



UNIVERSIDAD DE SALAMANCA

E. T. S. DE INGENIERÍA INDUSTRIAL DE BÉJAR

**DISEÑO Y CONTROL DE UNA ETAPA
DE POTENCIA DEL ROBOT ACMA X58
CON UN MICROCONTROLADOR Y EL
PROGRAMA LABVIEW**

Valeria Ospina Bohórquez
Béjar, julio de 2015



UNIVERSIDAD DE SALAMANCA

E. T. S. DE INGENIERÍA INDUSTRIAL DE BÉJAR

**DISEÑO Y CONTROL DE UNA ETAPA
DE POTENCIA DEL ROBOT ACMA X58
CON UN MICROCONTROLADOR Y EL
PROGRAMA LABVIEW**

Valeria Ospina Bohórquez
Béjar, julio de 2015

D^a Valeria Ospina Bohórquez autora de este proyecto titulado “DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT ACMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW” comparte su autoría con su tutor D^o José Torreblanca González para que pueda utilizarlo en continuas revisiones del mismo y poder desarrollar mejoras en cuanto a programación del microcontrolador y control de la muñeca.

Fdo: D^a Valeria Ospina Bohórquez

A mi tío Fidel por su inteligencia y valentía.

Índice

ÍNDICE.

MEMORIA.....	19
CAPÍTULO I INTRODUCCIÓN.	21
Sección I Antecedentes	23
Sección II Objetivos.....	25
Sección III Presentación de la memoria	27
CAPÍTULO II. MUÑECA DEL ROBOT ACMA X58	31
SECCIÓN I Mecanismo de la muñeca	33
1.- Introducción	33
2.- Definición de robot industrial.....	33
3.- Muñeca del robot	34
3.1.- Estructura mecánica	36
3.2.- Transmisiones	37
SECCIÓN II Actuadores	41
1.- Introducción	41
2.- Tipos de actuadores	41
2.1.- Actuadores neumáticos	41
2.2.- Actuadores hidráulicos	42
2.3.- Actuadores eléctricos	43
2.3.1.- Motores de corriente alterna	43
2.3.2.- Motores paso a paso.....	44
2.3.3.- Motores de corriente continua	46
2.3.3.1.- Frenos electromecánicos.....	53
SECCIÓN III Sensores	57
1.- Introducción	57
2.- Descripción de los sensores	58
2.1.- Tacómetro.....	58
2.1.1.- Tipos de tacómetros	58
2.1.2.- Tacómetro eléctrico de los motores de la muñeca	62
2.2.- Sensor de corriente	64
2.2.1.- Tipos de sensores de corriente	65
2.2.2.- Bobina de Rogowski. Sensor de corriente de los motores de la muñeca	66
2.3.- Codificadores ópticos o encoder.....	69
2.3.1.- Codificadores específicos de los ejes de la muñeca	70
2.4.- Sensores inductivos	73
2.4.1.- Tipos de sensores inductivos.....	78
2.4.2.- Sensores inductivos específicos de los ejes de la muñeca	79
2.5.- Sensores finales de carrera.....	81
CAPÍTULO III. ETAPA DE POTENCIA	85

SECCIÓN I Introducción.....	87
SECCIÓN II Alimentación.....	91
1.- Alimentación monofásica.....	91
1.1.- Transformador monofásico.....	91
1.2.- Rectificador monofásico de doble onda en puente.....	92
2.- Alimentación de los motores.....	95
2.1.- Contactor trifásico.....	95
2.2.- Transformador trifásico.....	96
2.3.- Rectificador trifásico de onda completa.....	100
2.3.1.- Resistencia de descarga del filtro capacitivo.....	102
3.- Control de la alimentación a través de relés.....	102
3.1.- Red Snubber.....	104
SECCIÓN III Placas electrónicas de control originales.....	105
1.-Introducción.....	105
2.-Placas de control.....	106
2.1.- Plano N°1.....	107
2.2.- Plano N°2.....	108
2.3.- Plano N°3.....	109
2.4.- Plano N°4.....	110
2.5.- Plano N°5.....	111
2.6.- Plano N°6.....	112
2.7.- Plano N°7.....	113
2.8.- Plano N°8.....	114
2.9.- Plano N°9.....	115
2.10.- Plano N°10.....	116
2.11.- Plano N°11.....	117
2.12.- Plano N°12.....	118
3.-Implementación física de las placas de control.....	119
3.1.- Circuito de los transformadores de impulsos.....	119
3.2.- Placa del puente en H.....	120
CAPÍTULO IV. PLACA DE CONTROL DISEÑADA. PLACAS AUXILIARES.....	123
SECCIÓN I Proceso de fabricación de la placa diseñada.....	125
1.- Introducción.....	125
2.- Proceso de fabricación.....	127
2.1.- Diseño de la placa en AutoCAD.....	127
2.2.- Fresado.....	130
2.2.1.- Programas de asistencia para el fresado. Win PC y VCarve Pro.....	130
2.2.2.- Proceso de fresado.....	131
2.3.- Soldaduras.....	141
2.3.1.- Soldaduras puente.....	141
2.3.2.- Soldaduras de componentes.....	141
2.3.2.- Soldaduras de elementos aislados.....	141
SECCIÓN II Componentes de la placa diseñada.....	143
1.- Introducción.....	143
2.- Microcontrolador PIC16F877A.....	144
2.1.- Introducción.....	144
2.2.- Arquitectura, diagrama de conexiones y repertorio de instrucciones.....	147
2.2.1.- Procesador RISC con arquitectura Harvard.....	147

2.2.2.- Organización de la memoria de programa.	149
2.2.3.- Organización de la memoria de datos RAM.....	150
2.2.4.- Diagrama de conexionado.	152
2.2.5.- Repertorio de instrucciones.	154
2.3.- Principales registros de control y manejo de las memorias FLASH y EEPROM.	159
2.3.1.- El registro estado (STATUS).	159
2.3.2.- El registro de opciones (OPTION).	161
2.3.3.- Los registro para controlar las interrupciones.	162
2.3.3.1.- Registro de control de interrupciones (INTCON).	162
2.3.3.2.- Registro de permiso de interrupciones 1 (PIE1).	163
2.3.3.3.- Registro de permiso de interrupciones 2 (PIE2).	164
2.3.3.4.- Registro de los señalizadores de interrupciones 1 y 2. (PIR1 y PIR2).	165
2.3.3.5.- Lectura y escritura de las memorias EEPROM y FLASH.....	167
2.3.3.6.- Registro PCON.	169
2.4.- Las puertas de E/S y recursos especiales del procesador.	169
2.4.1.- Puerta A.	170
2.4.2.- Puerta B.	170
2.4.3.- Puerta C.	171
2.4.4.- Puerta D.	172
2.4.5.- Puerta E.	172
2.4.6.- Puerta paralela esclava.	174
2.4.7.- Palabra de configuración.	174
2.4.8.- Palabra de identificación.	175
2.4.9.- Reinicialización o reset.	175
2.4.10.- Perro guardián (WDT, Watchdog Timer).	177
2.4.11.- Modo de reposo o de bajo consumo.	178
2.4.12.- Programación de los PIC16F87X.	179
2.5.- Los temporizadores.	179
2.5.1.- Estructura interna y funcionamiento del TMR0.	180
2.5.2.- Estructura interna y funcionamiento del TMR1.	181
2.5.2.1.- Registro de control del TMR1 (T1CON).	181
2.5.3.- Funcionamiento y programación del TMR2.	183
2.6.- Módulos de captura, comparación y modulación de anchura de pulsos	185
2.6.1.- Modo captura.	186
2.6.2.- Modo comparación.	187
2.6.3.- Modo de modulación de anchura de pulsos (pwm).	188
2.7.- El convertor A/D.	191
2.7.1.- Registros de trabajo.	191
2.8.- Módulo de comunicación serie síncrona: MSSP.	197
2.8.1.- Modo SPI.	198
2.8.2.- Manejo y programación en el modo spi.	199
2.8.3.- Módulo mssp trabajando en modo i2c.	202
2.8.4.- Conceptos fundamentales sobre el bus I2C.	202
2.8.5.- Características generales del bus I2C.	204
2.8.6.- Direccionamiento.	210
2.8.7.- Especificaciones eléctricas.	211
2.8.8.- El bus I2C y los PIC16F87X.	213
2.8.9.- Funcionamiento del modo maestro.	218
2.9.- El USART: Transmisor/ Receptor, Síncrono/ Asíncrono serie.	221
2.9.1.- Generador de baudios.	223
2.9.2.- Transmisión asíncrona.	225
2.9.3.- Receptor asíncrono.	227
2.10.- El microcontrolador PIC16F877A dentro del sistema de control diseñado.	230
3.- Decodificador 74LS138.	232

3.1.- El decodificador 74LS138 dentro del sistema de control diseñado	233
4.- Inversor 40106	233
4.1.- El inversor 40106 dentro del sistema de control diseñado.....	233
5.- Báscula 74LS373.	234
5.1.- La báscula 74LS373 dentro del sistema de control diseñado.....	236
6.- Buffer bidireccional 74LS245.....	237
6.1.- El buffer bidireccional 74LS245 dentro del sistema de control diseñado.....	238
7.- Circuito 8259, controlador programable de interrupciones (PIC).....	238
7.1.- Interrupciones en sistemas microordenadores.....	239
7.2.- El 8259.	240
7.3.- Resumen del funcionamiento del 8259.	241
7.4.- Descripción de los terminales.....	242
7.5.- Descripción de los bloques del controlador.	244
7.6.- Secuencia de interrupción.	246
7.7.- Programación del 8259.....	248
7.8.- Modos de funcionamiento.	255
7.8.1.- Modo completamente encajado (“Fully nested mode”).....	255
7.8.2.- Modo fin de interrupción EOI.....	256
7.8.3.- Modo fin automático de interrupción AEOL.	257
7.8.4.- Modo de rotación automática, dispositivos de igual prioridad.	257
7.8.5.- Modo de rotación específica, prioridad específica.	258
7.8.6.- Modo de máscaras de interrupció.	258
7.8.7.- Modo de comando poll o sondeo.	259
7.8.8.- Modos de disparo por flanco y por nivel.	260
7.8.9.- Modo special fully nested.	260
7.8.10.- Modo buffered o amplificador.	261
7.8.11.- Modo cascada.....	262
7.8.12.- Lectura del estado del 8259.	263
7.9.- El circuito 8259 dentro del sistema de control diseñado.....	264
8.- Módulo LCD.	264
8.1.- Conexiones de un módulo LCD.	265
8.2.- Instrucciones del módulo LCD.	267
8.3.- Módulo LCD del proyecto.	269
8.3.1.- El módulo LCD dentro del sistema de control diseñado.....	269
SECCIÓN III Placas auxiliares	271
1.- Introducción	271
2.- Placa 8255 – 8253.....	271
2.1.- Diseño de la placa	271
2.2.- Componentes de la placa	274
2.2.1.- Circuito 8253. Periférico programable de contadores-temporizadores.	274
2.2.1.1.- Diagrama de conexionado.....	276
2.2.1.2.- Descripción y programación del circuito.....	277
2.2.1.3.- Descripción de los modos de funcionamiento.	279
2.2.1.4.- Procedimiento de lectura - escritura	282
2.2.1.5.- El circuito 8253 dentro del sistema de control diseñado.....	284
2.2.2.- Circuito 8255. Periférico programable de entrada-salida.	284
2.2.2.1.- Diagrama de conexionado.....	287
2.2.2.2.- Modos de Funcionamiento del circuito.	288
2.2.2.3.- Ejemplos de conexiones en modo comunicaciones.	297
2.2.2.4.- Estados de lectura del puerto C.....	300
2.2.2.5.- El circuito 8255 dentro del sistema de control diseñado.....	301
3.- Placa de gestión de motores.	302
3.1.- Diseño de la placa	302

3.2.- Componentes de la placa	304
3.2.1.- DAC0800. Convertidor digital – analógico.	304
3.2.1.1.- El DAC0800. Convertidor digital – analógico dentro del sistema de control diseñado.	307
3.2.2.- TL074. Circuito amplificador.	308
3.2.2.1.- El circuito amplificador TL074. dentro del sistema de control diseñado.	309
4.- Placa de tratamiento de señales.	309
4.1.- Componentes de la placa.	313
4.1.1.- TL082. Circuito amplificador.	313
4.1.2.- Optoacopladores.	313
5.- Placa AND – OR.	314
5.1.- Componentes de la placa.	315
CAPÍTULO V. SISTEMA DE CONTROL DISEÑADO.....	317
SECCIÓN I Introducción	319
1.- Objetivo del sistema de control diseñado.	319
2.- Subsistemas de control diseñados: velocidad y posición.	320
3.- Subsistema de control de la alimentación de la etapa de potencia.....	321
4.- Interfaz gráfica en Labview.	321
SECCIÓN II Subsistemas de control	323
1.- Subsistema de control de posición.	323
1.1.- Estudio y adaptación de las señales de los encoder.	324
1.2.- Planteamiento del sistema de contaje.	326
1.2.1.- Contaje por interrupciones utilizando el circuito 8259.	326
1.2.2.- Contaje utilizando los circuitos contadores 8253.	326
1.2.2.1. - Descripción del esquema básico de contaje.....	326
1.2.2.2. - Frecuencia de entrada y recarga de contadores.	327
1.2.2.3. - Placa AND - OR.	328
1.3.- Maniobra de búsqueda de la posición inicial.	329
1.3.1.- Estudio de los sensores inductivos.	329
1.3.2.- Programación de la maniobra.	329
1.3.3.- Problema de desconexión del microcontrolador.	330
1.3.4.- Problema de reconocimiento de la interrupción.	331
1.4.- Delimitación del espacio de trabajo. Sensores finales de carrera.	331
1.4.1.- Problema de reconocimiento de la interrupción.	331
1.5.- Programación de movimientos automáticos de la muñeca.	331
2.- Subsistema de control de velocidad.	332
2.1.- Fallo en la conexión del tacómetro generador de cc.	333
2.2.- Subsistema de control de la velocidad diseñado.	334
2.3.- Problemas presentados.	334
2.3.1.- Problema de detención de los motores.....	334
2.3.1.1.- Solución mediante la variación de la tensión de referencia del DAC0800.	335
2.3.1.2.- Solución mediante compensación de tensión por software.	336
2.3.2.- Problema de pérdida de control de los motores.	336
3.- Subsistema de control de la etapa de potencia.	336
SECCIÓN III Interfaz gráfica.	339
1.- Introducción	339
2.- Descripción	340
2.1.- Control de los parámetros de la comunicación serie.....	340
2.2.- Control de la velocidad de los motores.....	341
2.3.- Lectura y recarga de registros contadores.	342

2.4.- Botones de acciones específicas.....	343
3.- Programación en labview.....	344
3.1.- Programación en Labview: envío y recepción de datos mediante comunicación serie.....	345
3.1.1.- Funciones utilizadas.....	345
3.1.2.- Programación de una estructura completa de lectura y escritura.....	349
3.2.- Programación en Labview: estructura de control de la pantalla informativa.....	350
4.- MAX 232.....	351
CAPÍTULO VI PROGRAMACIÓN EN LENGUAJE ENSAMBLADOR.....	355
CAPÍTULO VII CONCLUSIONES.....	417
Sección I Conclusiones generales.....	419
Sección II Posibles líneas de continuación.....	421
BIBLIOGRAFÍA.....	423
ÍNDICE DE FIGURAS.....	427
PLANOS.....	435
ÍNDICE DE PLANOS.....	437
Plano N° 1.....	439
Plano N° 2.....	441
Plano N° 3.....	443
Plano N° 4.....	445
Plano N° 5.....	447
Plano N° 6.....	449
Plano N° 7.....	451
Plano N° 8.....	453
Plano N° 9.....	455
Plano N° 10.....	457
Plano N° 11.....	459
Plano N° 12.....	461
Plano N° 13.....	463

Plano N° 14465

Plano N° 15467

PRESUPUESTO469

PLIEGO DE CONDICIONES GENERALES.....477

- 1.- Objeto.....479
- 2.- Documentos.479
- 3.- Número de copias autorizadas.479
- 4.- Pliego de condiciones generales.....479
 - 4.1.- Diseño y fabricación de la placa.479
 - 4.2.- Etapa de potencia.....480
 - 4.3.- Interfaz gráfica.480
 - 4.4.- Muñeca del robot ACMA X58.481
 - 4.5.- Pasos a seguir para controlar la muñeca del robot desde la interfaz gráfica.481

MEMORIA

Capítulo I

Introducción

Sección I

ANTECEDENTES

El presente trabajo de fin de grado surgió de la idea de aprovechar y mantener partes del robot ACMA X58 en desuso almacenadas en el laboratorio de tecnología electrónica. El robot ACMA X58 fue donado a la Escuela por la fábrica de vehículos Renault de Palencia en 1999 y estuvo funcionando hasta que dejó de hacerlo por motivos relacionados con las conexiones eléctricas.

El robot fue objeto de varias líneas de investigación desde el año 2001 hasta el 2003. En un momento se decidió que el robot debía de desmantelarse para ocupar el espacio en el desarrollo de otra actividad de la Escuela. Varias partes del robot fueron recicladas para servir de material de prácticas en el laboratorio de tecnología electrónica. Tanto la etapa de potencia como su sistema de alimentación se almacenaron en el laboratorio así como los dos ejes finales del robot que componen la muñeca del mismo. También se guardan dos de los motores de otros ejes del robot y una parte de la etapa de control antigua.

La realización de este trabajo de fin de grado se centró en la etapa de potencia y su sistema de alimentación y en los ejes finales (5 y 6) del robot, que componen la muñeca del mismo. El sistema de control diseñado suple las carencias que ocasionan las partes faltantes del sistema. El diseño integra el material del robot antiguo mencionado junto con el diseño e implementación de un control electrónico hecho a medida.

La muñeca del robot y su etapa de potencia estaban en estado de desuso en el laboratorio en el momento que se decidió realizar este trabajo de fin de grado. De ahí que las mejoras conseguidas a través de su realización constituyen la oportunidad de obtener nuevo material de prácticas para el laboratorio de tecnología electrónica. Además, se abre la posibilidad de indagar en líneas de continuación del trabajo que realicen mejoras sobre el sistema diseñado.

Sección II

OBJETIVOS

El objetivo de este trabajo fin de grado es diseñar e implementar un sistema de control a través de un microcontrolador que gobierne los movimientos de los dos ejes que conforman la muñeca del robot. Para lograr este objetivo se plantearon una serie de objetivos parciales que a continuación se detallan.

En primer lugar, se planteó como objetivo la puesta a punto de todos los elementos de la etapa de potencia y de la muñeca de robot. El correcto funcionamiento de estos elementos es fundamental para realizar el diseño de un sistema de control fiable y amoldado a las necesidades físicas reales sistema de controlar.

El segundo objetivo a conseguir fue el diseño y la fabricación de la placa de circuito impreso principal en la que se implementó el sistema de control diseñado. El diseño de la placa de control debió basarse en las necesidades del sistema “huérfano” heredado del robot ACMA X58. Fue claro desde un principio la necesidad fundamental de controlar los parámetros de velocidad y posición de los ejes. Para ello el sistema diseño debía de contar con subsistemas de control de cada una de estas dos magnitudes y una forma de interacción entre ellas y con el usuario. Además, se debió de establecer una unión entre el control electrónico original de la etapa de potencia y el sistema de control nuevo.

Una vez fabricada la placa de control principal se debió de probar el funcionamiento correcto de todas las características diseñadas. No todas las funciones ideadas en el momento de diseño de la placa tuvieron una implementación física posible, debido tanto a problemas de malas conexiones eléctricas como al desconocimiento de ciertos parámetros de funcionamiento de algunos circuitos impresos, conocimiento práctico que solo se adquiere tras sucesivos procesos de prueba y error. Dentro de la puesta a punto de todos los subsistemas de control ideados surgió la necesidad de aumentar las prestaciones del sistema de control diseñado. Varias placas auxiliares se añadieron al sistema, algunas se fabricaron específicamente para este trabajo mientras que otras se reciclaron de trabajos anteriores.

El cuarto objetivo parcial fue la interconexión física del sistema de control diseñado con la muñeca del robot. Todos los parámetros eléctricos que se controlaban con la placa y que se verificaron haciendo uso de instrumentos como los osciloscopios o los polímetros debieron de pasar de la teoría a la práctica. Como se esperaba, surgieron numerosos problemas derivados en su mayoría de la falta de aislamiento eléctrico entre la etapa de potencia y la de control y a errores de contaje de los pulsos de los encoders. Los microcontroladores son muy sensibles a picos de corriente, muy frecuentes en el arranque de los motores de corriente continua de los ejes de la muñeca del robot y la señal de los encoders poseía mucho ruido. Para cumplir con este objetivo debieron de idearse soluciones a la vez que surgían los problemas, proceso que retardó el trabajo considerablemente.

El quinto objetivo fue la programación de maniobras automáticas de movimiento de la muñeca en las que se demuestra el grado de control de los ejes que se posee. La idea es demostrar el rango de movimiento de la muñeca y hacerse a la idea de las numerosas tareas que podría desempeñar a nivel industrial.

Finalmente, tras realizarse una interconexión correcta del sistema de control y la muñeca del robot sin la ocurrencia de fallos eléctricos se planteó el objetivo final de programar una interfaz gráfica de control en Labview. Esta interfaz es el puente de unión entre el usuario y el sistema de control diseñado, permitiendo el gobierno de la muñeca del robot de una manera segura, clara e intuitiva.

Sección III

PRESENTACIÓN DE LA MEMORIA

La memoria del presente trabajo de fin de grado se encuentra estructurada en seis capítulos divididos en secciones para facilitar su comprensión y la estructuración del documento. A continuación se presentan cada uno de los capítulos.

El primer capítulo es la introducción del trabajo y se divide en tres secciones: antecedentes, objetivos y presentación de la memoria. En él se explica el origen de la muñeca de robot ACMA X58 utilizada en este trabajo y su trayectoria hasta que se decidió acometerlo. A su vez, se detallan los objetivos que se pretende cumplir a lo largo de la realización del trabajo.

El segundo capítulo se titula “Muñeca del robot ACMA X58” y está dividido en tres secciones: Mecanismo de la muñeca, Actuadores y Sensores. En él se introduce todos los componentes físicos que componen la muñeca del robot detallando el principio teórico de los elementos a la par que se profundiza en el sistema específico de la muñeca. La primera sección llamada “Mecanismo de la muñeca” se encarga de describir completamente el funcionamiento mecánico de dos ejes que componen esta extremidad del robot haciendo énfasis en el sistema de reductores que posee cada uno de los ejes. La sección de “Actuadores” realiza un repaso de los principales tipos de actuadores presentes en la industria robótica para posteriormente centrarse en los motores de corriente continua, que son los actuadores de la muñeca objeto de este trabajo. Se describe su principio de funcionamiento y la estrategia de control por variación de tensión del inducido, que es la implementada en el sistema de control de los motores. Por último, se explica el funcionamiento de los frenos electromecánicos, mecanismo fundamental para el funcionamiento de los motores.

El capítulo número tres se titula “Etapas de potencia” y en él se explican en detalle todos los componentes de la etapa de potencia del robot original. El capítulo se divide en tres secciones: Introducción, Alimentación y Placas electrónicas de control originales. La sección titulada “Alimentación” describe los dos esquemas de alimentación eléctrica que posee la etapa de potencia: uno para la electrónica de control y otro para la alimentación de los motores. La última parte de esta sección detalla el sistema de control de la alimentación

a través de relés implementado. La última sección de este capítulo titulada “Placas electrónicas de control originales” describe el circuito electrónico de control original de la etapa de potencia. La descripción de las placas de control se basa en la estructuración del circuito de control en doce planos en los manuales originales del robot. Cada uno de estos planos contiene una parte diferenciada del circuito que se explica en cada uno de los puntos. Por último, se detalla la implementación física de las placas de control cuyo circuito se describió en el punto anterior.

Como puede observarse, los capítulos II y III del presente trabajo realizan un recorrido teórico por todos los elementos preexistentes dentro del sistema de control diseñado. Por un lado, se explica en detalle el funcionamiento mecánico de los ejes de la muñeca. Por otro, el esquema de alimentación de la etapa de potencia y todo el circuito de control original de la misma. De esta forma, se pretende informar al lector de todo el sistema físico del que se partió para posteriormente explicar el diseño de control haciendo referencia a conceptos ya explicados.

El cuarto capítulo se titula “Placa de control diseñada. Placas auxiliares” y se divide en tres secciones: Proceso de fabricación de la placa diseñada, Componentes de la placa diseñada y Placas auxiliares. A lo largo de este capítulo se detalla el proceso de diseño, fabricación e implementación física de la placa de circuito impreso diseñada para controlar el sistema detallado en los capítulos II y III. Así, la sección I explica el proceso de fabricación de la placa diseñada centrándose en el diseño de la placa en AutoCAD, el fresado de la placa y el soldado de los componentes. La sección II describe individualmente cada uno de los componentes electrónicos utilizados para finalizar en cada caso con una introducción al papel desempeñado por cada uno dentro del sistema de control. Al realizar esta primera introducción al sistema de control se pretende dar un primer vistazo al esquema diseñado, explicando parte por parte para luego poder explicar con más facilidad la idea del funcionamiento global del control diseñado, que se detalla en el siguiente capítulo. Por último, la tercera sección llamada “Placas auxiliares” enumera las cuatro placas añadidas a la placa principal en el marco de la implementación del sistema de control. La implementación física de la placa de control original obligó a ampliar las prestaciones de control presupuestadas originalmente. Partiendo de esta idea se incluyeron dentro del sistema de control cuatro placas extra cuya morfología se detalla en esta sección. De la misma forma, se realiza una introducción al papel que juegan dentro del control diseñado.

El quinto capítulo llamado “Sistema de control diseñado” es la unión de todos los capítulos anteriormente presentados. En él se explica en detalle el esquema de control diseñado apoyándose en las bases mecánicas del sistema a controlar, eléctricas de la alimentación que necesita y en la implementación física de las placas de control. Se divide en tres secciones: Introducción, Subsistemas de control e Interfaz gráfica.

La sección II titulada “Subsistemas de control” divide el sistema de control diseñado en tres subsistemas: posición velocidad y control de la etapa de potencia. Los subsistemas de posición, velocidad, y control de la alimentación de la etapa de potencia constituyen el control de los ejes del robot. En esta sección se explica cada uno siguiendo el orden de eventos cronológico real en el que se realizaron. Se empieza describiendo los problemas de puesta a punto del sistema original a resolver para luego continuar con los diversos

planteamientos de control y centrarse finalmente en la explicación detallada del sistema de control elegido para cada caso. De esta forma, logra darse una idea clara no sólo de la teoría detrás del sistema de control elegido sino también de todos los procesos de prueba y error llevados a cabo para definirlo y de todos los problemas encontrados a lo largo del proceso, junto con las soluciones implementadas en cada caso.

La sección III llamada “Interfaz gráfica” se centra en el proceso de diseño, desarrollo y programación de la interfaz gráfica de control del sistema descrito en la sección anterior. En primer lugar, se realiza una descripción de los diversos módulos que componen la pantalla de la interfaz asociándolos al sistema de control diseñado. Posteriormente y puesto que esta interfaz se programó en Labview se hace especial énfasis a las estructuras de programación gráfica utilizadas. Se describe detalladamente el proceso de comunicación serie entre el ordenador y el microcontrolador y la programación de la estructura de control de la pantalla informativa de la interfaz. Este capítulo constituye por lo tanto el culmen de este trabajo, en el que todos los elementos descritos en los capítulos anteriores se ven envueltos en un sistema de control lógico en el que las magnitudes de posición y velocidad de los ejes de la muñeca logran ser controladas y manipuladas a través de una interfaz gráfica.

El sexto capítulo de este trabajo se titula “Programación en lenguaje ensamblador”. En él se incluye todo el código en lenguaje ensamblador programado en el microcontrolador.

Finalmente, el séptimo y último capítulo son las “Conclusiones”. Está dividido en dos secciones: Conclusiones generales y Posibles líneas de continuación. La sección titulada “Conclusiones generales” describe en detalle el desarrollo de todos los objetivos parciales que se plantearon al inicio del trabajo partiendo del objetivo principal que es el gobernar la muñeca del robot mediante el diseño de un nuevo sistema de control. Así, se realiza un recorrido por todo el trabajo realizado para luego centrarse en el sistema conseguido y en todas las funcionalidades que posee. De la misma forma, se describen los objetivos no conseguidos y da una explicación del por qué, englobándolos dentro de todo el proceso de implementación del sistema y de los numerosos problemas que se tuvo. La segunda sección de este capítulo, “Posibles líneas de continuación” enumera cuatro posibles líneas de continuación del trabajo realizado. Algunas de estas líneas de continuación constituyen ideas que han surgido a lo largo de estos meses de trabajo, en los que se ha recopilado gran cantidad de información sobre el funcionamiento de la muñeca del robot y de su etapa de potencia. Otras nacen de objetivos no alcanzados en la realización de este trabajo debido a la necesidad de solucionar numerosos problemas que se presentaron. Se espera que estas líneas de continuación propuestas se constituyan en trabajos individuales en un futuro.

Capítulo II

Muñeca del robot

ACMA X58

Sección I

MECANISMO DE LA MUÑECA

1.- INTRODUCCIÓN

El robot industrial ACMA X58 fue un robot de gama alta particularmente destinado a la soldadura por puntos hacia finales del S XX. En un principio el robot se componía de seis ejes, este trabajo se centra en el funcionamiento de los ejes 5 y 6 del robot que componen la muñeca del mismo. El eje 5 posee un rango de 180° de giro alrededor de su eje mientras que el eje 6 posee un movimiento de rotación de 360° .

A lo largo de esta sección se desarrollarán los elementos mecánicos de diversas unidades funcionales de la muñeca del robot. Específicamente se tratará la estructura mecánica y el sistema de transmisiones. De esta forma, se inicia todo el proceso de comprensión del funcionamiento del sistema objeto de este trabajo. A lo largo de este capítulo se complementará el conocimiento mecánico de la muñeca con el conocimiento del sistema de accionamiento de los ejes, de la alimentación y del sistema sensorial.

2.- DEFINICIÓN DE ROBOT INDUSTRIAL

La definición de robot industrial más comúnmente aceptada es posiblemente la de la Asociación de Industrias Robóticas (RIA) según la cual un robot industrial es un “manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar trayectorias diferentes”.

Esta definición, ligeramente modificada, ha sido adoptada por la Organización Internacional de Estándares (ISO) que define al robot industrial como un “manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas”. Se incluye en esta definición la necesidad de que el robot tenga varios grados de libertad.

Una definición más completa es la establecida por la Asociación Francesa de Normalización (AFNOR), que define primero el manipulador y, basándose en dicha definición, el robot.

- Manipulador: mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.
- Robot: manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectoria variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.

Por último, la Federación Internacional de Robótica (IFR, "International Federation of Robotics") distingue entre robot industrial de manipulación y otros robots.

"Por robot industrial de manipulación se entiende una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento".

En esta definición se debe entender que la reprogramabilidad y la multifunción se consiguen sin modificaciones físicas del robot.

Es común en todas las definiciones anteriores la aceptación del robot industrial como un brazo mecánico con capacidad de manipulación y que incorpora un control más o menos complejo. Un sistema robotizado, en cambio, es un concepto más amplio. Engloba todos aquellos dispositivos que realizan tareas de forma automática en sustitución de un ser humano y que pueden incorporar o no a uno o varios robots, siendo esto último lo más frecuente.

3.- MUÑECA DEL ROBOT

Como se explicó en la introducción, la parte del robot con la que se trabaja en el presente trabajo es la muñeca. La muñeca del robot está constituida por los ejes 5 y 6 del robot original que se pueden observar en la figura inferior. El eje 5 corresponde al eje con movimiento de giro de 180° y el eje 6 posee un movimiento de rotación de 360°.

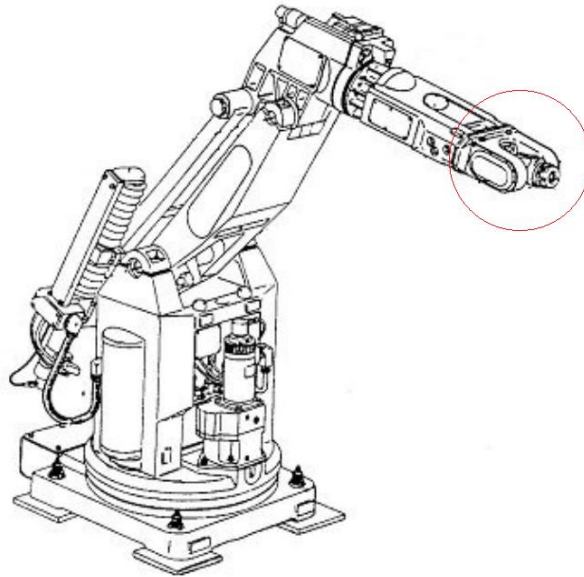


Ilustración 1 - Muñeca del robot

La muñeca es por lo tanto un subconjunto separable del robot que agrupa los mecanismos de los dos ejes finales. La base de la chapa constituye la cara de fijación de los dos motores reductores sobre la muñeca y de la unión de la muñeca sobre el brazo. En la imagen inferior se observa la muñeca del robot.



Ilustración 2 - Muñeca del robot ACMA X58

3.1.- ESTRUCTURA MECÁNICA

La constitución física de la mayor parte de los robots guarda cierta similitud con la anatomía de las extremidades superiores del cuerpo humano. De ahí que en ocasiones se haga referencia a ciertos elementos que componen el robot como cintura, brazo, codo, hombro o muñeca.

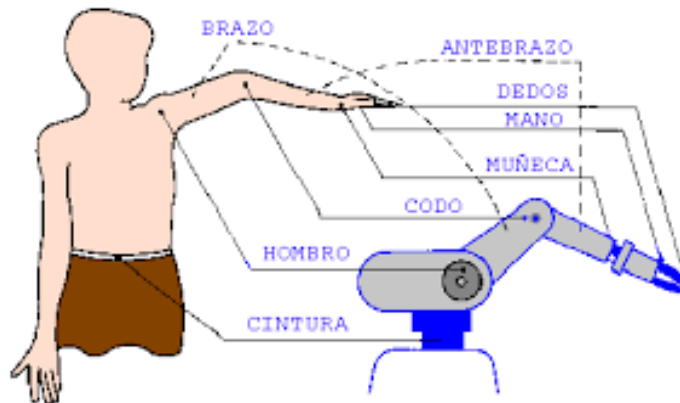


Ilustración 3 - Semejanza entre las partes de un robot y la anatomía humana

El manipulador es el elemento central de la mecánica del robot. Está formado por una serie de elementos estructurales sólidos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos. En este caso sólo se cuenta con la muñeca del robot. Cada articulación provee de un grado de libertad al manipulador. Se define grado de libertad como cada uno de los movimientos independientes (giros y desplazamientos) que puede realizar cada articulación con respecto de la anterior. El número de grados de libertad del robot viene dado por la suma de los grados de libertad de las articulaciones que lo componen, De esta forma pueden realizarse movimientos ya sean de tipo lineal o angulares. Existen dos tipos de articulación usados en las juntas del manipulador: prismática/ lineal y rotacional. En este caso en específico, ambos ejes o eslabones realizan movimientos angulares y poseen articulaciones de tipo rotacional.

El actuador final (también conocido como “gripper”) es un dispositivo que se une a la muñeca del brazo del robot con la finalidad de activarlo para la realización de una tarea específica. Se dividen dos grandes grupos: pinzas y herramientas. Se denomina Punto de Centro de Herramienta (TCP, “Tool Center Point”) al punto focal de la pieza o herramienta. Como el uso original del robot era el de la soldadura por arco el TCP podría haber estado en la punta de la antorcha de soldadura.

Las dimensiones de los elementos del manipulador, junto con los grados de libertad, definen la zona de trabajo del robot, referida únicamente al espacio dentro del cual pueda desplazarse el extremo de la muñeca. En este caso la zona de trabajo la conforma el espacio que llega a ocupar al girar el eje 5 girando 180° desde una de las posiciones horizontales hasta la otra. Para determinar el volumen de trabajo no se toma en cuenta el actuador final. Es importante definir los límites de movimiento de cada una de las articulaciones. Para ello se instalaron sensores finales de carrera en las posiciones extremas del eje 5 que se explicarán en la última sección de este capítulo.

Dentro del marco del sistema de control de la muñeca se estableció un lado A de referencia para poder describir los movimientos a derecha o a izquierda del eje 5. En la imagen inferior se observa la etiqueta del lado A de la muñeca.

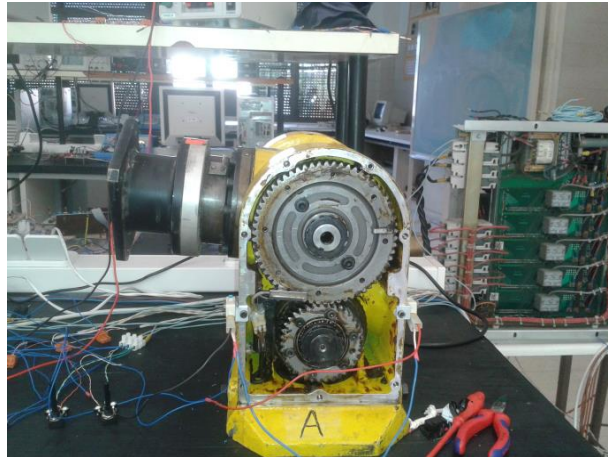


Ilustración 4 - Etiqueta del lado A convenido de la muñeca del robot.

3.2.- TRANSMISIONES

Las transmisiones son los elementos encargados de transmitir el movimiento desde los actuadores hasta las articulaciones. Se incluyen dentro de las transmisiones los reductores, encargados de adaptar el par y la velocidad a la salida del actuador a los valores adecuados para el movimiento de los elementos del robot.

Dado que un robot mueve su extremo con aceleraciones elevadas es de gran importancia reducir al máximo su momento de inercia. Del mismo modo, los pares estáticos que deben vencer los actuadores dependen directamente de la distancia de las masas al actuador. Por estos motivos se procura que los actuadores, por lo general pesados, estén lo más cerca posible de la base del robot. Esta circunstancia obliga a utilizar sistemas de transmisión que trasladen el movimiento hasta las articulaciones, especialmente a las situadas en el extremo del robot. Es de esperar que un buen sistema de transmisión cumple con una serie de características básicas: debe de tener un tamaño y peso reducido, ha de evitarse que presente juego u holguras considerables y deben buscarse transmisiones con gran rendimiento. Las transmisiones más habituales son aquellas que cuentan con un movimiento circular tanto a la salida como a la entrada. Incluidas en éstas se encuentran los engranajes, las correas dentadas y las cadenas.

Los reductores utilizados en robótica se les exigen unas condiciones de funcionamiento muy restrictivas. La exigencia de estas características viene motivada por las altas prestaciones que se le piden al robot en cuanto a precisión y velocidad de posicionamiento. Se buscan reductores de bajo peso, tamaño reducido, bajo rozamiento y que al mismo tiempo sean capaces de realizar una reducción elevada de velocidad en un único paso. Se tiende también a minimizar su movimiento de inercia, de negativa influencia en el funcionamiento del motor. Los reductores, por motivos de diseño, tienen una velocidad máxima admisible, que como regla general aumenta a medida que disminuye el tamaño de motor. A su vez, existe una limitación en cuanto al par de salida nominal

permisible (T_2) que depende del par de entrada (T_1) y de la relación de transmisión a través de la expresión:

$$T_2 = \eta T_1 (w_1/w_2)$$

η es el rendimiento que puede llegar a ser cerca del 100% y la relación de reducción de velocidades w_1 (velocidad de entrada) y w_2 (velocidad de salida) varía entre 50 y 300.

Puesto que los robots trabajan en ciclos cortos, que implican ciclos continuos de arranques y paradas, es de gran importancia que el reductor sea capaz de soportar pares elevados puntuales. También se busca que el juego angular sea lo menor posible. Éste se define como el ángulo que gira el eje de salida cuando se cambia su sentido de giro sin que llegue a girar el eje de entrada. Por último, es importante que los reductores para robótica posean una rigidez torsional, definida como el par que hay que aplicar sobre el eje de salida para que, manteniendo bloqueado el de entrada, aquél gire un ángulo unidad.

La muñeca del robot ACMA X58 posee un sistema importante de transmisiones y reductores que a continuación se describe.

Las cadenas cinemáticas están constituidas por:

- Una pareja de engranajes cónicos cuyo piñón está cortado dentro del árbol de salida del reductor.
- Una pareja de engranajes cilíndricos helicoidales cuyo piñón es solidario a la rueda cónica hacia arriba y la rueda respectivamente al árbol terminal del eje 5 y al piñón cónico del eje 6.

En las imágenes inferiores se observan esquemas de la cadena cinemática de la muñeca.

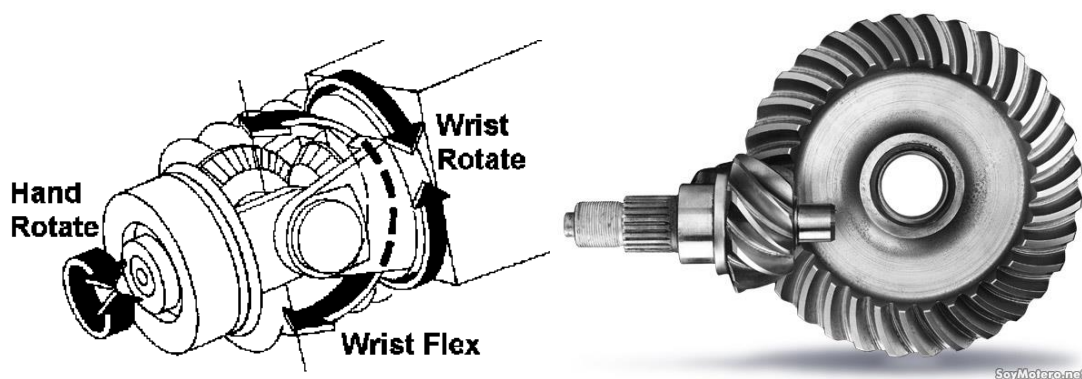


Ilustración 5 - Esquema cadena cinemática muñeca

A continuación se detallan algunas de las características más importantes de los elementos que componen las cadenas cinemáticas mencionadas.

- Eje 5

La velocidad máxima del motor es de 3000 rev./min, el eje 5 tiene un reductor planetario de 1/25. . La relación de piñones del eje es de 20/48 y la reducción total del eje es de 100.



Ilustración 6 - Reductor de eje tipo planetario

- Eje 6

El eje 6 está compuesto por un reductor de trenes paralelos de 1/10. La relación de los piñones es de 37/62. La señal cónica del eje 6 es de 22/35 y la reducción total del eje es de 64. Debido a su disposición el eje 6 sufre una interacción debido a los movimientos del eje 5 que debe ser compensada por el sistema de control.

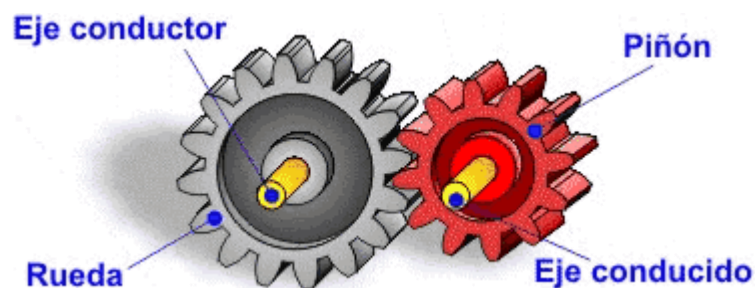


Ilustración 7 - Esquema básico de engranajes

Sección II

ACTUADORES

1.- INTRODUCCIÓN

Los actuadores tienen como misión generar el movimiento de los elementos del robot según las órdenes dadas por la unidad de control. Se clasifican en tres grandes grupos, según la energía que utilizan:

- Neumáticos
- Hidráulicos
- Eléctricos

2.- TIPOS DE ACTUADORES

2.1.- ACTUADORES NEUMÁTICOS

Los actuadores neumáticos utilizan el aire comprimido como fuente de energía y son muy indicados en el control de movimientos rápidos, pero de precisión limitada. Los motores hidráulicos son recomendables en los manipuladores que tienen una gran capacidad de carga, junto a una precisa regulación de velocidad. Los motores eléctricos son los más utilizados, por su fácil y preciso control, así como por otras propiedades ventajosas que establece su funcionamiento, como consecuencia del empleo de la energía eléctrica. Como la muñeca del robot posee únicamente actuadores de tipo eléctrico la descripción se centrará en estos.

Los actuadores neumáticos son aquellos en los que la fuente de energía es aire a presión entre 5 y 10 bar. Existen dos tipos de actuadores neumáticos: cilindros neumáticos y motores neumáticos (de aletas rotativas o de pistones axiales). En los primeros se consigue el desplazamiento de un émbolo encerrado en un cilindro como consecuencia de la diferencia de presión a ambos lados de aquél. Los cilindros neumáticos pueden ser de simple o de doble efecto. En los de simple efecto el émbolo se desplaza en un sentido como resultado del empuje ejercido por el aire a presión, mientras que en el otro sentido se desplaza como consecuencia del efecto de un muelle (que recupera al émbolo a su posición en reposo). En los cilindros de doble efecto el aire a presión es el encargado de empujar el

émbolo en las dos direcciones, al poder ser introducido de forma arbitraria en cualquiera de las dos cámaras.

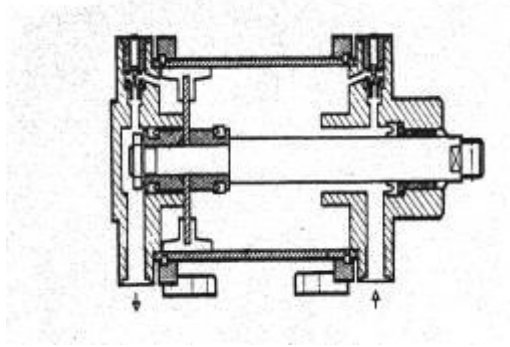


Ilustración 8 - Cilindro neumático de doble efecto

Normalmente, con los cilindros neumáticos sólo se persigue un posicionamiento en los extremos del mismo y no un posicionamiento continuo. Esto último se puede conseguir con una válvula de distribución (generalmente de accionamiento directo) que canaliza el aire a presión hacia una de las dos caras del émbolo alternativamente. Existen, no obstante, sistemas de posicionamiento continuo de accionamiento neumático, aunque debido a su coste y calidad todavía no resultan competitivos.

En los motores neumáticos se consigue el movimiento de rotación de un eje mediante aire a presión. Los dos tipos más utilizados son los motores de aletas rotativas y los motores de pistones axiales. Los motores de pistones axiales tienen un eje de giro solidario a un tambor que se ve obligado a girar ante las fuerzas que ejercen varios cilindros, que se apoyan sobre un plano inclinado. Otro método común más sencillo de obtener movimientos de rotación a partir de actuadores neumáticos, se basa en el empleo de cilindros cuyo émbolo se encuentra acoplado a un sistema de piñón-cremallera.

En general y debido a la compresibilidad del aire, los actuadores neumáticos no consiguen una buena precisión de posicionamiento. Sin embargo, su sencillez y robustez hacen adecuado su uso en aquellos casos en los que sea suficiente un posicionamiento en dos situaciones diferentes (todo o nada). Por ejemplo, son utilizados en manipuladores sencillos, en apertura y cierre de pinzas. Siempre debe tenerse en cuenta que el empleo de un robot con algún tipo de accionamiento neumático deberá disponer de una instalación de aire comprimido, incluyendo: compresor, sistema de distribución (tuberías, electro válvulas), filtros, secadores, etc. No obstante, estas instalaciones neumáticas son frecuentes y existen en muchas de las fábricas donde se da cierto grado de automatización.

2.2.- ACTUADORES HIDRÁULICOS

Este tipo de actuadores no se diferencia mucho de los neumáticos. En ellos, en vez de aire se utilizan aceites minerales a una presión comprendida normalmente entre los 50 y 100 bar, llegándose en ocasiones a superar los 300 bar. Existen, como en el caso de los neumáticos, actuadores del tipo cilindro y del tipo motores de aletas y pistones.

Sin embargo, las características del fluido utilizado en los actuadores hidráulicos marcan ciertas diferencias con los neumáticos. En primer lugar, el grado de compresibilidad de los aceites usados es considerablemente menor al del aire, por lo que la precisión obtenida en este caso es mayor. Por motivos similares, es más fácil en ellos realizar un control continuo, pudiendo posicionar su eje en todo un intervalo de valores (haciendo uso del servocontrol) con notable precisión. Además, las elevadas presiones de trabajo, diez veces superiores a las de los actuadores neumáticos, permiten desarrollar elevadas fuerzas y pares.

Por otra parte, este tipo de actuadores presenta estabilidad frente a cargas estáticas. Esto indica que el actuador es capaz de soportar cargas, como el peso o una presión ejercida sobre una superficie, sin aporte de energía (para mover el embolo de un cilindro sería preciso vaciar este de aceite). También es destacable su elevada capacidad de carga y relación potencia-peso, así como sus características de auto lubricación y robustez.

Frente a estas ventajas existen ciertos inconvenientes. Por ejemplo, las elevadas presiones a las que se trabaja propician la existencia de fugas de aceite a lo largo de la instalación. Asimismo, esta instalación es más complicada que la necesaria para los actuadores neumáticos y mucho más que para los eléctricos, necesitando de equipos de filtrado de partículas, eliminación de aire, sistemas de refrigeración y unidades de control de distribución. Los accionamientos hidráulicos se usan con frecuencia en aquellos robots que deben manejar grandes cargas (de 70 a 205kg).

2.3.- ACTUADORES ELÉCTRICOS

Las características de control, sencillez y precisión de los accionamientos eléctricos han hecho que sean los más usados en los robots industriales actuales. Dentro de los actuadores eléctricos pueden distinguirse tres tipos diferentes:

- Motores de corriente alterna
- Motores paso a paso
- Motores de corriente continua. Servomotores

2.3.1.- MOTORES DE CORRIENTE ALTERNA

Este tipo de motores no ha tenido aplicación en robótica hasta hace unos años, debido fundamentalmente a la dificultad de su control. Sin embargo, las mejoras que se han introducido en las maquinas síncronas hacen que se presenten como un claro competidor de los motores de corriente continua. Esto se debe principalmente a tres factores: la construcción de los motores síncronos sin escobillas, el uso de convertidores estáticos que permiten variar la frecuencia (y así la velocidad de giro) con facilidad y precisión y el empleo de la microelectrónica, que permite una gran capacidad de control. Existen dos tipos fundamentales de motores de corriente alterna:

- Motores asíncronos
- Motores síncronos

Los motores asíncronos son probablemente los más sencillos y robustos de los motores eléctricos. El rotor está constituido por varias barras conductoras dispuestas

paralelamente el eje del motor y por dos anillos conductores en los extremos. El conjunto es similar a una jaula de ardilla y por eso se le denomina también motor de jaula de ardilla.

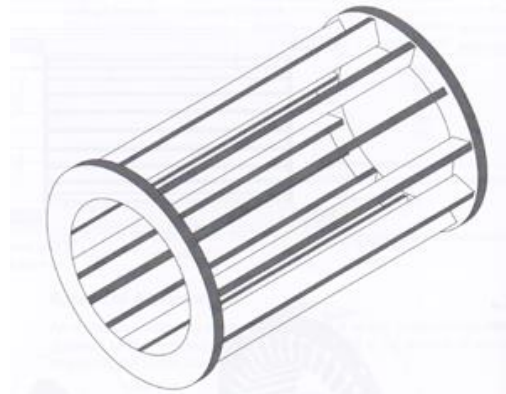


Ilustración 9 - Rotor de un motor asíncrono

El estator consta de un conjunto de bobinas, de modo que cuando la corriente alterna trifásica las atraviesa, se forma un campo magnético rotatorio en las proximidades del estator. Esto induce corriente en el rotor, que crea su propio campo magnético. La interacción entre ambos campos produce un par en el rotor. No existe conexión eléctrica directa entre estator y rotor. La frecuencia de la corriente alterna de la alimentación determina la velocidad a la cual rota el campo magnético del estator. El rotor sigue a este campo, girando más despacio. La diferencia de velocidades se denomina deslizamiento.

En el motor síncrono, como su nombre indica, el rotor opera exactamente a la misma velocidad que el campo del estator, sin deslizamiento. El inducido se sitúa en el rotor, que tiene polaridad constante (imanes permanentes o bobinas), mientras que el inductor situado en el estator está formado por tres devanados iguales decalados 120° eléctricos y se alimenta con un sistema trifásico de tensiones. En los motores síncronos la velocidad de giro depende únicamente de la frecuencia de la tensión que alimenta el inducido. Para poder variar esta precisión, el control de velocidad se realiza mediante un convertidor de frecuencia. Para evitar el riesgo de pérdida de sincronismo se utiliza un sensor de posición continuo que detecta la posición del rotor y permite mantener en todo momento el ángulo que forman los campos del estator y rotor. Este método de control se conoce como autosíncrono o autopilotado.

2.3.2.- MOTORES PASO A PASO

Los motores paso a paso generalmente no han sido considerados dentro de los accionamientos industriales, debido principalmente a que los pares para los que estaban disponibles eran muy pequeños y los pasos entre posiciones consecutivas eran grandes. En los últimos años se han mejorado notablemente sus características técnicas, especialmente en lo relativo a su control, lo que ha permitido fabricar motores paso a paso capaces de desarrollar pares suficientes en pequeños pasos para su uso como accionamientos industriales. Existen tres tipos de motores paso a paso:

- de imanes permanentes
- de reluctancia variable
- híbridos.

En los primeros, de imanes permanentes, el rotor, que posee una polarización magnética constante, gira para orientar sus polos de acuerdo al campo magnético creado por las fases del estator.

En los motores de reluctancia variable, el rotor está formado por un material ferromagnético que tiende a orientarse de modo que facilite el camino de las líneas de fuerza del campo magnético generado por las bobinas de estator. No contiene, por tanto, imanes permanentes. El estator es similar a un motor de corriente continua de escobillas. La reluctancia de un circuito magnético es el equivalente magnético a la resistencia de un circuito eléctrico. La reluctancia del circuito disminuye cuando el rotor se alinea con el polo del estator. Cuando el rotor está en línea con el estator el hueco entre el rotor y el estator es muy pequeño. En este momento la reluctancia está al mínimo. La inductancia del bobinado también varía cuando el rotor gira. Cuando el rotor está fuera de alineación, la inductancia es muy baja, y la corriente aumentará rápidamente. Cuando el rotor se alinea con el estator, la inductancia será muy grande. Esta es una de las dificultades del manejo de un motor de reluctancia variable.

Los motores híbridos combinan el modo de funcionamiento de los dos anteriores. En los motores paso a paso la señal de control consiste en trenes de pulsos que van actuando rotativamente sobre una serie de electroimanes dispuestos en el estator. Por cada pulso recibido, el rotor del motor gira un determinado número discreto de grados. Para conseguir el giro del rotor en un determinado número de grados, las bobinas del estator deben ser excitadas secuencialmente a una frecuencia que determina la velocidad de giro. Las inercias propias del arranque y parada (aumentadas por las fuerzas magnéticas en equilibrio que se dan cuando está parado) impiden que el rotor alcance la velocidad nominal instantáneamente, por lo que ésta, y por tanto la frecuencia de los pulsos que la fija, debe ser aumentada progresivamente. Para simplificar el control de estos motores existen circuitos especializados que a partir de tres señales (tren de pulsos, sentido de giro e inhibición) generan, a través de una etapa lógica, las secuencias de pulsos que un circuito de conmutación distribuye a cada fase. A continuación se muestran las configuraciones bipolar y unipolar respectivamente:

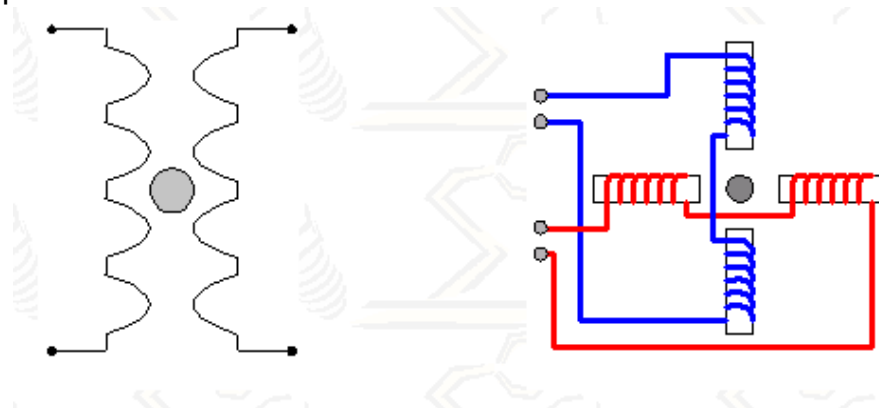


Ilustración 10 - Motor paso a paso. Configuración bipolar

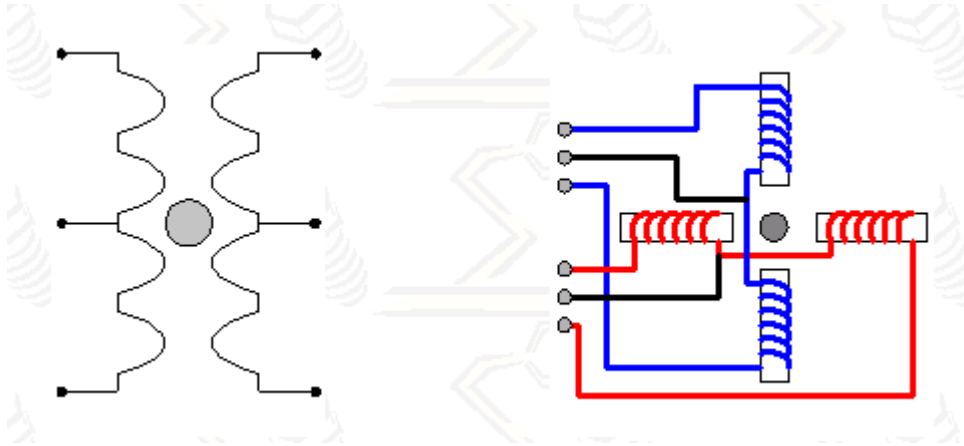
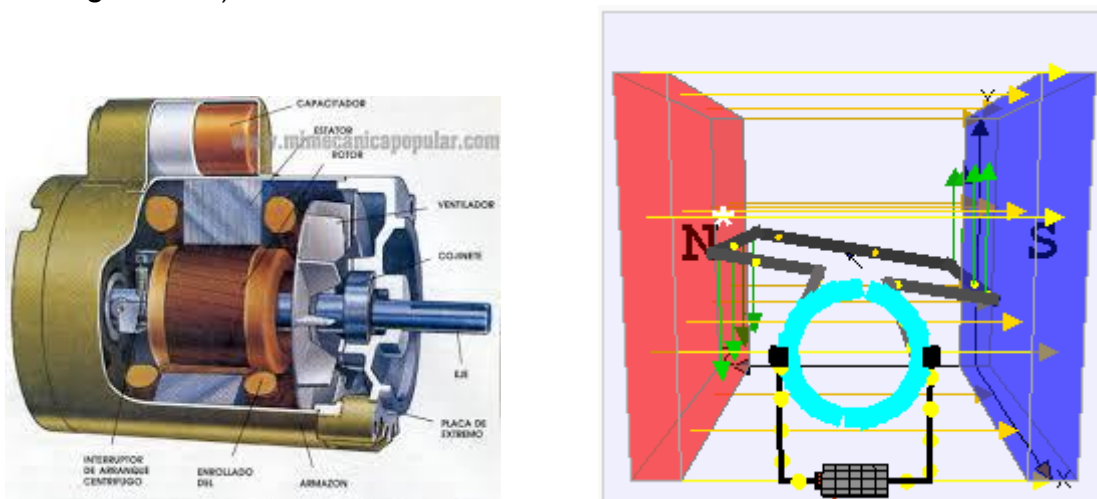


Ilustración 11 - Motor paso a paso. Configuración unipolar

2.3.3.- MOTORES DE CORRIENTE CONTINUA

El motor de corriente continua es una máquina que convierte la energía eléctrica en mecánica provocando un movimiento rotatorio gracias a la acción de un campo magnético. Una máquina de corriente continua se compone principalmente de dos partes. Por un lado, el estator es el encargado de dar soporte mecánico al aparato y contiene los imanes que crearán el campo magnético (en el caso de ser de imanes permanentes). Por otro, el rotor está formado por un devanado de cobre por el que se hace circular una intensidad.

Según la ley física de la Fuerza de Lorentz, cuando un conductor por el que pasa una corriente eléctrica se sumerge en un campo magnético el conductor sufre una fuerza perpendicular al plano formado por el campo magnético y la corriente que sigue la regla de la mano izquierda (puesto que la máquina de corriente continua funciona como motor y no como generador).



Los motores de corriente continua son accionados por una fuente de potencia de corriente continua. Hay cinco clases principales de motores de cc: de excitación separada, con excitación en derivación, de imán permanente, serie y compuesto.

Para que se pueda dar la conversión de energía eléctrica en energía mecánica de forma continua es necesario que los campos magnéticos del estator y del rotor permanezcan estáticos entre sí. Esta transformación es máxima cuando ambos campos se encuentran en cuadratura. El colector de delgas es un conmutador sincronizado con el rotor encargado de que se mantenga el ángulo relativo entre el campo del estator y el creado por las corrientes rotóricas. De esta forma se consigue transformar automáticamente, en función de la velocidad de la máquina, la corriente continua que alimenta al motor en corriente alterna de frecuencia variable en el inducido. Este tipo de funcionamiento se conoce con el nombre de autopilotado.

Al aumentar la tensión del inducido aumenta la velocidad de la máquina. Si el motor está alimentado a tensión constante, se puede aumentar la velocidad disminuyendo el flujo de excitación. Pero cuanto más débil sea el flujo, menor será el par motor que se puede desarrollar para una intensidad de inducido constante. Además, en los motores controlados por inducido se produce un efecto estabilizador de la velocidad de giro originado por la realimentación intrínseca que posee a través de la fuerza contraelectromotriz. Por estos motivos, de los dos tipos de motores de corriente continua es el controlado por inducido el que se usa en el accionamiento con robots.

Para mejorar el comportamiento de este tipo de motores, el campo de excitación se genera mediante imanes permanentes, con lo que se evitan fluctuaciones del mismo. Estos imanes son de aleaciones especiales como sumario-cobalto. Además, para disminuir la inercia que poseería un rotor bobinado, que es el inducido, se construye éste mediante una serie de espiras serigrafiadas en un disco plano, este tipo de rotor no posee apenas masa térmica, lo que aumenta los problemas de calentamiento por sobrecarga.

El motor de corriente continua presenta el inconveniente del obligado mantenimiento de las escobillas. Por otra parte, no es posible mantener el par con el rotor parado más de unos segundos, debido a los calentamientos que se producen en el colector. Para evitar estos problemas, se han desarrollado en los últimos años motores sin escobillas. En estos, los imanes de excitación se sitúan en el rotor y el devanado de inducido en el estator, con lo que es posible convertir la corriente mediante interruptores estáticos, que reciben la señal de conmutación a través de un detector de posición del rotor.

Las velocidades de rotación que se consiguen con estos motores son del orden de 1000 a 3000 rpm con un comportamiento muy lineal y bajas constantes de tiempo. Las potencias que pueden manejar pueden llegar a los 10KW. Como se ha indicado, los motores de corriente continua son controlados mediante referencias de velocidad. Éstas normalmente son seguidas mediante un bucle de retroalimentación de velocidad analógica que se cierra mediante una electrónica específica (accionador del motor). Se denominan entonces servomotores. Sobre este bucle de velocidad se coloca otro de posición, en el que las referencias son generadas por la unidad de control (microprocesador) sobre la base del error entre la posición deseada y la real.

Los dos motores de corriente continua encargados del movimiento de la muñeca del robot son de imanes permanentes. Este tipo de motores tienen algunas desventajas, como se dijo anteriormente, entre ellas que no pueden producir tan alta densidad de flujo magnético como la de un campo externo en derivación. En consecuencia el motor tendrá un par inducido por amperio de corriente del inducido menor que el de un motor en derivación del mismo tamaño y construcción. Además, este tipo de motores corren el riesgo de la desmagnetización. En una máquina de corriente continua la corriente del inducido produce su propio campo magnético. La fuerza magnetomotriz del inducido se resta de la de los polos bajo algunas zonas de las caras polares y se adiciona a la fuerza magnetomotriz de los polos en otras, reduciéndose el flujo neto total de la máquina. Éste es el efecto conocido como reacción del inducido. En un motor de imanes permanentes el flujo polar es justamente el residual de los imanes permanentes. Si la corriente del inducido llega a ser muy grande se corre el riesgo de que la fuerza magnetomotriz del inducido desmagnetice los polos, reduciendo y reorientando permanentemente el flujo residual en ellos. La desmagnetización también puede ser causada por calentamiento excesivo, que puede ocurrir durante periodos prolongados de sobrecarga.

En conclusión, un motor de imanes permanentes es básicamente la misma máquina que un motor de corriente continua en derivación, excepto que el flujo de un motor de imanes permanentes es fijo, por lo que no es posible controlar la velocidad del motor variando la corriente de campo. Así, los únicos métodos de control de velocidad disponibles para un motor de imanes permanentes son los de control de voltaje del inducido y control de la resistencia del inducido. Una vez hecha la analogía entre el motor de imanes permanentes y el motor con excitación en derivación se explicará el circuito equivalente del segundo, que salvando la diferencia mencionada será también el circuito equivalente para un motor de imanes permanentes.

Un motor con excitación en derivación es aquel cuyo circuito de campo obtiene su potencia directamente de los terminales del inducido del motor. El circuito equivalente de este motor es el siguiente:

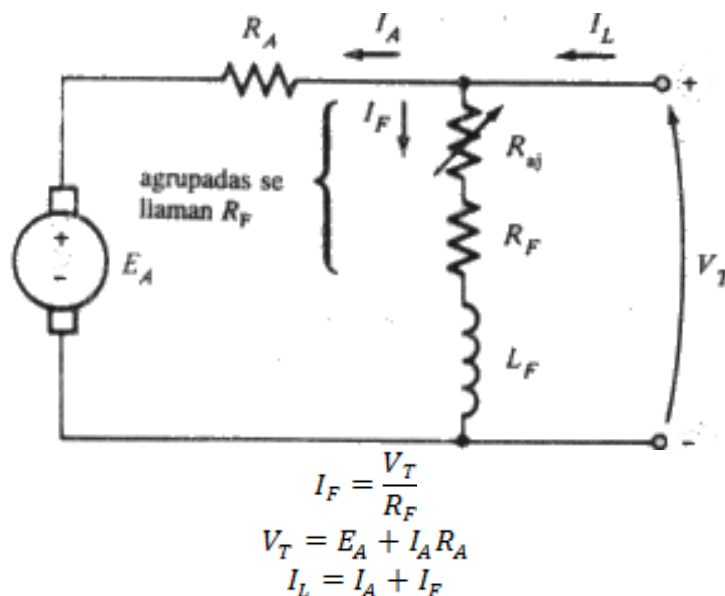


Ilustración 12 - Motor de corriente continua con excitación en derivación

La ecuación correspondiente a la ley de voltajes de Kirchoff para el circuito del inducido de estos motores es:

$$V_t = E_a + I_a * R_a$$

La respuesta de un motor de corriente continua con excitación en derivación ante un aumento del par de carga en cuanto este excede al par inducido en la máquina es reducir su velocidad. Cuando el motor se frena su voltaje interno generado disminuye ($E_a = K\phi\omega \downarrow$) por lo que la corriente del inducido del motor aumenta ($I_a = (V_t - E_a) / R_a$). Cuando aumenta la corriente del inducido aumenta a su vez el par inducido ($\tau_{ind} = K\phi I_a \uparrow$) y finalmente, el par inducido igualará al de carga a una velocidad mecánica de rotación ω más baja. El comportamiento descrito anteriormente se puede observar en la siguiente gráfica de par inducido frente a velocidad de rotación:

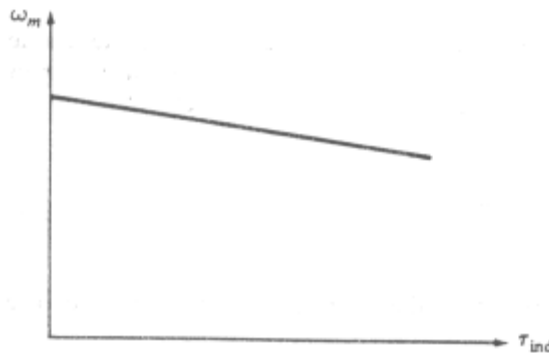


Ilustración 13 - Respuesta de un motor de corriente continua con excitación en derivación ante un aumento en el par de carga

Como ya se dijo anteriormente existen dos métodos de control de la velocidad en un motor de imanes permanentes: el cambio en la tensión del inducido y la inserción de una resistencia en serie con el circuito del inducido.

Los motores de la muñeca del robot se regulan variando su tensión de inducido V_a . Si el voltaje V_a aumenta, aumenta a su vez la corriente de inducido del motor ($I_a = (V_a \uparrow - E_a) / R_a$) y por lo tanto aumenta también el par inducido del motor ($\tau_{ind} = K\phi I_a \uparrow$). Llegados a este punto $\tau_{ind} > \tau_{carga}$ y la velocidad ω del motor aumenta. Como la velocidad ω aumenta, el voltaje interno generado también aumenta ($E_a = K\phi\omega \uparrow$) causando una disminución en la corriente del inducido ($I_a = (V_t - E_a) / R_a$). Esta disminución reduce el par inducido ($\tau_{ind} = K\phi I_a \downarrow$) y ocasiona que τ_{ind} igual a τ_{carga} a una mayor velocidad ω de rotación.

Los cables de alimentación de potencia de los motores y de los frenos, que se explican continuación, son 4 para los motores (dos por motor denominados M+ y M-), dos cables para frenos FR+ y FR-, un cable de tierra y un cable de TERM HN.

CABLES		
Marrón	M5+	M6+
Azul	M6-	M6-
Verde/Amarillo	Tierra M5	Tierra M6
Negro	FR+	FR+
Negro	FR-	FR-

Tabla I - Cables de alimentación de potencia de los motores y de los frenos

En las imágenes inferiores se observan los motores reales y su ubicación en la muñeca del robot.



Ilustración 14 - Motores de los ejes 5 y 6.



Ilustración 15 - Motores de los ejes 5 y 6. Detalle de la parte trasera.

Cada uno de los motores tiene sus especificaciones técnicas serigrafiadas en su carcasa. A continuación se observa la tabla de características del motor del eje 4 del robot original, que es igual a los motores de los ejes 5 y 6. En La tabla siguiente se resumen las características de la placa de características de la imagen inferior.

PARÁMETRO	VALOR	DESCRIPCIÓN
Protección IP	IP 55	Protección contra polvo y agua.
I (sin carga)	12,2 A	Corriente máx. en régimen continuo
I arranque	57 A	Corriente de arranque.
U máx.	99 V	Tensión nominal.
N máx.	2500 rpm	Velocidad en vacío.
Ke	37 mV/rpm	Constante de tensión, fuerza contraelectromotriz o tensión inducida.
Tacómetro/ Generador de CC	13 mV/rpm	Sensibilidad del tacómetro.
Freno electromecánico	24V / 5 Nm	Alimentación de corriente continua y par de frenado.



Ilustración 16 - Placa de características de los motores.



Ilustración 17 - Localización de motores de los ejes 5 y 6 (muñeca).

Además de todo lo explicado anteriormente, cada uno de los motores posee una bobina externa conectada al rotor inducido. Esta bobina tiene como objetivo evitar que la corriente caiga a cero cuando el factor de trabajo de la señal de modulación de ancho de pulso posee valores bajos. La bobina tiene un valor de 3mH y aguanta un máximo de 10 A. En la imagen inferior se observa una imagen de la bobina real.



Ilustración 18 - Bobina externa conectada al inducido de los motores.

2.3.3.1.- FRENOS ELECTROMECAÓNICOS

Dentro del sistema de control original del robot son de vital importancia los frenos electromecánicos adosados a los ejes de cada uno de los motores. Estos frenos tienen la función de permitir el movimiento de los motores si y solo si se ha realizado un proceso satisfactorio de reconocimiento del estado de los mismo y de la etapa de potencia y control. Además, en el caso de detectar un error en el funcionamiento del sistema, por ejemplo un exceso de corriente debido a un exceso de carga o un error de contaje de pulsos que conlleva un desconocimiento de la posición del robot (dentro del sistema de control original del robot, ver el Capítulo III), los frenos se activan deteniendo instantáneamente el movimiento de los motores.

Los frenos están físicamente implementados mediante electroimanes que al activarse permiten el movimiento de los ejes y cuando no se les alimenta bloquean el movimiento de los motores mediante fricción. En la imagen inferior se observa un esquema del electroimán.

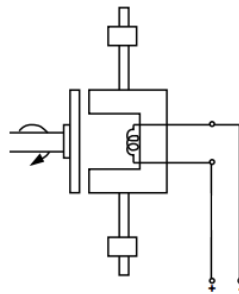


Ilustración 19 - Electroimán frenos. Esquema.

En la imagen inferior se observa un esquema de los frenos y sus partes.

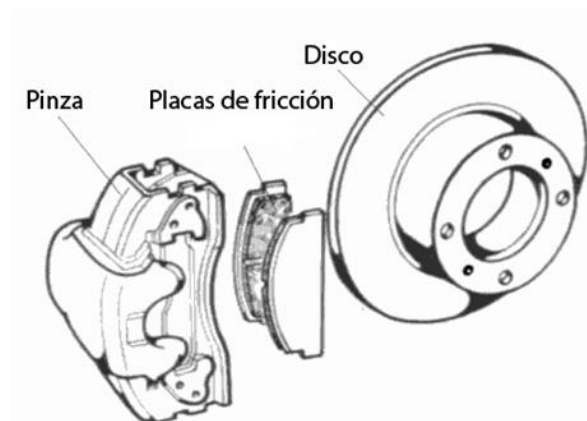


Ilustración 20 - Frenos mecánicos. Esquema.

Estos frenos necesitan una alimentación de 24V de corriente continua que se consigue a través de un contactor, un transformador monofásico y un rectificador monofásico. En la imagen inferior se observa el contactor.

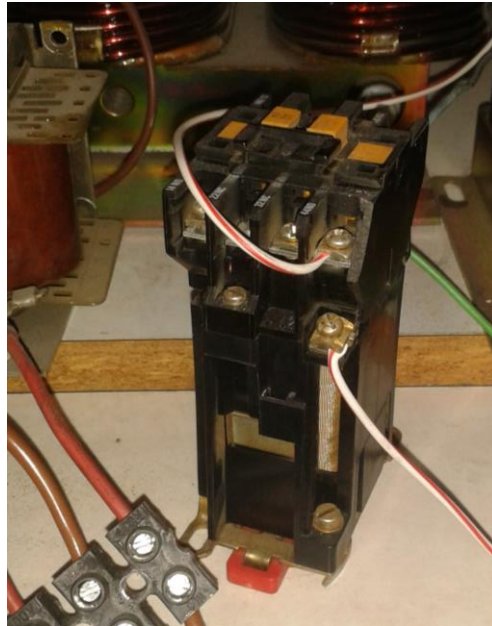
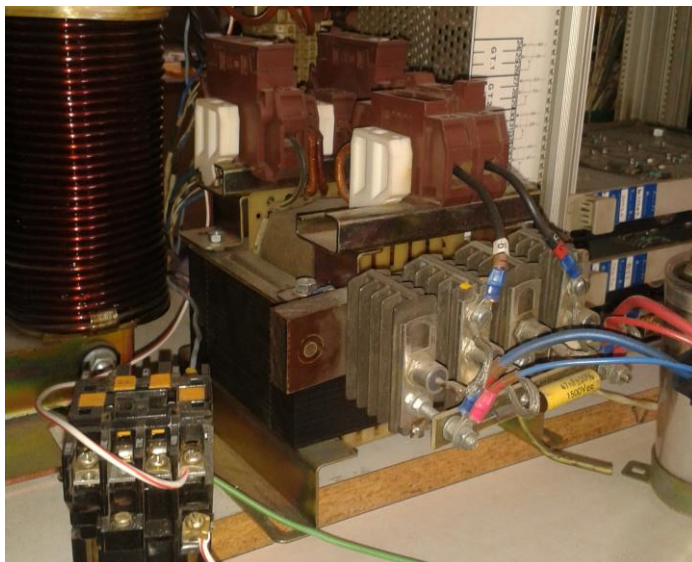


Ilustración 21 - Contactor monofásico del circuito de alimentación de frenos.

Además, se incluye un condensador de 22000 uF y 40V a la salida del rectificador que actuará como filtro antes de la carga. La imagen inferior muestra el transformador monofásico y el rectificador con el condensador del filtro capacitivo.



**Ilustración 22 – Transformador monofásico y rectificador para alimentación de frenos.
Condensador del filtro capacitivo.**

En la imagen inferior se observa un esquema eléctrico del sistema de alimentación de frenos.

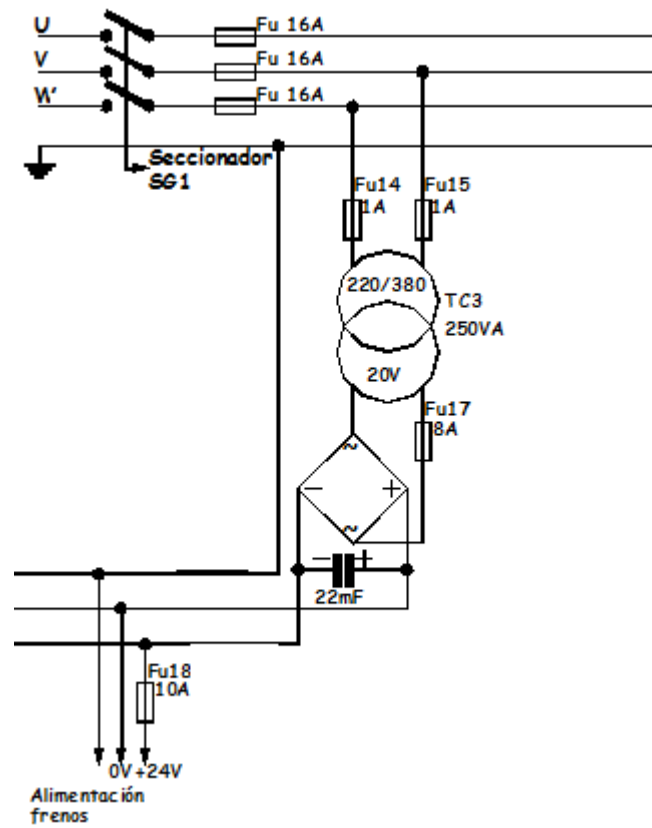


Ilustración 23 - Esquema eléctrico de la alimentación de los frenos.

Sección III

SENSORES

1.- INTRODUCCIÓN

Los sensores son un elemento fundamental para el funcionamiento tanto de la etapa de potencia como de la muñeca del robot y del nuevo sistema de control creado. Mediante las señales eléctricas que generan el sistema es capaz de actuar teniendo en cuenta las condiciones a tiempo real de su entorno. Las distintas actuaciones que permiten realizar son: actuaciones de control, seguridad en el funcionamiento de motores y de realimentación de lazos cerrados de control, de envío de órdenes directas al microcontrolador y de conocimiento de magnitudes en lazo abierto para su uso dentro del sistema de control. En total existen cinco sensores diferentes dentro del sistema. A continuación se describen cada uno de ellos.

Dentro de la etapa de potencia se encuentran dos sensores. Por un lado, un tacómetro envía señales eléctricas proporcionales a la fuerza contraelectromotriz generada en el rotor inducido de los actuadores. El conocimiento de esta magnitud permite crear un lazo cerrado de control de la velocidad de giro de los motores. La señal del sensor se resta con la de consigna obteniendo una señal de error que se transformará en una señal de control a aplicar sobre el sistema. Por otro, un sensor tipo sonda de corriente mide en todo momento la cantidad de corriente que atraviesa el inducido del motor. Esta corriente es indicativa del correcto funcionamiento de los motores en el sentido en que un exceso de la misma sería indicador de un fallo dentro del sistema (por ejemplo un exceso de carga). Mediante etapas de adaptación de señales se consigue una señal de tensión proporcional a la intensidad que se encargará de proporcionar valores lógicos que informarán del correcto o incorrecto funcionamiento de los motores. Ambas señales, tanto la del tacómetro como la del sensor de corriente se tratan dentro de las placas de control originales de la etapa de potencia de los motores que se describirán en el capítulo siguiente.

La muñeca del robot cuenta con dos sensores, un encoder en cada motor para medir el desplazamiento angular de los giros que realizan y un sensor inductivo para cada eje encargado de indicar la posición cero de inicio de todos los movimientos. En definitiva, estos

sensores le proporcionaban al robot un conocimiento de su estado y específicamente de la posición de sus articulaciones. Toda la etapa de control antigua que gestionaba estas señales se encuentra actualmente inutilizada. De ahí que el nuevo sistema de control diseñado las utilice como señales de entrada para ser gestionadas por el microcontrolador. El sistema de gestión de estas señales se basa en generar interrupciones dentro de la programación del microcontrolador a través de los circuitos integrados gestores de interrupciones 8259.

Por último, se agregaron unos sensores finales de carrera adosados mediante ángulos metálicos a la muñeca del robot. Estos sensores tienen como objetivo demarcar el área de trabajo de los dos ejes en cuestión, limitando el movimiento del eje 5 a 180° . La instalación de estos sensores finales de carrera se debió principalmente a cuestiones de seguridad en el manejo de la muñeca. Sin estos sensores no se tiene control total del movimiento del eje 5, que en caso de superar los 180° demarcados puede impactar sobre la superficie de la mesa que lo sostiene.

2.- DESCRIPCIÓN DE LOS SENSORES

A continuación se describirán cada uno de los sensores, su funcionamiento teórico y la implementación práctica que se ha tenido que realizar para algunos de ellos.

2.1.- TACÓMETRO

La captación de la velocidad se hace necesaria para mejorar el comportamiento dinámico de los actuadores del robot. La información de la velocidad de movimiento de cada actuador se realimenta a un bucle de control analógico que proporcionará una señal de actuación sobre los motores. Este bucle de control se encuentra ubicado en la unidad de control original del robot. Puesto que el bucle de control de velocidad es analógico la señal analógica de tensión proporcional a la velocidad de giro del tacómetro puede incluirse fácilmente en el sistema. Otra posibilidad al alcance del sistema existente, puesto que también se tiene encoders consistiría en derivar la información de posición.

Un sistema impulsado por un motor de corriente continua debe mantener constante su velocidad, ya que en aplicaciones de servomecanismos, como la tratada en este trabajo, la velocidad de giro debe ser conocida y controlada desde un circuito de control. La señal entra al circuito de control de velocidad del motor y con la configuración apropiada un circuito de realimentación negativa reajusta la corriente de alimentación del motor hasta llevar la velocidad de nuevo al valor de consigna deseado.

2.1.1.- TIPOS DE TACÓMETROS

Los tacómetros se dividen principalmente en dos tipos: tacómetros ópticos y de contacto. En los tacómetros ópticos no es necesario el contacto entre las partes mientras que en los de contacto se debe hacer contacto entre el instrumento y a pieza que rota para realizar la medición. En general la velocidad de rotación se puede determinar por muchas vías, por lo que existe gran variedad de tacómetros. Atendiendo a su principio de funcionamiento se dividen en: tacómetros de corrientes de Eddy, tacómetros centrífugos, tacómetros eléctricos, electrónicos de contadores de pulsos inducidos, ópticos y estroboscópicos.

- Tacómetro de corrientes de Eddy

Estos dispositivos basan su funcionamiento en la fuerza de arrastre que presenta un disco conductor debido a las corrientes inducidas en él cuando se encuentra cerca de un imán que gira, generalmente de aluminio. De ahí que también reciban el nombre de tacómetro de inducción. La rotación del imán produce un campo magnético que induce una corriente eléctrica en el disco de aluminio. La intensidad de las corrientes generadas aumenta al aumentar la velocidad que adquiere el imán. A su vez, aumentará el par de giro generado en el disco. De esta forma se logra que el ángulo de rotación de una aguja, que sufrirá la fuerza generada en el disco, varíe dentro de una escala apropiada para proporcionar valores correctos de velocidad de giro. En la imagen inferior se observa un esquema del funcionamiento de este tipo de tacómetros.

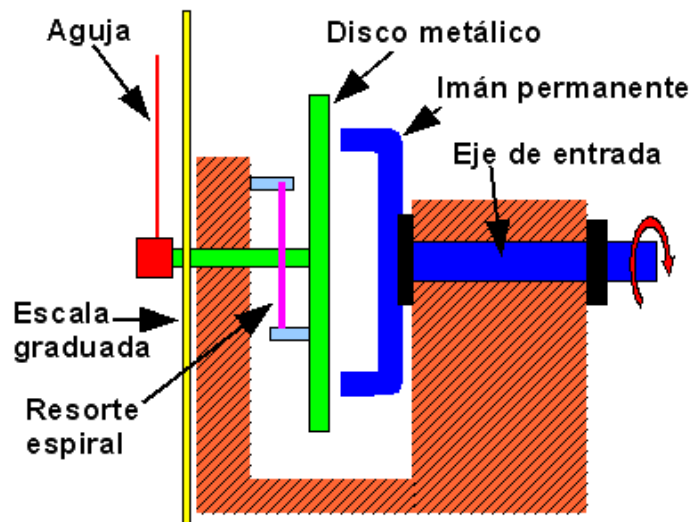


Ilustración 24 - Tacómetro de corrientes de Eddy. Esquema.



Ilustración 25 - Tacómetro de corrientes de Eddy.

- Tacómetro centrífugo

Los tacómetros centrífugos basan su funcionamiento en la fuerza centrífuga que se genera en una masa giratoria. En la imagen inferior se observa el esquema de las partes interiores del tacómetro.

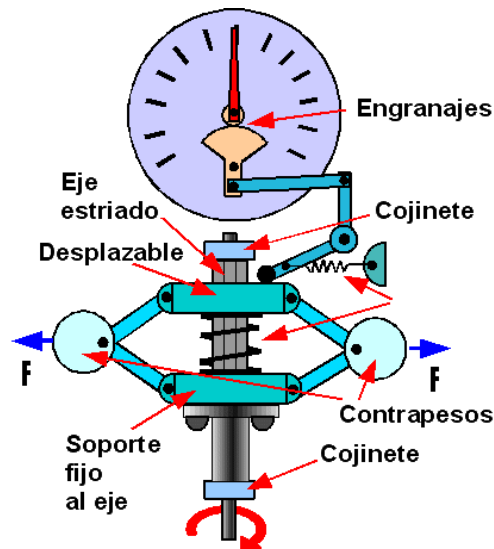


Ilustración 26 - Tacómetro centrífugo. Esquema.

Un eje soportado por cojinetes se acopla a la pieza en movimiento cuya velocidad de rotación se quiere determinar. En este eje están montadas dos piezas soporte separadas por un muelle, una fija y otra desplazable. Los dos contrapesos acoplados a los soportes a través de palancas giran solidariamente al eje y sufren la fuerza centrífuga (F) que tiene a separarlos al mover la pieza desplazable y vencer la resistencia del muelle. Un mecanismo de palancas y un juego de engranajes de amplificación transmiten el movimiento a una aguja que indica la magnitud deseada en la escala graduada en unidades de velocidad de rotación (por ejemplo r.p.m. revoluciones por minuto).

- Tacómetros ópticos

Un tacómetro óptico mide la velocidad de giro usando un haz de luz visible o infrarroja a una distancia máxima de 8 mm del elemento rotatorio. Pertenece a la clase de tacómetros que no necesitan contacto con la pieza a medir.

Estos dispositivos producen un haz de luz que se dirige a la pieza en movimiento donde se marca una zona de color blanco que refleja el haz luminoso en mayor proporción que el resto de la superficie. Este paso luminoso reflejado es detectado por un sensor dispuesto paralelamente al emisor de luz y convertido en una señal eléctrica lógica. Finalmente, un circuito convenientemente calibrado refleja el resultado de la medición de forma analógica o genera un número en una pantalla digital.

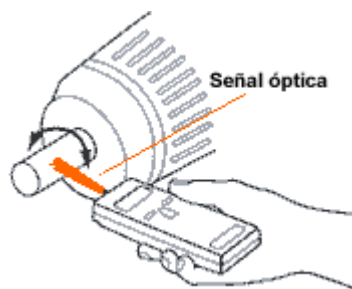


Ilustración 27 - Tacómetro óptico.

- Tacómetro estroboscópico

Estos tacómetros del tipo sin contacto basan su funcionamiento en el efecto estroboscópico, es decir, en la visión como estacionarios de objetos que rotan cuando son iluminados por una haz de luz intermitente sincronizado con su velocidad de rotación. Son en esencia una lámpara potente de encendido instantáneo cuya frecuencia de encendido y apagado se controla a través de un circuito interno.

Cuando se quiere medir la velocidad de rotación de un objeto se enfoca la lámpara hacia él y se va variando de manera gradual la frecuencia de encendido apagado. En el momento en el que el objeto se perciba detenido estarán sincronizadas la velocidad de rotación y la frecuencia de encendido apagado del haz. El valor de la velocidad de giro se obtiene mediante la conversión del valor de frecuencia a una tensión proporcional.



Ilustración 28 - Tacómetro estroboscópico.

- Tacómetro contador de pulsos

El tacómetro contador de pulsos es en definitiva un encoder utilizado para medir velocidad de giro y no posición, que es la medida más directa que proporciona. Una vez conocida la posición la medida de velocidad se obtiene derivando. Este tipo de sensores se describen en detalle al final de este capítulo.

- Tacómetro eléctrico

Este dispositivo basa su funcionamiento en la generación de tensión a partir de un movimiento giratorio en un generador de corriente. Los tacómetros de los motores de los robots son de este tipo por lo que se explican en detalle a continuación.

2.1.2.- TACÓMETRO ELÉCTRICO DE LOS MOTORES DE LA MUÑECA

La muñeca del robot posee dos ejes. Cada uno de los ejes es accionado por un motor y cada motor posee un tacómetro para la medición de la velocidad de giro. El tacómetro es en este caso un generador de corriente continua adosado al eje de motor e integrado en la misma carcasa. El tacómetro es por tanto el elemento que permite conocer la velocidad de los motores de corriente continua de cada eje proporcionando una señal de control a la electrónica del robot. Existe un tacómetro para el eje 5 y uno para el eje 6 y su posición se indica en el siguiente esquema:

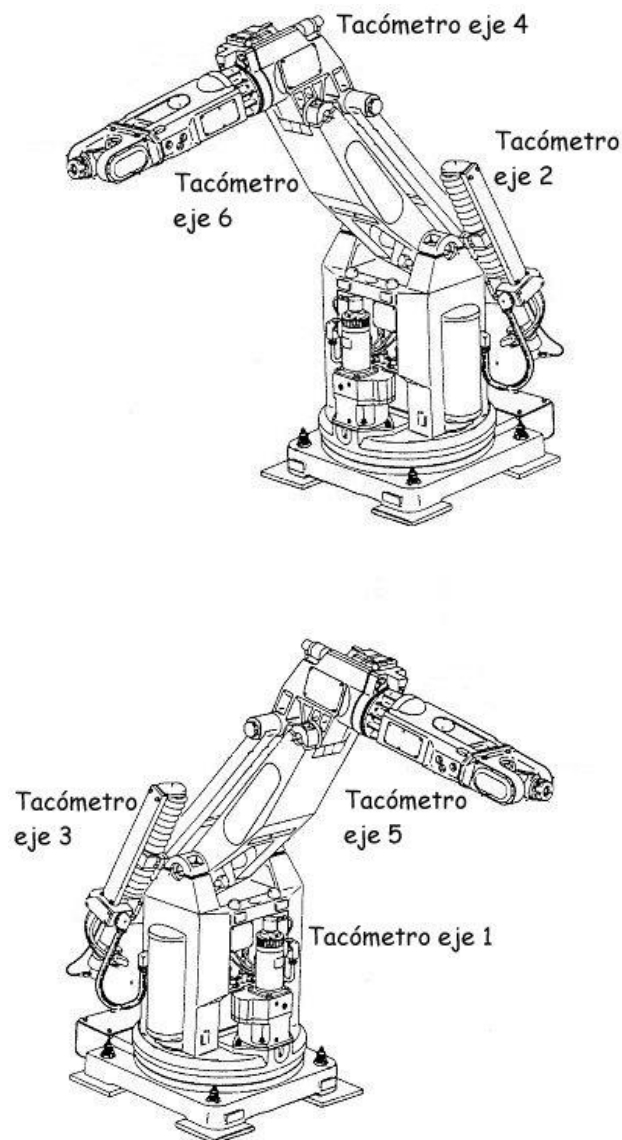


Ilustración 29 - Posición de los tacómetros originales del robot.

El tacómetro se introduce en el circuito del motor debido a la necesidad de mantener la velocidad de giro constante e igual a la de consigna. La velocidad de consigna se introduce indirectamente al motor a través de la tensión de armadura variable (inducido) generada mediante el nuevo sistema de control diseñado.

El tacómetro se incluye en el eje del motor y sirve como transductor de velocidad en la rama de realimentación negativa del lazo de control. A su vez, esta señal de realimentación negativa se introduce en la etapa de control para posteriormente gobernar la etapa de potencia, variando la corriente que alimenta al motor y por lo tanto su velocidad. El tacómetro adosado al eje del motor proporciona, en este caso, 10 mV por rpm. En la siguiente imagen se observa un esquema del control interno de los motores. El tacómetro está señalado en color rojo

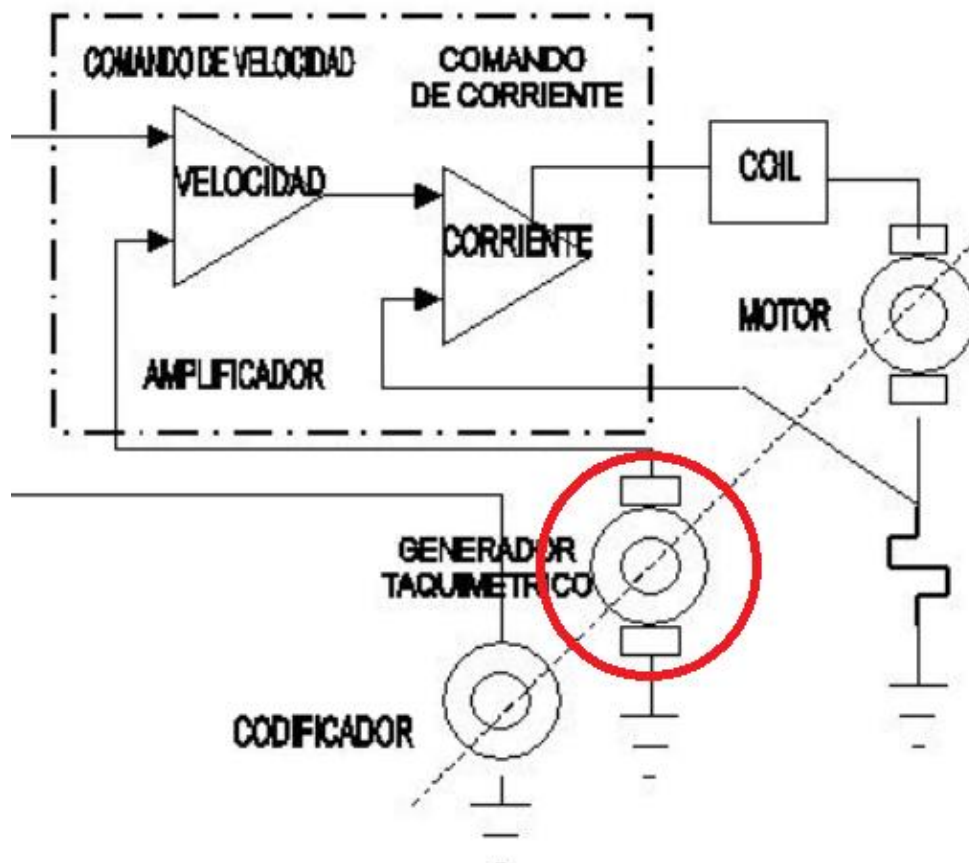


Ilustración 30 - Esquema de control de los motores del muñeca. Tacómetro

Como se observa en la imagen la salida del tacómetro constituye la señal de realimentación del bucle de velocidad dentro del esquema de control original del robot. A continuación la señal de salida se resta con otra señal de realimentación esta vez proveniente del sensor de corriente, que se explica a continuación.

Las señales de los tacómetros viajan por seis cables, un cable fin de carrera (no conectado), dos cables de alimentación 0V y +5V y un cable TERM BN. En la imagen inferior se observa la ubicación real de los cables del tacómetro a la salida del motor.

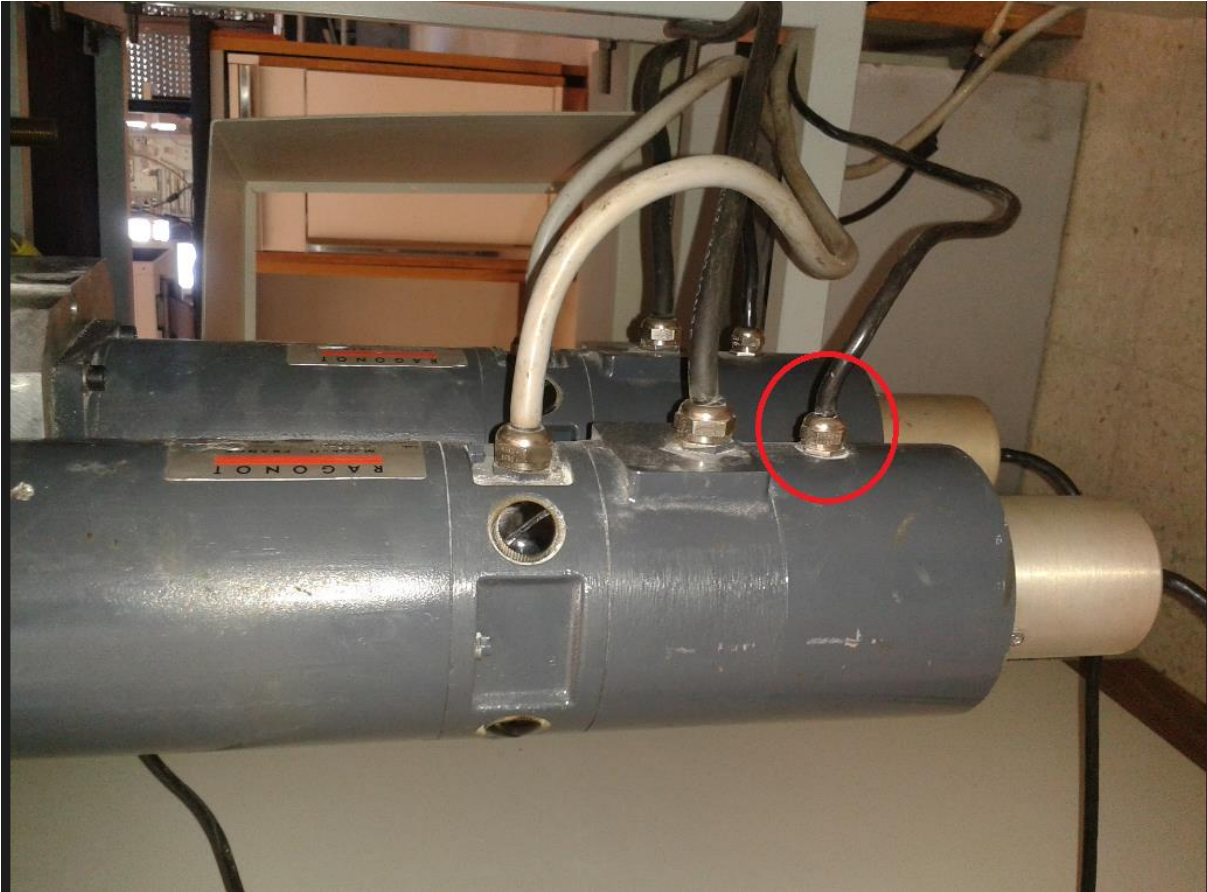


Ilustración 31 - Cable de salida de la señal del tacómetro.

2.2.- SENSOR DE CORRIENTE

Un sensor de corriente convierte corriente alterna o continua a una salida de tensión proporcional a la intensidad medida. Cuando una corriente atraviesa un conductor se produce una caída de tensión. A su vez, si se trata de corriente alterna se genera un campo magnético alrededor del cable que genera la corriente. Ambos efectos son utilizados en el diseño de sensores de corriente. Así, existen dos tipos principales de sensores: sensores de medición directa y sensores de medición indirecta.

La medición directa está basada en la ley de Ohm, mientras que la medición indirecta está basada en la ley de Faraday (cualquier cambio en el entorno magnético de una bobina induce una tensión en la misma) y la ley de Ampere (el campo magnético en el espacio alrededor de una corriente eléctrica es proporcional a dicha corriente eléctrica).

2.2.1.- TIPOS DE SENSORES DE CORRIENTE

- Sensor resistivo de intensidad

El método más sencillo para obtener una señal proporcional a la intensidad es una resistencia serie conectada a la carga. La resistencia es óhmica pura por lo que la tensión en sus bornes es proporcional a la corriente que circula por ella. Se basa en la ley de Ohm.

Debe tenerse en cuenta que la resistencia en el circuito provoca que la corriente disminuya, lo que induce un error en la medición. Para minimizar el error la resistencia debe tener un valor muy bajo, aunque un valor excesivamente bajo no proporcionará una tensión fácilmente medible. Se utilizan tanto para mediciones de corriente alterna como continua, Cuando el circuito trabaja con tensiones elevadas es necesario realizar un aislamiento eléctrico entre la resistencia y el circuito. Si la corriente es alterna puede emplearse un transformador de aislamiento. En la imagen inferior se observa un posible circuito de medida.

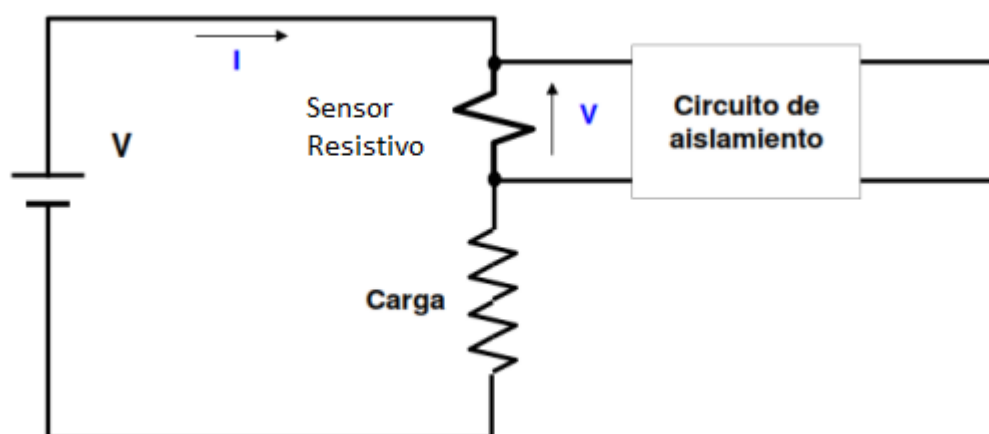


Ilustración 32 - Sensor de corriente resistivo.

- Sensor de efecto Hall

Este tipo de sensor de corriente, como su nombre lo indica, baso su principio de funcionamiento en el efecto Hall. Si una corriente fluye a través de un conductor situado dentro de un campo magnético éste campo ejerce una fuerza transversal sobre los portadores de cargas móviles que tiende a empujarlas hacia un lado del conductor. La acumulación de cargas en los lados del conductor equilibra ésta influencia magnética produciendo una tensión medible en bornes del conductor.

Dentro de las ventajas de este tipo de sensores se encuentra el aislamiento galvánico que proporcionan, la posibilidad de realizar mediciones tanto de corriente continua como de corriente alterna y la inexistencia de caídas de tensión. Por otro lado, muchas veces debe de agregarse al sistema un concentrador de campo para complementar sus prestaciones, lo que lo encarece.

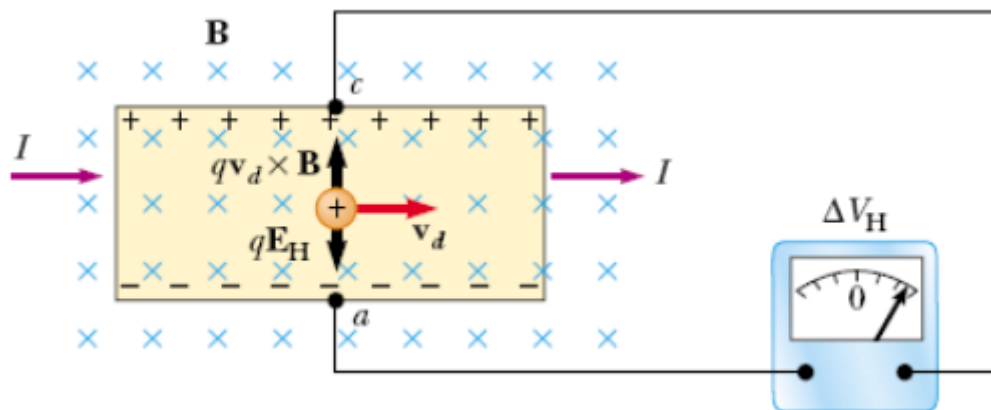


Ilustración 33 - Esquema demostrativo del efecto Hall.

- Bobina de Rogowski

Las bobinas de Rogowski son una de las tecnologías implementadas para la medición de corriente, Este dispositivo electrónico puede medir corriente alterna o pulsos de corriente de alta velocidad y es el transductor encargado de medir la intensidad que circula por los motores del robot por lo que se explica en detalle a continuación.

2.2.2.- BOBINA DE ROGOWSKI. SENSOR DE CORRIENTE DE LOS MOTORES DE LA MUÑECA

Las bobinas de Rogowski se han utilizado para la detección y medida de corriente desde la segunda mitad del siglo XX. Como se dijo anteriormente estas bobinas pueden utilizarse para medir corriente alterna o corrientes transitorias generadas por medio de la conmutación de corrientes continuas. Este último es el caso que tiene aplicación dentro del sistema de la etapa de potencia original del robot. La tensión que le llega a los motores es fruto de un proceso de modulación de ancho de pulso a través del sistema de control analógico implementado en las cartas de control electrónicas de los diferentes motores. De ahí que la corriente a medir en el inducido del motor corresponda con pulsos de alta velocidad.

Este sensor de corriente está constituido por un inductor con núcleo de aire de tipo toroidal, cuya salida de tensión es proporcional a la corriente a medir según la siguiente expresión.

$$V_{ROG} = M \times \frac{di}{dt}$$

M es la inductancia del arrollamiento. En la imagen inferior se observa el campo magnético en el interior de una bobina toroidal.

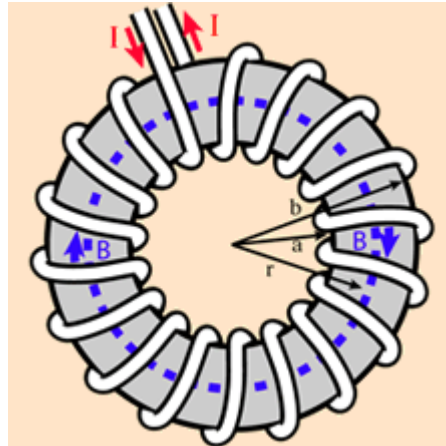


Ilustración 34 - Campo magnético en el interior de una bobina toroidal.

Dadas las características magnéticas del núcleo de aire la curva de respuesta, a diferencia del caso de hierro, es absolutamente lineal. No existe saturación del núcleo por lo cual puede medir grandes corrientes sin preocupación por deformación o rotura. La característica derivativa de la señal a la salida de la bobina obliga a realizar una integración en el tiempo a fin de obtener una señal proporcional a la corriente a medir.

$$V_{ROG}(t) = \int_i M \times \frac{di}{dt} = k \times i(t)$$

En la imagen inferior se observa un esquema de una bobina de Rogowski conectada a un circuito integrador.

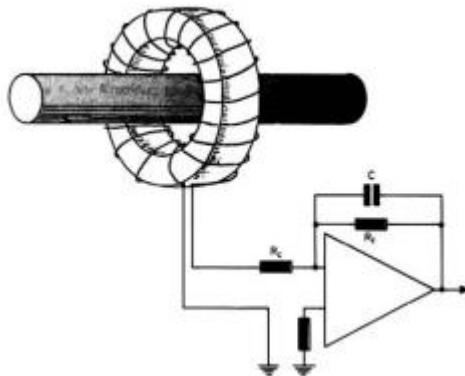


Ilustración 35 - Bobina de Rogowski conectada a un circuito integrador.

Si la bobina de Rogowski forma un circuito cerrado en torno a la corriente a ser medida I entonces la tensión inducida E en la bobina es teóricamente independiente de su forma y de la posición del conductor de corriente dentro de esta.

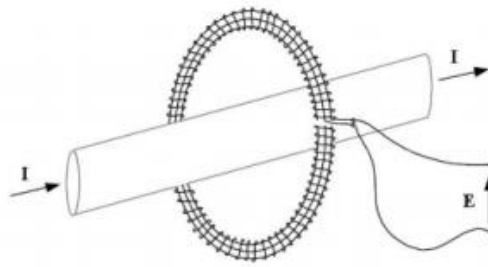


Ilustración 36 - Medición de una tensión inducida en una bobina de Rogowski.

Los sensores se encuentran implementados físicamente en la parte trasera del armario de control original del robot. En las imágenes inferiores se observan en detalle.

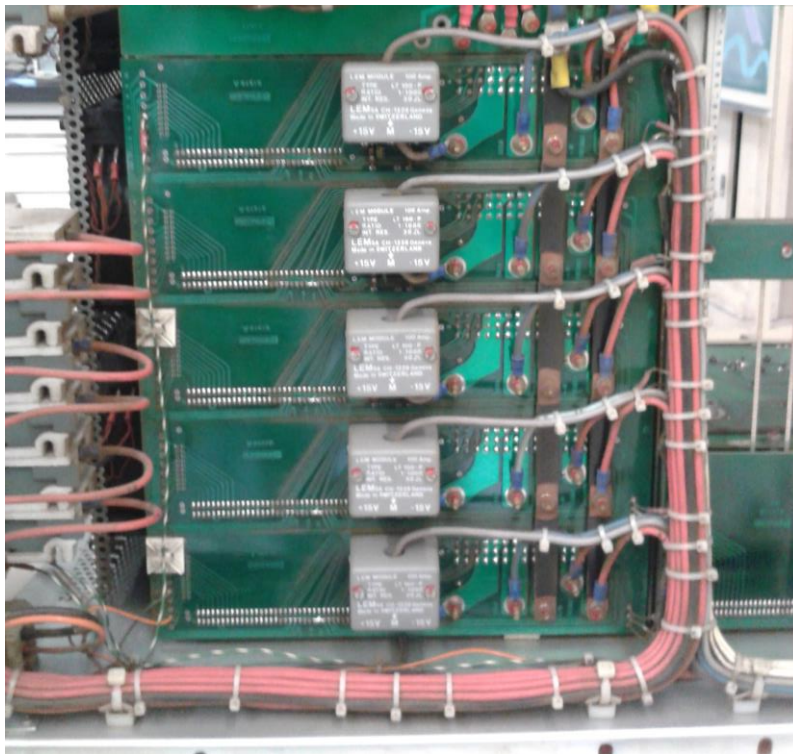


Ilustración 37 - Ubicación física de los sensores de corriente.



Ilustración 38 - Detalle sensor de corriente.

En definitiva, la imagen de corriente del motor se dirige a las placas de control de cada eje de la muñeca con objeto de valorar la temperatura (un exceso de corriente deriva en un aumento de temperatura) de los motores por el cálculo del valor eficaz de la corriente. De forma general, los defectos provocan el desenclavamiento de los frenos electromagnéticos de cada eje del robot y por lo tanto la parada inmediata de los motores.

2.3.- CODIFICADORES ÓPTICOS O ENCODER

Los codificadores ópticos o encoders incrementales constan, en su forma más simple, de un disco transparente con una serie de marcas opacas colocadas radialmente y equidistantes entre sí, de un sistema de iluminación en el que la luz es colimada de forma adecuada y de un elemento fotorreceptor. El eje cuya posición se quiere medir va acoplado al disco transparente. Con esta disposición, a medida que el eje gire se irán generando pulsos en el receptor cada vez que la luz atraviese cada marca. Llevando una cuenta de estos pulsos es posible conocer la posición del eje.

Existe, sin embargo, el problema del desconocimiento en un momento dado del sentido de giro, con el peligro que supone no estar contando adecuadamente. Una solución a este problema consiste en disponer de otra franja de marcas, desplazada de la anterior de manera que el tren de pulsos que con ella se genere esté desplazado 90° eléctricos con respecto al generado por la primera franja. De esta manera, con un circuito relativamente sencillo, es posible obtener una señal adicional que indique cuál es el sentido de giro y que actúe sobre el contador correspondiente indicándole que incremente o reduzca la cuenta que se está realizando. Es necesario además disponer de una marca de referencia sobre el disco que indique que se ha dado una vuelta completa y que, por tanto, se ha de empezar la cuenta de nuevo. Esta marca sirve también para poder comenzar a contar tras recuperarse de una caída de tensión.



Ilustración 39 - Esquema de funcionamiento del codificador angular de posición encoder.

La resolución de este tipo de sensores depende directamente del número de marcas que se pueden poner físicamente en el disco. Un método relativamente sencillo para aumentar esta resolución es no solamente contabilizar los flancos de subida de los trenes de pulsos, sino contabilizar también los de bajada, incrementando así la resolución del captador, pudiéndose llegar, con ayuda de circuitos adicionales, hasta 100.000 pulsos por vuelta.

El funcionamiento básico de los codificadores o encoders absolutos es similar al de los incrementales. Se tiene una fuente de luz con las lentes de adaptación correspondientes, un disco graduado y unos fotorreceptores. En este caso, el disco transparente se divide en un número determinado de sectores (potencia de 2), codificándose cada uno de ellos según un código binario cíclico (normalmente el código de Gray) que queda representado por zonas transparentes y opacas dispuestas radialmente. No es necesario ahora ningún contador o electrónica adicional para detectar el sentido del giro, pues cada posición (sector) es codificado de forma absoluta. Su resolución es fija, y vendrá dada por el número de anillos que posea el disco graduado. Las resoluciones habituales van desde 2^8 a 2^{19} bits (desde 256 a 524288 posiciones distintas).

Normalmente los sensores de posición se acoplan al eje del motor. Considerando que en la mayor parte de los casos entre el eje del motor y el de la articulación se sitúa un reductor de relación N, cada movimiento de la articulación se verá multiplicado por N al ser medido por el sensor. Éste aumentará así su resolución, multiplicándola por N. En algunos encoders absolutos se utiliza otro encoder absoluto más pequeño conectado por un engranaje reductor al principal, de manera que cuando éste gire una vuelta completa, el codificado adicional avanzará una posición. Son los denominados encoder absolutos multivuelta.

Esta misma circunstancia originará que en el caso de los codificadores incrementales la señal de referencia o marca de cero, sea insuficiente para detectar el punto origen para la cuenta de pulsos, pues habrá N posibles puntos de referencia para un giro completo de la articulación. Para distinguir cuál de ellos es el correcto se suele utilizar un detector de presencia denominado sincronismo, acoplado directamente al eslabón del robot que se considere. Este detector de presencia se encuentra implementado físicamente como un sensor inductivo en cada eje del robot y se explicará en el siguiente punto.

Los encoders pueden presentar problemas mecánicos debido a la gran precisión que se debe tener en su fabricación. La contaminación ambiental puede ser una fuente de interferencias en la transmisión óptica. Son dispositivos particularmente sensibles a golpes y vibraciones, estando su margen de temperatura de trabajo limitado por la presencia de componentes electrónicos.

2.3.1.- CODIFICADORES ESPECÍFICOS DE LOS EJES DE LA MUÑECA

Cada uno de los ejes del robot cuenta con su propio encoder incremental para el control de su posición. Su localización en la muñeca del robot puede observarse en la siguiente imagen:

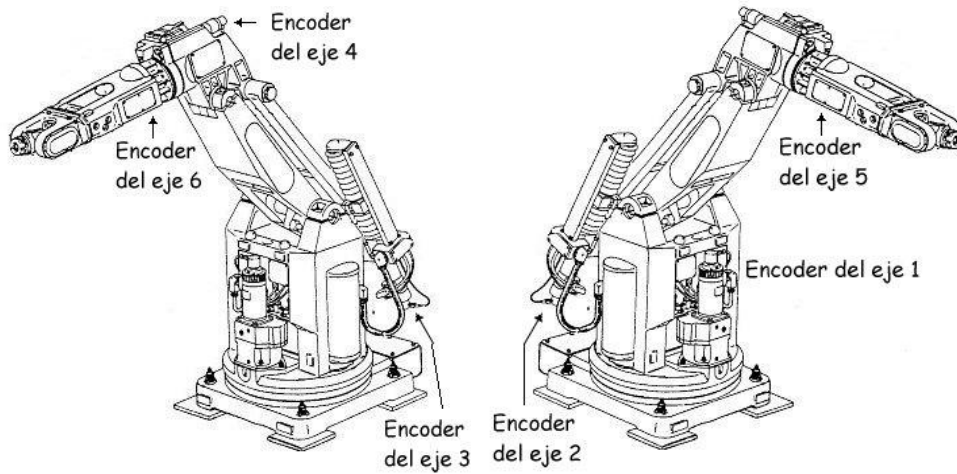


Ilustración 40 - Localización de los encoders en los ejes del robot original.

En las imágenes inferiores se observa un esquema del encoder y una imagen del encoder real.

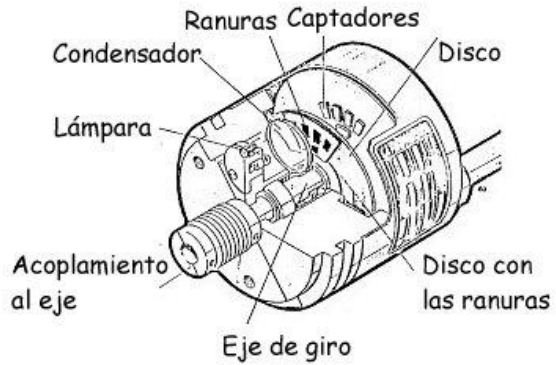


Ilustración 41 - Esquema del encoder.



Ilustración 42 - Encoder de los ejes de la muñeca del robot.

Los encoders se conectan a los ejes de los motores justo al lado contrario de donde se aprovecha el movimiento. Su proximidad con el colector de delgas del motor de corriente continua hace que se induzcan muchas señales de ruido en las señales del encoder. Este problema obligará a incluir filtros en las etapas contadoras para evitar fallos en la medición.

El encoder de cada eje proporciona dos formas de ondas cuadradas y desfasadas entre sí 90° grados. Con la lectura de una sola señal se dispone de la información correspondiente a la velocidad de rotación, con la otra señal es posible discriminar el sentido de rotación en base a la secuencia de datos que producen ambas señales. Además, los encoders de la muñeca del robot cuentan con una tercera señal que marca una vuelta completa.

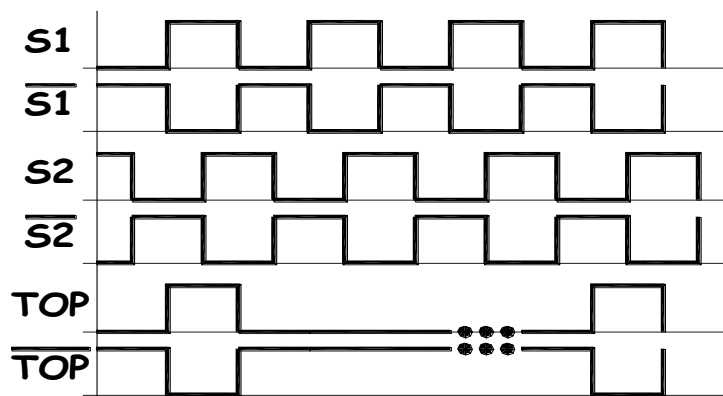


Ilustración 43 - Señales principales del encoder.

Para incrementar la calidad y estabilidad de las señales, el sistema de lectura se efectúa de manera diferencial, comparando señales idénticas pero desfasadas 180° . Su lectura se efectúa en base a la diferencia de las dos señales, eliminando de este modo las interferencias del modo común. En la realidad los encoders se encuentran unidos solidariamente al eje del motor y están ubicados en el lado contrario del eje de rotación. En las imágenes inferiores se observa la ubicación real del encoder y del cable por el que viajan las señales.

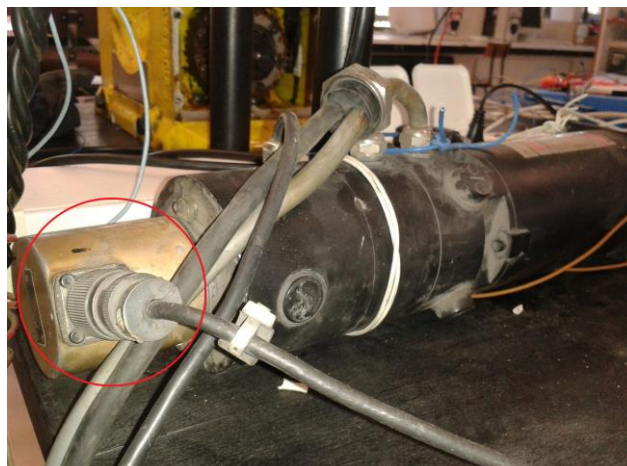


Ilustración 44 - Ubicación de los encoders en los motores y cable por el que viajan las señales.

2.4.- SENSORES INDUCTIVOS

Los sensores inductivos son una clase especial de sensores que sirven para detectar materiales metálicos ferrosos proporcionando una señal digital o analógica proporcional a la distancia. Pueden ser utilizados como transductores de velocidad, posición o proximidad, como es el caso. Constan de un devanado interno por el que circula una corriente que genera un campo magnético cambiante de alta frecuencia. Esta bobina forma parte de un circuito de resonancia. Al acercarse un material metálico a la zona de campo magnético cambiante el devanado induce corrientes de Foucault en él. De esta forma, se generan corrientes circulantes en la pieza que a su vez crean un campo magnético que se opone al de la bobina del sensor. Esto hace que el circuito en resonancia se altere, reduciendo la inductancia de la bobina del sensor e intrínsecamente el valor de la impedancia de la misma.

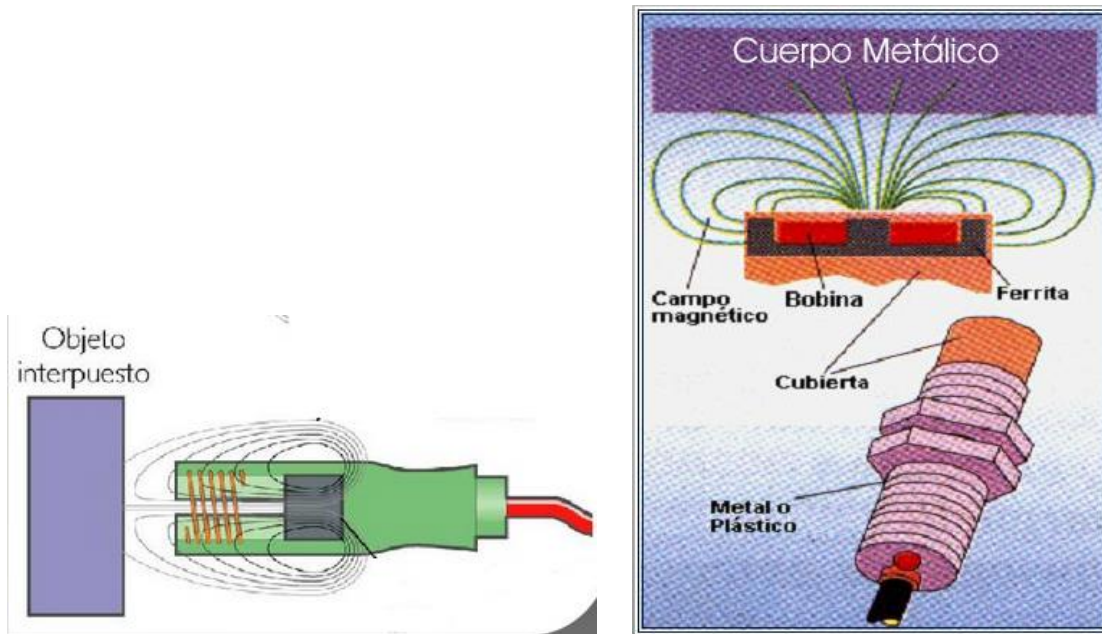


Ilustración 45 - Sensor inductivo. Esquema.

Un sensor inductivo se compone de una serie de elementos. Como ya se dijo anteriormente, la bobina detectora hace parte de un circuito oscilador (o circuito de resonancia) que no es más que un circuito formado por una bobina y un condensador eléctrico en paralelo. Su funcionamiento se basa en el almacenamiento de energía en forma de carga eléctrica en el condensador y en forma de campo magnético en la bobina.

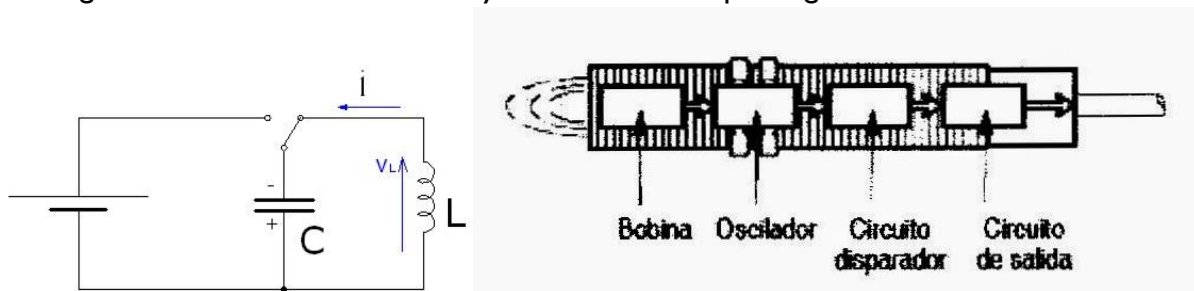


Ilustración 46 - Circuito oscilador base del sensor inductivo.

Al estar el condensador y la bobina en paralelo la energía almacenada por el campo eléctrico del condensador es absorbida por la bobina, que la almacena en el campo magnético para a continuación ser absorbida y almacenada por el condensador, nuevamente en forma de campo eléctrico. Esta variación constituye una oscilación electromagnética en la cual el campo eléctrico y el magnético son perpendiculares entre sí. Cabe aclarar que en cada ciclo de carga y descarga hay pérdidas debido a la resistencia eléctrica del conductor que conforma la bobina y a las fugas del dieléctrico del condensador. Debido a esto el voltaje al que se carga el condensador disminuye con el tiempo. Para evitar este fenómeno se disponen circuitos electrónicos amplificadores que reponen el voltaje perdido y mantienen las oscilaciones constantes con el tiempo.

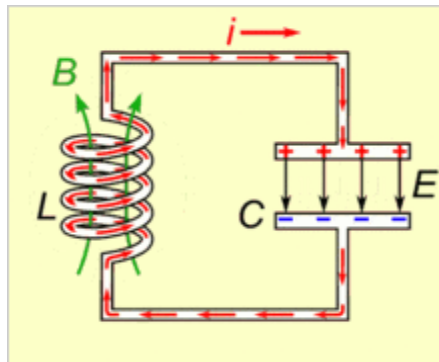


Ilustración 47 - Funcionamiento de un circuito oscilador LC.

Después del circuito oscilador se encuentra un circuito disparador. Este circuito disparador se conoce con el nombre de disparador Schmitt. Para su implementación se suele utilizar un amplificador operacional realimentado positivamente. Los niveles de referencia se controlan ajustando las resistencias R_1 y R_2 .

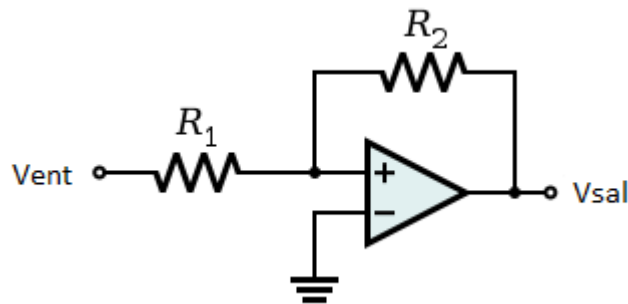


Ilustración 48 - Circuito disparador Schmitt.

Si el disparador está inicialmente activado, la salida estará en estado alto a una tensión $V_{sal} = +V_{sat}$ y las dos resistencias formarán un divisor de tensión entre la entrada y la salida. Para producir una transición a la salida la tensión V_+ (del terminal positivo del amplificador) debe descender a 0 V (pues el terminal negativo se encuentra conectado a masa). En este caso la tensión de entrada es:

$$V_l = -V_{sat} \frac{R_1}{R_2}$$

Llegado a este punto la tensión a la salida cambia $V_{sal} = -V_{sat}$ por lo que la tensión a la entrada será:

$$V_u = +V_{sat} \frac{R_1}{R_2}$$

En definitiva el circuito crea una banda centrada en cero con niveles de disparo en $\pm(R_1/R_2)V_{sat}$. La señal de entrada de salir de la banda para conseguir cambiar la tensión de salida. Si R_1 es cero y R_2 infinito (circuito abierto), la banda tendrá una anchura de cero y el circuito funcionaría como un comparador normal. En la siguiente gráfica puede observarse el comportamiento del disparador ante una entrada de tensión cambiante.

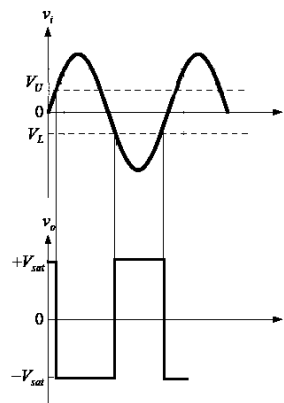


Ilustración 49 - Señal de salida del disparador Schmitt.

Por último, el sensor inductivo finaliza en un circuito de salida que se encarga de dar una señal de OFF cuando el circuito disparador conmuta a $+V_{sat}$ y una señal de ON cuando este conmuta a $-V_{sat}$. Esto es debido al efecto negativo del campo generado en la pieza sobre la bobina de circuito oscilador. El funcionamiento completo del sensor puede observarse en la gráfica siguiente.

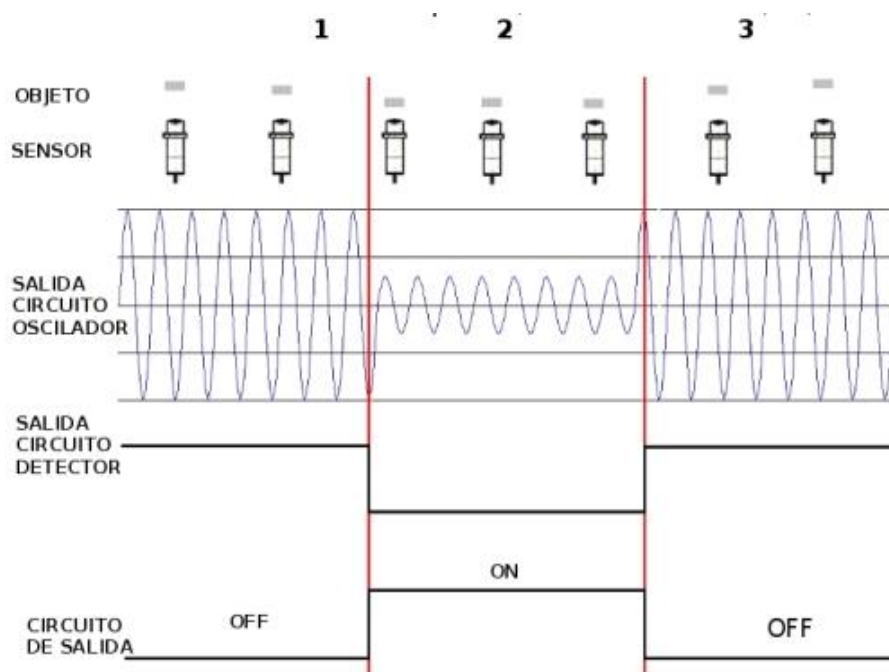


Ilustración 50 - Funcionamiento completo del sensor inductivo.

La distancia de detección depende directamente del tamaño de la bobina y del tipo de construcción del sensor, blindado o no blindado. Los sensores no blindados tienen mayor distancia de detección que los sensores blindados. Por regla general el alcance máximo del sensor inductivo es la mitad de su diámetro. Los sensores de proximidad tienen bobinas enrolladas en núcleo de ferrita. Estas pueden ser blindadas o no blindadas.

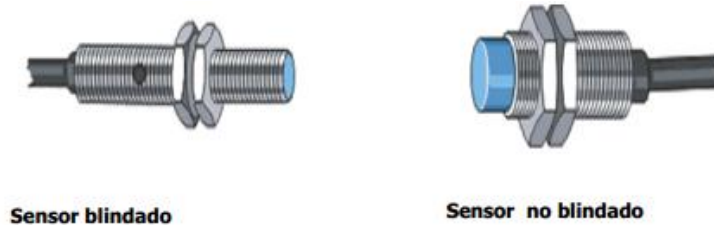


Ilustración 51 - Diferencias morfológicas entre el sensor inductivo blindado y el no blindado.

En sensores blindados el núcleo de ferrita concentra el campo radiado en la dirección de uso. Suele colocársele alrededor del núcleo un anillo metálico para restringir la radiación lateral del campo y pueden ser montados al ras de metal.

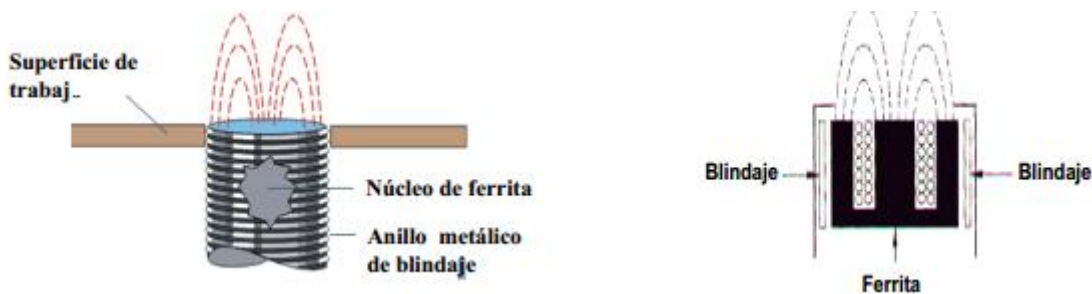


Ilustración 52 - Montaje del sensor blindado.

Los sensores no blindados no tiene el anillo de metal rodeando el núcleo ni pueden ser montados al ras de un metal, por el contrario, deben tener un área libre de metal alrededor de la superficie de detección.

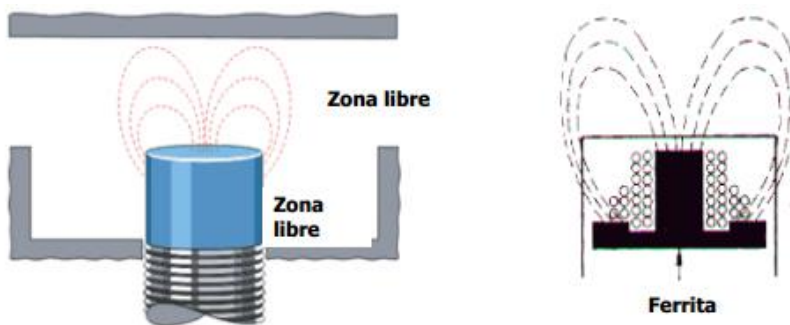


Ilustración 53 - Montaje del sensor no blindado.

La distancia de detección es constante para cualquier grosor del objetivo estándar: una placa de superficie plana, acero dúctil, lisa y de 1 mm de grueso. Sin embargo, para objetivos no ferrosos como el bronce, el aluminio o el cobre ocurre un fenómeno conocido

como “efecto epitelial” en el que la distancia de medición disminuye conforme aumenta el grueso del objetivo (se deberá multiplicar por el factor de corrección del eje y).

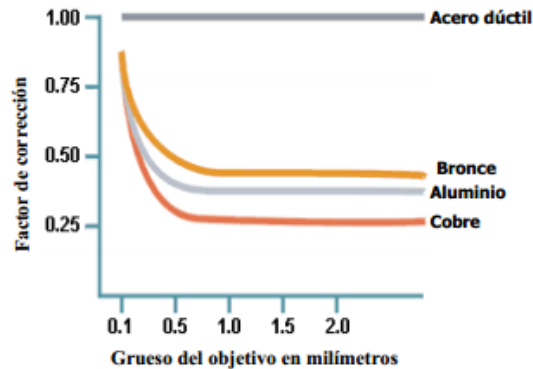


Ilustración 54 - "Efecto epitelial" presentado en diversos metales.

Si el material del objetivo es ferromagnético como es el caso del hierro, cobalto, níquel y sus aleaciones, la distancia de activación disminuye notablemente.

La característica de respuesta de este tipo de sensor está marcada por la histéresis que presenta. Los detectores de proximidad responden a un objeto solo cuando éste se encuentre en un área definida frente a la cara de detección del sensor. El punto en el cual el sensor reconoce al objeto y conmuta su salida a ON se conoce como “punto de operación”. El punto en el que un objetivo saliendo hace que el dispositivo conmute de nuevo a su estado normal se le conoce como “punto de desarme”. El área entre dos puntos es llamada zona de histéresis. Dependiendo de cómo se aproxime el objeto al sensor (movimiento axial o movimiento radial) éste se comportará de forma diferente. El fabricante proporciona comúnmente una curva de respuesta.

El tamaño y forma de la curva de respuesta depende del sensor de proximidad específico. En el siguiente ejemplo un objetivo a 0,45 mm del sensor hace que éste conmute cuando el objetivo cubre un 25% de la cara del sensor. Sin embargo, un objetivo a 0,8 mm del sensor debe cubrir la cara completa del sensor.

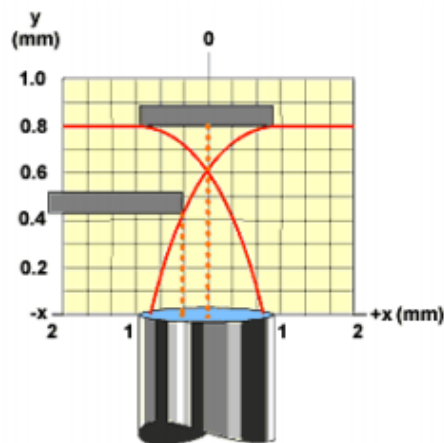


Ilustración 55 - Curva de respuesta del sensor en función de la proximidad.

La frecuencia de conmutación es la velocidad máxima a la que el sensor es capaz de entregar pulsos discretos individuales según el objetivo entra y sale del campo de detección. Este valor depende siempre del tamaño del objeto, de la distancia de éste a la cara de detección, de su velocidad y del tipo de sensor.

2.4.1.- TIPOS DE SENSORES INDUCTIVOS

Existen distintos tipos de sensores inductivos especiales para el trabajo en un ambiente industrial. Algunos sensores inductivos poseen una cubierta de teflón para ser empleados en cualquier atmósfera industrial. De este modo no pueden adherirse en ninguna superficie ni siquiera sobre restos de soldadura. Además, estos sensores no se ven afectados por campos magnéticos externos. Los sensores inductivos cubiertos completamente en metal pueden soportar altas presiones debido a su cubierta de acero inoxidable y suelen utilizarse en la industria de alimentos.

A su vez, existen sensores inductivos especiales para trabajar a altas temperaturas, entre $-20\text{ }^{\circ}\text{C}$ y $250\text{ }^{\circ}\text{C}$. Por último están los sensores especiales para zonas explosivas. Estos sensores están equipados de forma separada con amplificadores de aislamiento que cumplen con las normas ATEX. Estos equipos están certificados para trabajar en áreas con peligro de explosión por gas o por polvo.

En general, la suciedad de materiales no conductivos no tiene ninguna influencia en el funcionamiento del sensor. Tampoco influyen en el sensor los líquidos conductivos, la humedad, niebla o vapores de todo tipo ni el polvo. Las virutas de metal influyen según el tamaño del sensor, a mayor tamaño del sensor la influencia es menor. Siempre que sea posible el objeto debe ser el que se mueva y no el sensor, pues el movimiento de vaivén podría dañar el cable del sensor. En sensores de una cierta calidad las vibraciones no alteran el sensor.

Los sensores inductivos están disponibles en varios tamaños y configuraciones para apegarse a una gran variedad de requerimientos. La forma cilíndrica es la más común, para su montaje sólo se requiere de un agujero redondo. Además, hay sensores de forma cúbica de hasta 35 mm de alcance, sensores planos y rectangulares.



Ilustración 56 - Modelos comerciales de sensores inductivos.

2.4.2.- SENSORES INDUCTIVOS ESPECÍFICOS DE LOS EJES DE LA MUÑECA

El objetivo de los sensores inductivos ubicados en cada eje es reflejar la posición inicial del mismo en la ejecución de una determinada trayectoria. El sensor emite una señal cuando cada uno de los ejes llega a la posición marcada como inicio. Consta de tres hilos cuyas características se resumen en la siguiente tabla.

Hilos del sensor	
T-	25 V
Inic.	Referencia
T+	25 V

Tabla 2 - Características sensores inductivos de la muñeca del robot.

La ubicación de los sensores se puede observar en la siguiente figura:

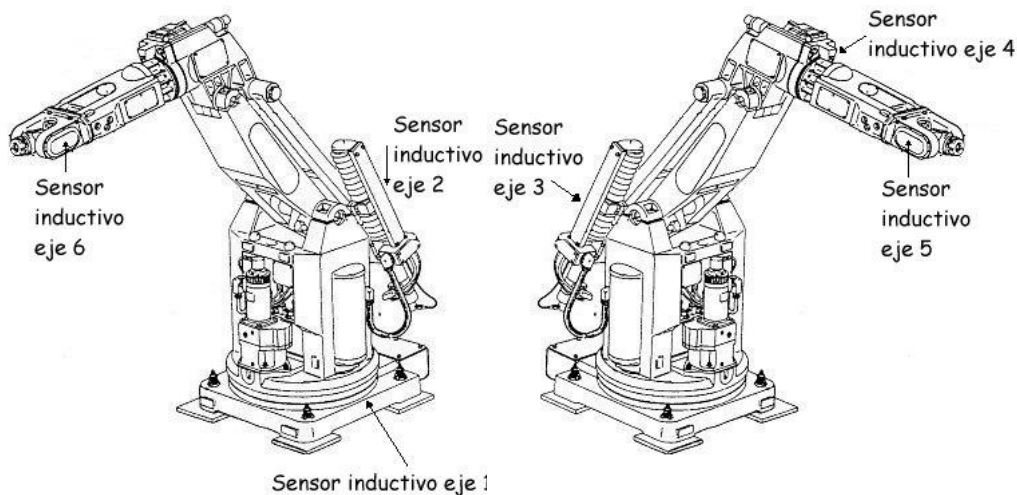


Ilustración 57 - Ubicación real de los sensores inductivos.

Los sensores inductivos de los ejes 5 y 6 se ubican en la zona de los engranajes. Sobre uno de ellos un sensor inductivo de tipo plano detecta el paso de una muesca que se encuentra unida a dicho engranaje.

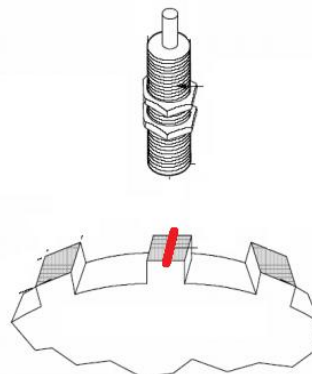


Ilustración 58 - Esquema de funcionamiento del sensor inductivo.

En la imagen inferior se observa el sensor inductivo real del eje 6 del robot. En rojo se demarca el sensor y en azul la pletina metálica que detecta el paso por cero. La razón por la que se produce la detección de la pletina y no del resto del engranaje metálico es la proximidad adecuada de ésta al sensor.

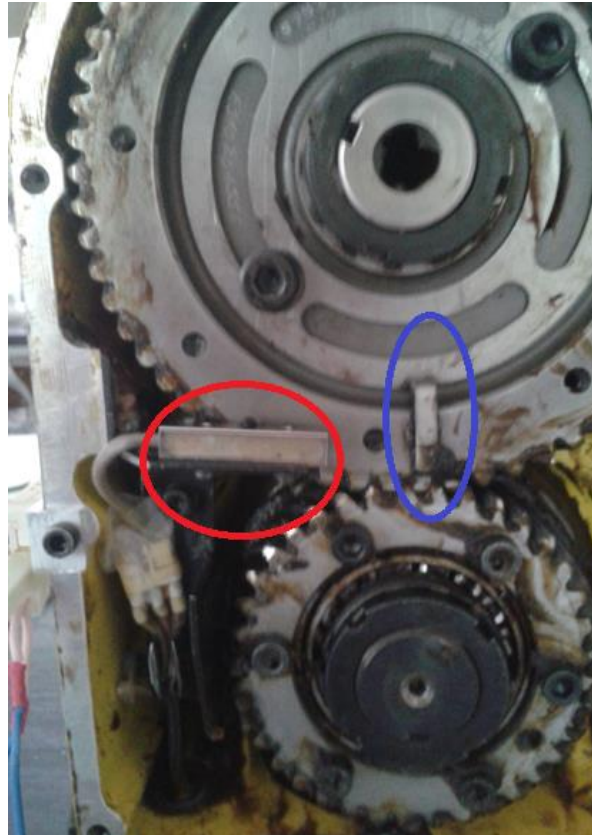


Ilustración 59 - Sensor inductivo real del eje 6.

Las señales de los dos sensores inductivos se conectan, a través de tres inversores (puesto que es una señal activa por bajos), a las interrupciones 2 y 3 del 8259 maestro. Para evitar problemas de rebote de señal se instaló un condensador en paralelo para cada uno de los sensores inductivos de los dos ejes de la muñeca. En la imagen inferior se observa el circuito montado.

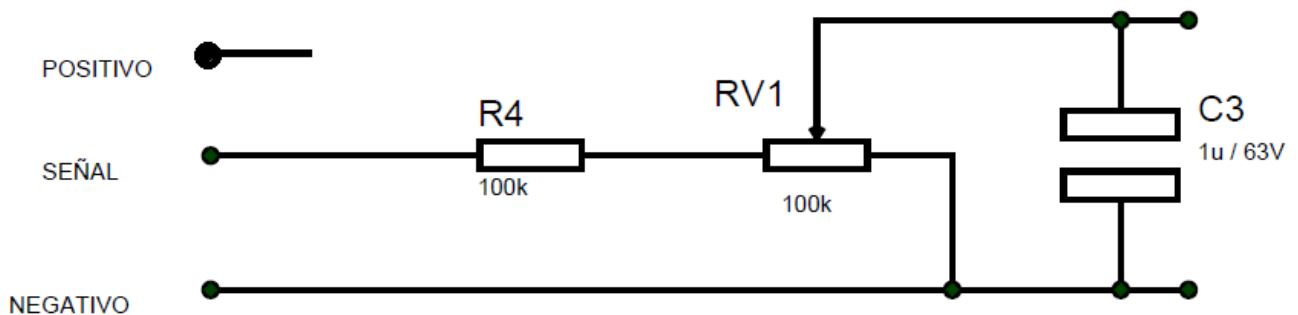


Ilustración 60 - Circuito de un sensor inductivo con condensador antirebote.

Los sensores inductivos de los dos ejes de la muñeca se alimentan a través de una fuente de alimentación externa a 24V de corriente continua. En la imagen inferior se observa dicha fuente.



Ilustración 61 - Fuente de alimentación externa de los sensores inductivos.

2.5.- SENSORES FINALES DE CARRERA

Estos sensores se incluyen dentro de los sensores de proximidad por contacto. Se utilizan en robótica para obtener información asociada con el contacto entre una mano manipuladora y objetos en el espacio de trabajo. En este caso en específico los sensores finales de carrera se agregaron en el marco del nuevo sistema de control diseñado. Se instalaron en total dos sensores finales de carrera, uno a cada lado del eje 5 del robot. Su objetivo es delimitar el espacio de trabajo de este eje, principalmente por razones de seguridad. Los finales de carrera son esencialmente conmutadores que responden ante la presencia o ausencia de un objeto. Para su correcta inserción dentro del sistema debió de instalarse una especie de pletina que será el objeto que impactará con el eje 5 de la muñeca y producirá la activación del sensor. En la imagen inferior se observa un esquema del montaje.

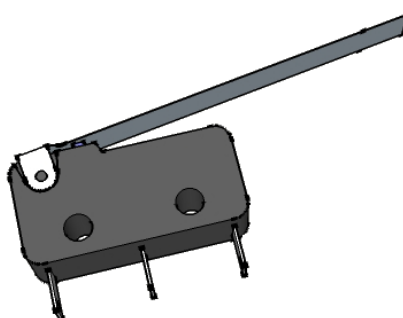


Ilustración 62 - Sensor final de carrera con pletina de contacto.

El montaje real se realizó con una goma como se observa en la imagen inferior.



Ilustración 63 - Montaje del sensor final de carrera de la muñeca del robot.

En la imagen inferior se observa la posición de los sensores en la muñeca del robot.



Ilustración 64 - Posición de los sensores finales de carrera en la muñeca del robot.

Las señales de los sensores finales de carrera se conectan, a través de dos inversores, a las interrupciones 0 y 1 del 8259 maestro. Para evitar problemas de rebote de señales se debieron de instalar dos condensadores en paralelo con la señal al igual que en el caso de los sensores inductivos. En la imagen inferior se observa el circuito montado.

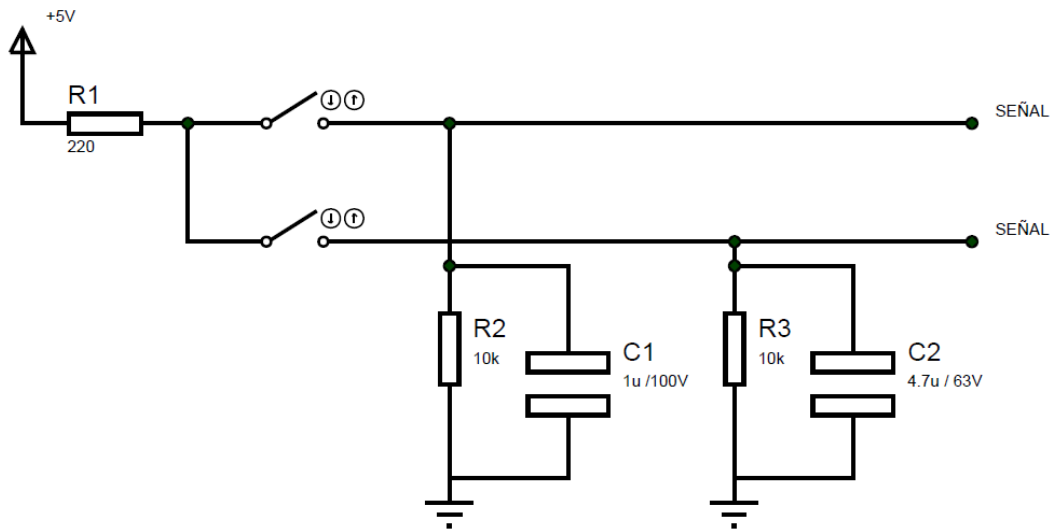


Ilustración 65 - Circuito de los finales de carrera con condensadores antirebote.

En las imágenes inferiores se observa el rango de movimiento de 180° del eje 5.

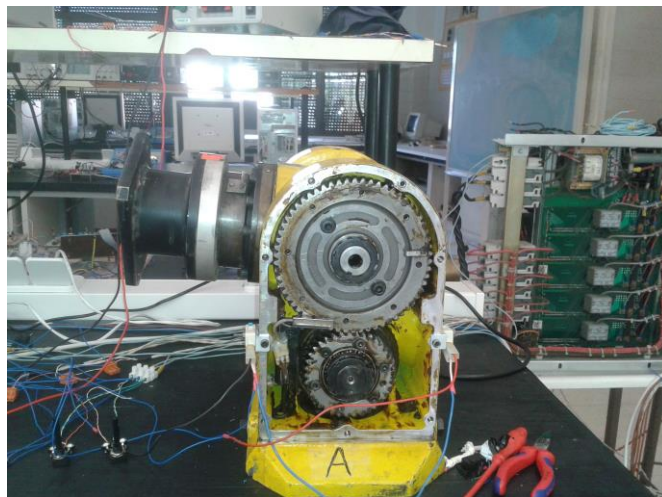


Ilustración 66 - Rango de movimiento de 180° del eje 5. Derecha.

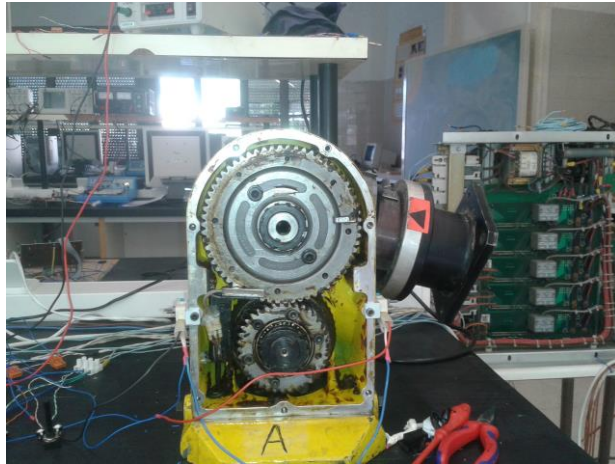


Ilustración 67 - Rango de movimiento de 180° del eje 5. Izquierda.

Capítulo III

Etapa de potencia

Sección I

INTRODUCCIÓN

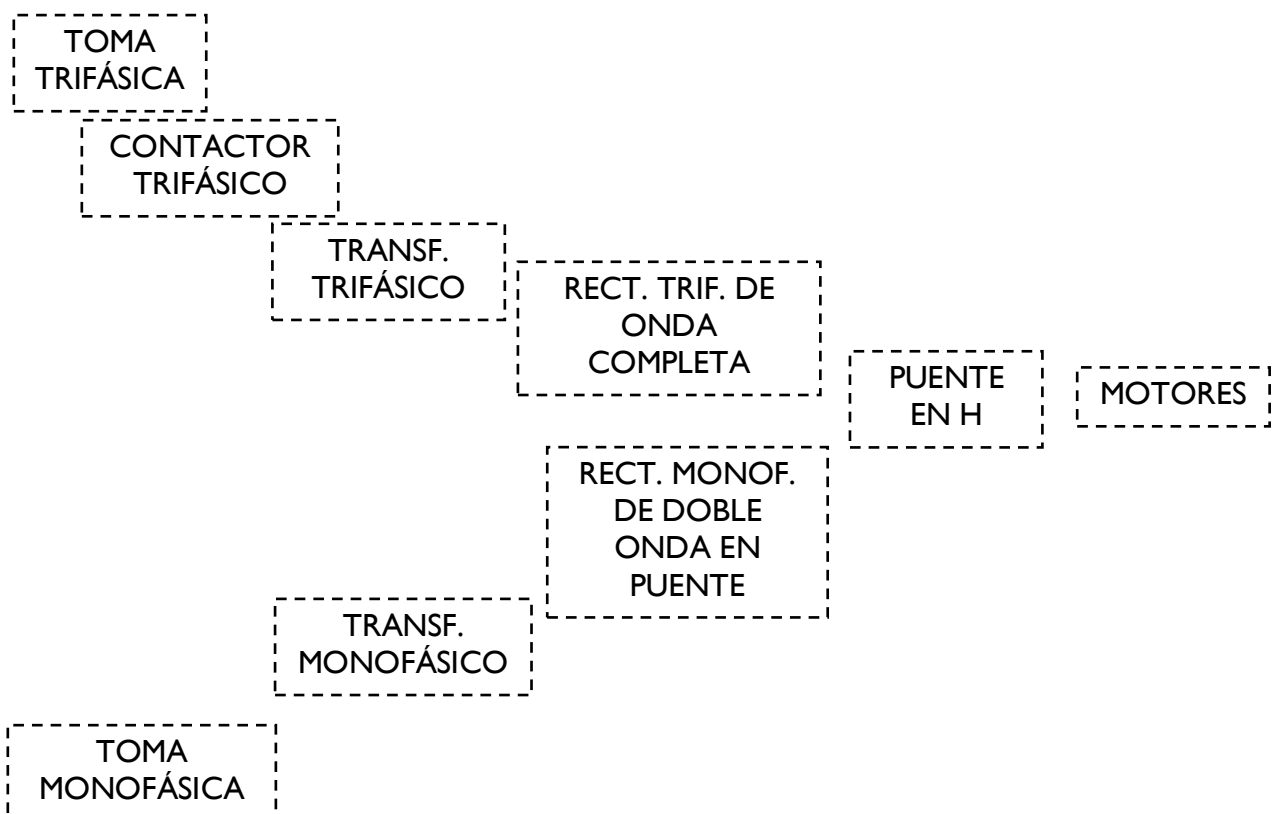
Las etapas de potencia tienen como misión proveer de la potencia necesaria al sistema para mover el manipulador final del brazo robótico. Las posibilidades para proveer esta fuerza son los sistemas eléctricos, neumáticos e hidráulicos. Dentro de estos tres se destacarán los eléctricos, puesto que es el sistema utilizado en el brazo robótico objeto de este proyecto. Los robots con actuadores eléctricos pueden utilizar motores pasos a paso, servomotores de corriente continua o servomotores de corriente alterna. Los sistemas de control y realimentación difieren dependiendo del motor usado. Los robots que usan motores paso a paso pueden funcionar con sistemas de control en lazo abierto. Sin embargo, son sistemas que tienen poca potencia y el control en lazo abierto da cabida a errores de posicionamiento. Los motores de corriente continua y alterna poseen sistemas de control en lazo cerrado. Las diferencias en relación a los motores convencionales similares se remiten al diseño del rotor (menor diámetro para tener un menor momento de inercia y un mejor control), a un mayor diámetro del alambre de la bobina para una menor pérdida de energía en forma de calor debido al efecto Joule y a una mayor sensibilidad a cambios en la tensión para un mejor control del par y velocidad, especialmente en el arranque.

El robot ACMA X58 usa servomotores de corriente continua de imán permanente, los más comunes en este tipo de robots de gama alta en su época (los años ochenta). Las características de velocidad y par presentadas por estos motores tienen un carácter bastante lineal. En el marco del sistema de control se busca que el motor usado responda linealmente a cambios de tensión, los sistemas de control pueden corregir las desviaciones de linealidad, pero se prefiere no corregir fallas de hardware mediante software. En cuanto a la forma en que se transmite la potencia, es importante contar con reductores potentes, pero de peso y tamaño reducidos, para no perder potencia ni capacidad de carga. El sistema mecánico de transmisión de potencia se explicó en detalle en el apartado anterior. Para que la etapa de potencia cumpla con el objetivo de proveer de la potencia necesaria a los motores de corriente continua del brazo robótico es necesario poseer una fuente de alimentación adecuada, tanto para la alimentación de los motores como de toda la circuitería electrónica de control asociada a su movimiento. Así, la alimentación de esta etapa de potencia está constituida por dos sistemas independientes. Por un lado, se encuentra el sistema de alimentación dirigido exclusivamente a los motores. Por otro, el sistema de alimentación de

toda la circuitería electrónica. La circuitería electrónica está compuesta por una serie de placas de circuito impreso en las que se encuentra implementado físicamente todo el sistema de control original de cada uno de los ejes del robot. Como el brazo robótico cuenta únicamente con dos ejes, sólo se utilizan dos de las placas de control y dos de las placas que implementan el puente en H, estructura que se explica al final de este capítulo.

El sistema de alimentación de los motores (alimentación de corriente continua) está constituido por una toma de corriente trifásica, un contactor trifásico, un transformador trifásico estrella-estrella y un rectificador trifásico de onda completa con filtro capacitivo compuesto por tres condensadores de 3300uF y 250V conectados en paralelo y cuya misión es la estabilizar la tensión de salida del rectificador. Todos estos elementos permiten transformar la corriente alterna trifásica con $U=400V$ en una corriente continua de aproximadamente 160V. Estos 160V se aplicarán a la estructura del puente en H encargado de gestionar el movimiento directo de los motores, su velocidad y sentido de rotación, a través de la conmutación de transistores tipo BJT.

El sistema de alimentación de la circuitería electrónica (alimentación monofásica) está compuesto por una toma de corriente monofásica con $U=230V$, un transformador monofásico y un rectificador monofásico de doble onda en puente con filtro capacitivo a la salida (condensador de 22000 uF y 40V). De esta forma, se consiguen las tensiones necesarias para la alimentación de los circuitos electrónicos: 5V, +15V, -15V, 12V y la masa del circuito. El siguiente esquema muestra las relaciones entre los distintos elementos de la etapa de potencia mencionados anteriormente.



Es importante destacar que el sistema de control de la etapa de potencia diseñado se integra dentro del sistema de control original del robot de forma que mantiene algunos de sus elementos mientras que añade nuevos elementos al control. Estos elementos añadidos permiten el conocimiento y manipulación de las variables de velocidad y posición. En la imagen inferior se observa un esquema de control de la etapa de potencia original.

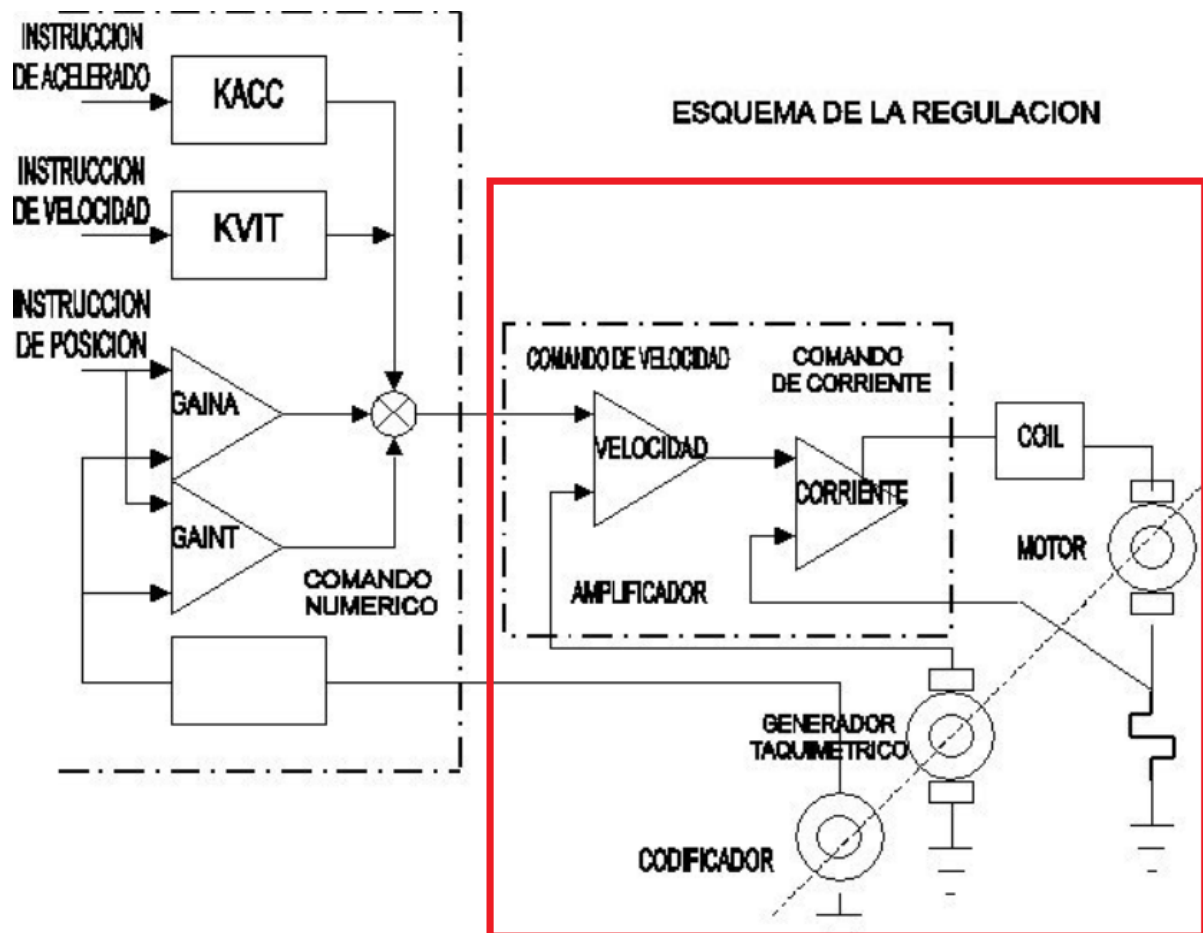


Ilustración 68 - Esquema de control original de la etapa de potencia de robot.

Los elementos fuera del cuadrado rojo corresponden con el sistema de control original del robot, que actualmente se encuentra en desuso. Por ejemplo, las señales de los encoder eran tratadas en esta parte del sistema de control. Dentro del cuadrado rojo se observa parte del bucle de control del sistema original que aún se mantiene dentro del sistema de control diseñado. El nuevo sistema de control adquiere señales de posición a través de la salida de los encoder de cada uno de los motores y envía la consigna de velocidad al bucle de velocidad que se observa en la imagen anterior. La señal de realimentación del bucle de corriente se obtiene de los sensores de corriente de cada uno de los motores y se realimenta internamente, sin interactuar con el sistema de control diseñado.

Sección II

ALIMENTACIÓN

Como se mencionó en la sección anterior de este capítulo, la alimentación de la etapa de potencia se divide en una alimentación dirigida a los motores y una alimentación monofásica. La alimentación trifásica está dirigida al puente en H, circuito mediante el cual puede controlarse la velocidad y sentido de giro de los motores basándose en la conmutación de transistores tipo MOSFET. La alimentación monofásica tiene como objetivo proveer a la circuitería electrónica del control de los motores de las tensiones de alimentación necesarias: +5V, +15V, -15V, +12V y la masa del circuito.

1.- ALIMENTACIÓN MONOFÁSICA

La alimentación monofásica de la etapa de potencia es el sistema encargado de proporcionar las tensiones de alimentación que necesita toda la circuitería de control. A continuación se detallan sus distintos componentes.

1.1.- TRANSFORMADOR MONOFÁSICO

Un transformador es un dispositivo que cambia potencia eléctrica alterna de un nivel de voltaje a potencia eléctrica alterna de la misma frecuencia pero de otro nivel de voltaje mediante la acción de un campo magnético. El devanado primario se conecta a una fuente de energía eléctrica alterna y el secundario suministra energía eléctrica a la carga. En este caso la tensión en el secundario del transformador pasa a través del rectificador monofásico de doble onda en puente y un transformador monofásico obteniéndose tensión continua para aplicar a la circuitería electrónica de control.

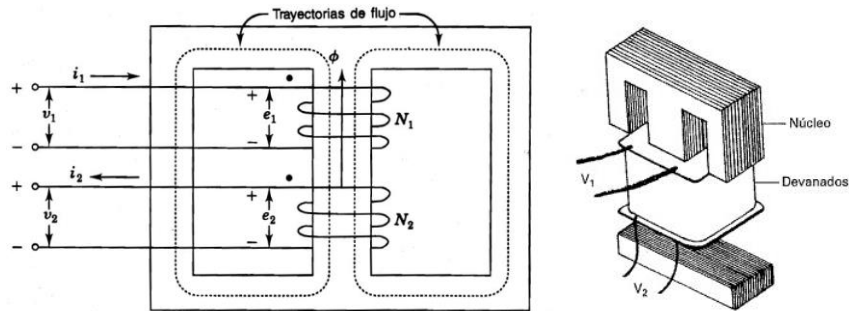


Ilustración 69 - Esquema transformador monofásico

En la siguiente imagen se observa el transformador monofásico real de la etapa de potencia.

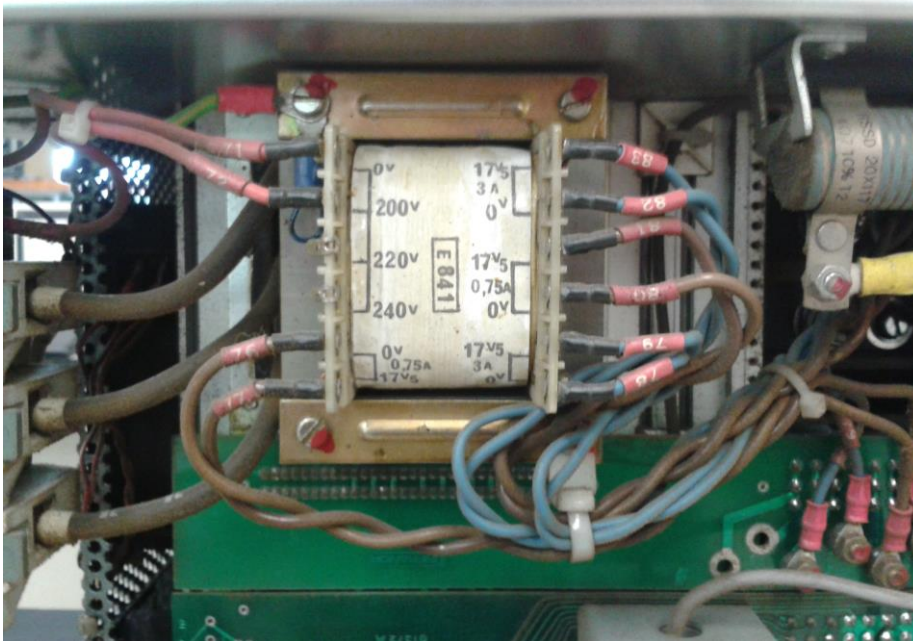


Ilustración 70 - Transformador monofásico real del proyecto

1.2.- RECTIFICADOR MONOFÁSICO DE DOBLE ONDA EN PUENTE

Este rectificador recibe el nombre de puente rectificador por estar formado por cuatro diodos conectados en puente. El circuito que se describirá es el siguiente:

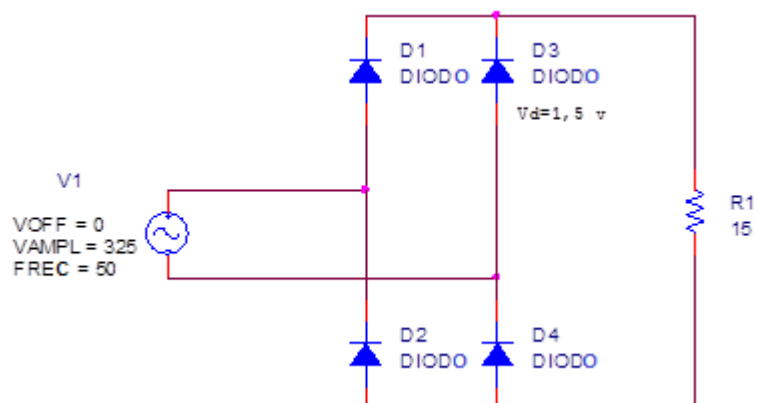


Ilustración 71 - Circuito Rectificador monofásico de onda completa

El paso de corriente alterna a corriente continua se consigue mediante la polarización en directa y en inversa de los pares de diodos D1/D4 y D3/D2 alternativamente. Dicha polarización la proporciona la fuente de tensión alterna al conmutar sus polaridades. De esta forma, la intensidad del circuito sigue uno de los dos caminos posibles según los diodos que conduzcan:

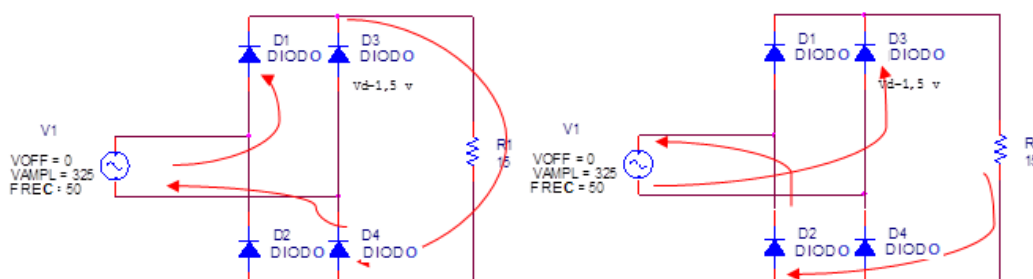


Ilustración 72 - Rectificador monofásico de doble onda en puente. Semiciclos

- Semiciclo positivo:
D1/D4 están polarizados directamente y por lo tanto conducen.
- Semiciclo negativo
D3/D2 conducen, D1/D4 están polarizados en inversa.

Los diodos D1/D4 conducen al mismo tiempo, al igual que D3/D2. La tensión de la carga, por lo tanto, es positiva durante la conducción de D1/D4 y negativa cuando conducen D3/D2.

La tensión inversa de pico de los diodos equivale a la tensión máxima del generador cambiada de signo ($-V_m$). Esta es la tensión que soportan la pareja de diodos durante el semiciclo en el que no conduzca. Este valor hace de este rectificador más adecuado para aplicaciones de alta tensión debido a la menor tensión de pico en bornes de los diodos (en el caso del rectificador monofásico de doble onda con transformador de toma intermedia es de $-2 \cdot V_m$). En la siguiente imagen se observa la corriente que entra al puente proveniente del generador en cada semiciclo:

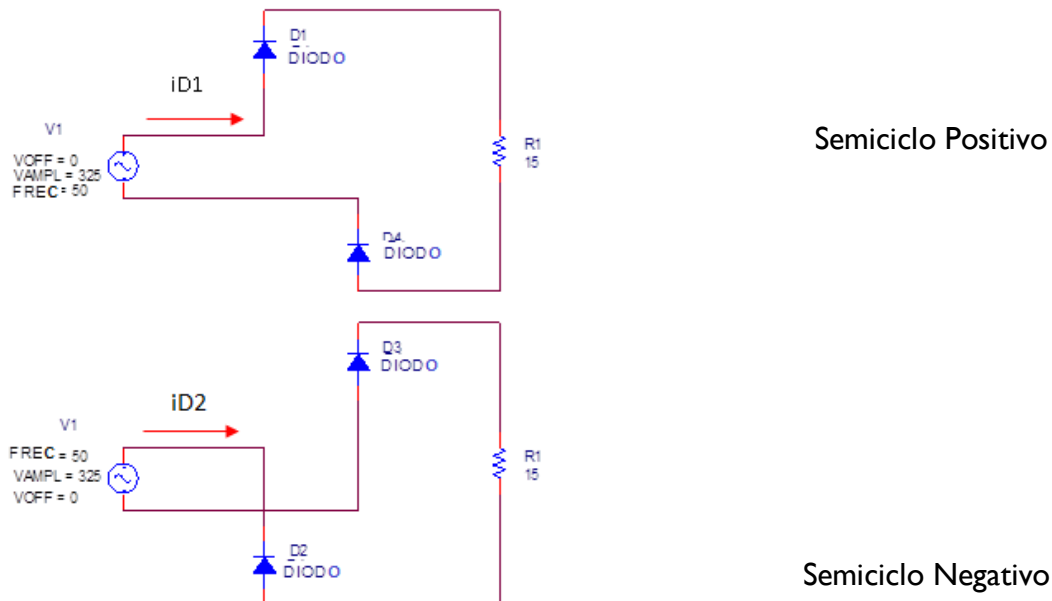


Ilustración 73 - Rectificador monofásico de onda completa. Corriente en los semiciclos

La corriente eficaz del generador es la misma que la corriente eficaz de la carga. La corriente del generador es la misma que la corriente de carga para la mitad del periodo del generador (i_{D1}); para la otra mitad es igual que la corriente de carga, pero con signo negativo ($-i_{D2}$). Los cuadrados de las corrientes de carga y del generador son iguales, por lo que las corrientes eficaces también lo serán. Además, las dos corrientes son simétricas respecto de cero por lo que la corriente media del generador es cero. En definitiva, se trata de dos circuitos serie y por lo tanto siempre será la misma intensidad.

La frecuencia a la salida es la mitad de la frecuencia de entrada, ya que se originan dos periodos a la salida para cada periodo a la entrada.

La segunda Ley de Kirchhoff describe la corriente que pasa por el circuito teniendo en cuenta que la carga del circuito es puramente resistivo:

$$V_m * \text{sen}(wt) = R * i$$

Para aplicar a la carga la corriente continua a la salida del rectificador es necesario incluir un filtro capacitivo que mantenga constante la tensión de salida. El filtro está constituido por un condensador de 22000 μF y 40V. El circuito completo se observa en la imagen inferior.

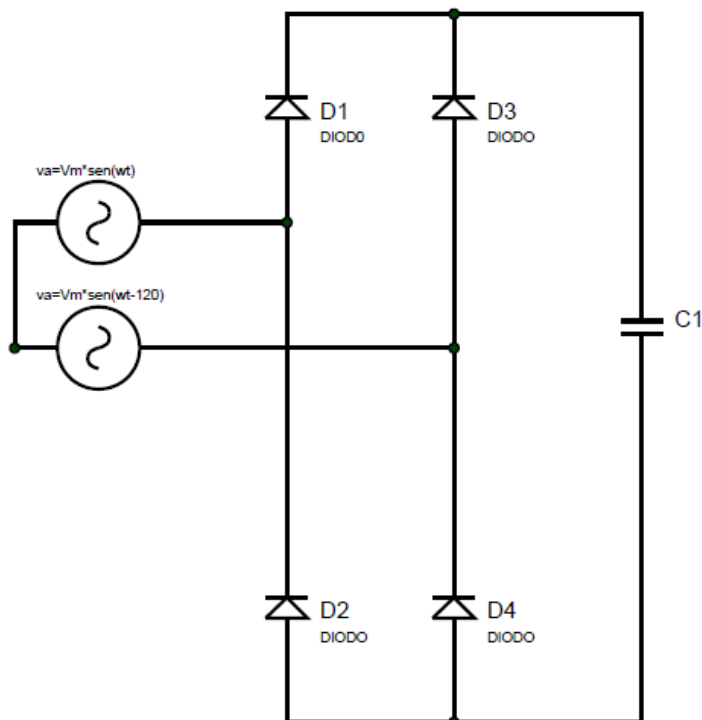


Ilustración 74 - Rectificador monofásico con filtro capacitivo.

Este filtro se encuentra implementado en las placas de control de la etapa de potencia original de la etapa del robot. Específicamente corresponde con la parte del circuito del Plano I2 explicado en la siguiente sección de este circuito.

2.- ALIMENTACIÓN DE LOS MOTORES

La alimentación de corriente continua de los motores está compuesta por una serie de elementos que permiten obtener una tensión continua de 160V a partir de una toma de corriente trifásica de $U=400V$. A continuación se describen en detalle cada uno de los elementos que componen este sistema de alimentación.

2.1.- CONTACTOR TRIFÁSICO

Un contactor es un componente electromecánico que tiene por objetivo establecer o interrumpir el paso de corriente, ya sea en el circuito de potencia o en el circuito de mando, tan pronto se dé tensión a una bobina. Un contactor es un dispositivo con capacidad de cortar la corriente eléctrica de un receptor o instalación, con la posibilidad de ser accionado a distancia, que tiene dos posiciones de funcionamiento: una estable o de reposo, cuando no recibe acción alguna por parte del circuito de mando, y otra inestable, cuando actúa dicha acción. Este tipo de funcionamiento recibe el nombre de "todo o nada".

Si bien constructivamente son similares a los relés, no son lo mismo. Su diferencia radica en la misión que cumple cada uno: ambos permiten controlar en forma manual o automática, ya sea localmente o a distancia toda clase de circuitos, pero mientras que los relés controlan corrientes de bajo valor los contactores se utilizan como interruptores electromagnéticos en la conexión y desconexión de circuitos de iluminación y fuerza motriz

de elevada tensión y potencia, como es el caso. En la imagen inferior se observa el esquema básico del contactor trifásico.

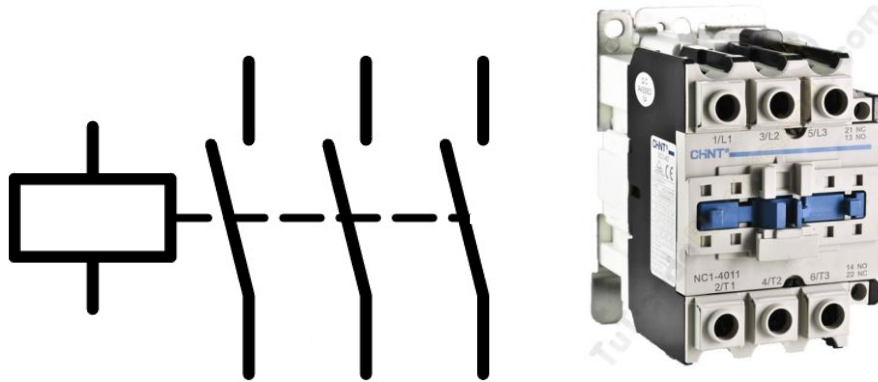


Ilustración 75 - Contactor trifásico.

2.2.- TRANSFORMADOR TRIFÁSICO

Al igual que el transformador monofásico, un transformador trifásico es un dispositivo que cambia potencia eléctrica alterna de un nivel de voltaje a potencia eléctrica alterna de la misma frecuencia pero de otro nivel de voltaje mediante la acción de un campo magnético. En este caso la tensión en el secundario del transformador pasa a través del rectificador trifásico de onda completa, obteniéndose tensión continua para aplicar al motor. Entendemos por transformador trifásico aquel que es utilizado para transformar un sistema trifásico equilibrado (corrientes son iguales y desfasados simétricamente) de tensiones en otro sistema equilibrado de tensiones trifásico pero con diferentes valores de tensiones e intensidades. Para conseguir ese propósito se pueden utilizar tres transformadores monofásicos, de manera que se tendrán tres núcleos magnéticos independientes y conexiones como indica la figura inferior.

El transformador trifásico del sistema de la muñeca del robot es de tipo acorazado. Así, consta de un núcleo laminado de tres columnas cuyas bobinas están enrolladas en la columna central. Las bobinas del primario y del secundario están físicamente enrolladas una sobre la otra, estando la bobina de menor voltaje situada en la parte interna más cercana al núcleo. Con esta disposición se simplifica el problema del aislamiento del devanado de alta tensión desde el núcleo. Además, resulta menor flujo disperso que en caso de disponer los dos devanados en el núcleo separados (transformador tipo núcleo).

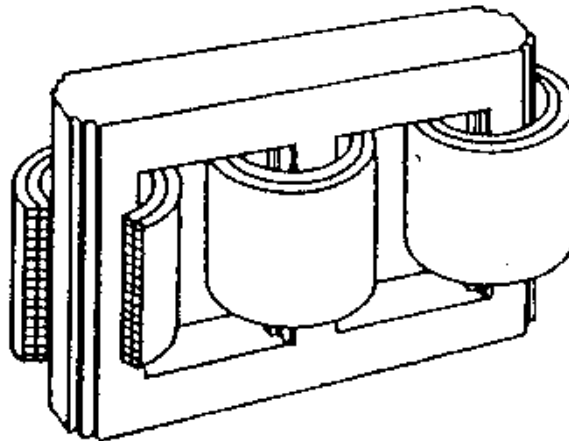


Ilustración 76 - Transformador trifásico de tipo acorazado

El conexionado de los devanados del transformador es del tipo estrella- estrella.

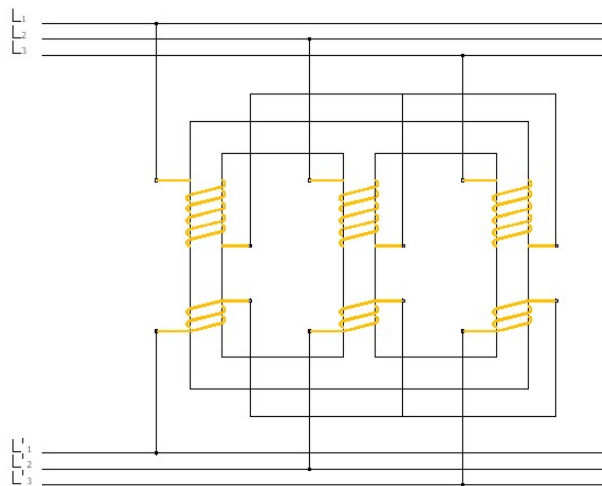


Ilustración 77 - Conexionado de los devanados del transformador trifásico estrella- estrella

A continuación se explican las relaciones entre las tensiones de línea y fase del primario y del secundario del transformador.

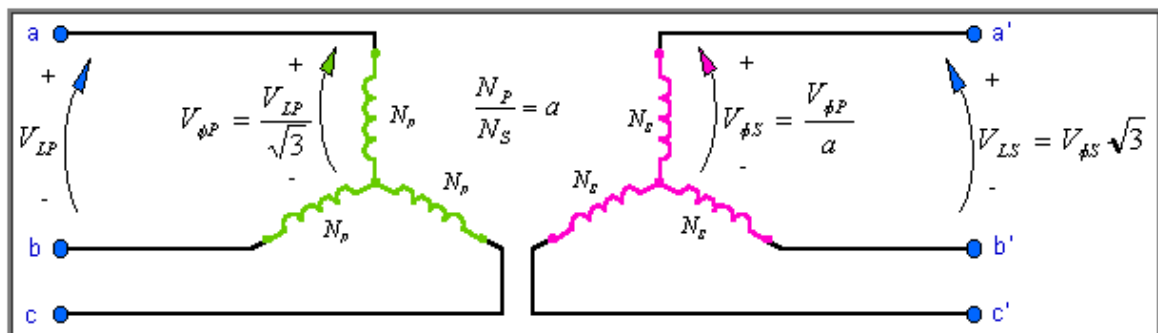


Ilustración 78 - Relaciones entre las tensiones de línea y fase

Las tensiones e intensidades primarias de línea y de fase cumplen las relaciones:

$$V_{LP} = \sqrt{3} V_{FP}$$

$$I_{LP} = I_{FP}$$

Las tensiones e intensidades de línea y de fase del secundario cumplen las relaciones:

$$V_{LS} = \sqrt{3} V_{FS}$$

$$I_{LS} = I_{FS}$$

La relación de tensiones de línea del primario y el secundario es:

$$\frac{V_{LP}}{V_{LS}} = a$$

En la siguiente imagen se observa el transformador trifásico real de la etapa de potencia.

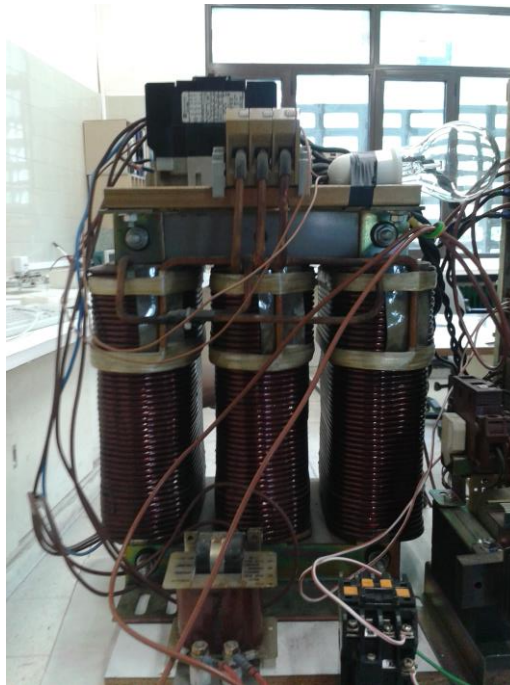


Ilustración 79 - Transformador trifásico real de la etapa de potencia



Ilustración 80 - Transformador trifásico. Detalle entrada primario.



Ilustración 81 - Transformador trifásico. Detalle salida secundario.

En la siguiente imagen se observa el sistema de alimentación del transformador trifásico.

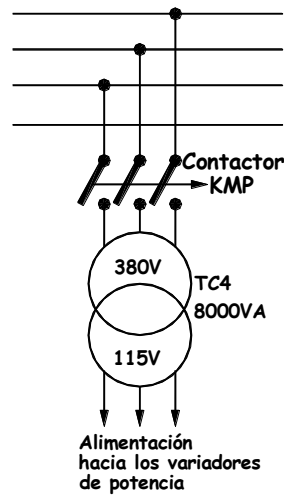


Ilustración 82 - Esquema eléctrico transformador trifásico.

2.3.- RECTIFICADOR TRIFÁSICO DE ONDA COMPLETA

El rectificador trifásico de onda completa es un circuito electrónico capaz de convertir una corriente alterna de entrada en una corriente continua de salida, mediante dispositivos semiconductores capaces de manejar grandes potencias como lo son los diodos de potencia. También se utilizan tiristores en el caso de rectificadores controlados. Los rectificadores trifásicos son alimentados por fuentes trifásicas, por lo que son más eficientes y pueden manejar grandes potencias, ya que en su salida presentan menor rizado de la señal. Son utilizados principalmente en la industria para producir voltajes y corrientes continuos que generalmente impulsan cargas de gran potencia, como motores de corriente continua.

En el marco de la etapa de potencia este rectificador tiene la función de producir tensión y corriente continua para la carga, en este caso el motor de corriente continua. Se considera que el generador trifásico está equilibrado y que la secuencia de fases es v_a, v_b, v_c .

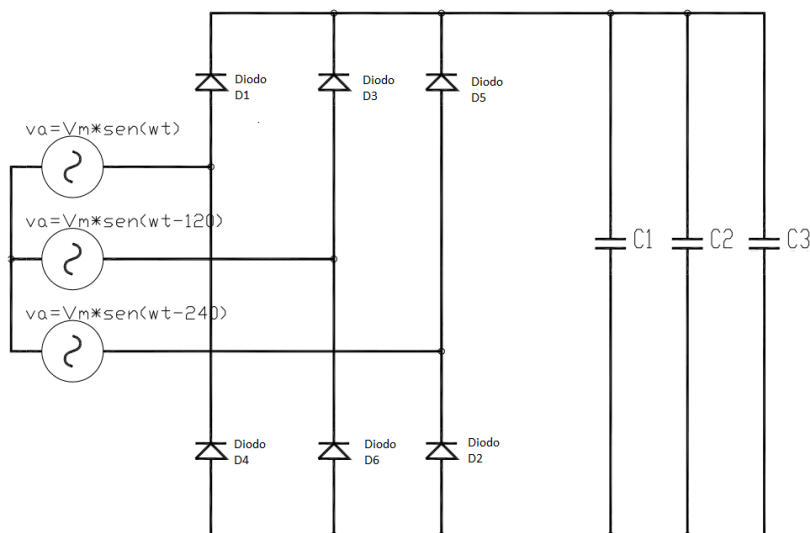


Ilustración 83 - Circuito Rectificador trifásico de onda completa

Sólo podrá conducir uno de los tres diodos de la mitad superior del puente: D1, D3 o D5. El diodo que conduzca será aquel que tenga su ánodo conectado a la tensión de fase de mayor valor en ese instante. Lo mismo sucede con los diodos de la mitad inferior del puente (D4, D6 y D2). El diodo en conducción tendrá su cátodo conectado a la tensión de fase de menor valor en ese instante. Además, la Ley de Kirchhoff para las tensiones demuestra que no pueden conducir al tiempo D1/D4, D3/D6 ni D5/D2; pues en estos casos la caída de tensión en el generador no sería la misma que la caída de tensión en la carga. Los diodos conducen por pares (6,1),(1,2),(2,3),(3,4),(4,5),(5,6),(6,1), activándose siguiendo la secuencia 1,2,3,4,5,6,1,... La tensión a la salida de la carga será una de las tensiones compuestas del generador, por lo tanto y al tratarse de un sistema trifásico, será igual a $\sqrt{3} * V_m$. La intensidad de corriente en el circuito posee seis “caminos” distintos, equivalentes a las seis combinaciones de tensiones compuestas existentes. Al considerar un periodo del generador de 360° se deduce que entre los picos de tensiones compuestas habrá 60. De ahí que este circuito se denomine rectificador de seis pulsos debido a las seis transiciones que se producen en cada periodo de la tensión del generador. Al realizarse la conexión de los diodos en puente se reduce la tensión inversa de pico que deben soportar los diodos haciendo el circuito más adecuado para aplicaciones de alta tensión. La segunda Ley de Kirchhoff para las tensiones describe la corriente por el circuito considerando el diodo ideal:

$$\sqrt{3} * V_m * \text{sen} \left(\omega t + \frac{\pi}{6} \right) = R * i_{ab}$$

Como se observa en el circuito de la imagen superior, la carga del rectificador trifásico está constituida por tres condensadores en paralelo. Esta distribución tiene como objetivo asegurar una tensión continua constante a la salida del rectificador. Cada uno de estos condensadores tiene una capacidad de 3300uF y 250V.



Ilustración 84 - Condensadores de la etapa de filtrado de la alimentación de los motores

2.3.1.- RESISTENCIA DE DESCARGA DEL FILTRO CAPACITIVO

Esta resistencia corresponde en la realidad a dos resistencias de potencia conectadas en serie de valor $4,7 \Omega$. Al poseer un valor óhmico tan bajo dejan pasar una gran cantidad de intensidad a través de ellas. Su propósito es el de servir de medio de descarga de los condensadores del filtro capacitivo. Una vez los condensadores se han descargado parcialmente un relé de conmutación las puentea. En la imagen inferior se observa un esquema del circuito.

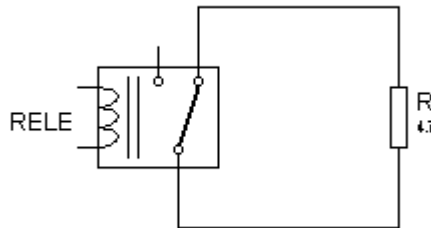


Ilustración 85 - Circuito resistencia de descarga del filtro capacitivo

En la imagen inferior se observa una de la resistencia descrita anteriormente. Se encuentra ubicada en la parte superior del reverso del armario de control.



Ilustración 86 - Resistencia en serie con el filtro capacitivo

3.- CONTROL DE LA ALIMENTACIÓN A TRAVÉS DE RELÉS

Todo el sistema de alimentación descrito anteriormente posee una etapa superior de control a través de relés. Esta etapa de control permite la activación de toda la etapa de potencia, tanto de la alimentación a 230V de toda la parte de control electrónico como la alimentación trifásica de 400V. El sistema de control se diseñó con el objetivo de poder manipular la etapa de potencia de forma segura, puesto que las diversas pruebas para la puesta en marcha del sistema de control diseñado requirieron del encendido y apagado de la etapa de potencia numerosas veces.

El sistema de control de la alimentación está compuesto de tres relés SRD – 05VDC-SL-C de corriente continua. Los relés son activos por bajos, conmutan a 5V y permiten el paso de hasta 10A. Dentro de su circuito interno poseen optoacopladores que mejoran el aislamiento entre la etapa de control y la etapa de potencia. En la imagen inferior se observan los relés que en la realidad vienen implementados en un grupo de ocho y los interruptores de control, que pertenecen a un entrenador del laboratorio.

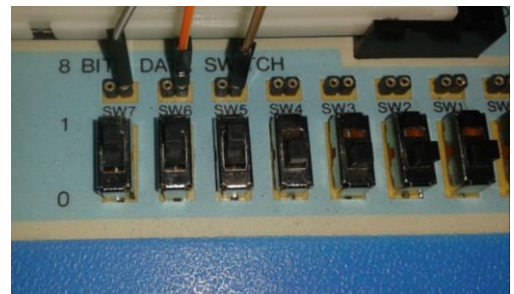
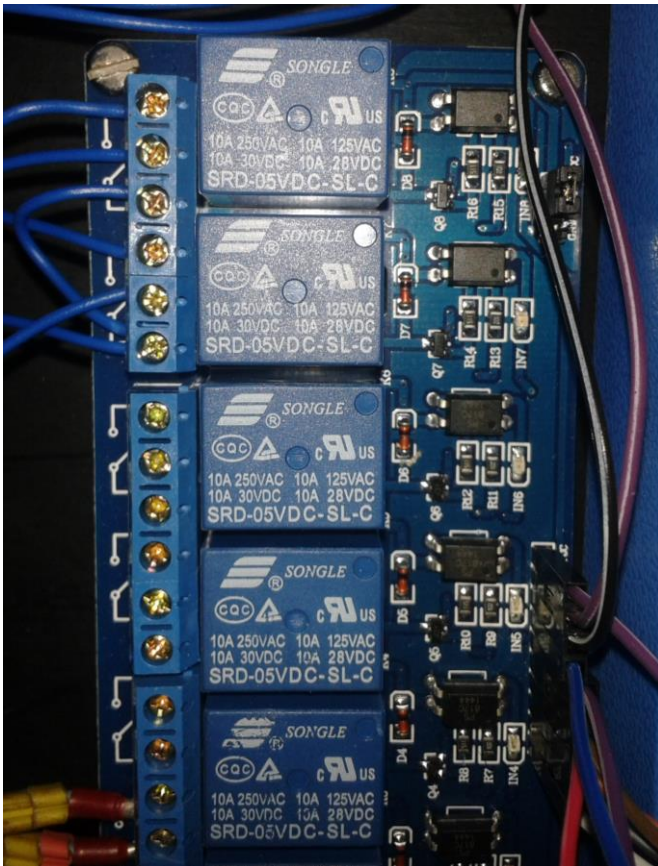


Ilustración 87 - Relés SRD-05VDC-SL-C para el control de la alimentación de la etapa de potencia. Interruptores de control.

En la imagen inferior se observa el circuito de los relés.

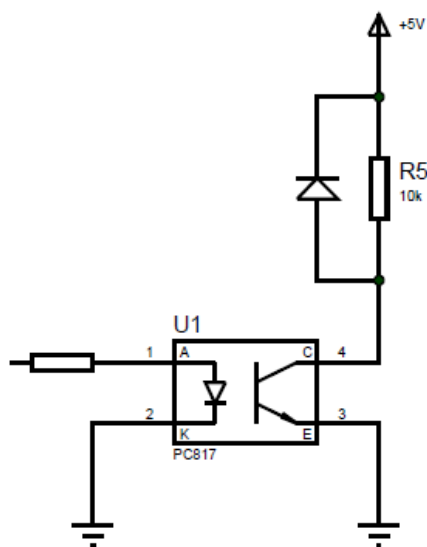


Ilustración 88 - Circuito relés de la etapa de potencia.

Cada uno de los relés implementados se encarga de un objetivo en concreto. Uno de ellos permite el paso de la alimentación de 230V de toda la electrónica de control original de los ejes del robot. Otro de los relés enclava el contactor trifásico del sistema de alimentación de potencia de los motores. Por último, un tercer relé envía al sistema de control la señal de “frenos desanclados”. Esta señal es una consigna de seguridad que permite detener todo el sistema de potencia sin necesidad de quitar la alimentación del mismo. De esta forma se evita someter al sistema a continuos encendidos y apagados durante las pruebas de puesta en marcha del sistema de control diseñado.

3.1.- RED SNUBBER

La red Snubber está compuesta por un conjunto de componentes que se incorporan al circuito de potencia para reducir el estrés eléctrico durante las conmutaciones y aseguran un régimen de trabajo seguro. En este caso fueron requeridas para evitar el efecto de la entrada en funcionamiento de la etapa de potencia sobre el microcontrolador. Cuando la etapa de potencia entra en funcionamiento se producen una serie de picos de corriente y tensión que afectan negativamente al sistema de control, provocando problemas como la reinicialización del microcontrolador.

La función principal del circuito es absorber la energía procedente de la etapa de potencia. La red protege el sistema de control de los picos y sobretensiones en la etapa de potencia. El circuito se implementa con un condensador de 100nF y una resistencia de 470 Ω en paralelo con la salida del relé. En la imagen inferior se observa la red a la salida del relé.

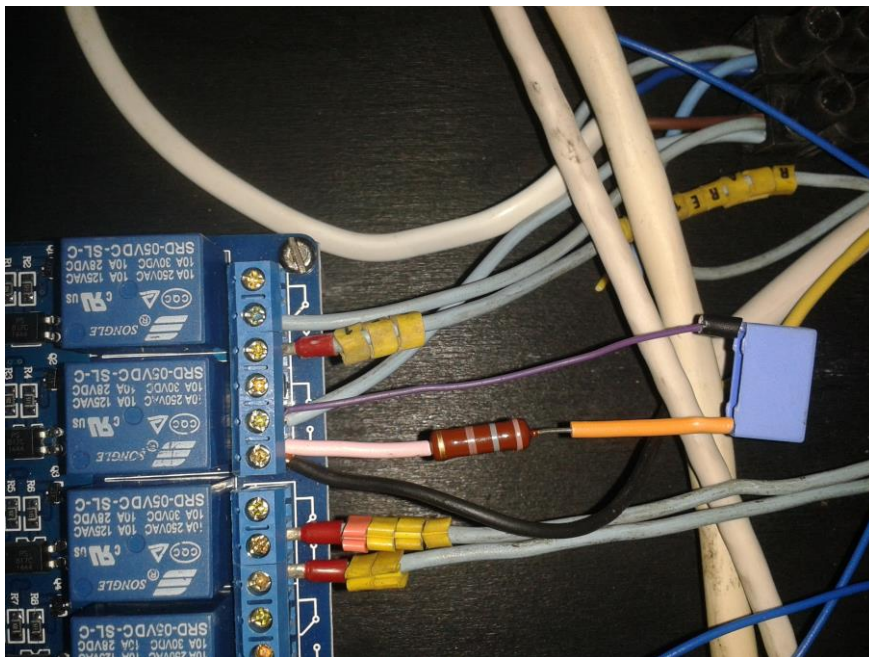


Ilustración 89 - Red Snubber a la salida de los relés de potencia.

Sección III

PLACAS ELECTRÓNICAS DE CONTROL ORIGINALES

1.-INTRODUCCIÓN

Cada uno de los seis motores originales del robot posee una placa de control y una placa de puente en H que conforman la unión entre las señales de control y la alimentación de potencia. La placa de control constituye el centro neurálgico de toda la información que gobierna la actuación sobre los robots. En el esquema de control actual la señal de control principal proviene de los generadores de corriente continua que funcionan como tacómetros de cada uno de los motores. En el caso de no recibir esta señal o recibir una señal errónea el circuito de control detiene los motores al impedir el funcionamiento de los puentes en H. Este proceso de control se logra mediante un circuito electrónico que se encuentra dividido para su estudio en doce planos.

Las señales de control atraviesan el circuito y en él se someten a procesos de tratamiento de señal, comparación con señales de frecuencia estable, chequeo de otras señales de verificación y finalmente, alimentación de los transformadores de impulsos conectados con el circuito tipo puente en H.

Las placas se encuentran ubicadas en un armario que les proporciona sujeción y conexión a las líneas de alimentación que se describieron en el capítulo anterior. La conexión entre las placas y los motores se realiza mediante un cable que sale de la parte frontal del armario y se conecta en la parte posterior de cada motor. En la imagen inferior puede observarse el armario de control en su parte frontal. En la siguiente imagen se observan las barras de conexión de la alimentación. Las placas de control se conectan por la parte posterior del armario.



Ilustración 90 - Armario de control.

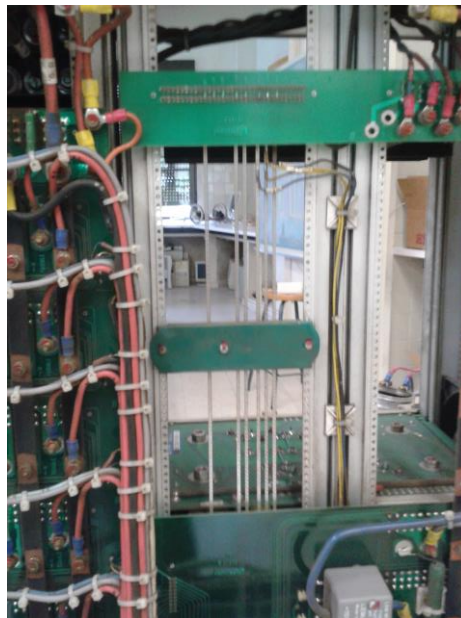
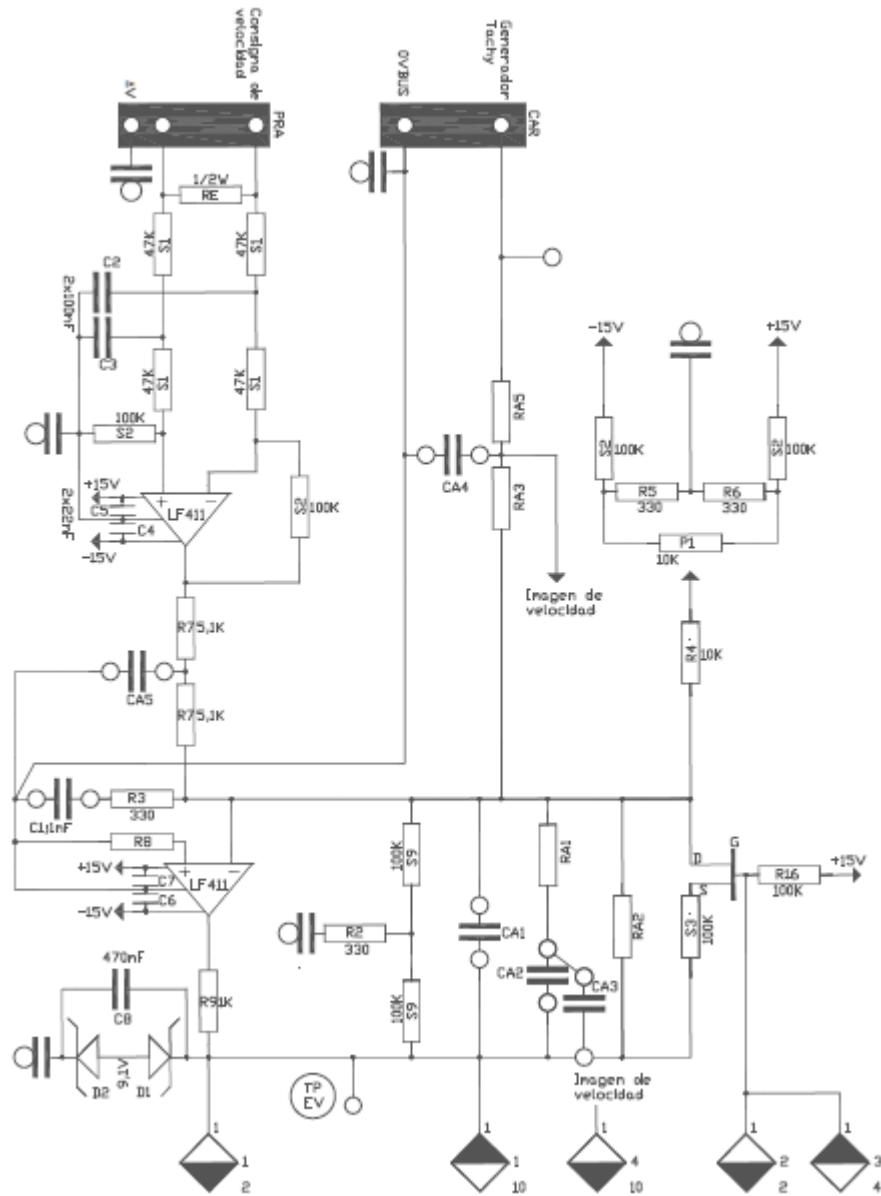


Ilustración 91 - Barras de alimentación posteriores del armario de control.

2.-PLACAS DE CONTROL

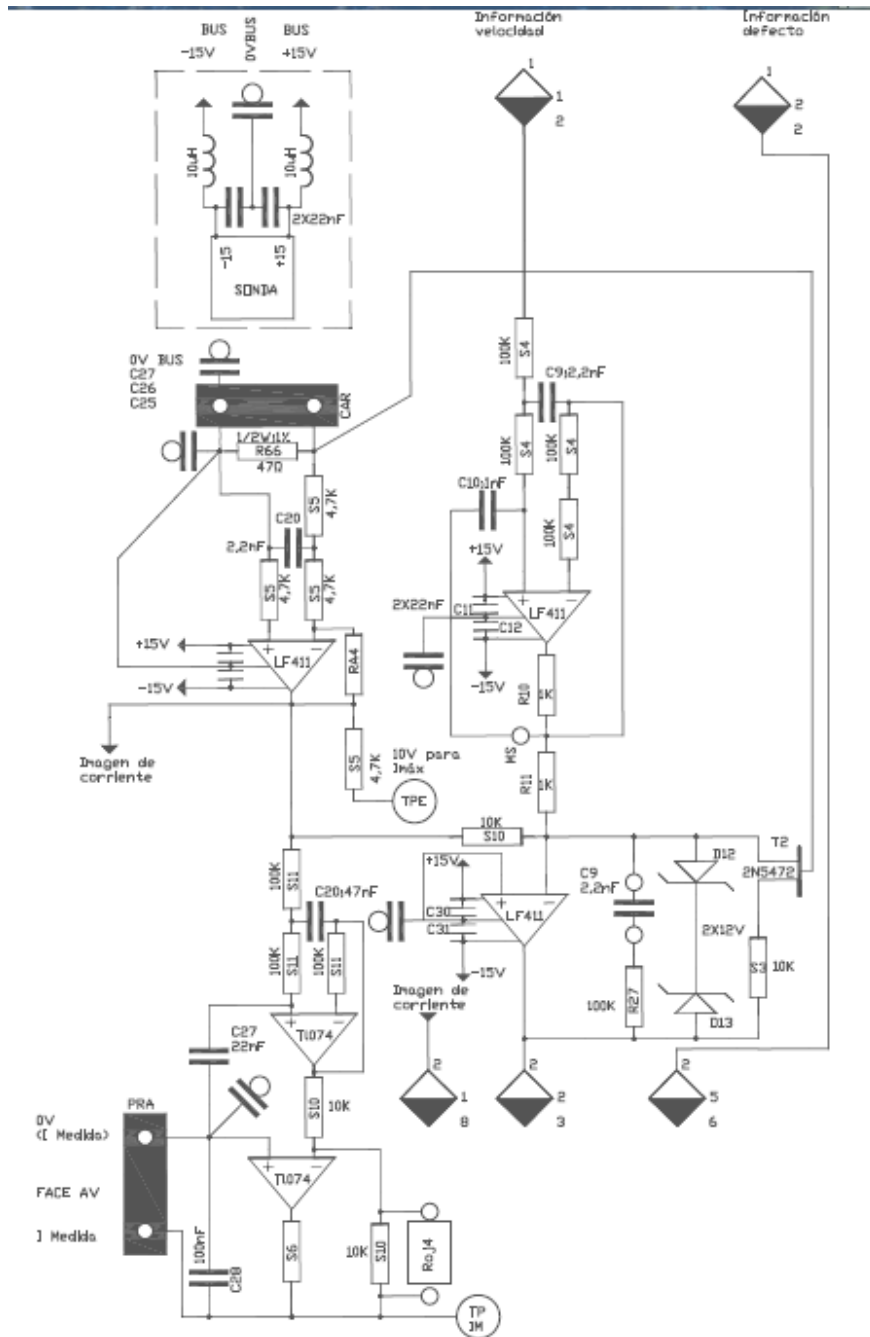
A continuación se describirá todo el circuito electrónico de control, tanto la placa de control como la placa de puente en H.

2.1.- PLANO N°1



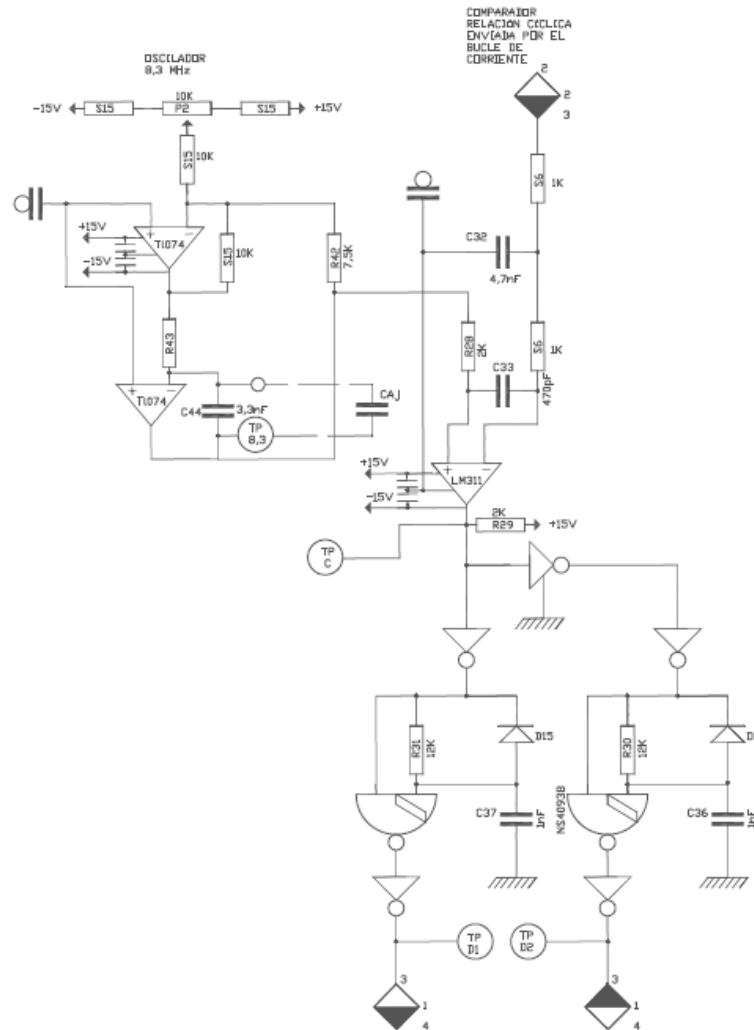
En esta parte del circuito las señales de entrada están constituidas por la señal de consigna de velocidad proveniente de las placas de control diseñadas (ver capítulo siguiente) y la señal de realimentación del tacómetro. Ambas señales son comparadas para generar una señal de control y actuar sobre el sistema.

2.2.- PLANO N°2



Las señales de entrada provienen del plano número I. En la esquina superior izquierda se observa el circuito sensor de corriente que se explicará al final de este capítulo. La señal proveniente del sensor, en caso de ser errónea, cortará la alimentación de consiga. En este punto la intensidad del bobinado de los motores aumentará hasta hacerlos detenerse por un exceso de la misma.

2.3.- PLANO N°3



En la esquina superior izquierda se observa un circuito oscilador de frecuencia igual a 8,3 kHz. La señal de salida del oscilador, que es de tipo triangular, se compara con la señal de salida del bucle de corriente. De esta forma se obtiene una modulación de pulso a la salida acorde a las necesidades de los motores a cada instante. Si la intensidad de los motores disminuye los pulsos serán mayores, en caso contrario los pulsos serán cada vez menores llegando incluso a detener los motores.

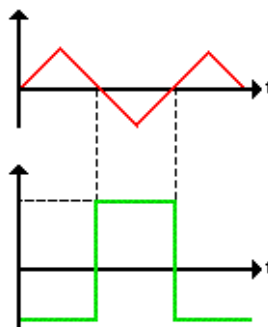
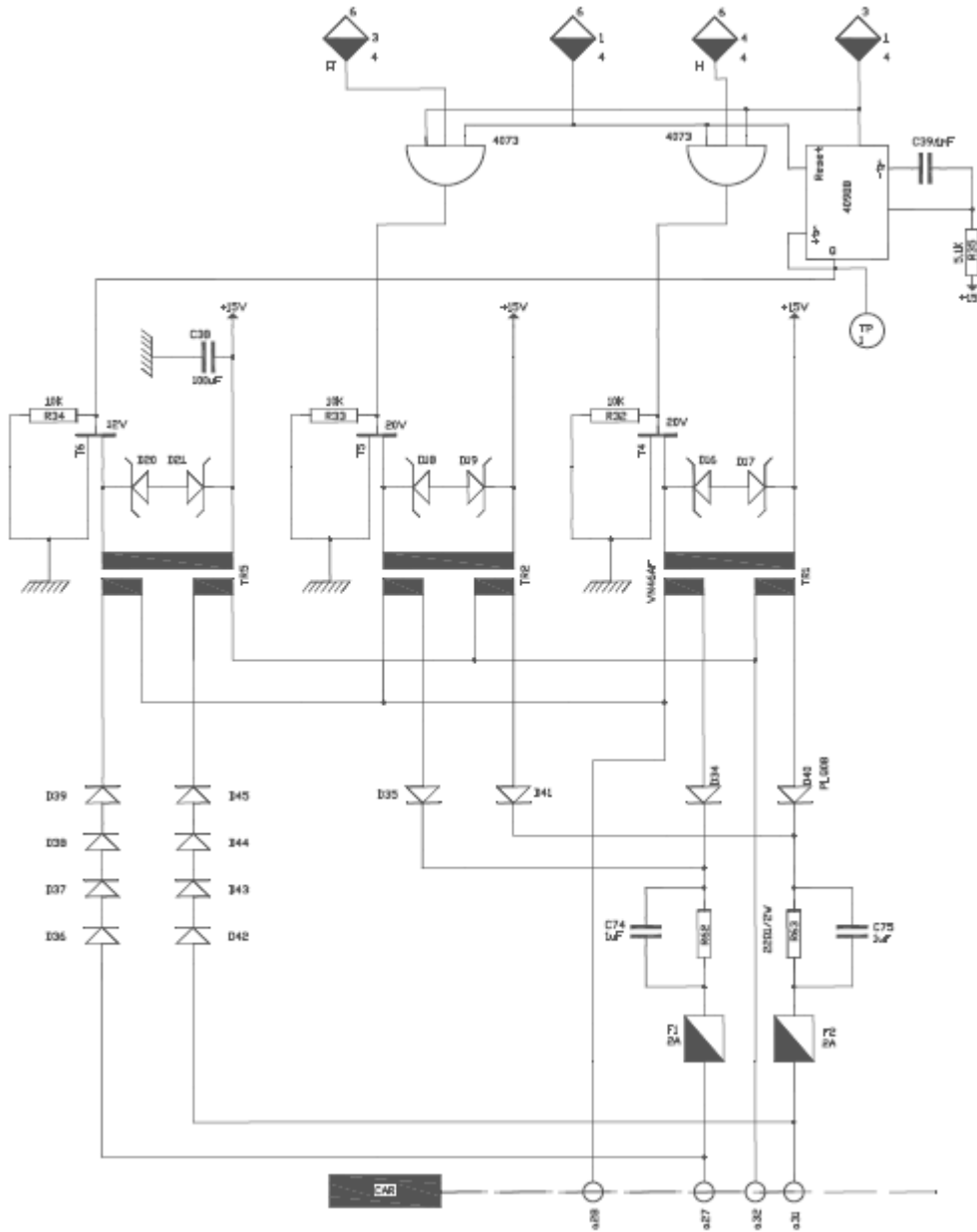


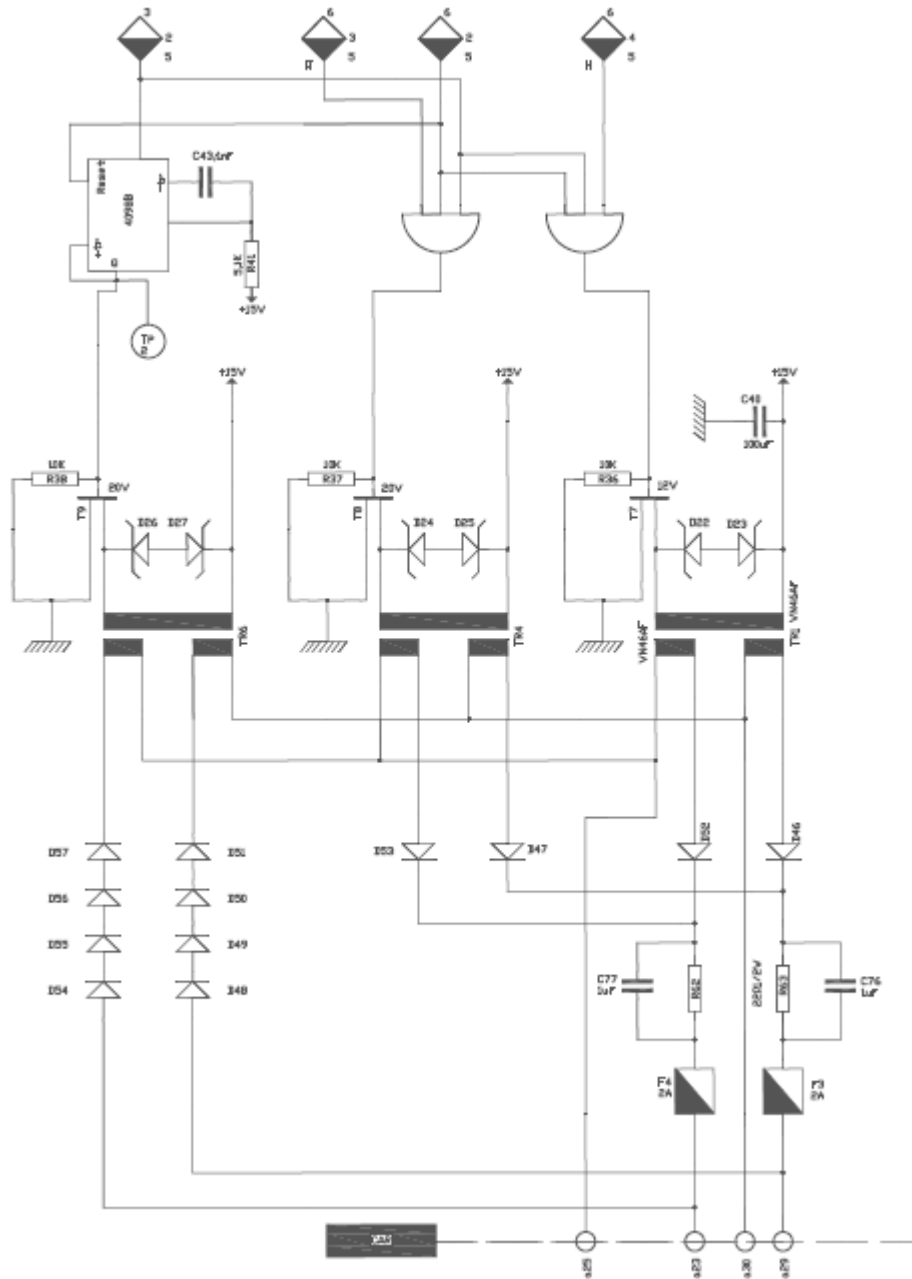
Ilustración 92 - Obtención de la señal de modulación de ancho de pulso

2.4.- PLANO N°4



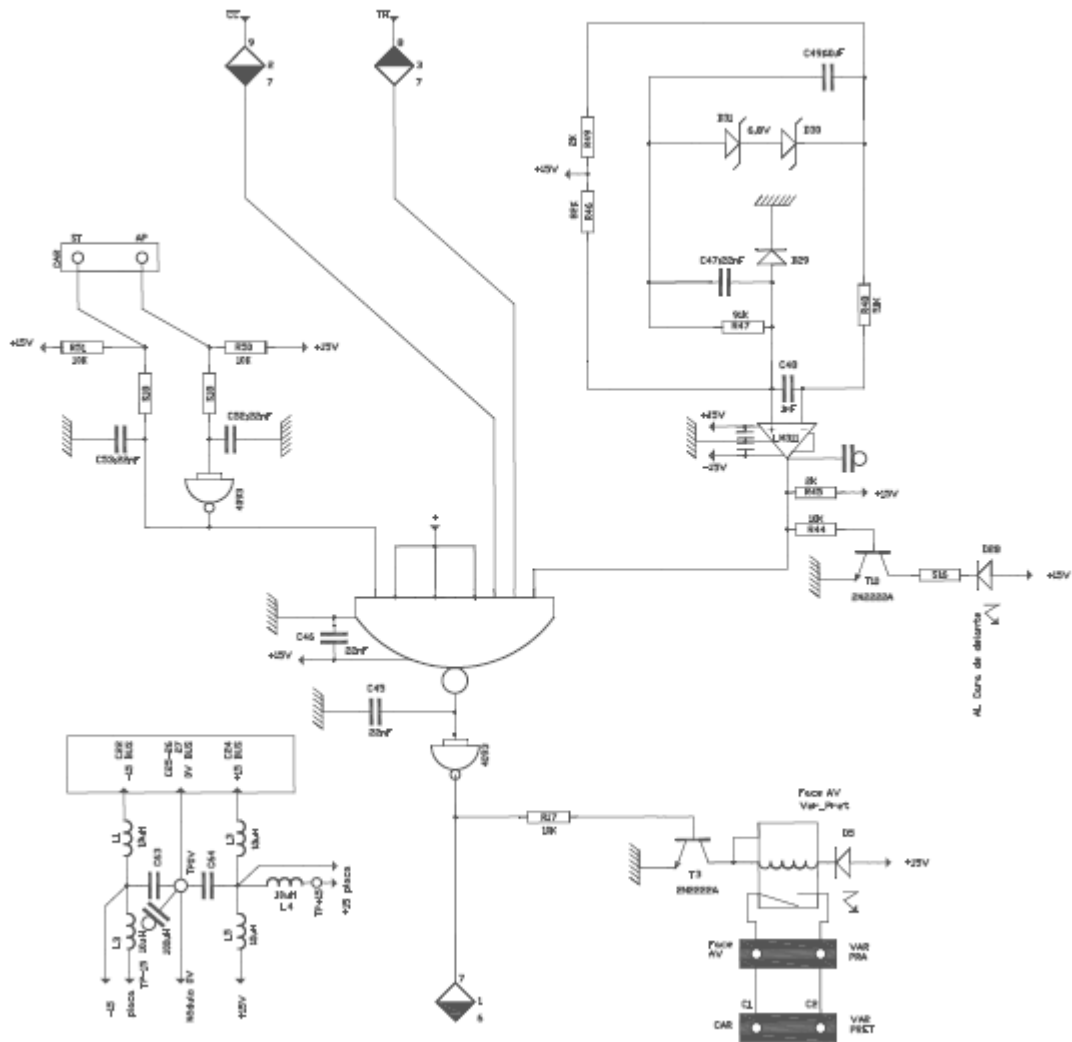
Las entradas de esta parte del circuito son las señales de control de velocidad provenientes del Plano 6 y una de las señales provenientes del bucle de corriente. El circuito 4098B realiza una comparación entre las señales de los osciladores de 400 kHz, 8,3 kHz y la señal de modulación de ancho de pulso. De esta forma se regula el funcionamiento de los transformadores de impulsos a través de un integrado siempre y cuando el sistema detecte que puede funcionar.

2.5.- PLANO N°5



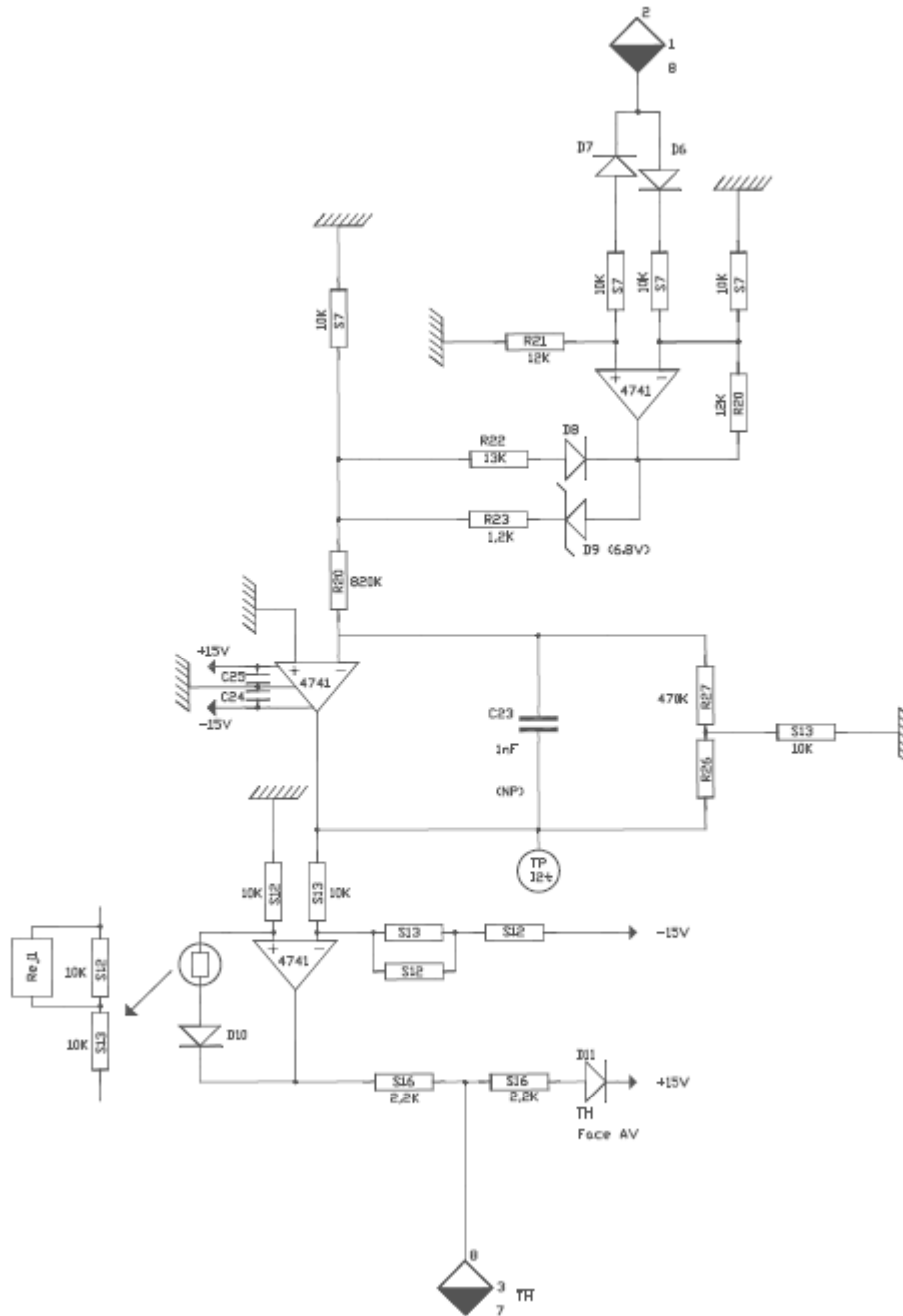
Este circuito contiene el segundo juego de transformadores de impulsos, encargados del movimiento de los motores en el otro sentido de giro. Su funcionamiento es análogo al del anterior.

2.7.- PLANO N°7



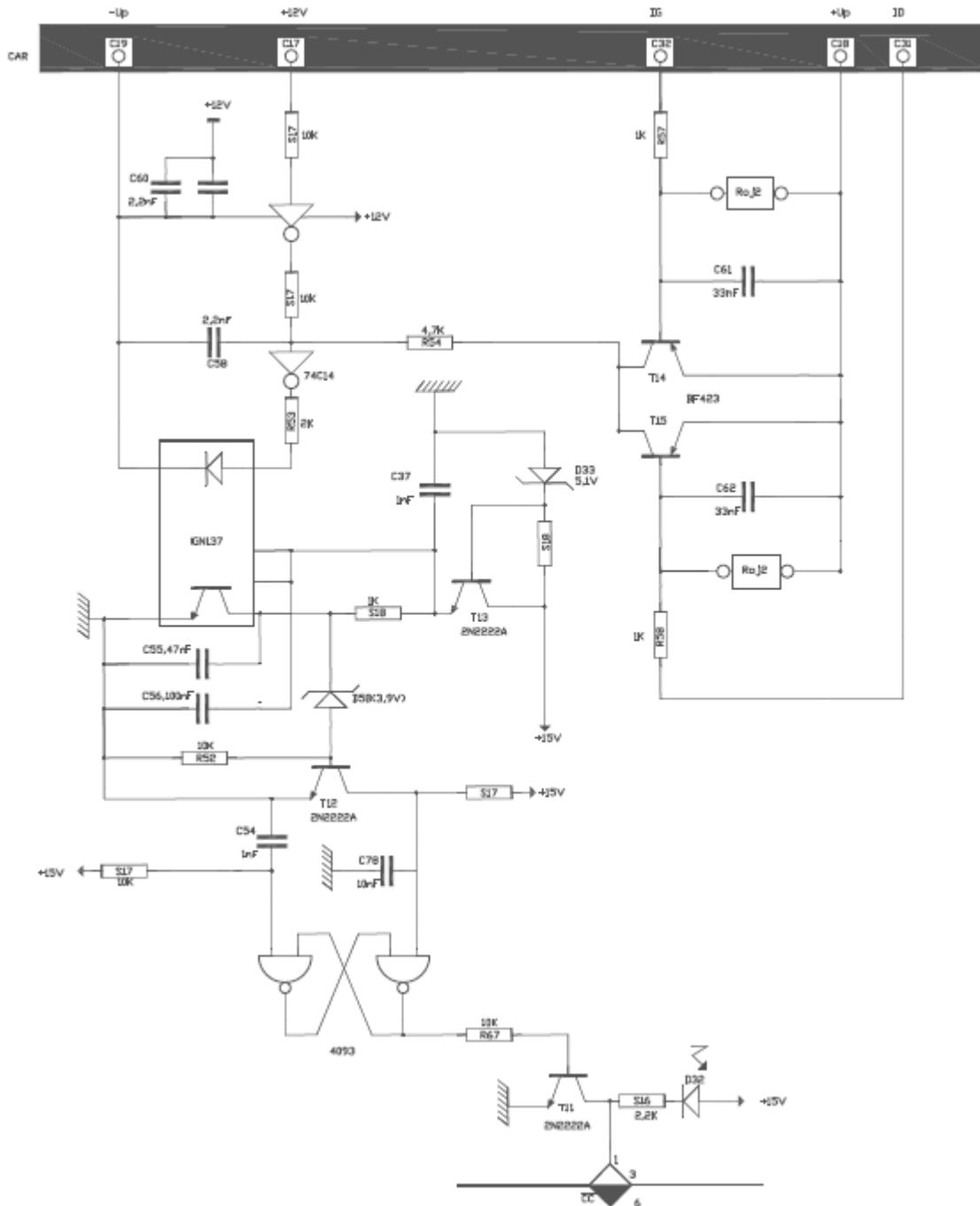
Las señales de entrada de este circuito están constituidas por una señal que indica el correcto funcionamiento del sistema de alimentación de los motores, una señal de control relacionada con el bucle de corriente del plano 2, y una señal de alimentación de los circuitos electrónicos. La señal de salida de este circuito sirve de mando de potencia y gobernará a través del circuito del plano 6 la activación de los transformadores de impulsos. Todas las señales deberán de ser correctas para habilitar el funcionamiento de los motores.

2.8.- PLANO N°8



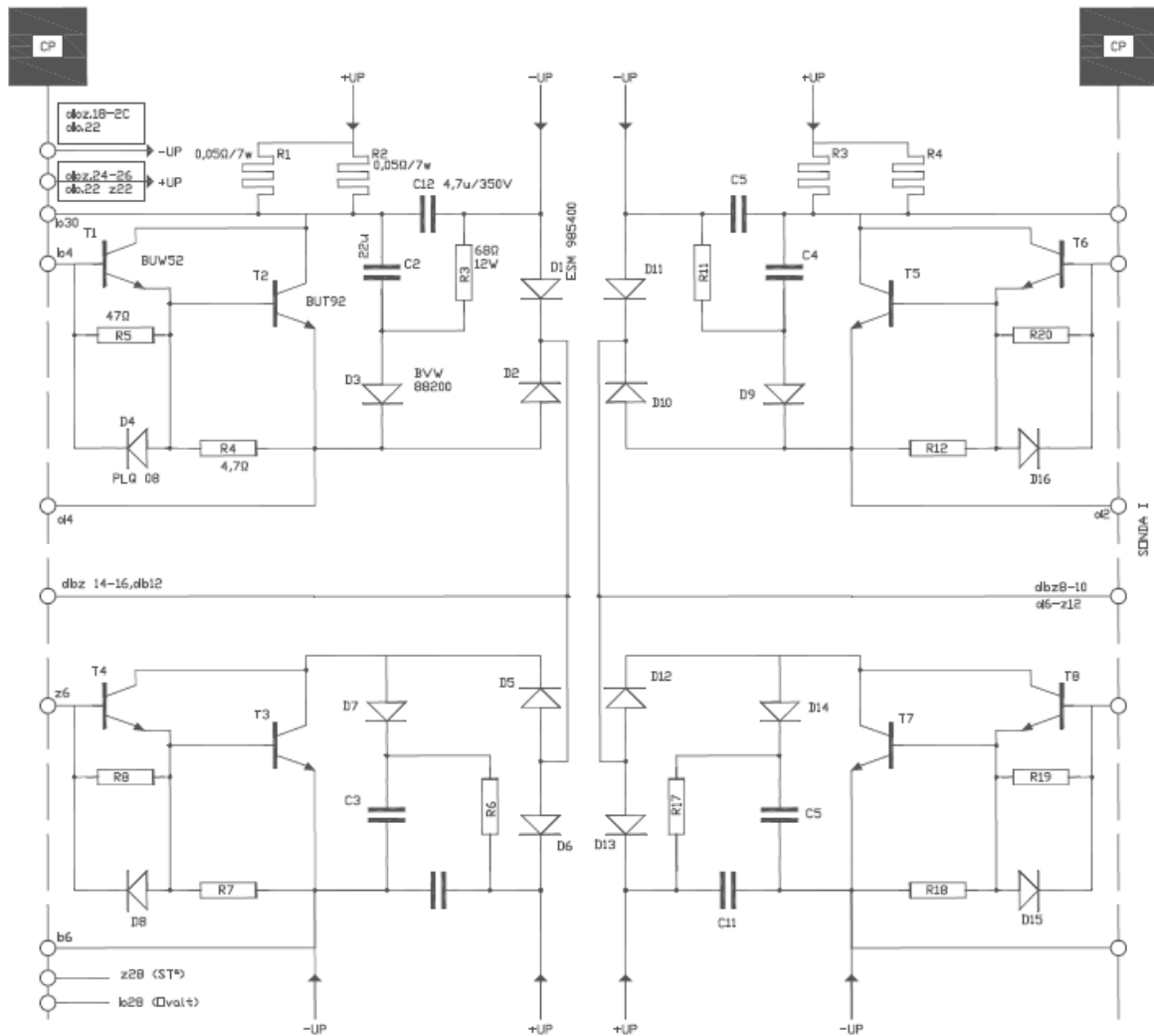
Este parte del circuito tiene una única señal de entrada proveniente del bucle de corriente del plano 2. Esta señal se trata para que la señal de corriente proporcione un nivel proporcional de tensión. La señal de salida se dirige a la sección de circuito del plano 7 y gobernará la activación o no de los transformadores de impulsos.

2.9.- PLANO N°9



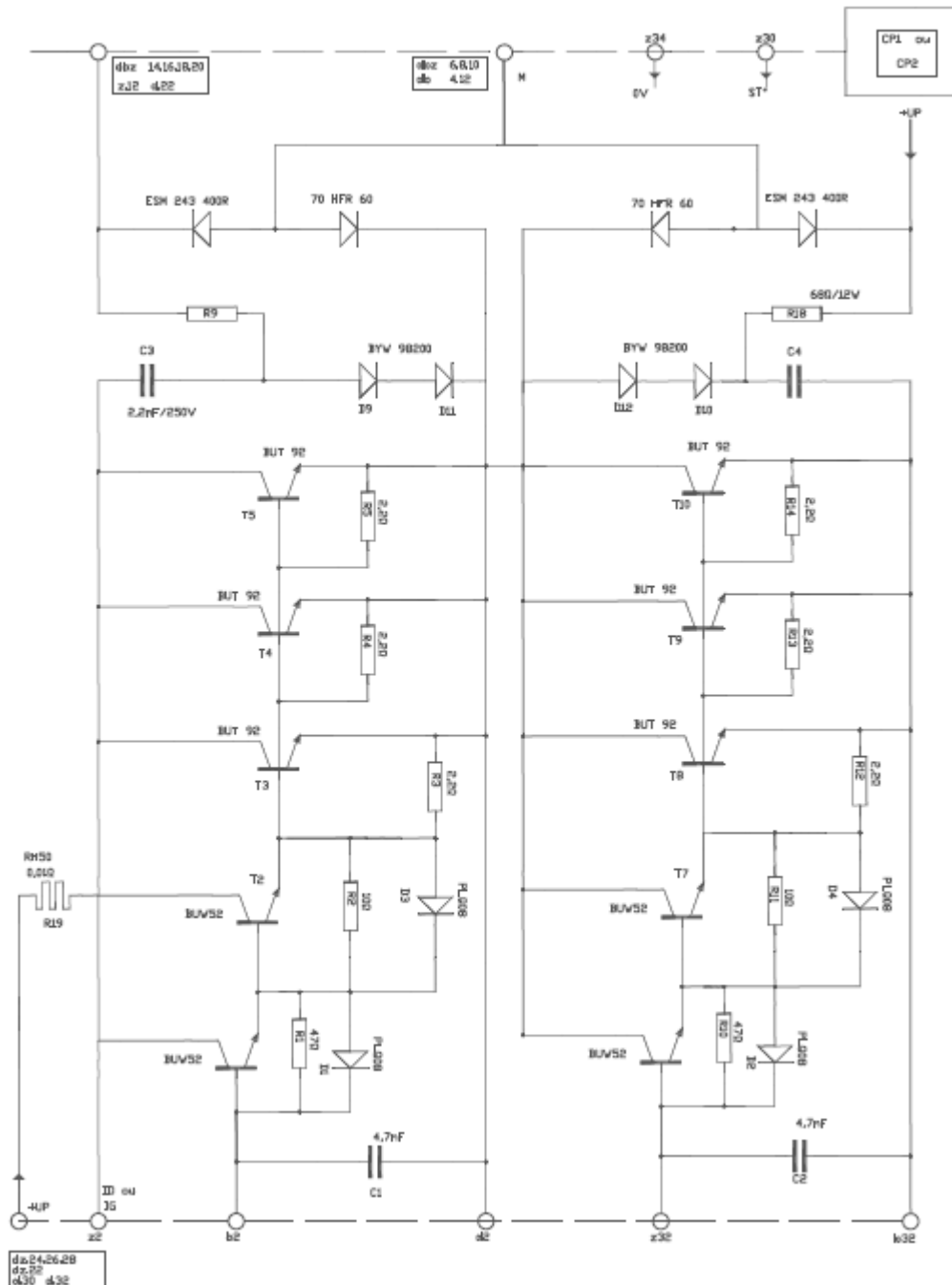
En esta parte del circuito se realiza una verificación de la alimentación de los motores. La señal de salida informará al resto del circuito de estado correcto o incorrecto de la misma, en cuyo caso se inhabilitará el funcionamiento del circuito tipo puente en H.

2.10.- PLANO N°10



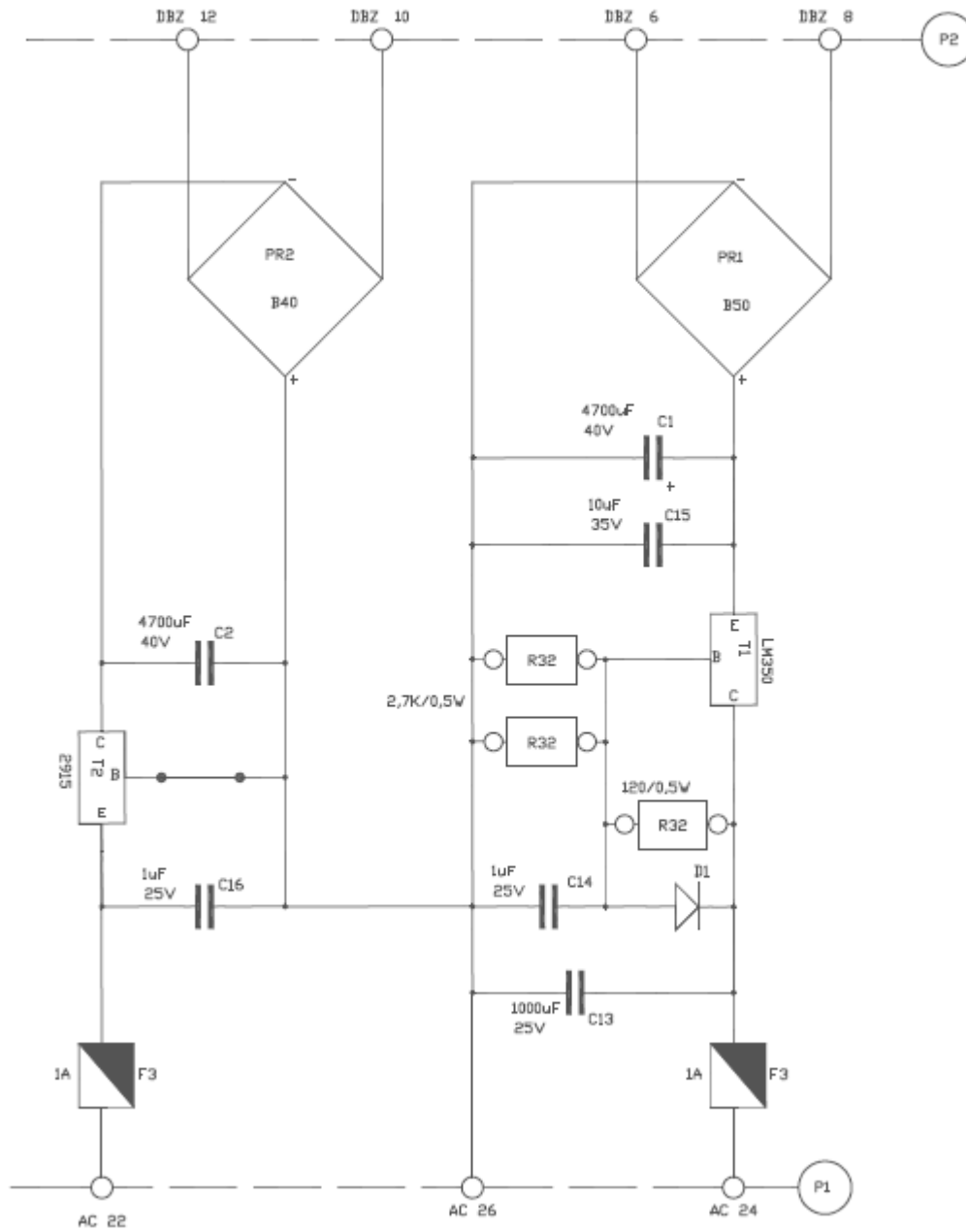
Esta parte del circuito corresponde a la placa del circuito del puente en H. Así, los transistores T2 y T7 serán los encargados de alimentar al motor para que gire en un sentido y los transistores T5 y T3 lo harán girar en el sentido contrario. Los cuatro pares de transistores se encuentran montados en una configuración tipo Darlington que les permite proporcionar una mayor ganancia de corriente para la alimentación de los motores. Las resistencias de potencia ubicadas en la parte superior permiten obtener una tensión proporcional a la intensidad y pueden disipar hasta 7 W.

2.11.- PLANO N°11



Este circuito corresponde al sistema de alimentación de un motor de mayor potencia y se añade a los otros circuitos para poder controlarlo. Este motor pertenecía al eje I del robot antiguo.

2.12.- PLANO N°12



Este circuito crea tensiones de +15 V y -15 V a partir de unos transformadores. Es en definitiva una fuente de alimentación.

3.-IMPLEMENTACIÓN FÍSICA DE LAS PLACAS DE CONTROL

En este punto se hace énfasis en la implementación física de dos de las partes más importantes del sistema de control descrito anteriormente: la placa del puente en H y el circuito que contiene los transformadores de impulsos.

3.1.- CIRCUITO DE LOS TRANSFORMADORES DE IMPULSOS

Esta placa constituye la implementación física de los planos 5 y 6. En este circuito se realiza un chequeo lógico de señales que indican el correcto funcionamiento de los sistemas del robot. Una vez comprobado lo anterior, una serie de transformadores de impulsos permiten alimentar al circuito puente en H. Los transformadores de impulsos son seis en total y son del tipo toroidal, con un primario y dos secundarios. La placa descrita se observa en la imagen inferior.

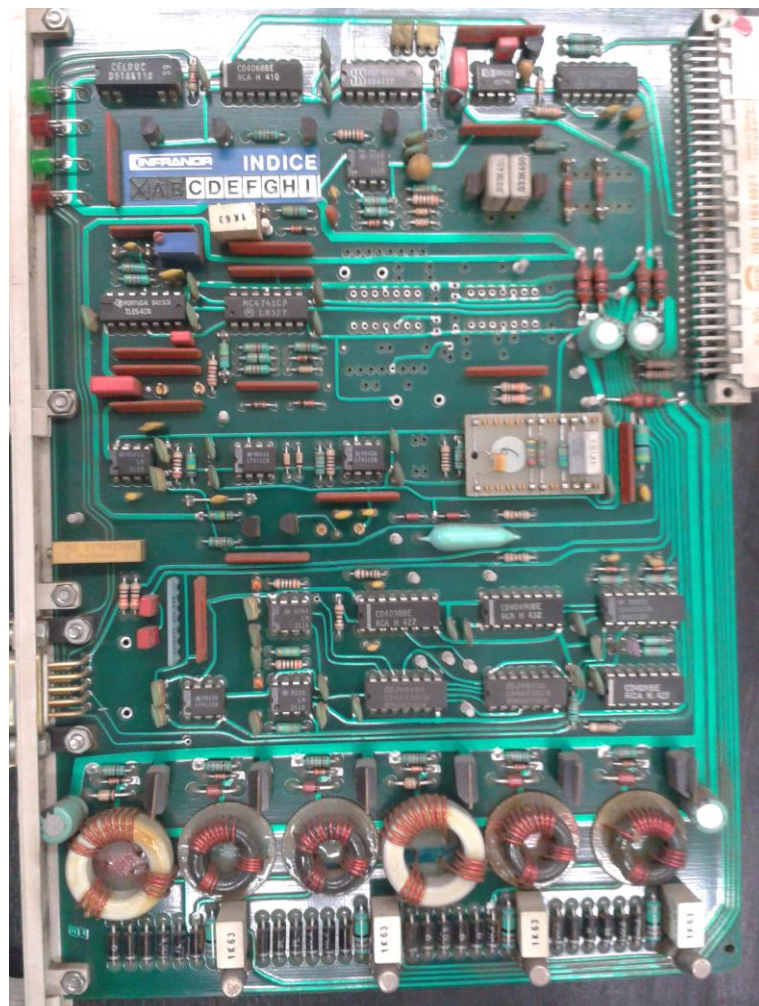


Ilustración 93 - Placa transformadores de impulsos

Los transformadores de impulsos constituyen la solución para el disparo de los transistores del puente en H, puesto que permiten disparar dejando a un lado el problema de poseer una referencia adecuada. Proporcionan un aislamiento galvánico al transferir energía entre sus bobinados por inducción electromagnética y sin que exista una conexión eléctrica entre ellos.

3.2.- PLACA DEL PUENTE EN H

El puente en H es un circuito electrónico que permite que un motor eléctrico de corriente continua gire en ambos sentidos. Son ampliamente usados en robótica y como convertidores de potencia, como es el caso. El término "puente en H" proviene de la típica representación gráfica del circuito. Un puente H se construye con 4 interruptores, implementados físicamente mediante transistores. Cuando los interruptores S1 y S4 (ver imagen inferior) están cerrados y S2 y S3 abiertos se aplica una tensión positiva al motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 y cerrando S2 y S3 el voltaje se invierte, permitiendo el giro en el sentido inverso. Con la nomenclatura utilizada los interruptores S1 y S2 nunca pueden estar cerrados al mismo tiempo, ya que se cortocircuitaría la fuente de tensión. Lo mismo sucedería con S3 y S4.

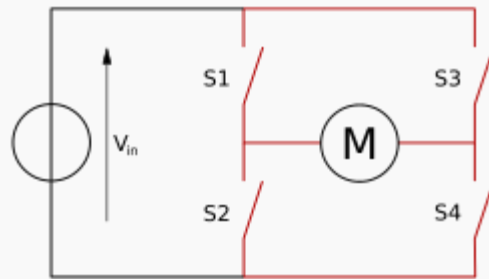


Ilustración 94 - Esquema puente en H

Los dos estados básicos del circuito se observan en la imagen siguiente.

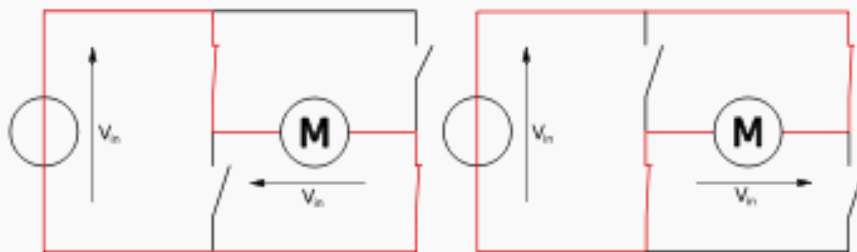


Ilustración 95 - Estados básicos del circuito puente en H

Como se dijo anteriormente, cada uno de los motores del robot posee una placa de puente en H encargada de alimentarlos, sirviendo de puente de unión entre los circuitos de control y de potencia. Para implementar el circuito se utilizan diodos y transistores de potencia. A su vez, es de vital importancia tener en cuenta la necesidad de refrigerar el circuito y el correspondiente cálculo de las superficies de intercambio de calor con el exterior. Los diodos ubicados como se observa en la imagen inferior permiten que la corriente circule en sentido inverso al previsto cada vez que se conmute la tensión, puesto que el motor está compuesto por bobinados que durante breves períodos de tiempo se opondrán a que la corriente varíe.

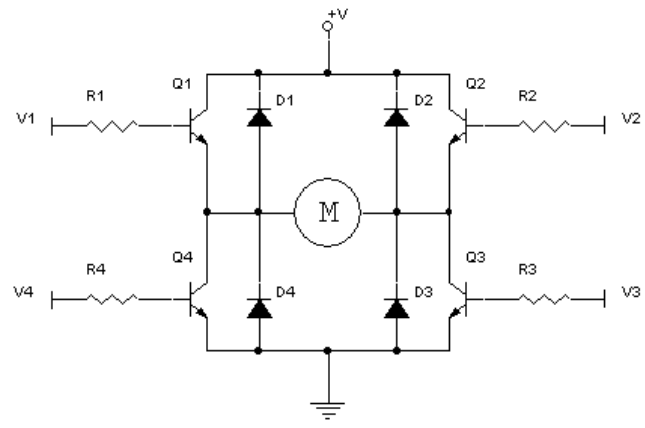


Ilustración 96 - Ubicación de los diodos en un puente en H

La placa del puente en H se observa en la imagen inferior.

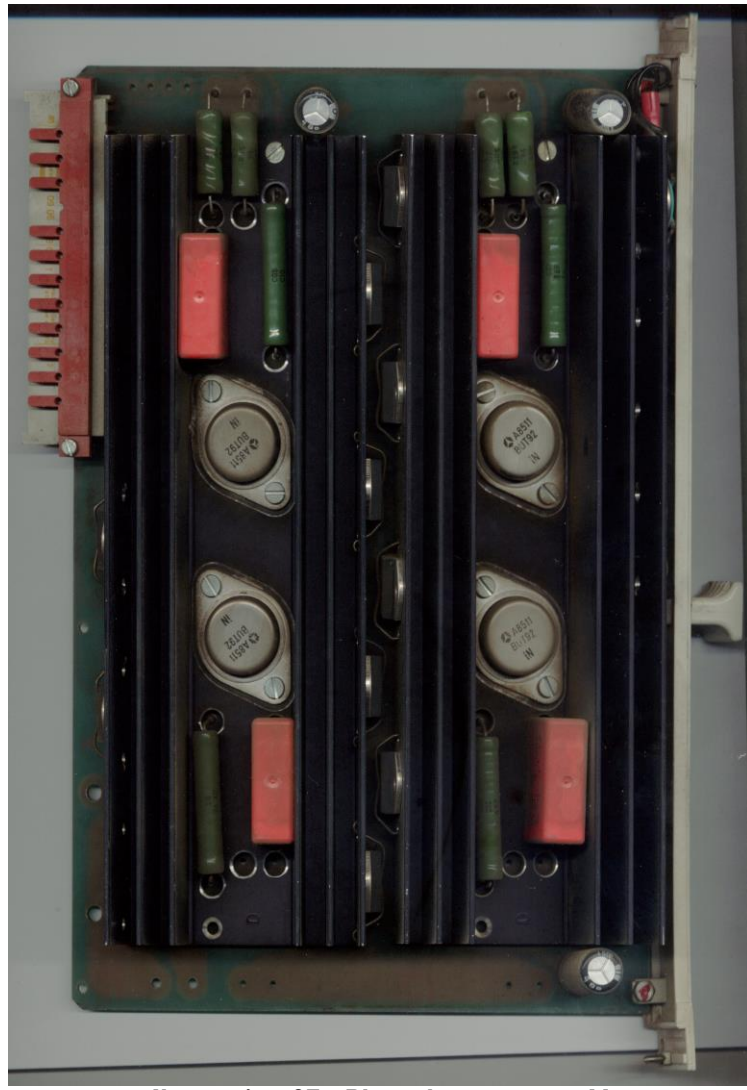


Ilustración 97 - Placa de puente en H

En la siguiente imagen se observa la parte inferior de la placa, con etiquetas correspondientes a cada uno de los elementos del circuito que pueden compararse con aquellos del circuito del plano 10.

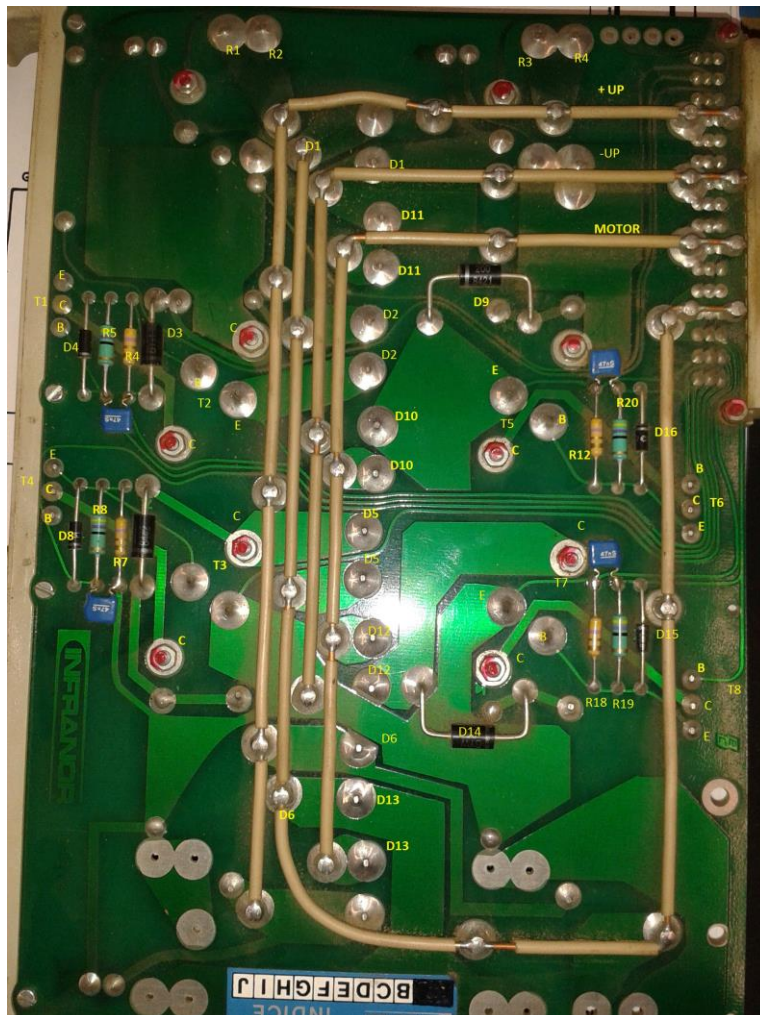


Ilustración 98 - Reverso placa del puente en H

Capítulo IV

Placa de control diseñada. Placas auxiliares

Sección I

PROCESO DE FABRICACIÓN DE LA PLACA DISEÑADA

1.- INTRODUCCIÓN.

El objetivo de la placa de control principal es proporcionar una base fija y confiable para las conexiones eléctricas entre los diversos circuitos integrados que la componen. La placa base es de cobre y carece de capa fotosensible, el proceso de fabricación consiste en devastar suficientes micras dentro de la placa de forma que se quite por completo la capa de material conductor. De esta forma, las diversas pistas quedan aisladas entre sí y forman las conexiones eléctricas que se necesitan.

El proceso de devastado se lleva a cabo con una fresadora tipo máquina CNC (máquina de control numérico). La fresadora se conecta a través de un cable puerto serie – USB a un ordenador desde el que se le cargan todas las trayectorias a realizar. Los patrones de pistas utilizados fueron realizados en AutoCAD y tratados con los programas Win PC y VCarve Pro, que permiten transformar el dibujo de AutoCAD en un lenguaje entendible por la máquina. Para fresar correctamente las pistas se deben de tener en cuenta diversos factores, entre ellos la horizontalidad de la placa y la facilidad de las brocas de desajustarse debido a las presiones a las que se ven sometidas. Además, es común que se produzcan fallos durante el fresado, algunas pistas pueden quedar sin fresar o, en un porcentaje muy pequeño de casos, la máquina puede perder sus coordenadas y cometer errores de posicionamiento durante la trayectoria de fresado. Debido a las numerosas fuentes de error todas las pistas fresadas se someten a un cuidadoso chequeo de conductividad. La placa también contiene numerosos huecos de taladro, en los que se ubican con posterioridad los zócalos que contendrán los circuitos integrados de la placa. La fresadora, cambiando a la broca correspondiente, realiza las tareas de taladro una vez introducidos los datos de trayectoria para realizar los diversos puntos.

El proceso de fabricación de la placa fue largo y complejo, especialmente porque la placa debió de fresarse por ambos lados (un lado de la placa corresponde al lado de componentes y el otro al lado de pistas), proceso que requiere de una gran precisión y de uniones entre

un lado y otro de la placa. Así, el paso a seguir una vez fresados ambos lado de la placa y realizado un proceso exhaustivo de chequeo de conductividades con un polímetro es el de la soldadura. En primer lugar, se sueldan las conexiones eléctricas que constituirán los puentes de pistas entre un lado y otro de la placa, luego se sueldan los zócalos de los circuitos impresos pin por pin.

Una vez soldados todos los zócalos se ubican los componentes y se conecta la alimentación. Es importante estar atentos en este punto a circuitos que se sobrecalienten, pues sería una indicación de que en algún punto de la placa se está produciendo un corto circuito. Si no se producen cortos detectables por el método descrito anteriormente el paso a seguir es comprobar el correcto funcionamiento de los diversos circuitos integrados. Esto se realiza programando pequeñas rutinas de comprobación de puertos para el microcontrolador y las básculas (74LS373), y rutinas un poco más complejas para programar los otros integrados que hacen parte de la placa y que se explicarán en la sección III de este capítulo. Debido a diversas necesidades surgidas después del proceso de fabricación de la placa original, que se explicará en detalle más adelante, fue necesario incluir placas de circuito impreso auxiliares (placa 8855-8853, placa DAC0800-74LS373 y placa de tratamiento de señales). Todas estas placas se detallan en las siguientes secciones de este capítulo. En la imagen inferior se observa la placa terminada.

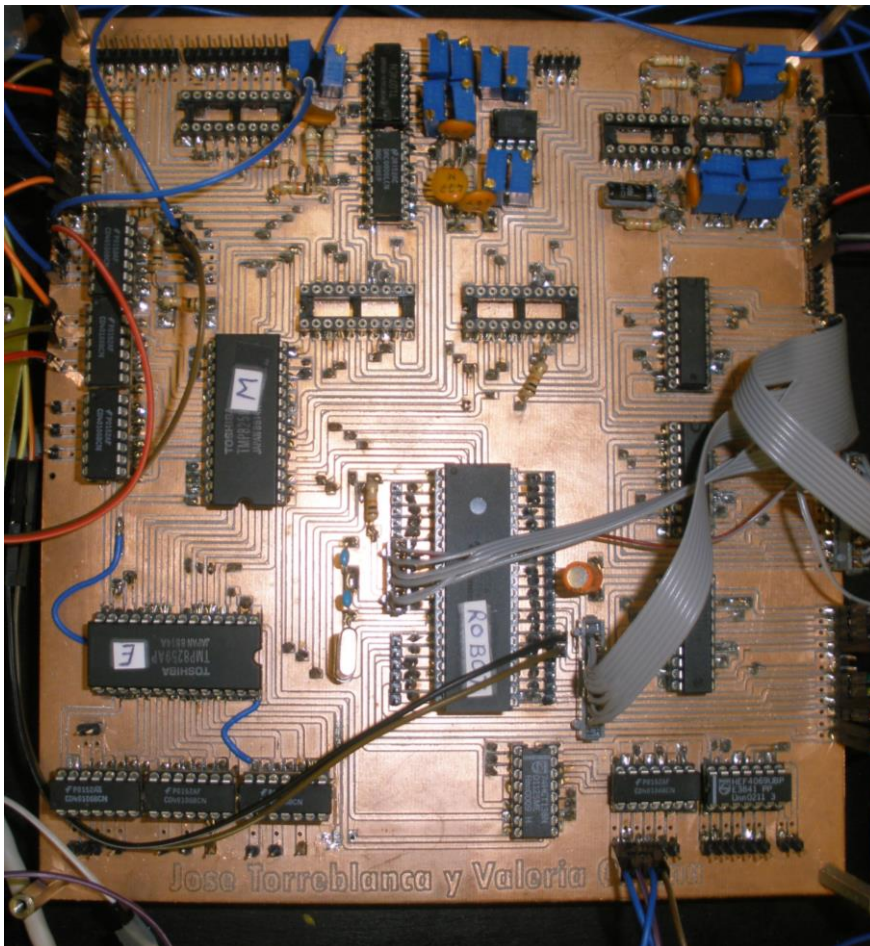


Ilustración 99 - Placa principal terminada.

2.- PROCESO DE FABRICACIÓN.

2.1- DISEÑO DE LA PLACA EN AUTOCAD

El proceso de fabricación de la placa inicia con el diseño de las pistas en AutoCAD. Se deben tener en cuenta multitud de factores para realizar un correcto diseño. Las pistas se trazan con un grosor de 1mm y se debe tener en cuenta la separación real de los pines de los circuitos integrados para situar correctamente los pines de inserción, que poseen una anchura de 2,5 mm. La placa se realiza devastando el cobre por ambas caras, por lo que son necesarios dos diseños, uno por cara. Los diseños reciben los nombres de lado de componentes y lado de pistas y, como lo indican sus nombres, en el lado de componentes se situarán los circuitos impresos y el lado de pistas alberga el grueso de las conexiones eléctricas. Ambos diseños deben realizarse como placas espejo, puesto que a la hora del fresado uno ira literalmente encima del otro. En cuanto a la alimentación de la placa, se diseña de tal modo que todos los puntos de 5V, GND, +12V y -12V (para los circuitos amplificadores) están unidos entre sí y con los bornes exteriores de la placa, donde se conecta la fuente de alimentación externa. El diseño más importante, puesto que es el que se importará más adelante a los programas de fresado es el que se realiza utilizando únicamente polilíneas de AutoCAD. En la siguiente imagen se observa este diseño de la cara de pistas.

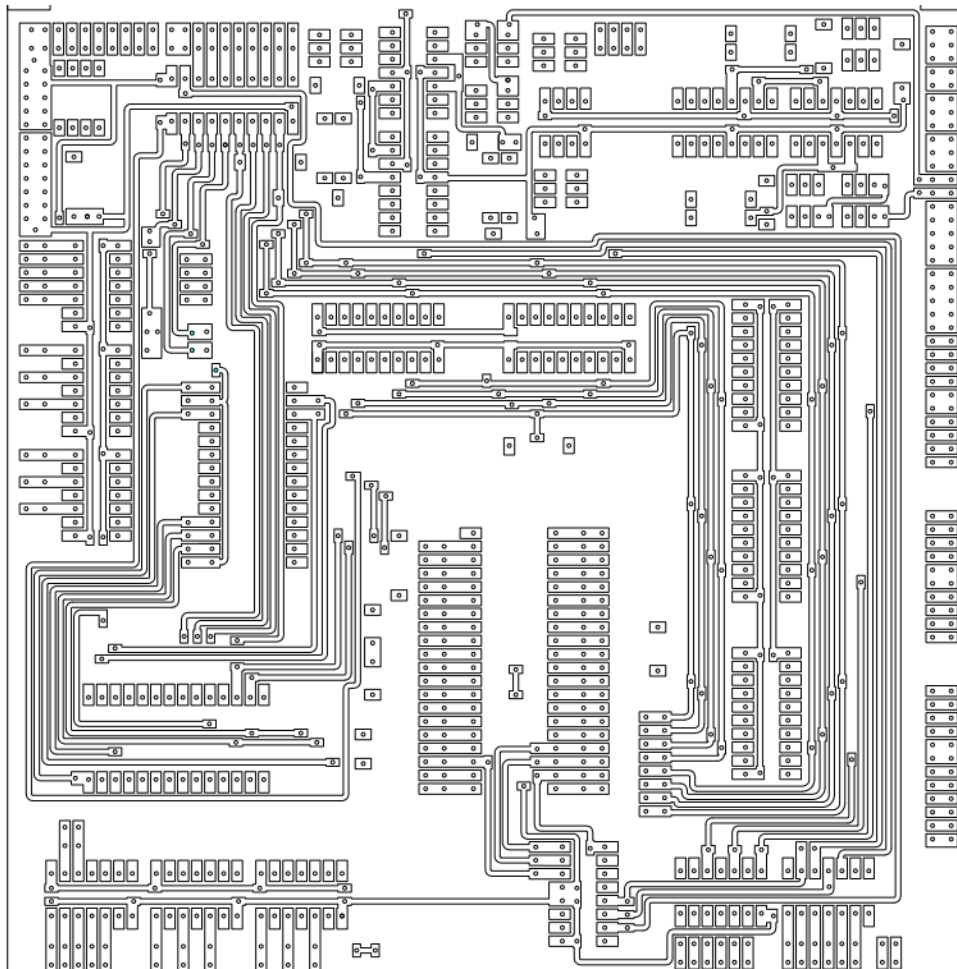


Ilustración 100 - Diseño en AutoCAD: cara de componentes.

El diseño se compone de una serie de capas que van conformando la disposición final de la placa. La cara de pistas de la imagen anterior se traza también a modo de esquema. Así, con la ayuda de colores se logra tener un mejor entendimiento del diseño y de la ubicación de los circuitos integrados. Se han agregado etiquetas con el nombre de los circuitos integrados.

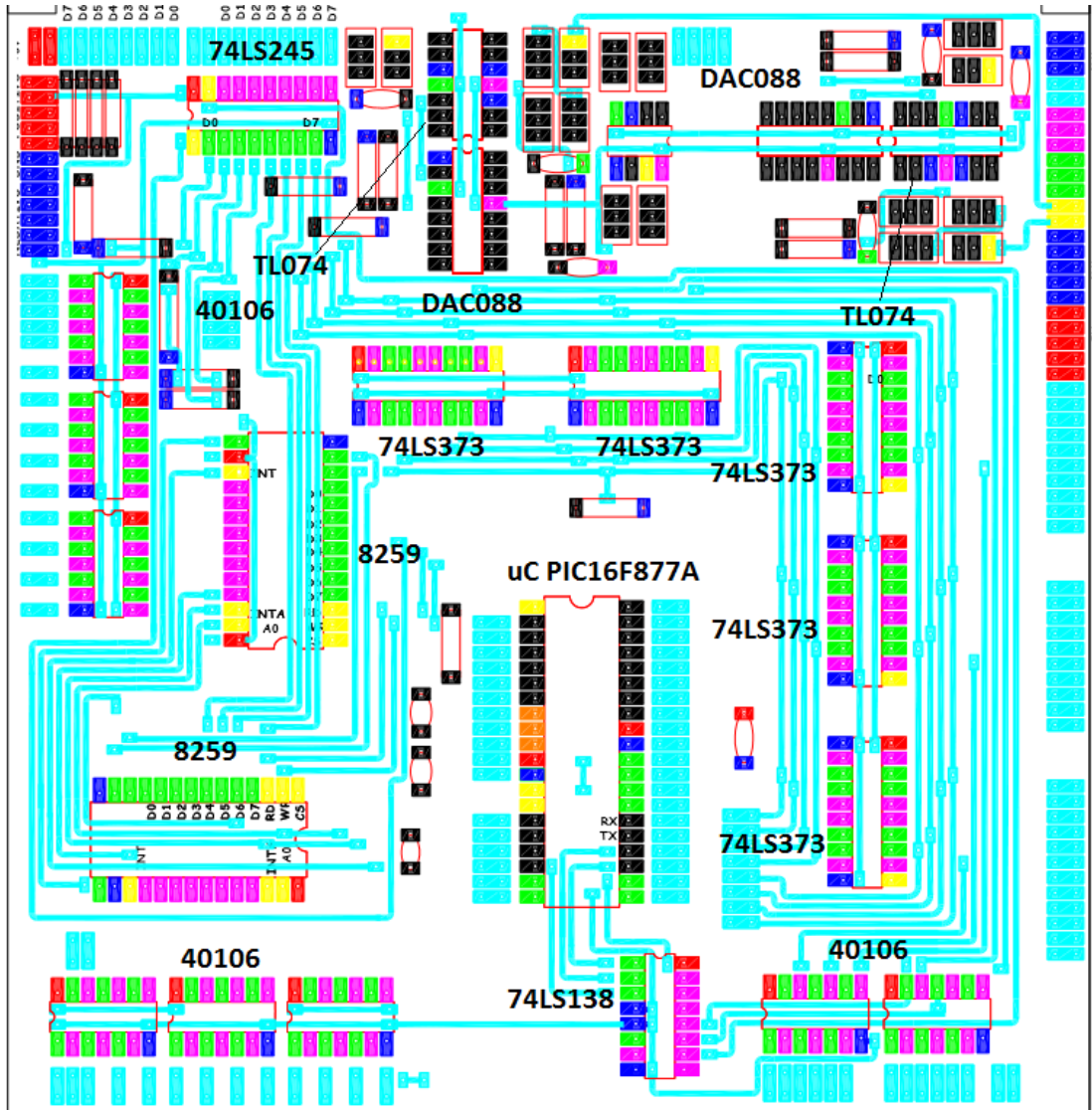


Ilustración 101 - Esquema a color: cara de componentes

En el esquema el color rojo hace referencia a conexiones a +5V y el azul a conexiones a masa. Los pines exteriores en rosa son alimentaciones a -12V y los verdes a +12V. Todas las pistas y sus conexiones a pines exteriores se rellenan en azul claro. En los circuitos integrados se observan las patillas de alimentación en rojo y en azul. Los elementos aislados corresponden a resistencias y condensadores que se conectan los circuitos integrados.

Como se dijo anteriormente, el lado opuesto de la placa corresponde al lado de pistas. En las imágenes inferiores puede observarse su diseño tanto en polilíneas como en formato esquemático.

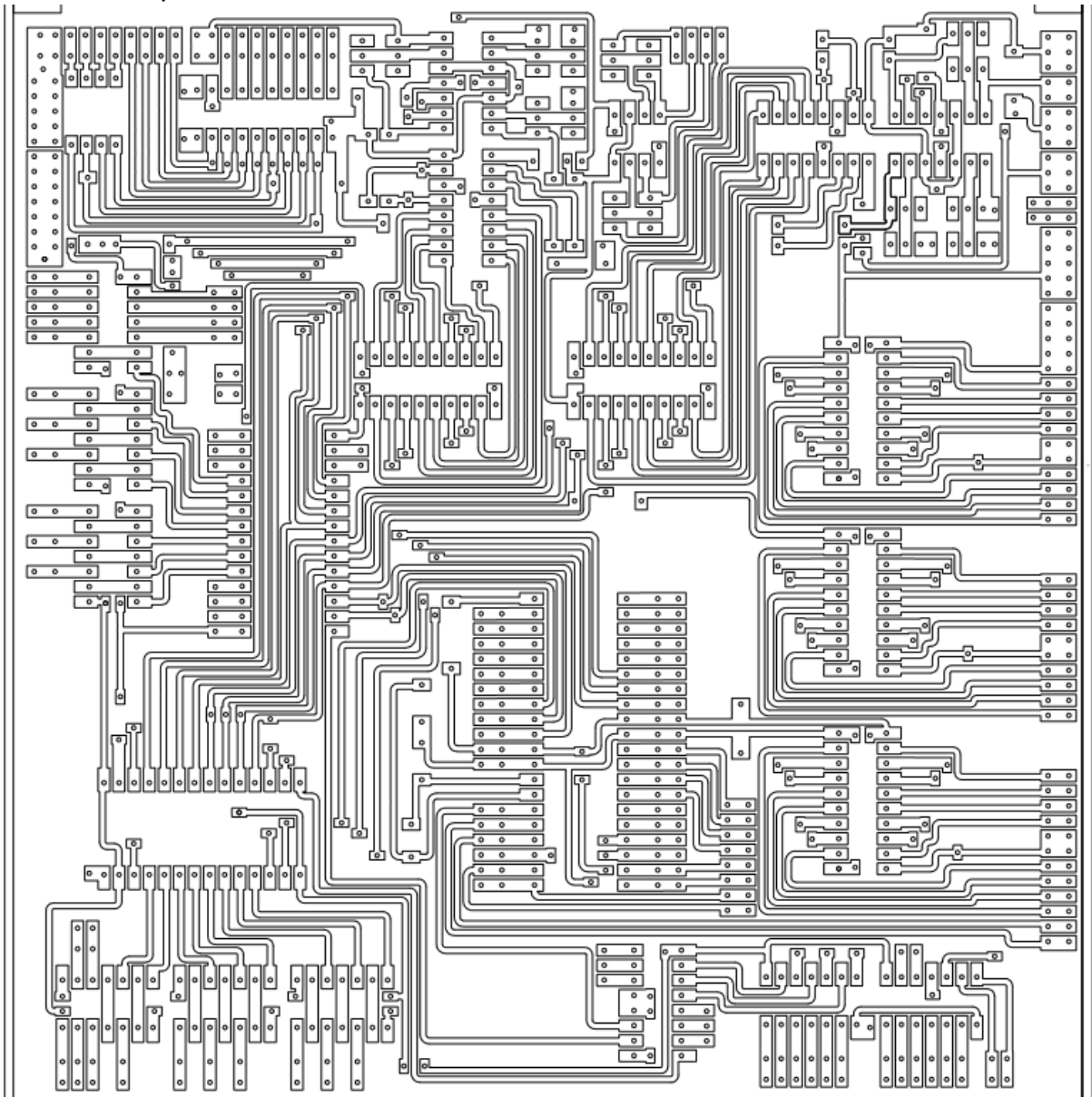


Ilustración 102 - Diseño de la placa: cara de pistas

La cara de pistas contiene el grueso del conexionado eléctrico. Después del fresado se realizarán los puentes de conexión entre el lado de componentes y el lado de pistas. De esta forma, existirá continuidad entre las pistas de un lado de la placa y las del otro. El proceso de soldadura de los zócalos de los componentes se realizará por este lado de la placa, insertando los zócalos por el lado de componentes para soldar luego los pines.

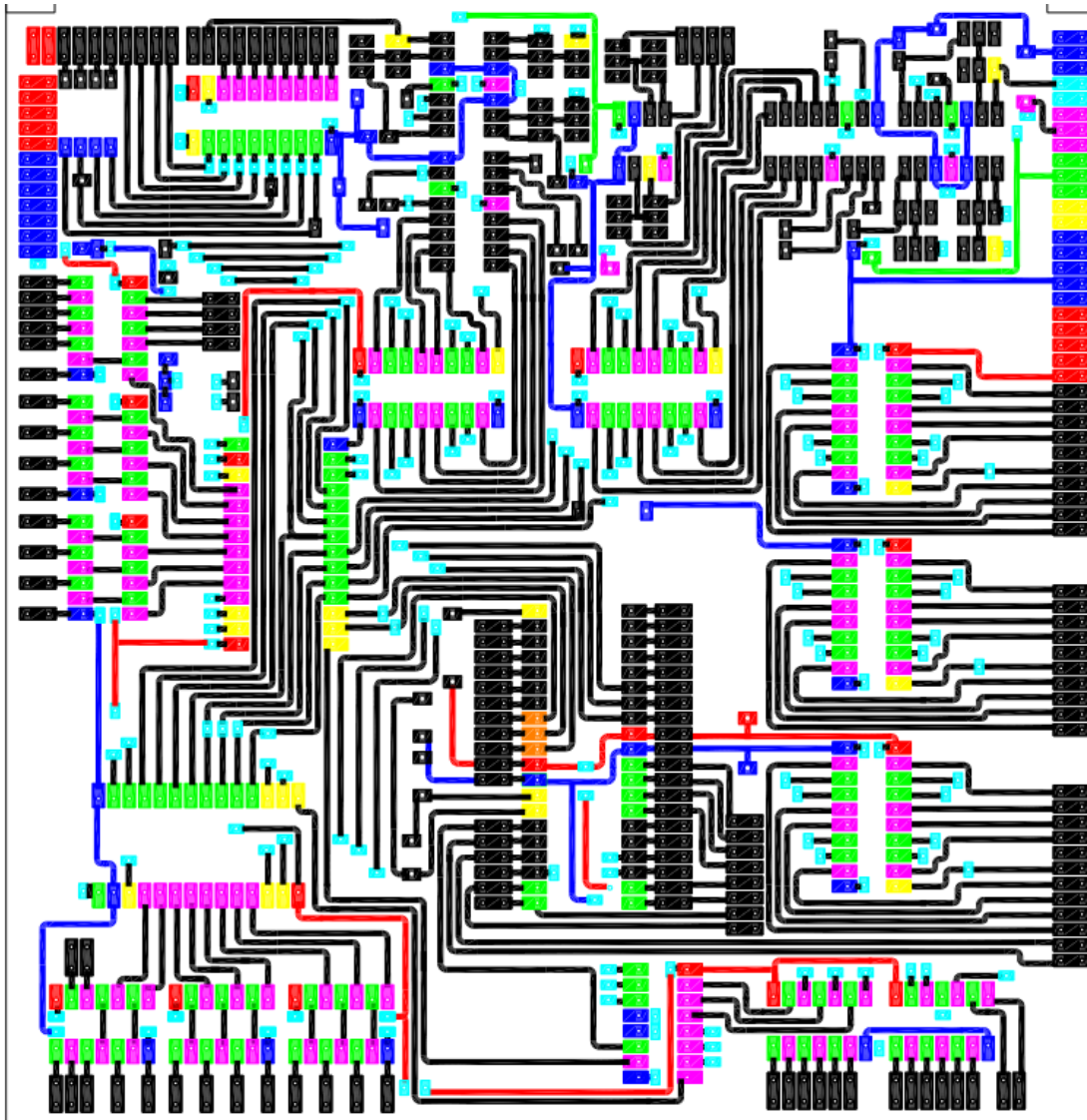


Ilustración 103 - Esquema a color: cara de pistas

Una vez finalizado el diseño de ambos lados de la paca el paso a seguir es fresar.

2.2.- FRESADO.

El fresado es el proceso por el cual los diseños creados a través de programas de diseño asistido por ordenador se transforman en órdenes que, transmitidas a una máquina de fresado tipo CNC, permiten devastar la placa de cobre virgen por las pistas demarcadas, a la profundidad requerida y con las características adecuadas.

2.2.1.- PROGRAMAS DE ASISTENCIA PARA EL FRESADO. WIN PC Y VCARVE PRO.

El software VCarvePro proporciona una solución potente e intuitiva para el diseño 2D y el cálculo de trayectorias de herramientas 2D para el mecanizado de piezas con una máquina de CNC (control numérico). Este software CAD CAM permite importar diseños

2D de otros programas (como Autocad). Las opciones de trayectoria cubren todas las operaciones de enrutamiento 2D típicas como el perfilado, incrustaciones y perforaciones. Cada trayectoria de la herramienta incluye opciones para personalizar los ajustes y proporcionar un alto nivel de control para diferentes tipos de operaciones. Además, todas las trayectorias se pueden previsualizar para saber cuál será el resultado.

Por otro lado, Win PC es un software de control numérico que permite el control de máquinas fresadores desde ordenadores con puerto USB. Este software puede ejecutarse en cualquier ordenador tipo PC o portátil (como es el caso), con sistema operativo Windows de 32 o 64 bits y que cuente con un puerto USB libre.

2.2.2.- PROCESO DE FRESADO.

El proceso de fresado inicia realizando los huecos de taladro de la placa. Para ello, se ejecuta Win PC a la vez que se instala una broca de 3 mm. La broca se posiciona en el punto de origen automático de la máquina dando la orden correspondiente en Win PC. Las coordenadas que le corresponden son 20, 20. A continuación, se ejecuta el VCarve PRO y abrimos el archivo de AutoCAD correspondiente cuya última modificación sea la de activar únicamente la capa de componentes en polilíneas. En el menú de la izquierda del VCarve se introducen la medida de la placa (230 X 321 mm) y el grosor del material (1,65 mm) para posteriormente hacer click en el botón OK.

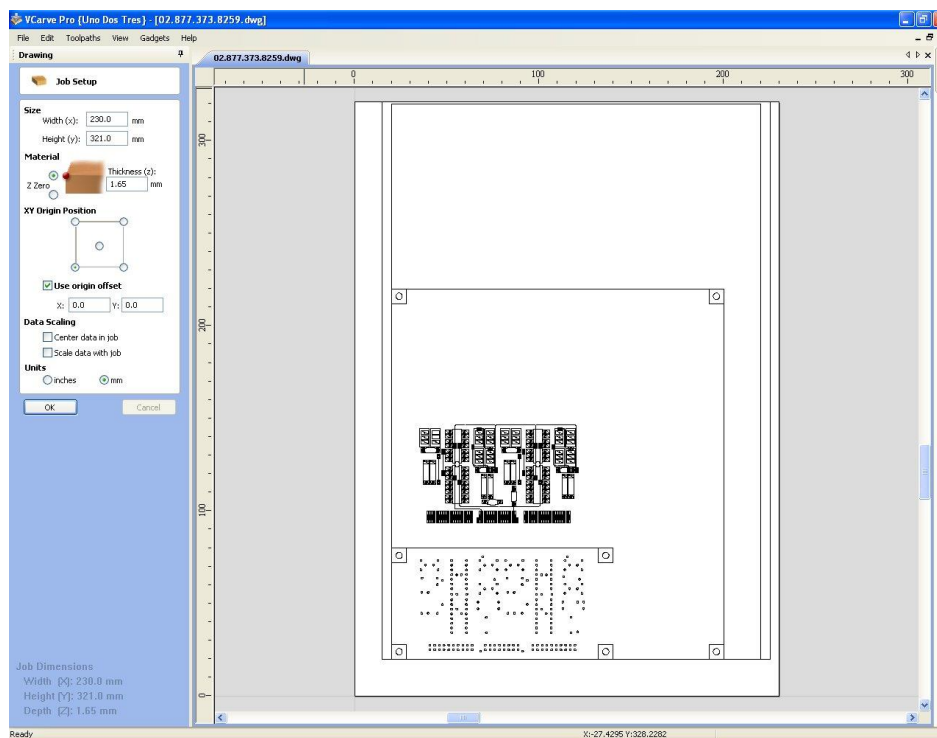


Ilustración 104 - Fresado: configuración inicial

Una vez marcados todos los agujeros (utilizando la tecla SHIFT) seleccionamos la herramienta dirigiéndonos al menú Herramientas.

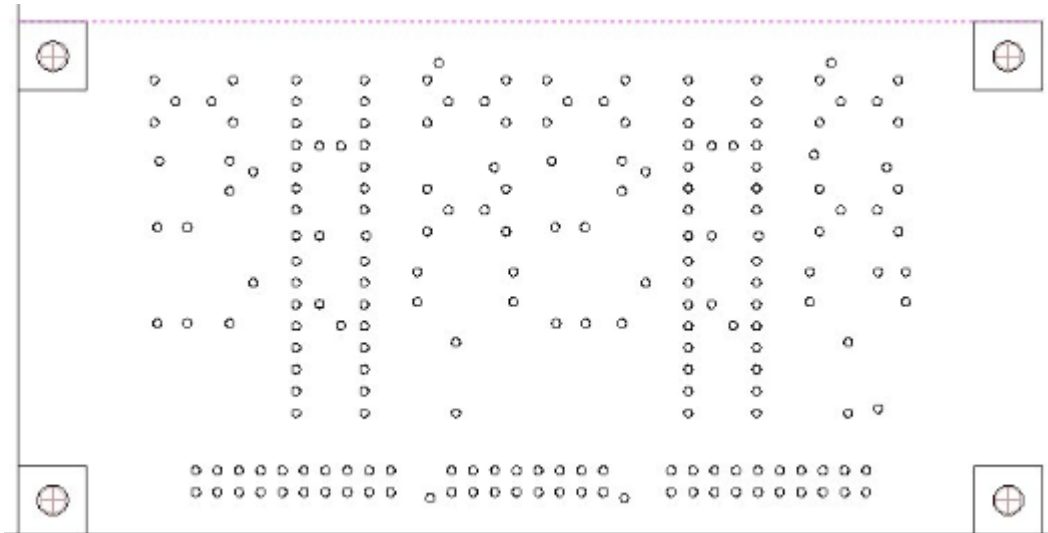


Ilustración 105 - Fresado: selección de taladros

Seleccionamos la herramienta “Drill” de 1 mm.

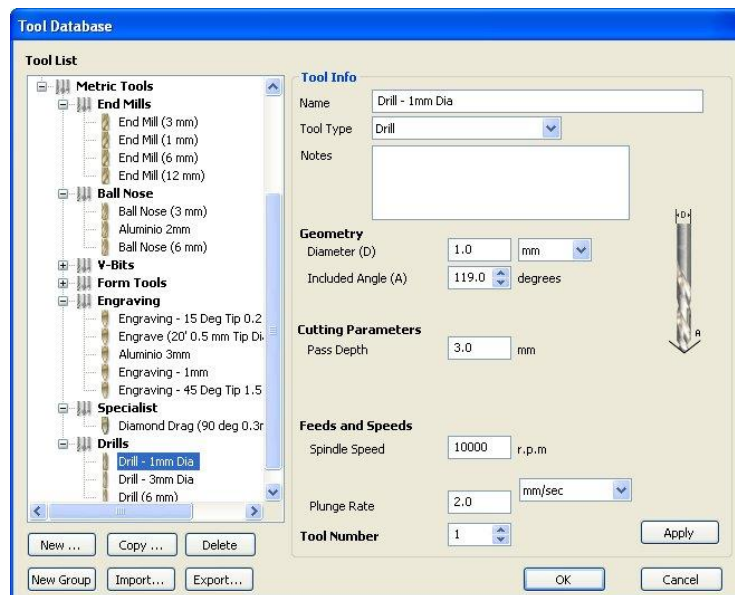


Ilustración 106 - Fresado: selección de la herramienta

Una vez seleccionada la herramienta marcamos la opción de “Peck drilling” en el menú de la derecha. Con la configuración indicada en la imagen inferior la máquina realizará en total dos pasadas para fresar por completo. Se configura una profundidad de taladrado de 2,5 mm, suficiente para atravesar la placa. Hacemos click en el botón de “Calculate”.

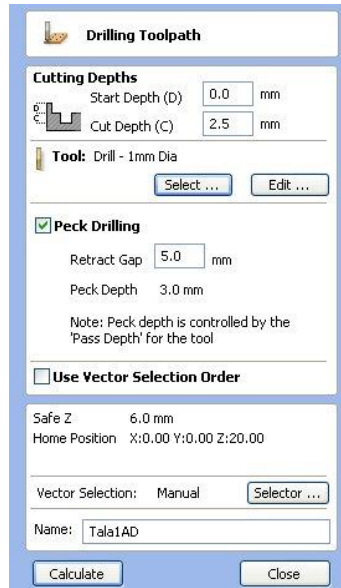


Ilustración 107 - Fresado: cálculo de la trayectoria

Se producirá una alerta puesto que la configuración introducida llevará a la máquina a atravesar el material.

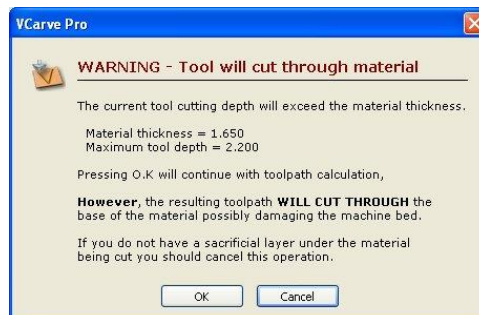


Ilustración 108 - Fresado: aviso de emergencia

Mediante la herramienta “Toolpaths” el programa VCarve PRO permite realizar una simulación del resultado final del fresado.

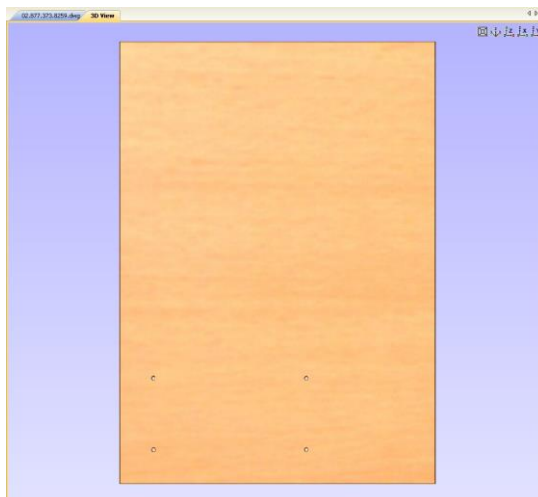


Ilustración 109 - Fresado: simulación del resultado final

Damos click en la opción “Save Toolpath” del menú de la derecha para guardar el programa que contiene la secuencia de ejecución configurada mediante los pasos anteriores. El programa se guarda con la extensión .nc.

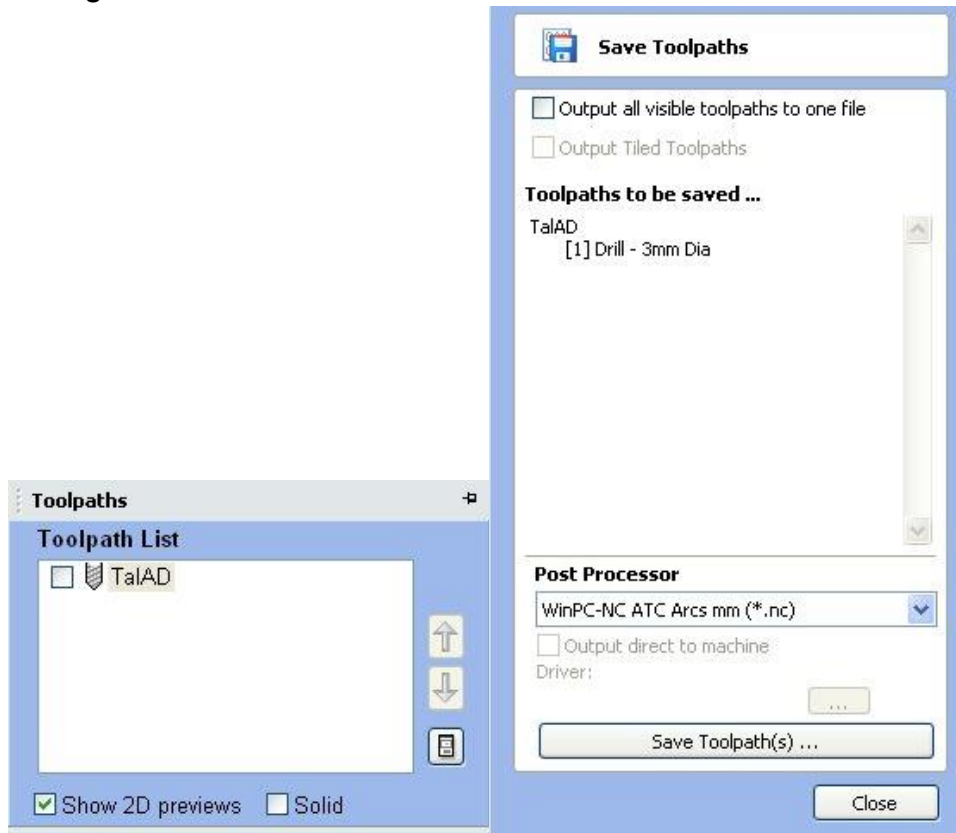


Ilustración 110 - Fresado: guardamos el archivo .nc del VCarve

Ahora ejecutamos Win PC y realizamos una medición de la altura de la placa. Vamos al menú Herramientas – Seleccionar herramienta y seleccionamos la herramienta número 1. Hacemos click en el botón OK.



Ilustración 111 - Fresado: elección de la herramienta

En el menú **Parámetros - Herramientas - Loop** obtenemos el resultado de la medición (52,03 mm) al que deberá sumársele la altura del sensor utilizado en la medición (82,3 mm).

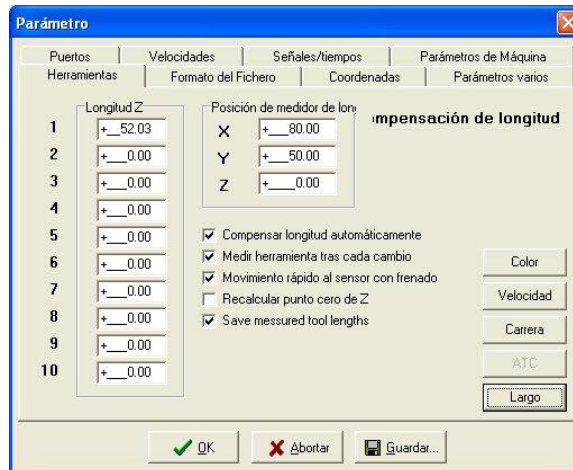


Ilustración 112 - Fresado: medición de la altura

Para iniciar la máquina de fresado se hace click en el botón de Inicio, aparecerá una ventana de avance del proceso como la siguiente:



Ilustración 113 - Fresado: ventana indicadora del proceso

Una vez realizados los taladros de la placa se procede a cortar el exceso de cobre de la placa virgen original. Para ello, se realiza en primer lugar una medida de la altura al ras de la madera. Una vez realizada la medición ejecutamos el VCarve y escogemos la línea que nos interesa cortar. La herramienta utilizada para el corte es el "End Mill" de 1mm.

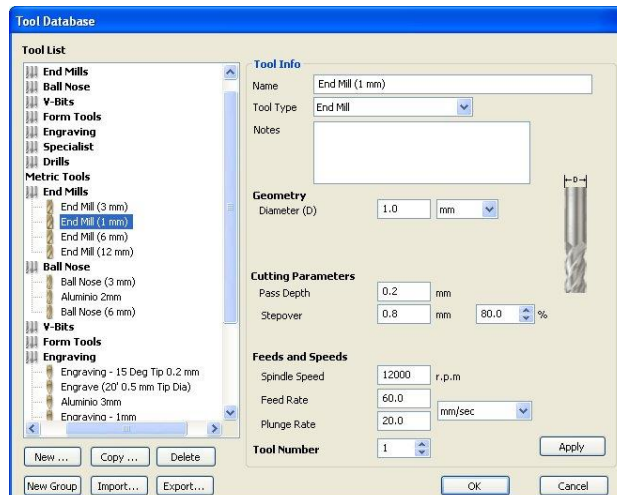


Ilustración 114 - Fresado: herramienta del corte

En el menú de la derecha se configura el corte de forma que la fresadora realice una trayectoria de rampa en el acercamiento a la superficie a cortar.

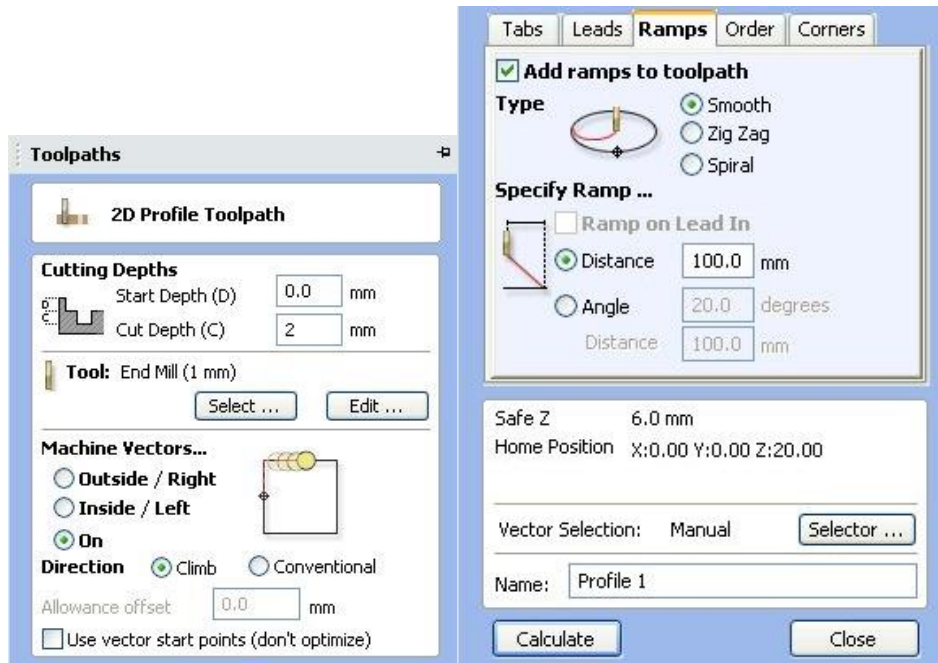


Ilustración 115 - Fresado: características del corte

Damos un nombre al nuevo archivo y damos click en Calcular. A continuación, damos click en Save Toolpaths.

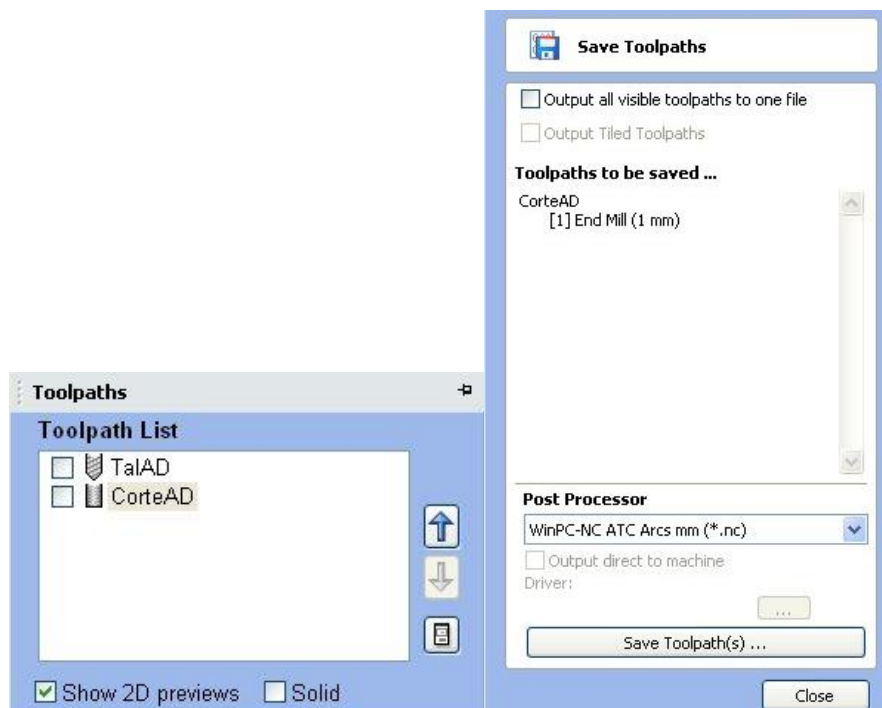


Ilustración 116 - Fresado: archivo de corte .nc

Ahora ejecutamos Win PC, cambiamos la broca y realizamos una nueva medición de altura. Hacemos click en Abrir y cuando nos pregunte por Guardar escogemos Sí. Damos a iniciar. Durante todo el proceso el programa Win PC actualiza una ventana con datos de las coordenadas de la máquina y las coordenadas de la pieza.

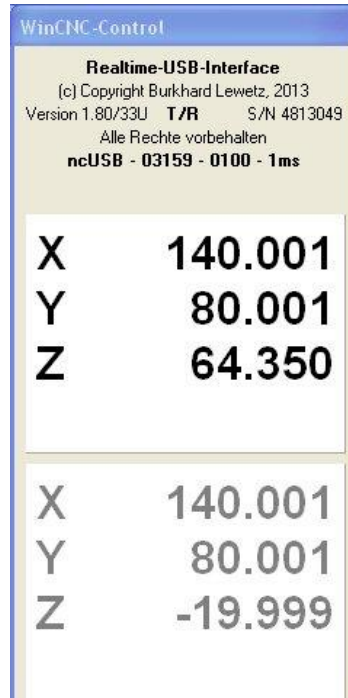


Ilustración I17 - Fresado: datos de coordenadas

A continuación se fresan las pistas del lado de componentes. Como el lado de pistas es un espejo del de componentes se debe girar la placa en AutoCAD sobre el eje x. Guardamos el archivo en AutoCAD con la capa que queremos fresar activa o escogemos las líneas a fresar directamente en el VCarve.

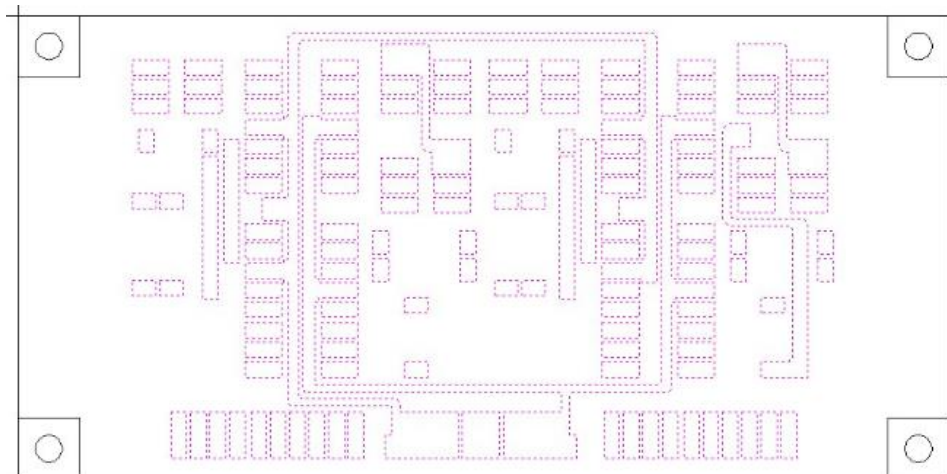


Ilustración I18 - Fresado: líneas a serigrafiar

Quitamos la broca de corte e instalamos la de serigrafía. Para el proceso de serigrafía se elige la herramienta “Engraving” de 0,2 mm.

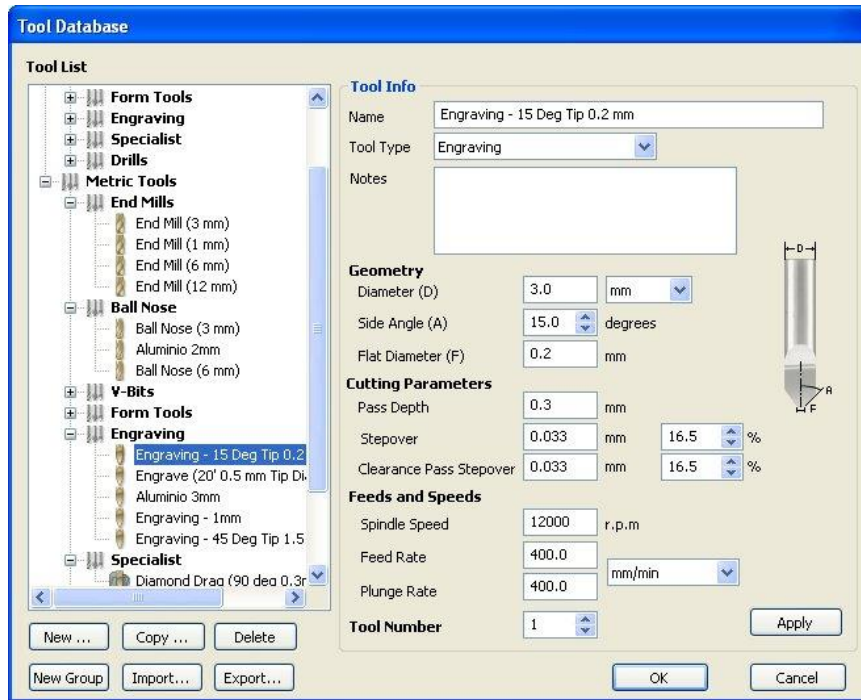


Ilustración 119 - Fresado: herramienta de serigrafía

En el menú de la derecha configuramos el corte a 0,2 mm.

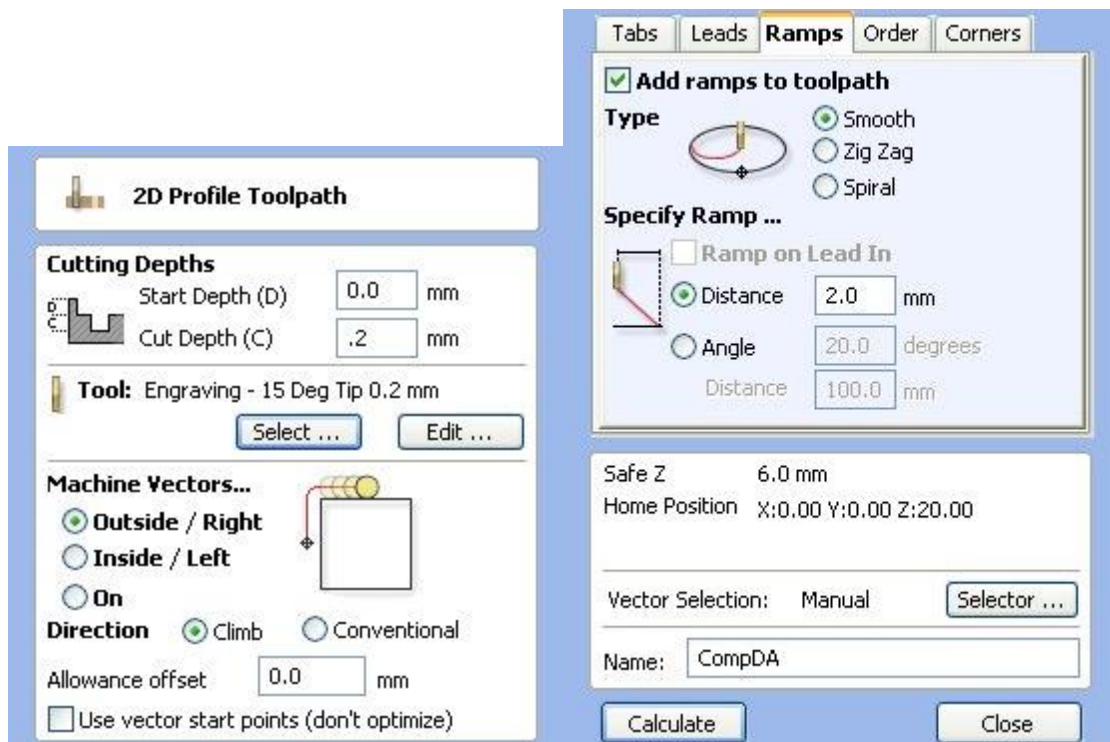


Ilustración 120 - Fresado: configuración del serigrafiado

A continuación hacemos click en Calcular y posteriormente en “SaveToolpath”. Ejecutamos Win PC y realizamos en primer lugar una medición de altura. Abrimos el archivo recientemente creado en VCarve y hacemos click en iniciar. En la imagen inferior observamos cómo la fresadora termina de serigrafiar la placa ya taladrada.

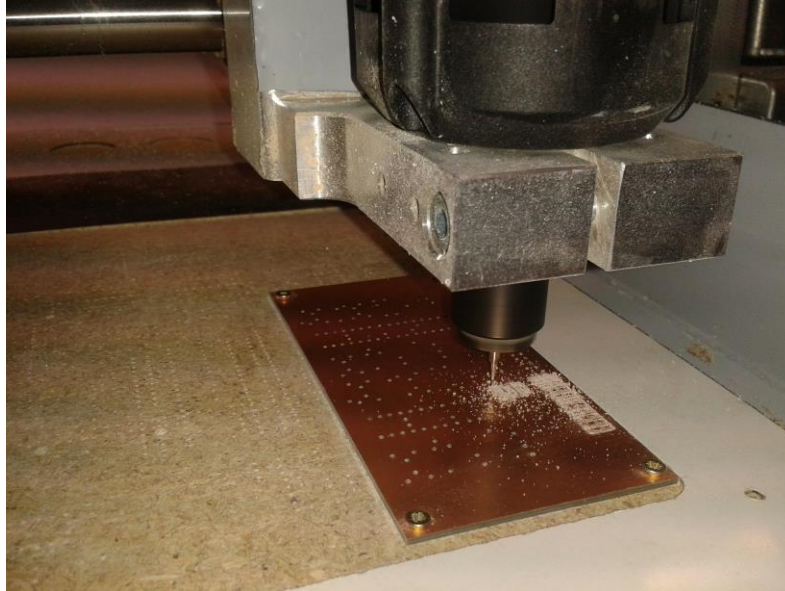


Ilustración 121 - Fresado: máquina CNC

Durante todo el proceso de fresado la placa se mantuvo sujeta a la plancha de la máquina CNC mediante cuatro tronillos ubicados en sus esquinas. Para realizar estos agujeros se siguió el mismo procedimiento que en el caso de los agujeros para inserción de pines, utilizando una herramienta “Drill” de 3 mm.

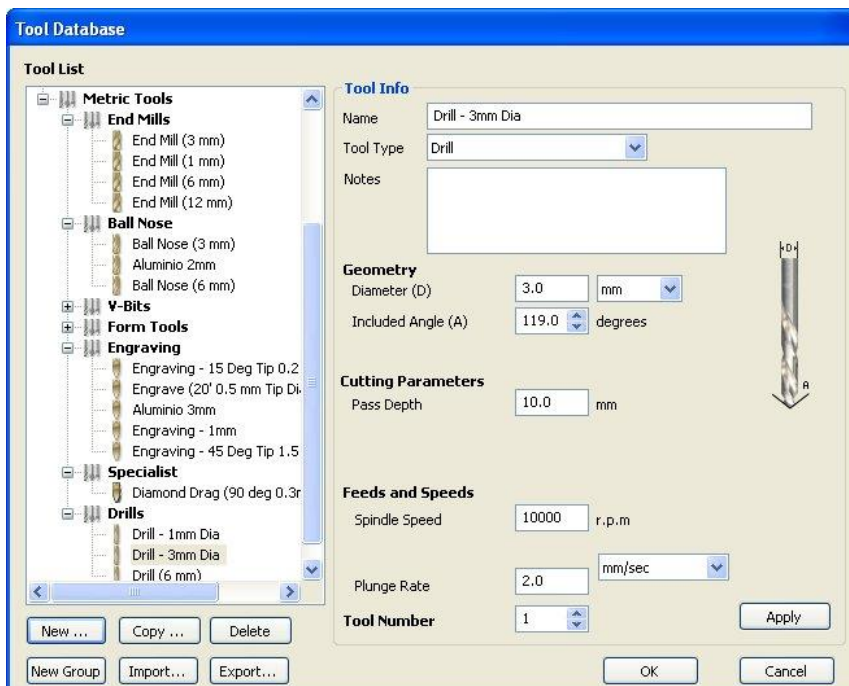


Ilustración 122 - Fresado: agujeros auxiliares

El taladrado se configuró en el menú de la derecha con una profundidad de 2,2 mm, suficiente para atravesar la placa de cobre por completo.

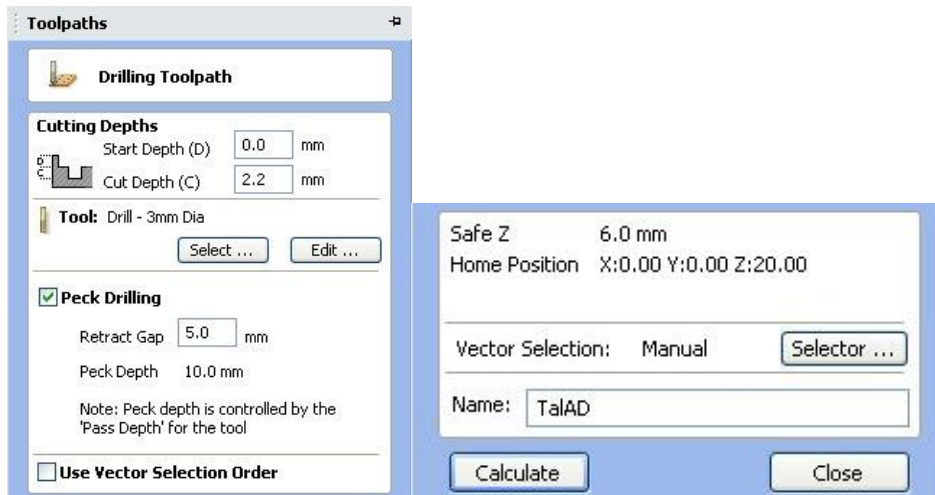


Ilustración 123 - Fresado: configuración taladros auxiliares

Una vez terminado el proceso de fresado de la placa se obtiene la base de conexiones eléctricas del sistema de control.

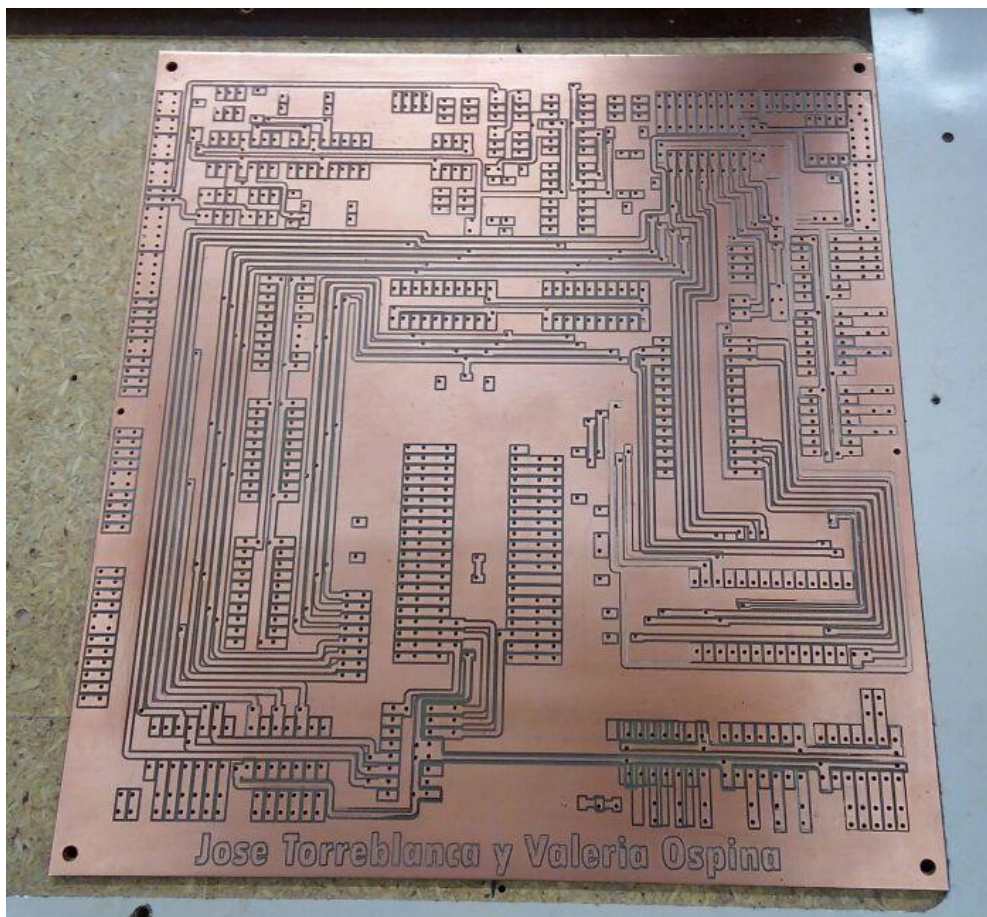


Ilustración 124 - Fresado: placa terminada

2.3.- SOLDADURAS.

2.3.1.- SOLDADURAS PUENTE

Las primeras soldaduras que se realizan son las soldaduras puente. Estas soldaduras son aquellas encargadas de unir el lado de componentes de la placa con el lado de pistas. Se realizan atravesando un pedazo de cable conductor de un lado a otro de la placa en el punto donde se diseñó el puente y soldando las dos puntas salientes. Es importante comprobar la conductividad de las pistas una vez soldados los puentes.

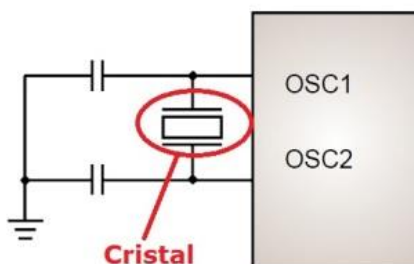
2.3.2.- SOLDADURAS DE COMPONENTES

Una vez soldados todos los puentes de la placa inicia el proceso de soldadura de los zócalos de los componentes. Es importante remarcar que son los zócalos los que se sueldan a la placa y no los componentes directamente. Esto se debe a que los circuitos integrados no resisten la temperatura del proceso de soldado y por lo tanto, de soldarse directamente, la exposición al calor los dañaría.

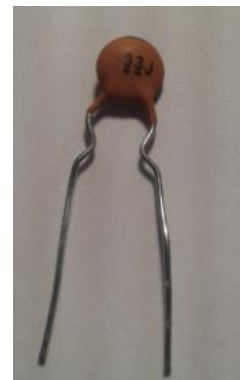
2.3.2.- SOLDADURAS DE ELEMENTOS AISLADOS

Por último, una vez soldados todos los zócalos de los circuitos integrados se sueldan los elementos aislados de la placa. Este grupo de elementos está compuesto por elementos pasivos y activos. Los elementos pasivos son resistencias, potenciómetros y condensadores. El oscilador de cuarzo del microcontrolador constituye el elemento activo y es el encargado de gobernar el tiempo de funcionamiento del sistema.

Este tipo de oscilador incluye en su realimentación un resonador piezoeléctrico, de ahí que su frecuencia de oscilación sea extremadamente estable, cualidad que lo hace perfecto para gobernar la frecuencia de funcionamiento del microcontrolador, es decir, la señal periódica que controla la evolución interna del microcontrolador, la ejecución de las instrucciones programadas. El oscilador se conecta como se ve en la imagen inferior. Los condensadores son de 33 pF con una tolerancia del 5% como lo indica la J en la lenteja.



El



Sección II

COMPONENTES DE LA PLACA DISEÑADA

1.- INTRODUCCIÓN.

La placa principal es el cerebro pensante de todo el proyecto. Se diseñó con el objetivo de poder manipular las señales recibidas de los diversos sensores del motor e integrarlas a un sistema de control por microcontrolador. Las señales tanto de comandos del usuario como señales del propio robot interactúan a través de los diversos circuitos integrados y, siguiendo instrucciones fijadas dentro de la memoria de código del microcontrolador, ejecutan rutinas de control que permiten al brazo del robot realizar los movimientos indicados por el usuario, de una forma controlada y con capacidad de visualización en tiempo real de la trayectoria a través del programa Labview. De esta forma, lo que podría ser un brazo hidráulico de tecnología antigua recupera un gran porcentaje de su funcionalidad en el área de la automatización industrial de procesos. Las funciones de la placa principal se complementan con otras tres placas que se han añadido al sistema de control debido a necesidades surgidas durante el proceso de montaje y programación. Una placa de tratamiento de señales se encarga de filtrar las señales del encoder mientras que otra placa, a la que nos referiremos como placa 8255- 8253, permite la interconexión del bus de datos con los DAC088 y los TL074 mediante el uso de un buffer bidireccional. Por último, los DAC088 y los TL074 se debieron de montar en una placa aparte debido a problemas de conexionado eléctrico entre las básculas 74LS373 y los DAC088. Esta placa recibe el nombre de placa de gestión de motores.

Todos los componentes de la placa cumplen una función determinada dentro del entorno de control del brazo. La funcionalidad básica es claramente el control del movimiento del brazo, el sentido de giro de los motores y cómo éste repercute en el movimiento conjunto de los ejes. De esta tarea se encargan principalmente los amplificadores operacionales del circuito TL074 y los convertidores digitales – analógicos DAC0800. La base de su programación es convertir un byte de información proveniente del usuario a través del microcontrolador en una determinada tensión que termine por variar el tiempo de conmutación de los transistores de potencia del puente en H que conforma la alimentación de los motores de corriente continua. Esto se logra introduciendo la señal de

tensión variante obtenida dentro del esquema de control – potencia original del motor, que se explicó en el capítulo III del presente trabajo. El byte de información es enviado por el usuario a través del bus de datos del microcontrolador. A su vez, el bus de datos se interconecta con los 8255 ubicados en la placa 8255-8253. El 8255 se comporta como un buffer bidireccional, interconectando el bus de datos con la placa que contiene los DAC088 y los TL074. La habilitación de los 8255 se produce mediante un envío de 8 bits desde el microcontrolador pasando a través de una báscula 74LS373.

Una vez controlado el movimiento de los motores interesa conocer su posición. Para ello, se deben interpretar y recibir correctamente las señales provenientes de los encoder incrementales de los motores de los ejes. En este proceso juega un papel muy importante la placa de tratamiento de señales que se explicará en la siguiente sección de este capítulo. Dentro de la placa principal son los circuitos 8259 (gestores de interrupciones) los encargados de iniciar el proceso de conteo de pulsos del encoder, que permite conocer la posición exacta de cada eje del motor una vez tratada la información obtenida. Los 8259 gestionan cada pulso de encoder como una interrupción independiente y direccionan al microcontrolador hacia un bucle que incrementa tres contadores asociados entre sí. Los tres contadores en conjunto pueden contar un máximo de $16.777.216$ pulsos o, lo que es lo mismo, 2 elevado a la 24 (3 registros de un byte enlazados). La placa de tratamiento de señal permite que la señal que entra a los 8259 sea lo más “limpia” posible, sin picos ni interferencias que puedan llevar a una interpretación errónea de la señal del encoder y por lo tanto a un mal conteo y a un desconocimiento de la posición del eje.

Sin embargo, no es suficiente con poder controlar el movimiento y conocer la posición de los ejes del brazo robótico. Para un correcto control de las trayectorias es vital concretar un punto de inicio de todos los movimientos que marque el reinicio de los contadores y las condiciones iniciales para una nueva trayectoria. El robot original realizaba este proceso mediante un chequeo de sensores inductivos ubicados en cada eje (ver capítulo II). En el nuevo sistema de control se programó un proceso similar en el que las señales provenientes de los sensores inductivos llegan al microcontrolador y, mediante una rutina de programación, permiten posicionar los dos ejes del brazo en su posición inicial.

El microcontrolador PIC16F877A permite, además de todo lo anterior, realizar una comunicación serie con el ordenador. Esta comunicación se gestiona a través de una interfaz de usuario creada mediante el programa Labview que, a través del envío de diversos pares de códigos al microcontrolador, permite la realización por parte del robot de diversas tareas preprogramadas. La información fluye en ambos sentidos, por lo que las lecturas de posición obtenidas de los encoder se gestionan dentro del entorno de Labview y permiten posicionar en tiempo real el esquema 3D del brazo robótico.

2.- MICROCONTROLADOR PIC16F877A.

2.1.- INTRODUCCIÓN.

El microcontrolador PIC16F877A es un modelo en el que evoluciona el PIC16F877, que pertenece a la subfamilia PIC16F87X conformada por cuatro modelos (PIC16F873, 4, 6,7).

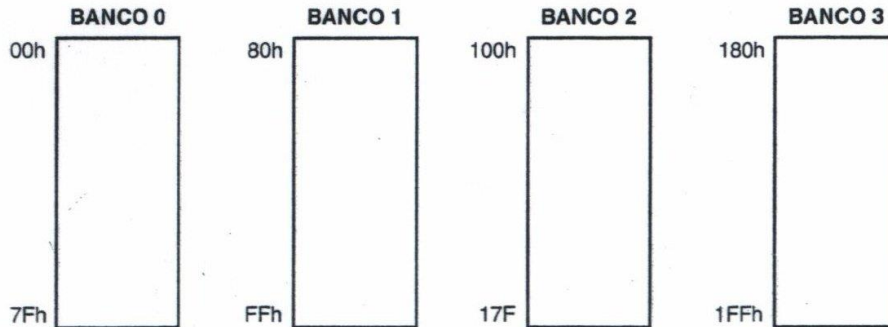
Ambos modelos difieren en muy pocos detalles y estos se resumen en la siguiente tabla. De ahí que en el presente capítulo se haga referencia al PIC16F877 y no al PIC16F877A.

CARACTERÍSTICA	PIC16F877	PIC16F877A
Número de terminales	40	40
Número de “timers”	3	3
Número de causas de interrupciones	13 o 14	14 o 15
Protocolos de comunicación	PSP, USART, SSP (SPI,I ² C Maestro/Esclavo)	PSP, USART, SSP (SPI,I ² C Maestro/Esclavo)
Frecuencia de funcionamiento	20 MHz	20 MHz
Voltaje	2.2 V – 5.5 V	2.0 V – 5.5 V
Convertor A/D	10-bits, 4 selecciones de relojes de conversión	10-bits, 7 selecciones de relojes de conversión
CCP	2	2
Comparador analógico	-	2
Referencia de voltaje del comparador	-	Sí
Memoria de programa	4K, 8K Flash (Borrar/ Escribir en byte único)	4K, 8K Flash (Borrar/ Escribir en bloques de cuatro bytes)
RAM	192, 368 bytes	192, 368 bytes
Datos de la EEPROM	128, 256 bytes	128, 256 bytes
Protección del código	Segmentada, inicia al final de la memoria de programa	On/Off
Protección de escritura de la memoria de programa	On/Off	Segmentada, inicia al principio de la memoria de programa
Otros	Depurador en-circuito, programación a bajo voltaje	Depurador en-circuito, programación a bajo voltaje

Tabla 3 - Table comparativa PIC16F877 / PIC16F877A

Estos microcontroladores tienen una memoria de tipo FLASH, por lo que es posible realizar acciones de lectura y escritura de múltiples posiciones un número muy alto de veces (hasta mil operaciones de escritura/ borrado). Además, se tiene la posibilidad de grabar y borrar programas eléctricamente lo cual permite que el microcontrolador esté insertado en el zócalo durante todo el proceso de diseño y depuración. De este modo, la tecnología flash permite mediante impulsos eléctricos velocidades de funcionamiento muy superiores a las de otras tecnologías anteriores como la EEPROM, utilizada en el modelo anterior de microcontrolador PIC16C84. El PIC16F877 cuenta específicamente con una memoria FLASH de 8192x14 (8K palabras de 14 bits), una memoria de datos EEPROM de 256 bytes (que soporta 100000 operaciones de escritura/borrado) y una RAM de 368 bytes. La estructura

de la memoria RAM se basa en cuatro bancos seleccionables con los bits RP0 y RPI del registro estado.



El PIC16F877 cuenta con hasta catorce posibles fuentes de interrupción, tres “timer” (temporizadores) y cinco puertas. El PIC mencionado dispone a su vez de dos módulos CCP (captura-comparación-PWM) que son capaces de capturar y comparar impulsos. La captura se efectúa con una precisión de 12,5 ns y una resolución de 16 bits, mientras que la comparación con igual resolución alcanza una precisión de 200 ns. Además, la sección PWM varía la anchura de los pulsos, técnica muy empleada en el control de motores.

En cuanto a la comunicación serie admite dos modelos. Por un lado, la USART (receptor/transmisor universal síncrono/asíncrono) orientada a la comunicación entre subsistemas o máquinas (RS232). Por otro, la MSSP (puerto serie maestro síncrono) que admite los protocolos I2C y SPI y está destinada a la comunicación entre diversos circuitos integrados. Para la comunicación en paralelo en este modelo de PIC está disponible el protocolo PSP que necesita todas las ocho líneas del puerto D y tres de control de la puerta E. Los de cuarenta terminales integran una puerta paralela esclava.

En todos los PIC16F87X existe un convertor analógico/digital de 10 bits con 8 canales de entrada en los de 40 terminales (PIC16F877). Diferencias entre el que termina en A y el que no.

A modo de resumen se nombran a continuación sus recursos fundamentales y los dispositivos periféricos de los que dispone.

- Procesador de arquitectura RISC avanzada.
- Juego de 35 instrucciones con 14 bits de longitud. Todas ellas se ejecutan en un ciclo de instrucción, menos las de salto que tardan dos.
- Frecuencia de 20 MHz.
- Hasta 8 k palabras de 14 bits para la memoria de programa, tipo FLASH.
- Hasta 368 bytes de memoria de datos RAM.
- Hasta 256 bytes de memoria de datos EEPROM.
- Encapsulados compatibles con los PIC16C73/74/76/77.
- Hasta 14 fuentes de interrupción internas y externas.
- Pila con 8 niveles.
- Modos de direccionamiento directo, indirecto y relativo.
- Perro guardián (WDT).

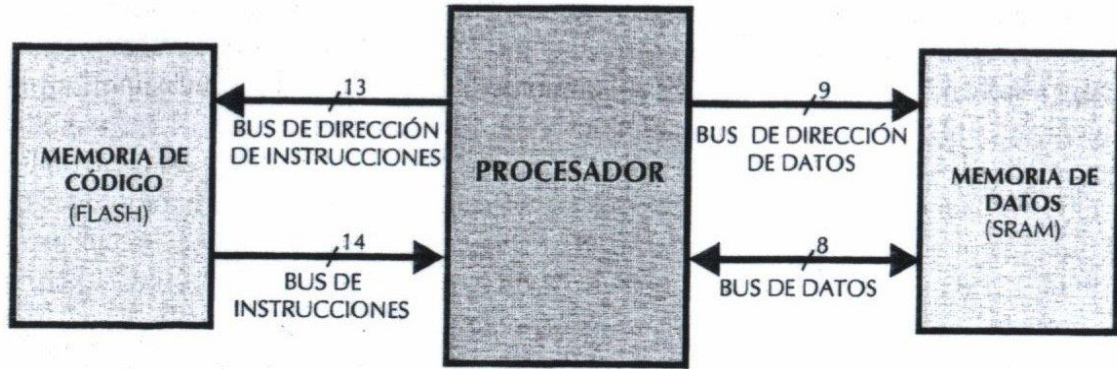
- Código de protección programable.
- Modo SLEEP de bajo consumo.
- Programación serie en circuito con dos terminales.
- Tensión de alimentación comprendido entre 2 y 5,5 V.
- Bajo consumo, menos de 2mA a 5V y 5MHz.
- Timr0: temporizador-contador de 8 bits con predivisor de 8 bits.
- Timer1: temporizador-contador de 16 bits con predivisor.
- Timer2: temporizador de 8 bits con predivisor y postdivisor.
- Módulo de Captura-Comparación-PWM.
- Conversor A/D de 10 bits.
- Puerto serie síncrono (SSP) con SPI e I2C.
- Transmisor/receptor síncrono/asíncrono (USART).
- Puerta paralela esclava (PSP), sólo en los de 40 terminales.

2.2.- ARQUITECTURA, DIAGRAMA DE CONEXIONES Y REPERTORIO DE INSTRUCCIONES.

2.2.1.- PROCESADOR RISC CON ARQUITECTURA HARVARD.

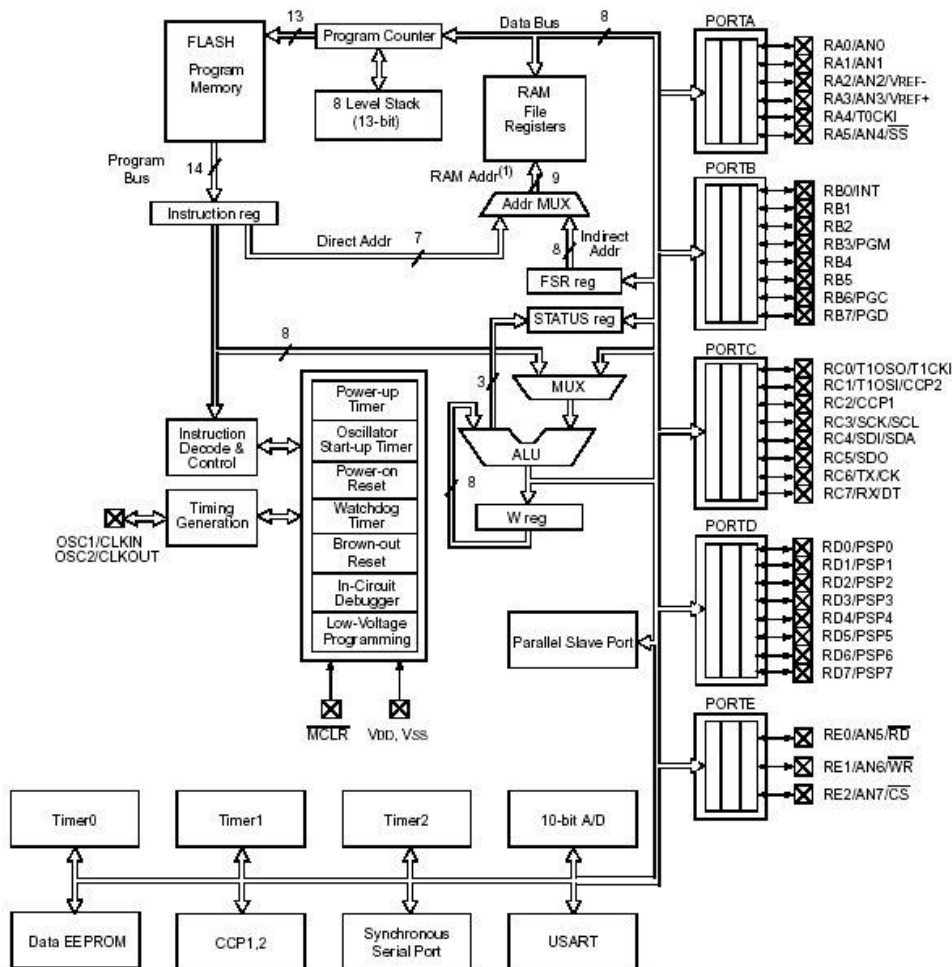
Esta arquitectura se caracteriza por la independencia entre la memoria de código y la de datos. Así, tanto la capacidad como el tamaño de los buses de cada memoria se adaptan estrictamente a las necesidades del diseño, facilitando el trabajo en paralelo de las dos memorias, lo que permite obtener las altas tasas de rendimiento características.

La arquitectura computacional RISC (del inglés “reduced instruction set computer”) es un tipo de diseño de CPU que se caracteriza por sus instrucciones de tamaño fijo y presentadas en un número reducido de formatos y porque sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos. La filosofía RISC se hace patente en el reducido número de instrucciones que forman su repertorio, sólo consta de 35 instrucciones que se ejecutan en un ciclo de instrucción equivalente a cuatro períodos de reloj, excepto las de salto que necesitan dos ciclos



La memoria de código está direccionada por el PC (contador de programa) en conexión con la pila de 8 niveles. La memoria de datos RAM contiene el banco de registros específicos y el banco de los registros de propósito general y transfiere información bidireccional por el bus de datos de ocho líneas que interconecta todos los elementos. Finalmente, el camino de datos está formado por una ALU de 8 bits que trabaja con el registro de trabajo W (acumulador).

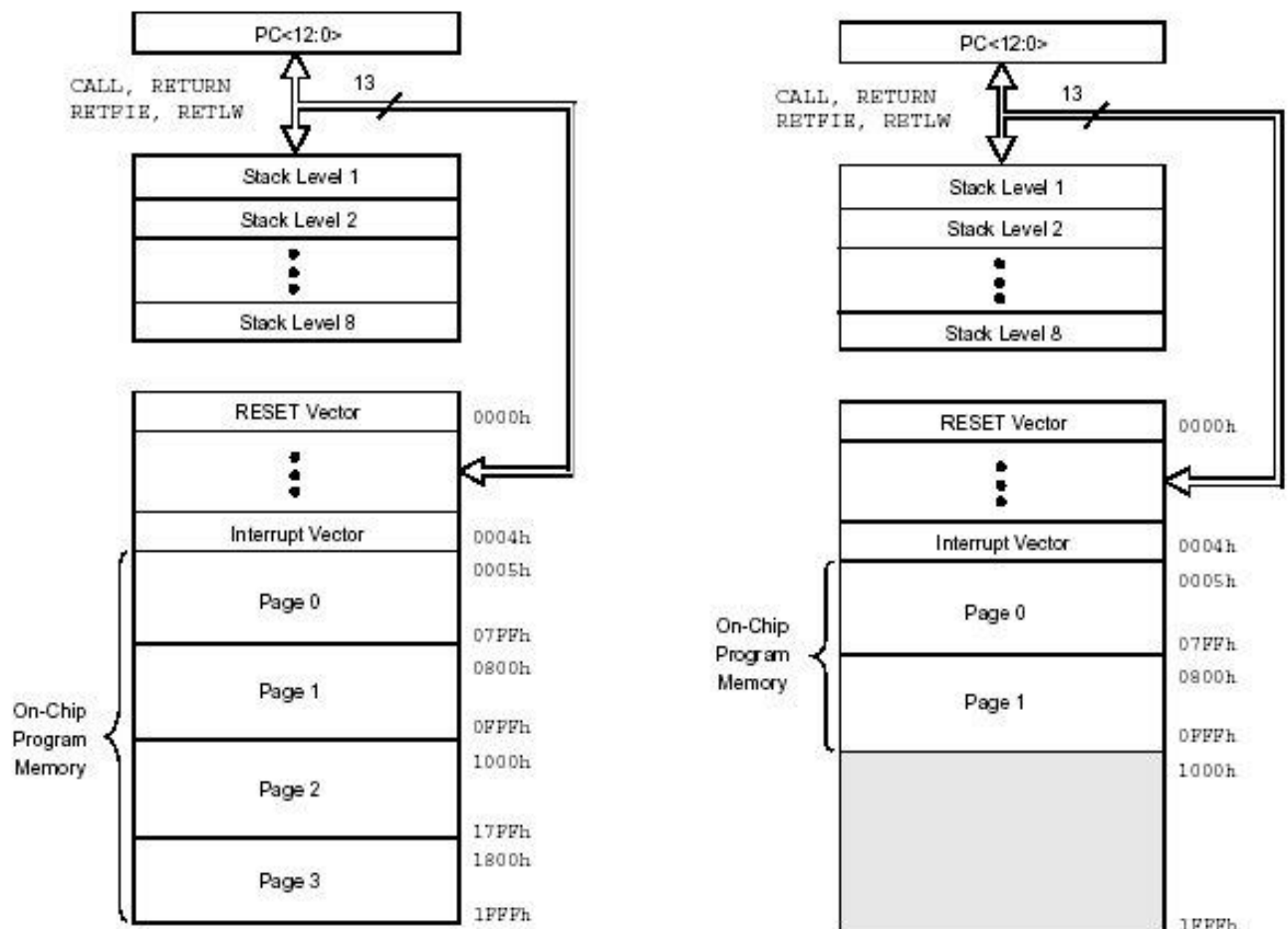
En la siguiente figura se observa en detalle la arquitectura correspondiente al modelo de PIC16F877.



Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes

2.2.2.- ORGANIZACIÓN DE LA MEMORIA DE PROGRAMA.

La memoria de código o programa tipo FLASH en la que se graba el programa de aplicación en los PIC16F877 tiene una capacidad de 8K palabras de 14 bits cada una. Dicha memoria está dividida en páginas de 2K palabras y está direccionada con el PC, que tiene un tamaño de 13 bits. Con la instrucción CALL el valor del PC se salva en el nivel superior de la pila. Con las instrucciones RETURN, RETFIE y RETLW el valor contenido en el nivel superior de la pila se carga en el PC. Al poseer la pila sólo ocho niveles le corresponde al programador preocuparse por los anidamientos en las subrutinas. Para no sobrepasar dicho valor el vector de reset ocupa la dirección 0000h y el vector de interrupción la 0004h, igual que en el PIC16F84.



2.2.3.- ORGANIZACIÓN DE LA MEMORIA DE DATOS RAM.

La memoria de datos tiene posiciones implementadas en RAM y otras en EEPROM. En la sección RAM, se alojan los registros operativos fundamentales en el funcionamiento del procesador y en el manejo de todos sus periféricos, además de registros que el programador puede usar para información de trabajo propia de la aplicación. La RAM estática consta de 4 bancos con 128 bytes cada uno. En las posiciones iniciales de cada banco se ubican los registros específicos que gobiernan al procesador y sus recursos. El modelo de PIC16F877A tiene 368 bytes, la siguiente figura presenta los cuatro bancos de la memoria RAM, indicando en las primeras posiciones de cada uno los nombres de los registros que contienen.

INDF	00h	INDF	80h	INDF	100h	INDF	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD	08h	TRISD	88h		108h		188h
PORTE	09h	TRISE	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reservado	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reservado	18Fh
TICON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h	Registros de propósito general	114h	Registros de propósito general	194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah	16 bytes	11Ah	16 bytes	19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh

ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh		
Registros de propósito general 96 bytes	20h	Registros de propósito general 80 bytes	A0h	Registros de propósito general 80 bytes	120h	Registros de propósito general 80 bytes	1A0h		
	7Fh		Mapeados con 70h-7Fh		EFh		16Fh	Mapeados con 70h-7Fh	1EFh
					F0h		170h		1F0h
		FFh		17Fh	1FFh				
	Banco 0		Banco 1		Banco 2		Banco 3		

Para seleccionar el banco al que se desea acceder en la RAM se emplean los bits 7, 6 y 5 del registro de estado, denominados IRP, RPI y RP0 respectivamente, según el código siguiente:

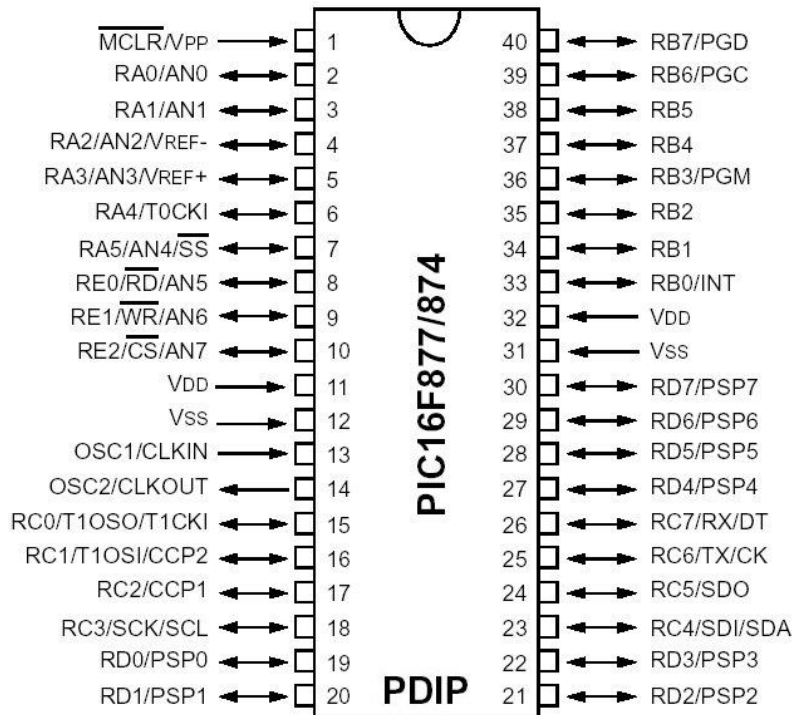
BANCO	IRP	RPI	RP0
0	0	0	0
1	0	0	1
2	1	1	0
3	1	1	1

Para manipular adecuadamente los elementos internos de los PIC hay que conocer la misión de los bits de los registros específicos. Los trece bits contenidos en el PC que direccionan la memoria de código están guardados en dos registros específicos. El registro PCL guarda los 8 bits de menos peso y se puede leer y escribir. Los bits de mayor peso del PC (12...8) se alojan en el registro PCH, que al no poder ser leído ni escrito, se accede a él a través del PCLATH. Las instrucciones de salto CALL y GOTO sólo proporcionan 11 bits de la dirección a saltar. Esto limita el salto dentro de cada banco de 2 K. Cuando se desea salir del banco actual hay que programar correctamente los bits PCLATH (4,3) que seleccionan al banco.

Para direccionar la memoria RAM de datos estructurada en cuatro bancos de 128 bytes cada uno existen dos métodos diferentes: direccionamiento indirecto y direccionamiento directo. En el modo de direccionamiento directo los bits RPI y RP0 del registro de estado (6 y 5) se encargan de seleccionar el banco, mientras que la dirección dentro del banco la determinan siete bits procedentes del código OP de la instrucción. Para el direccionamiento indirecto se usa el registro FSR en el que sus siete bits de menos peso señalan la dirección y el banco lo determina el bit de más peso de FSR concatenando con el bit IRP del registro de estado (7).

2.2.4.- DIAGRAMA DE CONEXIONADO.

En la siguiente figura se muestra el diagrama de distribución y asignación de los cuarenta terminales de los encapsulados DIP (del inglés “dual in-line package”) del PIC16F877.



La asignación de funciones es la siguiente:

Terminales de propósito general

- OSC1/CLKIN (9): entrada del cristal de cuarzo o del oscilador externo.
- OSC2/CLKOUT (10): salida del cristal de cuarzo. En modo RC el terminal OSC2 saca la cuarta parte de la frecuencia que se introduce por OSC1, que determina el ciclo de instrucción.
- Vss (8, 19): conexión a tierra.
- VDD (20): entrada de la alimentación positiva.
- MCLR#/VPP/THV (1): entrada de reset o entrada de tensión de programación o tensión alta en el modo test.

Puerta A

- RA0/AN0 (2): puede actuar como línea digital de E/S o como entrada analógica al convertor AD (canal 0).
- RA1/ANI (3): igual que la anterior.
- RA2/AN2/VREF- (4): puede ser línea digital de E/S, entrada analógica o entrada de tensión negativo de referencia.
- RA3/AN3/VREF+ (5): línea digital de E/S, entrada analógica o entrada de tensión de referencia positivo.
- RA4/T0CKI (6): línea digital de E/S o entrada del reloj del Timer0. Salida con colector abierto.

- RA5/SS#/AN4 (7): línea digital de E/S, entrada analógica o selección como esclavo de la puerta serie síncrona.

Puerta B

- RB0/INT (21): línea digital de E/S o entrada de petición de interrupción externa.
- RB1 (22): línea de E/S digital.
- RB2 (23): línea de E/S digital.
- RB3/PGM (24): línea de E/S digital o entrada del voltaje bajo para programación.
- RB4 (25): línea de E/S digital.
- RB5 (26): línea de E/S digital.
- RB6/PGC (27): línea digital de E/S. En la programación serie recibe las señales de reloj.
- RB7/PGD (28): línea digital de E/S. En la programación serie actúa como entrada de datos.

Las líneas RB0 y RB1 se conectan respectivamente al INT y al INTA de 8259, RB2 es la línea que indica la dirección del 74LS245 y RB6/RB7 son las líneas de la señal de entrada al amplificador del motor 1 y del motor 2 respectivamente.

Puerta C

- RC0/TIOSO/TICKI (11): línea digital de E/S o salida del oscilador del Timer1 o como entrada de reloj del Timer1.
- RC1/TIOSI/CCP2 (12): línea digital de E/S o entrada al oscilador del Timer1 o entrada al módulo Captura2/salida Comparación2/salida de PWM2.
- RC2/CCPI (13): E/S digital. También puede actuar como entrada Captura1/Salida Comparación1/salida de PWM1.
- RC3/SCK/SCL (14): E/S digital o entrada de reloj serie síncrona/salida de los modos SPI e I2C.
- RC4/SDI/SDA (15): E/S digital o entrada de datos en modo SPI o I/O datos en modo I2C.
- RC5/SDO (16): E/S digital o salida de datos en modo SPI.
- RC6/TX/CK (17): E/S digital o pin del transmisor del USART asíncrono o como reloj del síncrono.
- RC7/RX/DT (18): E/S digital o receptor del USART asíncrono o como datos en el síncrono.

En este caso específico se conecta el terminal RC0 al E3 del decodificador 74LS138, el RC3 al A2, el RC5 al A1 y el RC4 al A0. Se obtiene de esta forma un direccionamiento de hasta 8 dispositivos, la tabla de verdad es la siguiente:

TRUTH TABLE

INPUTS						OUTPUTS							
E ₁	E ₂	E ₃	A ₀	A ₁	A ₂	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

El terminal RC1 se conecta a la línea E del LCD, RC2 a la RS, RC6 a la línea TX RS232 y RC7 a la línea RX RS232. Incluir descripción de la pantalla LCD.

Puerta D

- RD0/PSP0-RD7/PSP7: los 8 terminales de esta puerta pueden actuar como líneas de E/S digitales o como líneas para transferencia de información en la comunicación de la puerta paralela esclava.

Puerta E

- Sólo tiene 3 terminales:
- RE0/RD#/AN5: E/S digital o señal de lectura para la puerta paralela esclava o entrada analógica (canal 5).
- RE1/WR#/AN6: E/S digital o señal de escritura para la puerta paralela esclava o entrada analógica al conversor A/D (canal 6).
- RE2/CS#/AN7: E/S digital o activación/desactivación de la puerta paralela esclava o entrada analógica (canal 7).

La línea RE0 se conecta al A0 del 8259, RE1 a WR y RE2 al RD.

2.2.5.- REPERTORIO DE INSTRUCCIONES.

Los mismos formatos, iguales modos de direccionamiento y las mismas 35 instrucciones que tenía el PIC16F84 sirven para todos los modelos PIC16F87X. En la siguiente tabla se muestran las instrucciones que manejan registros y que manejan bits.

<p>ADDWF W + F Sintaxis: [label] ADDWF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(W) + (f) \Rightarrow (\text{dest})$ Flags afectados: C, DC, Z Código OP: 00 0111 dfff ffff Descripción: Suma el contenido del registro W y el registro f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>	<p>ANDWF W AND F Sintaxis: [label] ANDWF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(W) \text{ AND } (f) \Rightarrow (\text{dest})$ Flags afectados: Z Código OP: 00 0101 dfff ffff Descripción: Realiza la operación lógica AND entre los registros W y f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>	<p>CLRF Borra un registro Sintaxis: [label] CLRF f Operandos: $0 \leq f \leq 127$ Operación: $0x00 \Rightarrow (f), 1 \Rightarrow Z$ Flags afectados: Z Código OP: 00 0001 1fff ffff Descripción: El registro f se carga con 0x00. El flag Z se activa.</p>
<p>CLRW Borra el registro W Sintaxis: [label] CLRW Operandos: Ninguno Operación: $0x00 \Rightarrow W, 1 \Rightarrow Z$ Flags afectados: Z Código OP: 00 0001 0xxx xxxx Descripción: El registro de trabajo W se carga con 0x00. El flag Z se activa.</p>	<p>COMF Complemento de f Sintaxis: [label] COMF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(f), 1 \Rightarrow (\text{dest})$ Flags afectados: Z Código OP: 00 1001 dfff ffff Descripción: El registro f es complementado. El flag Z se activa si el resultado es 0. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>	<p>DECF Decremento de f Sintaxis: [label] DECF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(f) - 1 \Rightarrow (\text{dest})$ Flags afectados: Z Código OP: 00 0011 dfff ffff Descripción: Decrementa en 1 el contenido de f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>
<p>INCF Incremento de f Sintaxis: [label] INCF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(f) + 1 \Rightarrow (\text{dest})$ Flags afectados: Z Código OP: 00 1010 dfff ffff Descripción: Incrementa en 1 el contenido de f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>	<p>IORWF W OR F Sintaxis: [label] IORWF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(W) \text{ OR } (f) \Rightarrow (\text{dest})$ Flags afectados: Z Código OP: 00 0100 dfff ffff Descripción: Realiza la operación lógica OR entre los registros W y f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>	<p>MOVF Mover a f Sintaxis: [label] MOVF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(f) \Rightarrow (\text{dest})$ Flags afectados: Z Código OP: 00 1000 dfff ffff Descripción: El contenido del registro f se mueve al destino d. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f. Permite verificar el registro, puesto que afecta a Z.</p>
<p>MOVWF Mover a f Sintaxis: [label] MOVWF f Operandos: $0 \leq f \leq 127$ Operación: $W \Rightarrow (f)$ Flags afectados: Ninguno Código OP: 00 0000 1fff ffff Descripción: El contenido del registro W pasa al registro f.</p>	<p>NOP No operar Sintaxis: [label] NOP Operandos: Ninguno Operación: No operar Flags afectados: Ninguno Código OP: 00 0000 0xxx 0000 Descripción: No realiza operación alguna. En realidad consume un ciclo de instrucción sin hacer nada.</p>	<p>RLF Rota f a la izquierda Sintaxis: [label] RLF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: Rotación a la izda Flags afectados: C Código OP: 00 1101 dfff ffff Descripción: El contenido de f se rota a la izquierda. El bit de menos peso de f pasa al carry (C), y el carry se coloca en el de mayor peso. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>
<p>RRF Rota f a la derecha Sintaxis: [label] RRF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: Rotación a la dcha Flags afectados: C Código OP: 00 1100 dfff ffff Descripción: El contenido de f se rota a la derecha. El bit de menos peso de f pasa al carry (C), y el carry se coloca en el de mayor peso. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>	<p>SUBWF Resta f - W Sintaxis: [label] SUBWF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(f) - (W) \Rightarrow (\text{dest})$ Flags afectados: C, DC, Z Código OP: 00 0010 dfff ffff Descripción: Mediante el método del complemento a dos el contenido de W es restado al de f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>	<p>XORWF W XOR F Sintaxis: [label] XORWF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(W) \text{ XOR } (f) \Rightarrow (\text{dest})$ Flags afectados: Z Código OP: 00 0110 dfff ffff Descripción: Realiza la operación lógica XOR entre los registros W y f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>
<p>SWAPF Intercambio de f Sintaxis: [label] SWAPF f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(f \langle 3:0 \rangle) \Rightarrow (f \langle 7:4 \rangle)$ Flags afectados: Ninguno Código OP: 00 1110 dfff ffff Descripción: Los 4 bits de más peso y los 4 de menos son intercambiados. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f.</p>	<p>BCF Borra un bit Sintaxis: [label] BCF f,b Operandos: $0 \leq f \leq 127, 0 \leq b \leq 7$ Operación: $0 \Rightarrow (f \langle b \rangle)$ Flags afectados: Ninguno Código OP: 01 00bb bfff ffff Descripción: Borra el bit b del registro f</p>	<p>BSF Activa un bit Sintaxis: [label] BSF f,b Operandos: $0 \leq f \leq 127, 0 \leq b \leq 7$ Operación: $1 \Rightarrow (f \langle b \rangle)$ Flags afectados: Ninguno Código OP: 01 01bb bfff ffff Descripción: Activa el bit b del registro f</p>

<p>BTFSZ Test de bit y salto Sintaxis: [label] BTFSZ f,b Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: Salto si $(f \ll b) = 0$ Flags afectados: Ninguno Código OP: 01 10bb bfff ffff Descripción: Si el bit b del registro f es 0, se salta una instrucción y se continúa con la ejecución. En caso de salto, ocupará dos ciclos de reloj.</p>	<p>BTFSZ Test de bit y salto Sintaxis: [label] BTFSZ f,b Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: Salto si $(f \ll b) = 1$ Flags afectados: Ninguno Código OP: 01 11bb bfff ffff Descripción: Si el bit b del registro f es 1, se salta una instrucción y se continúa con la ejecución. En caso de salto, ocupará dos ciclos de reloj.</p>	<p>INCFSZ Incremento y salto Sintaxis: [label] INCFSZ f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(f) - 1 \Rightarrow d$; Salto si $R=0$ Flags afectados: Ninguno Código OP: 00 1111 dfff ffff Descripción: Incrementa el contenido del registro f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f. Si la resta es 0 salta la siguiente instrucción, en cuyo caso costaría 2 ciclos.</p>
<p>DECFSZ Decremento y salto Sintaxis: [label] DECFSZ f,d Operandos: $d \in [0,1], 0 \leq f \leq 127$ Operación: $(f) - 1 \Rightarrow d$; Salto si $R=0$ Flags afectados: Ninguno Código OP: 00 1011 dfff ffff Descripción: Decrementa el contenido del registro f. Si d es 0, el resultado se almacena en W, si d es 1 se almacena en f. Si la resta es 0 salta la siguiente instrucción.</p>	<p>ANDLW W AND literal Sintaxis: [label] ANDLW k Operandos: $0 \leq k \leq 255$ Operación: $(W) \text{ AND } (k) \Rightarrow (W)$ Flags afectados: Z Código OP: 11 1001 kkkk kkkk Descripción: Realiza la operación lógica AND entre el contenido del registro W y k, guardando el resultado en W.</p>	<p>ADDLW Suma un literal Sintaxis: [label] ADDLW k Operandos: $0 \leq k \leq 255$ Operación: $(W) + (k) \Rightarrow (W)$ Flags afectados: C, DC, Z Código OP: 11 111x kkkk kkkk Descripción: Suma el contenido del registro W y k, guardando el resultado en W.</p>
<p>IORLW W OR literal Sintaxis: [label] IORLW k Operandos: $0 \leq k \leq 255$ Operación: $(W) \text{ OR } (k) \Rightarrow (W)$ Flags afectados: Z Código OP: 11 1000 kkkk kkkk Descripción: Se realiza la operación lógica OR entre el contenido del registro W y k, guardando el resultado en W.</p>	<p>MOVLW Cargar literal en W Sintaxis: [label] MOVLW k Operandos: $0 \leq k \leq 255$ Operación: $(k) \Rightarrow (W)$ Flags afectados: Ninguno Código OP: 11 00xx kkkk kkkk Descripción: El literal k pasa al registro W.</p>	<p>SUBLW Resta Literal - W Sintaxis: [label] SUBLW k Operandos: $0 \leq k \leq 255$ Operación: $(k) - (W) \Rightarrow (W)$ Flags afectados: Z, C, DC Código OP: 11 110x kkkk kkkk Descripción: Mediante el método del complemento a dos el contenido de W es restado al literal. El resultado se almacena en W.</p>
<p>XORLW W OR literal Sintaxis: [label] XORLW k Operandos: $0 \leq k \leq 255$ Operación: $(W) \text{ XOR } (k) \Rightarrow (W)$ Flags afectados: Z Código OP: 11 1010 kkkk kkkk Descripción: Se realiza la operación lógica XOR entre el contenido del registro W y k, guardando el resultado en W.</p>	<p>CALL Salto a subrutina Sintaxis: [label] CALL k Operandos: $0 \leq k \leq 2047$ Operación: $PC \Rightarrow \text{Pila}; k \Rightarrow PC$ Flags afectados: Ninguno Código OP: 10 0kkk kkkk kkkk Descripción: Salto a una subrutina. La parte baja de k se carga en PCL, y la alta en PCLATCH. Ocupa 2 ciclos de reloj.</p>	<p>CLRWDW Borra el WDT Sintaxis: [label] CLRWDW Operandos: Ninguno Operación: $0x00 \Rightarrow WDT, 1 \Rightarrow /TO$ $1 \Rightarrow /PD$ Flags afectados: /TO, /PD Código OP: 00 0000 0110 0100 Descripción: Esta instrucción borra tanto el WDT como su preescaler. Los bits /TO y /PD del registro de estado se ponen a 1.</p>
<p>GOTO Salto incondicional Sintaxis: [label] GOTO k Operandos: $0 \leq k \leq 2047$ Operación: $k \Rightarrow PC \ll 8:0$ Flags afectados: Ninguno Código OP: 10 1kkk kkkk kkkk Descripción: Se trata de un salto incondicional. La parte baja de k se carga en PCL, y la alta en PCLATCH. Ocupa 2 ciclos de reloj.</p>	<p>RETFIE Retorno de interrup Sintaxis: [label] RETFIE Operandos: Ninguno Operación: $1 \Rightarrow GIE; TOS \Rightarrow PC$ Flags afectados: Ninguno Código OP: 00 0000 0000 1001 Descripción: El PC se carga con el contenido de la cima de la pila (TOS): dirección de retorno. Consume 2 ciclos. Las interrupciones vuelven a ser habilitadas.</p>	<p>RETLW Retorno, carga W Sintaxis: [label] RETLW k Operandos: $0 \leq k \leq 255$ Operación: $(k) \Rightarrow (W); TOS \Rightarrow PC$ Flags afectados: Ninguno Código OP: 11 01xx kkkk kkkk Descripción: El registro W se carga con la constante k. El PC se carga con el contenido de la cima de la pila (TOS): dirección de retorno. Consume 2 ciclos.</p>
<p>RETURN Retorno de rutina Sintaxis: [label] RETURN Operandos: Ninguno Operación: $TOS \Rightarrow PC$ Flags afectados: Ninguno Código OP: 00 0000 0000 1000 Descripción: El PC se carga con el contenido de la cima de la pila (TOS): dirección de retorno. Consume 2 ciclos.</p>	<p>SLEEP Modo bajo consumo Sintaxis: [label] SLEEP Operandos: Ninguno Operación: $0x00 \Rightarrow WDT, 1 \Rightarrow /TO$ $0 \Rightarrow WDT \text{ Preescaler}, 0 \Rightarrow /PD$ Flags afectados: /PD, /TO Código OP: 00 0000 0110 0011 Descripción: El bit de energía se pone a 0, y a 1 el de descanso. El WDT y su preescaler se borran. El micro para el oscilador, llendo al modo "durmiente".</p>	

En el PIC16F877 existen nuevos registros específicos de control cuyos bits se deberán escribir o leer para su gobierno.

NEMÓNICO	DESCRIPCIÓN	CÓDIGO OP	FLAGS AFECTADOS	NOTAS
Instrucciones orientadas a registros				
ADDWF f,d	(W)+(f) a (destino)	00 0111 dfff ffff	C, DC, Z	1,2
ANDWF f,d	(W) AND (f) a (destino)	00 0101 dfff ffff	Z	1,2
CLRF f	00 a (f)	00 0001 1fff ffff	Z	2
CLRW	00 a (W)	00 0001 0000 0011	Z	
COMF f,d	Complemento de f [(#f)à (destino)]	00 1001 dfff ffff	Z	1,2
DECF f,d	(f)-I a destino	00 0011 dfff ffff	Z	1,2
DECFSZ f,d	(f)-I a destino y si resultado es 0 salta	00 1011 dfff ffff	Ninguno	1,2,3
INCF f,d	(f)+I a destino	00 1010 dfff ffff	Z	1,2
INCFSZ f,d	(f)+I a destino y si resultado es 0 salta	00 1111 dfff ffff	Ninguno	1,2,3
IORWF f,d	(W) OR (f) a destino	00 0100 dfff ffff	Z	1,2
MOVWF f	Mueve f a destino	00 1000 dfff ffff	Ninguno	
NOP	(W) a (f)	00 0000 1fff ffff	Ninguno	
RLF f,d	No operación	00 0000 0xx0 0000	C	1,2
RRF f,d	Rota f a la izq a través del carry a destino	00 1101 dfff ffff	C	1,2
SUBWF f,d	Rota f a la dcha a través del carry a destino	00 1100 dfff ffff	C, DC, Z	1,2
SWAPF f,d	(f)-(W) a (destino)	00 0010 dfff ffff	Ninguno	1,2
XORWF f,d	Cambia 4 bits de más peso por los 4 bits de menos peso de f a destino	00 1110 dfff ffff	Z	1,2
	(W) XOR (f) a (destino)	00 0110 dfff ffff		
Instrucciones orientadas a bits				

BCF f,b	Pone a 0 el bit b del registro f	01 00bb bfff ffff	Ninguno	1,2
BSF f,b	Pone a 1 el bit b del registro f	01 01bb bfff ffff	Ninguno	1,2
BTFSC f,b	Brinco si el bit b del reg. f es 0	01 10bb bfff ffff	Ninguno	3
BTFSS f,b	Brinco si el bit b del reg. f es 1	01 11bb bfff ffff	Ninguno	3
Instrucciones con literales y de control				
ADDLW K	(W)+ K a (W)			
ANDLW K	(W) AND K a (W)			
CALL K	Llamada a subrutina			
CLRWDT	Borado del temporizador del WD			
GOTO K	Go To dirección			
IORLW K	(W) OR K a (W)			
MOVLW K	K a (W)			
RETFIE	Retorno de una interrupción			
RETLW K	Retorno con un literal en W			
RETURN	Retorno de una subrutina			
SLEEP	Modo Standby			
SUBLW K	K – (W) a W			
XORLW K	(W) XOR K a (W)			

Notas:

1. Al modificar un registro de E/S con una operación sobre él mismo (por ejemplo MOVF PORTB, 1), el valor utilizado es el que esté presente en los pines del PORTB. Por ejemplo, si el biestable tiene un "1" para una patilla configurada como entrada y se pone a nivel bajo desde el exterior, el dato se volverá a escribir como "0".

2. Si se ejecuta esta instrucción sobre el TMR0 y d=1, será borrado el divisor de frecuencia (preescaler), si está asignado al TMR0.

3. Si se modifica el contador de programa (PC) o una condición de prueba es verdadera, la instrucción requiere dos ciclos máquina. El segundo ciclo se ejecuta como un NOP.

2.3.- PRINCIPALES REGISTROS DE CONTROL Y MANEJO DE LAS MEMORIAS FLASH Y EEPROM.

Para gobernar el funcionamiento de los recursos de los PIC existe un conjunto de registros específicos cuyos bits soportan el control del dispositivo. Dichos registros están ubicados en las primeras posiciones de cada banco de la memoria de datos RAM. En el apartado "Organización de la memoria de datos RAM" se muestra el contenido de la memoria RAM y la denominación y situación de los registros específicos junto a los de propósito general. Las posiciones 00h, 80h, 100h y 180h no son registros físicos sino direcciones indirectas. A continuación se describe la misión de los bits de los principales registros de control.

2.3.1.- EL REGISTRO ESTADO (STATUS).

Sin duda éste es el registro más usado de todos, pues sus bits están destinados a controlar las funciones vitales del procesador. Por este motivo, está duplicado en las cuartas posiciones de cada banco (03h, 83h, 103h y 183h).

Registro ESTADO	7	6	5	4	3	2	1	0
	IRP	RPI	RP0	TO#	PD#	Z	DC	C
	R/W	R/W	R/W	R	R	R/W	R/W	R/W
C=Señalizador de bit de acarreo 1=Acarreo en la suma y no en la resta 0=No acarreo en la suma y si en la resta								
DC=Señalizador de acarreo en el 4 bit de menor peso 1=Acarreo en la suma y no en la resta 0=No acarreo en la suma y si en la resta								
Z=Señalizador de cero 1=El resultado de un operación es cero 0= El resultado de un operación es distinto de cero								
PD#=Bit de POWER DOWN 1=Tras conectar la alimentación VDD o ejecutar la instrucción CLRWDT 0=Al ejecutar la instrucción SLEEP								
TO#=Bit de TIMER OUT 1=Tras conectar la alimentación VDD o ejecutar CLRWDT o SLEEP 0=Al rebasar el tiempo del WDT								
RPI-RP0=Selección del banco para el direccionamiento directo 0 0=Banco 0 (dirección 00 7F) 0 1=Banco 1 (dirección 80 FF) 1 0=Banco 2 (dirección 100 17F) 1 1=Banco 3 (dirección 180 1FF)								
IRP=Bit para seleccionar el banco para el direccionamiento indirecto. 0=Bancos 0 y 1 (00-FF) 1=Bancos 2 y 3 (100-1FF)								

Los tres bits de menos peso son los señalizadores (“flags”) de ciertas condiciones en las operaciones lógico-aritméticas.

- Z señalizador de cero. Se pone a 1 cuando el resultado es cero.
- C acarreo/llevada del 8º bit. Se pone a 1 automáticamente cuando existe acarreo en el bit de más peso en las instrucciones de suma. También actúa como señalizador de llevada en las instrucciones de resta, pero en este caso la correspondencia es inversa, es decir, si vale 0 hay llevada.
- DC acarreo/llevada en el 4º bit. Funciona igual que el señalizador C pero para el cuarto bit. Es muy útil en las operaciones con números expresados en BCD.

Los señalizadores PD# y TO# son activos por nivel bajo (#) y sirven para indicar la causa que ha provocado la reinicialización del procesador.

- PD#: se pone a 0 al ejecutarse la instrucción sleep (modo reposo). Se pone a 1 automáticamente tras la conexión de la alimentación o bien al ejecutarse la instrucción CLRWDR (refresco del perro guardián).
- TO#: se pone a nivel bajo al desbordarse el perro guardián. Toma el valor 1 tras la conexión de la alimentación o al ejecutarse las instrucciones CLRWDT o SLEEP.

Los señalizadores PD# Y TO# son muy importantes en el proceso de reset porque indican la causa que lo ha originado y permiten actuar en consecuencia. No son escribibles.

Los PIC se resetean al conectar la alimentación (POR: Power-on Reset). También se resetean cuando la tensión de alimentación baja de 4V (BOR: Brown-out Reset), aunque esta función es posible desactivarla poniendo a 0 el bit BODEN presente en la palabra de configuración. Tanto en el reset POR como en el BOR los bits PD# y TO# toman el valor 1, mientras que en los demás casos depende de la causa que ha provocado el reset. El bit IRP se usa concatenado con el bit de más peso del registro FSR para elegir el banco de RAM en el direccionamiento indirecto.

- IRP: Selección del banco en direccionamiento indirecto
 - 0: Banco 0 y 1 (00h...7Fh)
 - 1: Banco 2 y 3 (100h...1FFh)

Finalmente, los tres bits de más peso del registro de estado se emplean para seleccionar el banco de la RAM al que se desea acceder en el direccionamiento directo. Como se vio anteriormente, se utilizan los bits RPI y RP0 de acuerdo con la siguiente codificación.

RPI	RP0	Banco seleccionado
0	0	Banco 0 (00h...7Fh)
0	1	Banco 1 (80h...FFh)
1	0	Banco 2 (100h...17Fh)
1	1	Banco 3 (180h...1FF)

En algunas instrucciones, el registro ESTADO puede actuar como destino, en cuyo caso los bits de los señalizadores Z, C y DC no son escribibles en aquellas instrucciones que

les afecte la operación que realizan, al igual que sucede con PD# y TO#. Por ejemplo, al ejecutar la instrucción CLRF STATUS, el registro de estado pasa a valer 000uuuu. Los bits marcados con “u” no cambian el valor que tenían. Por este motivo se recomienda usar las instrucciones BCF, BSF, SWAPF y MOVF para manejar los bits de este registro, ya que no afectan a los señalizadores Z, C y DC. Los tres bits destinados al direccionamiento de la RAM (IRP, RPI y RP0) toman el valor 0 después del reset.

2.3.2.- EL REGISTRO DE OPCIONES (OPTION).

Sus funciones son:

- Asignar el divisor de frecuencias al Timer0 (TMR0) o al perro guardián.
- Elegir el rango en el que trabaja el divisor de frecuencia.
- Seleccionar el tipo de reloj del TMR0, que puede ser interno o externo, a través de la patita TOCKI. También seledcioa el flanco activo.
- Seleccionar el flanco activo para la interrupción externa por RB0/INT.
- Activar o desactivar las resistencias de pull-up de la puerta B.

Este registro toma el valor 1111 1111 en cualquier tipo de reinicialización que se produzca. En la siguiente figura se muestra el diagrama de distribución de los bits del registro OPTION.

Registro OPTION	7	6	5	4	3	2	1	0
	RBPU#	INTDG	TOCS	TOSE	PSA	PS2	PS1	PS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PS2-PS0=Rango con el que actúa el divisor de frecuencias								
	PS2	PS1	PS0	División del TMR0		División del WDT		
	0	0	0	1:2		1:1		
	0	0	1	1:4		1:2		
	0	1	0	1:8		1:4		
	0	1	1	1:16		1:8		
	1	0	0	1:32		1:16		
	1	0	1	1:64		1:32		
	1	1	0	1:128		1:64		
	1	1	1	1:256		1:128		
PSA=Asignación del divisor de frecuencias								
1=El divisor de frecuencias se le asigna al WDT								
0=El divisor de frecuencias se le asigna al TMR0								
TOSE=Activa el tipo de flanco para la señal en TOCKI								
1=Incremento del TMR0 en cada flanco descendente								
0=Incremento del TMR0 en cada flanco ascendente								
TOCS=Tipo de reloj para el TMR0								
1=Pulsos introducidos a través del TOCKI (contador)								
0=Pulsos de reloj internos Fosc/4 (Temporizador)								
INTDG=Tipo de flanco activo para la interrupción externa RB0/INT								
1=Flanco ascendente								
0=Flanco descendente								

RBP#=#Resistencias Pull-up para la puerta B
 1=Desactivadas
 0=Activadas

2.3.3.- LOS REGISTRO PARA CONTROLAR LAS INTERRUPTIONES.

Los PIC16F877A tienen en total catorce causas que pueden generar una interrupción. Al aceptarse una interrupción se salva el valor del PC en la pila y se carga aquel con el valor 0004h, que es el vector de interrupción. La mayoría de los recursos o periféricos de que dispone el PIC16F877A son capaces de ocasionar una interrupción.

El PIC16F877A tiene cuatro causas principales que pueden originar una interrupción: desbordamiento del TMR0, activación del terminal de interrupción RB0/INT, cambio del estado de una de los 4 terminales de más peso de la puerta B y finalización de la escritura de un byte en la EEPROM. Además de estas las causas de interrupción es este microcontrolador son las siguientes:

- Desbordamiento del Timer1.
- Desbordamiento del Timer2.
- Captura o comparación en el módulo CCPI.
- Captura o comparación en el módulo CCP2.
- Transferencia en la puerta serie síncrona.
- Colisión de bus en la puerta serie síncrona.
- Fin de la transmisión en el USART
- Fin de la recepción en el USART
- Fin de la conversión en el convertor A/D.
- Transferencia en la puerta paralela esclava. (Esta causa de interrupción no está disponible en los PIC16F87X de 28 terminales).

Para compensar el no disponer de más causas de interrupción el PIC16F877A tiene más registros para contener los bits señalizadores y de permiso o prohibición.

2.3.3.1.- REGISTRO DE CONTROL DE INTERRUPTIONES (INTCON).

Se trata de un registro leíble y escribible que, para facilitar su acceso, se halla duplicado en los cuatro bancos, ocupando las direcciones 0Bh, 8Bh, 10Bh y 18Bh, respectivamente. Tiene la misión de controlar las interrupciones provocadas por el TMR0, cambio de estado en las 4 líneas de más peso de la puerta B y activación del terminal RB0/INT. Con respecto a los PIC16F874 sólo cambia el bit 6, que es el PEIE (permiso de interrupción de los periféricos) en lugar del EEIE que tenían los microcontroladores mencionados y cuyo objetivo era permitir la interrupción cuando finalizase la escritura de un byte en la EEPROM. El bit PEIE actúa como una segunda llave parcial de permiso o prohibición de las causas de

TMR1IE=Permiso de interrupción para el TMR1 con su desbordamiento 1=Permite interrupción. 0=Prohíbe interrupción.
TMR2IE=Permiso de interrupción para el TMR2 con su desbordamiento 1: Permite interrupción 0: Prohíbe interrupción
CCP1IE=Permiso de interrupción para el módulo CCP1 cuando se produce una captura o comparación 1=Permite interrupción 0=Prohíbe interrupción
SSPIE=Permiso de interrupción para la puerta serie síncrona 1=Permite interrupción 0=Prohíbe interrupción
TXIE=Permiso de interrupción para el transmisor del USART cuando el buffer se vacía 1=Permite interrupción 0=Prohíbe interrupción
RCIE=Permiso de interrupción para el receptor del USART cuando el buffer se llena 1=Permite interrupción 0=Prohíbe interrupción
ADIE=Permiso de interrupción para el convertor A/D al finalizar la conversión 1=Permite interrupción 0=Prohíbe interrupción
PSPIE=Permiso de interrupción para la puerta paralela esclava al realizar una operación de lectura escritura en los modelos de 40 terminales. En los modelos de 28 terminales este bit es cero 1=Permite interrupción 0=Prohíbe interrupción

2.3.3.3.- REGISTRO DE PERMISO DE INTERRUPTACIONES 2 (PIE2).

Contiene los bits de permiso de interrupción de las tres causas que no figuraban en PIE1. La de fin de escritura en la EEPROM, colisión de bus en el modo SSP y producción de una captura o una comparación en el módulo CCP2. La siguiente tabla presenta la función de estos tres bits. El bit 6 es un bit reservado y su valor es siempre 0. Cuando se leen los bits que no tienen asignada función se obtiene 0.

Registro PIE2	7	6	5	4	3	2	1	0
	---	0	---	EEIE	BCLIE	---	---	CCP2IE
				R/W	R/W			R/W

CCP2IE=Permiso de interrupción en el módulo CCP2
1=Permite interrupción
0=Prohíbe interrupción

<p>BCLIE=Permiso de interrupción por colisión de bus en el SSP cuando dos o más maestros tratan de transferir al mismo tiempo. 1: Permite interrupción 0: Prohíbe interrupción</p>
<p>EEIE=Permiso de interrupción por fin de escritura en la EEPROM de datos. 1=Permite interrupción 0=Prohíbe interrupción</p>

2.3.3.4.- REGISTRO DE LOS SEÑALIZADORES DE INTERRUPTIONES 1 Y 2. (PIR1 Y PIR2).

En correspondencia con los bits de permiso/prohibición de las causas de interrupción recogidas en los registros PIE1 y PIE2, existen otros dos registros, PIR1 y PIR2, cuyos bits actúan de señalizadores del momento en el que se origina la causa que provoca la interrupción, independientemente de si está permitida o prohibida. Ocupan las direcciones 0Ch y 0Dh, respectivamente, figura 15 y 16.

Registro	7	6	5	4	3	2	1	0
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIIF	TMR2IF	TMR1IF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<p>TMR1IF=Flag de desbordamiento de TMR1 1=el registro se desbordo (debe ponerse a cero por software) 0=el registro de TMR1 no se desbordo</p>								
<p>TMR2IF=Flag de interrupción de TMR2 emparejado con PR2 1=TMR2 emparejado con PR2 ocurrió (debe ponerse a cero por software) 0=No ha ocurrido el emparejamiento de TMR2 con PR2</p>								
<p>CCPIIF=Flag de interrupción de CCPI</p> <p>Modo Captura 1=Ocurrió una captura de TMR1 (debe ponerse a cero por software) 0=No ocurrió ninguna captura de TMR1</p> <p>Modo Comparación 1=Se ha realizado una comparación de TMR1. (Con el registro emparejado debe ponerse a cero por software) 0=No se ha realizado comparación</p> <p>Modo PWM Este modo no se utiliza</p>								

<p>SSPIF=Flag de interrupción del Puerto Serie Síncrono (SSP) 1= La condición del SSP ha ocurrido, debe ponerse a cero por software antes de volver del programa de atención a la interrupción. Las condiciones que soporta este bit son: SPI=Ha tenido lugar una Transmisión / Recepción I2C Slave=Ha tenido lugar una Transmisión / Recepción I2C Master= Ha tenido lugar una Transmisión / Recepción La condición de salida iniciada se completó por el módulo de SSP La condición de parada inicio se completó por el módulo de SSP La condición de reinicialización se completo por el módulo SSP Una condición de la salida se ha realizado mientras el módulo de SSP estaba en estado de espera (sistema de Multimaster). 0=no ha ocurrido ninguna condición de interrupción del módulo SSP</p>
<p>TXIF=Flag de transmisión el USART 1= El buffer de datos a transmitir no está lleno 0= El buffer de datos a transmitir está lleno</p>
<p>RCIF=Flag de recepción por el USART 1=El buffer de datos recibidos por el USART está lleno 0=El buffer de datos recibidos por el USART no está lleno</p>
<p>ADIF=Flag de final de conversión del convertidor A/D 1=El convertidor A/D ha finalizado la conversión 0=El convertidor A/D no ha finalizado la conversión</p>
<p>PSPIF=Flag de Lectura/Escritura por el Puerto Paralelo esclavo 1=Concedido el permiso de interrupción para la puerta paralela esclava al realizar una operación de Lectura/Escritura. En los modelos de 40 terminales 0=No está permitida la interrupción PSPIF está reservado para los dispositivos de 28 terminales siendo siempre cero</p>

Registro	7	6	5	4	3	2	1	0
PIR2	---	0	---	EEIF	BCLIF	---	---	CCP2IF
				R/W	R/W			R/W

<p>CCP2IF=Flag de interrupción de CCP2 Modo Captura 1=Ha ocurrido una captura del registro TMR1 (debe ponerse a cero por software) 0=No se ha producido captura Modo Comparación 1=Se ha producido una comparación de TMR1 con su pareja (debe ponerse a cero por software). 0=No se ha producido comparación Modo PWM No se utiliza</p>
<p>BCLIF=Flag que indica la colisión en el bus 1=Se ha producido una colisión en el bus SSP, cuando se configura en el modo I2C Master. 0=No se ha producido colisión en el bus</p>
<p>EEIF=Flag que indica si se ha producido escritura en la EEPROM 1=Se ha completado la escritura en la EEPROM (Se pone a cero por software) 0=No se ha finalizado la escritura o no se ha comenzado</p>

2.3.3.5.- LECTURA Y ESCRITURA DE LAS MEMORIAS EEPROM Y FLASH.

En los PIC16F87X se pueden leer y escribir la memoria de datos EEPROM y la memoria de código FLASH. Esto significa que un programa dinámicamente puede generar información que se puede grabar en la FLASH directamente, sin necesidad de grabador externo. La propia aplicación puede reprogramar según las condiciones externas y es posible ampliar el área de la memoria de datos no volátil EEPROM con posiciones libres de la memoria de código FLASH.

Para manejar la memoria EEPROM de 64 bytes de los anteriores modelos de microcontroladores PIC bastaban dos registros para proporcionar la dirección a acceder y para guardar el dato de 8 bits que se leía o se iba a grabar. Como sólo existían 64 posiciones en la EEPROM bastaba con un registro de 8 bits para contener la dirección, el EEADR. El dato leído o a escribir, de tamaño byte, se colocaba en el registro EEDATA. Como ya se dijo anteriormente, en los PIC16F87X también se puede leer y escribir la memoria FLASH y ésta puede alcanzar un tamaño de 8 K palabras de 14 bits cada una. Por lo tanto, no es suficiente con un solo registro para la dirección (que alcanza los 13 bits) y lo mismo sucede para el dato, que tiene una longitud de 14 bits. Para cubrir esta necesidad el registro EEADR se concatena con el EEADRH que contiene los 5 bits de más peso de la dirección. Por otra parte, el registro EEDATAH se concatena con el EEDATA y contiene los 6 bits de más peso de la palabra leída o a escribir en la FLASH. Estos dos nuevos registros no se usan en las operaciones que afectan a la EEPROM.

Para controlar la operación de lectura/escritura de las memorias EEPROM y FLASH hay dos registros denominados EECON1 y EECON2. El EECON1 ocupa la dirección 18Ch, mientras que el EECON2, no está implementado físicamente y sólo se utiliza en la delicada operación de escritura, que tiene la elevada duración de 2 milisegundos aproximadamente. Antes de iniciar la escritura de una palabra se escribe en EECON2 el dato 55h y luego el AAh.

Registro	7	6	5	4	3	2	1	0
EECON1	EEPGD	---	---	---	WRERR	WREN	WR	RD
	R/W	0	0	0	R/W	R/W	R	R

RD=bit de inicio de lectura
 1=Cuando se le pone a 1 se inicia un ciclo de lectura. El bit RD se pone a cero por hardware.
 0=no ha comenzado el ciclo de lectura de la memoria no volátil.

WR=bit de inicio de escritura
 1=Cuando se le pone a 1 comienza el ciclo de escritura de la memoria no volátil.
 (El bit se pone de nuevo a cero por hardware cuando la escritura se completa)
 0=Toma este valor cuando completa el ciclo de escritura de la memoria no volátil

<p>WREN=bit de habilitación de escritura 1=Permite inicializar el ciclo de escritura 0=Inhibe la escritura</p>
<p>WRERR=Flag de error de escritura en la EEPROM 1=El proceso de lectura se ha producido prematuramente, (se ha producido un reset por MCLR o un WDT durante el proceso) 0=Se ha producido el proceso de escritura con éxito</p>
<p>EEPGD=selección del bit para la memoria de programa o de datos 1=Acceso a la memoria de programa FLASH 0=Acceso a la memoria de datos EEPROM (No se puede cambiar mientras la lectura o escritura está en proceso)</p>

Para evitar escrituras indeseadas en la EEPROM motivadas por espurios en la inicialización del microcontrolador se controla el bit WREN, que prohíbe cualquier operación de escritura mientras duran los 72 milisegundos que temporiza el Timer de Power-up. Para realizar la misma protección en la memoria FLASH se debe poner a 0 el bit WRT de la palabra de configuración, que sólo puede escribirse desde un grabador externo.

Dependiendo del valor del bit WRT y de los bits de protección de código CPI y CP0, ubicados en la palabra de configuración, se consiguen diversas alternativas de protección contra lectura y escritura de la FLASH, como puede observarse en la siguiente tabla.

CONFIGURACIÓN DE BITS			POSICIONES DE FLASH	LECTURA INTERNA	ESCRITURA INTERNA	LECTURA ICSP	ESCRITURA ICSP
CPI	CP0	WRT					
0	0	X	Toda la memoria de programa	Sí	No	No	No
0	1	0	Áreas no protegidas	Sí	No	Sí	No
0	1	0	Área protegida	Sí	No	No	No
0	1	1	Área no protegida	Sí	SI	Sí	No
0	1	1	Área protegida	Sí	No	No	No
1	0	0	Áreas no protegidas	Sí	No	Sí	No
1	0	0	Área protegida	Sí	No	No	No
1	0	1	Áreas protegidas	Sí	SI	Sí	No
1	0	1	Áreas protegidas	Sí	No	No	No
1	1	0	Toda la memoria de programa	Sí	No	Sí	Sí

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000000x	0000000u
10Dh	EEADR	Registro de direccionamiento de EEPROM, byte bajo								xxxxxxxx	uuuuuuuu
10Fh	EEADRH	---	---	---	Parte alta del direccionamiento de la EEPROM				xxxxxxxx	uuuuuuuu	
10Ch	EEDATA	Registro de dato de la EEPROM, byte bajo								xxxxxxxx	uuuuuuuu
10Eh	EEDATH	---	---	---	Parte alta del dato de la EEPROM				xxxxxxxx	uuuuuuuu	
18Ch	EECON1	EEPG D	---	---	---	WRERR	WREN	WR	RD	x---xxxx	x---u000
18Dh	EECON2	Registro de control EEPROM (registro físico no implementado)									
8Dh	PIE2	---	0	---	EEIE	BCLIE	---	---	CCP2I E	-r-00---0	-r-0---0
0Dh	PIR2	---	0	---	EEIF	BCLIF	---	---	CCP2I F	-r-00---0	-r-0---0

2.3.3.6.- REGISTRO PCON.

El registro PCON (Power Control) control de alimentación, contiene los flags que permiten diferenciar entre un Power-on Reset (POR), un Brown-out Reset (BOR), un reset por Watchdog (WDT) y un reset externo por MCLR.

Registro PCON	7	6	5	4	3	2	1	0
	---	---	---	---	---	---	#POR	#BOR
	0	0	0	0	0	0	R/W	R/W

#BOR=bit de estado de Brown-out Reset
1=no se ha producido ningún Brown-out Reset
0=se ha producido un Brown-out Reset (debe reestablecerse por software)

#POR=bit de estado Power-on Reset
1=no se ha producido un Power-on Reset
0=se ha producido un Power-on Reset (debe reestablecerse por software)

Los demás bits no están implementados: y se leen como 0

2.4.- LAS PUERTAS DE E/S Y RECURSOS ESPECIALES DEL PROCESADOR.

Los microcontroladores PIC16F877 disponen de cinco puertas de E/S (A, B, C, D, y E). Todas las líneas de estas puertas son multifuncionales, es decir, realizan diversas funciones según estén programadas. Sin embargo, todas ellas tienen la capacidad de trabajar como líneas de E/S digitales. Cada línea de E/S puede configurarse independientemente como entrada o como salida, según se ponga a 1 o a 0, respectivamente, el bit asociado del registro de configuración de cada puerta (TRISA, TRISB, TRISC, TRISD y TRISE).

2.4.1.- PUERTA A.

Sólo dispone de 6 líneas, denominadas RA0-RA5. Son bidireccionales y su sentido queda configurado según la programación de los bits del registro TRISA. Si el bit 0 del registro TRISA se pone a 1, la línea 0 (RA0) de la puerta A funciona como entrada. Si se pone a 0 funciona como salida y el contenido de la báscula de salida se aplica al terminal correspondiente.

Al leer el registro PORTA de la puerta A se lee el estado de los terminales, que es el que se halla escrito en la báscula de datos. La escritura entraña una operación de “lectura-modificación-escritura”, o sea, se leen los terminales, luego se modifica su valor y finalmente se escribe en la báscula de datos.

Los terminales RA0/AN0, RA1/AN1 y RA2/AN2, además de líneas de E/S digitales también pueden actuar como los canales 0, 1 y 2 por los que se puede aplicar una señal analógica al convertor A/D. El terminal RA3/AN3/Vref+, también puede actuar como entrada de la tensión de referencia para los periféricos que la precisan. El terminal RA4/T0CKI actúa como E/S digital y como entrada de la señal de reloj para el Timer 0. Por último, el terminal RA5/AN4/SS# tiene multiplexadas tres funciones: E/S digital, canal 4 para el convertor A/D y selección del modo esclavo cuando se trabaja con la comunicación serie síncrona.

Cuando se produce un reset al conectar la tensión de alimentación (POR: Power-on Reset) todos los terminales de la puerta A quedan configuradas como canales de entrada para el convertor A/D y son leídos como cero.

Para seleccionar si las líneas de la puerta A van a trabajar como E/S digitales o como canales de entrada para el convertor A/D, hay que escribir el valor adecuado sobre el registro ADCON1, que se estudiará con mayor profundidad al describir el convertor. De momento es suficiente con saber que si se carga en dicho registro el valor 011x en sus 4 bits de menos peso, todas las líneas de la puerta A funcionan como E/S digitales. La siguiente tabla recoge los registros que manejan la puerta A.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
05h	PORTA	-	-	RA5	RA4	RA3	RA2	RA1	RA0	--0x0000	--0u0000
85h	TRISA	-	-	Registro de configuración de la puerta A						--111111	--111111
9Fh	ADCON1	ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0	--0-0000	--0-0000

2.4.2.- PUERTA B.

Dispone de ocho líneas bidireccionales cuya función se elige mediante la programación del registro TRISB, igual que sucedía en la puerta A con TRISA.

Los PIC16F87X pueden ser programados con tensión alta y con tensión baja, siendo en este último caso la misma tensión de alimentación del microcontrolador VDD=5V la que se emplea en la grabación del programa. En la programación con tensión alta ésta se introduce por el terminal MCLR#/VPP y está comprendida entre 12 y 14 V. En la programación con

tensión baja se aplican 5V por el terminal MCLR#/VPP y el terminal RB3/PGM se conecta a nivel alto. Este último terminal es una E/S digital normal en la programación con tensión alta. Como la programación siempre se realiza síncronamente en serie, se introducen los pulsos de reloj por el terminal RB6/PGC y por la RB7/PGD, los bits de datos en serie.

Todos los terminales de la puerta B disponen de una resistencia interna de pull-up al positivo de la alimentación, que queda conectado cuando el bit RBPU#, que es el bit 7 del registro OPTION, tiene valor 0. La resistencia de pull-up es un transistor CMOS tipo P que se conecta automáticamente siempre que la línea esté configurada como salida. Cuando se produce un reset por conexión de la alimentación (POR) se desconectan todas las resistencias pull-up.

Las líneas RB(7...4) pueden programarse para generar una interrupción cuando una de ellas cambia de estado. Se deben configurar como entradas y el valor que se introduce por ellas se compara con el anterior para generar una interrupción en el caso de que no coincidan, siempre que lo autorice el bit de permiso. En cualquier caso el señalizador RBIF, que es el bit 0 del registro INTCON, se pondrá a 1. Esta interrupción es muy utilizada en el manejo de teclados matriciales.

El terminal RB0/INT también puede programarse como petición de interrupción externa si se autoriza con el correspondiente bit de permiso, que también está ubicado en INTCON.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
06h,106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxxxxxx	uuuuuuuu
86h,186h	TRISB	Registro de configuración de la puerta B									
81h,181h	OPTION	RBPU#	INTDGL	T0CS	T0SE	PSA	PS2	PS1	PS0		
0Bh,8Bh 10Bh,18B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000000x	0000000u

2.4.3.- PUERTA C.

Consta de ocho líneas bidireccionales cuyo sentido se configura mediante el registro TRISC. Todos los terminales de esta puerta tienen multiplexadas diferentes funciones que a continuación se exponen.

RC0/TIOSO/TICKI: Esta línea puede actuar como E/S digital, como salida del oscilador del Timer1 o como entrada de reloj o pulsos para el Timer1.

RC1/TIOSI/CCP2: E/S digital, entrada al oscilador del Timer1, entrada al módulo Captura2, salida comparador 2, salida de PWM2.

RC2/CCP1: E/S digital, entrada captura 1, salida comparador 1, salida de PWM1.

RC3/SCK/SCL: E/S digital, señal de reloj en modo SPI, señal de reloj en modo I2C.

RC4/SDI/SDA: E/S digital, entrada de datos en modo SPI, línea de datos en modo I2C.

RC5/SDO: E/S digital, salida de datos en modo SPI.

RC6/TX/CK: E/S digital, línea de transmisión en USART, señal de reloj síncrona en transmisión serie.

RC7/RX/DT: E/S digital, línea de recepción del USART, línea de datos en transmisión serie síncrona.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxxxxx x	uuuuuuu u
87h	TRISC	Registro de configuración de la puerta D								 	

2.4.4.- PUERTA D.

Esta puerta de ocho líneas bidireccionales. Ocupa la dirección 08h y su registro de configuración (TRISD) ocupa la dirección 88h. Todos los terminales disponen en su entrada de un Disparador Schmitt.

Además de usarse como líneas de E/S digitales normales, implementan una puerta paralela esclava de 8 líneas (PSP) que sirve para permitir la comunicación en paralelo con otros elementos del sistema.

Los terminales se denominan RD0/PSP0-RD7/PSP7 y para que funcionen como puerta de comunicación esclava en paralelo es preciso poner el bit PSPMODE=1. Este bit es el 4 del registro TRISE, que se comentará en la puerta E.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxxxxx x	uuuuuuu u
88h	TRISD	Registro de configuración de la puerta D								 	
89h	TRISE	IBF	OBF	IBOV	PSPMODE	---	Registro de configuración de la puerta E			0000-111	0000-111

Estructura de los registros que manejan la puerta D.

2.4.5.- PUERTA E.

Ocupa la dirección 09h dispone de tres terminales multifunción, que se configuran como entrada o salida según el valor de los tres bits de menos peso del registro TRISE, que está ubicado en la dirección 89h.

RE0/RD#/AN5: E/S digital, señal de lectura en el modo de puerta paralela esclava (PSP), canal 5 del convertor A/D.

RE1/WR#/AN6: E/S digital, señal de escritura en modo PSP, canal 6 del convertor A/D.

RE2/CS#/AN7: E/S digital, selección de chip en el modo PSP, canal 7 del conversor A/D.

La PSP actúa como un puerto de comunicación en paralelo de 8 líneas y para su activación hay que poner el bit PSPMODE a 1. Dicho bit es el 4 del registro TRISE, como ya se dijo anteriormente.

Además de las ocho líneas de transferencia de datos, se precisan tres señales de control, que determinan si la operación es de lectura, de escritura o de permiso de funcionamiento (RD#, WR# y CS#). Estas tres líneas de control están implementadas en la puerta E.

Registro	7	6	5	4	3	2	1	0
TRISE	IBF	OBF	IBOV	PSPMODE	---	BIT2	BIT1	BIT0
	R/W	R/W	R/W	R/W	0	R/W	R/W	R/W

BIT2-BIT0=bits que controlan la puerta E 1=esta línea trabaja en entrada 0=esta línea trabaja en salida
PSPMODE=modo de trabajo de la puerta paralela esclava 1=la puerta D trabaja como puerta paralela esclava 0=la puerta D trabaja como líneas de entrada/salida digitales
IBOV=la entrada del buffer se ha desbordado 1=a ocurrido cuando la palabra anterior no se había leído todavía 0=ningún desbordamiento ha ocurrido
OBF= estado de los bits de salida del buffer lleno, existe un dato que hay que enviar 1=la palabra a enviar todavía está en el registro 0=la palabra se ha enviado
IBF=estado de los bits de entrada al buffer lleno, existe un dato que hay que leer 1=la palabra se ha recibido y está esperando a ser leída por la CPU 0=ninguna palabra se ha recibido

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
09h	PORTE	---	---	---	---	---	RE2	RE1	RE0	----xxx	----uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	---	Registro de configuración de la puerta E			0000-1111	0000-1111
9Fh	ADCON1	ADFM	---	---	---	PCFG3	PCFG2	PCFG1	PCFG0	--0-0000	--0-0000

2.4.6.- PUERTA PARALELA ESCLAVA.

Esta puerta es usada para la comunicación en paralelo. Se utilizan las líneas de la puerta D como líneas de comunicación y las de la puerta E, como señales de control. En la siguiente tabla se representan todos los registros que intervienen en el control de esta puerta.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
08h	PORT D	Registro de entrada/salida para la puerta paralela esclava								xxxxxxx x	uuuuuuu u
09h	PORTE	---	---	---	---	---	RE2	RE1	RE0	-----xxx	-----uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	---	Registro de configuración de la puerta E			0000-111	0000-111
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIIF	TMR2IF	TMR1IF	0000000 0	0000000 0
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPIIE	TMR2IE	TMR1IE	0000000 0	0000000 0
9Fh	ADCON1	ADFM	---	---	---	PCFG3	PCFG2	PCFG1	PCFG0	--0-0000	--0-0000

2.4.7.- PALABRA DE CONFIGURACIÓN.

Es una posición reservada de la memoria de programa FLASH, que ocupa la dirección 2007h y que sólo es accesible durante la programación del PIC. El valor de sus bits determina algunas características fundamentales.

Palabra de configuración	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPI	CP0	DEBUG		WRT	CPD	LVP	BODEN	CPI	CP0	PWRTE#	WDTE	FOSCI	FOSC0
	W	W	W	W	W	W	W	W	W	W	W	W	W	W
FOSCI, FOSC0=Selecciona el tipo de oscilador														
	FOSCI	FOSC0	Tipo											
	0	0	LP (Baja Potencia. De 35 a 200 kHz)											
	0	1	XT (Estándar. De 100 kHz a 4 MHz)											
	1	0	HS (Alta velocidad. Más de 4 MHz)											
	1	1	RC (Resistencia-Condensador)											
WDTE=Bit de permiso del Timer del perro guardián														
	1=WDT activado													
	0=WDT desactivado													
PWRTE#=Bit de permiso para el Timer de conexión de alimentación														
	1=PWRT desactivado													
	0=PWRT activado													

BODEN=Bit de permiso para el reset por caída de tensión 1=BOR activada. 0=BOR desactivada.		
LVP=Bit de permiso para programación en baja tensión 1=RB3/PGM tiene permitida la grabación en baja tensión 0=RB3/PGM funciona como E/S digital. La programación se realiza en baja tensión		
CPD=Código de protección de la memoria EEPROM de datos 1=No hay protección en la EEPROM 0=Protección del código en la EEPROM		
WRT=Permiso de escritura en la memoria FLASH 1=Se puede escribir en la parte no protegida de la memoria FLASH 0=Prohibición de escritura.		
DEBUG=Modo depurador en circuito 1=Desactivado. RB7-RB6 actúan como líneas de E/S. 0=Activado. RB7-RB6 actúan en modo depurador. La depuración se puede hacer desde el MPLAB.		
CPI-CP0=Código de protección de la memoria de programa Están repetidos en los bits 13-12 y 5-4. Si los bits del código de protección no se programan, las posiciones de la memoria de código pueden ser leídas para verificación.		
CPI	CP0	Protección desde.....hasta
0	0	0000h 1FFFh
0	1	1000h 1FFFh
1	0	1F00h 1FFFh
1	1	No hay código protegido en la memoria FLASH

2.4.8.- PALABRA DE IDENTIFICACIÓN.

Se trata de cuatro palabras de la memoria de programa que se encuentran entre la dirección 2000h y la 2003h y están reservadas para que el usuario las pueda emplear en funciones de comprobaciones o “cheksums” (códigos de identificación, números de serie, fecha, modelo, lote, números secuenciales o aleatorios, etc). Estas cuatro posiciones sólo son accesibles en lectura y escritura durante la operación de programación/verificación. Sólo se deben emplear los 4 bits de menos peso de cada una de las palabras de identificación.

2.4.9.- REINICIALIZACIÓN O RESET.

Los PIC16F87X disponen de diversas maneras de reinicializarse:

- Reset por conexión de la alimentación (POR: Power-on Reset). El valor de la tensión de alimentación VDD sube entre 1,2 y 1,7V.
- Activación del terminal MCLR# (nivel bajo en dicho terminal durante una operación normal).
- Activación del terminal MCLR# estando el PIC trabajando en modo de reposo o sleep.
- Reset provocado por el desbordamiento del perro guardián en una operación normal.

- Reset provocado por el desbordamiento del perro guardián durante el estado de reposo.
- Reset provocado por una caída de tensión (BOR: Brown-on Reset). VDD baja entre 3,8 y 4,2 V.

Los bits TO# y PD# del registro de ESTADO toman un valor determinado en cada tipo de reset. A su vez, los bits 0 y 1 del registro PCON, llamados BOR# y POR#, respectivamente, sirven para especificar las causas de un reset. En la siguiente tabla se muestran estos bits y el tipo de reset asociado.

Las x significan que pueden tener valor 0 ó 1 indistintamente, ya que no se conoce su valor; las u, que su valor no varía con respecto al que tenía antes del reset.

POR#	BOR#	TO#	PD#	Tipo de reset
0	x	1	1	Conexión de alimentación (POR)
0	x	0	x	ilegal
0	x	x	0	ilegal
1	0	1	1	Por caída de tensión (BOR)
1	1	0	1	Por WDT (operación normal)
1	1	0	0	Por WDT (modo sleep)
1	1	u	u	Activación normal (MCLR#)
1	1	1	0	MCLR# en SLEEP o interrupción para despertar de SLEEP

Tras un reset el contador de programa queda cargado con el valor 0000h en todos los casos, menos cuando se produce por el desbordamiento del WDT o cuando se despierta del modo SLEEP por una interrupción. En estos casos el PC se carga con el valor PC+1.

En la siguiente tabla se indican los valores que toman los registros específicos del microcontrolador tras producirse los diversos tipos de reset.

REGISTRO	POR Y BOR	MCLR# Y WDT	DESPERTAR DE SLEEP
W	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	N/A	N/A	N/A
TMR0	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	0000h	0000h	PC+1
STATUS	0001 lxxx	000q quuu	uuuq quuu
FSR	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	--0x 0000	--0u 0000	--uu uuuu
PORTB	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	---0 0000	---0 0000	---u uuuu
INTCON	0000 000x	0000 000u	uuuu uuuu
PIR1	r000 0000	r000 0000	ruuu uuuu
PIR2	-r-0 0--0	-r-0 0--0	-r-u u--u
TMR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu

TICON	--00 0000	--uu uuuu	--uu uuuu
TMR2	0000 0000	0000 0000	uuuu uuuu
T2CON	-000 0000	-000 0000	-uuu uuuu
SSPBUF	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	0000 0000	0000 0000	uuuu uuuu
CCPR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	--00 0000	--00 0000	--uu uuuu
RCSTA	0000 000x	0000 000x	uuuu uuuu
TXREG	0000 0000	0000 0000	uuuu uuuu
RCREG	0000 0000	0000 0000	uuuu uuuu
CCPR2L	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	0000 0000	0000 0000	uuuu uuuu
ADRESH	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	1111 1111	1111 1111	uuuu uuuu
TRISA	--11 1111	--11 1111	--uu uuuu
TRISB	1111 1111	1111 1111	uuuu uuuu
TRISC	1111 1111	1111 1111	uuuu uuuu
PIE1	r000 0000	r000 0000	ruuu uuuu
PIE2	-r-0 0--0	-r-0 0--0	-r-u u--u
PCON	---- --qq	---- --uu	---- --uu
PR2	1111 1111	1111 1111	1111 1111
SSPADD	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	--00 0000	--00 0000	--uu uuuu
TXSTA	0000 -010	0000 -010	uuuu -uuu
SPBRG	0000 0000	0000 0000	uuuu uuuu
ADRESL	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	--0- 0000	--0- 0000	--u- uuuu
EEDATA	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATH	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADRH	xxxx xxxx	uuuu uuuu	uuuu uuuu
EECON1	x--- x000	u--- u000	u--- uuuu
EECON2	---- ----	---- ----	---- ----

2.4.10.- PERRO GUARDIÁN (WDT, WATCHDOG TIMER).

El WDT del PIC16F87X es un contador interno que origina un reset cuando se desborda. Su control de tiempos es independiente del TMR0 y está basado en una simple red R-C. Su actuación es opcional y puede bloquearse para que no funcione programando el bit WDTE de la palabra de configuración.

Para evitar que se desborde el perro guardián hay que refrescarlo previamente. Este refresco consiste en ponerlo a cero mediante las instrucciones CLRWDT y SLEEP. El programador debe analizar las instrucciones de la tarea y situar alguna de esas dos en sitios estratégicos por los que pase el flujo de control antes que transcurra el tiempo asignado al WDT. De esta manera, si el programa se “cuelga” no se refresca el perro guardián y se produce la reinicialización del sistema.

La instrucción CLRWDT borra al WDT y reinicia su cuenta. La instrucción SLEEP, además de borrar WDT, detiene el sistema y lo pone en un estado de “reposo” o “de bajo consumo”. Si no se desactiva el perro guardián al entrar en el modo de reposo provocará un reset al completar su contaje y sacará al microcontrolador del modo de bajo consumo. Para desactivar al perro guardián hay que escribir un 0 en el bit 2 (WDTE) de la palabra de configuración.

2.4.11.- MODO DE REPOSO O DE BAJO CONSUMO.

Este modo especial de funcionamiento del microcontrolador se introduce cuando se ejecuta la instrucción SLEEP. Este funcionamiento se caracteriza por su bajo consumo, parece que el PIC se ha “congelado”. Las líneas de E/S digitales que se utilizaban mantienen su estado, mientras que las que no se empleaban reducen al mínimo su consumo, se detienen los temporizadores y tampoco funciona el convertor A/D.

Al entrar en el modo de reposo se borra el WDT si estaba funcionando, pero sigue trabajando.

Para salir de este estado (despertar) y pasar a ejecutar la instrucción direccionada por PC+1 existen varias causas que se numeran a continuación:

- Activación externa del terminal MCLR#.
- Desbordamiento del WDT que sigue trabajando en reposo.
- Generación de interrupción por activación del terminal RB0/INT o por cambio de estado en los 4 terminales de más peso de la puerta B.
- Interrupción originada por alguno de los siguientes periféricos:
 - Interrupción del Timer 1.
 - Interrupción del módulo CCP en modo captura.
 - Disparo especial del Timer 1 funcionando en el modo asíncrono con reloj externo.
 - Interrupción en el módulo de comunicación SSP (Start/Stop).
 - Transmisión o recepción del MSSP en modo esclavo (SPI/I2C).
 - Transmisión o recepción del USART.
- Fin de la conversión en el convertor A/D.
- Fin de la operación de escritura sobre la EEPROM.

2.4.12.- PROGRAMACIÓN DE LOS PIC16F87X.

La posibilidad de programar estos PIC en serie permite grabar en la memoria de código el programa de trabajo, estando el microcontrolador colocado sobre el circuito o producto de aplicación final. Esta característica permite a los fabricantes construir y montar completamente la tarjeta de circuito impreso y dejar pendiente la grabación del programa en el PIC hasta el momento de la venta, pudiendo incluir el firmware más reciente que se disponga en cada momento.

La programación en serie típica, que se realiza con una tensión alta de 12 a 14 V aplicado por el terminal MCLR#/VPP, requiere el uso de 5 terminales del PIC:

VDD=5V.

GND o tierra.

VPP=12 ó 14 V que se introducen por el terminal MCLR#/VPP.

RB6: recibe los pulsos de reloj.

RB7: línea de datos con los bits en serie.

La programación con tensión baja que no requiere la tensión de 12 a 14 V constituye una gran aportación en los PIC16F87X. Para grabar en este modo hay que poner el bit LVP=1 (que reside en la palabra de configuración) y el terminal RB3/PGM se debe conectar a nivel alto. Por el terminal MCLR#/VPP se debe aplicar la tensión VDD de 5V mientras dure la operación de grabado. Cuando el microcontrolador no funciona en este modo de programación se puede usar el terminal RB3 como una línea de E/S digital.

2.5.- LOS TEMPORIZADORES.

Los PIC16F87X disponen de un potente conjunto de temporizadores para manejar eficientemente todas las operaciones que involucran tiempo y contaje. Dichos temporizadores son tres y se denominan técnicamente TMR0, TMR1 y TMR2.

Las funciones más representativas del TMR0 son:

- TMR0 es un contador/temporizador de 8 bits.
- Es leíble y escribible.
- Selección de reloj interno o externo.
- Selección del flanco en el reloj externo.
- Predivisor de la frecuencia de reloj programable.
- Generación de interrupción opcional en el desbordamiento.

Anteriormente se presentó el registro OPTION encargado del control del TMR0. Los nuevos temporizadores que incluyen los PIC16F87X son más potentes y flexibles que el sencillo TMR0. Así, el TMR1 se caracteriza por:

- TMR1 es un contador/temporizador de 16 bits.
- Es leíble y escribible.
- Selección de reloj interno o externo.

- Interrupción opcional por desbordamiento de FFFFh a 0000h.
- Posible reinicialización desde los módulos CCP.

El TMR2 tiene las siguientes características fundamentales.

- El TMR2 es un temporizador de 8 bits.
- Dispone de un registro de período de 8 bits (PR2).
- Leíble y escribible.
- Predivisor de frecuencia programable.
- Postdivisor de frecuencia programable.
- Interrupción opcional al coincidir TMR2 y PR2.
- Posibilidad de generar pulsos al módulo SSP.

2.5.1.- ESTRUCTURA INTERNA Y FUNCIONAMIENTO DEL TMR0.

El TMR0 se comporta como un registro de propósito especial (SFR) ubicado en la dirección 1 del banco 0 de la memoria de datos. En igual dirección, pero en el banco 1, se halla el registro OPTION.

El TMR0 puede ser leído y escrito en cualquier momento al estar conectado al bus de datos y funciona como un contador ascendente de ocho bits. Cuando funciona como temporizador conviene cargarle con el valor de los pulsos que se quiere temporizar, pero expresados en complemento a 2. De esta manera, al llegar el número de pulsos deseado se desborda y al pasar por 00H se activa el señalizador TOIF (bit 2 del INTCON) y/o se produce una interrupción.

Para calcular los tiempos a controlar con TMR0 se utiliza la siguiente fórmula práctica:

$$\text{Temporización} = 4 \cdot TOSC \cdot (256 - \text{Valor cargado en TMR0}) \cdot (\text{Rango del divisor})$$

$$TOSC = \text{Periodo de la frecuencia del reloj de funcionamiento.}$$

En cualquier momento se puede leer el valor que contiene TMR0, sin detener su conteo. La instrucción adecuada al caso es MOVF TMR0,W.

Los registros que intervienen en la programación del TMR0 son los de la siguiente tabla:

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
01h,101h	TMR0	Registro de carga del byte a contar								xxxxxxx x	uuuuuuu u
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000000 x	0000000 u
81h,181h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111111 1	1111111 1

2.5.2.- ESTRUCTURA INTERNA Y FUNCIONAMIENTO DEL TMR1.

El TMR1 es el único temporizador/contador ascendente con un tamaño de 16 bits, lo que requiere el uso de dos registros concatenados de 8 bits: TMR1H y TMR1L, que son los encargados de guardar el valor del contaje en cada momento. Dicho valor evoluciona desde 0000h hasta FFFFh, instante en el que se activa el señalizador TMR1IF y se regresa al valor inicial 0000h. Si se desea se puede provocar también una petición de interrupción. El valor contenido en TMR1H y TMR1L puede ser leído o escrito y los pulsos de reloj que originan el contaje ascendente pueden provenir del exterior o de la frecuencia de funcionamiento del microcontrolador ($F_{osc}/4$).

El TMR1 es capaz de funcionar de tres formas:

- Como temporizador.
- Como contador síncrono.
- Como contador asíncrono.

En el modo temporizador el valor concatenado TMR1H y TMR1L se incrementa con cada ciclo de instrucción ($F_{osc}/4$). En el modo contador, el incremento se puede producir con los flancos ascendentes de un reloj externo, cuya entrada se aplica a las líneas RC0 y RC1 de la puerta C, o por pulsos aplicados en la línea RC0.

2.5.2.1.- REGISTRO DE CONTROL DEL TMR1 (T1CON).

El funcionamiento del TMR1 está gobernado por el valor con el que se programan los bits del registro T1CON, que ocupa la dirección 10h de la memoria RAM y cuya denominación y distribución se presentan a continuación. El bit TMR10N gobierna el permiso o la prohibición de funcionamiento del timer1. En caso de poner un 0 en dicho bit el TMR1 no funciona. El bit TMR1CS selecciona la fuente de los pulsos de contaje. Si vale 0 elige el reloj interno ($F_{osc}/4$) y si vale 1 el reloj externo que se aplica por los terminales RC0 y RC1. Cuando los pulsos proceden de un reloj externo, es preciso que el bit T1OSCEN tenga valor 1, en cuyo caso los terminales RC0/T1OSO/TICK1 y RC1/T1OSI/CCP2 actúan como entradas del oscilador externo.

Si T1OSCEN vale 0 los pulsos vendrán a través de RC0/T1OSO/TICK1. En ambos casos, el TMR1 funciona como contador de eventos externos y los bits 1 y 0 (utilizados) del registro TRISC carecen de significado, ya que uno o ambos terminales RC1 y RC0 no pueden actuar al mismo tiempo como entrada de pulsos y como líneas de E/S. Entre los terminales RC1 y RC0 se puede poner un cristal de hasta una frecuencia de 200 KHz. En estas condiciones, el oscilador puede seguir funcionando aunque se fuerce el estado de bajo consumo o SLEEP.

El predivisor de frecuencia (“preescaler”) es un divisor de la frecuencia de los pulsos que se aplican al TMR1 por 1, 2, 4 u 8. El rango de división lo eligen los bits TICKPS1 y TICKPS0 según la siguiente tabla.

T2CKPS1	T2CKPS0	Rango del predivisor
0	0	1:1
0	1	1:2

1	0	1:4
1	1	1:8

El bit TISYNC# determina la posible sincronización o no de los pulsos del reloj externo con los del reloj interno, según valga 0 ó 1 respectivamente. El TMR1 puede generar una petición de interrupción cuando se produce el sobrepasamiento del contaje, es decir, cuando se pasa desde FFFFh a 0000h. En esta situación se pone automáticamente a 1 el flag TMR1IF (bit 0 del registro específico PIR1) que está ubicado en la dirección 0Ch de la RAM. El permiso o prohibición de la producción de interrupción del TMR1 está controlada por el bit TMR1IE, que ocupa la posición de menos peso del registro específico PIE1 (8Ch).

Registro TICON	7	6	5	4	3	2	1	0
	---	---	TICKPS1	TICKPS0	TIOSCEN	TISYNC#	TMR1CS	TMR1ON
	0	0	R/W	R/W	R/W	R/W	R/W	R/W
TMR1ON=Permiso de funcionamiento del Timer1 1=Funciona 0=No funciona								
TMR1CS=Selección de la fuente de los pulsos de contaje 1=Si vale 1 selecciona el reloj externo que se aplica por los terminales RC0 y RC1 0=Selecciona el reloj interno								
TISYNC#=Sincronización de los pulsos de reloj 1=No hay sincronización de los pulsos de reloj externo con los del reloj interno 0=Si hay sincronización								
TIOSCEN=Procedencia de los pulsos cuando TMR1 funciona como contador de eventos externos 1=Los pulsos proceden de un reloj externo, los terminales RC0/TIOSO/TICK1 y RC1/TIOSI/CCP2 actúan como entradas del oscilador externo 0=Los pulsos vendrán a través de RC0/TIOSO/TICK1								
T2CKPS1-T2CKPS0=Rango del predivisor de frecuencias								
	T2CKPS1	T2CKPS0	Rango del predivisor					
	0	0	1:1					
	0	1	1:2					
	1	0	1:4					
	1	1	1:8					

En la siguiente tabla se resumen los registros principales que controlan el comportamiento del TMR1 y la distribución de los bits.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	0000000 x	0000000 u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000000 0	0000000 0
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000000 0	0000000 0

0Eh	TMRIL	Registro de carga del byte de menos peso del registro de 16 bits TMR1								xxxxxxx x	uuuuuuu u
0Fh	TMRIH	Registro de carga del byte de más peso del registro de 16 bits TMR1								xxxxxxx x	uuuuuuu u
10h	TICON	---	---	TICK PSI	TICK PS0	TIOS CEN	TISYN C#	TMR ICS	TMR ION	--000000	--uuuuuu

Cuando el módulo CCP (captura-comparación-PWM) está configurado como comparador para generar un disparo especial (CCPIM3-CCPIM0=1011), dicha señal resetea el TMR1. Para aprovechar esta característica el TMR1 debe estar configurado en modo temporizador o contador síncrono. En otro caso no se produce el reset.

2.5.3.- FUNCIONAMIENTO Y PROGRAMACIÓN DEL TMR2.

Se trata de un temporizador ascendente de 8 bits que se puede leer y escribir. También puede realizar funciones especiales para la puerta serie síncrona (SSP) y con los módulos de captura y comparación. Ocupa la dirección 11h del mapa de los registros específicos. La señal de reloj del TMR2 es interna $F_{osc}/4$ y antes de ser aplicada pasa por un predivisor de frecuencia con rangos de 1:1, 1:4 y 1:16. La salida del TMR2 atraviesa un postdivisor de frecuencia con rangos de división desde 1:1 a 1:16, pasando por los 16 valores posibles. Al entrar el microcontrolador en modo de reposo o SLEEP se detiene el oscilador interno y al no existir la señal $F_{osc}/4$ deja de funcionar el TMR2. Para controlar el funcionamiento del TMR2 se usa el registro T2CON que ocupa la dirección 12h del mapa de los registros específicos y cuya distribución y asignación de bits se muestra a continuación.

Registro	7	6	5	4	3	2	1	0
T2CON	---	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T2CKPS1-T2CKPS0=Selección del rango de división del predivisor de pulsos								
T2CKPS1 T2CKPS0 Rango del predivisor								
0 0 1:1								
0 1 1:4								
1 *								
1:16								
TMR2ON= Selecciona el funcionamiento del TMR2								
1=Permite el funcionamiento.								
0=Prohíbe el funcionamiento.								

TOUTPS0-TOUTPS3=Selección del rango del postdivisor de frecuencia

TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	Rango del postdivisor
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
0	0	1	1	1:4
0	1	0	0	1:5
0	1	0	1	1:6
0	1	1	0	1:7
0	1	1	1	1:8
1	0	0	0	1:9
1	0	0	1	1:10
1	0	1	0	1:11
1	0	1	1	1:12
1	1	0	0	1:13
1	1	0	1	1:14
1	1	1	0	1:15
1	1	1	1	1:16

El señalizador de desbordamiento del TMR2 es el bit 1 (TMR2IF) del registro PIR1, que ocupa la dirección 0Ch. El predivisor y el postdivisor se ponen a 0 al escribir el TMR2, al escribir el T2CON o con un reset. Sin embargo, cuando se escribe el T2CON no se borra el TMR2. El contenido del TMR2 pasa a valer 0 cuando se produce un reset. Este temporizador también tiene la capacidad de producir una petición de interrupción si se pone a 1 el bit TMR2IE de habilitación, que reside en el registro PIE1 (8Ch). Además, con la programación adecuada, los pulsos de salida del TMR2 (antes del postdivisor) pueden actuar como base de tiempos para el módulo SSP.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000000 x	0000000 u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIE	CCPIIF	TMR2IF	TMR1IF	0000000 0	0000000 0
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIF	CCPIIE	TMR2IE	TMR1IE	0000000 0	0000000 0
11h	TMR2	Registro del modulo TMR2								xxxxxxx x	uuuuuuu u
12h	T2CON	---	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	R2CKPSI	R2CKPS0	xxxxxxx x	uuuuuuu u
92h	PR2	Registro del periodo de TMR								--000000	--uuuuuu

2.6.- MÓDULOS DE CAPTURA, COMPARACIÓN Y MODULACIÓN DE ANCHURA DE PULSOS

Los microcontroladores PIC16F87X disponen de dos módulos CCP, llamados CCPI y CCP2, que son idénticos excepto en lo referente a la modalidad de disparo especial que luego se comenta. Dada esta similitud, la descripción se orienta hacia el módulo CCPI. Estos módulos pueden realizar tres funciones principales:

- **Modo captura:** una pareja de registros de un módulo CCPx captura el valor que tiene el TMR1 cuando ocurre un evento especial en el terminal RC2/CCPI (para el módulo CCPI) o en el RC1/TIOSI/CCP2 (para el módulo CCP2).
- **Modo comparación:** se compara el valor de 16 bits del TMR1 con otro valor cargado en una pareja de registros de un módulo CCPx y cuando coincide se produce un evento en el/los terminal/es RC2/CCPI y/o RC1/TIOSI/CCP2.
- **Modo modulación de anchura de pulsos (PWM):** dentro del intervalo del período de un pulso controla la anchura en que la señal vale nivel alto, se usa el temporizador TMR2.

El módulo CCPI utiliza un registro de trabajo de 16 bits que está formado por la concatenación de los registros CCPRIH-CCPRI L (direcciones 16h y 15h). El registro de control del módulo CCPI es el CCPICON, que ocupa la dirección 17h. El módulo CCP2 tiene como registros de trabajo a CCPR2H-CCPR2L (direcciones 1Ch y 1Bh) y como registro de control a CCP2CON en la dirección 1Dh. Las parejas de registros son las encargadas de capturar el valor del TMR1, de comparar el valor que tienen con el del TMR1 o, de modular la anchura del pulso, en el modo PWM.

Registro CCPxCON	7	6	5	4	3	2	1	0
X puede ser	---	---	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
1 ó 2	0	0	R/W	R/W	R/W	R/W	R/W	R/W

CCPxM3-CCPxM0=modo de trabajo del módulo.

CCPxM3	CCPxM2	CCPxM1	CCPxM0	Modo de trabajo del módulo
0	0	0	0	Módulo CCPx desconectado
0	1	0	0	Modo captura con cada flanco descendente en RC1/CCP2 o en RC2/CCPI
0	1	0	1	Modo captura con cada flanco ascendente en RC1/CCP2 o en RC2/CCPI
0	1	1	0	Modo captura con cada 4 flancos ascendentes RC1/CCP2o en RC2/CCPI
0	1	1	1	Modo captura con cada 16 flancos ascendentes en RC1/CCP2 o en RC2/CCPI
1	0	0	0	Modo comparación que activa el terminal RCy/CCPx a coincidir valores
1	0	0	1	Modo comparación que desactiva el terminal RCy/CCPx al coincidir valores
1	0	1	0	Modo comparación que genera una interrupción software
1	0	1	1	Modo comparación en el que se produce un disparo especial diferente para cada módulo
1	1	*	*	Modo PWM

CCPxX-CCPxY: bit menos significativos de PWM
 Modo Captura=sin usar
 Modo Comparación=sin usar
 Modo PWM=Estos dos bits son los menos significativos del ciclo de PWM. Los ocho bits más significativos se encuentran en CCPRXL.

La interacción de los tres modos de funcionar es la siguiente:

Modo CCPx	Modo CCPy	Interacción
Captura	Captura	La misma base de tiempos de TMR1
Captura	Comparación	El comparador debe configurarse para el modo de disparo especial que pone a cero el TMR1
Comparación	Comparación	El Comparador(es) debe configurarse para el modo de disparo especial que pone a cero el TMR1
PWM	PWM	El PWM tendrá la misma frecuencia y proporción de actuación (interrupción de TMR2)
PWM	Captura	Ninguna
PWM	Comparación	Ninguna

2.6.1.- MODO CAPTURA.

En este modo, la pareja de registros CCPxH-L del módulo CCPx captura el valor de 16 bits que contiene el TMR1 cuando sucede un evento en el terminal RCy/CCPx de la puerta C, que previamente ha sido configurada como entrada poniendo a 1 el bit correspondiente del registro TRISC.

Los eventos posibles que pueden ocurrir sobre el terminal RCy/CCPx para producir la captura del valor del TMR1 sobre la pareja de registros CCPxH-L son:

- Un flanco ascendente.
- Un flanco descendente.
- Cada 4 flancos ascendentes.
- Cada 16 flancos ascendentes.

Los cuatro bits CCPIM3-0 del registro CCPICON seleccionan el evento adecuado en el módulo CCP1 y otro tanto sucede con el módulo CCP2. Al efectuar la captura, se activa el señalizador CCP1IF del registro PIR1. Además, si se pone a 1 el bit de permiso de interrupción PIE1 (CCPIIE), se genera una petición de interrupción cuando se carga en CCPR1H-L el valor del TMR1.

Cuando se emplea el módulo CCP1 en modo captura el TMR1 debe estar configurado para trabajar como temporizador o como contador síncrono, nunca en modo asíncrono.

Si se van a cambiar las condiciones de funcionamiento en el modo captura conviene detener o desactivar al módulo CCP para evitar que se produzcan falsas interrupciones durante la operación.

En el caso de que no se haya leído el contenido de los registros CCPR1H-L y se produce una nueva captura dichos registros pasan a contener el nuevo valor.

Cuando se desactiva el módulo CCP o deja de funcionar en modo captura se borra la codificación del previsor de frecuencia que determinan los bits CCPIM3-0.

Una aplicación muy interesante del modo captura es la medición de los intervalos de tiempo que existen entre los pulsos que llegan al terminal RC2/CCPI que se halla configurada como entrada. En este caso el TMR1 debe trabajar con entrada de reloj externo sincronizada.

2.6.2.- MODO COMPARACIÓN.

En esta forma de trabajo, la pareja de registros CCPRIH-L compara su contenido de forma continua con el valor del TMR1. Cuando coinciden ambos valores el terminal RC2/CCPI, que se halla configurada como salida, sufre algunos de los siguientes eventos, de acuerdo con la programación de los bits CCPIM3-0:

Pasa a nivel alto.

Pasa a nivel bajo.

No cambia su estado pero se produce una interrupción.

Al coincidir los valores del TMR1 con los de la pareja de registros CCPRIH-L se pone a 1 el señalizador CCPIIE. El TMR1 debe trabajar en modo temporizador o contador síncrono, nunca en modo asíncrono.

Si el bit de permiso de interrupción está a 1, cuando coinciden los valores mencionados se origina una petición de interrupción.

Cuando se selecciona el modo de trabajo de “disparo especial” con los bits CCPIM3-0 el módulo CCPI pone a 0 el TMR1 y el CCPRI funciona como un registro de período, capaz de provocar periódicamente interrupciones. En ese modo, el CCP2 pone a 0 el TMR1 y, además, inicia una conversión en el conversor A/D. De esta forma, puede realizar conversiones analógicos/digitales sin el control del programa de instrucciones de forma periódica.

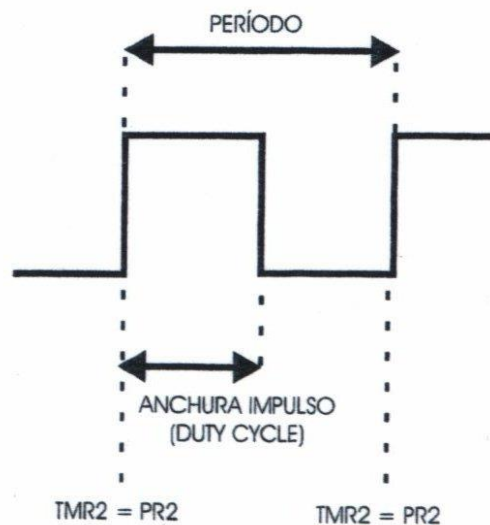
En la siguiente tabla se presentan los registros (con la distribución de sus bits) que participan en la programación de los módulos de captura y comparación de los dispositivos CCPI y CCP2:

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIIF	TMR2IF	TMR1IF
87h	TRISC	Registro de configuración de la puerta C							
8Ch	TMR1L	Registro de carga del byte de menos peso del registro de 16 bits TMR1							
0Eh	TMR1H	Registro de carga del byte de más peso del registro de 16 bits TMR1							
10h	TICON	----	----	TICKPS1	TICKPS0	TIOSCEN	TISYNC #	TMR1CS	TMR1ON
15h	CCP1L	Registro del byte de menos peso del módulo de captura/comparación /modulación							
16h	CCP1H	Registro del byte de más peso del módulo de captura/comparación /modulación							
17h	CCP1CON	----	----	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0

2.6.3.- MODO DE MODULACIÓN DE ANCHURA DE PULSOS (PWM).

Con este modo de trabajo se consiguen pulsos lógicos con anchura del nivel alto de duración variable, que son de enorme aplicación en el control de dispositivos como los motores y los triacs.

El terminal RC2/CCPI está configurado como salida y bascula entre los niveles lógicos 0 y 1 a intervalos variables de tiempo. Lo que se intenta es obtener un pulso cuyo nivel alto tenga una anchura variable ("duty cycle", factor de trabajo) dentro del intervalo del período de trabajo, como se observa en la figura inferior. Para lograr el basculado del terminal de salida RC2/CCPI se usa un comparador que pone a 1 (set) un flip-flop cuando el valor del registro PR2 coincide con la parte alta del TMR2, momento en que el TMR2 toma el valor 00h. El flip-flop se resetea (se pone a 0) cuando otro comparador detecta la coincidencia del valor existente en CCPR1H con el de la parte alta del TMR2.



De esta manera, variando los valores que se cargan en PR2 y en CCPR1L (que luego se traspassa al CCPR1H) se varía el intervalo de tiempo en el que el terminal de salida está a 1 y a 0.

Cuando se trabaja con una precisión de 10 bits, los 2 bits CCPICON (5,4) se concatenan con los 8 de CCPR1L y, de la misma forma, los 8 bits de más peso del TMR2 se concatenan con los dos bits de menos peso del reloj interno, haciendo que el TMR2 cuente cada T_{osc} en vez de cada $4 \cdot T_{osc}$.

El tiempo que dura el período de la onda depende del valor cargado en PR2, según la fórmula siguiente:

$$Periodo = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot Valor_predivisor_TMR2$$

Cuando el valor del TMR2 coincide con el del PR2 se concatenan tres acontecimientos:

- Se borra el TMR2.
- El terminal RC2/CCPI se pone a 1.

- El valor de CCPRIL, que es el que determina la anchura de pulso, se carga en CCPRIH.

El tiempo que el terminal de salida está a nivel alto, que es la anchura del pulso, depende del contenido cargado en CCPRIL y de los dos bits 5 y 4 del CCPICON cuando se trabaja con una precisión de 10 bits.

$$\text{Anchura_de_pulso} = (\text{CCPRIL} : \text{CCPICON}(5,4)) \cdot \text{Tosc} \cdot \text{Valor_previsor_TMR2}$$

El valor CCPRIL: CCPICON(5,4) puede cargarse en cualquier momento, puesto que el mismo no se traspasa a CCPRIH y se compara hasta que coincidan PR2 con TMR2. En el modo PWM el registro CCPRIL sólo puede ser leído. En la siguiente tabla se resumen registros que intervienen en la programación del modo PWM.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000000x	0000000u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIIF	TMR2IF	TMR1IF	00000000	00000000
08h	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPIIE	TMR2IE	TMR1IE	00000000	00000000
87Ch	TRISC	Registro de configuración de la puerta C								11111111	11111111
11Eh	TMR2	Registro del módulo TMR2								xxxxxxxx	uuuuuuuu
12h	T2CON	---	TOUT PS3	TOUT PS2	TOUT PS1	TOUT PS0	TMR2O N	T2CK PS1	T2C KPS0	xxxxxxxx	uuuuuuuu
92h	PR2	Registro de período del TMR2								--000000	--uuuuuu
15h	CCPRIL	Registro del byte de menos peso del módulo de captura/comparación /modulación								xxxxxxxx	uuuuuuuu
16h	CCPRIH	Registro del byte de más peso del módulo de captura/comparación /modulación								xxxxxxxx	--000000
17h	CCPICON	---	---	CCPIX	CCPIY	CCPIM3	CCPIM2	CCPIMI	CCPIM0	--000000	--000000

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000000x	0000000u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIIF	TMR2IF	TMR1IF	00000000	00000000
0Dh	PIR2	---	0	---	EEIF	BCLIF	---	---	CCP2IF	-----0	-----0
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPIIE	TMR2IE	TMR1IE	00000000	00000000
8Dh	PIE2	---	0	---	EEIE	BCLIE	---	---	CCP2IE	-----0	-----0

87Ch	TRISC	Registro de configuración de la puerta C									
0Eh	TMR1L	Registro de carga del byte de menos peso del registro de 16 bits TMR1								xxxxxxxx	uuuuuuuu
0Fh	TMR1H	Registro de carga del byte de más peso del registro de 16 bits TMR1								xxxxxxxx	uuuuuuuu
10h	TICON	----	----	TICK PS1	TICK PS0	TIOS CEN	TISYN C#	TMR1 CS	TMR ION	--000000	--uuuuuu
15h	CCPR1L	Registro 1 del byte de menos peso del módulo de captura/comparación /modulación								xxxxxxxx	uuuuuuuu
16h	CCPR1H	Registro 1 del byte de más peso del módulo de captura/comparación /modulación								xxxxxxxx	--000000
17h	CCP1CON	---	---	CCP1 X	CCP1 Y	CCP1 M3	CCP1M 2	CCP1 MI	CCP IM0	--000000	--000000
1Bh	CCPR2L	Registro 2 del byte de menos peso del módulo de captura/comparación /modulación								xxxxxxxx	uuuuuuuu
1Ch	CCPR2H	Registro 2 del byte de más peso del módulo de captura/comparación /modulación								xxxxxxxx	--000000
1Dh	CCP2CON	---	---	CCP2 X	CCP2 Y	CCP2 M3	CCP2M 2	CCP2 MI	CCP 2M0	--000000	--000000

En la siguiente tabla donde se recogen todos los registros que intervienen en la captura, comparación y el TMR1. De igual forma se recogen los registros que intervienen en la modulación de ancho de pulso y el TMR2.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000000x	0000000u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	00000000	00000000
0Dh	PIR2	---	0	---	EEIF	BCLIF	---	---	CCP2IF	-----0	-----0
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	00000000	00000000
8Dh	PIE2	---	0	---	EEIE	BCLIE	---	---	CCP2IE	-----0	-----0
87Ch	TRISC	Registro de configuración de la puerta C									
11Eh	TMR2	Registro del módulo TMR2								xxxxxxxx	uuuuuuuu
92h	PR2	Registro de período del TMR2								--000000	--uuuuuu
12h	T2CON	---	TOUT PS3	TOUT PS2	TOUT PS1	TOUT PS0	TMR2O N	T2CK PS1	T2C KPS0	xxxxxxxx	uuuuuuuu
15h	CCPR1L	Registro del byte de menos peso del módulo de captura/comparación /modulación								xxxxxxxx	uuuuuuuu
16h	CCPR1H	Registro del byte de más peso del módulo de captura/comparación /modulación								xxxxxxxx	--000000
17h	CCP1CON	---	---	CCP1 X	CCP1 Y	CCP1 M3	CCP1M 2	CCP1 MI	CCP IM0	--000000	--000000
1Bh	CCPR2L	Registro 2 del byte de menos peso del módulo de captura/comparación /modulación								xxxxxxxx	uuuuuuuu

ICh	CCPR2H	Registro 2 del byte de más peso del módulo de captura/comparación /modulación								xxxxxxxx	--000000
IDh	CCP2CON	---	---	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2MI	CCP2M0	--000000	--000000

Los pasos a seguir para realizar la configuración del modo PWM son los siguientes:

- Asignar el período cargando el oportuno valor en PR2.
- Asignar la anchura de pulso cargando el registro CCPR1L y los dos bits 5 y 4 del CCP1CON.
- Configurar la línea RC2/CCPI como salida.
- Asignar el valor del predivisor y activar el TMR2 escribiendo en T2CON.
- Configurar el módulo CCP1 en modo PWM.

2.7.- EL CONVERTOR A/D

Los microcontroladores PIC16F877 poseen un convertor A/D de 10 bits de resolución y 8 canales. La resolución que tiene cada bit procedente de la conversión tiene un valor que es función de la tensión de referencia Vref, de acuerdo con la fórmula siguiente:

$$\text{Resolución} = \frac{(V_{\text{ref}+} - V_{\text{ref}-})}{1024} = \frac{V_{\text{ref}}}{1024}$$

Así, por ejemplo, si la Vref+ = 5 VDC y la Vref- es tierra, la resolución es de 4,8 mV/bit. Por tanto, a la entrada analógica de 0V le corresponde un valor digital de 00 0000 0000 y para la de 5V uno de 11 1111 1111. La tensión de referencia determina los límites máximo y mínimo de la tensión analógica que se puede convertir. La tensión diferencial mínima es de 2V. A través del canal de entrada seleccionado se aplica la señal analógica a un condensador de captura y mantenimiento ("sample" and "hold") y luego se introduce al convertor, el cual proporciona un resultado digital de 10 bits de longitud usando la técnica de "aproximaciones sucesivas".

El convertor A/D es el único dispositivo que puede funcionar en modo reposo (SLEEP); para ello el reloj del convertor debe conectarse al oscilador RC interno. La tensión de referencia puede implementarse con la tensión interna de alimentación VDD, o bien, con una externa que se introduce por el terminal RA3/AN3/Vref+, en cuyo caso la polaridad negativa se aplica por el terminal RA2/AN2/Vref-.

2.7.1.- REGISTROS DE TRABAJO.

El funcionamiento del convertor A/D requiere la manipulación de cuatro registros:

- ADRESH: parte alta del resultado de la conversión.
- ADRESL: parte baja del resultado de la conversión.
- ADCON0: registro de control 0.
- ADCON1: registro de control 1.

En la pareja de registros ADRESH-ADRESL se deposita el resultado de la conversión, sólo son significativos 10 de los bits de dicha pareja. El registro ADCON0 controla la

operación del convertor A/D, mientras que el ADCON1 sirve para configurar los terminales de la puerta A como entradas analógicas o E/S digitales (ver las tablas inferiores). Los bits ADCON0 (7,6) sirven para seleccionar la frecuencia de reloj que se emplea en la conversión, con la siguiente asignación:

ADCS1-0	Frecuencia
00	FOSC/2
01	FOSC/8
10	FOSC/32
11	FRC (procede del oscilador RC interno)

Se designa como TAD el tiempo que dura la conversión de cada bit. En el caso de trabajar con valores digitales de 10 bits se requiere un tiempo mínimo de $12 \cdot TAD$. El valor de TAD se selecciona por software mediante los bits ADCS1:ADCS0, en los PIC16F87X nunca debe ser menor de 1,6 microsegundos.

En la tabla se presentan los diversos valores que toma TAD según los bits ADCS1:ADCS0 y la frecuencia de trabajo del microcontrolador. Así, por ejemplo, si dichos bits toman el valor 00 $TAD=2 \cdot T_{osc}$ y si el PIC funciona a 20 MHz resulta que $TAD=100ns$, tiempo no válido. Si la frecuencia es de 5 MHz, $TAD=400ns$, valor que tampoco es válido. Si la frecuencia es de 1,25 MHz, $TAD=1,6$ microsegundos, valor válido. Finalmente, para una frecuencia de 333,33 KHz, $TAD=6$ microsegundos.

Selección T_{AD}		Frecuencia de trabajo			
T_{AD}	ADCS1:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz
$2 \cdot T_{osc}$	00	100 ns	400 ns	1.6 μs	6 μs
$8 \cdot T_{osc}$	01	400 ns	1.6 ns	6.4 μs	24 μs
$32 \cdot T_{osc}$	10	1.6 μs	6.4 μs	2.6 μs	96 μs
RC oscilador interno en el C A/D	11	2-6 μs	2-6 μs	2-6 μs	2-6 μs

Figura 1.- Tabla que presenta los valores que toma TAD, tiempo de conversión por bit, según la programación de ADCS1-ADCS0 y la frecuencia de funcionamiento del microcontrolador. Para el convertor A/D de los PIC16F87X el valor mínimo de TAD es de 1,6 microsegundos.

Los bits CHS2-0 seleccionan el canal por el que se introduce la señal analógica a convertir de acuerdo con el siguiente código:

CH2-0	Canal
000	Canal 0 (RA0/AN0)
001	Canal 1 (RA1/AN1)
010	Canal 2 (RA2/AN2)
011	Canal 3 (RA3/AN3)
100	Canal 4 (RA5/AN4)
101	Canal 5 (RE0,AN5). Los PIC16F87X de 28 terminales no tienen este canal
110	Canal 6 (RE1,AN6). Los PIC16F87X de 28 terminales no tienen este canal
111	Canal 7 (RE2,AN7). Los PIC16F87X de 28 terminales no tienen este canal

El bit **GO/DONE#** es el “bit de estado de la conversión”. Poniéndolo a 1 se inicia la conversión y mientras esté a 1 está realizándose dicha operación. Cuando **GO/DONE#** pasa a 0 confirma el final de la conversión y la puesta del resultado en la pareja de registros **ADRESH-L**. El bit **ADON** sirve para activar el conversor A/D poniéndolo a 1 y para inhibir su funcionamiento poniéndolo a 0.

Registro ADCON0	7	6	5	4	3	2	1	0																																				
	ADCSI	ADCS0	CHS2	CHS1	CHS0	GO/DONE#	---	ADON																																				
	R/W	R/W	R/W	R/W	R/W	R/W	0	R/W																																				
ADON=Activación del conversor A/D 1=conversor A/D activo 0=conversor A/D inactivo																																												
GO/DONE#=Estado de la conversión 1=poniéndolo a 1 se inicia la conversión y mientras esté a 1 está realizándose dicha operación. 0=confirma el final de la conversión y la puesta del resultado en la pareja de registros ARDES-L .																																												
CHS2-CHS0=Selección del canal <table border="1"> <thead> <tr> <th>CH2</th> <th>CH1</th> <th>CH0</th> <th>Canal</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Canal 0 (RA0/AN0)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Canal 1 (RA1/AN1)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Canal 2 (RA2/AN2)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Canal 3 (RA3/AN3)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Canal 4 (RA5/AN4)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Canal 5 (RE0,AN5). Los PIC16F87X de 28 terminales no tienen este canal</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Canal 6 (RE1,AN6). Los PIC16F87X de 28 terminales no tienen este canal</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Canal 7 (RE2,AN7). Los PIC16F87X de 28 terminales no tienen este canal</td> </tr> </tbody> </table>									CH2	CH1	CH0	Canal	0	0	0	Canal 0 (RA0/AN0)	0	0	1	Canal 1 (RA1/AN1)	0	1	0	Canal 2 (RA2/AN2)	0	1	1	Canal 3 (RA3/AN3)	1	0	0	Canal 4 (RA5/AN4)	1	0	1	Canal 5 (RE0,AN5). Los PIC16F87X de 28 terminales no tienen este canal	1	1	0	Canal 6 (RE1,AN6). Los PIC16F87X de 28 terminales no tienen este canal	1	1	1	Canal 7 (RE2,AN7). Los PIC16F87X de 28 terminales no tienen este canal
CH2	CH1	CH0	Canal																																									
0	0	0	Canal 0 (RA0/AN0)																																									
0	0	1	Canal 1 (RA1/AN1)																																									
0	1	0	Canal 2 (RA2/AN2)																																									
0	1	1	Canal 3 (RA3/AN3)																																									
1	0	0	Canal 4 (RA5/AN4)																																									
1	0	1	Canal 5 (RE0,AN5). Los PIC16F87X de 28 terminales no tienen este canal																																									
1	1	0	Canal 6 (RE1,AN6). Los PIC16F87X de 28 terminales no tienen este canal																																									
1	1	1	Canal 7 (RE2,AN7). Los PIC16F87X de 28 terminales no tienen este canal																																									
ADCSI-ADCS0=Selección de la frecuencia del reloj <table border="1"> <thead> <tr> <th>ADCSI</th> <th>ADCS0</th> <th>Frecuencia</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>$F_{osc}/2$</td> </tr> <tr> <td>0</td> <td>1</td> <td>$F_{osc}/8$</td> </tr> <tr> <td>1</td> <td>0</td> <td>$F_{osc}/32$</td> </tr> <tr> <td>1</td> <td>1</td> <td>F_{RC} (procede del oscilador RC interno)</td> </tr> </tbody> </table>									ADCSI	ADCS0	Frecuencia	0	0	$F_{osc}/2$	0	1	$F_{osc}/8$	1	0	$F_{osc}/32$	1	1	F_{RC} (procede del oscilador RC interno)																					
ADCSI	ADCS0	Frecuencia																																										
0	0	$F_{osc}/2$																																										
0	1	$F_{osc}/8$																																										
1	0	$F_{osc}/32$																																										
1	1	F_{RC} (procede del oscilador RC interno)																																										

Figura 2.- Asignación de los bits del registro de control del conversor A/D ADCON0.

El bit de menos peso **ADFM** del registro **ADCON1** selecciona el formato del resultado de la conversión. Si vale 1 el resultado está justificado en el registro **ADRESH**, que tiene sus 6 bits de más peso a 0; mientras que si vale 0 la justificación se realiza sobre el registro **ADRESL**, que tiene sus 6 bits de menos peso a 0.

Registro ADCON1	7	6	5	4	3	2	1	0
	ADFM	---	---	---	PCFG3	PCFG2	PCFG1	PCFG0
	R/W	0	0	0	R/W	R/W	R/W	R/W

PCFG3-PCFG0=Configuración			de					los		terminales
PCFG3	PCFG2	PCFG1	PCFG0	AN4/RA5	AN3/RA3	AN2/RA2	AN1/RA1	AN0/RA0		
V_{REF+}	V_{REF-}	REFS								
0	0	0	0	A	A	A	A	A	A	
V_{DD}	V_{SS}	8/0								
0	0	0	1	A	V_{REF+}	A	A	A	A	
RA3	V_{SS}	7/1								
0	0	1	0	A	A	A	A	A	A	
V_{DD}	V_{SS}	5/0								
0	0	1	1	A	V_{REF+}	A	A	A	A	
RA3	V_{SS}	4/1								
0	1	0	0	D	A	D	A	A	A	
V_{DD}	V_{SS}	3/0								
0	1	0	1	D	V_{REF+}	D	A	A	A	
RA3	V_{SS}	2/1								
0	1	1	*	D	D	D	D	D	D	
V_{DD}	V_{SS}	0/0								
1	0	0	0	A	V_{REF+}	V_{REF-}	A	A	A	
RA3	RA2	6/2								
1	0	0	1	A	A	A	A	A	A	
V_{DD}	V_{SS}	6/0								
1	0	1	0	A	V_{REF+}	A	A	A	A	
RA3	V_{SS}	5/1								
1	0	1	1	A	V_{REF+}	V_{REF-}	A	A	A	
RA3	RA2	4/2								
1	1	0	0	A	V_{REF+}	V_{REF-}	A	A	A	
RA3	RA2	3/2								
1	1	0	1	D	V_{REF+}	V_{REF-}	A	A	A	
RA3	RA2	2/2								
1	1	1	0	D	D	D	D	D	A	
V_{DD}	V_{SS}	1/0								
1	1	1	1	D	V_{REF+}	V_{REF-}	D	A	A	
RA3	RA2	1/2								

ADFM=Selección del formato del resultado de la conversión
 1=el resultado está justificado en el registro ADRESH, que tiene sus 6 bits de más peso a 0
 0=la justificación se realiza sobre el registro ADRESL, que tiene sus 6 bits de menos peso a 0

Registro ADCON1	7	6	5	4	3	2	1	0
	ADFM	---	---	---	PCFG3	PCFG2	PCFG1	PCFG0
	R/W	0	0	0	R/W	R/W	R/W	R/W

Bit 3	Bit 2	Bit 1	Bit 0	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	V_{REF+}	V_{REF-}	CAN/ REF
0	0	0	0	A	A	A	A	A	A	A	A	V_{DD}	V_{SS}	8/0

0	0	0	1	A	A	D	A	V _{REF+}	A	A	A	RA3	V _{SS}	7/1
0	0	1	0	D	D	D	A	A	A	A	A	V _{DD}	V _{SS}	5/0
0	0	1	1	D	D	D	A	V _{REF+}	A	A	A	RA3	V _{SS}	4/1
0	1	0	0	D	D	D	D	A	D	A	A	V _{DD}	V _{SS}	3/0
0	1	0	1	D	D	D	D	V _{REF+}	D	A	A	RA3	V _{SS}	2/1
0	1	1	*	D	D	D	D	D	D	D	D	V _{DD}	V _{SS}	0/0
1	0	0	0	A	A	A	A	V _{REF+}	V _{REF-}	A	A	RA3	RA2	6/2
1	0	0	1	D	D	A	A	A	A	A	A	V _{DD}	V _{SS}	6/0
1	0	1	0	D	D	A	A	V _{REF+}	A	A	A	RA3	V _{SS}	5/1
1	0	1	1	D	D	A	A	V _{REF+}	V _{REF-}	A	A	RA3	RA2	4/2
1	1	0	0	D	D	D	A	V _{REF+}	V _{REF-}	A	A	RA3	RA2	3/2
1	1	0	1	D	D	D	D	V _{REF+}	V _{REF-}	A	A	RA3	RA2	2/2
1	1	1	0	D	D	D	D	D	D	D	A	V _{DD}	V _{SS}	1/0
1	1	1	1	D	D	D	D	V _{REF+}	V _{REF-}	D	A	RA3	RA2	1/2

ADFM=Selección del formato del resultado de la conversión
 1=el resultado está justificado en el registro ADRESH, que tiene sus 6 bits de más peso a 0
 0=la justificación se realiza sobre el registro ADRESL, que tiene sus 6 bits de menos peso a 0

Figura 3.- Asignación de los bits del registro de control del convertor A/D ADCON1. Tabla que determina los terminales del PIC que actúan como entradas analógicas o como E/S digitales, según el valor de los 4 bits de ADCON1(PCFG3-).

Esto significa que los 16 bits que forman la concatenación de ARDESH-ADRESL unas veces tienen a 0 los 6 bits de más peso y otras los 6 bits de menos peso (alineación a la derecha o a la izquierda). Los restantes cuatro bits PCFG3-0 de ADCON1 se usan para configurar los terminales de los canales de entrada al convertor como analógicos o como E/S digitales.

A continuación se indican los pasos que hay que efectuar para realizar una conversión A/D:

1. Configurar el módulo A/D.
 - Configurar los terminales que actuarán como entradas analógicas, las que trabajan como E/S digitales y las usadas para la tensión de referencia (ADCON1).
 - Seleccionar el reloj de la conversión (ADCON0).
 - Seleccionar el canal de entrada A/D (ADCON0).
 - Activar el módulo A/D (ADCON0).
2. Activar, si se desea, la interrupción escribiendo sobre PIE1 y PIR1.
 - Borrar el señalizador ADIF (bit 6 del registro PIR1).
 - Poner a 1 el bit ADIE (bit 6 del registro PIE1).
 - Poner a 1 los bits habilitadores GIE y PEIE (bit 7 y 6, respectivamente, del registro INTCON).
3. Tiempo de espera para que transcurra el tiempo de adquisición.

4. Inicio de la conversión.
 - Poner a 1 el bit GO/DONE# (ADCON0).
5. Tiempo de espera para completar la conversión A/D que puede detectarse.
 - Por la exploración del bit GO/DONE#, que al completarse la conversión pasa a valer 0.
 - Esperando a que se produzca la interrupción si se ha programado, al finalizar la conversión.
 - Aunque no se produzca interrupción, el señalizador ADIF se pondrá a 1 al finalizar la conversión.
6. Leer el resultado en los 10 bits válidos de ARDESH-ADRESL y borrar el flag ADIF.
7. Para un nueva conversión regresar al paso 1° o al 2°. El tiempo de conversión por bit está asociado por TAD. Se exige esperar un mínimo de $2 \cdot TAD$ para reiniciar una nueva conversión.

Si se elige como reloj para la conversión al oscilador RC interno del convertor A/D, éste puede seguir funcionando cuando se introduce el microcontrolador al modo de reposo o SLEEP. En los restantes modos se aborta la conversión y se desactiva el convertor A/D. En la siguiente figura se observa el organigrama sobre la operatividad del convertor A/D.

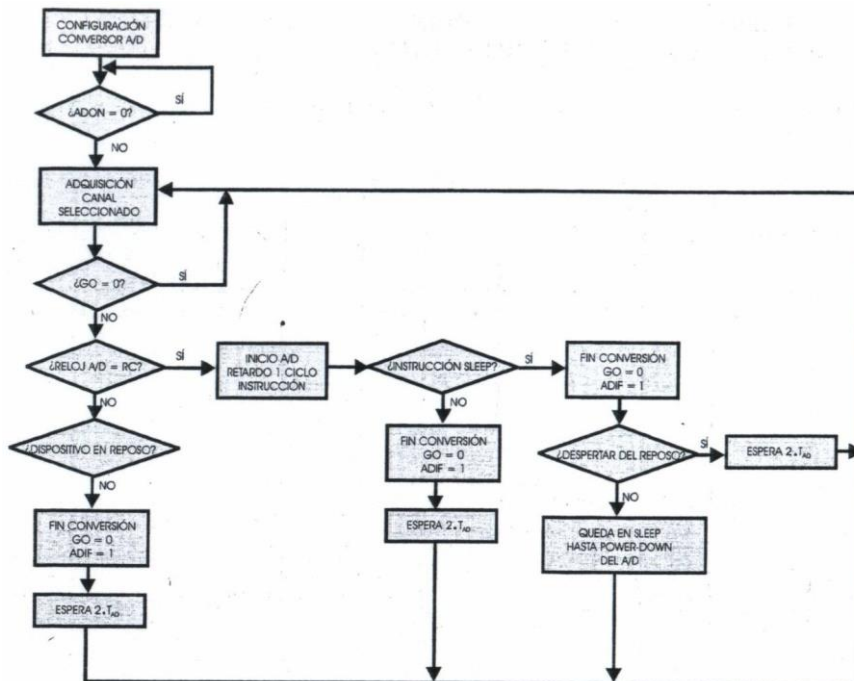


Figura 4.- Organigrama del funcionamiento del convertor A/D.

La siguiente tabla contiene todos los registros que participan en la programación del convertor A/D.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000000 x	0000000 u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPI1F	TMR2IF	TMR1IF	0000000 0	0000000 0
08h	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPI1E	TMR2IE	TMR1IE	0000000 0	0000000 0
1Eh	ADRESH	Bit de mayor peso del resultado de la conversión A/D							xxxxxxx x	uuuuuuu u	
9Eh	ADRESL	Bit de menos peso del resultado de la conversión A/D							---	---	
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE#	---	ADON	xxxxxxx x	uuuuuuu u
9Fh	ADCON1	ADFM	---	---	---	PCFG3	PCFG2	PCFG1	PCFG0	000000-0	000000-0
85h	TRISA	---	---	Registro de configuración de la puerta A					--0-0000	--0-0000	
05h	PORTA	---	---	RA5	RA4	RA3	RA2	RA1	RA0	--0x0000	--0u0000
89h	TRISE	IBF	OBF	IBOV	PSPMODE	---	Registro de configuración de la puerta E			0000-111	0000-111
09h	PORTE	---	---	---	---	---	RE2	RE1	RE0	-----xxx	-----uuu

Figura 5.- Tabla que ofrece la distribución de los bits de los registros que intervienen en la programación del conversor A/D.

2.8.- MÓDULO DE COMUNICACIÓN SERIE SÍNCRONA: MSSP

La comunicación serie es una forma muy útil de transferir datos digitales entre sistemas y circuitos integrados, dada la reducida cantidad de líneas que precisa.

En el PIC16F84, cuando se quería comunicarle en serie con otro dispositivo, había que destinar varios de sus terminales generales de E/S a soportar las líneas de datos y de reloj. También había que desarrollar un programa específico para soportar el protocolo de comunicación.

En los PIC16F87X se ha implantado el módulo MSSP ("Master Synchronous Serial Port"), que proporciona un excelente interfaz de comunicación entre microcontroladores y entre microcontroladores y diversos periféricos, entre los que destacan las memorias EEPROM serie, los conversores A/D y los controladores de displays. De esta forma, el usuario dispone de algunos terminales de la puerta C que programados correctamente sirven para controlar directamente la comunicación serie. Además, el módulo MSSP admite dos de las alternativas más usadas en la comunicación serie síncrona:

- SPI ("Serial Peripheral Interface").
- I2C ("Inter.-Integrated Circuit").

La comunicación serie en modo SPI la utilizan principalmente las memorias (RAM y EEPROM) y se utilizan tres líneas para llevarla a cabo. En el modo I2C sólo se emplean dos líneas y ha conseguido una importante implantación en la comunicación de circuitos integrados, existiendo en el mercado todo tipo de periféricos capaces de trabajar con este protocolo (memorias, controladores, relojes, convertidores, etc.).

El módulo MSSP consta básicamente de dos registros: el SSPSR, que es un registro de desplazamiento que transforma la información serie en paralelo y viceversa, y el registro SSPBUF, que actúa como buffer de la información que se recibe o transmite.

El funcionamiento del módulo MSSP es el siguiente. En transmisión, el byte que se quiere transmitir se carga en el registro SSPBUF a través del bus de datos interno y automáticamente se traspassa al registro SSPSR, que va desplazando bit a bit el dato sacándolo ordenadamente al exterior al ritmo de los pulsos de reloj. En recepción los bits van entrando al ritmo de reloj por un terminal y se van desplazando en el SSPSR hasta que lo llenan, en cuyo momento la información se traspassa al SSPBUF, donde queda listo para su lectura. Este doble almacenamiento del dato recibido permite iniciar la recepción de un nuevo dato antes de que se haya leído el último.

Cuando se han recibido 8 bits se traspassa dicha información a SSPBUF y el bit señalizador BF (Buffer Full) se pone a 1, al igual que el flag de interrupción SSPIF. Cualquier escritura en el SSPBUF se ignora durante una transferencia de información y se señala poniendo a 1 el bit WCOL, que advierte de este tipo de colisiones. Es responsabilidad del programador pasar el bit WCOL a 0 una vez completada la escritura en SSPBUF.

2.8.1.- MODO SPI.

Permite la transferencia de datos de 8 bits en serie que pueden ser transmitidos y recibidos de forma síncrona y simultánea. Para el establecimiento de la comunicación se utilizan tres líneas:

- SD0 (Serial Data Out): salida de datos en serie.
- SDI (Serial Data In): entrada de datos en serie.
- SCK (Serial Clock): reloj de sincronización

Puede ser necesaria utilizar una cuarta línea de control más cuando el PIC que se utiliza trabaja en modo esclavo. En este caso, el terminal SS# (selección de esclavo) se debe conectar a tierra. En la siguiente figura se observa un esquema de un sistema intercomunicado en modo SPI en el que un PIC actúa como maestro y otros dos como esclavos. Las cuatro líneas que utilizan se corresponden con los terminales multifunción RC3/SDO, RC4/SDI, RC5/SCK y RA5/SS#.

La conexión habitual del PIC maestro se suele realizar con circuitos de memoria con el objeto de ampliar su capacidad. Como se aprecia en la figura la línea SDO del maestro se corresponde con las SDI de los esclavos y la línea SCK por la que circulan los pulsos de reloj siempre parte del maestro, que es el encargado de generar y controlar la sincronización.

Si, por ejemplo, actuase como esclavo un chip de memoria RAM de 256x8 de tamaño, la comunicación SPI la iniciaría el maestro enviando por la línea SDO un byte con la dirección de la memoria a acceder, seguido de otro byte que especificaría la operación

(lectura/escritura) y un tercero que contendría el dato a escribir en caso de que se tratase de una operación de escritura. En el caso de que se tratase de una lectura, el maestro, después de enviar los 2 bytes iniciales, quedaría a la espera del byte que sacaría el esclavo por su línea SDO y que se introduciría al maestro por su línea SDI. Si fuese una memoria con más posiciones, la dirección se tendría que especificar con más de un byte.

Cuando el PIC trabaja como maestro hay que programar la línea RC3/SDO como salida, la línea RC4/SDI como entrada y la línea RC5/SCK también como salida. Si actuase como esclavo la línea RC5/SCK debería configurarse como entrada y la RA5/SS# debería conectarse a tierra.

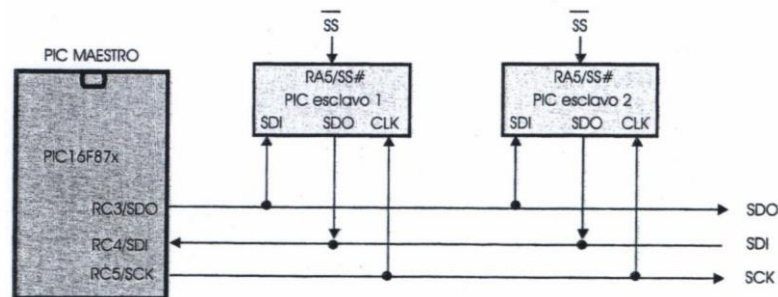


Figura 6.- Comunicación SPI entre un PIC maestro y otros dos esclavos.

2.8.2.- MANEJO Y PROGRAMACIÓN EN EL MODO SPI.

En el comienzo del funcionamiento del modo SPI es necesario establecer las condiciones de trabajo programando adecuadamente los bits (5...0) del registro SSPCON y los (7,6) del registro SSPSTAT. De esta forma, se determinan las siguientes especificaciones:

- Si trabaja en modo maestro, SCK debe ser salida.
- Si trabaja en modo esclavo, la línea SCK del reloj debe ser entrada.
- Hay que determinar la polaridad de la señal de reloj, es decir, su estado de inactividad.
- Se debe precisar el momento en el que se toma la muestra del dato de entrada, que puede ser en la mitad o al final del pulso.
- Flanco de reloj activo, que puede ser el ascendente o descendente de SCK.
- Rango de trabajo del reloj en el modo maestro.
- Selección del modo esclavo, en caso de trabajar de esta forma.

En la siguiente tabla se observa la distribución y nomenclatura de los bits del registro SSPSTAT, que ocupa la dirección 94h y es uno de los que seleccionan las características mencionadas.

A continuación se describe la misión de tres de los bits del registro SSPSTAT que participan en la configuración del modo SPI. A su vez, participan en la configuración del modo I2C junto a los restantes, que se verán al presentar dicho protocolo.

- SMP.
 - El bit 7 del registro SSPSTAT determina el momento del muestreo del bit de entrada.

- En modo maestro SPI, si SMP=1 el dato de entrada se muestrea al final del pulso de reloj y si vale 0 en la mitad del mismo.
- En modo esclavo el bit SMP no tiene cometido y debe valer 0.
- CKE.
 - Con este bit se selecciona el flanco de reloj activo.
 - Si el bit CKP=0 en el registro SSPCON significa que el estado de inactividad en la señal de reloj es el bajo y entonces si CKE=1 el dato se transmite en el flanco ascendente de SCK y si CKE=0 en el flanco descendente.
 - Si CKP=1, lo que supone que es el estado alto el de inactividad en la señal de reloj, cuando CKE=1 el dato se transmite en el flanco descendente de SCK y si CKE=0 en el ascendente.
- BF.
 - Indica cuando vale 1, que el registro SSPBUF está lleno.

Registro SSPSTAT	7	6	5	4	3	2	1	0
	SMP	CKE	D/A#	P	S	R/W#	UA	BF
	R/W	R/W	R	R	R	R	R	R

BF=Señalizador del buffer de datos 1=tiene un dato y la transmisión está en progreso sin terminar 0=no tiene dato
UA=Señalizador del número de bits 1=la dirección es de 10 bits 0=la dirección es de 7 bits
R/W#=Indica si se trata de una escritura o lectura 1=lectura 0=escritura
S=Detecta la condición de inicio 1=inicio 0=no inicio
P=Detecta la condición de stop o llegada del bit 1=llegada del bit. 0=no llegada de bit
D/A#=Indica si el dato recibido es de información o es una dirección 1=información 0=dirección
CKE=Selección de los niveles de los terminales SCL y SDA 1=flanco descendente 0=flanco ascendente
SMP=Bit de muestreo 1=los bits de datos se muestrean al final del periodo 0=los bits de datos se muestrean en la mitad del período

En la siguiente figura se presenta la distribución de los bits del registro SSPCON, que está ubicado en la dirección 14h y sirve para programar el módulo MSSP en los dos modos que puede trabajar.

Registro SSPCON	7	6	5	4	3	2	1	0
	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SSPM3-SSPM0=Modo de trabajo para el módulo SSP								
SSPM3	SSPM2	SSPM1	SSPM0	Modo de trabajo				
0	0	0	0	Modo maestro SPI, con reloj a $F_{osc}/4$				
0	0	0	1	Modo maestro SPI, con reloj a $F_{osc}/16$				
0	0	1	0	Modo maestro SPI, con reloj a $F_{osc}/64$				
0	0	1	1	Modo maestro SPI, con reloj igual a la salida del TMR2/2				
0	1	0	0	Modo esclavo SPI, con reloj igual a la patita SCK y SS# a nivel bajo				
0	1	0	1	Modo esclavo SPI, con reloj igual a la patita CSK y SS# no activo				
0	1	1	0	Modo esclavo I2C, con dirección de 7 bits				
0	1	1	1	Modo esclavo I2C, con dirección de 10 bits				
1	0	0	0	Modo I2C maestro con reloj $F_{osc}/[4 \cdot (SSPAD+1)]$				
1	0	1	1	Modo I2C maestro controlado por firmware				
1	1	1	0	Modo I2C maestro controlado por firmware, con bits inicio y stop,				
1	1	1	1	activada interrupción, dirección de 7 bits				
1	1	1	1	Modo I2C maestro controlado por firmware, con bits inicio y stop,				
				activada interrupción, dirección de 10 bits				
CKP=Activar el reloj en el modo esclavo, no se usa en el modo maestro								
1=activo								
0=activo								
SSPEN=Bit de activación del módulo SSP								
1=la puerta serie queda configurada con las patitas SCK, SDO, SDA y SS#								
0=las patitas anteriores se utilizan para E/S digitales								
SSPOV=Bit de desbordamiento								
1=Ilega un byte a SSPBUF sin haberse leído el anterior								
0=no hay desbordamiento								
WCOL=Detector de colisiones en escritura								
1=se ha intentado escribir en SSPBUF en condiciones no válidas								
0=no hay colisión								

Figura 7.- Estructura interna del registro SSPCON.

El registro SSIPSR no es direccionable directamente. En los programas hay que acceder a él a través del SSPBUF. En la tabla de la figura se representan los registros relacionados con esta operación.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000000x	0000000u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIIF	TMR2IF	TMR1IF	00000000	00000000
08h	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPIE	TMR2I	TMR1	00000000	00000000

								E	IIE		
13h	SSPBUF	Registro para el sincronismo del puerto serie en recepción y transmisión de datos								xxxxxxxx	uuuuuuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	00000000	00000000
94h	SSPSTAT	SMP	CKE	D/A#	P	S	R/W#	UA	BF	00000000	00000000

2.8.3.- MÓDULO MSSP TRABAJANDO EN MODO I2C.

El protocolo de comunicación serie I2C fue desarrollado por Philips para cubrir sus propias necesidades en la implementación de diversos productos electrónicos que requerían una elevada interconexión de circuitos integrados. El protocolo I2C (“Inter Integrated Circuit”) utiliza únicamente dos líneas para la transferencia de información entre los elementos que se acoplan al bus. Una de dichas líneas, la SDA, es bidireccional y se dedica a soportar los datos. La otra lleva los pulsos de reloj para la sincronización, es unidireccional y recibe el nombre de SCL. Los pulsos de reloj siempre los genera el maestro y tienen la función de sincronizar las transferencias con todos los esclavos colgados a las dos líneas, como se observa en la figura inferior.

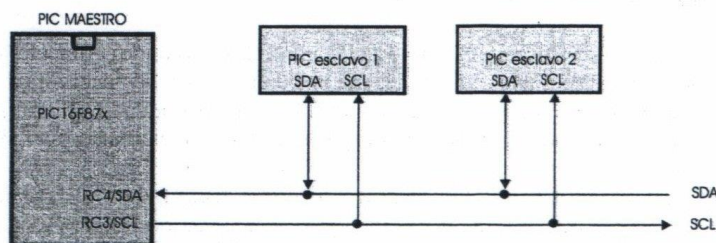


Figura 8.- El maestro se conecta con todos los esclavos en el bus I2C con sólo dos líneas. Por SDA circulan los datos en serie y por SCL el maestro envía los pulsos de reloj que genera.

2.8.4.- CONCEPTOS FUNDAMENTALES SOBRE EL BUS I2C.

En algunas aplicaciones, especialmente las que emplean microcontroladores monochip, se deben establecer ciertos criterios.

- Cierta tipo de microcontroladores no son ampliables desde el exterior, por lo que la aplicación se tiene que ceñir al tipo y cantidad de elementos que contenga en su interior: memoria RAM de datos, EEPROM, líneas de E/S, temporizadores, contadores, etc. Si éstos no son suficientes habrá que elegir otro modelo de microcontrolador con más prestaciones o emplear un sistema ampliable externamente.
- Normalmente, un sistema completo, además del microcontrolador, necesita de otros circuitos periféricos como memorias, ampliación de las líneas de E/S, convertidores A/D y D/A, relojes en tiempo real, controladores de displays, de LCD, etc.
- El coste de la conexión de todos estos periféricos con el microcontrolador debe reducirse al mínimo.

- Normalmente, la aplicación no requiere una gran velocidad de transferencia entre el microcontrolador y los distintos dispositivos.
- La eficacia general del sistema depende de la versatilidad de los dispositivos conectados al mismo.

Estos criterios se pueden implementar con una estructura de bus serie. Este tipo de bus no puede alcanzar velocidades similares a las conseguidas con una estructura de bus paralelo, pero requiere mucho menos cableado y el hardware es mucho más sencillo. Los distintos dispositivos conectados al bus serie deben comunicarse entre sí mediante un protocolo que evite el bloqueo de la información y garantice la comunicación entre todos ellos. Además, el sistema debe ser independiente de los dispositivos conectados al bus con objeto de poder realizar futuras ampliaciones o modificaciones. Así, mediante el protocolo adecuado, se decide qué dispositivo se hace cargo del bus y cuándo. Todos estos conceptos se contemplan en las especificaciones del bus I2C.

Dos líneas, SDA (datos) y SCL (reloj), son las encargadas de transportar la información entre los diferentes dispositivos conectados al bus. Cada dispositivo se identifica por una única dirección y puede transmitir o recibir dependiendo de sus características. Un controlador de LCD, por ejemplo, sólo recibe mientras que una memoria de tipo RAM puede transmitir o recibir datos en función de que se vaya a leer o a escribir.

Los dispositivos pueden clasificarse en maestro (master o principal) o esclavo (slave o secundario). El maestro es el que inicia la transferencia de datos y genera la señal de reloj, mientras que cualquiera de los dispositivos direccionados por un maestro se considera un esclavo.

El I2C es un bus multimaestro por lo que puede haber más de un maestro conectado y controlando el bus. Normalmente se trata de microcontroladores o microcomputadores.

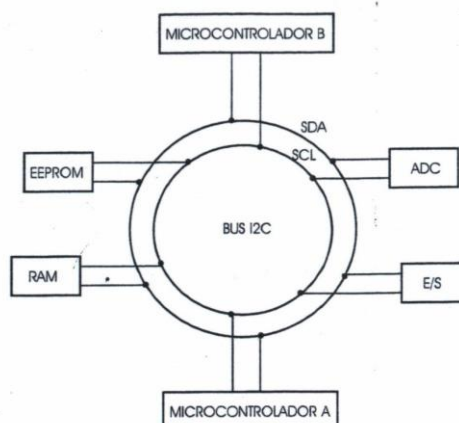


Figura 9.- Conexiones al bus.

Supongamos que en un momento dado el microcontrolador A actúa como maestro:

- 1er caso: envía información al B.
 - A, emisor, direcciona a B.
 - A, emisor, envía el dato a B, receptor.
 - A termina la transferencia.

- 2do caso: recibe información desde B.
 - A, emisor, direcciona B.
 - A, receptor, recibe el dato desde el B, transmisor.
 - A termina la transferencia.

Esta configuración puede no ser permanente por lo que, en otro momento, puede ser el B quien actúe de maestro, todo depende de la dirección del dato a transferir en un momento dado.

Dada la posibilidad de que existieran varios maestros, podría ocurrir que más de uno inicie la transferencia en un mismo instante de tiempo. Para evitar el caos que pueda producir esta situación se ha desarrollado un sistema de arbitraje del bus. El procedimiento consiste en una conexión tipo AND entre todos los dispositivos conectados al bus. Cuando uno o más maestros colocan información en la línea SDA verifican si el bit que ellos sacan coincide con el nivel lógico de dicha línea. Si un maestro saca un 1 pero la línea SDA está a 0 coincidiendo con un pulso de reloj pierde la posesión del bus cancelando desde ese momento la transmisión. El nivel lógico 0 (bit dominante) presente en la línea SDA procederá de un maestro distinto.

Las señales de reloj durante el arbitraje del bus son una combinación entre las señales de clock de los distintos maestros conectadas entre sí a la línea SCL mediante una conexión tipo AND.

La terminología para el bus I2C es la siguiente:

- Receptor: Dispositivo que recibe información por la línea SDA.
- Maestro: Dispositivo que inicia la transferencia, genera la señal de reloj y finaliza la transferencia.
- Esclavo: Dispositivo seleccionado por el maestro.
- Multi-maestro: Cuando más de un maestro puede hacerse con el control del bus sin corromper la información.
- Arbitraje: Procedimiento que garantiza que, si más de un maestro accede simultáneamente al bus, únicamente uno de ellos se hace cargo del mismo con objeto de no alterar la información.
- Sincronización: Procedimiento para sincronizar la señal de reloj en un sistema multi-maestro.

2.8.5.- CARACTERÍSTICAS GENERALES DEL BUS I2C.

Tanto las líneas SDA como SCL son líneas bidireccionales que se conectan a +V_{dd} mediante resistencias de carga pull-up, tal y como se muestra en la figura. Cuando el bus está libre ambas líneas están a nivel lógico 1. Los transistores de salida conectados a las líneas del bus I2C deben ser de colector abierto para que todos ellos se puedan conectar entre sí formando una conexión tipo AND.

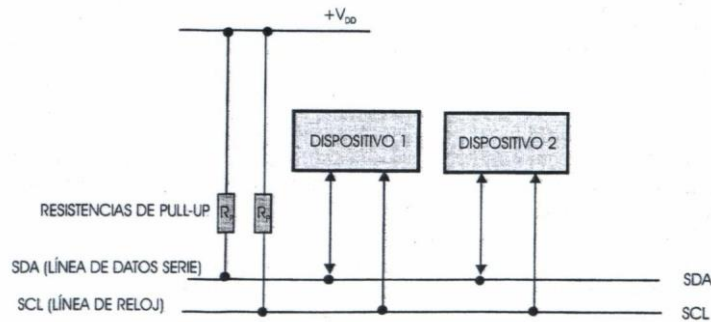


Figura 10.- Conexión de SDA y SCL al bus.

Los bits de datos sobre el bus pueden transferirse a una velocidad de 100 Kbits/s. La capacidad máxima en el bus es de 400 pF y el número de dispositivos conectados no debe superarla.

▪ Transferencia del bit.

Debido a la variedad de tecnologías empleadas en los dispositivos diseñados para conectarse al bus I2C (CMOS, NMOS, TTL, etc.) los niveles lógicos 0 y 1 de los bits transferidos no tienen una tensión fija, sino que dependen de la tensión V_{DD} de alimentación. Cada bit que se trasfiere por la línea SDA debe ir acompañado de un pulso de reloj por la línea SCL.

- Validez del bit

El bit de datos transferido por la línea SDA debe mantenerse estable durante el período en que la señal de reloj está a nivel 1. La línea de datos SDA sólo puede cambiar de estado durante el período en que la señal de reloj esté a 0, tal y como se muestra en la figura.

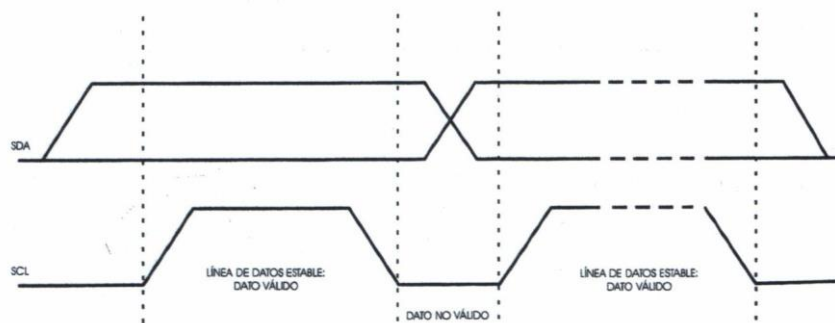


Figura 11.- Validación del bit de datos.

- Condiciones de inicio ("start") y parada ("stop")

Existen dos situaciones únicas que son definidas como las condiciones de inicio (inicio) y parada (parada) y que determinan el inicio y final de toda transferencia de datos entre el maestro y el esclavo.

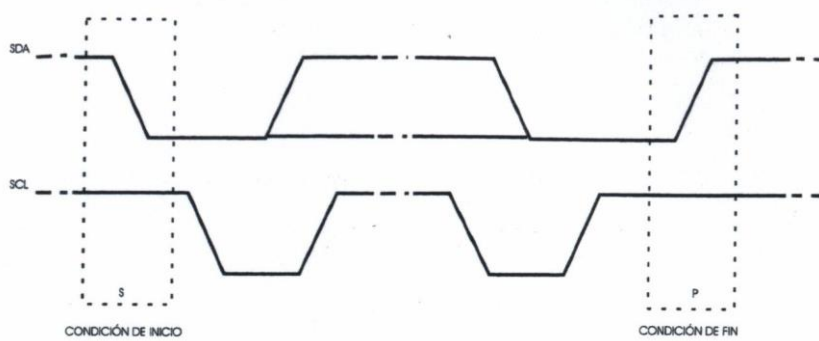


Figura 12.- Condiciones de inicio y parada.

Por un lado, una transición de 1 a 0 (flanco descendente) en la línea SDA al tiempo que SCL está a nivel lógico 1, determina la condición de inicio. Por otro, una transición de 0 a 1 (flanco ascendente) sobre la línea SDA al tiempo que SCL permanece a nivel 1 determina la condición de parada. Ambas condiciones son siempre generadas por el maestro que, en todo momento, inicia y finaliza toda transferencia. El bus se considera ocupado (BUSY) tras la condición de inicio y se considera libre cierto tiempo después de la condición de parada.

- **Transferencia de datos.**

- **Formato del Byte**

Todos los bytes colocados sobre la línea SDA deben constar de 8 bits. Cada byte va seguido de un bit de reconocimiento, ACK, y comienza siempre por el bit de más peso (MSB), tal y como se muestra en la figura.

Entre un byte y el siguiente puede haber un tiempo t de espera que se puede aprovechar para, por ejemplo, dar servicio a una interrupción. En este caso, la línea SCL se mantiene a 0 para forzar al transmisor a un estado de espera.

Un mensaje cualquiera puede finalizar mediante la generación de la condición de parada durante la transferencia de un byte. En este caso no se generará el bit de reconocimiento ACK.

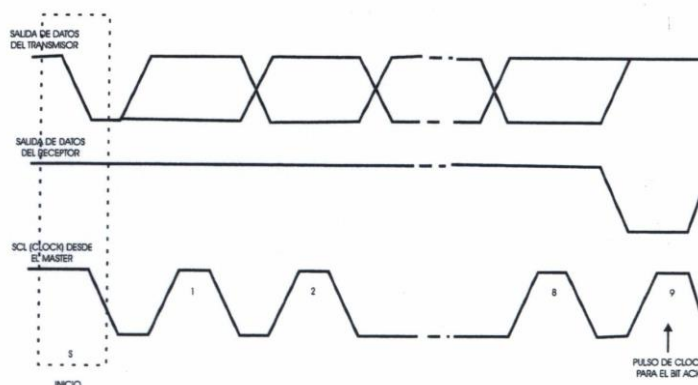


Figura 13.- Formato del byte.

- **Reconocimiento**

El bit de reconocimiento ACK es obligatorio en la transferencia de cada byte. El pulso de reloj asociado a este bit número 9 lo genera el maestro. El transmisor pone la línea SDA a 1 durante dicho pulso de reloj. El receptor, por su parte, pone a 0 la línea SDA durante el

pulso de reloj correspondiente al ACK y lo mantiene estable durante todo el período de dicho pulso. En la línea SDA prevalece por tanto el nivel 0, como se aprecia en la figura.

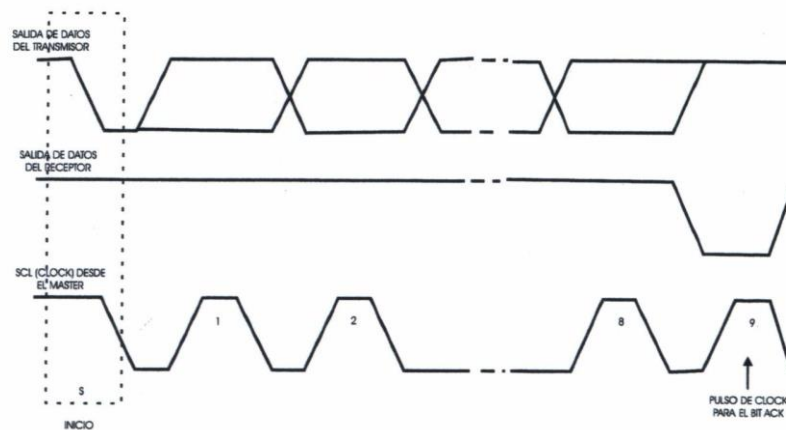


Figura 14.- El reconocimiento.

Normalmente, el receptor que ha sido direccionado está obligado a generar el bit ACK después de recibir cada byte. Cuando un receptor esclavo no genera el bit ACK (NACK) al ser direccionado (por ejemplo, cuando está realizando una función interna y no está dispuesto a recibir), debe mantener la línea SDA a nivel 1 durante el bit ACK (NACK). Esta situación es detectada por el maestro, que debe generar la condición de parada y abortar así la transferencia. Igualmente, si un esclavo receptor reconoce ser direccionado pero no está dispuesto a recibir más bytes el maestro debe abortar la transferencia. Esta situación se identifica porque el esclavo tampoco genera el ACK en el siguiente byte que recibe. En su lugar pone la línea SDA a 1 (NACK) y el maestro genera la condición de parada.

Si un maestro está recibiendo debe generar ACK tras cada byte enviado por el esclavo excepto en el último, en el que el bit ACK lo mantiene a 1 (NACK) y genera la condición de parada.

- **Arbitraje sin sincronización de reloj.**

Es la técnica por la cual se evitan conflictos en el bus y pérdidas de información en un sistema donde existen varios maestros.

- **Sincronización.**

Todos los maestros generan una señal de reloj sobre la línea SCL para transferir mensajes sobre el bus I2C. Los bits de información son válidos cuando dicha señal de reloj está a nivel lógico 1.

La sincronización del reloj se realiza mediante una conexión tipo AND de todos los maestros existentes en el bus.

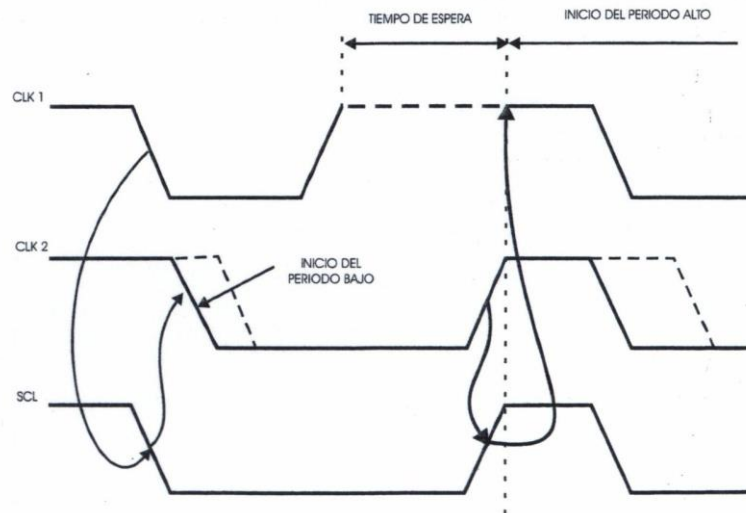


Figura 15.- Sincronización del reloj.

Cuando un maestro cualquiera, por ejemplo el 1, genera un flanco descendente en CLK1 la línea SCL pasa a 0. En este momento todos los dispositivos comienzan a temporizar sus respectivos períodos bajos de reloj. Sin embargo, un flanco ascendente en CLK1 no cambia el estado de SCL, ya que en ese instante CLK2 está a 0. La línea SCL se mantiene a 0 durante el período de aquel dispositivo cuyo nivel 0 sea más largo. El resto de dispositivos entran en estado de espera.

Cuando todos los dispositivos finalizan su período bajo, SCL pasa a 1. De esta forma, todos inician el período alto al mismo tiempo. El primer dispositivo en finalizar este periodo alto pone SCL a 0 y el proceso se repite.

La señal de reloj presente en la línea SCL queda sincronizada y el período bajo se determina por aquel cuyo periodo sea el de mayor duración. El período alto está condicionado por aquel dispositivo cuyo periodo alto sea el de menor duración.

- Arbitraje.

El arbitraje consiste en determinar qué maestro se hace cargo del bus en un sistema multimaestro. Cada maestro debe comparar si el bit de datos que transmite junto con su pulso de reloj en un momento dado coincide con el nivel lógico presente en la línea de datos SDA. De no ser así, pierde el acceso al bus. Si un maestro saca un nivel lógico 1 al tiempo en que otro saca un 0, en la línea SDA prevalece el segundo (bit dominante), por lo que el primero debe dejar el bus libre al menos hasta que se detecte la condición de parada generada por el maestro que se hizo cargo de dicho bus.

La figura inferior muestra el caso de dos maestros. En el momento en que hay diferencia entre el nivel lógico interno del maestro que genera el DATOI y el nivel actual presente en la línea SDA, éste pierde el derecho a usar el bus y su nivel interno pasa a 1, liberando así la línea SDA.

La comparación para determinar el arbitraje puede continuar en varios bits hasta que haya una falta de coincidencia. El maestro que pierde el arbitraje puede seguir generando pulsos de reloj hasta que finalice el byte en que lo perdió.

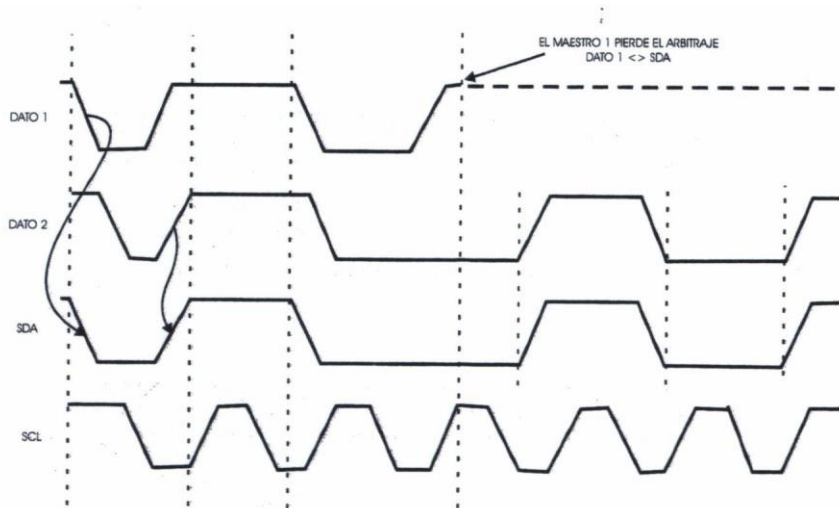


Figura 16.- Arbitraje del bus.

Si un maestro pierde el arbitraje en el momento en que está direccionando a un esclavo es posible que el maestro ganador lo estuviera direccionando precisamente a él. Es por ello que el maestro perdedor se debe poner inmediatamente en modo receptor y como esclavo.

▪ **Formato.**

Los datos que se transfieren tienen el formato mostrado en la figura 61.

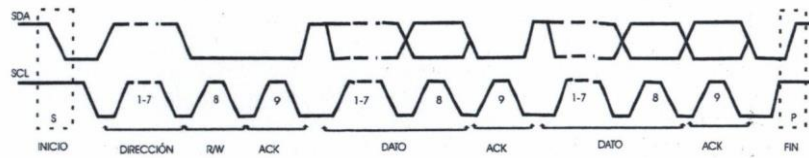


Figura 17.- Formato de los datos transferidos.

Tras la condición de inicio, el maestro envía la dirección del esclavo al que se quiere dirigir. Esta dirección es de 7 bits más el bit R/W# que indica lectura (R/W#=1) o escritura (R/W#=0) del esclavo. Toda transferencia de datos finaliza con la condición de parada, también generada por el maestro. A pesar de todo, si un maestro aún desea comunicar por el bus puede generar otra condición de inicio y direccionar a otro esclavo sin generar previamente la condición de parada.

En las siguientes figuras se muestran posibles formatos de transferencia.

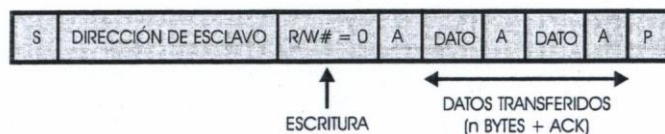


Figura 18.- El maestro transmite al esclavo receptor. No cambia el byte de dirección.

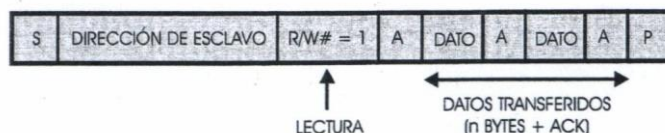


Figura 19.- El maestro recibe desde el esclavo después de enviarle a éste la dirección.

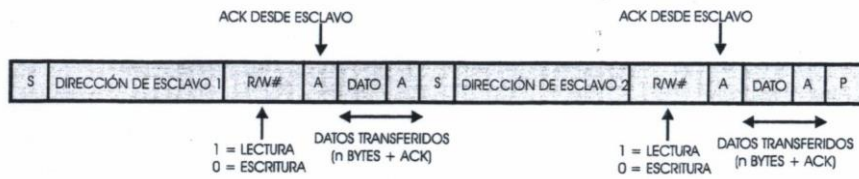


Figura 20.- El maestro cambia el byte de dirección y selecciona un esclavo distinto.

Tras el primer ACK enviado por el esclavo el maestro transmisor se convierte en receptor y el esclavo receptor en transmisor. El formato combinado de la figura 64 permite seleccionar un esclavo para leerlo, por ejemplo, y luego para escribirlo. En el caso de un dispositivo I2C de tipo memoria RAM puede ser útil.

El tipo de datos o comando que se envían pueden servir para realizar distintos tipos de operaciones según el dispositivo I2C empleado consultando las características y posibilidades de cada cual.

2.8.6.- DIRECCIONAMIENTO.

El proceso de direccionamiento del bus I2C se basa en que el primer byte enviado por el maestro tras la condición de inicio es el código que determina y selecciona a un determinado esclavo. Existe una excepción en el código denominada llamada general. Cuando un maestro realiza una llamada general (código 0000 000) todos los esclavos existentes deben responder con el bit ACK. Puede darse el caso de dispositivos que ignoren esta llamada. El siguiente byte tras la misma indicará la operación a realizar.

- **Definición del primer byte**

Los 7 bits de más peso del primer byte se emplean para direccionar a un determinado esclavo. El de menos peso, el octavo enviado, determina si se realizará una operación de lectura o de escritura (R/W#) sobre el esclavo direccionado, tal y como se muestra en la tabla inferior. Cuando el maestro envía una dirección por el bus, todos los esclavos conectados la comparan con la suya. Aquel en que coincida se considerará seleccionado por el maestro.

De los 7 bits que forman una dirección hay una parte que son fijos y ya están definidos internamente por el propio dispositivo esclavo y otra parte que son programables. De esta forma, es posible conectar al mismo bus dispositivos idénticos cuya parte fija de dirección es la misma pero la programable es diferente.

El número de dispositivos iguales que se pueden instalar a un mismo bus depende del número de bits programables disponibles. Por ejemplo, si un determinado modelo de dispositivo recibe una dirección de 7 bits, 4 de los cuales son fijos y los 3 restantes programables, se podrán conectar al bus un máximo de 8 dispositivos de ese mismo modelo.

MSB(mayor peso)		LSB (menor peso)
S	Dirección del esclavo	R/W#

DIRECCIÓN	R/W#	FUNCIÓN
0000 000	0	Dirección de "llamada general" (1)
0000 000	1	Byte de inicio (2)
0000 001	x	Dirección CBUS (3)
0000 010	x	Reservada para un formato diferente (4)
0000 011	x	No definida
0000 100	x	No definida
0000 101	x	No definida
0000 110	x	No definida
0000 111	x	No definida

(1) Para aquellos dispositivos que la admitan y/o necesiten.
(2) No está permitido el bit ACK tras la recepción del byte. Empleado por ciertos dispositivos.
(3) La dirección CBUS esta reservada para permitir la combinación de dispositivos I2C y CBUS en un mismo sistema. Los dispositivos I2C no responden a esta dirección.
(4) Esta dirección está reservada para combinar el formato I2C con otros protocolos. Únicamente los dispositivos I2C que estén diseñados para trabajar con esos formatos y protocolos responden a esta dirección.

Figura 21.- El primer byte

El comité I2C creado al efecto coordina las direcciones asignadas a los diferentes dispositivos. La dirección 1111 111 está reservada como extensión de dirección en el que el proceso de direccionamiento de un dispositivo continúa en los siguientes bytes transferidos por el maestro. La dirección 1111 xxx está reservada también con propósitos de direccionamientos extendidos y la emplean ciertos dispositivos que tienen 10 bits de direccionado. La dirección 0000 xxx está reservada para un grupo especial de dispositivos.

2.8.7.- ESPECIFICACIONES ELÉCTRICAS.

El bus I2C permite la comunicación entre dispositivos fabricados con diferentes tecnologías y diferentes tensiones de alimentación. Para conexiones con niveles de entrada fijos y alimentación de +5 V se definen los siguientes valores:

- $V_{IL}=1,5V$ (máxima tensión de entrada a 0).
- $V_{IH}=3V$ (mínima tensión de entrada a 1).

Para dispositivos capaces de trabajar con un rango variado de tensiones de alimentación (por ejemplo, CMOS) se definen los siguientes niveles:

- $V_{IL}=0,3V_{DD}$ (máxima tensión de entrada a 0).
- $V_{IH}=0,7V_{DD}$ (mínima tensión de entrada a 1).

En ambos casos, la máxima tensión de salida a 0 es:

- $V_{OL}=0,4V_{ARM}$

La corriente máxima de entrada a 0 en las líneas SDA y SCL de un dispositivo compatible con el bus es de $-10\mu\text{A}$. A nivel 1 la corriente de entrada es de $10\mu\text{A}$. La capacidad de dichas líneas es de 10 pF máximo.

Los dispositivos con niveles fijos de entrada pueden alimentarse desde diferentes fuentes de alimentación de $+5\text{V}$, como se muestra en la figura.

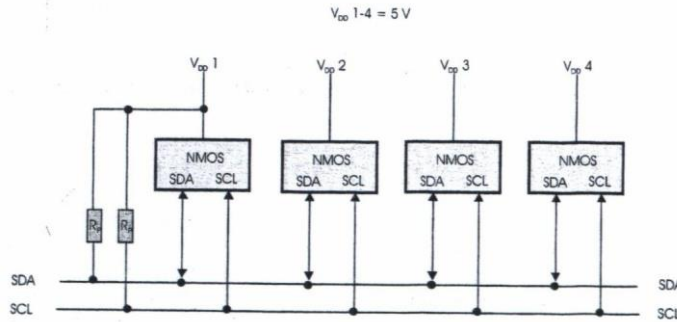


Figura 22.- Alimentación de dispositivos con niveles fijos de entrada.

Los dispositivos I2C cuyos niveles lógicos de entrada están en relación a VDD deben alimentarse conjuntamente según la figura 67.

Cuando se combinan ambos tipos de dispositivos, los de niveles fijos de entrada (NMOS, TTL) y los dependientes de VDD (CMOS), estos últimos deben conectarse a una alimentación común de $+5\text{V}$, como se muestra en la figura.

La capacidad máxima del bus es de 400 pF , que incluye la capacidad entre cables y la de los dispositivos conectados.

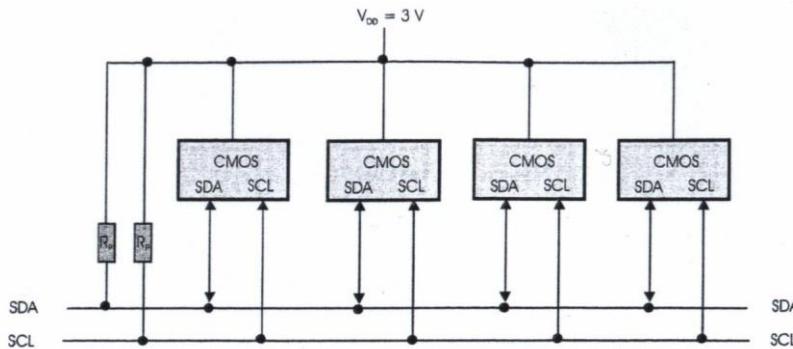


Figura 23.- Alimentación de dispositivos con niveles de entrada dependientes de VDD.

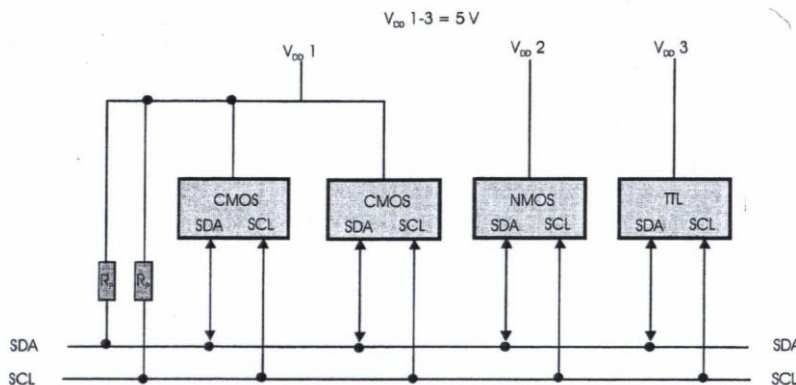


Figura 24.- Alimentación de dispositivos con niveles de entrada fijos junto con dependientes de VDD.

▪ **Tiempos.**

La señal de reloj en el bus I2C tiene un período a 0 de como mínimo 4,7ms y a 1 de como mínimo 4ms. De esta forma, el maestro puede generar una frecuencia de hasta 100 KHz. La tabla de la figura muestra el valor de cada tiempo.

SÍMBOLO	PARÁMETRO	MÍNIMO	MÁXIMO	UNIDAD
TBUF	Tiempo en que el bus debe estar libre antes de iniciar una nueva transmisión	4,7		μs
THD,STA	Tiempo entre el inicio y primer pulso en SCL	4		μs
TLOW	Duración del período bajo del reloj	4,7		μs
THIGH	Duración del período alto del reloj	4		μs
TSU,STA	Tiempo antes de una nueva condición de inicio	4,7		μs
THD,DAT	Tiempo de mantenimiento del dato	0		μs
TSU,DAT	Tiempo de puesta del dato	250		μs
TR	Tiempo del flanco ascendente para las líneas SDA y SCL		1	μs
TF	Tiempo del flanco descendente para las líneas SDA y SCL		300	μs
TSU,STO	Tiempo para la condición de parada	4,7		μs

Figura 25.- Valor y nomenclatura de los diferentes tiempos.

2.8.8.- EL BUS I2C Y LOS PIC16F87X.

En los PIC16F87X el bus I2C está implementado en silicio tanto en el modo maestro (“master”) como en el esclavo (“slave”). En general, el modo maestro es el más usado. En modo maestro es el dispositivo, en este caso el PIC16F87X, el que inicia y finaliza la transferencia y genera los pulsos de reloj. A su vez, selecciona el esclavo al que se destina la información.

Como ya se ha comentado, cada transferencia comienza con la condición de inicio (“start”) y termina con la condición de parada (“stop”). Ambas condiciones las genera el maestro y la primera consiste en un flanco descendente en SDA mientras SCL tiene nivel alto. La condición de parada es un flanco ascendente mientras SCL tiene nivel alto.

Los datos que se colocan sobre la línea SDA son bytes de 8 bits que comienzan con el MSB y terminan con el bit de menos peso, al cual le sigue en el noveno bit la condición de reconocimiento (ACK). Para este caso, el transmisor pone SDA=1 mientras que el receptor pone SDA=0 en dicho pulso de reloj, prevaleciendo sobre la línea el nivel bajo.

El primer byte que envía el maestro tras la condición de inicio contiene la dirección del esclavo con el que se desea realizar la comunicación. El código 0 se usa para realizar una llamada general sobre todos los esclavos. La dirección en realidad consta de 7 bits, estando el octavo destinado a indicar la operación a realizar (R/W#, lectura/escritura). Tras este byte inicial de direccionamiento se manda otro byte que especifica las características de la operación a realizar.

Para activar el bus I2C hay que poner el bit SSPEN=1 que es el bit 5 del registro SSPCON. A partir de ese momento los terminales SDA y SCL quedan configurados para soportar el protocolo I2C. Previamente dichos terminales de la puerta C (RC3/SCL) y RC4/SDA) han de estar configuradas como entradas mediante la escritura de los bits correspondientes del registro TRISC. Hay seis registros para controlar el bus I2C:

- SSPSTAT: registro de estado, explicado en el apartado 8.2.
- SSPCON: registro de control, explicado en el apartado 8.2.
- SSPCON2: registro de control 2, explicado más adelante.
- SSPBUF: buffer para los datos, soporta el dato que se va a transmitir o que se ha recibido.
- SSPSR: registro de desplazamiento no accesible directamente, realiza la conversión serie/paralelo y viceversa.
- SSPADD: registro de dirección, guarda la dirección del esclavo.

Mediante el bit CKE se ponen a nivel alto los terminales SDA y SCL en el modo maestro y esclavo. Cuando CKE=0 los niveles de dichos terminales se configuran según las especificaciones del bus I2C.

Los 4 bits de menos peso del registro SSPCON sirven para seleccionar el modo de trabajo entre estos tres posibles:

- Modo I2C maestro. $\text{Reloj} = (\text{SPADD} + 1) \cdot \text{OSC} / 4$.
- Modo I2C esclavo con 7 bits; para la dirección.
- Modo I2C esclavo con 10 bits para la dirección.

El registro de estado SSPSTAT contiene la información que representa el estado de la transferencia de datos. Detecta las condiciones de inicio y parada, así como la recepción del byte de la dirección.

SSPBUF es el registro que actúa como buffer y soporta el dato que se va a transmitir o que se ha recibido. El registro de desplazamiento SSPSR realiza la conversión serie/paralelo y viceversa junto al SSPBUF. Cuando se ha completado la llegada del byte, que se recibe en el registro SSPSR, se traspassa a SSPBUF y se activa el señalizador SSPIF. Si se recibe otro byte antes de haber leído al SSPBUF, se activa el señalizador de desbordamiento SSPOV (SSPCON, bit 6).

El registro SSPADD guarda la dirección del esclavo. Cuando se emplean direcciones de 10 bits, el usuario debe escribir el byte alto de la dirección con el siguiente código donde A9-A8 son los dos bits de más peso: 1-1-1-1-0-A9-A8-0.

En la siguiente figura se muestra la distribución de los bits del registro SSPSTAT, que ocupa la dirección 94h de los registros específicos de la memoria RAM.

El bit SMP del registro SSPSTAT en el modo maestro es un bit de muestreo que vale 1 cuando los bits de datos se muestrean al final del período, o vale 0 cuando se hace en la mitad del período. El bit CKE selecciona los niveles de los terminales SCL y SDA en el modo maestro o multi-maestro. El bit D/A# (dato/dirección) indica si el dato recibido es de información, 1 o es una dirección, 0. La llegada del bit o condición de stop la detecta el bit P,

que se pone a 1. El bit S detecta la condición de inicio o start. R/W# indica si se trata de una lectura/escritura.

El bit UA cuando vale 1 indica que la dirección es de 10 bits y que hay que cargar el byte alto, según la codificación antes expresada, en el registro SSPADD. Si es 0 la dirección es de 7 bits. Finalmente, el bit BF actúa como señalizador del buffer de datos. Si BF=1 indica que tiene un dato y la transmisión está en progreso sin terminar.

Registro	7	6	5	4	3	2	1	0
SSPSTAT	SMP	CKE	D/A#	P	S	R/W#	UA	BF
	R/W	R/W	R	R	R	R	R	R

BF=Señalizador del buffer de datos 1=tiene un dato y la transmisión está en progreso sin terminar 0=no tiene dato
UA=Señalizador del número de bits 1=la dirección es de 10 bits 0=la dirección es de 7 bits
R/W#=Indica si se trata de una escritura o lectura 1=lectura 0=escritura
S=Detecta la condición de inicio 1=inicio 0=no inicio
P=Detecta la condición de stop o llegada del bit 1=llegada del bit. 0=no llegada de bit
D/A#=Indica si el dato recibido es de información o es una dirección 1=información 0=dirección
CKE=Selección de los niveles de las patitas SCL y SDA 1=flanco descendente 0=flanco ascendente
SMP=Bit de muestreo 1=los bits de datos se muestrean al final del periodo 0=los bits de datos se muestrean en la mitad del período

Figura 26.- Distribución de los bits del registro de estado SSPSTAT. Todos los bits son leíbles, siendo sólo escribibles los dos de más peso. Tras un Reset todos se ponen a 0.

En la tabla siguiente se muestra la estructura del registro de control SSPCON, que ocupa la dirección 14h del mapa de memoria.

El bit WCOL es un detector de colisiones en escritura y cuando vale 1 significa que se ha intentado escribir en SSPBUF en condiciones no válidas para el protocolo.

El bit SSPON si vale 1 indica desbordamiento, o sea, que llega un byte a SSPBUF sin haberse leído el anterior. Si SSPEN=1 la puerta serie queda configurada con los terminales SCL y SDA. Si SSPEN=0 los terminales RC3/SCL y RC4/SDA funcionan como líneas de E/S digitales.

El bit CKP sirve para activar el reloj en el modo esclavo, no usándose en el modo maestro. Mediante los 4 bits de menos peso de SSPCON se selecciona la frecuencia de reloj. Se indican los códigos más interesantes para el modo maestro:

Registro SSPCON	7	6	5	4	3	2	1	0
	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SSPM3-SSPM0=Modo de trabajo para el módulo SSP								
SSPM3	SSPM2	SSPM1	SSPM0	Modo de trabajo				
0	0	0	0	Modo maestro SPI, con reloj a $F_{osc}/4$				
0	0	0	1	Modo maestro SPI, con reloj a $F_{osc}/16$				
0	0	1	0	Modo maestro SPI, con reloj a $F_{osc}/64$				
0	0	1	1	Modo maestro SPI, con reloj igual a la salida del TMR2/2				
0	1	0	0	Modo esclavo SPI, con reloj igual a la patita SCK y SS# a nivel bajo				
0	1	0	1	Modo esclavo SPI, con reloj igual a la patita CSK y SS# no activo				
0	1	1	0	Modo esclavo I2C, con dirección de 7 bits				
0	1	1	1	Modo esclavo I2C, con dirección de 10 bits				
1	0	0	0	Modo I2C maestro con reloj $F_{osc}/[4 \cdot (SSPAD+1)]$				
1	0	1	1	Modo I2C maestro controlado por firmware				
1	1	1	0	Modo I2C maestro controlado por firmware, con bits inicio y stop, activada interrupción, dirección de 7 bits				
1	1	1	1	Modo I2C maestro controlado por firmware, con bits inicio y stop, activada interrupción, dirección de 10 bits				
CKP=Activar el reloj en el modo esclavo, no se usa en el modo maestro								
1=activo								
0=inactivo								
SSPEN=Bit de activación del módulo SSP								
1=la puerta serie queda configurada con las patitas SCK, SDO, SDA y SS#								
0=las patitas anteriores se utilizan para E/S digitales								
SSPOV=Bit de desbordamiento								
1=llega un byte a SSPBUF sin haberse leído el anterior								
0=no hay desbordamiento								
WCOL=Detector de colisiones en escritura								
1=se ha intentado escribir en SSPBUF en condiciones no válidas								
0=no hay colisión								

Figura 27.- Registro de control SSPCON. Todos sus bits son leíbles y escribibles y se ponen a 0 tras un reset.

La siguiente tabla muestra la distribución de los bits del registro de control 2 SSPCON2.

El bit GCEN sólo se usa en el modo esclavo. Cuando se pone el bit ACKSTAT=1 significa que se ha recibido el bit de reconocimiento ACK del esclavo. ACKDT es el bit de reconocimiento en el modo maestro en recepción. Si ACKDT=0 el maestro ha transmitido el bit de reconocimiento, pero si vale 1 no lo ha hecho. Cuando se pone ACKEN=1 se inicia la secuencia de generación de la condición de reconocimiento. Este bit se borra automáticamente por hardware.

Para habilitar el modo de recepción del maestro hay que poner el bit RCEN=1. Para generar la condición de parada en las líneas SCL y SDA hay que poner PEN=1. El bit RSEN cuando se pone a 1 inicia la repetición de la condición de inicio. Finalmente, para iniciar la condición de inicio hay que poner SEN=1.

Registro SSPCON2	7	6	5	4	3	2	1	0
	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SEN=Inicia la condición de inicio 1=iniciada 0=no iniciada								
RSEN=Inicia la repetición de la condición de inicio 1=iniciada 0=no iniciada								
PEN=Genera la condición de parada en las líneas SCL y SDA 1=hay parada 0=no hay parada								
RCEN=Bit que habilita el modo de recepción del maestro 1=habilitado 0=no habilitado								
ACKEN=Selección de la secuencia de generación de la condición de reconocimiento 1=se inicia la secuencia 0=no se inicia la secuencia								
ACKDT=Bit de reconocimiento del modo maestro 1=el maestro no ha transmitido el bit de reconocimiento 0=si lo ha transmitido								
ACKSTAT=Bit de reconocimiento del modo esclavo 1=se ha recibido el bit de reconocimiento 0=no se ha recibido								
GCEN=Sólo se usa para el modo esclavo								

Figura 28.- Registro de control 2 SSPCON2. Todos sus bits son leíbles y escribibles y quedan a 0 tras un reset.

En la tabla se observan todos los registros que participan en la programación del bus I2C.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000000x	0000000u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPI1F	TMR2IF	TMR1IF	00000000	00000000
0Dh	PIR2	---	0	---	EEIF	BCLIF	---	---	CCP2IF	-----0	-----0
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPI1E	TMR2IE	TMR1IE	00000000	00000000
8Dh	PIE2	---	0	---	EEIE	BCLIE	---	---	CCP2IE	-----0	-----0

0Eh	SSPBUF	Registro de recepción/transmisión de datos en puerto serie síncrono								xxxxxxxx	uuuuuuuu
0Fh	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	00000000	00000000
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	00000000	00000000
93h	SSPADDD	Registro direcciones del esclavo/registro para el maestro								00000000	00000000
94h	SSPSTAT	SMP	CKE	D/A#	P	S	R/W#	UA	BF	00000000	00000000

Figura 29.- Tabla que recoge los principales registros que intervienen en el control del módulo MSSP con el protocolo bus I2C.

2.8.9.- FUNCIONAMIENTO DEL MODO MAESTRO.

El trabajo en el modo maestro se pone en marcha escribiendo los valores adecuados en los bits del registro de control SSPCON y SSPEN=1. Una vez puesto en funcionamiento este modo el usuario tiene seis opciones:

- Generar la condición de inicio.
- Generar la condición de repetición de inicio.
- Escribir sobre SSPBUF para iniciar la transmisión de un dato o una dirección.
- Generar la condición de parada.
- Generar el modo I2C en recepción.
- Generar la condición de reconocimiento al final de la recepción de un byte de datos.

El maestro es el encargado de generar los pulsos de reloj y las condiciones de inicio y parada en cada byte transferido. Cuando el maestro es receptor, el primer byte indica la dirección del esclavo (7 bits más el bit R/W#). Por cada byte recibido por el maestro se transmite un bit de reconocimiento ACK.

A continuación se describe la secuencia de pasos que hay que realizar en una transmisión del maestro:

1. El usuario genera la condición de inicio, poniendo SEN=1 en el registro SSPCON2.
2. Se pone SSPIF=1. El módulo esperará a que finalice la operación de inicio antes de empezar otra.
3. Se carga en SSPBUF la dirección del esclavo a transmitir.
4. Se saca por SDA la dirección del esclavo.
5. El esclavo genera el bit de reconocimiento ACK y en el maestro se escribe en SSPCON2 sobre el bit 6.
6. Se genera una interrupción al final del noveno pulso de reloj y SSPIF=1.
7. El usuario carga en SSPBUF un dato de 8 bits.
8. Se transmiten los 8 bits por SDA.

9. Se recoge el bit de reconocimiento del esclavo y se escribe el bit ACKSTAT del SSPCON2.
10. El módulo genera una interrupción en el noveno pulso de reloj y pone SSPIF=1.
11. Se genera la condición de parada con PEN=1 en SSPCON2.
12. Una vez completada la condición de parada, se genera una interrupción.

En el modo maestro el valor usado en la generación de la frecuencia se escribe en SSPADD, según los 4 bits de menos peso. Después, este valor se carga en el contador descendente BRG, que lo va decrementando hasta llegar a 0, donde se para hasta que se vuelve a recargar. El BRG se decrementa 2 unidades por cada ciclo de instrucción.

- **Condición de inicio (start).**

Para generar la condición de start, el usuario debe escribir el bit SEN=1 en SSPCON2. Si SDA y SCL se mantienen a nivel alto, el contador BRG se carga con el valor de 7 bits de SSPADD. Cuando termina el conteo de BRG la línea SDA pasa a nivel bajo. La condición de inicio consiste en un flanco descendente de SDA mientras SCL permanece con nivel alto. Pone el bit S=1 del registro SSPSTAT. Luego se recarga el BRG con el valor SSPADD (6...0) y vuelve a decrementarse. Al llegar a 0 el contador el bit SEN pasa a 0 automáticamente.

- **Condición de repetición de inicio (start).**

Esta situación se produce cuando RSEN=1. Entonces SCL pasa a nivel bajo y el contador BRG se carga con los 7 bits de menos peso de SSPADD y comienza el conteo. Al finalizar el conteo SCL pasa de nivel bajo a nivel alto mientras SDA sigue en nivel bajo. Luego nuevamente SCL=1 y SDA=1 y se vuelve a cargar BRG.

- **Modo maestro en transmisión.**

La transmisión de un byte de datos debe ir precedida de la transmisión de la dirección del esclavo, la cual debe cargarse en el registro SSPBUF, en cuyo momento se pone el bit BF=1 y se pone en marcha el reloj y el inicio de la transmisión. Cada bit que sale por la línea SDA lo hace en el flanco descendente del pulso de reloj en SCL. Tras sacar los 8 bits se borra el bit BF y el esclavo debe generar la condición de reconocimiento en el noveno pulso de reloj. El bit ACK se coloca en el bit ACKDT en el flanco descendente de SCL. Tras el noveno pulso SSPIF=1 y se detiene el generador de frecuencia hasta que no se cargue otro dato en SSPBUF.

En transmisión BF=1 cuando la CPU escribe en el SSPBUF y BF pasa a 0 cuando han salido los 8 bits. El señalizador WCOL vale 1 cuando el usuario escribe sobre el SSPBUF estando una transmisión del dato previo en progreso. Este bit hay que pasarlo a 0 por software. El bit ACLSTAT pasa a 0 cuando el esclavo ha enviado el bit de reconocimiento.

- **Modo maestro en recepción.**

Para poner en marcha el modo maestro en recepción, hay que poner el bit RCEN=1 estando en reposo el módulo MSSP. El generador de frecuencia empieza a funcionar y SCL cambia de estado en cada período controlado por el decremento del contador BRG. Entonces el dato empieza a desplazarse sobre el registro SSPSR. Al final del octavo pulso de reloj se ponen BF=SSPIF=1 y se transfiere el contenido de SSPSR al SSPBUF, con lo que RCEN=0. El módulo queda en estado de reposo a la espera de un nuevo comando. Cuando

el buffer es leído por la CPU, BF se pone a 0 automáticamente. El usuario puede enviar un bit de reconocimiento al finalizar la recepción, escribiendo ACKEN=1, SSPCON2 bit 4.

Cuando un dato o una dirección se carga en SSPBUF desde SSPSR, el bit BF=1 y este bit pasa a 0 cuando la CPU lee SSPBUF. El señalizador de desbordamiento SSPOV es 1 cuando se recibe un byte en SSPSR y aún no ha leído la CPU el dato anterior.

Si el usuario escribe en SSPBUF mientras hay una recepción en progreso, el bit WCOL se pone a 1 y el contenido del buffer no cambia.

- **Secuencia de reconocimiento.**

Se produce automáticamente en cuanto se pone ACKEN=1. Entonces el terminal SCL pasa a nivel bajo y aparece en SDA el bit de reconocimiento al finalizar el dato. Al poner ACKEN=1 y ACKDT=0 el terminal SCL queda a nivel bajo, mientras el generador de frecuencia cuenta un período (TBRG). Luego SCL pasa a nivel alto y al completarse otro período SCL vuelve a pasar a nivel bajo y ACKEN=0. El flag SSPIF=1 cuando finaliza la secuencia de reconocimiento.

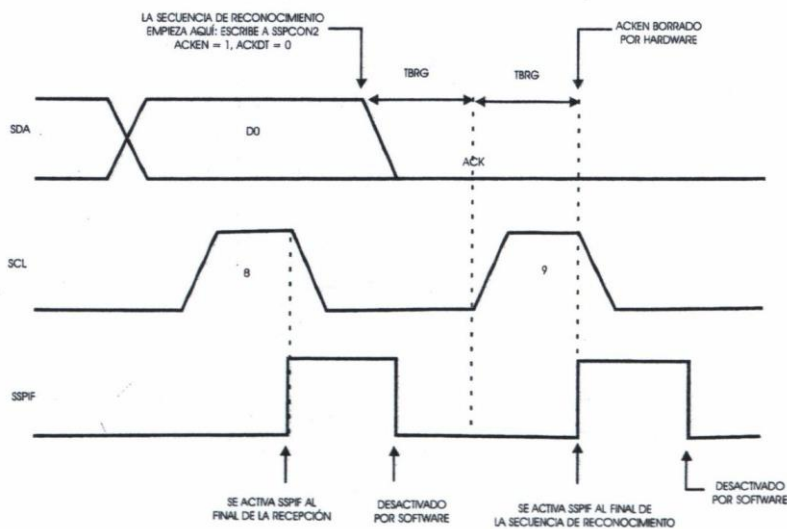


Figura 30.- Diagrama de tiempos de una secuencia de reconocimiento.

- **Condición de parada (stop).**

Se genera la condición o bit de stop al final de la transmisión/recepción, poniendo el bit PEN=1, registro SSPCON2 bit 2. En el flanco descendente del noveno pulso de reloj la línea SCL queda a nivel bajo. Entonces si PEN=1 la línea SDA pasa a nivel bajo y el generador cuenta un período, tras el cual SCL pasa a nivel alto. Tras otro período TBRG, la línea SDA pasa a nivel alto y P=1. Tras otro período PEN, se pone a 0 automáticamente, figura 75.

El pulso de reloj en SCL se produce cuando es muestreada dicha línea cada ciclo T_{osc} 4, y está a nivel alto, en cuyo momento se carga el contador BRG con los 7 bits de menos peso de SSPADD y comienza a contar otro período, pasando SCL a 0 al completarlo.

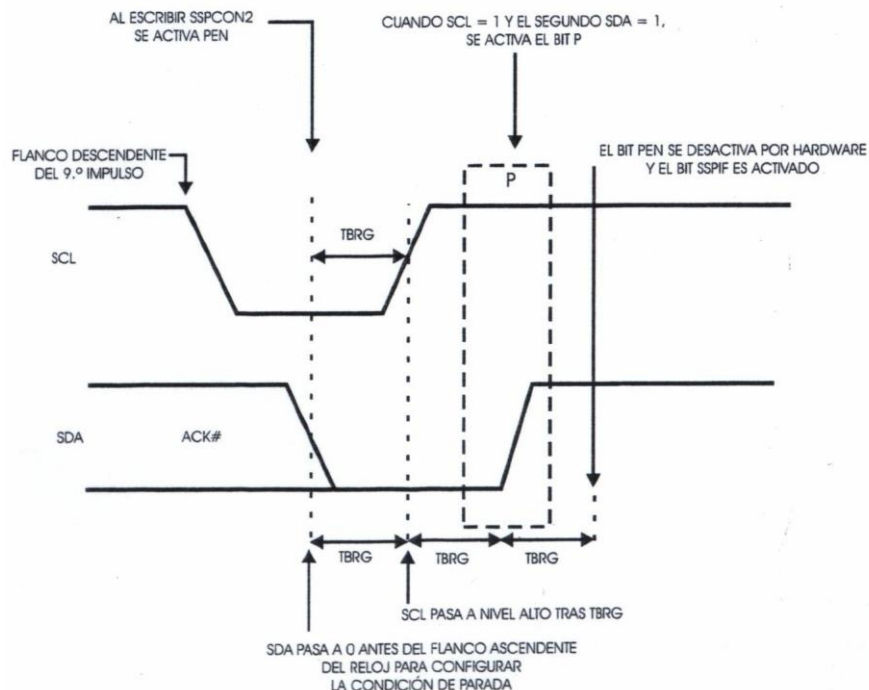


Figura 31.- Diagrama de tiempos de la condición de parada.

2.9.- EL USART: TRANSMISOR/ RECEPTOR, SÍNCRONO/ ASÍNCRONO SERIE

El USART, llamado SCI (“Serial Communications Interface”) puede funcionar como un sistema de comunicación full dúplex (los datos circulan simultáneamente en ambas direcciones) o bidireccional asíncrono, adaptándose a multitud de periféricos y dispositivos que transfieren información de esta forma, tales como el ordenador PC. También puede trabajar en modo síncrono unidireccional o half duplex para soportar periféricos como memorias, conversores, etc. En resumen, el USART puede trabajar de tres maneras:

- Asíncrona, full duplex bidireccional.
- Síncrona maestro, half duplex unidireccional.
- Síncrona esclavo, half duplex unidireccional.

En la siguiente figura se muestra un esquema del comportamiento del USART en modo asíncrono y síncrono. En el primero, las transferencias de información se realizan sobre dos líneas TX (transmisión) y RX (recepción), saliendo y entrando los bits por dichas líneas al ritmo de una frecuencia controlada internamente por el USART. En el modo síncrono, la comunicación se realiza sobre dos líneas, la DT que traslada en los dos sentidos los bits a la frecuencia de los pulsos de reloj que salen por la línea CK desde el maestro. En ambos modos las líneas de comunicación son las dos de más peso de la puerta C: RC6/TX/CK y RC7/RX/DT.

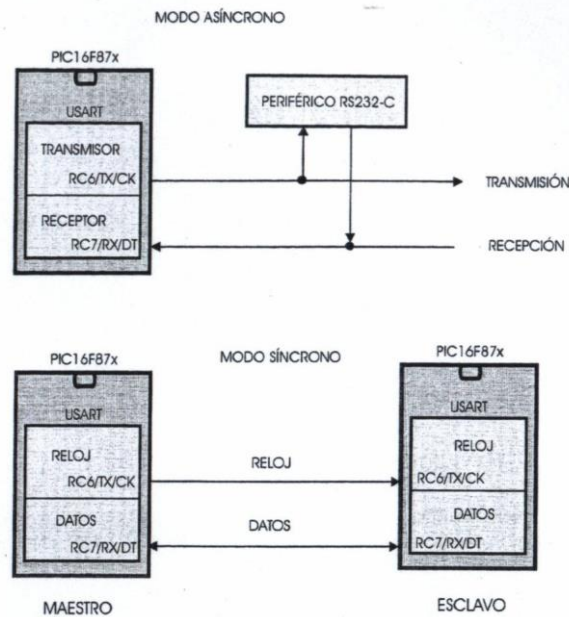


Figura 32.- En el modo asíncrono, la comunicación serie del USART en los PIC16F87X está soportada por las líneas RC6/TX/CK y RC7/RX/DT por las que se mueven los bits a la frecuencia interna de reloj. En el modo síncrono, los bits de información circulan en ambos sentidos por la línea DT a la frecuencia de los pulsos que genera el maestro por la línea CK.

Dada la importancia del protocolo asíncrono del USART, según la popular norma RS-232-C, se dedica este apartado a estudiar este modo de comunicación. En esta forma de comunicación serie, se usa la norma RS-232-C, donde cada palabra de información o dato se envía independientemente de los demás.

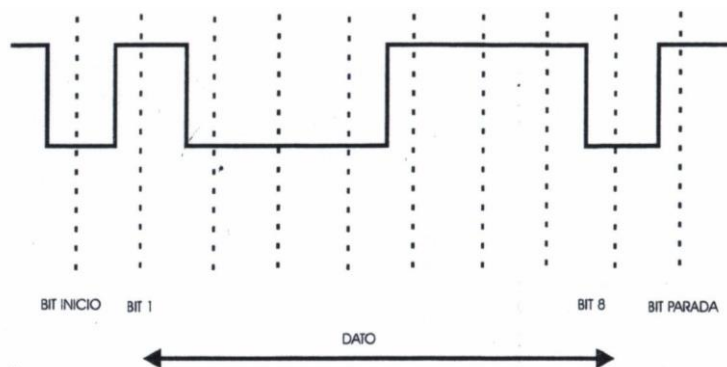


Figura 33.- En el protocolo de comunicación asíncrono, cada palabra de información o dato consta de 8 ó 9 bits, estando precedidos por un bit de INICIO y les sigue un bit de PARADA. Los bits se transfieren al ritmo de la frecuencia de reloj establecida.

Suele constar de 8 o 9 bits y van precedidos por un bit de START (inicio) y detrás de ellos se coloca un bit de STOP (parada), de acuerdo con las normas del formato estándar NRZ (“NonReturn-to-Zero”). Los bits se transfieren a una frecuencia fija y normalizada.

Los cuatro bloques que configuran la arquitectura del USART, en modo asíncrono, son:

- Circuito de muestreo.
- Generador de baudios.

- Transmisor asíncrono.
- Receptor asíncrono.

El circuito de muestreo actúa sobre el terminal RC7/RX/DT, que es por donde se recibe el bit de información o control y se encarga de muestrear tres veces su valor, para decidir éste por mayoría.

2.9.1.- GENERADOR DE BAUDIOS.

En el protocolo asíncrono RS-232-C la frecuencia en baudios (bits por segundo) a la que se realiza la transferencia se debe efectuar a un valor normalizado: 330, 600, 1200, 4800, 9600, 19200, 38400, etc. Para generar esta frecuencia el USART dispone de un generador de frecuencia en baudios, BRG, cuyo valor es controlado por el contenido grabado en el registro SPBRG.

Además del valor X cargado en el registro SPBRG, la frecuencia en baudios del generador depende del bit BRGH del registro TXSTA, bit 2, que se explicará en el apartado siguiente. En el caso de que BRGH sea 0 se trabaja en baja velocidad y si BRGH es 1 se trabaja en alta velocidad. El valor de este bit gobernará el valor de una constante K necesaria en la determinación de la frecuencia de funcionamiento.

$$\text{Frecuencia en Baudios} = \text{FOSC} / [k \cdot (X + 1)]$$

X: es el valor cargado en el registro SPBRG

Si BRGH=0, baja velocidad y k = 64

Si BRGH=1, alta velocidad y k = 16

De donde se desprende que: $X = \text{FOSC} / \text{Frecuencia} / K - 1$.

A modo de ejemplo si en un PIC16F873 funcionando con una FO SC=16MHz se desea trabajar con el USART en modo asíncrono y baja velocidad, con una frecuencia 9600 baudios, calcular el valor X que hay que cargar en el registro SPBRG y el error que se origina.

Baja velocidad K=64

$$\text{Frecuencia} = \text{FO SC} / [64 \cdot (X + 1)]$$

$$9600 = 16000000 / [64 \cdot (X + 1)]$$

$$X = 25$$

Si se carga el registro SPBRG con 25 la frecuencia real de trabajo será:

$$\text{Frecuencia} = 16000000 / [64 \cdot (25 + 1)] = 9615 \text{ baudios.}$$

El error que se origina cargando 25 en el SPBRG será:

$$\text{Error} = (9615 - 9600) / 9600 = 0,16\%$$

En la siguiente tabla se presenta la estructura interna junto a otros datos de interés de los registros que intervienen en la determinación de la frecuencia en baudios a la que transfiere información el USART.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
98h	TXSTA	CSR C	TX9	TXEN	SYNC	----	BRGH	TRMT	TX9D	0000-010	0000-010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000000x	0000000x
99h	SPBRG	Registro generador de baudios								00000000	00000000

Figura 34.- Características relevantes de los registros que intervienen en la determinación de la frecuencia en baudios de la transferencia de información en el modo asíncrono del USART.

Baudios (KB)	Fosc=20MHz			Fosc=16MHz			Fosc=10MHz		
	Kbaud.	% error	SPBRG Decima I	Kbaud.	% error	SPBRG Decima I	Kbaud.	% error	SPBRG Decima I
0,3	-	-	-	-	-	-	-	-	-
1,2	1,221	1,75	255	1,202	0,17	207	1,202	0,17	129
2,4	2,404	0,17	129	2,404	0,17	103	2,404	0,17	64
9,6	9,766	1,73	31	9,615	0,16	25	9,766	1,73	15
19,2	19,531	1,72	15	19,231	0,16	12	19,531	1,72	7
28,8	31,250	8,51	9	27,778	3,55	8	31,250	8,51	4
33,6	34,722	3,34	8	35,714	6,29	6	31,250	6,99	4
57,6	62,500	8,51	4	62,500	8,51	3	52,083	9,58	2

Baudios (KB)	Fosc=4MHz			Fosc=3,6864kHz		
	Kbaud.	% error	SPBRG Decima I	Kbaud.	% error	SPBRG Decima I
0,3	0,300	0	207	0,3	0	191
1,2	1,221	1,75	51	1,2	0	47
2,4	2,404	0,17	25	2,4	0	23
9,6	8,929	6,99	6	9,6	0	5
19,2	20,833	8,51	2	19,2	0	2
28,8	31,250	8,51	1	28,8	0	1
33,6	-	-	-	33,6	-	-
57,6	62,500	8,51	0	57,6	0	0

Valores para el registro SPBRG en función de la frecuencia y del valor puesto en el bit BRGH del registro TXSTA.

Valor BRGH=0

Baudios (KB)	Fosc=20MHz			Fosc=16MHz			Fosc=10MHz		
	Kbaud.	% error	SPBRG Decima I	Kbaud.	% error	SPBRG Decima I	Kbaud.	% error	SPBRG Decima I
0,3	-	-	-	-	-	-	-	-	-
1,2	-	-	-	-	-	-	-	-	-
2,4	-	-	-	-	-	-	2,441	1,71	255

9,6	9,615	0,16	129	9,615	0,16	103	9,615	0,16	64
19,2	19,231	0,16	64	19,231	0,16	51	19,531	1,72	31
28,8	29,070	0,94	42	29,412	2,13	33	28,409	1,36	21
33,6	33,784	0,55	36	33,333	0,79	29	32,895	2,10	18
57,6	59,524	3,34	20	58,824	2,13	16	56,818	1,36	10

Baudios (KB)	Fosc=4MHz			Fosc=3,6864kHz		
	Kbaud.	% error	SPBRG Decima I	Kbaud.	% error	SPBRG Decima I
0,3	-	-	-	-	-	-
1,2	1,202	0,17	207	1,2	0	191
2,4	2,404	0,17	103	2,4	0	95
9,6	9,615	0,16	25	9,6	0	23
19,2	19,231	0,16	12	19,2	0	11
28,8	27,798	3,55	8	28,8	0	7
33,6	35,714	6,29	6	32,9	2,04	6
57,6	62,500	8,51	3	57,6	0	3

Valores para el registro SPBRG en función de la frecuencia y del valor puesto en el bit BRGH del registro TXSTA.

Valor BRGH=1

2.9.2.- TRANSMISIÓN ASÍNCRONA.

En la figura inferior se muestra un esquema simplificado de la comunicación entre un USART emisor y otro receptor, contenidos en sus respectivos microcontroladores.

El dato que se desea transmitir por el USART transmisor se deposita en el registro TXREG y a continuación se traspasa al registro de desplazamiento TSR, que va sacando los bits secuencialmente y a la frecuencia establecida. Además, incluye un bit de INICIO antes de los bits del dato de información y después de sacar todos los bits añade un bit de PARADA. El USART receptor recibe uno a uno los bits, elimina los dos de control y los de información una vez que han llenado el registro de desplazamiento RSR los traslada automáticamente al registro RCREG, donde quedan disponibles para su posterior procesamiento.

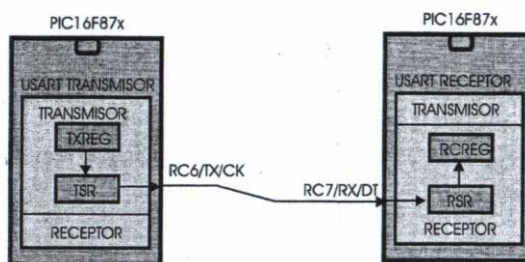


Figura 35.- Esquema simplificado del contenido entre dos módulos USART funcionando en modo asíncrono.

En la figura se presenta el diagrama por bloques de la sección transmisora del USART. El núcleo está constituido por el registro de desplazamiento TSR, que obtiene el dato desde el registro TXREG y luego lo va desplazando y sacando bit a bit, en serie, por el terminal RC6/TX/CK. El primer bit que sale es el de menos peso. El dato a transferir se carga por software en TXREG y se transfiere al TSR en cuanto se haya transmitido el bit de PARADA

del dato anterior. La transferencia entre los dos registros se realiza en un ciclo y entonces el señalizador TXIF se pone a 1, para advertir que el registro de transmisión se ha vaciado. En este momento puede producirse una interrupción si se ha habilitado programando el bit TXIE=1 en el registro PIE1 (bit 4). Cuando se escribe otro dato sobre TXREG, el señalizador TXIF se pone a 0. El bit TRMT sirve para indicar el estado del registro TSR, vale 1 cuando está vacío.

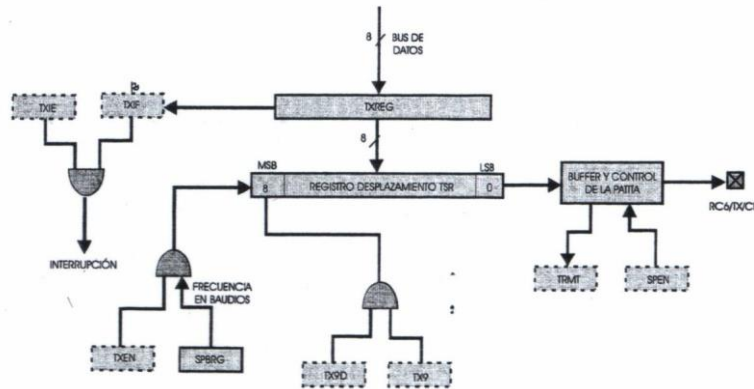


Figura 36.- Diagrama de bloques de la sección de transmisión del USART en modo asíncrono.

La secuencia de pasos a seguir para implementar una transmisión en el USART es la siguiente:

- Hay que configurar las líneas RC6/TX/CK como salida y RC7/RX/DT como entrada.
- Poner SYNC=0 y SOEN=1 para activar el USART en modo asíncrono.
- Si se desea trabajar con interrupción, poner TXIE=1, además de habilitar las interrupciones en general.
- Si el dato consta de 9 bits, en lugar de los 8 típicos, poner el bit TX9=1. El noveno bit se colocará en TX9D (TXSTA).
- Se carga el valor X adecuado en el registro SPBRG, para producir la frecuencia de trabajo deseada. Hay que controlar el bit BRGH (alta y baja velocidad).
- Activar la transmisión con TXEN=1. El bit TXIF tendrá valor 1, ya que TXREG se encuentra vacío.
- Cargar en TXREG el dato a transmitir. Comienza la transmisión.

En las figuras siguientes se especifica la distribución y asignación de funciones de los bits del registro TXSTA y se resumen las características más relevantes de los registros asociados con la transmisión asíncrona.

Registro TXSTA	7	6	5	4	3	2	1	0
	CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D
	R/W	R/W	R/W	R/W	0	R/W	R/W	R/W

TX9D=Bit 9 del dato a transmitir
TRMT=Bit de estado del registro de desplazamiento de transmisión
1=TSR vacío
0=TSR no vacío

BRGH=Bit de selección de la velocidad de baudios Modo síncrono: no utilizado 1=Alta velocidad 0=Baja velocidad	Modo asíncrono:
SYNC=Bit de selección del modo del USART 1=Modo síncrono 0=Modo asíncrono	
TXEN=Activa la transmisión 1=Transmisión activada 0=Transmisión desactivada	
TX9=Habilita el bit 9 de transmisión 1=Selecciona transmisión de 9 bits 0=Selecciona transmisión de 8 bits	
CSRC=Bit de selección del reloj Modo síncrono: no influye 1=Modo maestro (reloj generado internamente desde BRG) 0=Modo esclavo (reloj generado por una fuente externa)	Modo asíncrono:

Figura 37.- Asignación de funciones de los bits del registro TXSTA para el control de la sección transmisora del USART.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPIE	TMR2IE	TMR1IE	00000000	00000000
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIF	TMR2IF	TMR1IF	00000000	00000000
98h	TXSTA	CSR C	TX9	TXE N	SYNC	----	BRGH	TRMT	TX9D	0000-010	0000-010
19h	TXREG	Registro de transmisión del USART								00000000	00000000
18h	RCSTA	SPEN	RX9	SRE N	CREN	ADDEN	FERR	OERR	RX9D	0000000x	0000000x
99h	SPBRG	Registro generador de baudios								00000000	00000000

Figura 38.- Características más relevantes de los registros asociados con la transmisión asíncrona.

2.9.3.- RECEPTOR ASÍNCRONO.

Los datos se reciben en serie, bit a bit, por el terminal RC7/RX/DT y se van introduciendo secuencialmente en el registro desplazamiento RSR, que funciona a una frecuencia 16 veces más rápida que la de trabajo. Cuando el dato consta de 9 bits hay que programar el bit RX9=1 y el noveno bit de información se colocará en el bit RX9D del registro RCSTA. Obsérvese en la figura el control sobre el noveno bit con las puertas de control y las señales que se aplican ADDEN=1. Cuando CREN=1 en el registro RCSTA (bit 4) se habilita la recepción.

Cuando un procesador maestro intenta enviar información a uno de los esclavos, envía primero un byte de dirección que identifica al destinatario. El byte de dirección se identifica porque el bit RX9D que llega vale 1. Si el bit ADDEN=1 en el esclavo se ignoran todos los

- Si se ha direccionado el dispositivo, poner ADDEN=0 para permitir la recepción de la información.

En la figura se muestra la estructura interna y la asignación de funciones de los bits del registro RCSTA.

Registro RCSTA	7	6	5	4	3	2	1	0
	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RX9D = Bit 9 del dato recibido, puede ser el bit de paridad								
OERR = Bit de error de sobrepasamiento 1=error de sobrepasamiento (puede ser borrado escribiendo un 0 en el bit CREN) 0=no hay error de sobrepasamiento								
FERR=Bit de error de trama 1=error de trama (puede ser actualizado leyendo el registro RCREG y recibir el siguiente dato válido. 0=no hay error de trama								
ADDEN=Detección de dirección Modo asíncrono con 9 bits (RX9=1): 1=activa la detección de dirección, activa la interrupción y descarga el buffer de recepción al activarse RSR, bit 8 0=desactiva la detección de dirección, todos los bits son recibidos y el bit 9 puede ser utilizado como bit de paridad								
CREN = Configura al recepción continua Modo asíncrono: 1=habilita modo de recepción continua 0=deshabilita recepción continua Modo síncrono: 1=habilita recepción continua hasta que el bit CREN es borrado 0=deshabilita recepción continua								
SREN=Configura la recepción sencilla Modo asíncrono: no influye 1=Habilita recepción sencilla 0=Deshabilita recepción sencilla Modo síncrono esclavo: no se utiliza Modo síncrono maestro:								
RX9=Habilita el bit 9 de recepción 1=selecciona recepción de 9 bits 0=selecciona recepción de 8 bits								
SPEN=Habilitación del puerto serie 1=puerto serie habilitado, se configuran los terminales RC7/RX/DT y RC6/TX/CK 0=puerto serie deshabilitado								

Figura 40.- Registro RCSTA para el control de la sección receptora del USART, en modo asíncrono.

En la tabla se muestran las características más relevantes de los registros que intervienen en la recepción asíncrona de datos en el USART.

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Valor en POR. BOR	Valor en el resto de resets
8Ch	PIEI	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPIIE	TMR2IE	TMR1IE	00000000	00000000
0Ch	PIRI	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIIF	TMR2IF	TMR1IF	00000000	00000000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	----	BRGH	TRMT	TX9D	0000-010	0000-010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000000x	0000000x
1Ah	RCREG	Registro de recepción del USART								00000000	00000000
99h	SPBRG	Registro generador de baudios								00000000	00000000

Figura 41.- Características de interés de los registros que intervienen en la sección receptora del USART en modo asíncrono.

2.10.- EL MICROCONTROLADOR PIC16F877A DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

El microcontrolador PIC16F877A es el cerebro central del sistema de control diseñado. Todas las señales de control del sistema llegan a él de un modo u otro y dependiendo del tipo de señal se obtendrá una variable de salida determinada.

En el marco del sistema de control diseñado son fundamentales las posibles fuentes de interrupción que posee el microcontrolador. En primer lugar, la interrupción por la línea RB0 (cuyas funciones se ven tremendamente ampliadas mediante la utilización de los circuitos gestores de interrupciones 8259, ver sección III de este capítulo) establece la conexión entre el microcontrolador y las señales de control provenientes de los sensores finales de carrera y de los sensores inductivos ubicados en la muñeca del robot (ver Capítulo II Sección III). En segundo lugar, la interrupción por comunicación por el puerto serie es fundamental puesto que la unión de los subsistemas de control de posición y velocidad que hacen parte del sistema de control diseñado (ver Capítulo V). Esta unión se realiza mediante la interacción de los subsistemas de control con la interfaz gráfica y está basada en la comunicación serie entre el ordenador y el microcontrolador que se explica en detalle en el Capítulo V del presente trabajo.

Además de las interrupciones cabe destacar el sistema de comunicación de información implementado en la placa de control principal. La comunicación entre el microcontrolador y los demás circuitos se realiza mediante un bus de datos de ocho bits implementado en la puerta D del microcontrolador y una serie de básculas 74LS373. De esta forma se consigue una comunicación dinámica en la que el microcontrolador solo se ocupa los instantes requeridos por cada circuito y queda libre para actuar en todos los otros

instantes. Este sistema de comunicación se complementa con los circuitos 8255 ubicados en la placa auxiliar 8255 – 8253 (ver la última sección de este capítulo) que permiten aumentar el número de puertos de salida del microcontrolador (hasta un máximo de tres puertos de salida extra por cada 8255). A través de estos circuitos el microcontrolador puede comunicarse con la placa de gestión de motores como con los circuitos contadores, relaciones de control que se explican en detalle en el Capítulo V del presente trabajo.

En conclusión, el microcontrolador sí que es el centro de control del sistema diseñado. Sin embargo, son sus circuitos integrados periféricos, que aumentan exponencialmente sus prestaciones los que permiten que el sistema de control pueda realizar todas las tareas que se le exigen. Dentro de estos circuitos periféricos se encuentran las básculas 74LS373 y el decodificador 74LS138 que permiten la comunicación dentro del microcontrolador a través del bus de datos de la placa principal. A su vez, los gestores de interrupciones 8259 aumentan el número de posibles fuentes de interrupción que puede gestionar el microcontrolador, incrementando exponencialmente las posibilidades de control. Por último se destacan los circuitos 8255 que permiten ampliar los puertos de salida (en este caso, puesto que también pueden programarse en entrada) del microcontrolador.

En la siguiente imagen se observa las patillas del microcontrolador junto con su respectiva conexión dentro del esquema de la placa principal.

VPP	1		40	
	2		39	
	3		38	
	4		37	
	5		36	
	6		35	
AI	7		34	INTA
A0	8		33	INT
WR	9		32	VDD
RD	10		31	VSS
VDD	11		30	BUS
VSS	12		29	BUS
OSC1	13		28	BUS
OSC2	14		27	BUS
CSI38	15		26	RXD
E LCD	16		25	TXD
RS LCD	17		24	AI I38
A2 I38	18		23	A0 I38
BUS	19		22	BUS
BUS	20		21	BUS

Tabla 4 - Conexiones del microcontrolador en la placa principal.

Para programar el microcontrolador se utilizó un programador de Microchip de cuarenta terminales como el que se observa en la imagen inferior.



Ilustración 125 - Programador de Microchip para dispositivos de hasta cuarenta terminales.

3.- DECODIFICADOR 74LS138.

El decodificador 74LS138 es un decodificador binario, es decir, un circuito combinacional con n entradas y 2ⁿ salidas, en este caso 3 entradas y 8 salidas. EL 74LS138 es un circuito MSI (de mediana escala de integración de integración por sus siglas en inglés “médium scale of intergration”) fabricado con diodos Schottky.

Para cada dato binario de entrada A_i se fija una única salida O_i a 1, cuyo índie “i” corresponde al valor binario del dato de entrada. Su tabla de verdad es la siguiente:

INPUTS						OUTPUTS							
E ₁	E ₂	E ₃	A ₀	A ₁	A ₂	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

U - HIGH Voltage Level

Ilustración 126 - Tabla de verdad del decodificador 74LS138

A continuación puede observarse el diagrama lógico del 74LS138:

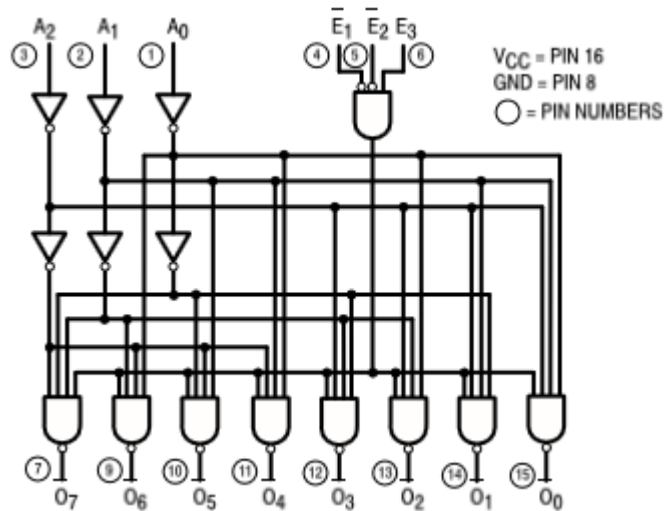


Ilustración 127 - Diagrama lógico del decodificador 74LS138

Las patillas E1 y E2 se conectan a masa de forma que la activación del decodificador dependerá únicamente de E3 (llamada CS a efectos de programación), conectada a la línea RC0 del microcontrolador.

3.1.- EL DECODIFICADOR 74SLS138 DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

Este decodificador se encuentra ubicado en la placa principal y su misión es la de gobernar la activación de las básculas 74LS373 presentes en la misma placa. De esta forma, se establece una comunicación con numerosos dispositivos utilizando únicamente el bus de datos de la placa principal implementado en el puerto D del microcontrolador. Las salidas del decodificador se encuentran negadas, lo cual obliga a utilizar un inversor 40106 (que se explica a continuación) para la conexión con las básculas 74LS373, ya que la báscula es activa por altos. En cambio, la conexión con el buffer bidireccional 74LS245 y los 8259 se realiza directamente (ver esquema eléctrico de la placa principal).

4.- INVERSOR 40106.

Cada inversor 40106 provee de seis buffers inversores a la placa de control, en todas se ubicaron siete circuitos 40106 en la placa. Cada entrada tiene un circuito disparador Schmitt. El inversor cambia de estado en puntos diferentes dependiendo de la dirección del cambio de estado de la entrada (ya sea de nivel alto a nivel bajo o viceversa). Esta diferencia de tensión se define como histéresis de tensión.

4.1.- EL INVERSOR 40106 DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

Los inversores conectados en la placa principal cumplen tres funciones. En primer lugar, la de permitir la conexión de la salida de decodificador 74LS138 con las básculas 74LS373, ya que éstas son activas por altos y la salida del decodificador es de nivel bajo. En segundo lugar, la de asegurar que sea un 1 o un 0 lógico claro lo que llegue a los circuitos, evitando así malas interpretaciones de los niveles de salida. En último lugar, los inversores se

utilizan como barrera física aislante entre los niveles de tensión de las interrupciones provenientes del exterior y las entradas a los circuitos 8259. Las señales de interrupción provienen de las salidas de los sensores finales de carrera y los sensores inductivos de la muñeca del robot. De esta forma, se consigue una protección extra para estos circuitos. A continuación puede observarse el patillaje del 40106:

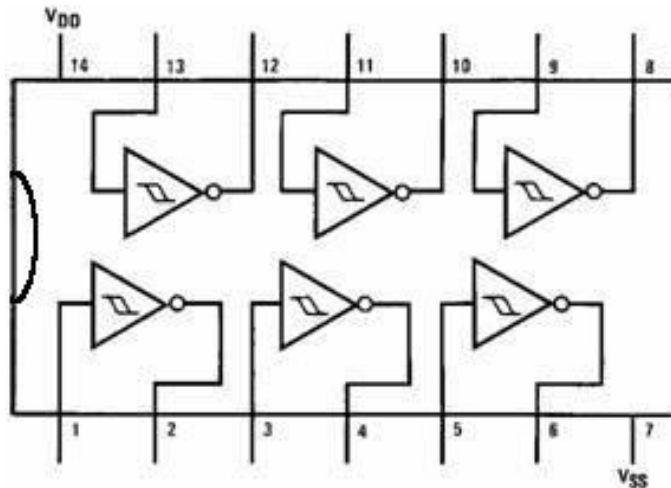


Ilustración 128 - Patillaje del 40106

La señal que entra por la patilla 1 sale invertida por la patilla 2 y así sucesivamente. La patilla 14 es la alimentación y debe conectarse a +5 V. La patilla 7 se conecta a masa.

5.- BÁSCULA 74LS373.

Este circuito consta de ocho buffers, que son el conjunto de 8 latch para almacenar datos formados por flip flop de tipo D y de 8 drivers para amplificar la señal de salida y poder adoptar el estado de alta impedancia. El funcionamiento de los circuitos se basa en dos señales de control, una que hace pasar el dato de entrada en el flip flop a la salida del mismo y otra que se encarga de activar los drivers de salida para poner el dato en el bus de salida del circuito. El patillaje del 74LS373 es el siguiente:

(Señal de activación) \overline{OE}	1E	E20	Vcc (Tensión +5V)
Línea de salida < O0	2S	E19	O7 > Línea de salida
Líneas de entrada { D0	3E	S18	D7 } Líneas de entrada
	4E	S17	D6 } Líneas de entrada
Líneas de salida { O1	5S	S16	O6 } Líneas de salida
	6S	S15	O5 } Líneas de salida
Líneas de entrada { D2	7E	E14	D5 } Líneas de entrada
	8E	E13	D4 } Líneas de entrada
Línea de salida < O3	9S	S12	O4 > Línea de salida
(Tierra 0V) GND	10E	E11	LE (Activar flip flop)

Ilustración 129 - Patillaje del 74LS373

A continuación puede observarse la tabla de verdad y el diagrama lógico del circuito:

74LS373			
D _n	LE=CP	\overline{OE}	O _n
1	1	0	1
0	1	0	0
*	0	0	Q _n
*	*	1	Alta Z

Ilustración I30 - Tabla de verdad 74LS373

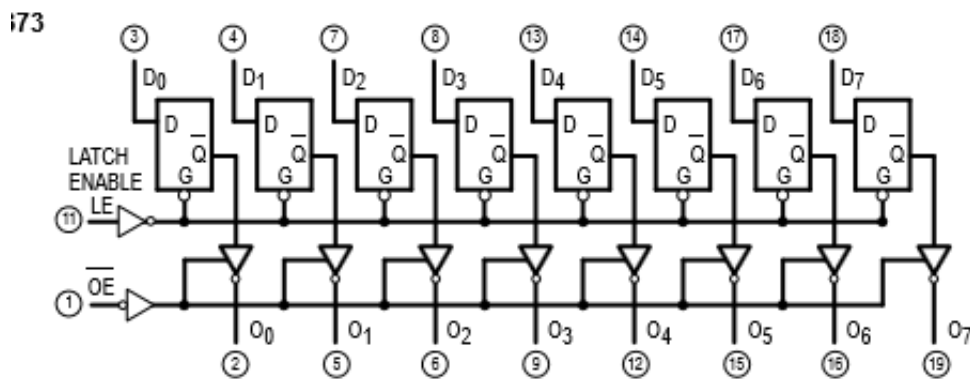


Ilustración I31 - Diagrama lógico 74LS373

A continuación se describen las funciones de cada uno de los terminales de los circuitos:

D0...D7: son los terminales por donde se introducen los datos a los flip flop. Los flip flop pasan dicho datos a su salida por medio de la activación de la señal LE ("latch enable").

O0...O7: son los terminales de salida del circuito que provienen de los drivers que amplifican los datos y que además pueden adoptar el estado de alta impedancia. En el estado de alta impedancia la salida se comporta como si no estuviera conectada al circuito, excepto por una pequeña corriente de fuga que puede fluir hacia adentro o hacia afuera de la terminal de salida.

LE=CS: es la señal de entrada encargada de activar los flip flop para almacenar los datos que tienen a la entrada. Esta señal es activa por altos, es decir, hace funcionar a los flip flop cuando la señal pasa del valor lógico "0" al valor lógico "1". Esta señal se conecta a la salida del decodificador 74LS138 para poder activar la báscula a voluntad.

OE: es una señal de entrada encargada de activar los drivers de salida de los circuitos. Está conectada a los drivers para hacerlos funcionar o en caso contrario ponerlos en estado de alta impedancia. En el circuito este terminal se conecta a masa, por lo que los drivers están en estado de alta impedancia.

5.1.- LA BÁSCULA 74LS373 DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

En total se ubican cinco básculas dentro de la placa principal numeradas desde la 1 a la 5. En el sistema de control diseñado solo se utilizan dos: una para la comunicación con el módulo LCD y otra para la comunicación con el bus de datos de la placa 8255 – 8253 explicada al final de este capítulo. Otras dos básculas de la placa principal fueron diseñadas en un principio para la comunicación con los circuitos conversores digitales - analógicos DAC0800. Sin embargo, como se explica en la última sección de este capítulo, esta sección de la placa era objeto de constantes fallos eléctricos por lo que se debió de fabricar de nuevo.

En la nueva placa auxiliar la conexión entre el puerto A del 8255 – 7 (ver la última sección de este capítulo) y en conversor DAC0800 se realiza prescindiendo de las básculas puesto que se llegó a la conclusión de que el fallo eléctrico presentado en la placa original se debió a la imposibilidad de los 74LS373 de atacar al conversor. Esto ocurre debido a que el circuito no produce a su salida suficiente intensidad como para comunicarse correctamente con el conversor digital - analógico DAC0800.

En la imagen inferior se observa la numeración convenida de las cinco básculas de la placa principal.

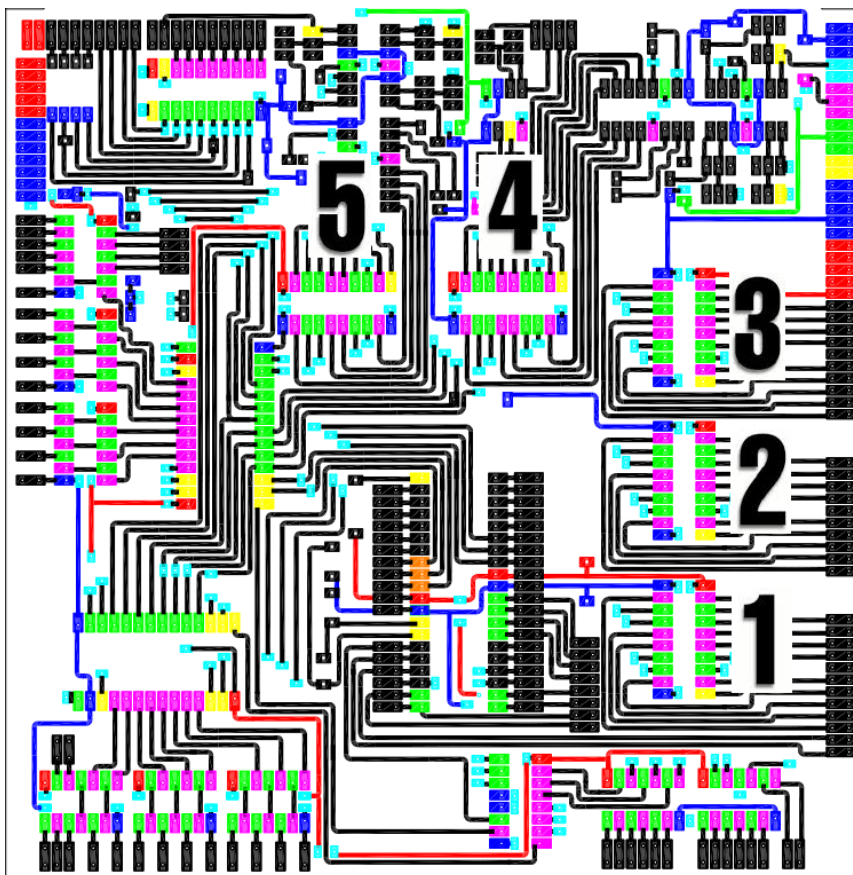


Ilustración 132 - Numeración de las básculas 74LS138 de la placa principal.

6.- BUFFER BIDIRECCIONAL 74LS245.

El buffer bidireccional 74LS245 es un amplificador de señales de ocho bits (bus de datos). Al introducirlo en el bus de datos del sistema se aumenta la capacidad de carga del mismo a la hora de actuar sobre muchos circuitos. Este transceptor de ocho bits está diseñado para permitir comunicaciones asíncronas entre buses de datos. Además, la función de control implementada minimiza los requerimientos externos de tiempo. Este dispositivo permite la transmisión de información desde el bus A hasta el bus B o desde el bus B hasta el bus A, dependiendo del nivel lógico de la patilla de control DIR. La entrada OE (“output enable”) puede deshabilitar el dispositivo para aislar efectivamente los buses.

Este circuito dispone de dos terminales de control, uno que se encarga de activar el circuito y otro que se encarga de dar paso a los datos en un sentido u en otro ya que el circuito es bidireccional. A continuación se describen las funciones de cada uno de los terminales del circuito 74LS245.

A0...A7: son los terminales por donde se introducen o salen los datos ya amplificados. El dato puede entrar o salir por estos terminales en función de cómo se conecten en el circuito y se active una de las señales de control.

B0...B7: son los terminales por donde se sacan o introducen los datos ya amplificados. El dato puede entrar o salir por estos terminales en función de cómo se conecten en el circuito y se active una de las señales de control.

T/R: es una señal de control que hace que los datos entren por unas líneas y salgan por las otras o viceversa. Todo depende de cómo se conecte el circuito con el sistema.

OE: es una señal de entrada encargada de activar los drivers de salida de los circuitos, está conectada a los drivers para hacerlos funcionar o ponerlos en el estado de alta impedancia.

El patillaje del 74LS245 se observa en la siguiente imagen:

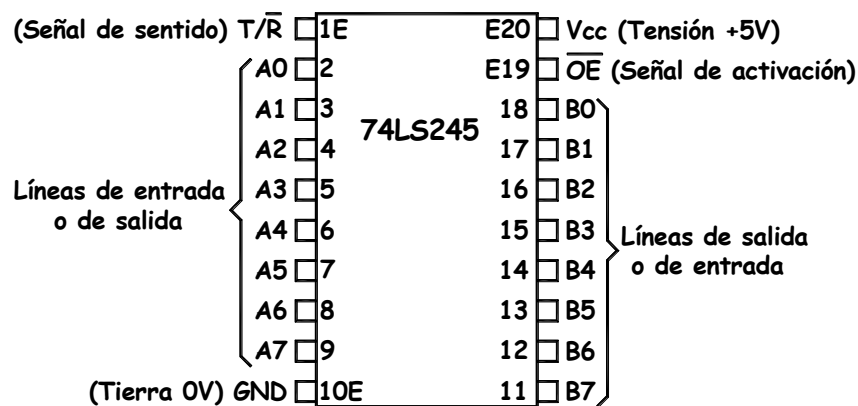


Ilustración 133 - Diagrama del conexionado del 74LS245

En la siguiente tabla se muestra el funcionamiento del circuito:

\overline{OE}	$\overline{T/R}$	Descripción
-----------------	------------------	-------------

0	0	Datos de B hacia A
0	1	Datos de A hacia B
1	*	Alta impedancia

Ilustración 134 - Tabla de verdad del 74LS245

La siguiente imagen muestra el diagrama lógico del circuito:

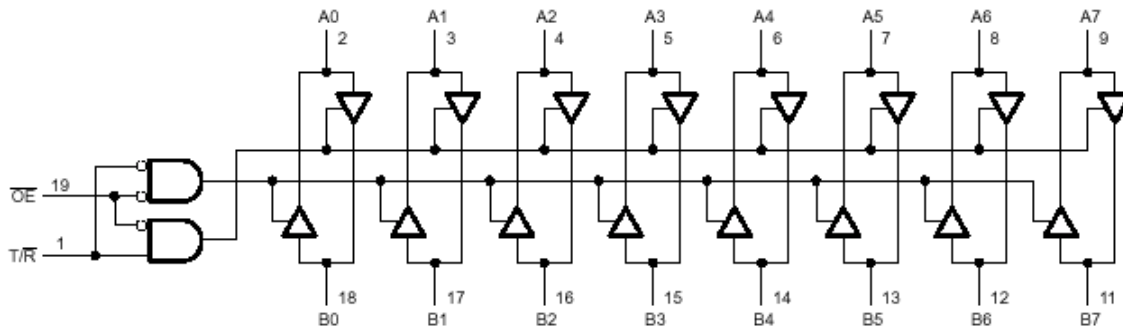


Ilustración 135 - Diagrama lógico del 74LS245

6.1.- EL BUFFER BIDIRECCIONAL 74LS245 DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

Dentro del sistema de control diseñado el buffer bidireccional 74LS245 permitiría una comunicación con un teclado hexadecimal destinado al control sin ordenador de la muñeca del robot. Sin embargo, esta parte del sistema nunca llegó a diseñarse puesto que el control por ordenador requirió mucho más tiempo del que se previó en un principio.

7.- CIRCUITO 8259, CONTROLADOR PROGRAMABLE DE INTERRUPCIONES (PIC).

Un sistema microordenador está diseñado para el control y tratamiento de información y para ello requiere el uso de dispositivos para su comunicación con el exterior tales como teclados, pantallas, sensores, etc. La gestión de dichos periféricos se realiza por la CPU del modo siguiente:}

- El periférico correspondiente emite una interrupción.
- La CPU se da cuenta de que dicho dispositivo solicita su servicio.
- La CPU pasa a ejecutar la subrutina correspondiente del programa, en la que realizará las instrucciones programadas que convertirá en ordenes al exterior si es necesario.

La CPU tiene dos formas de enterarse de que un dispositivo necesita su servicio:

- Método de sondeo: consiste en que la CPU mira cada dispositivo en una secuencia que se repetirá cada cierto tiempo.
- Método de interrupción: el dispositivo llama a la CPU cuando la necesita.

Es claro que el segundo método es más efectivo pues el procesador sólo dedica a los periféricos el tiempo estrictamente necesario. De esta tarea se encarga el 8259 y es capaz de gestionarla con gran eficiencia. Dicho dispositivo sirve para gobernar ocho interrupciones y conectándolo en cascada con otros 8259 se pueden gestionar hasta 64 interrupciones.

El 8259, además de hacer posible con sencillez el uso del ya citado método de interrupción, permite establecer un orden de prioridades en la gestión de periféricos dándoles servicio en el orden deseado. La asignación de prioridades puede hacerla de muchas formas según en el modo de trabajo en que haya sido programado. Además, la programación de la 8259 puede ser cambiada durante cualquier momento en la ejecución del programa principal, aportando dinamismo al sistema de control.

7.1.- INTERRUPTACIONES EN SISTEMAS MICROORDENADORES.

El diseño de un sistema microordenador requiere que dispositivos de entrada-salida tales como teclados, displays, sensores y otros componentes reciban servicio de una manera eficiente, de forma que la gran cantidad de tareas del sistema puedan ser asumidas por el microprocesador con poco o nulo efecto en su rendimiento global.

El método más común para prestar servicio a tales dispositivos es el enfoque sondeo. Este consiste en un proceso de chequeo por parte del procesador de cada dispositivo en una secuencia, “preguntando” a cada uno si necesita servicio. Es fácil comprender que una gran parte del programa principal está continuamente ejecutando un bucle a través de este ciclo de sondeo y que tal método tiene un efecto perjudicial sobre el rendimiento global del sistema, limitando de esta forma las tareas que podría asumir el microcomputador y reduciendo la efectividad que debería suponer el uso de tales dispositivos de entrada-salida.

Un método más deseable sería aquel que permitiera al microprocesador ejecutar un programa principal deteniéndose para servir a los distintos periféricos al ser llamado por alguno de ellos. En realidad, el método proporciona una entrada asíncrona que informa al procesador sobre la inmediata necesidad de completar cualquier instrucción en curso de ejecución, forzándolo a realizar un salto a una nueva rutina de servicio del dispositivo solicitante. Sin embargo, una vez que se ha completado este servicio, el procesador reasume el mando exactamente donde lo dejó (dirigiéndose a la dirección almacenada en el stack pointer antes del salto a la subrutina). Este método se llama de interrupciones y gracias a él el rendimiento del sistema aumenta drásticamente. De esta forma, el microprocesador podrá asumir más tareas, incrementando así su efectividad. El Controlador Programable de Interrupciones (PIC) funciona en un entorno de sistema atacado por interrupciones como un director global. Acepta peticiones desde el periférico, determina cuál de las peticiones que llegan es la de más importancia, mayor prioridad, averigua si la petición tiene un valor de prioridad más alto que el nivel actualmente en servicio y emite a la CPU una interrupción basada en esta determinación.

Cada dispositivo o estructura periférica tiene generalmente un programa o rutina que está asociada a sus requerimientos funcionales u operacionales. Esto es conocido como una rutina de servicio. Después de emitir una interrupción a la CPU el PIC debe introducir de alguna manera información en la misma que permita apuntar, mediante el contador de programa, a la rutina de servicio asociada al dispositivo solicitante. Este puntero es una

dirección en una tabla que está almacenada en la 8259 y que se introducirá mediante las palabras de comando de iniciación.

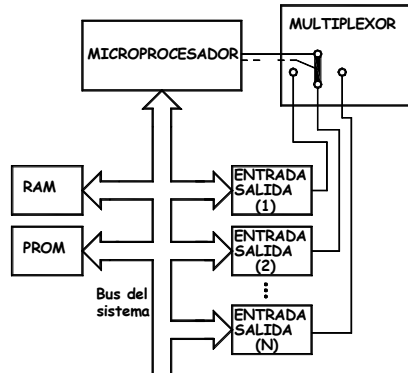


Ilustración I36 - Método de interrupciones

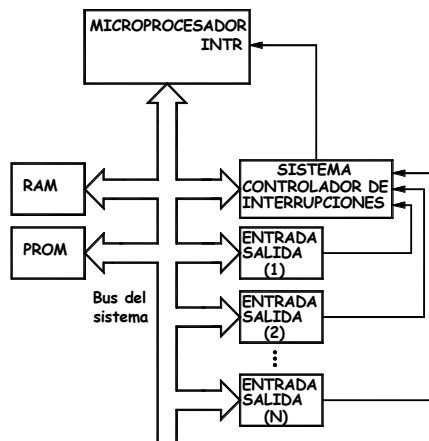


Ilustración I37 - Método de sondeo

7.2.- EL 8259.

Este dispositivo está diseñado específicamente para su uso en sistemas microcomputadores atacados por interrupciones. Tiene la capacidad de gobernar ocho niveles de peticiones y tiene incorporadas características para conectarse en cascada con otros 8259 (hasta 64 niveles). Se programa como un periférico de entrada- salida mediante el software del sistema.

El programador dispone de una selección de modos de prioridad de manera que se puede configurar la forma en la cual el 8259 procesa las peticiones, para así encajar con más requerimientos del sistema. Los modos de prioridad pueden cambiarse dinámicamente en cualquier momento. Esto quiere decir que se puede definir, según las necesidades, la estructura completa de interrupción basada en el entorno del sistema completo. A continuación puede observarse el esquema general de funcionamiento del 8259.

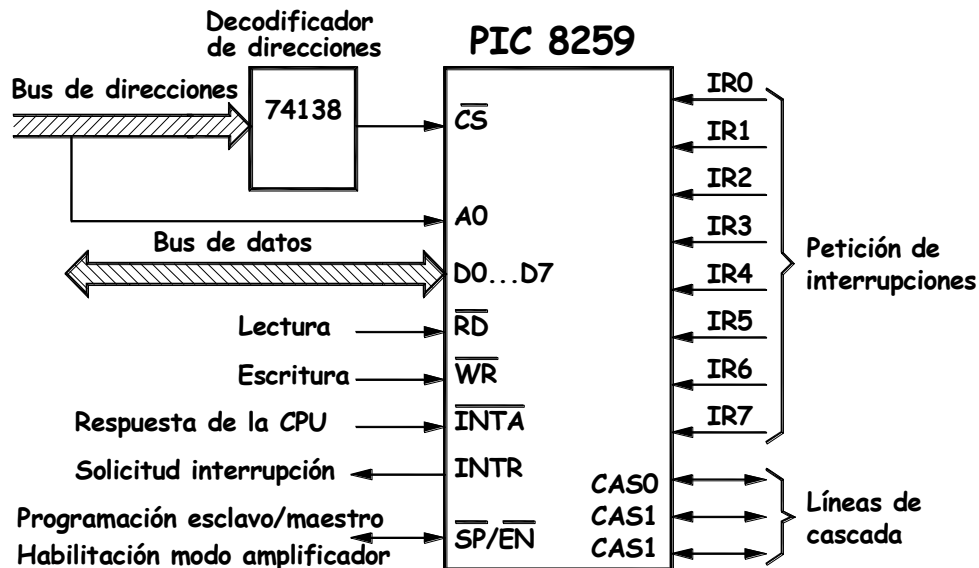


Ilustración I38 - Esquema general de conexión del 8259 dentro de un sistema.

7.3.- RESUMEN DEL FUNCIONAMIENTO DEL 8259.

En este apartado se pretende dar una idea general del funcionamiento del circuito, características, puesta en funcionamiento y programación. Este circuito sirve para controlar desde ocho hasta 64 interrupciones (con otros 8259 conectadas en cascada). También es programable por lo que algunas de sus características de funcionamiento pueden cambiarse por software, entre ellas, la selección de prioridades y el enmascaramiento de algunas interrupciones. A continuación se explica por pasos el funcionamiento de este circuito:

- Se inicia el controlador de interrupciones. Esto se realiza enviando desde la CPU instrucciones OUT de 2 a 4 palabras de iniciación, ICW, según las funciones que haya de realizar el sistema. En estas palabras va información de aspectos como la dirección de la subrutina de cada interrupción, si el 8259 es único en el sistema o está conectado en cascada con otros, con qué CPU está trabajando, etc.
- El 8259 ya está preparado para funcionar. Si suponemos que le llegan varias peticiones de interrupción, las almacena, evalúa su prioridad o si están enmascaradas, envía por el bus de datos a la CPU una interrupción y seguidamente una vez aceptada la petición por parte de la CPU, le manda a ésta la dirección de memoria correspondiente a la subrutina de la interrupción seleccionada.
- El 8259 tiene varios modos de funcionamiento. Se diferencian unos de otros en la forma de evaluar las prioridades, en las máscaras, entre otras características.

Se ha visto en los dos pasos anteriores que una vez iniciado el circuito podía ya funcionar, pero lo hace de un modo determinado y preestablecido. Si se desea que funcione en otro modo debe de programarse enviando desde la CPU palabras de comando, OCW, las cuales pondrán al circuito en el modo de funcionamiento deseado. Como la selección de modo se hace por software, puede cambiarse en cualquier momento la programación del 8259 mediante unas nuevas palabras de comando. El dispositivo programado sigue haciendo

lo mismo, como en el segundo paso; admite interrupciones, selecciona la más prioritaria y hace que la CPU ejecute la subrutina correspondiente.

7.4.- DESCRIPCIÓN DE LOS TERMINALES.

A continuación se describen las funciones de cada uno de los terminales del circuito 8259.

- **DO...D7**: conforman el bus de datos y está formado por ocho líneas que entran en un buffer bidireccional que puede adoptar el estado de alta impedancia. Estas ocho líneas se conectan con el bus de datos del sistema para poder enviar o recibir datos.
- **\overline{CS}** : este terminal es el “chip select” del circuito, a nivel bajo en su entrada habilita el circuito para su funcionamiento. Los valores de \overline{INTA} son independientes del \overline{CS} . Generalmente, este terminal se activa desde la salida de un decodificador de direcciones que se encarga de posicionar al PIC en una zona determinada del mapa de puertos.
- **\overline{RD}** : es una señal de entrada que indica, cuando está negada, que se realiza una operación de lectura sobre el circuito. Se conecta a la misma línea del microprocesador y los datos que se pueden leer son:
 - El estado del registro de petición de interrupción IRR.
 - El estado del registro de servicio ISR.
 - El estado del registro de máscara de interrupción.
 - El nivel de interrupción sobre el bus de datos.
- **\overline{WR}** : es una señal de entrada que indica, cuando está negada, que se realiza una operación de escritura sobre el circuito. Se conecta a la misma línea del microprocesador para poder recibir palabra de iniciación y de comando.
- **A0**: es una línea de entrada al PIC que normalmente se conecta con la línea A0 o A8 de la CPU ya que por ellas sale la misma información cuando se dirige a puertos. Esta señal conjuntamente con \overline{CS} , \overline{RD} y \overline{WR} sirven para seleccionar las operaciones de escritura y lectura sobre diversos registros internos de del 8259.
- **$\overline{SP/EN}$** : es un terminal que soporta una señal bidireccional con una doble función según se trabaje en modo amplificador (“buffered”) o no. El modo amplificador se usa en grandes sistemas con muchos circuitos integrados, en los que son imprescindibles amplificadores para la adaptación de señales con el bus de datos del sistema, al que también están conectados los PIC 8259. Si se trabaja en este modo, se configura el 8259 para enviar una señal de activación de los buffers que les comunica con el bus de datos a través del terminal $\overline{SP/EN}$ con $\overline{EN} = 0$. Así cuando las salidas del PIC al bus de datos están inactivas, también lo está la señal $\overline{SP/EN}$. En modo amplificador hay que configurar a cada PIC como maestro o esclavo mediante software. El bit 3 de la palabra ICW4 programa el modo amplificador y el bit 2 de la misma palabra, determina si el 8259 se trata de un maestro o de un esclavo. Cuando no se trabaja en

modo amplificador, este terminal se usa como una entrada para seleccionar el modo de trabajo del PIC, si $\overline{SP}=0$ se trata de un esclavo y si $\overline{SP}=1$ se trata de un maestro.

- **INTA**: Esta señal de entrada al PIC la utiliza la CPU para comunicarle la aceptación de la petición de interrupción. En contestación a su actuación el 8259 deposita en el segundo \overline{INTA} , que es semejante a un ciclo de lectura, el valor de la dirección de interrupción sobre el bus de datos.
- **INTR**: Es una señal de salida que se pone a nivel alto cuando se confirma una petición de interrupción válida por parte del 8259, se conecta al terminal INTR de la CPU y por tanto se usa para interrumpir a ésta.
- **IR0...IR7**: Son estradas asíncronas para la solicitud de interrupción desde ocho posibles dispositivos. La petición puede generarse mediante un flanco ascendente hasta que se reconozca la petición de interrupción o también pueden activarse por nivel alto en sus terminales.
- **CAS 0, CAS 1 y CAS 2**: Son tres líneas bidireccionales que conforman un bus local, en el conexionado en cascada de varios 8259. Actúan como salida en el PIC maestro que deposita en ellos el código del PIC esclavo favorecido. Los PIC esclavos reciben el código depositado por el maestro y reconocen si se les concede la petición de interrupción. El PIC esclavo seleccionado será el que deposite en el bus de datos del sistema, la dirección de la interrupción a gestionar.
- **Vcc**: Entrada de la tensión de alimentación +5V.
- **GND**: Tierra o masa.

A continuación puede se detalla los terminales del circuito:

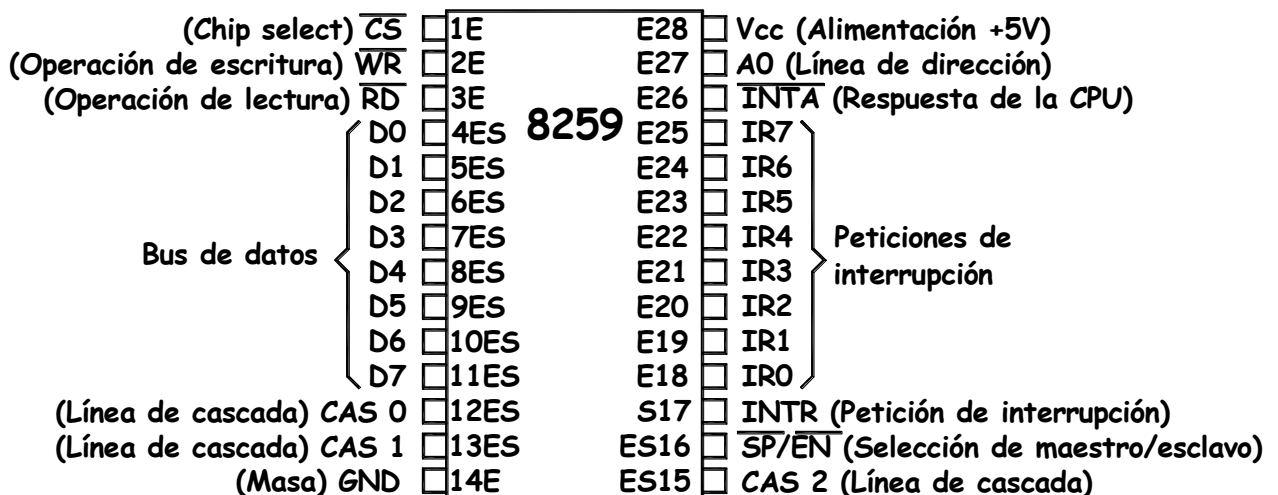


Ilustración 139 - Diagrama detallado de los terminales del 8259

7.5.- DESCRIPCIÓN DE LOS BLOQUES DEL CONTROLADOR.

- **Buffer del bus de datos:** este buffer es bidireccional triestado y tiene 8 bits. Se usa para interconectar el 8259 con el bus de datos del sistema. La información de estados del 8259 a la CPU y palabras de control de la CPU al 8259A se transfieren a través de éste buffer.
- **Lógica de control lectura-escritura:** la función de este bloque es almacenar comandos de la CPU. Contiene los registros de iniciación que almacenan las palabras de iniciación ICW y los registros de comando que almacenan las palabras de operación OCW. El conjunto de dichas palabras configurarían los diversos formatos de control para los distintos modos de funcionamiento del dispositivo. Este bloque funcional permite también que el estado del 8259 sea transferido sobre el bus de datos. A continuación puede observarse el diagrama de bloques del circuito:

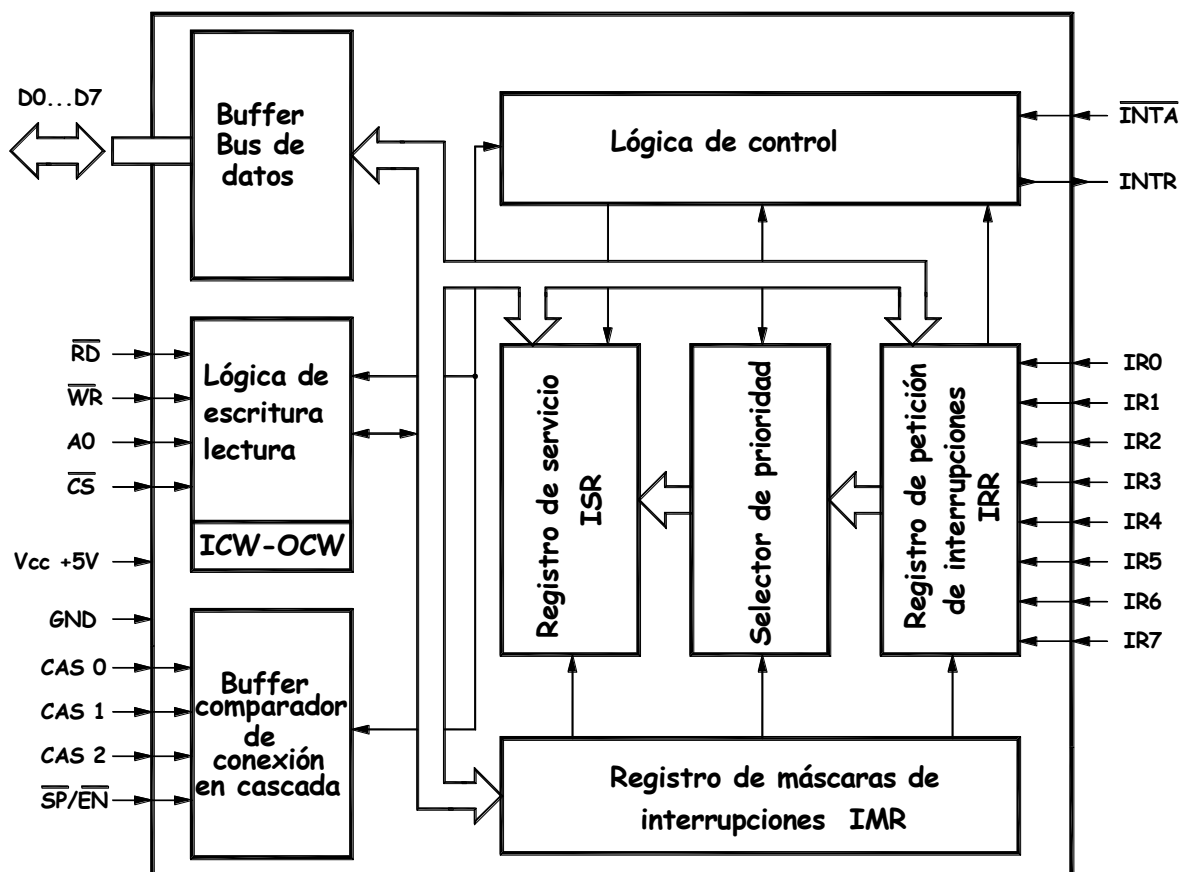


Ilustración I40 - Diagrama de bloques del 8259.

- **Selector de prioridad:** este bloque lógico determina las prioridades de los bits que están a "1" dentro del IRR. Se selecciona el de mayor prioridad y se permite que su correspondiente bit del ISR se ponga a "1" durante los impulsos \overline{INTA} , los cuales provienen de la CPU cuando es aceptada una interrupción en ella.

- **Registro de máscara de la interrupción IMR:** el IMR almacena los bits que enmascaran las líneas de interrupción que se quieren enmascarar o prohibir. Para hacer efectivas estas máscaras el IMR actúa sobre las salidas del IRR. El enmascarar una entrada de mayor prioridad no afectará a las líneas de petición de interrupción de menor prioridad. Es un registro de ocho bits.
- **Registro de petición de interrupción IRR y registro de servicio ISR:** las interrupciones de las líneas de entrada IRO...IR7 las mantienen dos registros en cascada:
 - El registro de petición de interrupción IRR y el registro de servicio ISR.
 - El IRR es un registro de ocho bits se usa para almacenar todos los niveles de interrupción que están solicitando servicio, cuando se reconoce una interrupción, se borra del IRR el nivel de petición más alto no enmascarado. El ISR es otro registro de ocho bits y es utilizado para almacenar todos los niveles de interrupción que están siendo servidos. Como veremos pueden estar sirviéndose más de un nivel a la vez. Por ejemplo en casos en que es solicitada una interrupción de mayor prioridad que la que está actualmente en ejecución.
- **Buffer-comparador de cascada:** este bloque funcional almacena el identificador ID, de cada 8259 esclavo utilizado en el sistema, uno por esclavo.

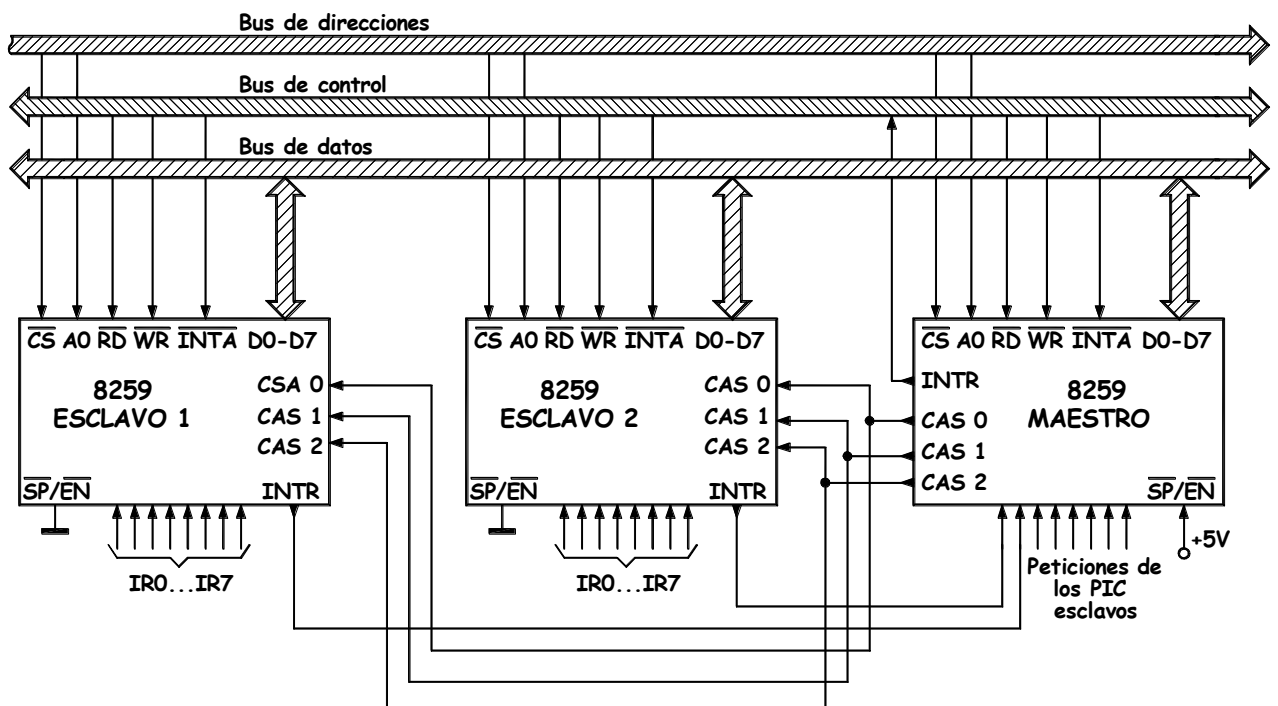


Ilustración 141 - Acoplamiento en cascada de dos PIC esclavos y un maestro

El ICW del dispositivo maestro y el ICW del dispositivo esclavo se observan a continuación:

ICW3 Dispositivo maestro

A0	D7	D6	D5	D4	D3	D2	D1	D0
I	S7	S6	S5	S4	S3	S2	S1	S0

I=La entrada IR tiene un esclavo
0=La entrada IR no tiene un esclavo

ICW3 Dispositivo esclavo

A0	D7	D6	D5	D4	D3	D2	D1	D0
I	0	0	0	0	0	ID2	ID1	ID0

0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1

Tabla 5 - ICW 8259 dispositivo maestro y dispositivo esclavo

D2, D1 y D0 son identificadores que constan de un conjunto de tres bits y que permiten por tanto diferenciar hasta ocho de estos circuitos. Son enviados desde la CPU en ICW3, como se observa en la tabla anterior. En resumen es similar a asignar un nombre a cada 8259 esclavo del sistema.

Las líneas CAS 0...CAS 2 son salidas cuando se programa el 8259 como maestro y entradas si es esclavo. Como maestro el 8259 envía el ID del dispositivo esclavo que desea seleccionar por las líneas CAS 0...CAS 2, el esclavo compara si su ID internamente almacenado en este bloque es igual al enviado por el maestro. El esclavo así elegido enviará la dirección de su subrutina preprogramada al bus de datos durante los dos siguientes impulsos \overline{INTA} .

7.6.- SECUENCIA DE INTERRUPCIÓN.

Las características más potentes del 8259 en un sistema microordenador son su capacidad de programación y su capacidad de direccionamiento de rutinas de interrupción. La última permite saltos directos o indirectos a la rutina específica de interrupción solicitada sin ninguna clase de sondeo ("polling") del dispositivo de interrupción. La secuencia normal durante una interrupción depende del tipo de CPU que se utiliza en el sistema. La secuencia de interacción entre el microcontrolador y el 8259 es la siguiente:

1. Suben a "1" una o más de las líneas de petición de interrupción, poniendo a "1" los bits correspondientes del IRR.
2. El 8259 evalúa estas peticiones y envía un INTR a la CPU si procede.
3. La CPU reconoce el INT y responde con un pulso \overline{INTA} .
4. Al recibir un \overline{INTA} de la CPU, se pone a "1" el bit de ISR que tenga mayor prioridad, y se borra el correspondiente bit del IRR. El 8259 liberará también

un código de interrupción CALL (11001101) sobre las ocho líneas del bus de dato D0...D0.

5. Esta instrucción CALL hará que la CPU envíe al 8259 dos pulsos $\overline{\text{INTA}}$ más.
6. Estos dos pulsos $\overline{\text{INTA}}$ permiten al 8259 liberar la dirección de su subrutina preprogramada sobre el bus de datos. Los ocho bits de direcciones menos significativos se liberan al primer pulso $\overline{\text{INTA}}$, y los ocho bits de direcciones de mayor peso se liberan en el segundo pulso $\overline{\text{INTA}}$.
7. Esto completa los tres bytes de la instrucción CALL liberados por el 8259. En el modo AEOI (que es un modo de funcionamiento que se describe más adelante), el bit de ISR se borra al final del tercer impulso $\overline{\text{INTA}}$. De otro modo el bit de ISR permanece a "1" hasta que se emite un comando EOI (también se describe más adelante) apropiado mediante la palabra de comando de operación OCW2, al final de la secuencia de interrupción.

Los bytes que pone el 8259 sobre el bus de datos son la salida de la secuencia de interrupción. Estos bytes los lee el microcontrolador para seguir la instrucción completa e ir a la rutina de servicio correspondiente. La secuencia de interrupción está sincronizada por tres pulsos $\overline{\text{INTA}}$. Durante el primer pulso $\overline{\text{INTA}}$ se libera el código de instrucción CALL sobre el bus de datos, como se observa en la tabla siguiente.

Código CALL							
D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	1	1	0	1

Tabla 6 - Código de instrucción CALL

Durante el segundo pulso $\overline{\text{INTA}}$ es puesto en el bus de datos el byte menos significativo de la dirección de la rutina de servicio perteneciente a la interrupción reconocida. Cuando el intervalo de separación de las rutinas es 4 serán puestos por el programador los bits A5, A6 y A7, mientras que el 8259 introduce automáticamente A0...A4. Si el intervalo es 8 serían programados A6 y A7 poniendo el resto el circuito PIC.

IR	Intervalo=4							
	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	A5	0	0	0	0	0
1	A7	A6	A5	0	0	1	0	0
2	A7	A6	A5	0	1	0	0	0
3	A7	A6	A5	0	1	1	0	0
4	A7	A6	A5	1	0	0	0	0
5	A7	A6	A5	1	0	1	0	0
6	A7	A6	A5	1	1	0	0	0
7	A7	A6	A5	1	1	1	0	0

IR	Intervalo=8							
	D7	D6	D5	D4	D3	D2	D1	D0

0	A7	A6	0	0	0	0	0	0
1	A7	A6	0	0	1	0	0	0
2	A7	A6	0	1	0	0	0	0
3	A7	A6	0	1	1	0	0	0
4	A7	A6	1	0	0	0	0	0
5	A7	A6	1	0	1	0	0	0
6	A7	A6	1	1	0	0	0	0
7	A7	A6	1	1	1	0	0	0

Tabla 7 - Byte menos significativo de la dirección de interrupción

Durante el tercer pulso $\overline{\text{INTA}}$ se libera sobre el bus de datos el byte más significativo de la dirección de rutina, dicho byte habrá sido puesto anteriormente mediante la programación.

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

Tabla 8 - Byte más significativo de la dirección de interrupción

7.7.- PROGRAMACIÓN DEL 8259.

Como se ha dicho anteriormente, para el correcto funcionamiento del 8259 han de seguirse dos pasos: iniciación y programación. Una vez iniciado, el circuito puede ya funcionar, pero lo hará en el "modo completamente encajado", es decir, con unas prioridades y funciones determinadas. Habitualmente, es necesario que el dispositivo funcione en otro modo de funcionamiento, para ello ha de programarse según los requerimientos del sistema.

Estos dos pasos se llevan a cabo mediante palabras de comando, que es un conjunto de 9 bits que envía el microcontrolador al 8259, un bit lo pone la línea de dirección A0 y las ocho restantes se mandan a través del bus de datos. Estos bytes enviados desde la CPU se realizan mediante instrucciones OUT. Dichas palabras pueden ser de dos tipos:

1. Palabras de iniciación ICW: son una secuencia de dos a cuatro palabras que permitirán al 8259 ponerse en su funcionamiento normal. Deberá enviarse una secuencia de estas a cada 8259 del sistema convenientemente direccionado. En la siguiente figura se ve el orden de envío de las ICW.

2.- Palabras de comando OCW: son las palabras de comando que se mandan al circuito para los diversos modos de funcionamiento tales como:

- Enmascaramiento de interrupción.
- Fin de interrupción.
- Prioridad rotativa.

Las OCW pueden ser escritas en cualquier momento después de la iniciación. Las diversas palabras deben ser enviadas por el microcontrolador sincronizada con pulsos $\overline{\text{WR}}$ para permitir su escritura en el 8259. Ambos tipos de palabras pueden cambiarse en cualquier lugar de la ejecución del programa del sistema. A continuación se describen los

diversos pasos en la programación del 8259. Se tienen en cuenta no sólo los bytes puestos en el bus de datos del 8259 sino también los de la línea A0.

Palabras de iniciación

Palabra de iniciación ICW1

Cuando se emite una palabra de iniciación con la línea D4=1 y con A0=0 el 8259 la interpreta como la primera palabra de inicialización ICW1. Esta palabra inicia la secuencia de iniciación en la que ocurre lo siguiente:

1. Se borra el circuito sensor de flanco, es decir, después de la iniciación, una petición de interrupción debe efectuar una transición de bajo a alto para generar una interrupción.
2. Se borra el registro de máscara de interrupción.
3. A la entrada IR7 se le asigna prioridad 7, o sea la menor prioridad.
4. La dirección en modo esclavo sobre las líneas de cascada se pone a 7.
5. Se borra el modo de máscara especial y la lectura de estado corresponde a IRR.
6. Si IC4=0 todas la funciones seleccionadas en ICW4 se ponen a 0.

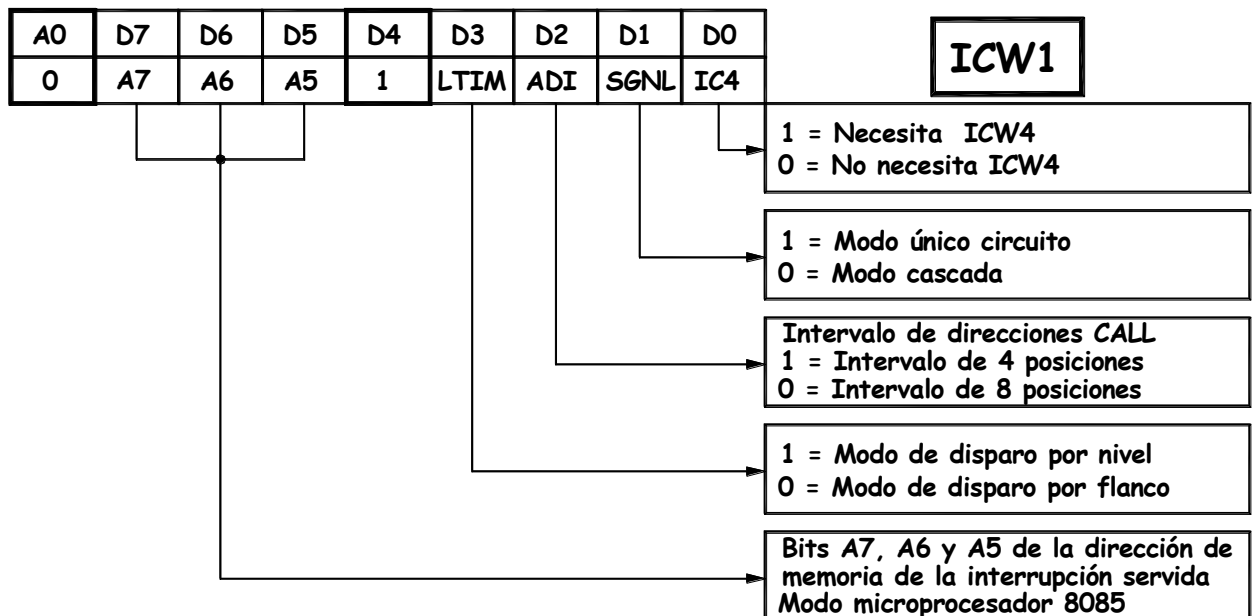


Ilustración I42 - Palabra de iniciación ICW1

Palabra de iniciación ICW2

Cuando se emite una palabra de iniciación con la línea A0=1 y dicha palabra va detrás de la palabra de iniciación con la línea D4=1 y con A0=0, el 8259 interpreta que esta es la palabra ICW2 donde está contenida la primera dirección de memoria para la interrupción IR0.

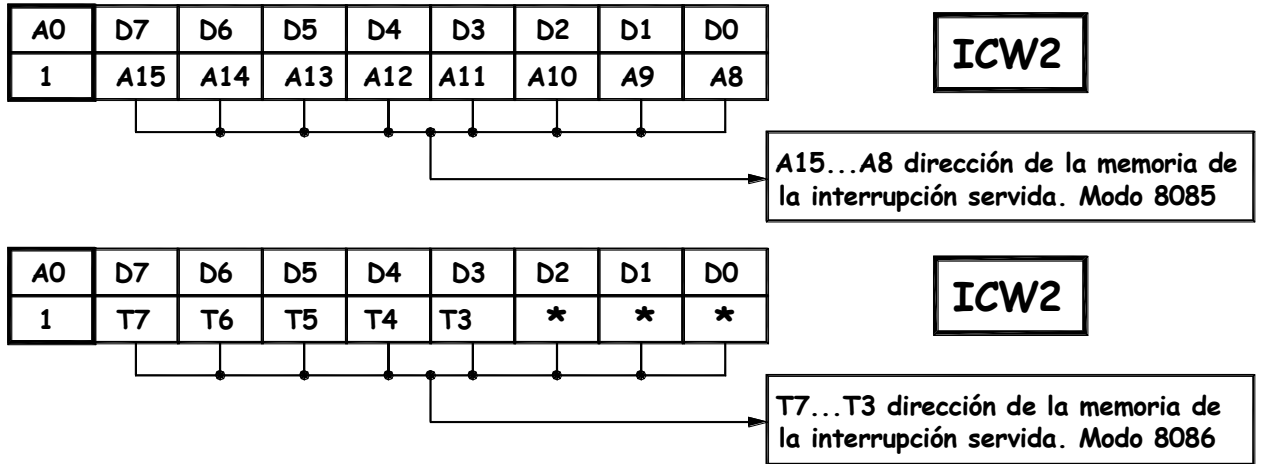


Ilustración I43 - Palabra de iniciación ICW2

A continuación se describen algunas generalidades acerca de las rutinas de servicio. Estas rutinas deben de estar cargadas en unas direcciones de memoria. El funcionamiento del 8259 obliga a que lo estén de un modo determinado.

Código CALL							
D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	1	1	0	1

Primer byte que envía el 8259 al primer pulso INTA

Tabla 9 – Primer byte que envía el 8259 al primer pulso INTA

IR	Intervalo de 4 posiciones							
	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	A5	0	0	0	0	0
1	A7	A6	A5	0	0	1	0	0
2	A7	A6	A5	0	1	0	0	0
3	A7	A6	A5	0	1	1	0	0
4	A7	A6	A5	1	0	0	0	0
5	A7	A6	A5	1	0	1	0	0
6	A7	A6	A5	1	1	0	0	0
7	A7	A6	A5	1	1	1	0	0

IR	Intervalo de 8 posiciones							
	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	0	0	0	0	0	0
1	A7	A6	0	0	1	0	0	0
2	A7	A6	0	1	0	0	0	0
3	A7	A6	0	1	1	0	0	0
4	A7	A6	1	0	0	0	0	0
5	A7	A6	1	0	1	0	0	0
6	A7	A6	1	1	0	0	0	0
7	A7	A6	1	1	1	0	0	0

Tabla 10 - Byte menos significativo de la dirección de la interrupción

Modo microprocesador 8085							
D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

Byte más significativo de la dirección de la interrupción

Tabla 11 - Byte más significativo de la dirección de la interrupción

Modo microprocesador 8086							
D7	D6	D5	D4	D3	D2	D1	D0
T7	T6	T5	T4	T3	*	*	*

Tabla 12 - Byte menos significativo de la dirección de la interrupción

Palabra de iniciación ICW3

Esta palabra se escribe sólo cuando hay más de un 8259 en el sistema y se utilizan en cascada, dicha palabra será aceptada solamente si en ICW1 tenemos SNGL=0 (habilita el modo cascada, el "1" habilita el modo único circuito, que se utiliza cuando solo se trabaja con un 8259). Cabe recalcar que se deben de enviar palabras diferentes a esclavos y a maestros. La pala de iniciación ICW3 se carga en el registro esclavo de 8 bits. Las funciones de este registro son las siguientes:

- En el modo maestro (bien cuando $\overline{SP} = 0$ o cuando M/S=1 en el modo buffer) se pone un "1" para cada esclavo en el sistema. El maestro liberará entonces el byte primero de la secuencia CALL y permitirá a través de las líneas de cascada, al esclavo correspondiente poner los bytes segundo y tercero sobre el bus de datos.
- En el modo esclavo (bien cuando $\overline{SP} = 0$ o BUF=1 y M/S=0 en ICW4) los bits D3, D2 y D1, es decir, ID2, ID1 e ID0 respectivamente identifican al esclavo receptor de la palabra ICW3. El esclavo compara lo que el maestro envía por las líneas de cascada y si son iguales los tres bits, libera los bytes segundo y tercero de la secuencia CALL.

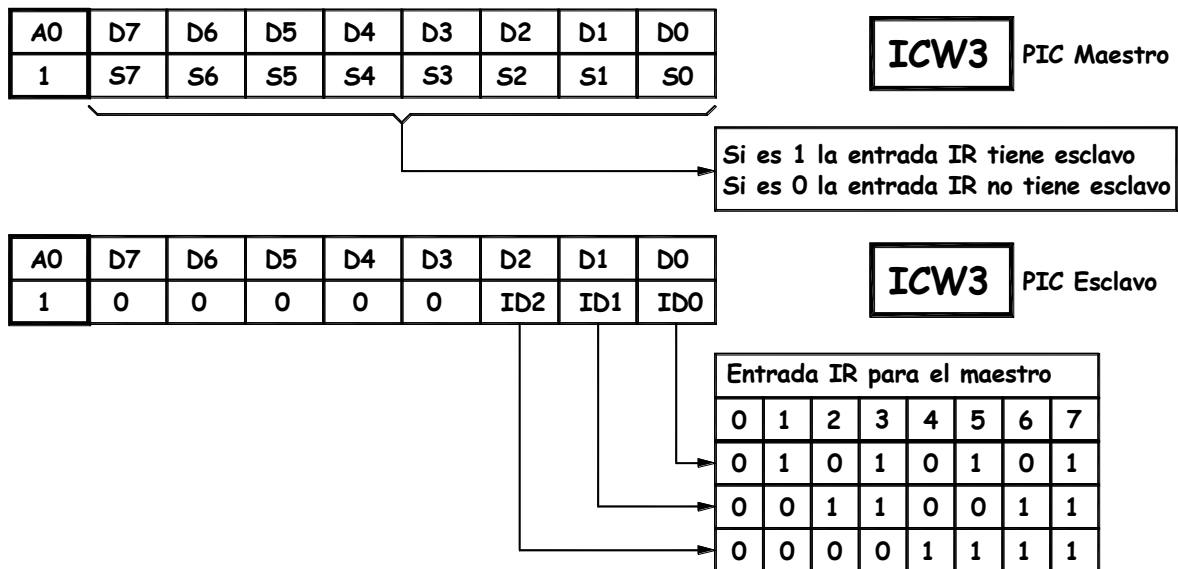


Ilustración I44 - Palabra de iniciación ICW3

Palabra de iniciación ICW4

A continuación se detalla la función de cada uno de los bits que contiene la palabra de iniciación ICW4:

- **SFNM:** si SFNM=1 se programa al 8259 en el modo completamente encajado especial ("special fully nested"). Si SFNM=0 se programa en el modo no especial.
- **BUF:** si BUF=1 se programa el modo amplificador ("buffered"). En este modo $\overline{SP/EN}$ se convierte en una salida de permiso para validar los buffer externos del bus de datos y la determinación de maestro-esclavo se hace mediante M/S. Si BUF=0 se programa el modo no amplificador.

- **M/S:** si se selecciona el modo amplificador (“buffered”) M/S=1 significa que el 8259 está programado como maestro y M/S=0 como esclavo. Si BUF=0, M/S no tiene ninguna función.
- **AEOI:** Si AEOI=1 se programa el modo de fin automático de la interrupción. Si AEOI=0 se programa en el modo normal.
- **μPM:** Modo microprocesador, si μPM=0 prepara al 8259 para trabajar con sistemas microprocesador 8080/8085 (modo que se acopla perfectamente con el microcontrolador PIC16F877A) y si μPM=1 lo prepara para funcionar en sistemas microprocesador 8086/8088.

Hay que recordar que si IC4=0 en la palabra de iniciación ICWI no se lee ICW4 y que todos los bits de los registros que seleccionaría se ponen a "0".

Palabras de comando

Constará de tres palabras que se pueden enviar al 8259 cuando se quiera cambiar el modo de funcionamiento del circuito dentro del sistema. Después de que las palabras de iniciación ICW han sido programadas en el 8259 el circuito está listo para aceptar peticiones de interrupción en sus líneas de entrada. A partir de este momento, se puede hacer que el 8259 funcione en otros modos distintos mediante las palabras de comando OCW.

OCW1								
A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0
OCW2								
A0	D7	D6	D5	D4	D3	D2	D1	D0
0	R	SL	EOI	0	0	L2	L1	L0
OCW3								
A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	ESMM	SMM	0	I	P	RR	RIS

Tabla 13 - Palabras de comando

Palabra de iniciación OCWI

Esta palabra activa o borra los bits de máscara del registro de interrupción IMR. M7...M0 son los ocho bits de máscara. Si Mx=1 significa que el canal x queda enmascarado (prohibido). Si Mx=0 hace que el canal x quede permitido.

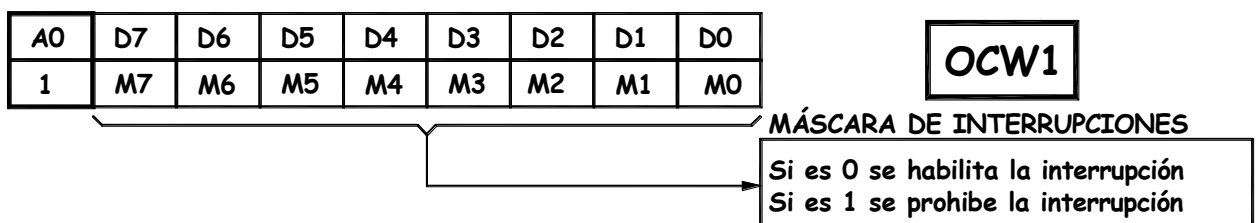


Ilustración 145 - Palabra de comando OCW1

Palabra de iniciación OCW2

Los bits R, SL y EOI controlan los modos rotativo y de fin de interrupción y combinaciones de ambos, dichos modos se explican más adelante. En la figura inferior puede observarse una tabla con estas combinaciones. Los bits L2, L1 y L3 determinan el nivel de interrupción sobre el que se actúa cuando SL es activo, es decir, toma el valor "1".

OCW2		A0	D7	D6	D5	D4	D3	D2	D1	D0	
		0	R	SL	EOI	0	0	L2	L1	L0	
Fin de interrupción	Comando EOI no específico	0	0	1				0	0	0	0
	Comando EOI específico	0	1	1				0	0	1	1
Rotación automática	Comando EOI de rotación o no específico	1	0	1				0	1	0	2
	Actuación modo EOI rotación automática SET	1	0	0				0	1	1	3
	Desactivación modo EOI rotación automática CLEAR	0	0	0				1	0	0	4
Rotación específica	Rotación o comando específico EOI	1	1	1				1	0	1	5
	Comando de actuación de prioridad	1	1	0				1	1	0	6
	No operativa	0	1	0				1	1	1	7

Bits que establecen el nivel de la interrupción

Ilustración 146 - Palabra de comando OCW2

Palabra de iniciación OCW3

Los bits de esta palabra controlan otros modos de funcionamiento del 8259:

- **ESMM:** permiso de modo de máscara especial. Cuando este bit está a "1" permite al bit SMM activar o desactivar el modo de máscara especial.
- **SMM:** modo de máscara especial. Si ESMM=1 y SMM=1 el 8259 trabaja en modo de máscara especial, y si ESMM=1 y SMM=0 el 8259 volverá al modo de máscara normal.

OCW3		A0	D7	D6	D5	D4	D3	D2	D1	D0
		0	*	ESMM	SMM	0	1	P	RR	RIS
No actúan		0	0					0	0	No actúan
No actúan		0	1					0	1	No actúan
RESET enmascaramiento especial		1	0					1	0	Leer registro IRR en el siguiente pulso \overline{RD}
SET enmascaramiento especial		1	1					1	1	Leer registro ISR en el siguiente pulso \overline{RD}

Si es 1 comando poll (SONDEO)
Si es 0 no comando poll (NO SONDEO)

Ilustración 147 - Palabra de comando =CW3

En resumen, para poner en funcionamiento el sistema controlador de interrupciones hay que enviar de 2 a 4 bytes de iniciación para cada dispositivo 8259 (maestro y esclavos si los hubiera). Además, hay que mandar las palabras de comando OCW necesarias a cada circuito para que funcione en el modo deseado. Debe de tenerse en cuenta una diferencia esencial entre las palabras de iniciación y de comando.

	A0	D7	D6	D5	D4	D3	D2	D1	D0	
ICW1	0	A7	A6	A5	1	LTIM	ADI	SGNL	IC4	
ICW2	1	A15	A14	A13	A12	A11	A10	A9	A8	80/85
ICW2	1	T7	T6	T5	T4	T3	*	*	*	86/88
ICW3	1	S7	S6	S5	S4	S3	S2	S1	S0	Maestro
ICW3	1	0	0	0	0	0	ID2	IDI	ID0	Esclavo
ICW4	1	0	0	0	SFNM	BUF	M/S	AEOI	μPM	
OCW1	1	M7	M6	M5	M4	M3	M2	M1	M0	
OCW2	0	R	SL	EOI	0	0	L2	L1	L0	
OCW3	0	*	ESMM	SMM	0	1	P	RR	RIS	

Tabla 14 - Palabras de iniciación ICW y de comando OCW

Las OCW pueden ser enviadas por separado, es decir, en cualquier momento de la ejecución del programa se puede enviar cualquiera de ellas (para cambio de máscaras, cambio o restauración de modo de operación, etc.) según las necesidades del sistema. El 8259 las reconoce y diferencia entre sí. Se diferencia OCW2 de OCW3 por los bits D3 y D4 y ambas de OCW 1 mediante el bit A0. En cambio, las ICW deben ser enviadas en conjunto según la secuencia indicada en el apartado anterior. El 8259 sabrá que se inicia esta secuencia cuando le llegue por la línea A0 un "0" y por D4 un "1", este último bit las distingue además de OCW2 y OCW3.

Nótese que si se envía al 8259 direccionado una pretendida palabra de comando después de ICW1 ésta no sería interpretada como tal, en todo caso el 8259 la entendería como una ICW. Así, si se quiere cambiar alguna de las características de funcionamiento incluidas en las ICW debe comenzarse enviando ICW1. De este sencillo modo el dispositivo es capaz de diferenciar e interpretar las órdenes de programación que le llegan.

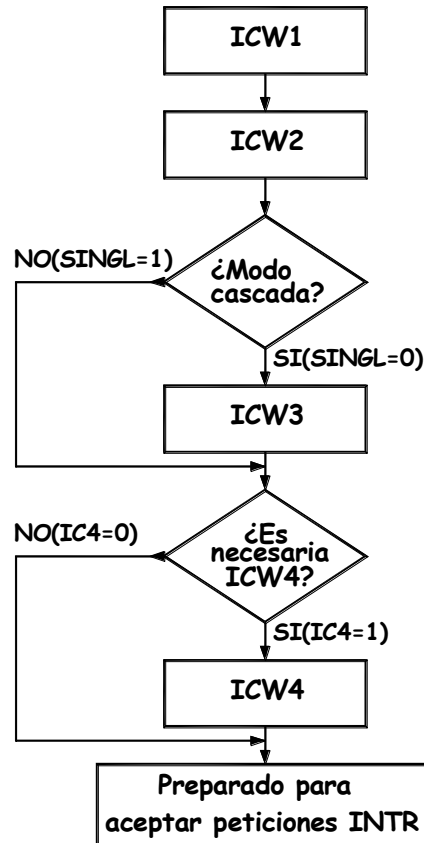


Ilustración 148 - Secuencia de las palabras de iniciación

7.8.- MODOS DE FUNCIONAMIENTO.

A continuación se describen los aspectos básicos de funcionamiento de los distintos modos del 8259. Un modo de funcionamiento se describe como la forma en que el 8259 responderá, según su programación, a cada conjunto de órdenes que le llegan del exterior. No puede decirse que cada modo de funcionamiento sea único y excluyente, pues cada modo incluye numerosas posibilidades. Sin embargo, en vistas a una descripción más académica se explicarán por separado, sin entrar en las combinaciones entre ellos.

Cuando una interrupción es reconocida se determina la petición de mayor prioridad y su vector de interrupción se sitúa en el bus de datos. Adicionalmente se pone a "1" el bit del registro de servicio correspondiente a la interrupción reconocida.

7.8.1.- MODO COMPLETAMENTE ENCAJADO ("FULLY NESTED MODE").

Este modo entra después de la secuencia de iniciación si el circuito no se ha programado en otro modo. A continuación de dicha frecuencia las prioridades están ordenadas de 0 a 7, siendo IR0 la entrada más prioritaria e IR7 la de prioridad menor.

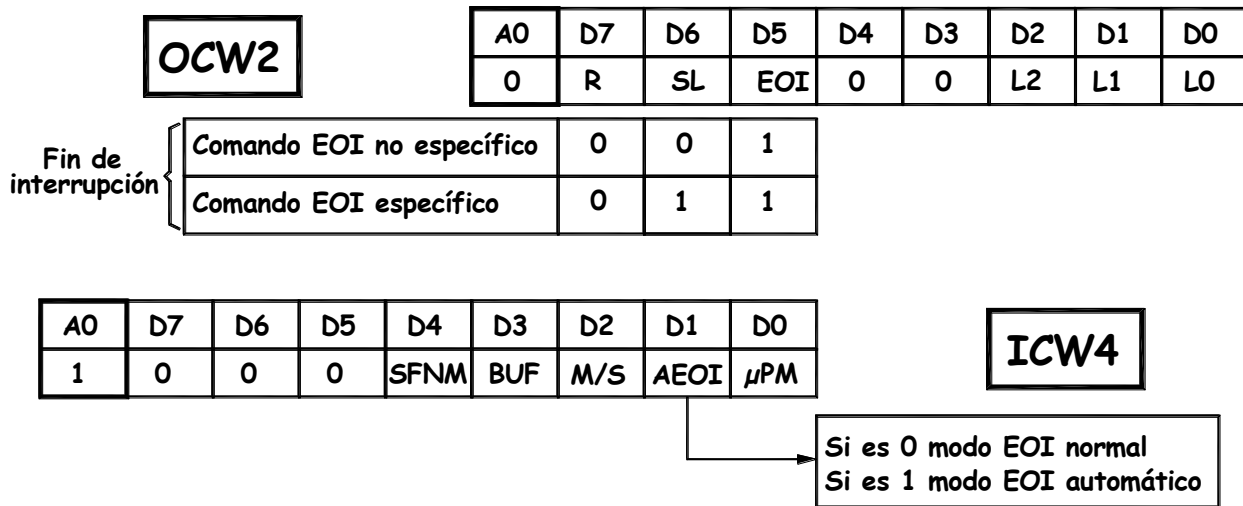


Ilustración 149 - Palabra OCW2 y OCW4

El bit del registro ISR que se ha puesto a “1” con el reconocimiento de una interrupción permanece con ese valor hasta que el microcontrolador emita un comando de fin de interrupción. Esta orden se emite mediante una OCW2 inmediatamente antes de volver de la rutina de interrupción o después del último \overline{INTA} si está puesto a “1” el bit AEOI de ICW4 (fin de interrupción automático). Mientras el bit ISR está a “1” quedan prohibidas todas las demás interrupciones de igual o menor prioridad. Niveles superiores generarían una interrupción que sería reconocida sólo si el flip-flop de permiso de interrupción interno del microcontrolador ha sido permitido de nuevo por software (bit GIE 7 y/o bit RBIE 4 del registro INTCON).

7.8.2.- MODO FIN DE INTERRUPCIÓN EOI.

El bit del registro ISR puesto a “1” cuando la interrupción es reconocida puede borrarse de dos formas:

- Automáticamente a continuación de la subida del último pulso \overline{INTA} de la secuencia de interrupción. Esto ocurre si se ha programado el bit AEOI de ICW4 a “1”.
- Mediante una palabra de comando OCW2 que debe ser enviada al 8259 antes de volver de una rutina de servicio, se llamará a esto comando EOI. Un comando EOI debe emitirse dos veces si está en el modo cascada, una vez para el maestro y otra para el esclavo correspondiente. Hay dos tipos de comando EOI: específico y no específico. A continuación se describe cada uno.
 - Un comando EOI no específico en el modo completamente encajado borrará automáticamente el bit ISR de mayor peso de los que están a “1”, ya que en este modo el nivel ISR más alto fue necesariamente el último nivel reconocido y servido. En OCW2 deberá enviarse EOI=1, SL=0 y R=0.

- Cuando se usa un modo en que las prioridades no son fijas como, por ejemplo, el rotativo en el que el 8259 no es capaz de determinar el último nivel reconocido. En este caso debe emitirse un comando específico de final de interrupción, el cual incluye en su código el nivel ISR a borrar. Un EOI específico será enviado con OCW2, EOI=1, SL=1, R=0 y L2...L0 con el nivel binario del bit ISR a ser borrado.

OCW2	A0	D7	D6	D5	D4	D3	D2	D1	D0
	0	R	SL	EOI	0	0	L2	L1	L0

Ilustración 150 - Palabra OCW2

Debe tenerse en cuenta que un nivel enmascarado por un bit IMR (registro de la máscara de interrupción) no se borrará con un EOI específico si el 8259 está en el modo máscara especial.

7.8.3.- MODO FIN AUTOMÁTICO DE INTERRUPTIÓN AEOI.

Si en ICW4 AEOI=1 el 8259 funcionará continuamente en el modo AEOI hasta que sea vuelto a programar por ICW4. En este modo el 8259 ejecutará una operación EOI no específica en el frente del último impulso \overline{INTA} . Este modo sólo debe usarse cuando no se precisa una estructura de interrupciones multinivel dentro de un único 8259. El modo AEOI sólo se puede usar en un maestro y no en un esclavo.

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	SFNM	BUF	M/S	AEOI	μ PM

ICW4

Ilustración 151 - Palabra ICW4

7.8.4.- MODO DE ROTACIÓN AUTOMÁTICA, DISPOSITIVOS DE IGUAL PRIORIDAD.

Es útil en aplicaciones con un conjunto de dispositivos que requieren interrupciones con igual prioridad. En este modo un dispositivo como por ejemplo un teclado, después de haber sido servido, recibe la prioridad más baja de forma que un dispositivo que solicita una interrupción, tendrá que esperar en el peor de los casos hasta que se sirva como mucho una vez cada uno de los otros siete dispositivos restantes.

Este funcionamiento se explica a continuación con un ejemplo. Se suponen cinco teclados conectados a las líneas IR0...IR4. En la figura se aprecia que el indicativo IS2 correspondiente al teclado solicitante de servicio IR2 está activado y es el de prioridad más alta entre los solicitantes, dos en este caso, IS4 e IS2. En la siguiente figura se observa como IR2 ha sido ya servido y las demás prioridades han rotado en consecuencia.

- Antes de la rotación de prioridad.

	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Estado IS	0	0	0	1	0	1	0	0
Estado de prioridad	7	6	5	4	3	2	1	0

- Después de la rotación de prioridad.

	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Estado IS	0	0	0	1	0	0	0	0
Estado de prioridad	4	3	2	1	0	7	6	5

Tabla 15 - Niveles de rotación con igual prioridad

Existen dos formas de realizar la rotación automática utilizando OCW2: la rotación sobre comando no específico EOI, R=1, SL=0 y EOI=1 y la rotación en modo automático EOI, que es activada por R=1, SL=0, EOI=0 y borrada por R=0, SL=0 y EOI=0.

OCW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	R	SL	EOI	0	0	L2	L1	L0

Ilustración 152 - Palabra OCW2

7.8.5.- MODO DE ROTACIÓN ESPECÍFICA, PRIORIDAD ESPECÍFICA.

En este modo el programador puede cambiar las prioridades fijando la prioridad más baja, quedando así puestas las demás prioridades. Por ejemplo, si se programa IR5 como el dispositivo de prioridad más baja, IR6 tendrá automáticamente la más alta. El comando de activación de prioridad se emita en OCW2 donde hay que poner R=1, SL=1 y en L2...L0 el código binario del nivel de prioridad del dispositivo que deba tener más baja prioridad. En este modo el estado interno se actualiza por control de software durante OCW2. Sin embargo, es independiente del comando de final de interrupción EOI también puesto con OCW2. Por tanto, los cambios de prioridad se pueden realizar durante un comando EOI usando en OCW2 la rotación sobre comando EOI específico R=1, SL=1, EOI=1 y L2...L0 igual al nivel IR a recibir más baja prioridad.

OCW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	R	SL	EOI	0	0	L2	L1	L0

Ilustración 153 - Palabra OCW2

7.8.6.- MODO DE MÁSCARAS DE INTERRUPTIO.

Cada entrada IR puede enmascarse individualmente cargando con OCWI el registro de máscaras de interrupción (IMR) de una forma adecuada. Cada bit a "1" en el IMR

enmascara el canal de interrupción correspondiente. El enmascarar un canal IR no afecta al funcionamiento de los demás canales.

Algunas aplicaciones pueden requerir que durante la ejecución de su rutina de servicio se altere dinámicamente la estructura de prioridades. Un caso ejemplo podría ser que la rutina necesite inhibir las peticiones de menor prioridad durante una parte de su ejecución, pero permitir algunas de ellas durante otra parte. Este ejemplo en principio no tendría dificultad alguna si no se reconoce ninguna petición de interrupción, pues en el momento deseado se enmascararían unas mediante OCWI y cuando fuera necesario se quitarían las máscaras del mismo modo. Pero, si se reconoce una petición de interrupción y no se pone a "0" su bit ISR mediante un comando EOI mientras se ejecuta una rutina de servicio, el 8259 tendría inhibidas todas las peticiones de menor prioridad y la rutina no tendría forma de poder permitir las.

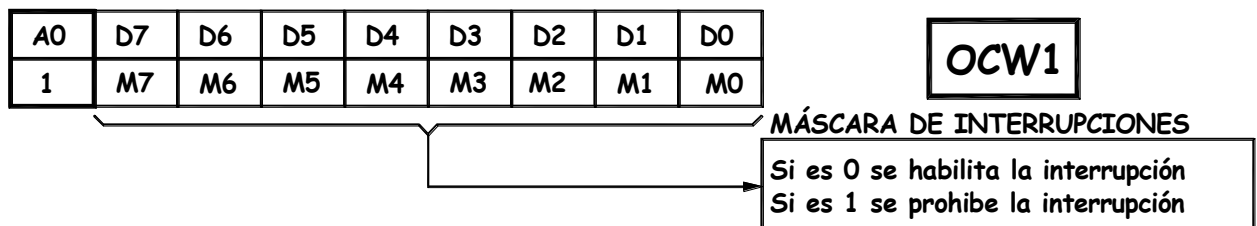


Ilustración 154 - Palabra OCW1

Es aquí donde tiene gran utilidad el modo de máscara especial. En este modo, que se programa mediante OCW3 cuando OCW1 pone a "1" algún bit de máscara, se inhiben otras interrupciones en este nivel y se permite interrupciones en los otros niveles, que no estén enmascarados, tanto de mayor prioridad como de menor. De esta manera, se puede permitir selectivamente cualquier interrupción cargando el registro de máscara. El modo de máscara especial se activa por OCW3 con ESMM=1 y SMM=1 y se borra con ESMM=1 y SMM=0.

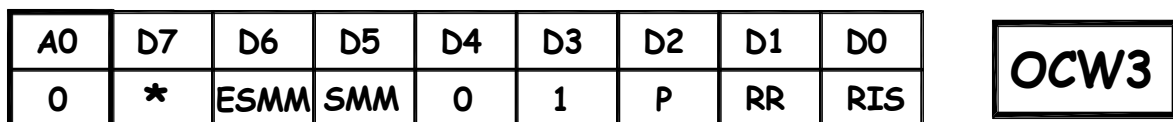


Ilustración 155 - Palabra OCW3

7.8.7.- MODO DE COMANDO POLL O SONDEO.

En este modo ocurre una de las siguientes situaciones: o no se utiliza la salida INTR o está borrado el flip-flop de permiso de interrupciones interno al microcontrolador, impidiendo su entrada de interrupción. El servicio se lleva a cabo mediante software usando un comando poll o sondeo que se emite poniendo P=1 en OCW3. El 8259 trata el siguiente impulso \overline{RD} que le llega, es decir, $\overline{RD}=0$ y $\overline{CS}=0$ como un reconocimiento de interrupción. Si hay alguna petición de interrupción en el IRR evalúa la prioridad, pone a "1" el bit ISR apropiado y envía al bus de datos durante \overline{RD} el byte indicado en la figura.

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	W2	W1	W0

W2-W1-W0	Código binario del nivel de mayor prioridad que está solicitando servicio
D7	Igual a 1 si hay interrupción

Tabla 16 - Comando sondeo

Este modo se usa si hay un comando de rutina común a varios niveles, de forma que no se necesita la secuencia \overline{INTA} , ahorrando así espacio en la memoria. También se utiliza para expandir el número de niveles de prioridad a más de 64.

7.8.8.- MODOS DE DISPARO POR FLANCO Y POR NIVEL.

Este modo se programa usando el bit LTIM de ICW1. Si LTIM=0 la petición de interrupción será reconocida por flanco, es decir, por la transición de bajo a alto de una entrada IR. Seguidamente, la entrada IR puede permanecer alta sin generar otra interrupción. Si LTIM=1 una petición de interrupción será reconocida cuando una entrada IR está a nivel alto, no siendo necesaria la detección de un flanco. La petición de interrupción debe anularse antes de que el correspondiente bit del ISR, después de darse servicio, se vuelva a poner a 0 para así impedir que tenga lugar una nueva interrupción.

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	A5	1	LTIM	ADI	SGNL	IC4

ICW1

Ilustración 156 - Palabra ICW1

En ambos modos las entradas IR deben permanecer altas hasta después del frente de bajada del primer \overline{INTA} . Si la entrada IR va a nivel bajo antes de este tiempo ocurrirá, por defecto, un IR7 cuando el microcontrolador reconozca la interrupción. Esto puede usarse como una protección para detectar falsas interrupciones creadas por ruidos. Para ello se usa en la rutina IR7 una instrucción RETURN y así se ignora la interrupción.

7.8.9.- MODO SPECIAL FULLY NESTED.

Este modo se emplea en el caso de un gran sistema donde se utilice el modo cascada y se tenga que mantener la prioridad dentro de cada esclavo. En este caso se programará al maestro en el modo "special fully nested" con ICW4. Este modo funciona de manera parecida al modo "fully nested" normal pero han de tenerse en cuenta las siguientes excepciones:

- En el modo "fully nested" normal, cuando una interrupción de un esclavo está siendo servida, ninguna otra petición del mismo esclavo ni de prioridad mayor se atenderá. Sin embargo, en el modo especial las interrupciones de más alta prioridad dentro de o de los esclavos que está sirviendo seguirán siendo reconocidas por el maestro.

- Cuando finaliza la rutina de interrupción el software tiene que chequear si la interrupción servida era la única de ese esclavo. Esto se hace enviando al esclavo un comando EOI no específico, leyendo su registro ISR y viendo los bits que están a 0. Si está vacío, o sea todos a 0, puede enviarse un comando EOI también al maestro, en caso contrario, no será mandado ningún comando EOI al maestro.

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	SFNM	BUF	M/S	AEOI	μPM

ICW4

Ilustración 157 - Palabra ICW4

7.8.10.- MODO BUFFERED O AMPLIFICADOR.

Cuando el 8259 se emplea en un sistema grande donde se necesitan buffers amplificadores sobre el bus de datos para que lo ataquen y se usa el modo cascada, existen problemas a la hora de hacer válidos los buffers necesarios en cada momento. En el modo "buffered" la línea $\overline{SP/EN}$ será una salida activada por el 8259 cuando deban permitirse sus salidas del bus de datos, validando con ella los buffers.

La línea $\overline{SP/EN}$ tiene la función de decir si el 8259 era esclavo, cuando estaba a "0", o si era maestro, cuando estaba a "1". En este modo ya no se puede decir al 8259 si es esclavo o maestro mediante dicha línea. Esta información se le transmite por software en la palabra de comando ICW4 durante la iniciación. Así, con el bit 3 de ICW4, se programa al 8259 en este modo y con el bit 2 se le dice si es esclavo o maestro.

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	SFNM	BUF	M/S	AEOI	μPM

ICW4

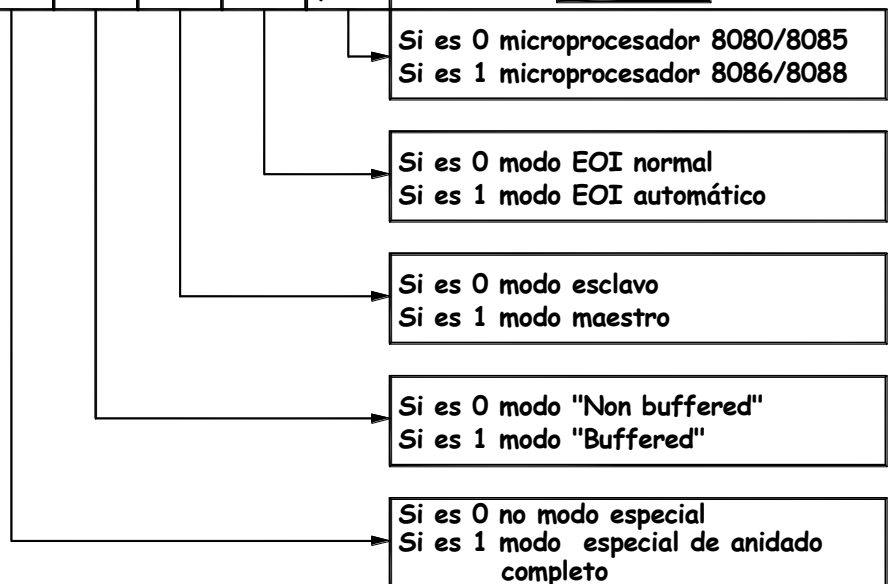


Ilustración 158 - Palabra ICW4

7.8.11.- MODO CASCADA.

El 8259 puede interconectarse fácilmente en un sistema de un maestro con ocho esclavos para manipular 64 niveles de prioridad. El maestro controla a los esclavos a través de un bus de cascada de tres líneas. El bus de cascada actúa como un “chip select” de los esclavos y es activado durante la secuencia \overline{INTA} .

En la configuración en cascada, las salidas de interrupción del esclavo se conectan a las entradas de petición de interrupción del maestro. Cuando se activa una línea de petición de interrupción de un esclavo y más tarde es reconocida, el maestro permitirá que el esclavo correspondiente libere la dirección de la rutina del dispositivo durante los pulsos 2 y 3 de \overline{INTA} .

Las líneas del bus de cascada están normalmente a nivel bajo, pero desde el flanco del primer impulso \overline{INTA} hasta el flanco del tercero el maestro pondrá en ellas el código de dirección del esclavo del que haya sido reconocida la interrupción. Cada 8259 del sistema debe seguir una secuencia de iniciación independiente y puede ser programado para trabajar en modos diferentes. Un comando EOI debe ser emitido dos veces, una para el maestro y otra para el esclavo correspondiente. Es necesario un decodificador de direcciones para activar la entrada \overline{CS} de cada 8259. Las líneas de cascada del 8259 maestro se activan para cualquier entrada de interrupción, incluso si no hay ningún esclavo conectado a esa entrada.

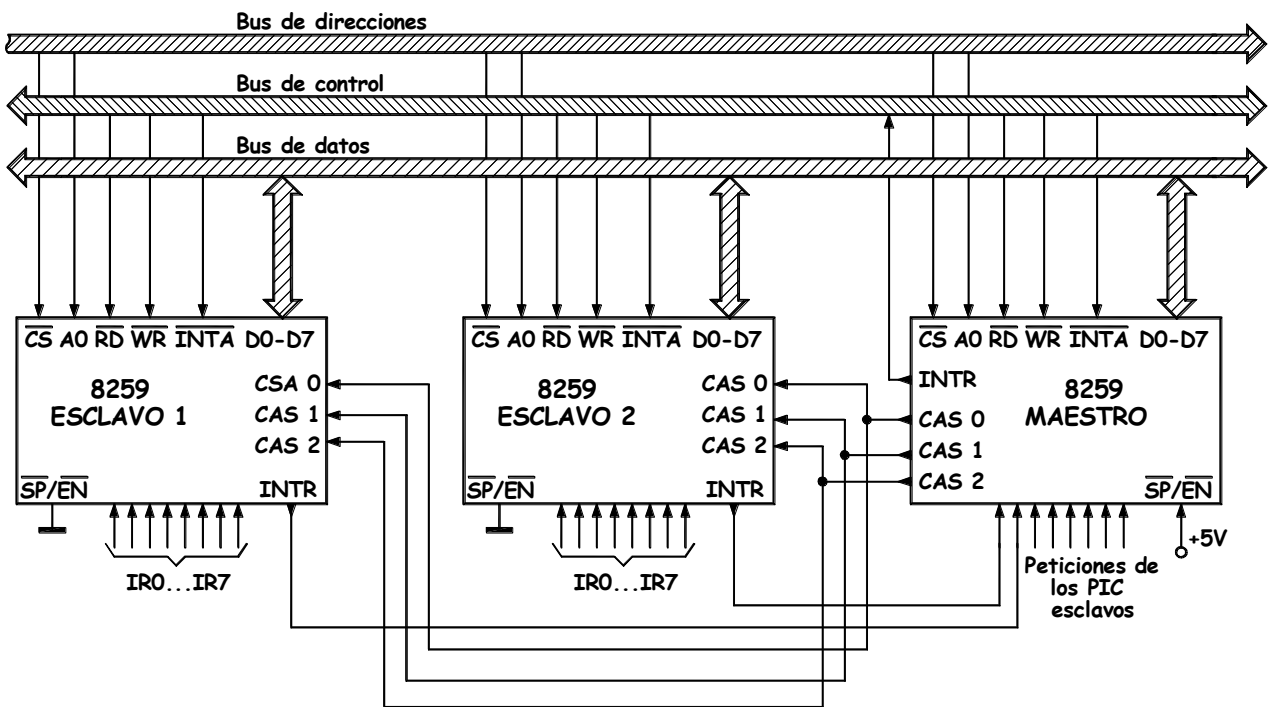


Ilustración 159 - Acoplamiento en cascada de dos PIC esclavos y un maestro

7.8.12.- LECTURA DEL ESTADO DEL 8259.

Se puede leer el estado de varios registros internos para actualizar la información del usuario sobre el sistema. Mediante OCW3 se leen el IRR y el ISR y con OCW1 el IMR.

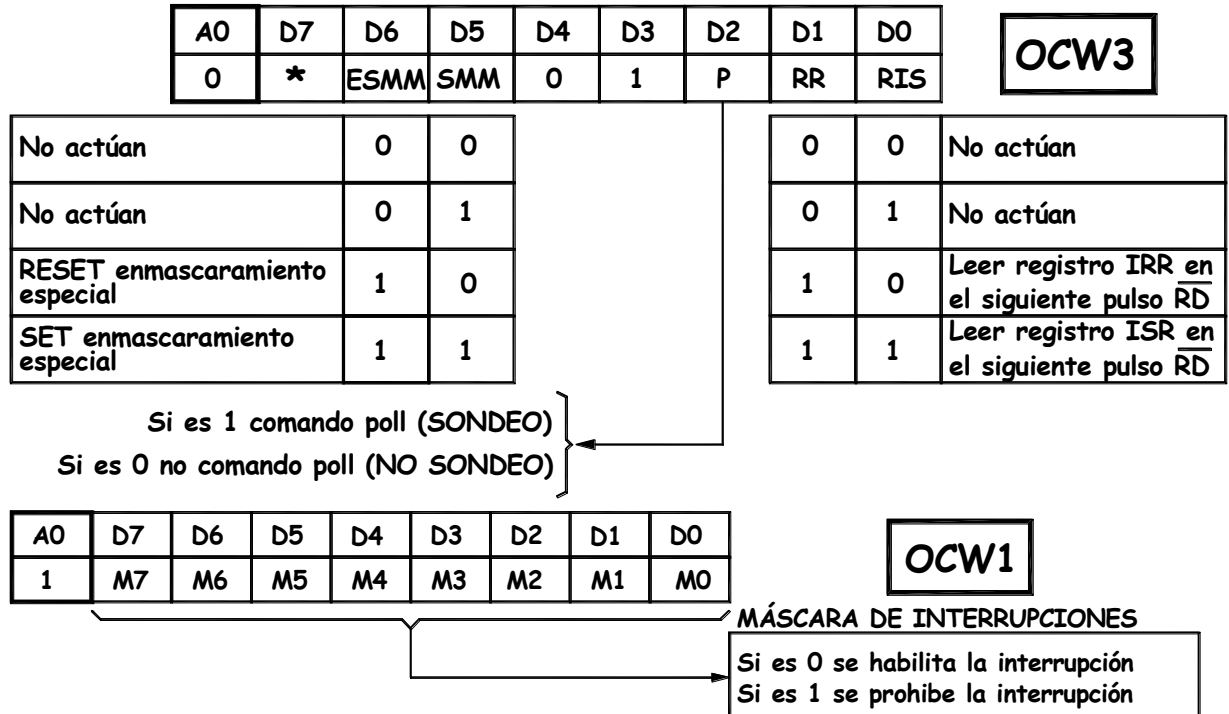


Ilustración 160 - Palabras de comando OCW1 y OCW3

- El registro de petición de interrupción IRR se puede leer cuando, antes de un impulso \overline{RD} , se emite un comando de lectura de registro en OCW3, RR=1 y RIS=0.
- El registro de servicio ISR se puede leer como en el caso anterior, enviando antes de un \overline{RD} , la palabra OCW3 con RR=1 y RIS=1. En ambos casos el 8259 recuerda si OCW3 ha seleccionado anteriormente una lectura del IRR o del ISR, por tanto no es necesario escribir una OCW3 cada vez que se vaya a leer del mismo registro. Esto no se cumple cuando se trabaja en modo poll o sondeo.
- Para leer el registro de máscara de interrupción no se necesita OCW3 pues el bus de datos contendrá el IMR siempre que sea activa \overline{RD} y que A0=1, byte OCW1.

Un ejemplo de la utilidad que supone poder leer los registros es el siguiente: en la ejecución de un programa se lee el registro, se compara con un valor dado y como resultado de la comparación, mediante una instrucción de salto condicional, se va a la ejecución de una rutina u otra; lo que habrá permitido tomar decisiones que pueden ser muy útiles en un sistema concreto.

7.9.- EL CIRCUITO 8259 DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

En la placa principal de control se encuentran implementados dos circuitos integrados 8259 configurados como maestro y esclavo. En total permiten la gestión de quince fuentes de interrupción externas de las cuales se utilizan cuatro, pertenecientes al 8259 maestro. Las señales de interrupción gestionadas proceden de los sensores inductivos y los finales de carrera ubicados en la muñeca del robot y pasan a través de una serie de inversores antes de conectarse a las entradas del 8259.

Para la gestión correcta de las interrupciones se debió de programar el 8259 en modo rotativo. De esta forma, una vez atendida una interrupción el 8259 la etiqueta con la prioridad más baja quedando libre para atender a las demás interrupciones. En la imagen inferior se observa la localización del 8259 maestro y el 8259 esclavo en la placa principal.

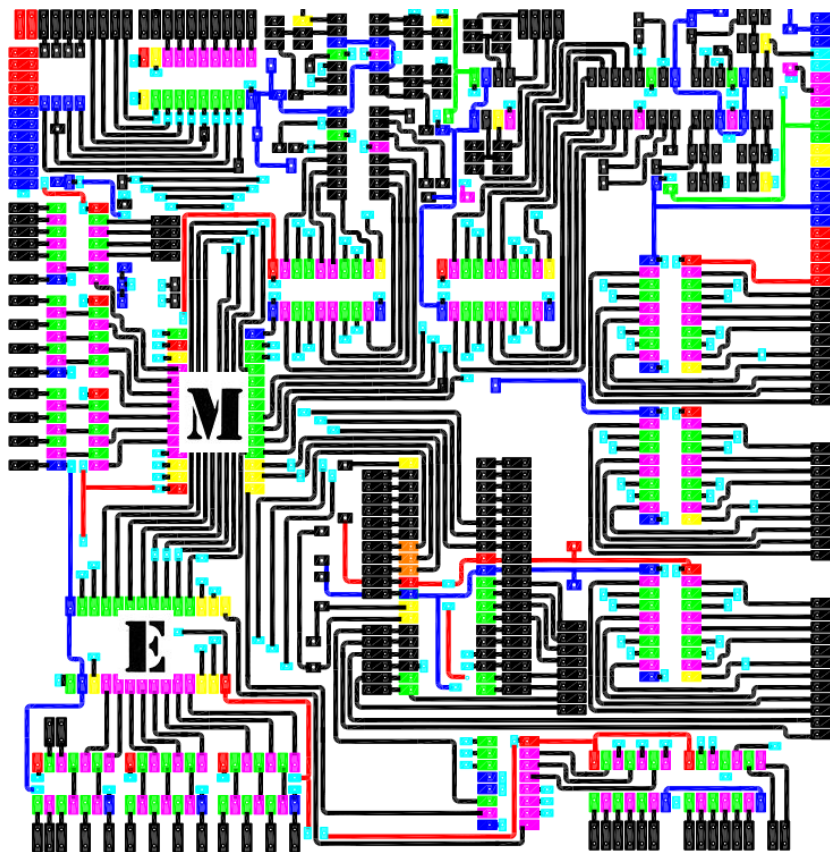


Ilustración 161 - Localización de los dos 8259 en la placa principal.

8.- MÓDULO LCD.

Los módulos LCD (“Liquid Crystal Display”) son compactos y necesitan muy pocos componentes externos para un funcionamiento correcto. La función principal de estos módulos es la de visualizar los caracteres deseados por el usuario. Existen módulos de distintas dimensiones que van desde 1 a 4 líneas y de 6 a 80 caracteres por línea. El módulo utilizado es de 4x16 líneas.

En la imagen inferior se observa el aspecto y posición de los pines de un módulo LCD estándar. Existen también módulos LCD que disponen de 16 pines, son los llamados LCD con retroiluminación donde el pin 15 (+5 V) corresponde al ánodo y el pin 16 (masa) al cátodo de la retroiluminación.

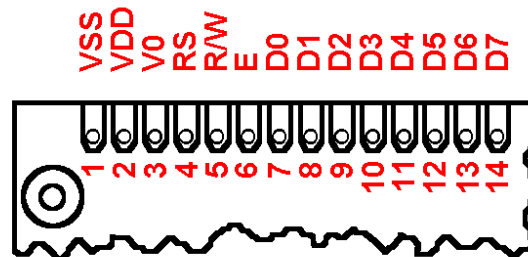


Ilustración 162 - Pines de un módulo LCD estándar.

Estos dispositivos vienen gobernados por un microcontrolador que normalmente va incorporado sobre la misma placa de circuito impreso que soporta el LCD. En el mercado es muy habitual encontrarse con el controlador HD44780 de Hitachi, que se describe a continuación. El controlador se encarga de gestionar el display líquido: polarizar los puntos de la pantalla, generar los caracteres, desplazar la pantalla, mostrar el cursor,... El usuario necesita conocer una serie de instrucciones de alto nivel (limpia display, posiciona cursor,...) que le permiten mostrar mensajes en la pantalla de forma sencilla. Para comunicarse con el controlador del display se dispone de una interfaz de tipo paralelo al exterior de fácil conexión a otros microcontroladores. En la imagen inferior se observa el controlador HD44780 de Hitachi.



Ilustración 163 - Vista trasera del display LCD. Controlador HD44780 incrustado.

Los caracteres que se pueden representar están formados por una matriz de puntos que vienen almacenados en la zona de memoria ROM dentro del propio controlador.

8.1.- CONEXIONES DE UN MÓDULO LCD.

El número de pines de un display LCD es normalmente de 14 o 16 pines en caso de ser retroiluminado. En la tabla siguiente se muestra el significado de las señales de cada pin y su interconexión con el microcontrolador PIC16F877A conectado al módulo LCD a través de una de las básculas unidireccionales 74LS274 descritas anteriormente.

Pin	Conexión	Descripción
1 - Vss	Vss	Masa

2 – Vdd	Vdd	Alimentación 5v.
3 – Vo	Vo	Control de contraste de la pantalla.
4 – RS	RC2	Selección del registro de control/registro de datos: RS =0 Selección del registro de control. RS=1 Selección del registro de datos.
5 – RW	RE1/RE2	Señal de lectura/escritura: R/W=0 El Módulo LCD en modo escritura. R/W=1 El Módulo LCD en modo lectura.
6 – E	RC1	Señal de activación del módulo LCD: E=0 Módulo desactivado. E=1 Modulo activado.
7-14 – D0-D7	RD0-RD7	Bus de datos bidireccional. A través de estas líneas se realiza la transferencia de información entre el módulo LCD y PIC.

La línea RS del módulo se conecta con el bit 2 del puerto C del microcontrolador, la línea E al bit 1 del mismo puerto, la línea de lectura se encuentra implementada sobre el bit 2 del puerto E y la línea de lectura sobre el bit 1 del mismo puerto. Hay tres tipos de señales en el LCD: de alimentación, de control y de datos.

- Señales de alimentación

Son tres señales correspondientes a los pines 1, 2 y 3. El pin 1 corresponde a la masa, el 2 a la alimentación positiva (+5V) y el 3 al juste de contraste. Para actuar sobre el contraste de la pantalla debe instalarse un potenciómetro de 10k tal y como se observa en la imagen inferior. Cuanto más cercano a masa esté la tensión en el pin 3 (Vo) mayor será el contraste.

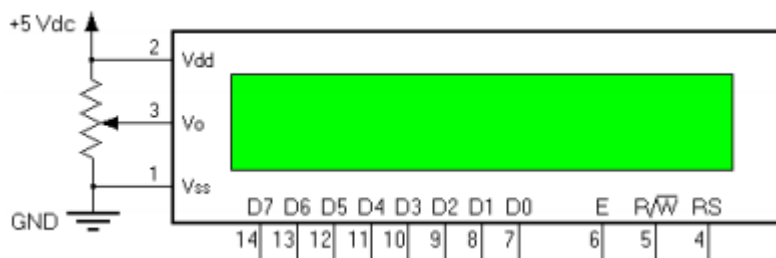


Ilustración 164 - Instalación de un potenciómetro de contraste para la pantalla LCD.

- Señales de control

Los pines 4, 5 y 6 forman las señales de control del LCD. El pin 4 (RS) sirve para seleccionar el registro de datos (DR) o el de instrucciones (IR), poniendo RS=1 o RS=0 respectivamente. EL pin 5 permite leer (R/W(negada)=1) o escribir (R/W(negada)=0) tanto datos como instrucciones. Por último, el pin 6 permite habilitar con E=1 o deshabilitar el display (E=0). Solo estando habilitado nos podemos comunicar con él.

- Señales de datos

Los pines 7 al 14 forman un bus de datos bidireccional de 8 bits (DB7 – DB0) por donde se pueden escribir datos e instrucciones y se puede leer el estado del display (si está o no ocupado, la posición actual del cursor,...).

8.2.- INSTRUCCIONES DEL MÓDULO LCD.

Las instrucciones del módulo LCD se resumen a continuación.

Instrucción	Código										Descripción	Tiempo máx. de ejecución	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Clear display (Borra display)	0	0	0	0	0	0	0	0	0	0	1	Borra el display y coloca el cursor en la posición 0.	82us - 1,64 ms
Return home (Vuelve a inicio)	0	0	0	0	0	0	0	0	0	1	*	Cursor en la posición de inicio e inicia desplazamiento.	40us - 1,64ms
Entry mode set (Modo inicial de funcionamiento)	0	0	0	0	0	0	0	0	1	I/D	S	Establece sentido de desplazamiento de la info. En el display.	40us
Display ON/OFF control (Control de encendido del display)	0	0	0	0	0	0	0	1	D	C	B	Pone en ON/OFF tanto el display D-0 (off) o C-1 (on) y establece si parpadea o no B-0 (off) o B-1 (on).	40us
Cursor display shift (Control del cursor)	0	0	0	0	0	0	1	S/C	R/L	*	*	Mueve el cursor y desplaza el display sin cambiar el contenido de la DDRAM.	40us
Function set (Establecimiento de funciones)	0	0	0	0	1	DL	N	F	*	*	*	Establece el tamaño de interface con el bus de datos (DL), el número de líneas del display (N) y la fuente de los caracteres.	40us

CGRAM address set (Establecimiento de la dirección del CGRAM)	0	0	0	1	Dirección CGRAM	Establece la dirección CGRAM a partes de la cual se almacenan los caracteres de usuario.	40us
DDRAM address set (Establecimiento de la dirección del DDRAM)	0	0	1		Dirección de la DDRAM	Establece la dirección de la DDRAM a partir de la cual se almacenan los datos a visualizar.	40us

Read Busy Flag and Address (Leer flag Ocupado y dirección)	0	1	BF		Dirección DDRAM o CGRAM	Lectura del flag de Busy e indicación de la dirección de la CGRAM o DDRAM empleada.	1us
Write data into the CGRAM or the DDRAM (Escribir en la memoria CGRAM o DDRAM)	1	0			Dato a escribir	Escribe en DDRA o CGRAM los datos que se quieren presentar en el LCD.	40us
Read data from the CGRAM or the DDRAM (Leer datos de la CGRAM o DDRAM)	1	1			Dato a leer	Lee de la DDRAM o la CGRAM los datos que se direccionen.	1
					I/D – 1 incrementa la dirección del cursor. I/D – 0 decreuenta la dirección del cursor. S – 1 desplaza la visualización cada vez que se escribe un dato. S/C – 1 se desplaza la visualización. S/C – 0 se desplaza el cursor. R/L – 1 desplazamiento a la derecha. R/L – 0 desplazamiento a la izquierda. DL – 1 trabaja con el bus de datos de 8 bits. DL – 0 trabaja con el bus de datos de 4 bits. N – 1 la presentación se hace en dos líneas.	-DDRAM: Display Data RAM -CCRAM: Generador de caracteres RAM -ACC: Dirección de CGRAM -ADD:	

		N – 0 la presentación se hace en 1 línea. F – 1 caracteres de 5x10 puntos (“dots”). F – 0 caracteres de 5x7 puntos (“dots”). BF – 1 LCD ocupado en una operación interna. BF – 0 LCD disponible para aceptar instrucciones. D – 1 pantalla activa (ON). C – 1 cursor activo (ON). * indeterminado.	Contador de direcciones usado para la CGRAM y DDRAM.	
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------	--

8.3.- MÓDULO LCD DEL PROYECTO.

Como ya se dijo anteriormente, el módulo LCD del proyecto es de 4 líneas y 16 caracteres. Las direcciones de memoria DDRAM para las cuatro líneas son las siguientes:

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
10	11	12	13	14	15	16	17	18	19	2A	2B	2C	2D	2E	2F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F

En la imagen inferior se observa el módulo LCD real del proyecto.

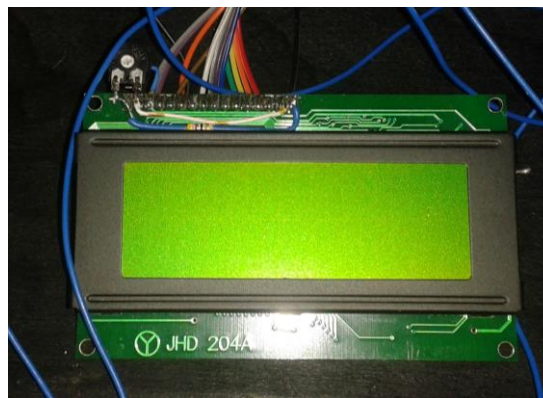


Ilustración 165 - Módulo LCD del proyecto.

8.3.1.- EL MÓDULO LCD DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

La pantalla LCD sirve para enviar pequeños avisos al usuario del correcto funcionamiento del sistema. No se realiza el envío de información de importancia del proceso de control puesto que esta información está fácilmente accesible para el usuario a través de la interfaz gráfica de control.

Sin embargo, la pantalla fue de vital importancia durante todo el proceso de diseño e implementación del sistema de control. Mediante el envío de diversos caracteres a la pantalla se realizaron numerosas pruebas de ensayo error con el sistema. Además, constituye la forma más fiable de conocer el estado del microcontrolador puesto que, en el caso de el

microcontrolador se desconecte del sistema, la pantalla deja de responder inmediatamente a las órdenes del usuario.

Sección III

PLACAS AUXILIARES

1.- INTRODUCCIÓN

Como ya se dijo anteriormente, las funciones de la placa principal se complementan con funciones desarrolladas por placas auxiliares conectadas a ella. Estas placas son cuatro en total. En primer lugar se encuentra la placa 8255 – 8253, en la que se conectan entre sí y con la placa principal (a través del bus de datos) cuatro circuitos 8253 y dos circuitos 8255. Los 8253 son contadores triples que se encargarán de realizar el conteo de los pulsos de los encoders. Los 8255 son dispositivos de entrada/ salida programables que se utilizan como puertos de comunicación. En segundo lugar, se encuentra la placa de gestión de motores que alberga los convertidores digital- analógicos DAC0800 y los amplificadores operacionales del circuito TL074. En esta placa se realiza el control de la velocidad y sentido de giro de los motores mediante la conversión de un byte de información que llega a la placa a través de un 8255 proveniente del bus de datos del microcontrolador. En tercer lugar, está la placa de tratamiento de señales. En esta placa se acondiciona la señal de los encoders mediante una serie de filtros, inversores y optoacopladores para poder conectarla a los circuitos de conteo 8253. De esta forma, se evita ruido en la señal y picos que distorsionarían o incluso podrían llegar a imposibilitar el conteo de los pulsos. En último lugar, la placa AND – OR se diseñó en el marco del manejo correcto del conteo de pulsos de los encoders por parte de los 8253. Mediante una serie de puertas lógicas AND y OR se logra establecer un control del paso de las señales de los encoders a la entrada de los contadores. Así, se tiene la posibilidad de iniciar el conteo en el momento justo (cuando se sabe que los 8253 están respondiendo correctamente) y de detenerlo a voluntad, pudiendo realizar una lectura de las señales en el mismo instante de tiempo.

2.- PLACA 8255 – 8253.

2.1.- DISEÑO DE LA PLACA

Esta placa se diseñó para un proyecto anterior a este y por sus características fue adecuada para insertar dentro del sistema de control diseñado. El proceso de fabricación utilizado fue el diseño en AutoCAD y posteriormente el uso del método fotográfico.

Para seguir este procedimiento se debe disponer de placas presensibilizadas, es decir, con una película de emulsión fotosensible encima del cobre. El tipo de emulsión fotosensible más usual es la fotopositiva, que es aquella en la que las partes que reciben la radiación ultravioleta se reblandecen y se disuelven tras un proceso llamado revelado, que se realiza después de insolar la placa a través de su dibujo en papel de plástico opaco. Una insoladora posee unos tubos fluorescentes que emiten luz actínica (rayos ultravioleta capaces de ejercer una acción química donde sean aplicados) y una bomba de vacío para asegurar un perfecto acoplamiento entre el patrón y la placa, sin que se formen burbujas de aire entre ellos. El revelado consiste en sumergir la placa, una vez insolada, en una cubeta con agua y una base fuerte (como la sosa cáustica) eliminándose de esta manera la emulsión fotosensible en aquellas zonas que hayan recibido la luz y permaneciendo en las que han sido cubiertas. De este modo sólo permanecerá el cobre en las zonas en las que haya permanecido la emulsión fotosensible que es resistente al ácido. Por lo tanto, en una placa fotopositiva, las futuras pistas y zonas que no deban ser corroídas serán negras en el patrón, siendo el resto transparente. Con las placas que utilizan emulsión fotonegativa todo sucederá a la inversa. Serán atacadas por el ácido y disueltas las zonas que no hayan sido expuestas a la luz, permaneciendo las otras.

Se debe de trabajar cuidadosamente si se tienen placas presensibilizadas, guardarlas en su bolsa original, de color negro, y en un sitio oscuro, para que no reciban la luz ambiental. El tiempo de exposición a la luz está comprendido entre 1 y 5 minutos, dependiendo de la lámpara utilizada y del tipo de placa. Una vez realizada la insolación, en la placa no se observa ningún dibujo visible, pero habrá quedado transferido el patrón sobre la película fotosensible de la placa.

Terminada la insolación, se procede al revelado con la base. En este caso, el procedimiento de transferencia fotográfico consistió en utilizar una placa con película fotosensible positiva y someterla a un proceso de insolación en una máquina insoladora, El patrón se pone al lado del cual procederán las radiaciones ultravioleta y la placa se ubica debajo de él. Los patrones opacos se obtienen a través de AutoCAD y se imprimen en papel transparente para posteriormente adherirlos encima de la placa fotosensible.

El diseño en AutoCAD se realiza siguiendo el mismo procedimiento que en la placa principal. Se diseña tanto el lado de componentes como el lado de pistas. La cara de componentes es donde se encuentran colocados los componentes y los conectores de entrada-salida de la placa. La cara de pistas es aquella que alberga las pistas conductoras impresas y los "pads", que son las superficies para el contacto o soldadura de los terminales de los componentes.

En las imágenes inferiores pueden observarse los diseños del lado de componentes y del lado de pistas, respectivamente.

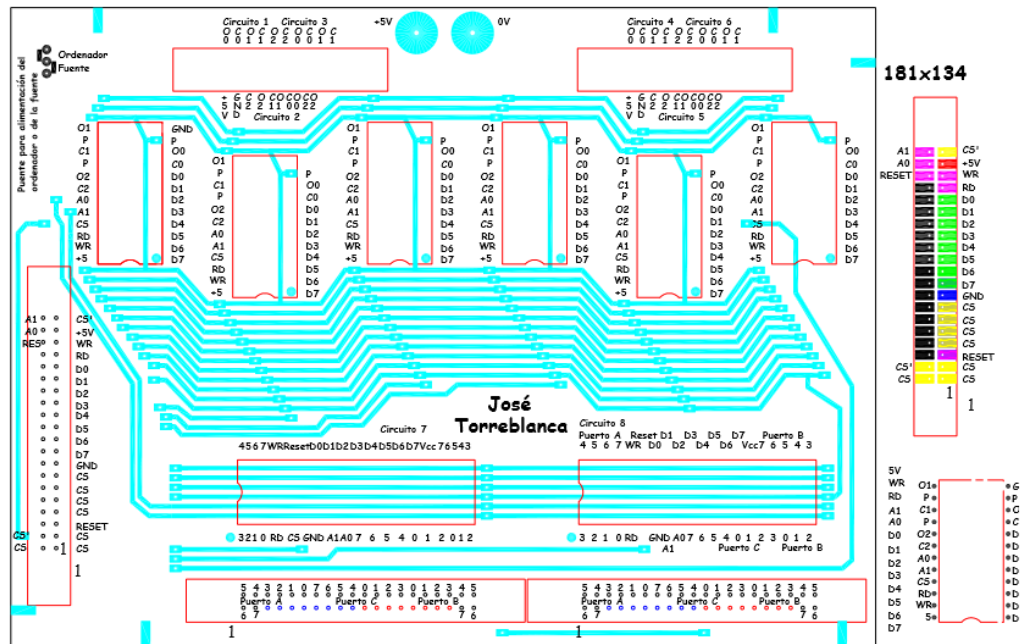


Ilustración 166 - Placa 8253-8255: Cara de componentes

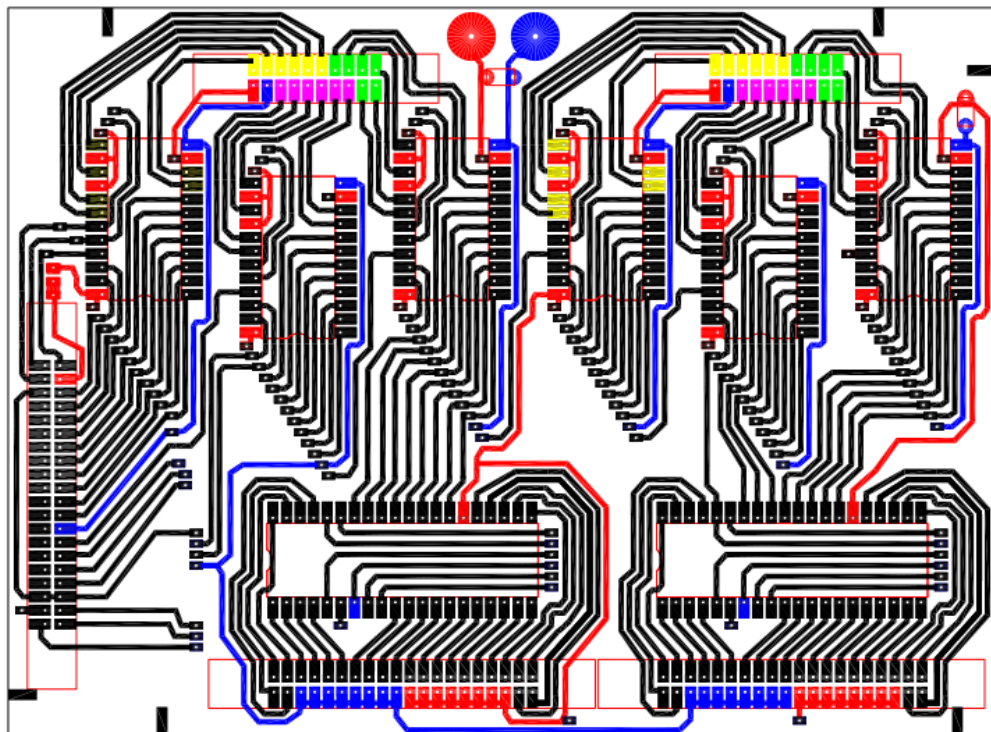


Ilustración 167 - Placa 8253-8255: Cara de pistas

En la cara de pistas el color azul simboliza las pistas conectadas a masa y el color rojo simboliza la alimentación de los circuitos a +5V. En la imagen inferior se observa la placa terminada.

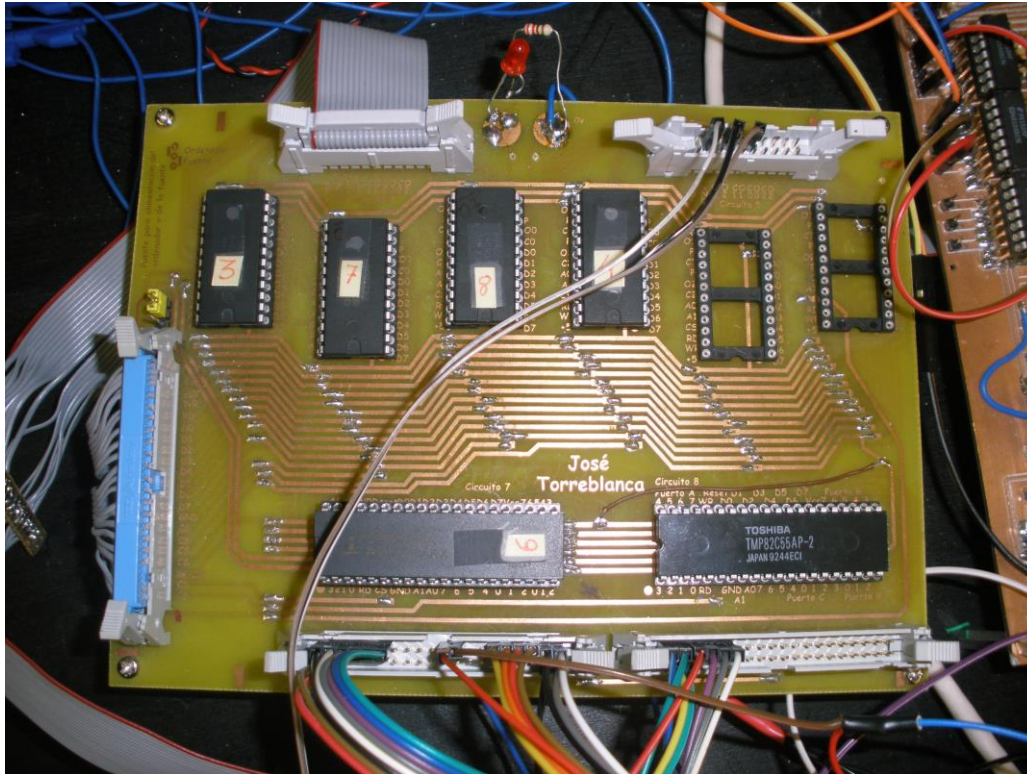


Ilustración 168 - Placa 8255 - 8253 terminada.

2.2.- COMPONENTES DE LA PLACA

2.2.1.- CIRCUITO 8253. PERIFÉRICO PROGRAMABLE DE CONTADORES-TEMPORIZADORES.

Este periférico contiene tres contadores-temporizadores programables de 16 bits cada uno y se conecta con el microcontrolador como un periférico de puertos de entrada-salida.

Este circuito integrado permite generar tiempos de retardo muy precisos mediante el control por programa. Los tres contadores de 16 bits admiten frecuencias de entrada comprendidas entre 0 y 4 MHz (hasta 240 millones de pulsos por minuto). El sistema de control por software de la carga y arranque de los contadores permite conseguir una gran precisión en dichos contajes y temporizaciones. El 8253 es un medidor de eventos programable específicamente diseñado para usarse con un sistema microcomputador. Su función de forma general, es la de un dispositivo que puede medir varios intervalos, ya sean de tiempo o eventos, y que puede ser tratado como una red de puertos de entrada o salida al programarse por software.

Este circuito resuelve uno de los problemas más comunes en algunos de los sistemas microcomputador, como es la generación de retardos exactos de tiempos controlados mediante software, en lugar de retrasar el tiempo que estará la señal en el circuito, lo que hace el programador para que el 8253 produzca retardos exactos de tiempo es cargar uno de los contadores del 8253 con la cantidad deseada y después de una orden el 8253

decrementará el valor del contador e interrumpirá al microprocesador cuando su cuenta haya llegado a cero. En la imagen inferior puede observarse la estructura interna de este circuito.

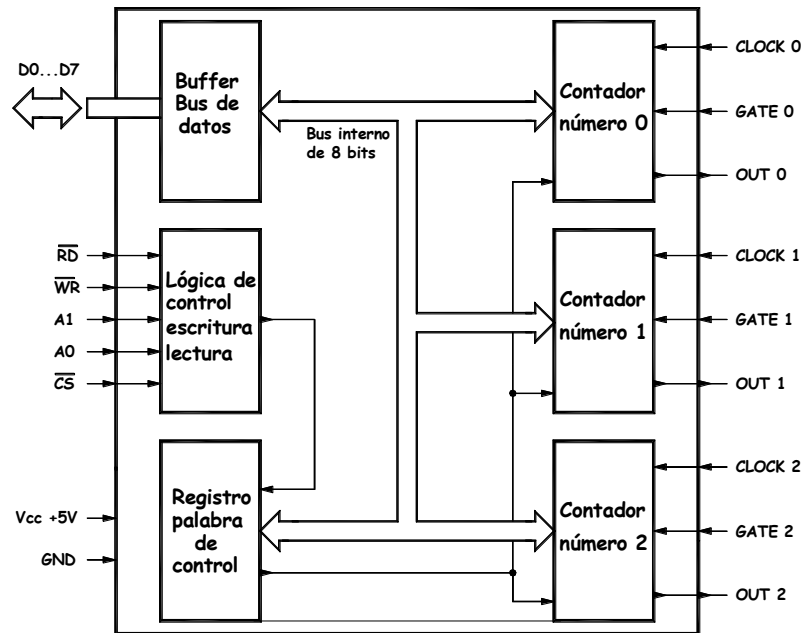


Ilustración 169 - Estructura interna del 8253

El buffer del bus de datos contiene ocho amplificadores bidireccionales con control de alta impedancia. Su función es conectar el 8253 con el bus de datos del sistema. Además, se encarga de transmitir o recibir datos de acuerdo con las instrucciones IN, OUT ejecutadas por el microcontrolador. Existen tres funciones básicas que se realizan a través de este bloque: la programación de modos, la carga de los datos en los registros y el leer los valores contabilizados.

La lógica de lectura-escritura se encarga de controlar todas las funciones del 8253 y está gobernado por las señales de control que vienen del microcontrolador a través del bus de control. El registro de la palabra de control sirve para almacenar el funcionamiento del circuito, es decir, la forma en que se quiere que trabajen los contadores-temporizadores. En este bloque se encuentran además los registros para cada contador.

Los contadores 0, 1 y 2 de 16 bits son idénticos e independientes entre sí, permitiendo poder trabajar con modos independientes y velocidades de conteo diferentes y no sincronizadas. Las funciones de conteo pueden ser en binario o en BCD, con la señal GATE o sin ella y la configuración de las líneas de entrada y salida vienen programadas por la palabra de control almacenada en el registro palabra de control.

El programador, mediante operaciones de lectura tiene acceso al contenido de cada uno de los contadores. Este periférico dispone de las señales de control y los comandos necesarios para que las operaciones de lectura no afecten al conteo. En la figura inferior puede observarse el diagrama detallado de los terminales del circuito.

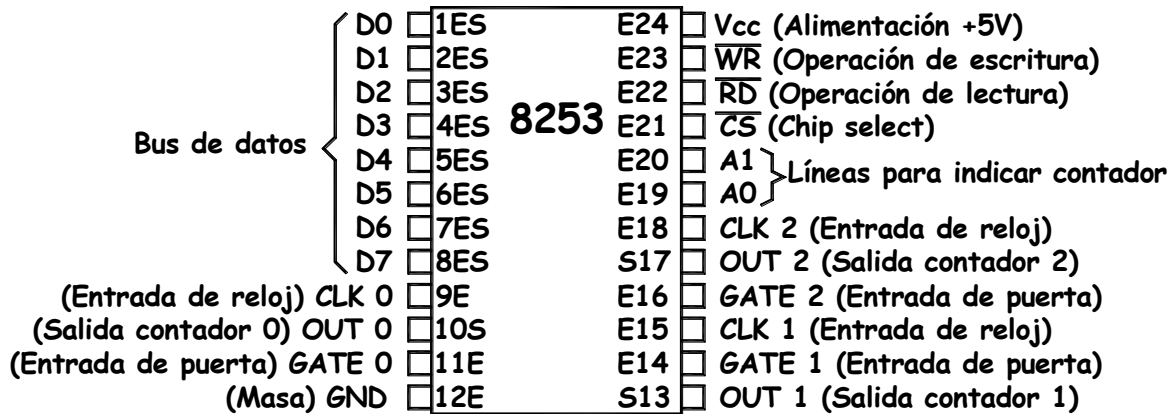


Ilustración 170 - 8253: diagrama detallado de los terminales del circuito

En la tabla siguiente se muestra el control de los contadores en función de las señales que provienen del microcontrolador.

\overline{CS}	A1	A0	\overline{RD}	\overline{WR}	Descripción
0	0	0		0	Escritura en el contador 0
0	0			0	Escritura en el contador 1
0		0		0	Escritura en el contador 2
0				0	Escritura en el registro de control
0	0	0	0		Lectura en el contador 0
0	0		0		Lectura en el contador 1
0		0	0		Lectura en el contador 2
	*	*	*	*	Alta impedancia
0			0		Alta impedancia
0	*	*			Alta impedancia

Tabla 17 - Señales de control de los 8253

2.2.1.1.- DIAGRAMA DE CONEXIONADO.

A continuación se describen las funciones de cada uno de los terminales del circuito 8253.

DO...D7: Conforman el bus de datos y está formado por 8 líneas que entran en un buffer bidireccional que puede adoptar el estado de alta impedancia (en el estado de alta impedancia, la salida se comporta como si aún no estuviera conectada al circuito, excepto por una pequeña corriente de fuga que puede fluir hacia adentro o hacia afuera de la terminal de salida),, estas 8 líneas se conectan con el bus de datos del sistema para poder enviar o recibir datos.

CS: Este terminal es el “chip select” del circuito, a nivel bajo en su entrada habilita el circuito para su funcionamiento. Esta entrada no tiene efecto sobre la operación actual del contador.

\overline{RD} : Es una señal de entrada que indica, cuando está negada, que se realiza una operación de lectura sobre el circuito. Se conecta a la misma línea del microcontrolador.

\overline{WR} : Es una señal de entrada que indica, cuando está negada, que se realiza una operación de escritura sobre el circuito. Se conecta a la misma línea del microcontrolador.

A0 y A1: Son dos señales de entrada que permiten controlar la selección de uno de los contadores 0, 1 y 2. Así mismo, también permiten dirigirse al registro de control para poder programar el funcionamiento del circuito. Estas dos entradas se conectan normalmente a las líneas de dirección menos significativas del microcontrolador.

CLK 0, CLK 1 y CLK 2: Son las líneas pertenecientes a las entradas de los contadores o temporizadores. A estas líneas pueden llegar señales de reloj o de eventos que se quieran contar.

GATE 0, GATE 1 y GATE 2: Son las líneas pertenecientes a las entradas de los contadores o temporizadores. A estas líneas llegan señales que controlan el funcionamiento de los contadores-temporizadores.

OUT 0, OUT 1 y OUT 2: Son las líneas pertenecientes a las salidas de los contadores o temporizadores para indicar que se ha llegado al final del conteo.

Vcc: Entrada de la tensión de alimentación +5V.

GND: Tierra o masa.

2.2.1.2.- DESCRIPCIÓN Y PROGRAMACIÓN DEL CIRCUITO.

Todas las funciones del 8253 son programadas por software. Un conjunto de palabras de control deben ser enviadas por el microcontrolador para inicializar cada contador en el modo deseado y la cantidad que deben contar. Estas palabras de control programan el modo, la secuencia de carga y la selección de cuenta en binario o en BCD. Una vez programado, el 8253 está listo para realizar el trabajo que le ha sido asignado. La operación de cuenta de cada contador es completamente independiente.

El programador puede seleccionar uno de los seis modos de funcionamiento de cada contador, utilizando las palabras de programación adecuadas y cargándolas en el registro de control seleccionado con los bits A0 y A1. El formato de la palabra de control o de programación y la función de cada uno de los ocho bits es el siguiente:

B7	B6	B5	B4	B3	B2	B1	B0
SCI	SC0	RLI	RL0	M2	M1	M0	BCD

Tabla 18 - Formato de la palabra de programación.

Los dos bits B7 y B6 sirven para seleccionar el contador sobre el que queremos programar un modo de funcionamiento. En función de la elección de los bits B7 y B6 ya sean “0” ó “1” tendremos el siguiente modo de funcionamiento.

B7 SCI	B6 SC0	Descripción
0	0	Selección del contador número 0.
0	1	Selección del contador número 1.
1	0	Selección del contador número 2.
1	1	Invalidada.

Tabla 19 - 8253: Selección de modos de funcionamiento

Con el bit B0 se selecciona el modo en el que queremos que se realice la cuenta, de forma que si es un “1” la cuenta se realiza en BCD, se cuentan 4 décadas lo que hace un número de 9999. Sin embargo, si el valor introducido en este bit es un “0” la cuenta se realiza en binario con 16 bits y el número que se puede contar es de $2^{16}=65536$. Con los bits B5 y B4 se selecciona el modo en el cual se escriben los bytes dentro de los contadores, el orden y los bytes a escribir.

B5 RLI	B4 RLO	Descripción
0	0	Operación de latching, es decir, se almacena el dato por donde va la cuenta del contador para posteriormente leerla. No afecta para nada al contaje.
0	1	Lectura o escritura solamente del byte menos significativo del contador seleccionado siendo el otro byte igual a 0.
1	0	Lectura o escritura solamente del byte más significativo del contador seleccionado siendo el otro byte igual a 0.
1	1	Lectura o escritura del byte menos significativo en primer lugar y después el byte más significativo del contador seleccionado.

Tabla 20 - 8253: Selección del modo de escritura en los contadores

Con los bits B3, B2 y B1 se selecciona el modo de funcionamiento de los contadores.

B3 M2	B2 M1	B1 M0	Descripción
0	0	0	Modo 0
0	0	1	Modo 1
*	1	0	Modo 2
*	1	1	Modo 3
1	0	0	Modo 4
1	0	1	Modo 5

Tabla 21 - 8253: Modo de funcionamiento de los contadores

2.2.1.3.- DESCRIPCIÓN DE LOS MODOS DE FUNCIONAMIENTO.

Modo 0. Interrupción al final del contaje.

Al iniciarse el modo de programación, la salida pasa a nivel bajo. Cuando se carga el valor a contar en el contador seleccionado, se inicia el contaje. La salida se mantiene a nivel bajo mientras dura el proceso de contaje. Cuando se llega al final, la salida pasa a nivel alto y se mantiene en este nivel hasta que aparece el flanco de caída del segundo pulso de carga del nuevo valor de contaje (2^{do} byte), es decir, permanecerá en alto hasta que el registro de cuenta seleccionado sea recargado con el modo.

Mode 0

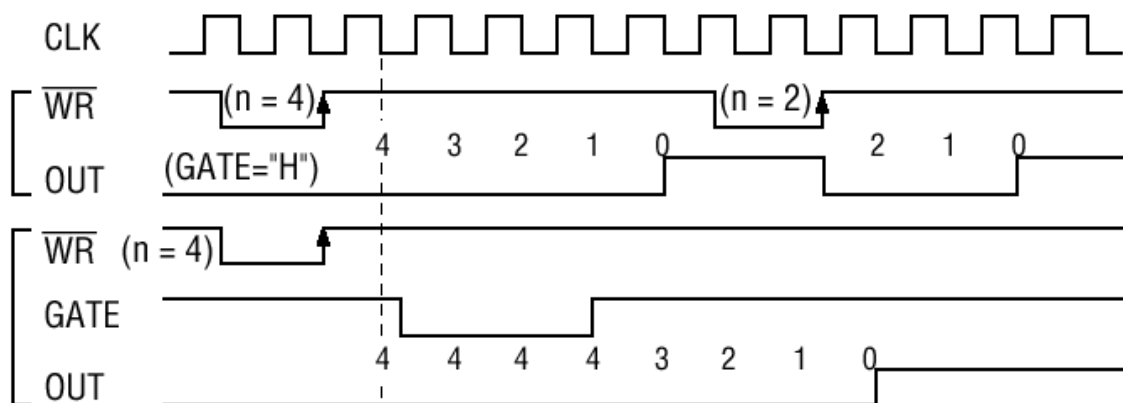


Ilustración 171 - Modo 0 8253. Interrupción al final del contaje

Si se introduce una señal de GATE (nivel bajo en dicha entrada), el contaje se interrumpe mientras dure la señal GATE. Cuando vuelve a pasar a nivel alto, continúa el contaje a partir del punto en que había quedado interrumpido. La continuación del contaje se produce a partir del flanco de bajada de la señal en CLK que hay después de haber puesto a nivel alto la señal de GATE. Si se hace una recarga de un registro mientras se está contando se produce lo siguiente:

- Al cargar el primer byte para el contador actual.
- Al cargar el segundo byte empieza de nuevo el contaje en el registro modificado.

Este es el modo implementado en el sistema de contaje de los pulsos del encoder. La señal de GATE se inhabilitó conectándola a +5V.

Modo I. Monoestable programable.

La salida se pone a nivel bajo en el flanco de bajada de la señal de entrada en CLK que se produce después del flanco de subida de la señal GATE. La salida pasa a nivel alto al final del periodo de contaje.

Mode 1

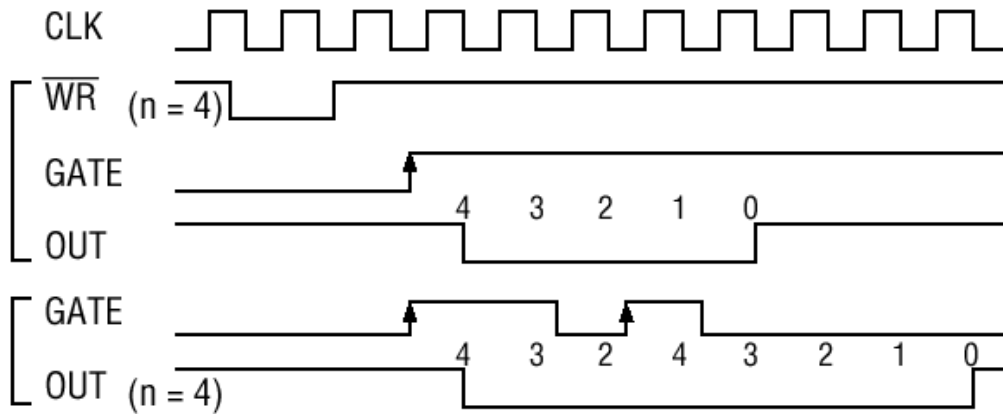


Ilustración 172 - Modo 1 8253 Monoestable programable

Si durante el proceso de contaje se carga un nuevo dato, dicho contaje no queda afectado. El nuevo dato programado se carga en el pulso de subida del siguiente pulso de la señal de entrada en CLK. La inserción de un pulso de GATE cuando la salida está a nivel bajo pone en RESET al contador y lo vuelve a arrancar cuando pasa a nivel alto. La salida se mantiene a nivel bajo durante la señal en la entrada GATE.

Modo 2. Divisor de frecuencia.

El divisor de frecuencia es un contador de módulo programable. La salida se mantiene a nivel bajo durante un periodo completo de la señal de entrada en CLK. El dato que se carga en el contador es el módulo del contador/divisor. Si el registro del contador se recarga entre dos pulsos de salida no queda afectado el periodo de contaje siguiente. La salida se mantiene a nivel alto durante la inserción de un pulso en la entrada GATE. Durante dicho tiempo, el contador permanece parado aunque no pierda el valor acumulado. El funcionamiento normal se reanuda en el flanco de bajada de la señal de entrada en CLK siguiente al flanco de subida de la entrada GATE. Así mismo, esta señal se puede utilizar como sincronizador del contador.

Mode 2

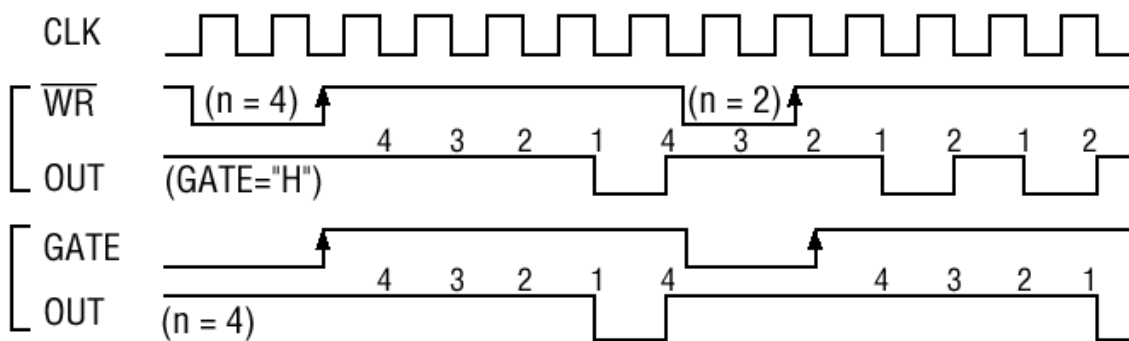


Ilustración 173 - Modo 2 8253 Divisor de frecuencia

Modo 3. Generador de onda cuadrada.

Este modo de programación es casi idéntico al Modo 2. Difieren en que la salida permanece alta durante la primera mitad del conteo y pasa a nivel bajo durante la segunda mitad de dicho conteo. Esto ocurre para valores pares del módulo. Para valores impares, la salida está a nivel alto durante un periodo más de la señal de entrada en CLK que con respecto al tiempo que permanece en estado bajo.

- Semiperiodo alto $N+1/2$
- Semiperiodo bajo $N-1/2$

Siendo N el valor decimal del módulo del divisor, o sea, el dato cargado.

Mode 3

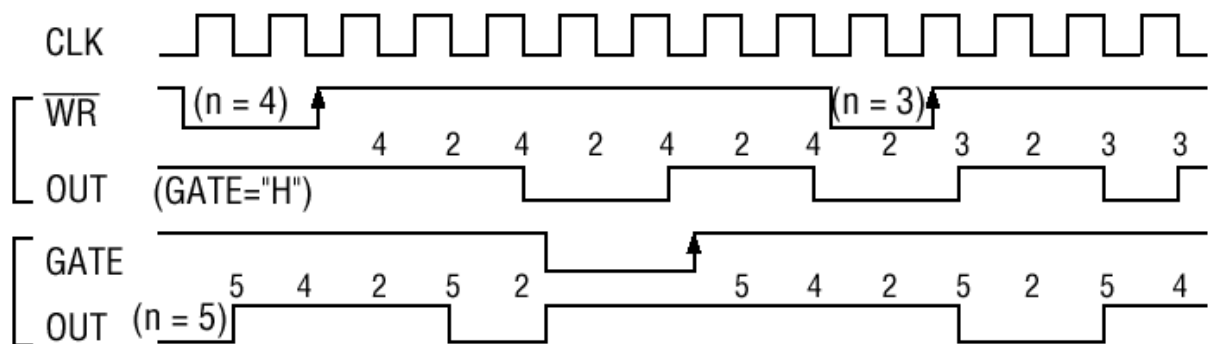


Ilustración 174 - Modo 3. 8253 Generador de onda cuadrada.

Si el registro del contador es recargado con un nuevo valor durante el conteo, éste queda reflejado inmediatamente después del conteo en curso. La salida pasa a estado alto cuando se activa a nivel bajo la entrada GATE. El conteo se vuelve a iniciar cuando dicha entrada pasa a nivel alto.

Modo 4. Retardo activado por software.

La salida pasa a nivel alto cuando se programa este modo. El conteo se inicia después de cargar el 2^{do} byte del dato. Cuando se alcanza el final del conteo, la salida pasa a nivel bajo durante un periodo de la señal de entrada en CLK, como se aprecia en la imagen inferior. Los cambios en el dato a contar se reflejan en la salida como datos a tener en cuenta desde el mismo momento que se realiza la nueva carga. Durante la carga del nuevo dato, la salida se mantiene a nivel alto y se inhibe el conteo.

Si en la entrada GATE se aplica un nivel bajo, la salida se mantiene en nivel alto, se pone a RESET el contador y después de pasar a nivel alto dicha entrada, se empieza a contar desde el valor inicial.

Mode 4

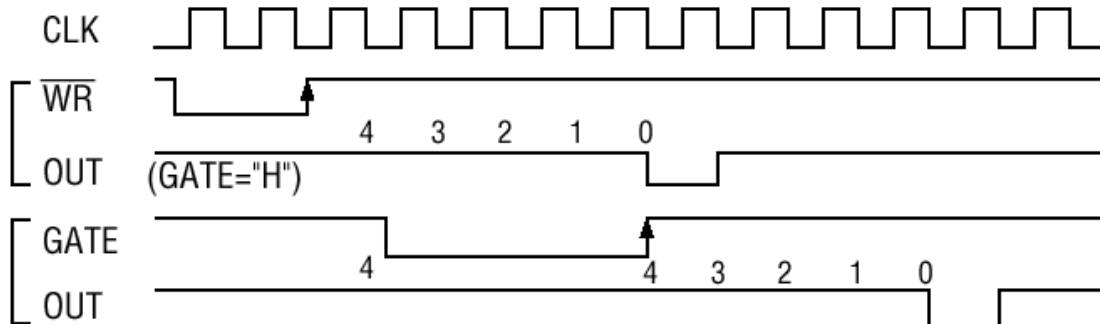


Ilustración 175 - Modo 4 8253 Retardo activado por software.

Modo 5. Retardo activado por hardware.

La salida pasa a nivel alto cuando se programa este modo de funcionamiento. El conteaje se inicia después de cargar el dato y de que la entrada GATE haya pasado a nivel alto. Al alcanzarse el final del conteaje, la salida pasa a nivel bajo durante un periodo de la señal de entrada en CLK. Los siguientes pulsos aplicados a la entrada GATE ponen a RESET el contador. El conteaje se inicia desde el último pulso aplicado a la entrada GATE. En la imagen inferior pueden observarse los correspondientes diagramas de tiempo.

Mode 5

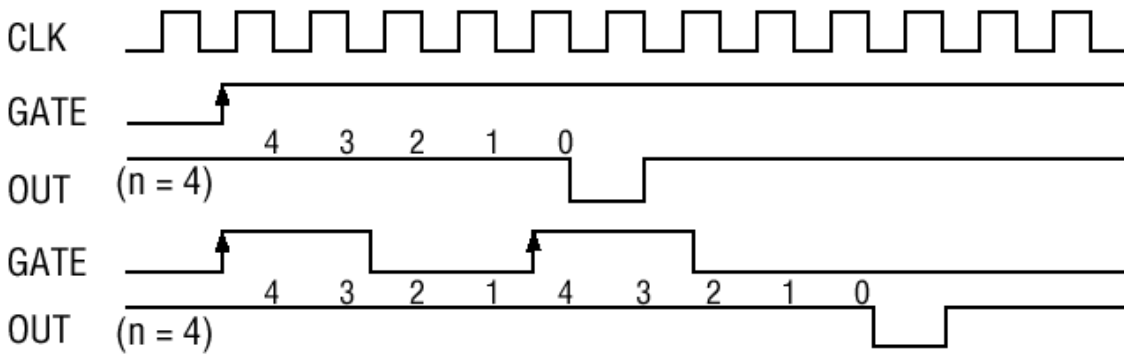


Ilustración 176 - Modo 5 8253 Retardo activado por hardware.

2.2.1.4.- PROCEDIMIENTO DE LECTURA - ESCRITURA

Operación de escritura.

Se puede programar cada uno de los contadores del 8253 con el modo y la cantidad deseada mediante software. El programador puede escribir en el 8253 una palabra de control de modo y el número programado de los bytes uno y dos del contador de registro antes de usar el contador seleccionado. El orden actual de la programación es muy flexible. Mediante la palabra de control de modo, podemos escoger cualquier secuencia de contadores. El contador 0 no tiene por qué ser el primero ni el contador 2 no tiene por qué ser el último. Cada registro de la palabra de control de modo del contador tiene una

dirección separada, así su carga es completamente independiente SC0 y SCI. Sin embargo, la carga del registro contador con el valor actual de cuenta puede ser realizada exactamente en el orden programado en la palabra de control de modo RL0 y RLI. Esta carga del valor del registro contador es independiente de la carga de la palabra de control de modo pero cuando el registro contador seleccionado está siendo cargado, debe ser cargado con el número de bytes programados en la palabra de control e modo RL0 y RLI. El primero o los dos primeros bytes en ser cargados en el registro contador no tienen que seguir a la palabra de control de modo asociada. Pueden ser programados en cualquier instante posterior a la carga de la palabra de control de modo, tan largo como para que el número correcto de bytes sean cargados en orden.

Todos los contadores cuentan hacia atrás, de este modo, el valor cargado en el registro contador irá decrementándose. Cuando todos los contadores equivalgan cero dentro del registro contador se observará el máximo contaje, 65536 en binario y 9999 en BCD. En el Modo 0 la nueva cuenta no vuelve a empezar hasta que la carga haya sido completada. El circuito aceptará uno o dos bytes dependiendo de cómo ha sido programada la palabra de control de modo RL0 y RLI, entonces realizará la operación de vuelta a empezar.

Operación de lectura.

En muchas de las aplicaciones del contador es necesario leer el valor del contador que está funcionando y tomar una decisión para el sistema basado en este valor. Los contadores son seguramente las aplicaciones más comunes que hacen uso de esta función. El 8253 está hecho de tal manera que permite al programador leer fácilmente el contenido de cualquier de los tres contadores sin tener consecuencias sobre el contador que está funcionando.

Hay dos métodos que el programador puede usar para leer el valor de los contadores. El primer método supone simplemente el uso de operaciones de lectura de entrada-salida del contador seleccionado. Controlando las entradas A0 y A1 del 8253 el programador puede hacer que el contador sea leído. La única condición de este modo es que para asegurar un contador de lectura permanente, la operación actual del contador seleccionado deberá ser inhabilitada controlando la entrada GATE, o mediante instrucciones externas que inhabiliten la entrada de la señal de entrada en CLK. El contador seleccionado estará disponible como se indica a continuación. La primera lectura lee el contenido del byte menos significativo y el segundo el contenido del byte más significativo. Debido a la configuración interna del 8253 esto es absolutamente necesario para completar el procedimiento completo de lectura. Si dos bytes son programados para ser leídos, estos deberán ser leídos antes de que a este mismo contador le llegue una orden de escritura.

El segundo método es leer mientras cuenta. Para que el programador pueda leer el contenido de cualquier contador sin afectar ni perturbar a la operación de cuenta, el 8253 tiene instrucciones internas a las que se puede acceder usando simplemente el comando de escritura del registro de modo. Básicamente, cuando el programador desea leer el contenido de un contador seleccionado, mientras cuenta, se carga el registro de modo con un código especial el cual introduce el actual valor del contador dentro de un registro almacén, para

que éste contenga la cantidad estable. Posteriormente, el programador emitirá un comando normal de lectura al contador seleccionado y el contenido del registro almacén será leído.

2.2.1.5.- EL CIRCUITO 8253 DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

El objetivo de los circuitos 8253 dentro del sistema de control diseñado es permitir el conteo de los pulsos de los encoder. Cada uno de los dos encoder (ver Capítulo II Sección III) produce seis señales a su salida de las cuales se toman en cuenta solo tres (las señales negadas no se tienen en cuenta). En total se implementan cuatro circuitos 8253 ubicados en la placa 8255 – 8253. Para el conteo de las seis señales (tres de cada uno de los dos encoder) se utilizan dos de los cuatro circuitos 8253. Los otros dos contadores constituyen un sistema de conteo encadenado de modo que una vez decrementando el valor programado de los 8253 el pulso de salida decremento en una unidad el contador encadenado. De esta forma se logra aumentar la cantidad de pulsos que el sistema puede contar. Como se comentó existen cuatro contadores implementados en la placa 8255 – 8253. Para referirse a ellos se han numerado del 1 al 4 como se muestra en la imagen inferior.

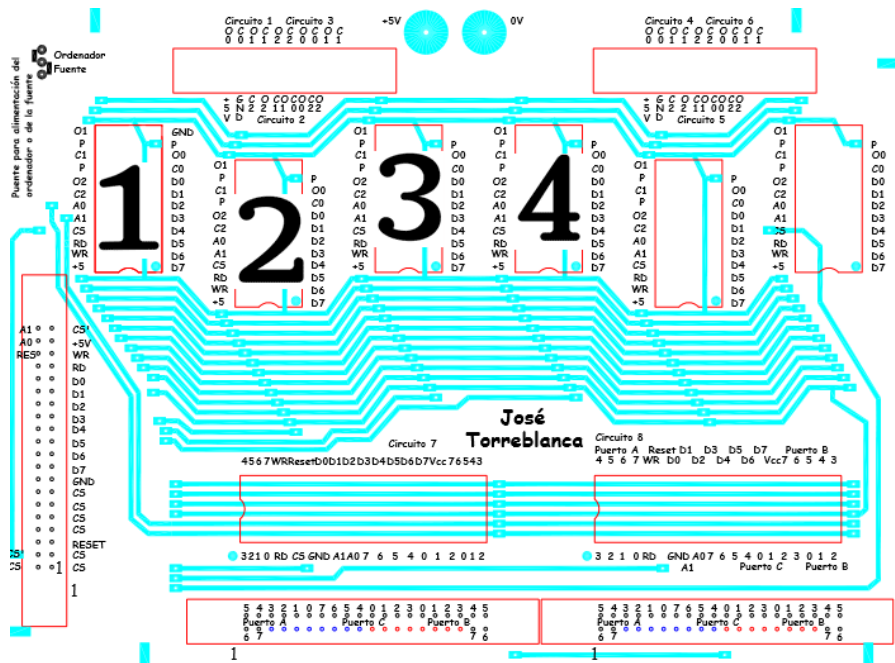


Ilustración 177 - Numeración de los circuitos 8253 de la placa 8255 - 8253.

2.2.2.- CIRCUITO 8255. PERIFÉRICO PROGRAMABLE DE ENTRADA-SALIDA.

Este circuito es un periférico programable de entrada-salida de aplicación general y compatible con el microcontrolador PIC16F877A. Dispone de un total de 24 líneas de entrada-salida divididas en tres puertos A, B y C, de los cuales el puerto C se puede dividir en dos grupos de cuatro bits cada uno.

El periférico 8255 se divide en dos grupos o bloques, el A y el B. El grupo A está formado por el puerto A de ocho bits y los cuatro bits más significativos del puerto C, mientras que el grupo B está formado por los ocho bits del puerto B y los cuatro bits menos significativos del puerto C. El grupo A se puede programar en tres modos diferentes de funcionamiento mientras que el grupo B sólo se puede programar en dos modos de funcionamiento.

El 8255 es un buffer bidireccional con control de alta impedancia utilizado para interconectar al periférico con el bus de datos del sistema microprocesador. Los datos se reciben o transmiten del periférico mediante instrucciones IN, OUT respectivamente del microcontrolador. A través de este buffer también se transfieren las palabras de programación y de estatus.

La función del bloque de control, lectura y escritura es controlar las transferencias interna de los datos. Recoge la información de direcciones y las señales de control que provienen del microcontrolador. A continuación se muestra el efecto combinado de las líneas que entran en el bloque de control, se detalla lo que cada combinación de líneas realiza sobre el circuito.

CS	AI	A0	RD	WR	Descripción
0	0	0	1	0	Escritura en el puerto A
0	0	1	1	0	Escritura en el puerto B
0	1	0	1	0	Escritura en el puerto C
0	1	1	1	0	Escritura en el registro de control
0	0	0	0	1	Lectura en el puerto A
0	0	1	0	1	Lectura en el puerto B
0	1	0	0	1	Lectura en el puerto C
1	*	*	*	*	Alta impedancia
0	1	1	0	1	No se puede realizar
0	*	*	1	1	Alta impedancia

Ilustración 178 - 8255: Líneas de control

En la imagen inferior puede observarse la estructura interna de este circuito.

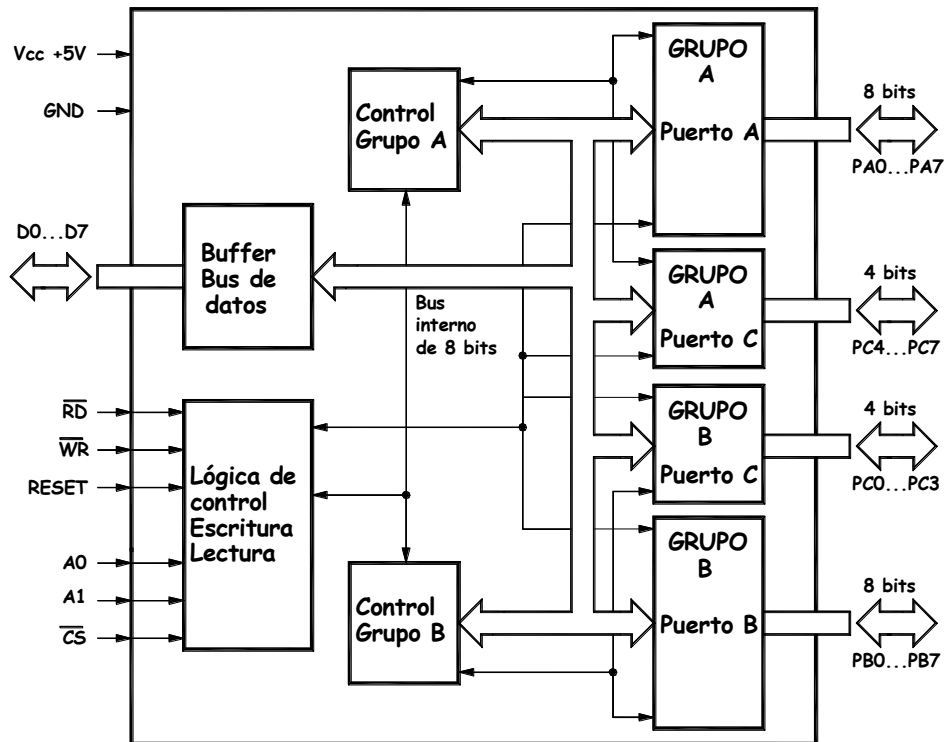


Ilustración 179- Estructura interna del 8255

El modo de funcionamiento de cada uno de los puertos se programa por software. La palabra de control que escribe el microcontrolador sobre el periférico contiene la información del modo de funcionamiento y sirve para inicializar al circuito. Cada uno de los grupos de control acepta comandos de programación para los puertos asociados a él.

- Control del grupo A: Puerto A y parte alta del puerto C (Pc4...Pc7).
- Control del grupo B: Puerto B y parte baja del puerto C (Pc0...Pc3).

En el registro de control solamente se puede escribir, no permitiéndose las operaciones de lectura. Los puertos son de 8 bits cada uno y todos ellos son programables de diferentes maneras mediante las adecuadas palabras de programación. Además, el puerto C tiene la particularidad de que sus bits se pueden programar independientemente. En la imagen inferior se encuentra el diagrama detallado de los terminales de este circuito.

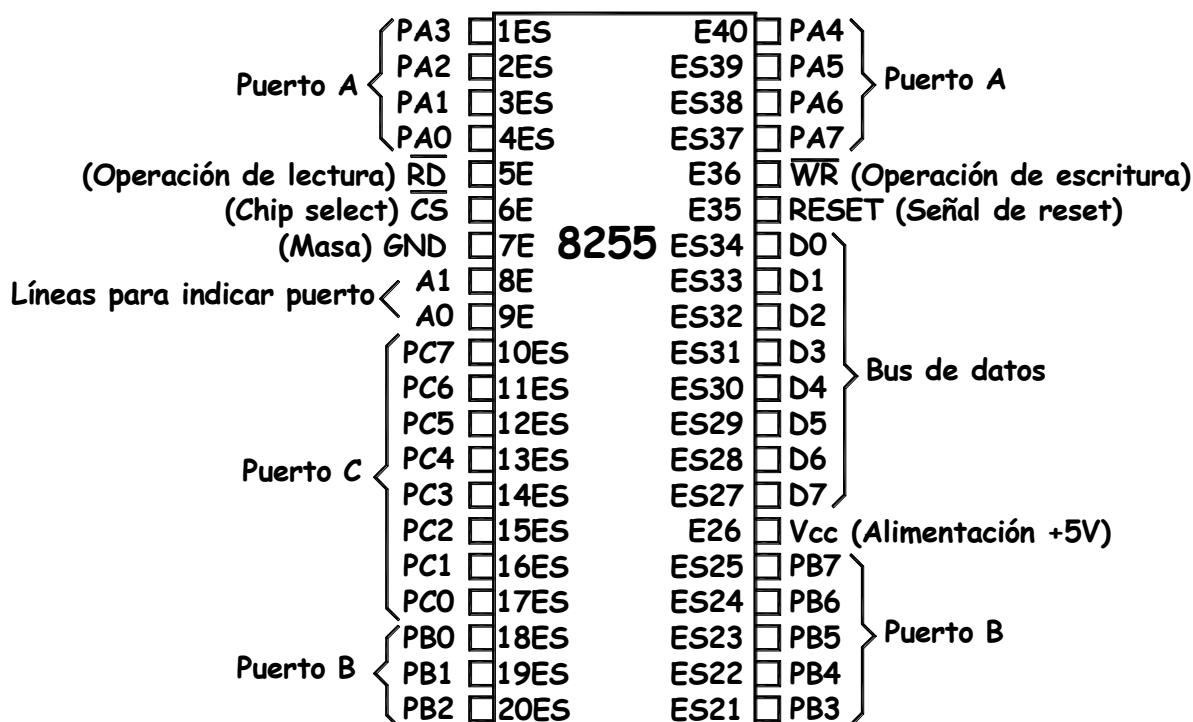


Ilustración 180 - Diagrama detallado del conexionado del 8255, la letra "E" indica entradas al circuito y la letra "S" salidas del circuito.

2.2.2.1.- DIAGRAMA DE CONEXIONADO.

A continuación se describen las funciones de cada uno de los terminales del circuito 8255.

DO...D7: Conforman el bus de datos y está formado por 8 líneas que entran en un buffer bidireccional que puede adoptar el estado de alta impedancia, estas 8 líneas se conectan con el bus de datos del sistema para poder enviar o recibir datos.

CS: Este terminal es el "chip select" del circuito, a nivel bajo en su entrada habilita el circuito para su funcionamiento.

RD: Es una señal de entrada que indica, cuando está negada, que se realiza una operación de lectura sobre el circuito. Se conecta a la misma línea del microcontrolador.

WR: Es una señal de entrada que indica, cuando está negada, que se realiza una operación de escritura sobre el circuito. Se conecta a la misma línea del microcontrolador.

A0 y A1: Son dos señales de entrada que permiten controlar la selección de uno de los puertos A, B y C. Así mismo, también permite dirigirse al registro de control para poder programar el funcionamiento del circuito. Estas dos entradas se conectan normalmente a las líneas de dirección menos significativas del microcontrolador.

RESET: Es una señal de entrada que es utilizada para producir un reset en el circuito. Se conecta a la línea RESET OUT del microcontrolador y así cuando éste recibe un reset hace que el circuito conectado a él también reciba un reset.

PA0...PA7: Son las líneas pertenecientes al puerto A, están formadas por un buffer de 8 bits que permite la entrada o salida de datos según se programe.

PB0...PB7: Son las líneas pertenecientes al puerto B, están formadas por un buffer de 8 bits que permite la entrada o salida de datos según se programe.

PC0...PC7: Son las líneas pertenecientes al puerto C, están formadas por un buffer de 8 bits que permite la entrada o salida de datos según se programe. Además estas líneas se pueden dividir en dos bloques de 4 bits cada una pudiéndose utilizar cada una de ellas como señales de control o de estado junto con los puertos A y B.

Vcc: Entrada de la tensión de alimentación +5V.

GND: Tierra o masa.

2.2.2.2.- MODOS DE FUNCIONAMIENTO DEL CIRCUITO.

Existen tres modos de funcionamiento que pueden escogerse por software:

1. Modo 0. Modo básico de entradas-salidas.
2. Modo 1. Modo comunicaciones unidireccionales.
3. Modo 2. Modo comunicaciones bidireccionales.

Cuando la entrada de RESET pasa a estado alto, todos los puertos quedan programados en el modo entrada. Después de pasar la señal de RESET a estado bajo, los puertos continúan programados en el modo 0 y en entrada. Durante la ejecución del proceso se puede cambiar la programación de modo en cualquier momento. Los modos del puerto A y del puerto B se pueden definir separadamente. Sin embargo, el puerto C queda dividido en dos mitades, dependientes cada una de ellas del puerto A y del puerto B respectivamente. Todos los registros de salida son puestos en reset igual que los biestables de estado cuando se efectúa un cambio de modo. Los modos se pueden combinar según las necesidades.

A continuación se detallan los modos de funcionamiento partiendo de la palabra de control o de programación del circuito 8255.

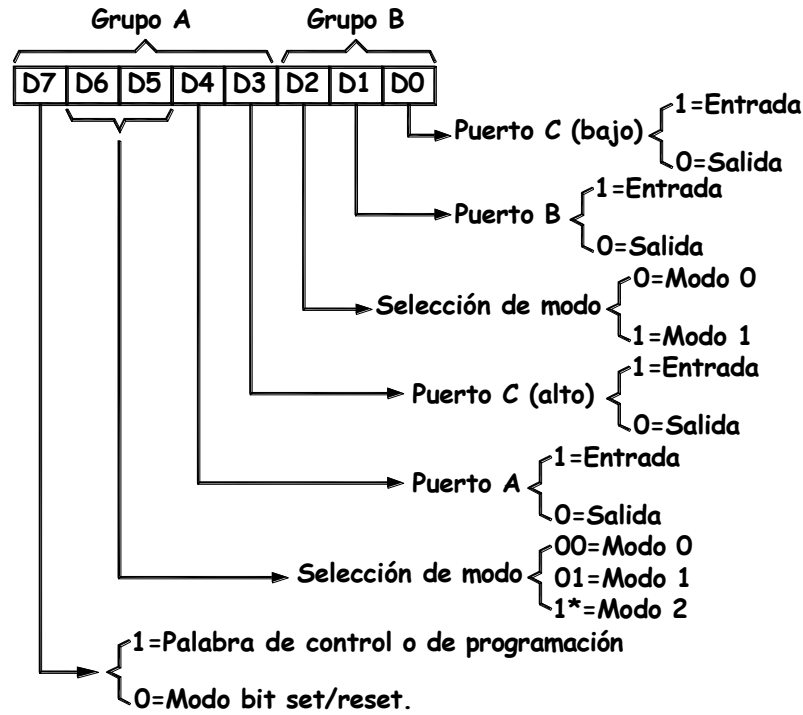
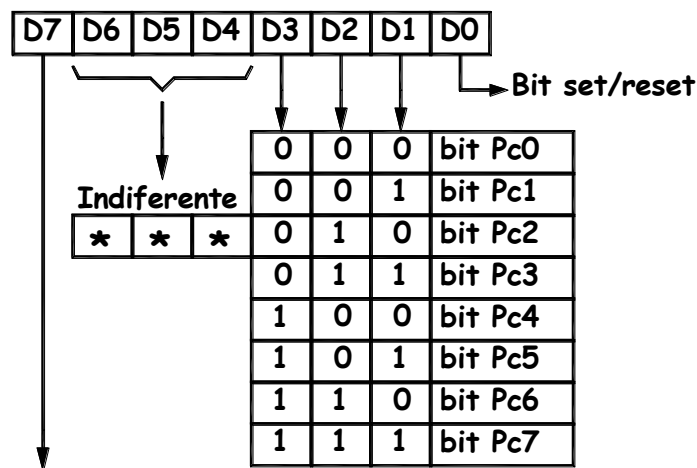


Ilustración 181 - Palabra de programación del circuito 8255.



0=Modo bit set/reset.
Poner a 0 ó 1 un bit del puerto C

Ilustración 182 - Palabra para asignar valores de control a los bits del puerto C. En los modo 1 y 2 con esta configuración se puede poner a 1 ó 0 ciertos flip flops que permiten o prohíben la generación de peticiones de interrupción.

Modo 0

Este modo de funcionamiento permite una programación fácil de cada uno de los tres puertos para trabajar como entradas o salidas. En este modo no se necesitan señales de control; los datos son simplemente escritos o leídos sobre un puerto específico.

Las características de este modo y la función de las diferentes líneas son:

- Dos puertos de 8 bits.

- Dos puertos de 4 bits con capacidad de control set/reset de cada línea.
- Las salidas son latched, es decir, almacenan el dato.
- Las entradas no son latched, es decir, no almacenan el dato, cuando se hace una operación de lectura del puerto se lee lo que en ese momento hay.

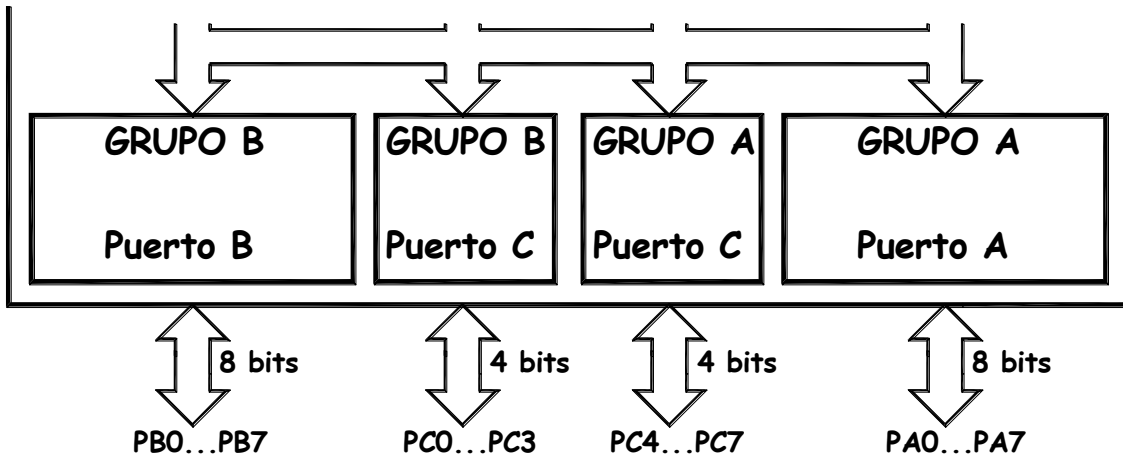


Ilustración 183 - Comportamiento de las líneas de los diferentes puertos cuando funcionan en Modo 0

Diagrama de tiempos para la entrada y para la salida.

Mode 0 (Basic Input)

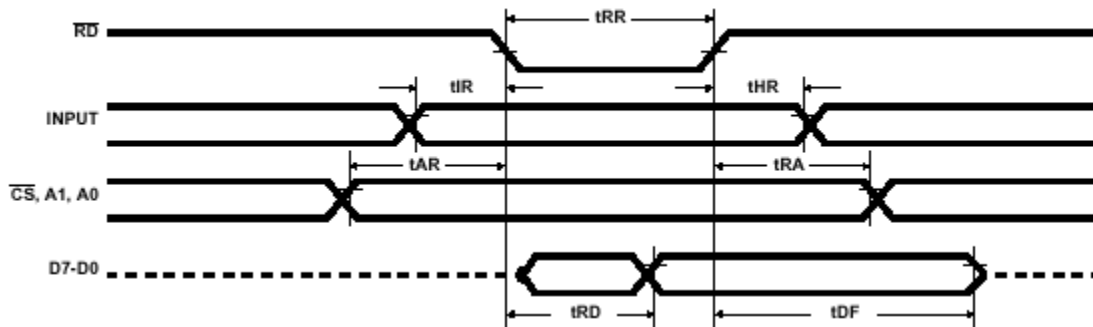


Ilustración 184 - Diagramas de tiempos para el funcionamiento en Modo 0 cuando se está produciendo una lectura de datos.

Mode 0 (Basic Output)

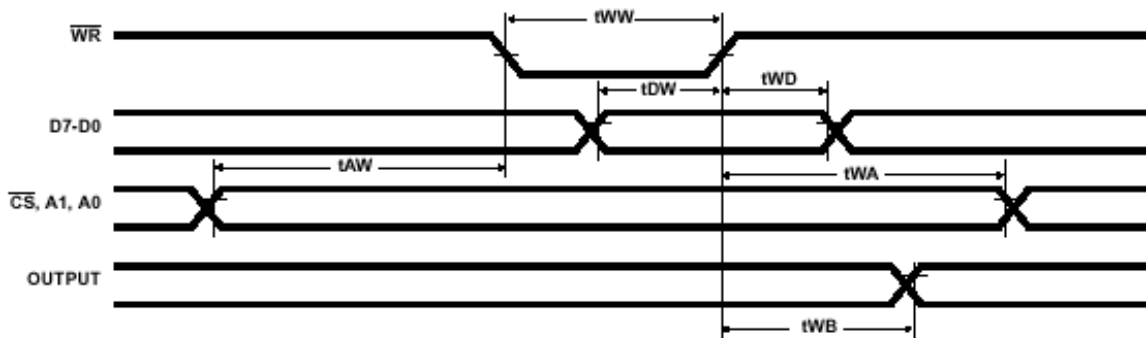


Ilustración 185 - Diagramas de tiempos para el funcionamiento en Modo 0 cuando se está produciendo una escritura de datos.

Modo I

Esta configuración permite hacer la transferencia de datos hacia o desde un puerto seleccionado, junto con las señales de control necesarias para conseguir una transferencia con dialogo. En este modo los puertos A y B utilizan las varias líneas del puerto C para generar o aceptar dichas señales de control. Las características de este modo son:

- Uno o dos puertos en modo comunicaciones.
- Cada puerto en modo I se configura con:
 - Un puerto de 8 bits.
 - Tres líneas de control del puerto C.
 - Control por interrupciones.
- Cualquier puerto puede ser de entrada o salida.
- Si se programa un solo puerto en comunicaciones, el resto de las 13 líneas se pueden programar en modo 0, es decir en entradas o en salidas.
- Si se programan dos puertos en comunicaciones, las dos líneas que sobran en el puerto C se pueden programar en modo 0 e incluso con capacidad de control por el bit set/reset.

En la imagen inferior se observa la definición de las señales de control para la entrada de datos por el puerto A y el puerto B.

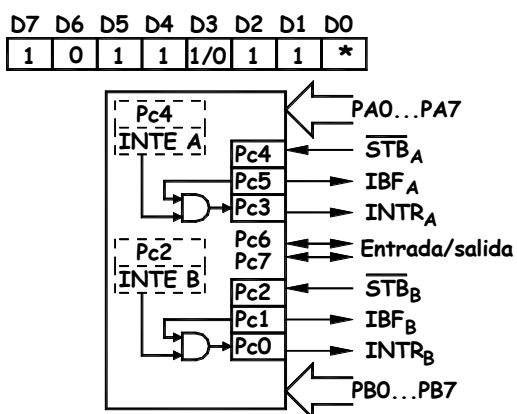


Ilustración 186 - Configuración en Modo I con el puerto A y B en entrada.

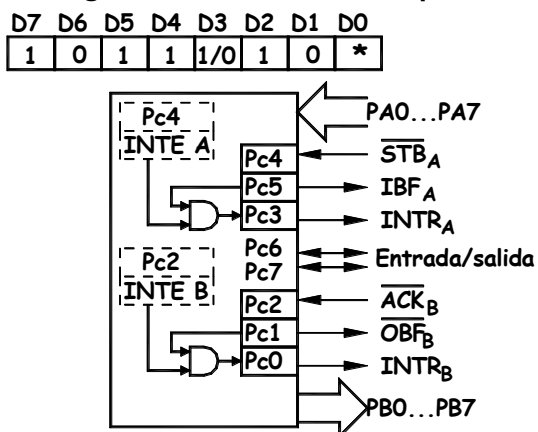


Ilustración 187 - Configuración en Modo I con el puerto A en entrada y el puerto B en salida.

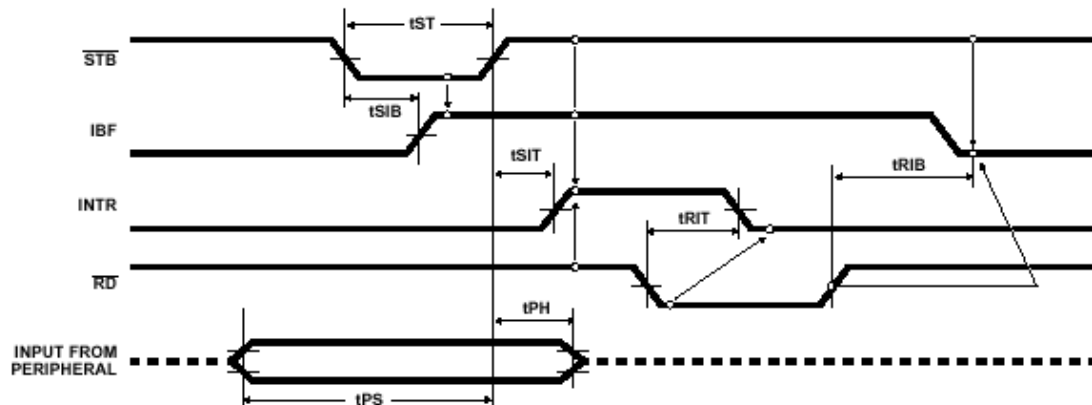


Ilustración 188 - Diagrama de tiempos cuando el puerto funciona en modo comunicaciones y como entrada.

En la imagen inferior se indica a modo de ejemplo la configuración adoptada para la entrada de datos en modo comunicaciones o diálogo a través del puerto A y B, así como los correspondientes diagramas de tiempo de las señales de control que se aprecian en la figura 20.

\overline{STB} (Entrada strobe): Un nivel bajo en esta entrada da la orden de memorizar los datos en los latches de entrada.

IBF (Entrada buffer lleno): Un nivel alto en esta salida indica que el dato ha sido cargado en el latch de entrada. La señal IBF es activada por el flanco de bajada de la señal **\overline{STB}** y es desactivada por el flanco de subida de la entrada \overline{RD} .

INTR (Interrupción solicitada): El nivel alto que aparece en esta salida se puede utilizar como interrupción hacia el microprocesador cuando un periférico de entrada requiere su servicio. La INTR se activa al pasar la señal \overline{STB} a nivel alto y en el supuesto de que IBF e INTE estén a nivel alto. Esta señal de petición de interrupción INTR pasa a nivel bajo con el flanco de bajada de la señal \overline{RD} . Este procedimiento permite a un periférico de entrada activar la señal de interrupción activando, con la señal de strobe, la entrada de datos al periférico.

INTE A. Este bit es controlado por el bit 4 set/reset del puerto C.

INTE B. Este bit es controlado por el bit 2 set/reset del puerto C.

Los puertos A y B pueden programarse también como puertos de entrada o de salida, de una manera individual. La transferencia de las señales a través de dichos puertos se realiza por el sistema strobe. La definición de las señales de control para la salida de datos por el puerto A y el puerto B se observa en las imágenes inferiores.

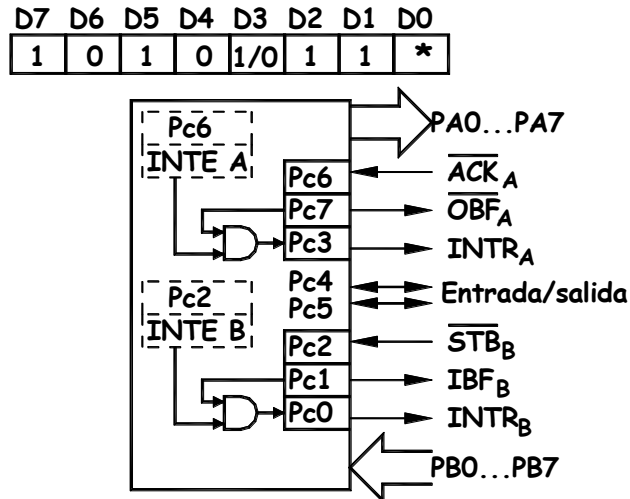


Ilustración 189 - Configuración en Modo I con el puerto A en salida y el puerto B en entrada.

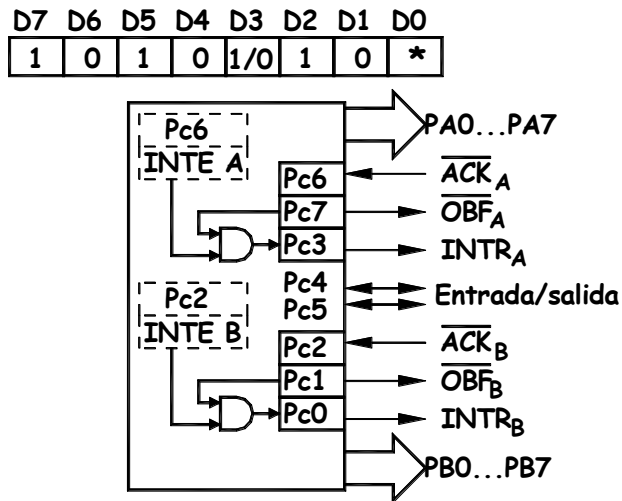


Ilustración 190 - Configuración en Modo I con el puerto A y B en salida.

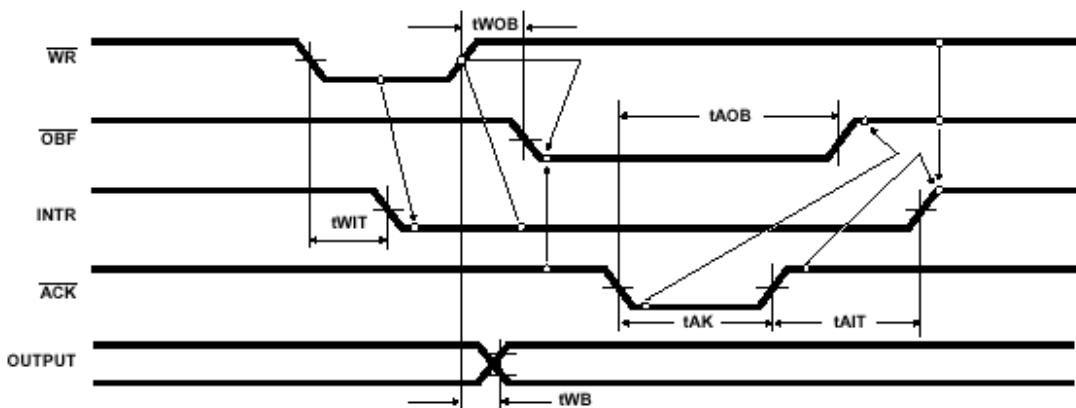


Ilustración 191 - Diagrama de tiempos cuando el puerto funciona en modo comunicaciones y como salida.

En las imágenes inferiores se indica a modo de ejemplo la configuración adoptada para la salida de datos en modo comunicaciones o dialogo a través del puerto A y B, así como los correspondientes diagramas de tiempo de las señales de control.

$\overline{\text{OBF}}$ (Buffer lleno): Esta salida se pondrá a nivel bajo para indicar que el microcontrolador ha escrito datos de salida para el puerto especificado.

$\overline{\text{ACK}}$ (Entrada de reconocimiento): Esta entrada se pondrá a nivel bajo para informar que los datos del puerto han sido aceptados.

INTR (Interrupción solicitada): El nivel alto que aparece en esta salida se puede utilizar como interrupción hacia el microcontrolador cuando un periférico de entrada requiere su servicio.

INTE A. Este bit es controlado por el bit 6 set/reset del puerto C.

INTE B. Este bit es controlado por el bit 2 set/reset del puerto C.

Este modo de trabajo permite el establecimiento de diálogo o comunicaciones de los periféricos con los puertos A y B, soportando las líneas del puerto C las señales de diálogo. El puerto B emplea como líneas auxiliares Pc0...Pc2 y el puerto A usa Pc3...Pc7. Las líneas del puerto C no utilizadas en el diálogo pueden actuar como líneas convencionales de entrada-salida.

En la figura 78 puede observarse la configuración de los terminales del circuito 8255 cuando el puerto A y B actúan como entradas. Las líneas PC6 y PC7 funcionan como líneas de salida convencionales, como consecuencia de la adecuada programación con un 0 del bit que les corresponda. En la imagen la línea de diálogo $\overline{\text{STB}}$, que genera el periférico, sirve para cargar en las básculas internas del 8255 el dato que envía aquél hacia los puertos A o B. Al cargarse un dato en las puertos del 8255, se activa la señal IBF que indica que el buffer de entrada de datos está lleno. El microcontrolador explora periódicamente las líneas IBF y cuando alguna está a nivel alto, o sea, activa, reconoce que en el puerto que controla hay un dato disponible. Al leerse el puerto con información, se desactiva la línea IBF y también la $\overline{\text{STB}}$, que estaba memorizada o "lacheada". El diálogo periférico-8255 funciona en este caso de la siguiente forma:

1. Cuando el periférico dispone de un dato, lo coloca en las líneas correspondientes del puerto del 8255 y activa la señal $\overline{\text{STB}}$, (dato listo) que lo carga en el registro del puerto.
2. Al contener un dato el registro del puerto, se activa IBF (nivel alto) señalizando a la CPU la presencia de información. Cuando la CPU lee el puerto se desactiva IBF y $\overline{\text{STB}}$.

Para dotar de mayor velocidad al sistema, se puede hacer uso de las interrupciones. Así puede generarse una petición de interrupción en el momento que el periférico carga un dato en el 8255. En este caso, previamente hay que poner a 1 la línea del puerto C que actúa

como puesta a 1 o set de un flip-flop INTE asociado con algunas de las líneas del puerto C (PC2 o PC4). Así, la puerta AND que provoca la petición de interrupción (INTR) a la CPU, se activa cuando IBF está a 1.

En la figura 82 se muestra la configuración de las líneas del 8255 cuando funciona en Modo 1 y los puertos A y B como salidas. También se aprecia la programación del registro de control, La señal \overline{OBF} se activa cuando queda lleno el buffer de salida del puerto, es decir, cuando la CPU escribe un dato en el puerto que corresponda. La señal \overline{ACK} , procedente del periférico, sirve para señalar el reconocimiento de la salida del dato cuando el periférico lee el puerto, momento en el que se desactiva la señal \overline{OBF} . Finalmente, las líneas INTR son las de petición de interrupción al microcontrolador cuando el puerto dispone de información y está lista para ser leída por el periférico. En el caso de querer avisar al microcontrolador mediante interrupción hay que poner previamente a 1 los flip-flops INTE, que corresponden con un par líneas del puerto C. Ya se ha explicado que es posible sacar un 1 o un 0 por las líneas del puerto C con la adecuada programación del registro de control. Esto supone el set o el reset de los flip-flops INTE asociados a dichas líneas. En el Modo 1 es factible realizar diversas combinaciones, como por ejemplo, que la puerta A trabaje como entrada y la puerta B como salida y viceversa.

En resumen, cuando el 8255 y el periférico admiten diálogo, el elemento que envía información activa una señal que representa "dato enviado" y el que la recibe activa otra similar a "dato recibido". Cuando un puerto del 8255 actúa como entrada, las señales de diálogo son IBF y \overline{STB} , mientras que si actúa como salida son \overline{OBF} y \overline{ACK} .

Modo 2

Este tipo de configuración permite la transmisión de datos en los dos sentidos, como si de un bus de datos se tratase, la transmisión se controla con las correspondientes señales de control para el modo dialogo o comunicaciones. Las características de este modo son:

- Un bus bidireccional, de 8 bits, el cual contiene:
 - 8 bits bidireccionales, a través de la puerta A.
 - 5 líneas de control.
 - Control por interrupciones.
 - Entradas y salidas latcheadas.
- El resto de las once líneas se pueden programar tanto en modo 0 como en modo 1.
-

En este modo sólo puede funcionar el puerto A, aunque simultáneamente el puerto B puede hacerlo en modo 0 o modo 1. En el modo 2 el puerto A puede trabajar de forma bidireccional, como entrada y salida, usando como líneas auxiliares 5 del puerto C.

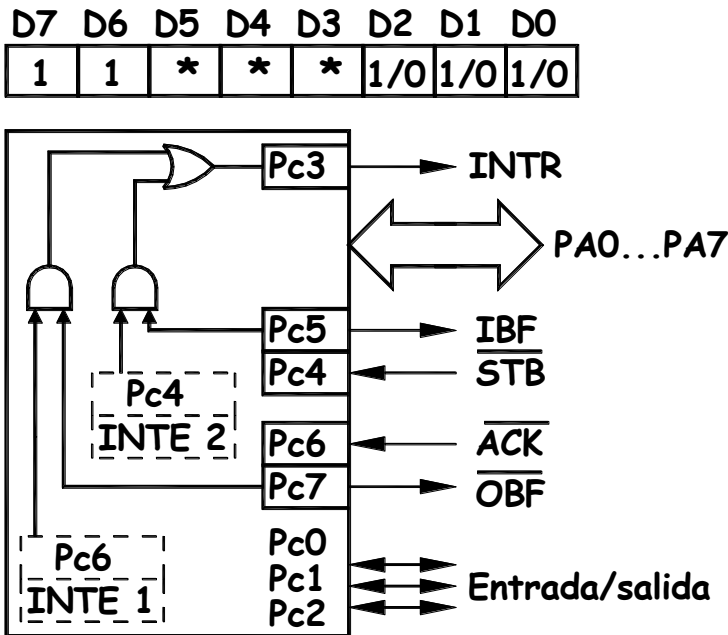


Ilustración 192 - Configuración en Modo 2.

Las 3 líneas restantes del puerto C pueden funcionar como líneas de entradas-salidas convencionales, o bien, como líneas auxiliares del puerto B, si trabaja en modo I.

Al trabajar el puerto A como entrada y salida, utiliza las señales de control \overline{ACK} y \overline{OBF} cuando es salida e IBF y \overline{STB} cuando es entrada. La petición de interrupción a la CPU puede producirse desde la línea IBF o desde la \overline{OBF} usando una puerta OR.

En la imagen 84 se tiene el formato correspondiente a este modo y en la 85 se observa su diagrama de tiempos. A continuación se definen las señales de control para la salida-entrada de datos por el puerto A en modo 2.

a.-) Como salida:

INTR (Solicitud de interrupción): Es una salida que pasa a nivel alto cuando se cargan datos en un sentido o en otro, que puede servir para generar una interrupción hacia el microcontrolador.

\overline{OBF} (Salida buffer lleno): Esta salida se pondrá a nivel bajo para indicar que el microcontrolador ha escrito datos de salida en el puerto A.

\overline{ACK} (Entrada de reconocimiento): Esta entrada se pondrá a nivel bajo para habilitar el buffer de salida del puerto A para sacar o enviar el dato. Mientras no ocurra tal cosa el buffer de salida permanece en estado de alta impedancia.

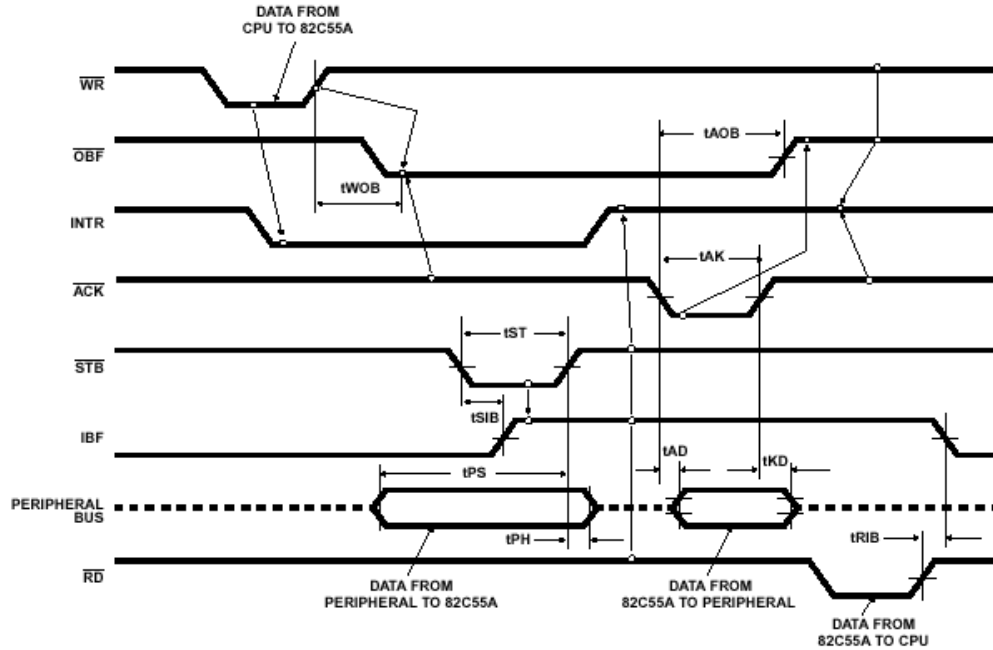
INTE I (Flip-flop asociado con \overline{OBF}): Este flip-flop está controlado por el bit set reset, Pc6, del puerto C para activarlo o desactivarlo.

b.-) Como entrada:

\overline{STB} (Entrada strobe): Un nivel bajo en esta entrada da la orden de memorizar los datos en los latches de entrada.

IBF (Entrada buffer lleno): Un nivel alto en esta salida indica que el dato ha sido cargado en el latch de entrada.

INTE 2 (Flip-flop asociado con IBF): Este flip-flop está controlado por el bit set reset, Pc4, del puerto C para activarlo o desactivarlo.



NOTE: Any sequence where \overline{WR} occurs before \overline{ACK} and \overline{STB} occurs before \overline{RD} is permissible. ($INTR = IBF \cdot MASK \cdot \overline{STB} \cdot \overline{RD} + \overline{OBF} \cdot MASK \cdot \overline{ACK} \cdot \overline{WR}$)

Ilustración 193 - Diagrama de tiempos cuando el puerto funciona en modo comunicaciones bidireccionales.

2.2.2.3.- EJEMPLOS DE CONEXIONES EN MODO COMUNICACIONES.

En primer lugar, se describe cómo conectar dos 8255 trabando uno en salida de datos y el otro recibiendo datos, modo unidireccional para los dos circuitos.

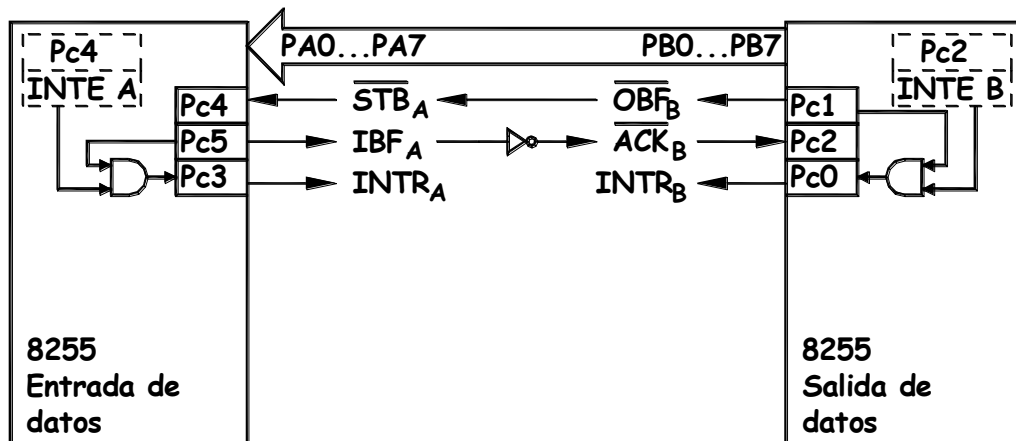


Ilustración 194 - Modo comunicaciones unidireccional.

El circuito en entrada tiene:

\overline{STB} es una entrada, a nivel bajo se da la orden de memorizar los datos en el puerto A.

IBF es una salida, a nivel alto indica que se han memorizado los datos en el puerto A.

INTR es una petición de interrupción.

El circuito en salida tiene:

\overline{OBF} es una salida, se pone a nivel bajo para indicar que se ha puesto un dato en el puerto B

\overline{ACK} es una entrada, a nivel bajo indica que los datos han sido aceptados y memorizados.

INTR es una petición de interrupción.

Este tipo de circuito puede funcionar de dos formas: una es por medio de la interrupción, solicitando al microcontrolador que se ejecute y la otra es por programa, leyendo constantemente el bit \overline{ACK} cuando se envía un dato y el bit IBF para indicar que se ha memorizado el dato.

En segundo lugar, vamos a ver como conectar dos 8255 trabando en modo bidireccional. Este tipo de circuito puede funcionar de dos formas al igual que el anterior. Una es por medio de las interrupciones y la otra es por medio de programa, leyendo constantemente los bits encargados de la comunicación.

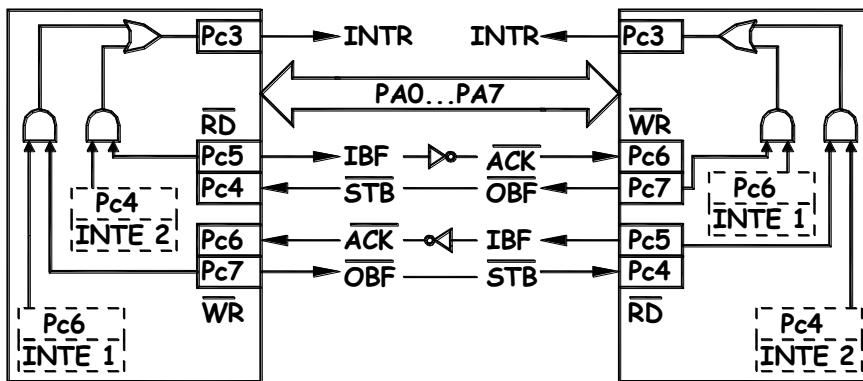


Ilustración 195 - Modo comunicaciones bidireccional.

En tercer lugar, se describe cómo conectar el 8255 a una impresora y el pequeño programa de control de la trasmisión de datos. Los datos serán enviados desde la posición 3000 de la memoria hasta encontrar el código 38 que indica fin de los datos enviados a la impresora, el código 0D le indica a la impresora que hay un salto de línea y la impresora imprime 80 caracteres en cada línea pasado a hexadecimal 50. La dirección del puerto A es la 30.

Dirección	Ensamblador	Hex	Comentarios
2000	MVI A,A0	3E	Programamos el circuito 8255 en modo comunicaciones del puerto A unidireccionales, el resto de líneas las programamos en salida
2001		A0	
2002	OUT 33	D3	Sacamos el dato anterior al puerto de programación del 8255

2003		33	cuya dirección es la 33
2004	LXI H,3000	21	Cargamos en el registro doble HL la dirección 3000, que será el encargado de ir leyendo los datos de memoria para enviárselos a la impresora
2005		00	
2006		30	
2007	MVI B,50	3E	Cargamos el registro B con el dato 50 para ir contando los caracteres impresos en cada línea. En decimal son 80
2008		50	
2009	IN 32	DB	Leemos el puerto C del 8255 para ver como está el bit encargado de la transmisión de datos
200A		32	
200B	ANI 80	E6	Realizamos la operación AND entre el dato leído del puerto C y el dato 80 para obtener un 00 en el acumulador cuando OBF=0
200C		80	
200D	JZ 2009	CA	Cuando el dato leído no coincida con 80, el flag Z estará a 0 y se continua con la ejecución del programa, en caso contrario salta a la dirección 2009, esto le indica que la impresora está lista
200E		09	
200F		20	
2010	MOV A, M	7E	Leemos dato de memoria y lo llevamos al acumulador
2011	CPI 38	FE	Comparamos el dato leído de la memoria con 38 para ver si es el fin de la comunicación
2012		38	
2013	JZ 2024	CA	Cuando los datos comparados son iguales el flag Z se pone a 1 y entonces se salta a la dirección indicada
2014		24	
2015		20	
2016	OUT 30	D3	Sacamos el dato por el puerto A para enviárselo a la impresora
2017		30	
2018	INX H	23	Incrementamos pareja de registros HL
2019	DCR B	05	Decrementamos registro B
201A	JNZ 2009	C2	Si el registro B llega a cero, el flag Z=1 y se indica que se ha escrito una línea y que hay que saltar a la siguiente, para ello se continua con el programa
201B		09	
201C		20	
201D	MVI A, 0D	3E	Llevamos el dato 0D al acumulador para enviárselo a la impresora y que salte una línea
201E		0D	
201F	OUT 30	D3	Sacamos por puerto el dato anterior
2020		30	
2021	JMP 2007	C3	Saltamos al inicio del programa para comenzar con otra impresión de caracteres de una línea
2022		07	
2023		20	
2024	HLT	76	Fin del programa

Tabla 22 - 8255. Conexión con una impresora

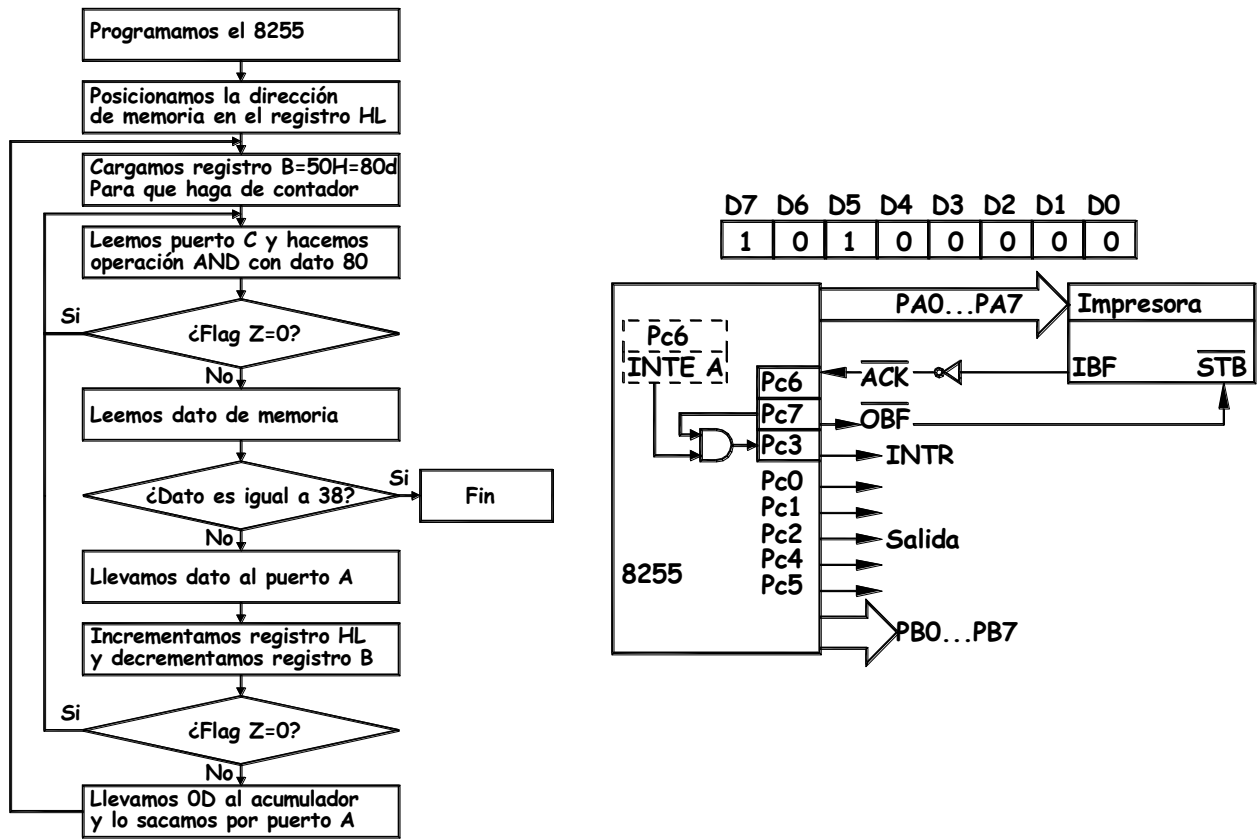


Ilustración 196 - Organigrama que corresponde con el programa de comunicaciones desde un 8255 hacia una impresora. Esquema de conexiones.

2.2.2.4.- ESTADOS DE LECTURA DEL PUERTO C.

En modo 0 el 8255 transfiere datos o lee datos del puerto C conectado a cualquier dispositivo. Cuando el 8255 es programado para funcionar en los modos 1 ó 2, el puerto C genera o acepta las señales de dialogo con el periférico. Leyendo los contenidos del puerto C se permite al programador probar o verificar los estados de cada periférico y cambiar en conformidad con los cambios del programa. No hay instrucciones especiales para leer los datos de información del puerto C. Una operación normal de lectura del puerto C sirve para realizar esta función.

Configuración de entrada en modo 1.

Pc7	Pc6	Pc5	Pc4	Pc3	Pc2	Pc1	Pc0
E/S	E/S	IBF _A	INTE _A	INTR _A	INTE _B	IBF _B	INTR _B
Grupo A				Grupo B			

Configuración de salida en modo 1.

Pc7	Pc6	Pc5	Pc4	Pc3	Pc2	Pc1	Pc0
OBF _A	INTE _A	E/S	E/S	INTR _A	INTE _B	OBF _B	INTR _B

Grupo A	Grupo B
---------	---------

Configuración en modo 2.

Pc7	Pc6	Pc5	Pc4	Pc3	Pc2	Pc1	Pc0
$\overline{\text{OBF}}$	INTE1	IBF	INTE2	INTR	*	*	*
Grupo A				Grupo B			

Cada una de las 8 salidas del puerto respectivo puede suministrar una corriente de 1 mA a 1,5 V, lo cual permite activar directamente una etapa “darlington” de mayor tensión o de mayor corriente.

2.2.2.5.- EL CIRCUITO 8255 DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

Dentro del sistema de control diseñado se implementan en total dos circuitos 8255 ubicados en la placa auxiliar 8255 – 8253. Como ya se dijo anteriormente, los 8255 permiten aumentar las prestaciones del microcontrolador al añadir tres puertos de salida o entrada por circuito al sistema. Por un lado, uno de los dos circuitos 8255 permite la conexión del microcontrolador con la placa de gestión de motores. En este caso se utilizan los puertos A, B y C del circuito 8855- 7 (nombre convenido de uno de los dos 8255). El puerto A realiza en envío del byte de entrada a uno de los convertores D/A mientras que el puerto B envía el byte de entrada al otro convertor. Dos bits del puerto C se utilizan para enviar la referencia de tensión a la etapa amplificadora de cada uno de los dos sistemas DAC0800 – TL074 (ver Capítulo V). Como se mencionó anteriormente los dos circuitos 8255 presentes en la placa auxiliar 8255 – 8253 se numeraron para poder referirse a ellos. La numeración convenida se observa en la imagen inferior.

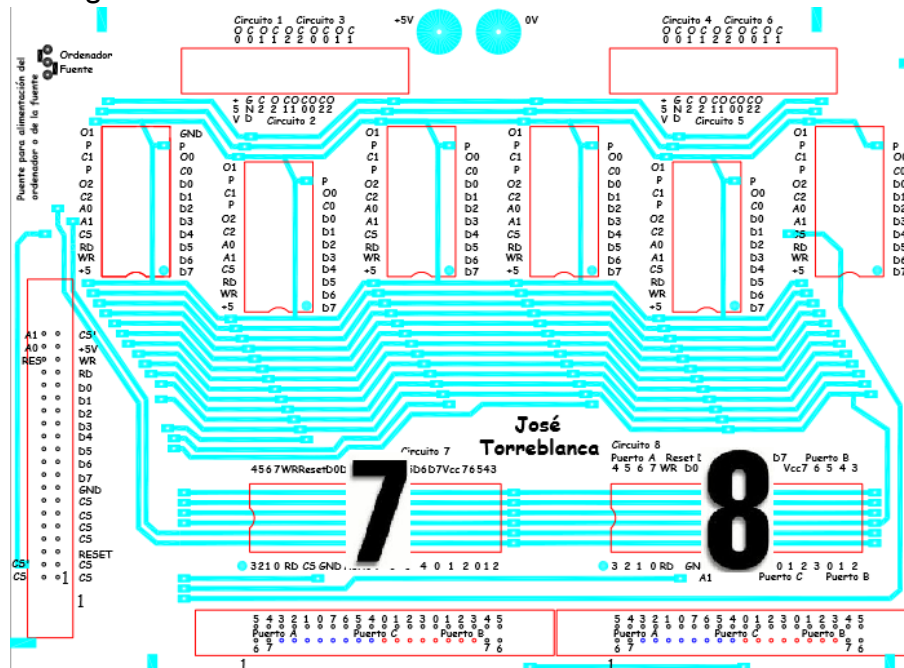


Ilustración 197 - Numeración de los circuitos 8255 en la placa 8255 - 8253.

3.- PLACA DE GESTIÓN DE MOTORES.

La placa de gestión de motores es una de las cuatro placas auxiliares que componen en su totalidad el sistema de control diseñado. Al igual que la placa anterior fue diseñada para otro proyecto y adaptada al sistema de control diseñado. Como su nombre lo indica, esta placa se encarga del control de los motores del brazo robótico. El brazo cuenta con dos motores de corriente continua, uno para cada eje, encargados del movimiento del mismo. Por lo tanto, la placa está constituida por dos sistemas gemelos de control que mediante un conversor digital analógico (DAC0800) y una etapa amplificadora (TL074) convierten un byte de información en una tensión variante entre 0 y 10V aproximadamente.

Esta placa se interconecta con la placa principal a través de la placa 8253- 8255 descrita anteriormente. De esta forma, un byte de información puesto en el bus de datos a través de software del microcontrolador es enviado a uno de los puertos de uno de los dos 8255 presentes en la placa auxiliar. El puerto A del 8255 -7 se conecta con la entrada de datos de uno de los dos convertidores DAC800 de control de un motor mientras que el puerto B del 8255 -7 se conecta con la entrada del otro conversor para realizar el control del otro motor.

Controlado la referencia de la tensión de la etapa amplificadora se tiene un control absoluto del movimiento de los motores puesto que, de desearse una parada total de los mismos, basta con enviar por software y desde el microcontrolador un "0" lógico a la patilla de la referencia del circuito TL074.

Además de los circuitos integrados mencionados, la placa cuenta también con una serie de potenciómetros que permiten ajustar el valor analógico de la tensión de salida. De esta forma, se logra acotar el rango de dicha tensión a los valores necesarios para el correcto funcionamiento de los motores del brazo. La tensión final de salida varía entre -1,53 V y +1,53 V aproximadamente para cada motor y no solo es acotada a través de este método sino que también se acota mediante software.

3.1.- DISEÑO DE LA PLACA

Al igual que la placa principal, la placa de gestión de motores se realizó con la fresadora partiendo de su diseño en AutoCAD tanto del lado de componentes como del lado de pistas. Se utilizaron los programas VCarve PRO y Win PC para transformar el diseño en Autocad en comandos para la máquina de fresado tipo CNC. En las imágenes inferiores puedes observarse los diseños del lado de componentes y del lado de pistas de la placa, respectivamente.

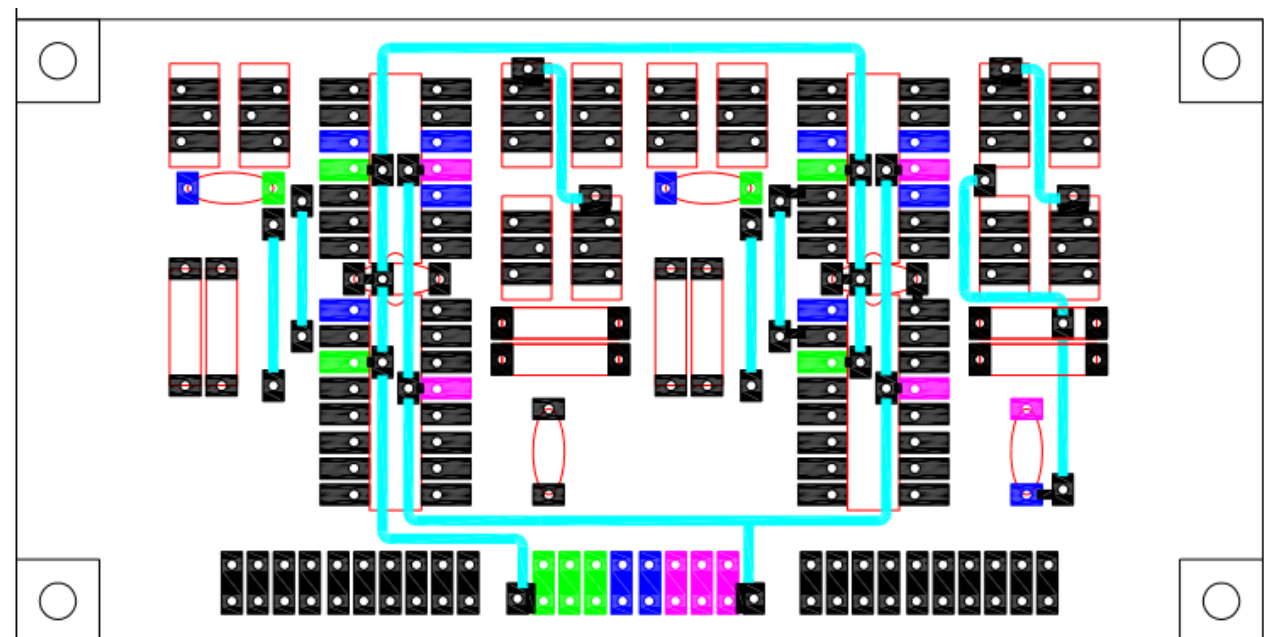


Ilustración 198 - Placa motores. Lado de componentes

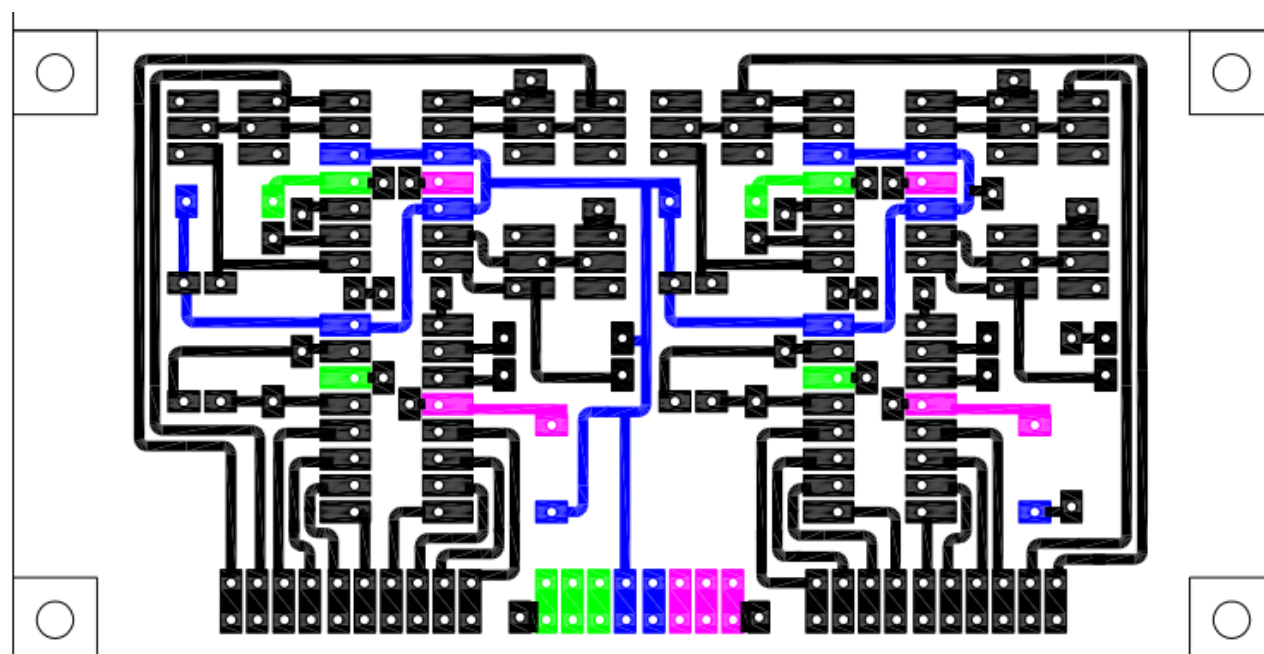


Ilustración 199 - Placa motores. Lado de pistas

En la imagen inferior se observa la placa terminada.

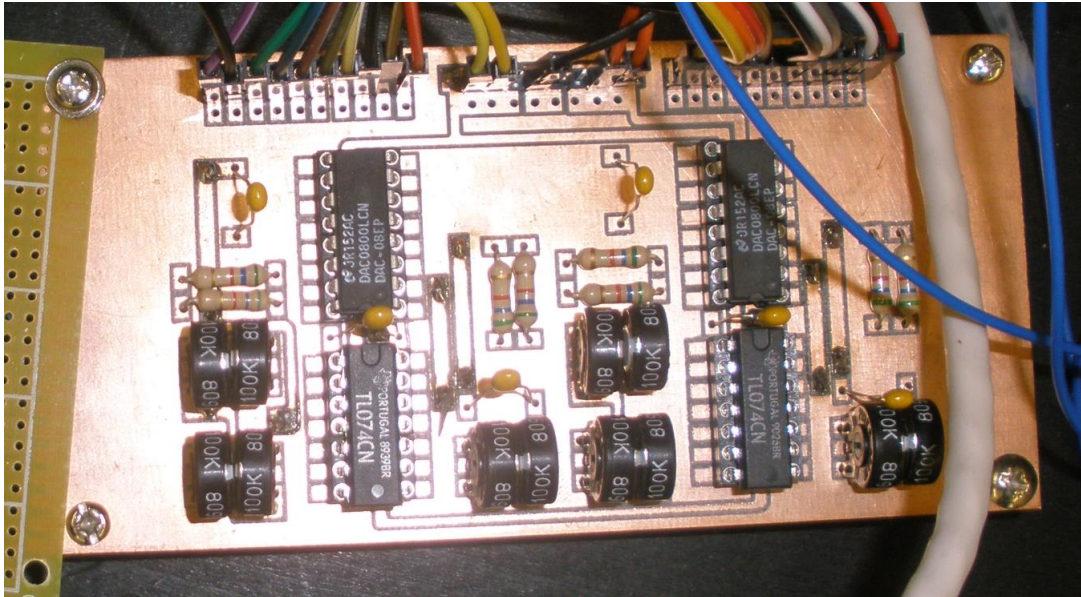


Ilustración 200 - Placa de gestión de motores terminada.

3.2.- COMPONENTES DE LA PLACA

Como ya se dijo anteriormente, la placa de gestión de motores está conformada por dos sistemas de control, uno para cada motor. Cada uno de los sistemas de control contiene un circuito DAC0800 convertidor digital analógico, una etapa amplificadora que adecúa la tensión analógica de salida conformada por un circuito amplificador TL074 y una serie de potenciómetros para ajustar la tensión de salida a las necesidades del circuito. A continuación se describen a fondo cada uno de los circuitos integrados.

3.2.1.- DAC0800. CONVERTIDOR DIGITAL – ANALÓGICO.

El DAC0800 es un convertidor digital analógico de ocho bits, monolítico y de corriente de salida de alta velocidad. Su tiempo de conversión ronda los 100 ns. El error máximo con respecto al fondo de la escala equivale a ± 1 LSB ($\pm 17,58$ mV) y su alimentación puede realizarse en el rango desde los 4,5 V hasta los 18 V. El porcentaje de resolución del convertidor es igual a 0,39 % y si consideramos que la tensión de referencia es igual a 4,5 V la resolución con respecto al fondo de escala (4,5V) será igual a 17,58 mV.

El diagrama de conexiones del circuito puede observarse en la imagen inferior.

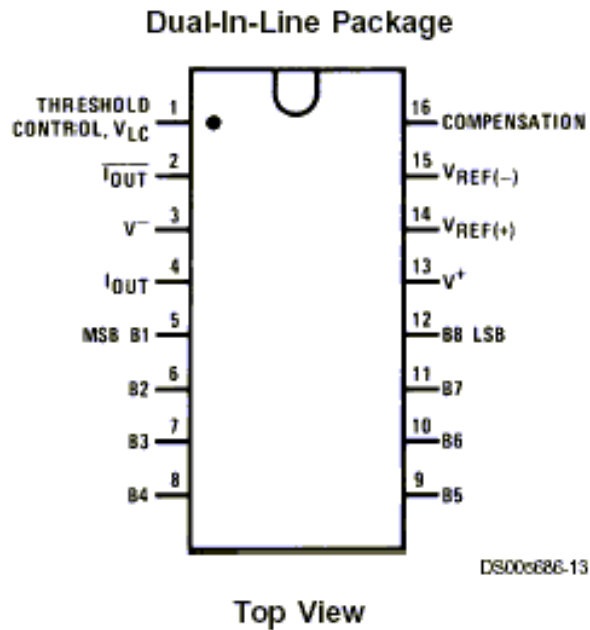


Ilustración 201 - DAC0800 Diagrama de pines

El bus de datos de la placa principal se interconecta con el bus de datos del convertidor (B1-B8) a través de uno de los 8255. Internamente, el circuito convertidor está constituido por un convertidor digital analógico con resistencia ponderada.

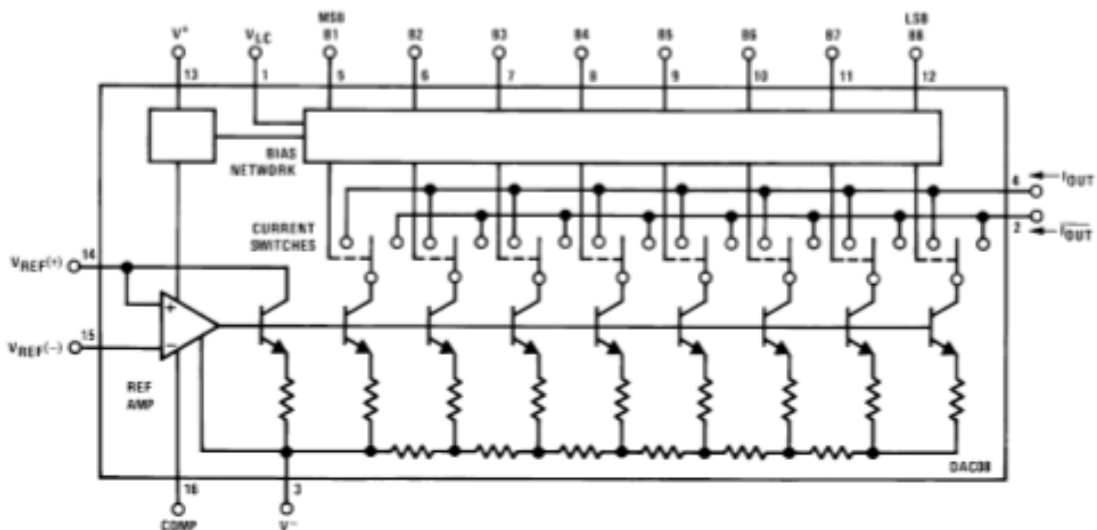
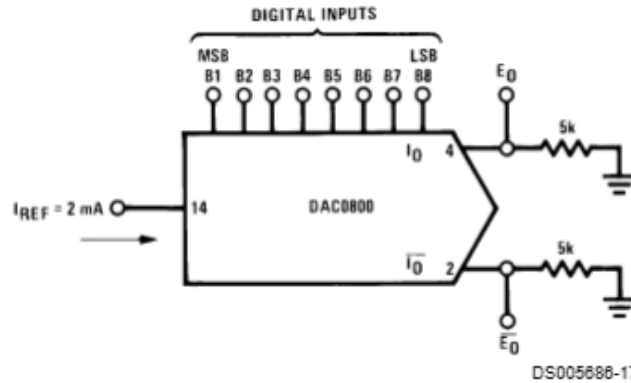


Ilustración 202 - DAC0800 Diagrama de bloques

El DAC0800 posee varias configuraciones descritas en su data sheet que permiten obtener a la salida diversas series de tensiones analógicas. A continuación se describen las más importantes.

Configuración negativa básica unipolar

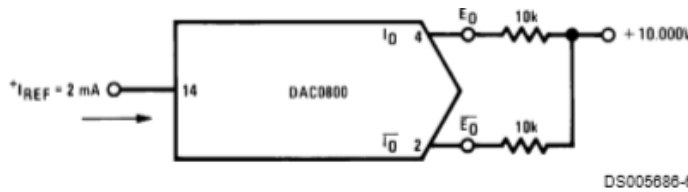


	B1	B2	B3	B4	B5	B6	B7	B8	I_0 mA	\bar{I}_0 mA	E_0	\bar{E}_0
Full Scale	1	1	1	1	1	1	1	1	1.992	0.000	-9.960	0.000
Full Scale-LSB	1	1	1	1	1	1	1	0	1.984	0.008	-9.920	-0.040
Half Scale+LSB	1	0	0	0	0	0	0	1	1.008	0.984	-5.040	-4.920
Half Scale	1	0	0	0	0	0	0	0	1.000	0.992	-5.000	-4.960
Half Scale-LSB	0	1	1	1	1	1	1	1	0.992	1.000	-4.960	-5.000
Zero Scale+LSB	0	0	0	0	0	0	0	1	0.008	1.984	-0.040	-9.920
Zero Scale	0	0	0	0	0	0	0	0	0.000	1.992	0.000	-9.960

Ilustración 203 - DAC0800 Configuración básica unipolar negativa

Como se observa en la imagen superior, esta configuración permite obtener a la salida del conversor una tensión variante entre 0V y casi -10V (el tope máximo dependerá del valor de la tensión de referencia).

Configuración básica bipolar

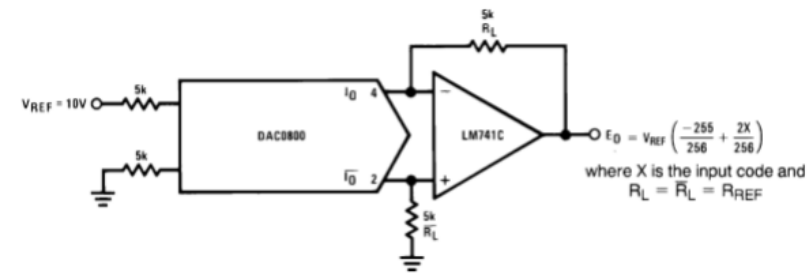


	B1	B2	B3	B4	B5	B6	B7	B8	E_0	\bar{E}_0
Pos. Full Scale	1	1	1	1	1	1	1	1	-9.920	+10.000
Pos. Full Scale-LSB	1	1	1	1	1	1	1	0	-9.840	+9.920
Zero Scale+LSB	1	0	0	0	0	0	0	1	-0.080	+0.160
Zero Scale	1	0	0	0	0	0	0	0	0.000	+0.080
Zero Scale-LSB	0	1	1	1	1	1	1	1	+0.080	0.000
Neg. Full Scale+LSB	0	0	0	0	0	0	0	1	+9.920	-9.840
Neg. Full Scale	0	0	0	0	0	0	0	0	+10.000	-9.920

Ilustración 204 - Configuración básica bipolar

En esta configuración se obtiene una tensión bipolar no simétrica a la salida.

Configuración de offset simétrico



	B1	B2	B3	B4	B5	B6	B7	B8	EO
Pos. Full Scale	1	1	1	1	1	1	1	1	+9.960
Pos. Full Scale-LSB	1	1	1	1	1	1	1	0	+9.880
(+)Zero Scale	1	0	0	0	0	0	0	0	+0.040
(-)Zero Scale	0	1	1	1	1	1	1	1	-0.040
Neg. Full Scale+LSB	0	0	0	0	0	0	0	1	-9.880
Neg. Full Scale	0	0	0	0	0	0	0	0	-9.960

Ilustración 205 - DAC0800 Configuración de offset simétrico

Esta configuración es de extrema importancia puesto que es la utilizada para la gestión de los motores. Lo que hace esta configuración perfecta para el control de la velocidad de los motores es su simetría y la posibilidad de obtener a la salida valores de tensión tanto positivos como negativos. De esta forma, puede controlarse tanto la velocidad como el sentido de giro. La ecuación que gobierna la tensión de salida de este circuito es la siguiente. La x representa el byte de entrada al convertidor en formato decimal.

$$V_{sal} = V_{REF} \left(\frac{-255}{256} + \frac{2x}{256} \right)$$

3.2.1.1.- EL DAC0800. CONVERTIDOR DIGITAL – ANALÓGICO DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

Como ya se dijo anteriormente, este circuito hace parte del sistema de gestión de motores diseñado en el marco del sistema de control de la muñeca del robot. En total se implementan dos de estos circuitos ubicados en la placa auxiliar de gestión de motores (ver la última sección de este Capítulo). Cada uno de ellos es responsable del control de uno de los dos motores de la muñeca del robot (uno del eje 5 y otro del eje 6, ver Capítulo II Sección I).

Utilizando estos circuitos se logra convertir un byte de información proveniente de la interfaz de control en una tensión analógica que servirá de consigna de velocidad y se introducirá dentro del esquema de control de velocidad original de la etapa de potencia (ver Capítulo V). La configuración del DAC0800 utilizada es la del offset simétrico en la que es imposible obtener una tensión nula a la salida del convertidor. Este problema se soluciona variando la tensión de referencia del circuito a través de software. Para obtener una tensión nula a la salida la entrada de la tensión de referencia debe conectarse a masa. Los convertidores se complementan con una etapa amplificadora que permite variar la tensión de

referencia del circuito al modificar el valor de una serie de potenciómetros. La etapa amplificadora se implementa mediante el circuito integrado TL074 que se explica en detalle a continuación.

La orden de detener los motores es dada por el usuario a través de la interfaz de control por lo que la puesta a masa de la referencia es una acción que se ejecuta a través de software. Físicamente, los pines de entrada de la tensión de referencia de los circuitos DAC0800 se encuentran conectados con el microcontrolador a través de los bits 0 y 1 del puerto C del circuito 8255 – 7 ubicado en la placa auxiliar 8255 – 8253 (ver última sección de este Capítulo).

3.2.2.- TL074. CIRCUITO AMPLIFICADOR.

El TL074 es un circuito integrado compuesto por cuatro amplificadores operacionales independientes entre sí. La tecnología interna de los amplificadores es del tipo J_FET (“junction field effect transistor”) lo que permite un comportamiento con tasas altas de velocidad. El encapsulado es de tipo DIP con 14 patillas y su alimentación se realiza a $\pm 12V$. Su diagrama de pines puede observarse en la imagen inferior:

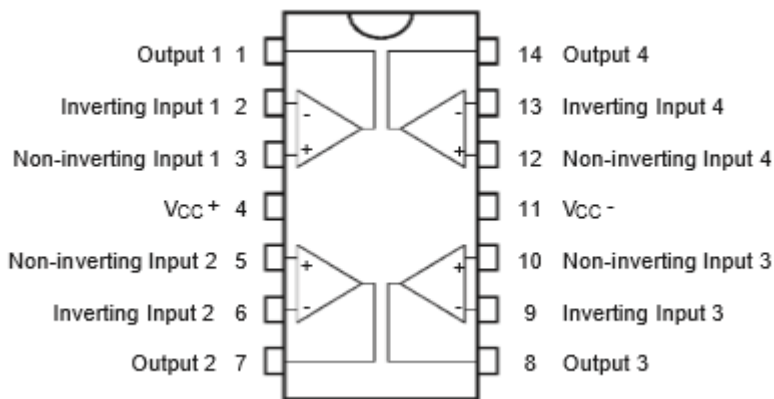


Ilustración 206 - TL074 Diagrama de pines

Dentro de la placa de gestión de motores este circuito cumple con la función de ajustar la tensión analógica de salida a valores útiles dentro del marco del control de movimiento de los motores y limitados por la tensión bipolar de saturación del circuito. Para ello, se conecta con la salida analógica del convertidor DAC0800 descrito anteriormente como se observa en la imagen inferior:

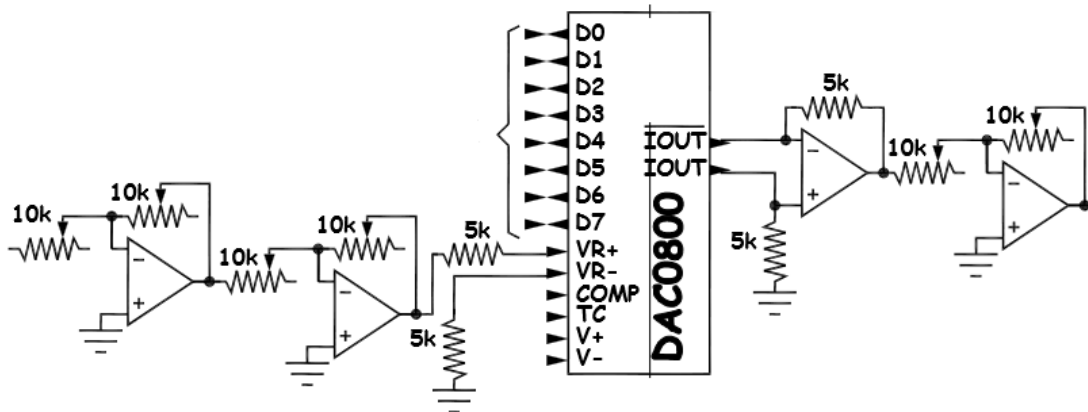


Ilustración 207 - Conexión DAC088 y TL074

Como se observa en la imagen la tensión de referencia del convertidor entra a través de la primera etapa amplificadora compuesta por dos de los cuatro amplificadores operacionales del TL074. De esta forma y puesto que la patilla de referencia negativa VR- está conectada a masa, se asegura un cero absoluto a la salida de convertidor cuando se quiera detener por completo uno de los motores.

Tanto la etapa de entrada como la de salida se complementan con una serie de potenciómetros y resistencias que permiten ajustar el nivel de la tensión.

3.2.2.1.- EL CIRCUITO AMPLIFICADOR TL074. DENTRO DEL SISTEMA DE CONTROL DISEÑADO.

Como ya se dijo anteriormente, el circuito TL074 constituye la implementación de una etapa amplificadora complementaria dentro del sistema de gestión de motores a través de los convertidores digitales - analógicos DAC0800. Estos circuitos se añadieron al sistema de gestión del usuario para permitir ajustar la tensión de referencia del convertidor y por lo tanto la tensión de salida. De esta forma se logra una salida de tensión positiva y negativa completamente simétrica.

4.- PLACA DE TRATAMIENTO DE SEÑALES.

El control de posición de los motores del brazo robótico es posible gracias a un encoder solidario al eje de rotación de los motores que permite conocer la posición exacta de los mismos mediante un proceso de conteo de pulsos. En el conteo de los pulsos de los encoder intervienen tanto elementos físicos como de software programados en rutinas del microcontrolador.

Podría pensarse en realizar el conteo de los pulsos conectando la señal de salida del encoder directamente a una línea de interrupciones del microcontrolador, en este caso, a una de las líneas de uno de los de 8259 (maestro y esclavo) conectados al microcontrolador en la placa principal e control. Sin embargo, se comprobó empíricamente mediante el estudio con osciloscopio de las señales del encoder que no era viable la conexión directa mencionada anteriormente. La señal observada poseía numerosos picos y distorsiones que

imposibilitarían el conteo y resultarían en valores de posición erróneos y por tanto inútiles en el marco de control del brazo. En la imagen inferior pueden observarse las señales puras en el osciloscopio.

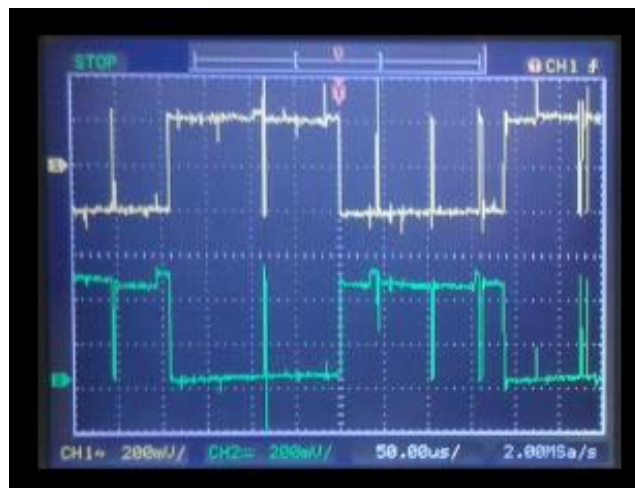


Ilustración 208 - Señales del encoder sin tratamiento

Para solucionar este problema se planteó el uso de una placa de tratamiento de las señales realizada para otro proyecto anterior. Esta placa fue diseñada específicamente para el tratamiento de las señales de los encoder del brazo robótico. Posee seis entradas para seis señales diferentes, el encoder tiene seis señales de salida. Dos señales principales S1 y S2 que informan sobre la velocidad de giro y la posición, puesto que se encuentran desfasadas 90° entre sí. Una tercera señal llamada TOP emite un pulso cada 400 pulsos de S1 o S2, que equivalen a una vuelta del motor. Las últimas tres señales son las señales invertidas de S1, S2 y TOP. El tratamiento de la señal se realiza a través de un proceso de filtrado. Además, la señal atraviesa una serie de optoacopladores, amplificadores e inversores, todo con el fin de obtener la mejor señal a la salida. Como en el caso de las placas anteriores, esta placa fue diseñada en AutoCAD. Sin embargo, se fabricó siguiendo el método fotográfico descrito en el apartado de la placa auxiliar 8253- 8255. En las imágenes inferiores se observa el diseño de la placa tanto del lado de componentes como del lado de pistas, respectivamente.

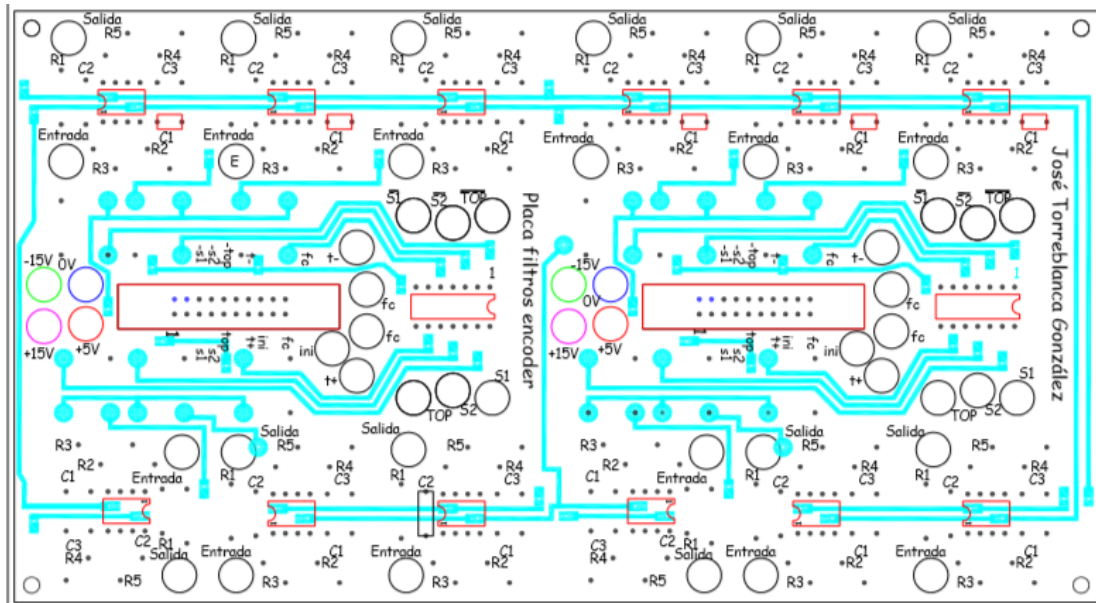


Ilustración 209 - Placa de tratamiento de señales. Lado de componentes

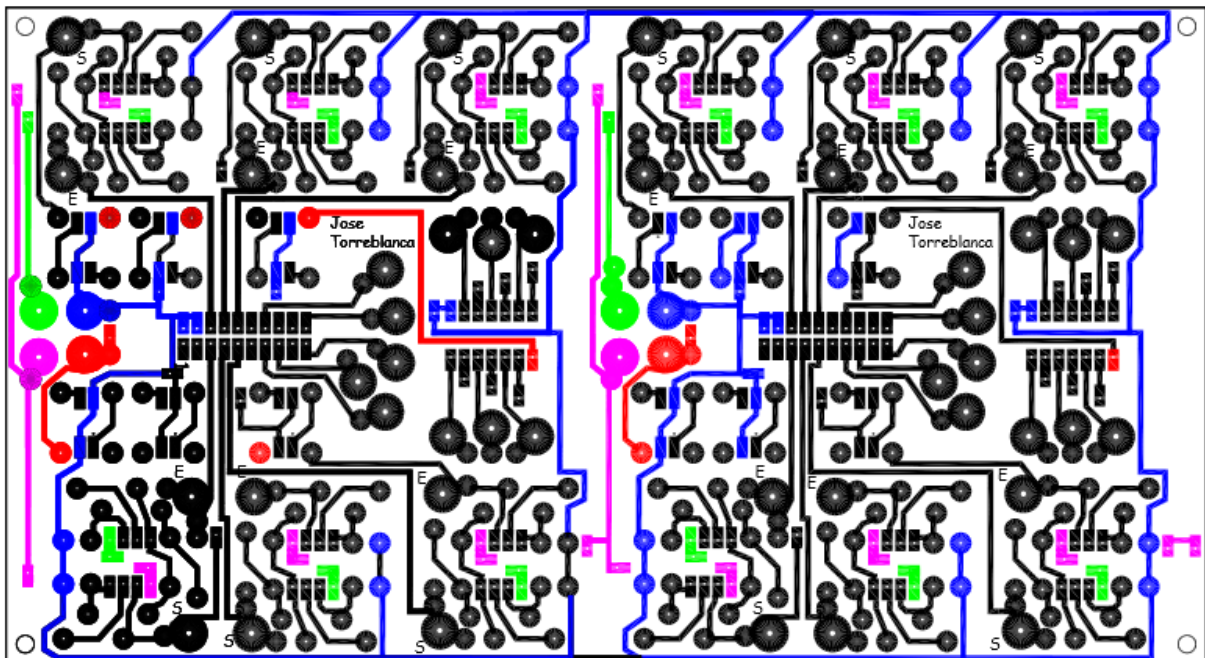


Ilustración 210 - Placa de tratamiento de señales. Lado de Pistas

En la imagen inferior se observa la señal a la salida de la placa de tratamiento.

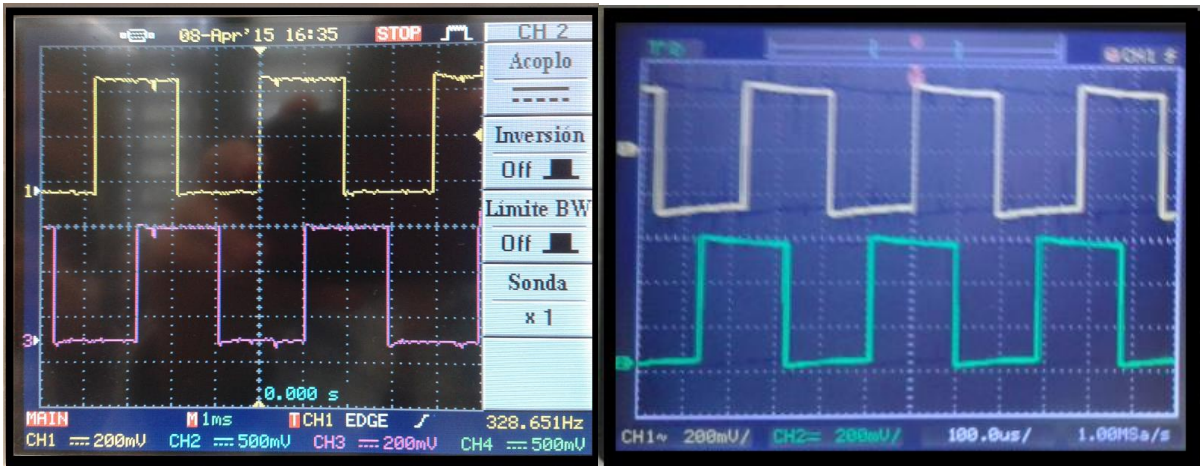


Ilustración 211 - Placa de tratamiento de señales. Señales después del proceso de filtrado

Como puede concluirse al comparar las señales antes y después de la placa de tratamiento, las segundas constituyen una señal mucha más apta para el contaje. La señal tratada carece de picos y distorsiones y en ella se diferencia claramente el valor lógico “1” del “0”. La señal una vez tratada se conecta al 8259 maestro para así poder realizar el proceso de contaje de pulsos mediante software, es decir, mediante una rutina encargada de incrementar una serie de contadores programada dentro de la memoria de código del microcontrolador. En la imagen inferior se observa la placa terminada.

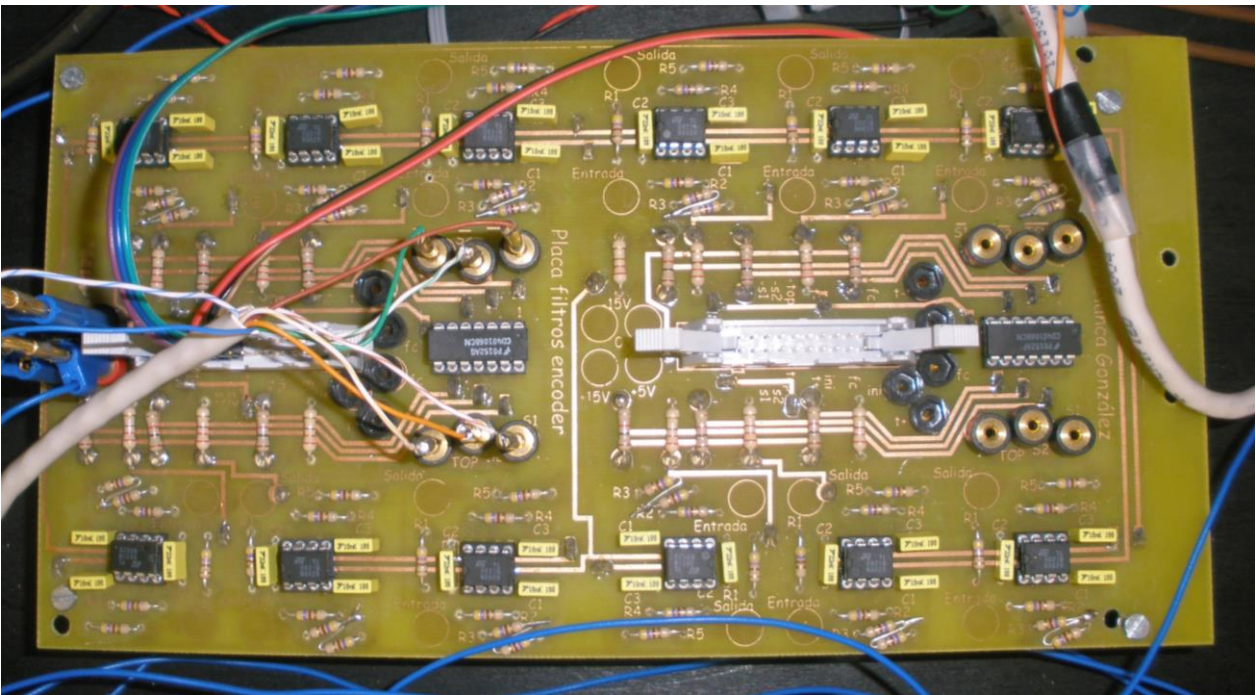


Ilustración 212 - Placa de tratamiento de señales terminada.

4.1.- COMPONENTES DE LA PLACA.

Como ya se dijo anteriormente, la placa está compuesta por una serie de filtros, optoacopladores, inversores y amplificadores. A continuación se describen en detalle cada uno de los circuitos.

4.1.1.- TL082. CIRCUITO AMPLIFICADOR.

El TL082 es un circuito amplificador compuesto por dos amplificadores operacionales con tecnología tipo J-FET de alta velocidad, al igual que los TL074 descritos anteriormente. Se alimenta a $\pm 12V$ y su diagrama de pines se observa en la imagen inferior.

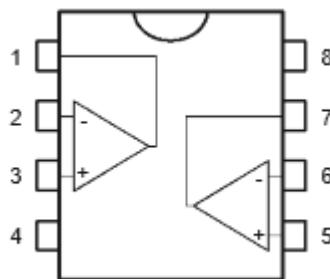


Ilustración 213 - TL082 Diagrama de pines

4.1.2.- OPTOACOPLADORES.

Un optoacoplador es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo LED que satura un componente optoelectrónico, normalmente en forma de fototransistor o fototriac. De este modo, se combinan en un solo dispositivo semiconductor un fotoemisor y un fotorreceptor, conectados ópticamente. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar para aislar eléctricamente a dispositivos muy sensibles.

En el esquema de la placa de filtro los optoacopladores son un elemento más de adecuación de la señal. La adaptación optoeléctrica que realizan permite desacoplar un bloque funcional del circuito de otro, en este caso, la salida de señales del encoder de la entrada de las mismas a los circuitos integrados encargados del conteo (8253). Así, las interferencias y ruidos que posee la señal a la salida de los encoder se anulan casi completamente, obteniendo una señal de pulsos definidos y fácilmente contabilizables. En la imagen inferior se observa el esquema básico de un optoacoplador con receptor tipo fototransistor.

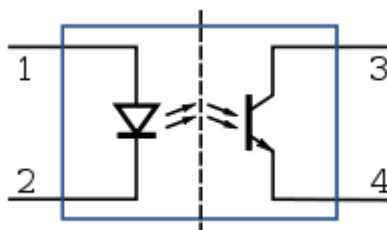


Ilustración 214 - Circuito básico del optoacoplador con fototransistor.

5.- PLACA AND – OR.

Esta placa se realizó dentro del proceso de tratamiento informático de las señales de los encoder. Por cuestiones de programación y para realizar un correcto contaje de los pulsos de los encoder se debe de refrescar el valor de los contadores antes de conectar a su entrada las señales del sensor. La solución encontrada se basa en el envío “manual” de pulsos desde el puerto A de uno de los dos circuitos 8255 de la placa 8255 – 8253 (8255 – 8).

Como puede deducirse, es necesario contar con un sistema electrónico que permita el envío de los pulsos manuales a la vez que prohíbe la entrada de los pulsos de los encoder. El circuito ideado es bastante simple, contiene puertas AND y OR conectadas como se observa en la imagen inferior.

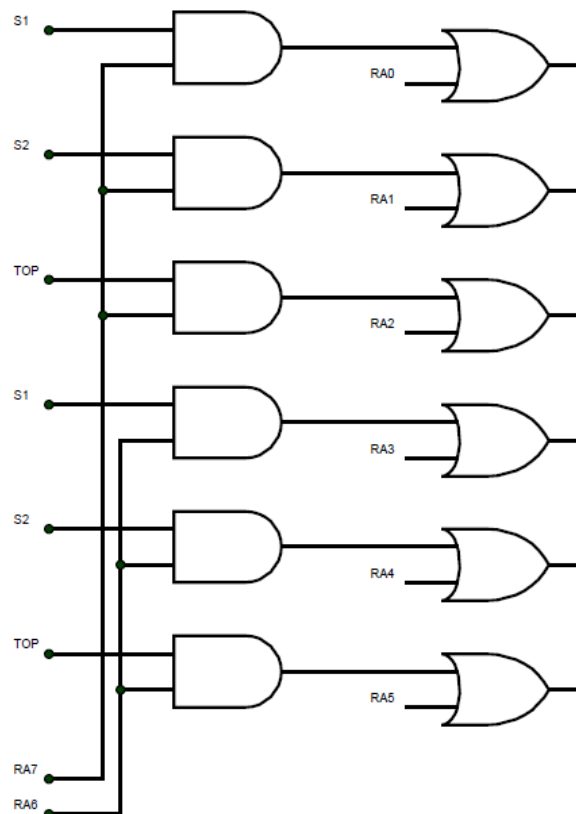


Ilustración 215 - Esquema eléctrico. Placa AND - OR.

Los puertos RA0 – RA7 referidos en la imagen corresponden con el puerto A del circuito 8255 – 8. Como cada uno de los dos motores posee un encoder es necesario duplicar el circuito. El funcionamiento del circuito es el siguiente: cuando las salidas RA6 o RA7 se ponen a 1 permiten la entrada de las señales de los encoder (S1, S2 y TOP) al sistema de contaje. En el caso contrario, de tener un 0 en RA6 o RA7, la entrada de las señales de los encoder se inhabilita. En este punto se produce el envío de pulsos manuales a los contadores principales (contador 8253 1 y 2) a través de las salidas RA0 – RA5.

El sistema también permite una lectura de las señales en el mismo instante, puesto que al inhabilitar el contaje el valor de los contadores quedará estático, pudiendo realizar mediciones certeras del valor de los contadores. En la siguiente imagen se observa la placa.

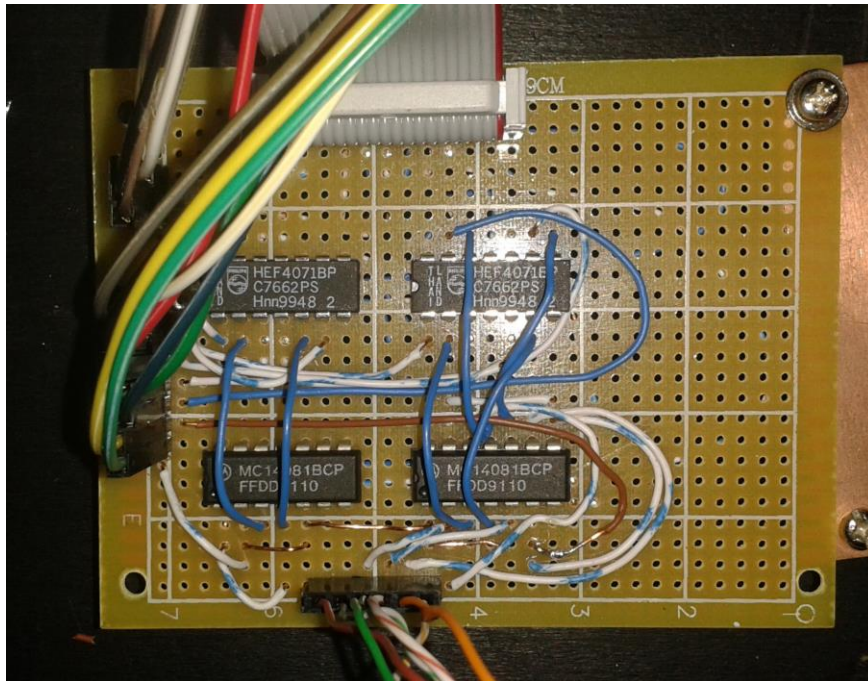


Ilustración 216 - Placa AND - OR

En la imagen inferior se observa en detalle la entrada de las seis señales provenientes de los dos encoder de los ejes 5 y 6.

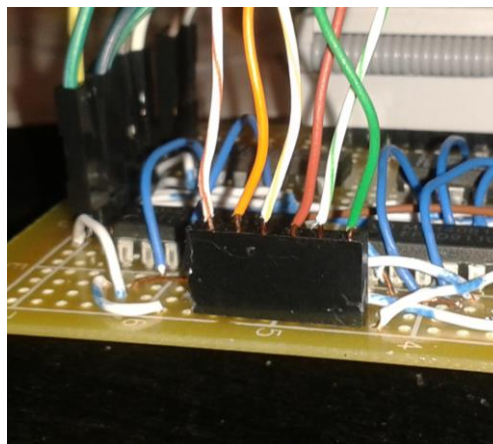


Ilustración 217 - Placa AND - OR. Detalle conexión señales encoder.

5.1.- COMPONENTES DE LA PLACA.

Como se observa en la imagen los circuitos integrados utilizados son dos 4071 que incluye cuatro puerta lógicas OR cada uno, y dos 4081 que incluyen cuatro puertas lógicas AND cada uno. En las imágenes inferiores se observa el patillaje y circuito interno de cada uno.

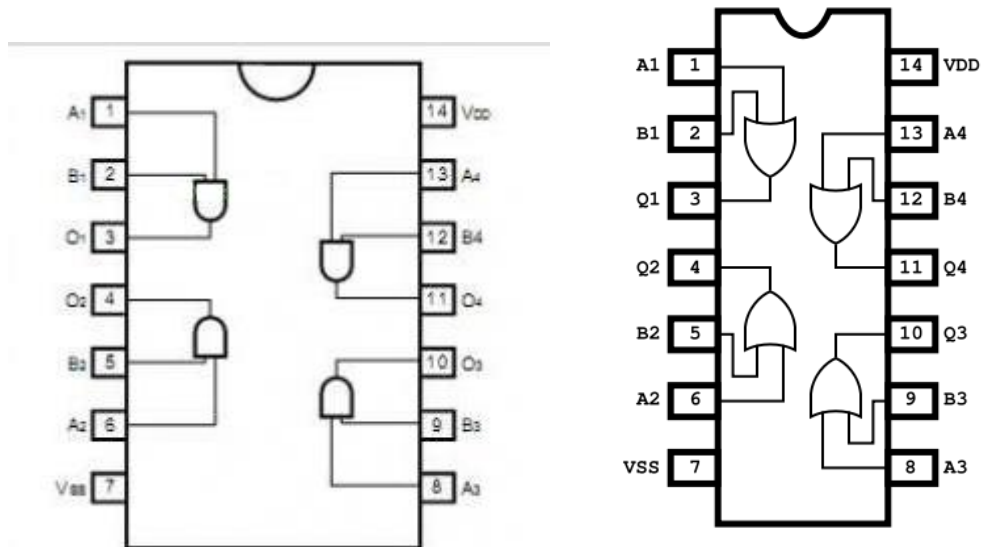


Ilustración 218 - Patillaje y circuito interno del 4081 (izquierda) y el 4071 (derecha).

Capítulo V

Sistema de control diseñado

Sección I

INTRODUCCIÓN

1.- OBJETIVO DEL SISTEMA DE CONTROL DISEÑADO.

El objetivo del sistema de control diseñado es el de controlar desde el ordenador los movimientos de la muñeca del robot. Este control incluye por tanto a los ejes 5 y 6 del robot original y sus respectivos movimientos característicos. Para un correcto manejo de la muñeca del robot se debe conocer y gobernar tanto la velocidad como la posición de los ejes, a la vez que se tiene un control seguro y confiable de la etapa de potencia, que involucra la alimentación de la electrónica de control y de los motores y un conocimiento del estado del robot, adquirido gracias a su sistema de sensores.

Una vez conseguido un control manual de la muñeca del robot basado en la fiabilidad de los movimientos y la seguridad del usuario, se pretendió programar una serie de maniobras que permitan observar el rango de movimiento de la muñeca y el nivel de control que se posee, tanto en parámetros de velocidad como de posición. Estas maniobras se concibieron como un acercamiento a lo que son los movimientos reales exigidos a un robot industrial en un ambiente de trabajo normal.

Por último, se planteó como objetivo la programación de una interfaz gráfica de control de usuario. Esta interfaz se realizó en el programa Labview y está basada en la comunicación serie entre el ordenador y el microcontrolador central del sistema de control. Constituye la interacción entre los subsistemas de control de velocidad y posición. De esta forma, se establece el culmen del sistema de control permitiendo al usuario interactuar en tiempo real con la muñeca del robot al ser capaz de controlar sus movimientos y obtener datos instantáneos en pantalla de su posición.

2.- SUBSISTEMAS DE CONTROL DISEÑADOS: VELOCIDAD Y POSICIÓN.

Para el gobierno de la posición y la velocidad de los ejes se diseñaron sendos subsistemas de control implementados en las placas de control (diseñadas y auxiliares) descritas en el Capítulo IV Sección III del presente trabajo. Los subsistemas de control están basados en el control a través de microcontrolador, concretamente se utilizó el microcontrolador PIC16F877A programado en lenguaje ensamblador (ver Capítulo VII). El microcontrolador es el punto central del sistema de control e interacciona con el exterior a través de circuitos que aumentan sus prestaciones como los contadores 8253 y los gestores de interrupciones 8259 (descritos en el Capítulo IV Sección II). De esta forma, se logra un control descentralizado en la medida en que las señales externas obtenidas son tratadas por circuitos externos al microcontrolador y solo cuando es necesario la información viaja al control central para ser tratada y generar otras nuevas señales de control para actuar sobre los actuadores, en este caso los motores de corriente continua de los ejes de la muñeca. El microcontrolador atenderá cada una de las señales (de entrada y salida) a petición del usuario o al producirse una interrupción que requiera atención inmediata.

El subsistema de control de la posición se basa en la correcta adquisición de las señales de los encoder de los ejes del robot y su posterior lectura. La lectura de estas señales se realiza mediante cuatro circuitos contadores 8253, dos para cada uno de los encoder. Los contadores 8253 cuentan decrementando dos bytes (desde un máximo de 65535 en decimal) por lo que el sistema de contaje debe de contener un proceso de recargo de contadores. Este recargo se realiza mediante el envío de comandos desde la interfaz gráfica. En el marco del control manual de la muñeca, el proceso de lectura de contadores se realiza por envío de datos a la interfaz gráfica aprovechando la comunicación serie, una vez el microcontrolador recibe el comando correspondiente del usuario. A su vez, el subsistema incluye la programación de la maniobra de búsqueda de la posición inicial, en la que intervienen las señales provenientes de los sensores inductivos originales de cada uno de los ejes del robot (ver Capítulo II Sección III).

Las señales de entrada del subsistema de control de posición provienen de sensores ubicados en la muñeca del robot. Estos sensores son tres: los sensores inductivos, los finales de carrera y los encoder, descritos en el Capítulo II Sección III. Las señales de los dos primeros sensores interaccionan con el sistema de control en forma de interrupciones gestionadas por el microcontrolador través de los circuitos 8259. Los sensores inductivos intervienen en la rutina de búsqueda de la posición inicial de la muñeca mientras que los finales de carrera son principalmente elementos de seguridad que delimitan el espacio de trabajo de los ejes (ver Capítulo II Sección I). Las señales de salida del encoder son las entradas de los circuitos contadores 8253 una vez atraviesan la placa AND – OR de control de contaje (ver Capítulo III Sección III).

El subsistema de control de velocidad crea la señal de consigna del bucle de control de la velocidad original del robot. Esta señal de consigna es una señal de tensión variable entre -10V y +10V, las tensiones positivas o negativas indican un sentido u otro de giro del eje de la muñeca. En miras a aumentar la seguridad del usuario esta velocidad se limitó por software trabajando dentro de un rango de tensiones entre -2,5V y +2,5 aproximadamente. El control

se basa en la obtención de dicho rango de tensiones mediante un circuito conversor digital – analógico DAC0800 complementado con una etapa amplificadora (ver Capítulo II Sección II). El byte a convertir es enviado por el usuario al circuito conversor a través de la interfaz gráfica utilizando la comunicación serie entre el ordenador y el microcontrolador. De esta forma, se tiene un control completo del movimiento de cada uno de los ejes de la muñeca.

3.- SUBSISTEMA DE CONTROL DE LA ALIMENTACIÓN DE LA ETAPA DE POTENCIA.

El subsistema de control de la alimentación de la etapa de potencia le permite al usuario aislarse físicamente de la conexión de la alimentación tanto de 230V de la electrónica de control original como de la alimentación de potencia de los motores (ver Capítulo III Sección I). Se diseñó con el objetivo de aumentar la seguridad para el usuario sobretodo en la conexión del contactor trifásico que conecta la alimentación de potencia de los motores.

El subsistema se implementó mediante tres relés de corriente continua, activos por bajos y que conmutan a +5V. El primero de los relés alimenta a 230 V toda la electrónica de control original de la etapa de potencia. El segundo relé alimenta el contactor trifásico que proporciona 400V a la etapa de alimentación de los motores de corriente continua (ver Capítulo II Sección II). Por último, el tercer relé envía la señal de frenos desactivados a la electrónica de control original. Esta señal es una de las señales de verificación que deben de ser correctas para permitir el funcionamiento de los motores según el esquema de control original del robot (ver Capítulo III Sección III). Mediante la activación consecutiva de los tres relés el usuario alimenta la etapa de potencia de una forma segura y su vez tiene la posibilidad de cortar la alimentación inmediatamente en cualquier momento. Este subsistema no está vinculada a la interfaz gráfica, su control es puramente manual.

4.- INTERFAZ GRÁFICA EN LABVIEW.

La interfaz gráfica en Labview constituye la ventana a través de la cual el usuario puede observar los parámetros de control y gobernar los movimientos de la muñeca del robot. La interfaz está compuesta por una serie de botones de comando y etiquetas indicadoras que permiten al usuario realizar el envío de códigos de actuación al microcontrolador y recibir la información enviada en respuesta, respectivamente.

Como ya se dijo anteriormente, la base de la interfaz es su conexión con el microcontrolador utilizando la comunicación serie. Este tipo de comunicación se instaura mediante el envío de una serie de códigos numéricos hexadecimales desde el ordenador al microcontrolador. Una vez recibidos los códigos el microcontrolador los lee e interpreta a qué dirección de la memoria de código debe dirigirse, ejecutando en cada momento la rutina deseada por el usuario. De esta forma, mediante una interfaz clara y sencilla se logra un nivel de control superior de fácil entendimiento para el usuario y en comunicación constante con el microcontrolador.

Labview es un software de grandes capacidades que permite crear entornos completamente amoldados a las necesidades del proyecto. De ahí que la interfaz creada constituya el nivel de control perfecto para las necesidades del sistema de control diseñado.

Es importante atender a los retardos de programación que son necesarios incluir en el entorno de Labview para que funcionen en armonía el microcontrolador y la comunicación serie, puesto que la comunicación con el ordenador consume un tiempo extra que ha de tenerse en cuenta dentro de la secuencia de funcionamiento del microcontrolador.

Sección II

SUBSISTEMAS DE CONTROL

1.- SUBSISTEMA DE CONTROL DE POSICIÓN.

Como ya se dijo anteriormente, el subsistema de control de la posición es uno de los tres subsistemas de control que compone el sistema de control diseñado junto con el subsistema de control de velocidad, el subsistema de control del encendido de la etapa de potencia y la interfaz gráfica, que permite la interacción entre subsistemas. Este subsistema permite conocer la posición de los dos ejes de la muñeca del robot mediante la adquisición de señales de los encoder de cada uno de los motores y su posterior tratamiento. A su vez, permite la realización de la maniobra de búsqueda del cero mediante las señales provenientes de los sensores inductivos de cada uno de los ejes de la muñeca.

Para llegar a este punto fue necesario realizar en primer lugar una adaptación de las señales de los encoder mediante una etapa de filtrado. Después, se plantearon dos posibles sistemas de contaje de pulsos; uno utilizando los circuitos integrados gestores de interrupciones 8259 y otro utilizando los contadores triples 8253, método escogido finalmente. Una vez se lograron programar los cuatro contadores utilizados en el control de posición se procedió a la programación de la maniobra inicial de movimiento de la muñeca, la maniobra de búsqueda de la posición inicial. Para programar esta maniobra se utilizaron las señales provenientes de los sensores inductivos originales de cada uno de los ejes del robot (ver Capítulo II Sección III) conectadas a los circuitos 8259. Así, las señales de los sensores inductivos son interpretadas por el microcontrolador como interrupciones que debe atender inmediatamente. Finalmente, el subsistema de control está conectado con la interfaz gráfica de forma que las órdenes de mando, como la de lectura de contadores o inicio de la maniobra de búsqueda de la posición inicial, son ejecutadas por el usuario desde el ordenador.

A continuación se describirá en detalle todo el subsistema de control, los elementos que los componen, los problemas que surgieron durante su programación e implementación y su conexión con la interfaz gráfica del sistema de control.

1.1.- ESTUDIO Y ADAPTACIÓN DE LAS SEÑALES DE LOS ENCODER.

En primer lugar, se realizó un estudio de las señales a la salida de los encoder. Para ello se empleó un motor de corriente continua externo y se acopló el encoder a su eje. Se utilizó un osciloscopio para observar las señales. Las señales vistas en el osciloscopio contenían gran cantidad de ruido y picos que sin duda generarían errores de conteo. En la imagen se observa dos de las señales estudiadas con el osciloscopio.

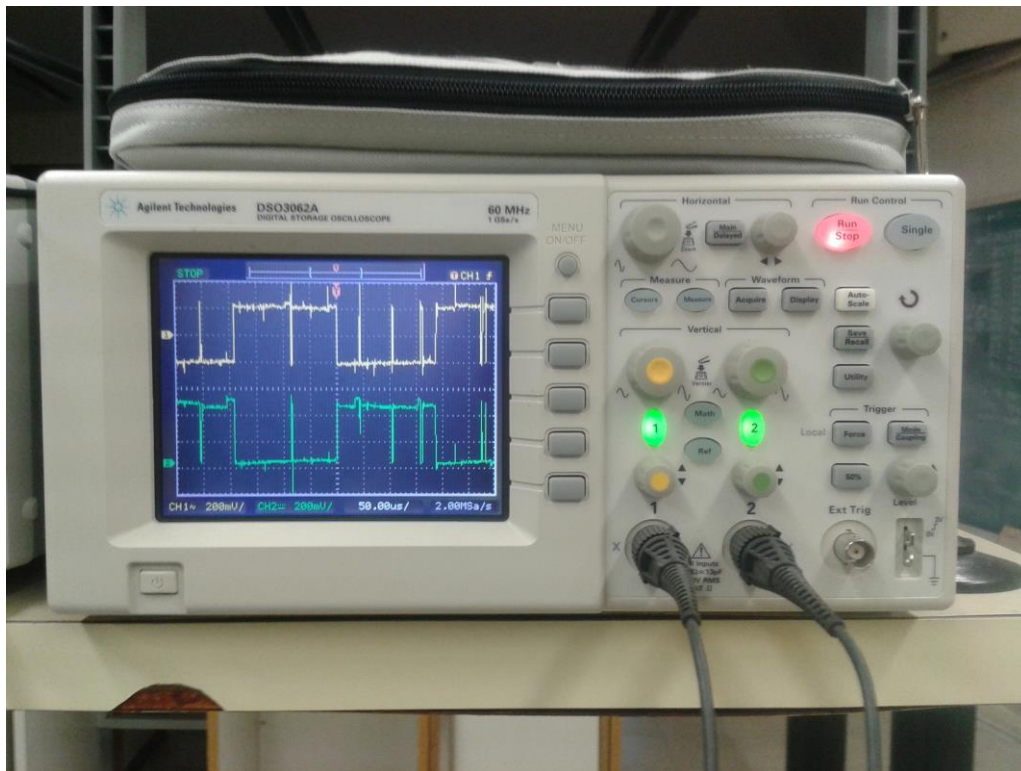


Ilustración 219 - Señales a la salida del encoder vistas con osciloscopio.

Para solucionar este problema se planteó el uso de una etapa de filtrado implementada sobre una placa perteneciente a un proyecto anterior (ver Capítulo III Sección III). La placa posee entradas para dos encoder, es decir, se pueden tratar en total doce señales (ver Capítulo IV Sección III). Sin embargo, se decidió que sólo se trataría con las señales originales y no con las complementadas, que también se encuentran a la salida del encoder, por lo que sólo se utiliza la mitad de la placa. Se tratan seis señales en total tres de cada encoder S1, S2 y la señal de TOP. S1 y S2 son dos señales desfasadas entre sí 90° que permiten conocer el sentido de giro del motor. La señal de TOP indica que el motor ha dado una vuelta completa y equivale a 400 pulsos de las dos señales anteriores.

Utilizando esta placa se logró filtrar la señal por medio de una serie de optoacopladores, amplificadores e inversores y obtener a la salida una señal sin tanto ruido apta para ser la entrada del sistema de conteo. En la imagen inferior se observa una de las pruebas realizadas a la placa de adaptación, la fuente de alimentación se encarga de polarizar correctamente todos los circuitos.

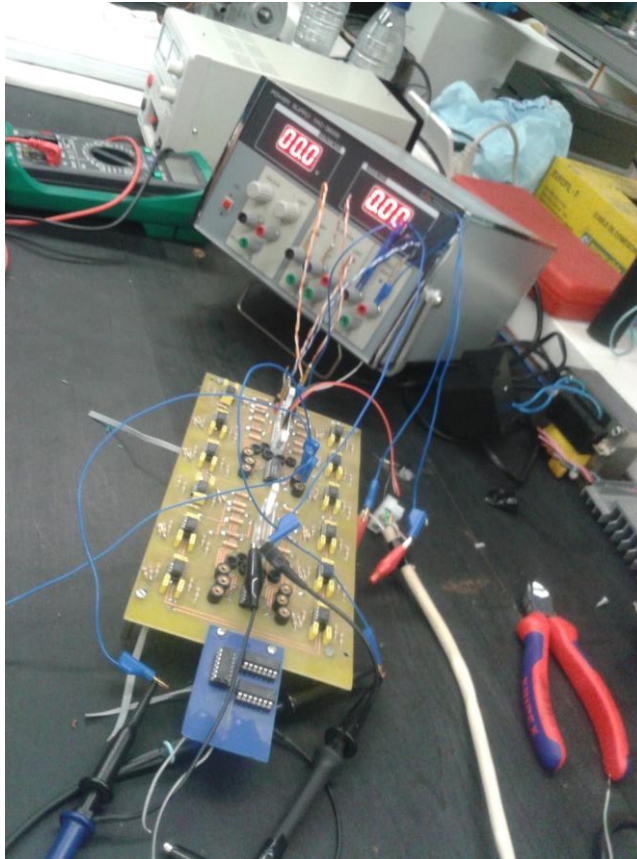


Ilustración 220 - Pruebas con la placa de tratamiento de señales.

En la imagen inferior se observan las señales vistas en el osciloscopio después de pasar por la etapa de tratamiento.

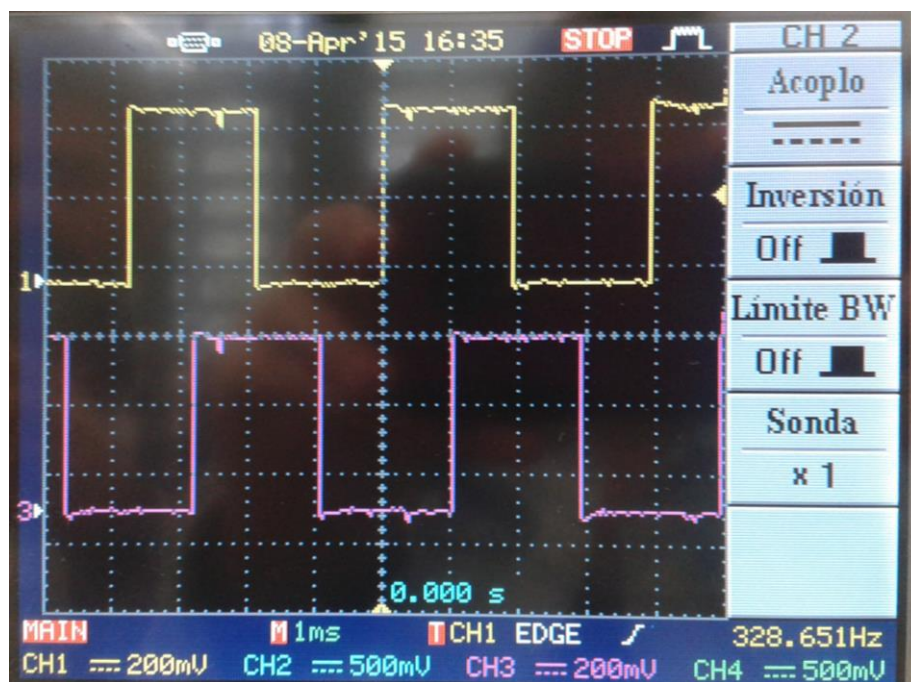


Ilustración 221 - Señales de los encoders a la salida de la placa de tratamiento.

1.2.- PLANTEAMIENTO DEL SISTEMA DE CONTAJE.

Una vez conseguida la adaptación de las señales del encoder se procedió al planteamiento de distintos sistemas de conteo de los pulsos. La idea básica del proceso de conteo es la de conseguir traducir los pulsos generados por los encoder a un dato numérico que pueda incluirse dentro del sistema general de control. Siguiendo esta idea se plantearon inicialmente dos posibles sistemas de conteo. Por un lado, conectar las señales de los encoder a la entrada de los circuitos 8259 y realizar el conteo mediante un sistema basado en interrupciones. Por otro, se planteó conectar las señales a la entrada de los circuitos contadores 8253, realizando el conteo al decrementar un contador desde un valor asignado.

1.2.1.- CONTAJE POR INTERRUPTIONES UTILIZANDO EL CIRCUITO 8259.

Los circuitos integrados 8259 son circuitos gestores de interrupciones que permiten aumentar las prestaciones del microcontrolador en este campo, puesto que éste solo posee una línea de entrada para interrupciones de este tipo (ver Capítulo IV Sección II). Como se dijo anteriormente, la idea que se tuvo fue conectar las señales del encoder a la entrada de estos circuitos de forma que cada uno de los pulsos del encoder activara una interrupción. La interrupción generada por el 8259 llegaría al microcontrolador y lo enviaría a un bucle en el que se aumentaría el valor de un registro determinado. De esta forma, se tendría siempre a disposición del usuario el valor de conteo guardado en un registro interno del microcontrolador y fácilmente accesible en cualquier momento.

Cuando se implementó el sistema descrito en el párrafo anterior se observó que el sistema de conteo por interrupciones no era viable. El microcontrolador se desbordaba debido a una cantidad excesiva de interrupciones ejecutándose al mismo tiempo. Además, la jerarquía de las interrupciones impedía que el microcontrolador atendiera a más de una señal a la vez, pues una vez se ejecutaba la interrupción de mayor prioridad ésta anulaba las demás peticiones de interrupción. El problema de la jerarquía de las interrupciones se intentó solucionar reprogramando el 8259 para que atendiera las interrupciones de forma rotativa (ver Capítulo IV Sección II), es decir, una vez gestionada una interrupción ésta pasa a ser la interrupción de menos importancia, dándole la oportunidad al microcontrolador de gestionar las demás interrupciones. Sin embargo, el fallo persistió y en un momento se decidió optar por el conteo a través de los contadores 8253, que se explica a continuación.

1.2.2.- CONTAJE UTILIZANDO LOS CIRCUITOS CONTADORES 8253.

1.2.2.1. - DESCRIPCIÓN DEL ESQUEMA BÁSICO DE CONTAJE

El conteo mediante circuitos contadores 8253 se basa en provocar que cada uno de los pulsos de las señales de los encoder decremente en una unidad el valor del registro de un contador de dos bytes, que puede contar desde cualquier valor programado por el usuario (desde un máximo de 65535, pues el registro de cada contenedor es de dos bytes). Cada uno de los 8253 posee tres contadores independientes y de las mismas características.

Como las señales a tratar de los dos encoders son seis en total basta con dos contadores para realizar el conteo principal. Sin embargo, con vistas a poder realizar un conteo tan extenso como se quisiese se implementaron otros dos contadores encadenados de forma que una unidad de conteo de los contadores encadenados equivale a 65535 unidades de conteo de los contadores principales.

La lectura de los registros de contadores se realiza mediante una orden de usuario desde la interfaz gráfica y el resultado del conteo se envía por comunicación serie de vuelta a la misma. El esquema básico de conteo con los 8253 se observa en la imagen siguiente.



Ilustración 222 - Esquema de conteo mediante contadores 8253.

1.2.2.2. - FRECUENCIA DE ENTRADA Y RECARGA DE CONTADORES.

Es importante tener en cuenta que los 8253 pueden contar señales de entrada de hasta un máximo de 2,6 MHz. Si consideramos que los encoders proporcionan 400 pulsos por vuelta y que el rango de velocidad en el que se funcionarán los motores va desde 0 hasta 1000 rpm la frecuencia máxima de entrada que se tendrá es de uno 400 KHz, valor de entrada viable para realizar un conteo con los 8253.

Dentro del sistema de conteo con los 8253 es fundamental contar con un proceso de recarga de contadores al alcance del usuario. Este proceso es una rutina dentro de la memoria de código del microcontrolador que cambia el valor de uno de los registros internos del 8253 (ver Capítulo IV Sección III) que permite reiniciar el conteo decremental desde cualquier valor dentro del rango de 0 hasta 65535, herramienta de programación muy útil para el usuario. Además, el proceso de recarga de contadores se programó para realizarse desde la interfaz gráfica de forma que la recarga de los contadores siempre está al alcance de un click.

1.2.2.3. - PLACA AND - OR.

Como se dijo en el Capítulo IV Sección III del presente trabajo, la placa AND – OR es una de las placas auxiliares que componen la implementación física del sistema de control diseñado. Está compuesta por una serie de puertas lógicas AND y OR que permiten el envío de pulsos tanto manuales como a través de los encoder a las entradas de los 8253.

El objetivo de esta placa fue el de solventar un problema en el proceso de recarga de los contadores. Debido a la programación interna de los 8253 no basta con cambiar el registro de contaje para que éste empiece a decrementar el contador desde el valor indicado. Es necesario que lleguen a las entradas dos pulsos para que el valor de contaje se actualice. El problema se resolvió enviando dos pulsos a través de la circuitería lógica implementada en esta placa. De esta forma se recargan correctamente los contadores en el valor deseado. Además, el envío de pulsos manuales permitió realizar pruebas sobre el sistema sin necesidad de conectar la etapa de potencia, ahorrándole un estrés innecesario al sistema.

Para el envío de pulsos manuales se utilizó el puerto A del 8255 – 8 ubicado en la placa auxiliar 8255 – 8253 (ver Capítulo IV Sección III). Para enviar pulsos es necesario habilitar el circuito receptor 8255 – 8 enviando una señal de masa a su “chip select” a través del bus de datos de la placa principal que está conectado con el bus de datos de la placa 8255 – 8253. Una vez habilitado el circuito se envía la información a través el bus de datos y de una de las básculas 74LS373 de la placa principal (ver Capítulo IV Sección II). El comando de envío de pulsos se realiza a través de la interfaz gráfica. En la siguiente imagen se observa un esquema explicativo del proceso.

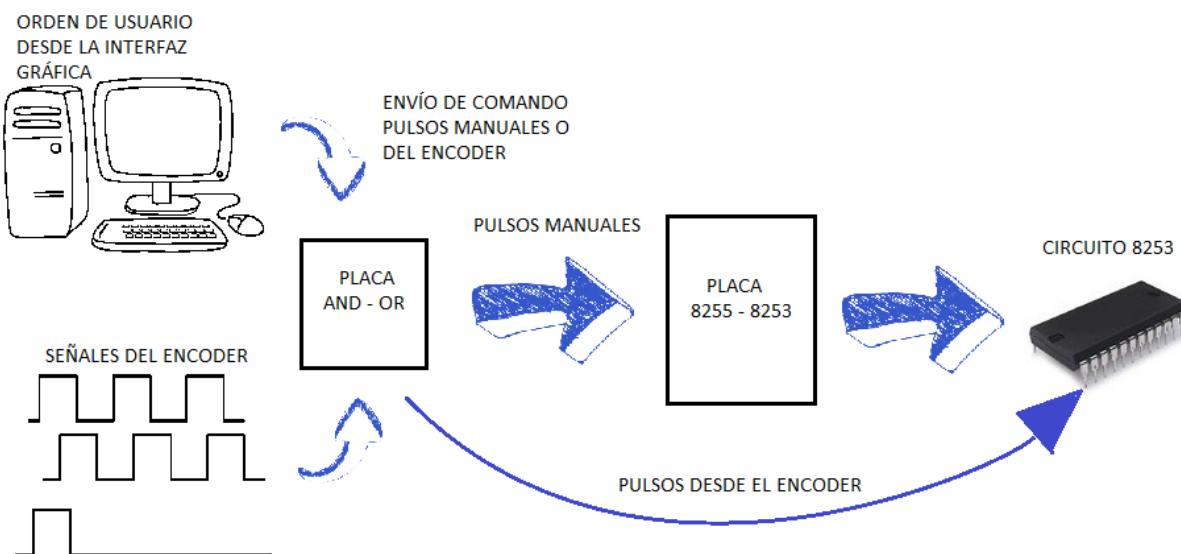


Ilustración 223 - Esquema del envío de pulsos manuales al 8253.

1.3.- MANIOBRA DE BÚSQUEDA DE LA POSICIÓN INICIAL.

Como se dijo anteriormente, el subsistema de control de la posición incluye tanto el estudio y tratamiento de las señales provenientes de los encoders como el desarrollo de la maniobra de búsqueda de la posición inicial realizada a partir de las señales provenientes de los sensores inductivos originales de cada uno de los ejes del robot (ver Capítulo II Sección III).

La maniobra de búsqueda de la posición inicial es básica dentro del marco de la realización de un sistema de control para un elemento robótico. Lo anterior dado que todo movimiento de los ejes de la muñeca, en este caso, es un movimiento controlado tanto en parámetros de posición como de velocidad y de ahí que se necesite un punto de partida para todos los sistemas de control. Los contadores de pulsos necesitan reprogramarse a su valor inicial y los ejes de los motores deben conducirse en un movimiento suave hasta una posición convenida como posición de arranque de todos los movimientos.

1.3.1.- ESTUDIO DE LOS SENSORES INDUCTIVOS.

Como ya se dijo anteriormente, los sensores inductivos son parte de equipamiento original del robot y su función original era también la de proporcionar información de posición para la maniobra de búsqueda de la posición cero. Están compuestos por un sensor y una pletina metálica sobre los engranajes externos de ambos ejes de la muñeca (ver Capítulo II Sección III).

Los sensores deben de alimentarse desde una fuente de alimentación externa de 25V y proporcionan una señal de +5V excepto cuando la pletina enfrente al sensor, momento en que la señal cae a 0 V. Debido a este funcionamiento la señal debe de atravesar un inversor para obtener una señal de +5V en el momento que la pletina cruza en frente del sensor.

La posición inicial convenida es aquella en la que el eje 5 se encuentra vertical y el sensor del eje 6 sobre su respectiva pletina. Sin embargo, la posición original de la pletina del eje 5 lo hacía detenerse formando un ángulo de 30° con la vertical. Para solucionar este problema se debió de reposicionar la pletina, desajustando unas tuercas adosadas al engranaje externo y girando la chapa metálica hasta ubicar la pletina metálica en la posición correcta.

1.3.2.- PROGRAMACIÓN DE LA MANIOBRA.

Para programar la maniobra se debió de decidir en primer lugar la forma de entrada de la señal al circuito de control diseñado. Se decidió incluir la señal como una entrada al circuito gestor de interrupciones 8259. El sistema ideado gestiona las interrupciones provocadas por los sensores de los ejes 5 y 6 dentro de la rutina programada en el microcontrolador,

La maniobra se programa en tres subrutinas de la memoria de código del microcontrolador. La primera subrutina es ejecutada por una orden dada por el usuario a través de la interfaz gráfica. Esta orden indica al robot que debe moverse en un sentido

determinado y habilita el funcionamiento de la interrupción conectada al eje 5 del robot. Este movimiento obliga al eje 5 a pasar por la posición convenida como posición inicial, formando un ángulo recto con la mesa. Al activarse la interrupción del sensor inductivo del eje 5 se ejecuta la segunda subrutina de la maniobra: se detiene el motor del eje 5, se inhabilita la interrupción del eje 5 y se habilita la del eje 6 y se inicia el movimiento de este eje. Cuando se activa la interrupción del eje 6 se ejecuta la última subrutina de la maniobra que detiene el motor del eje 6 e inhabilita esta interrupción. En la imagen inferior se observa un esquema de los movimientos que realiza la muñeca del robot durante la maniobra.

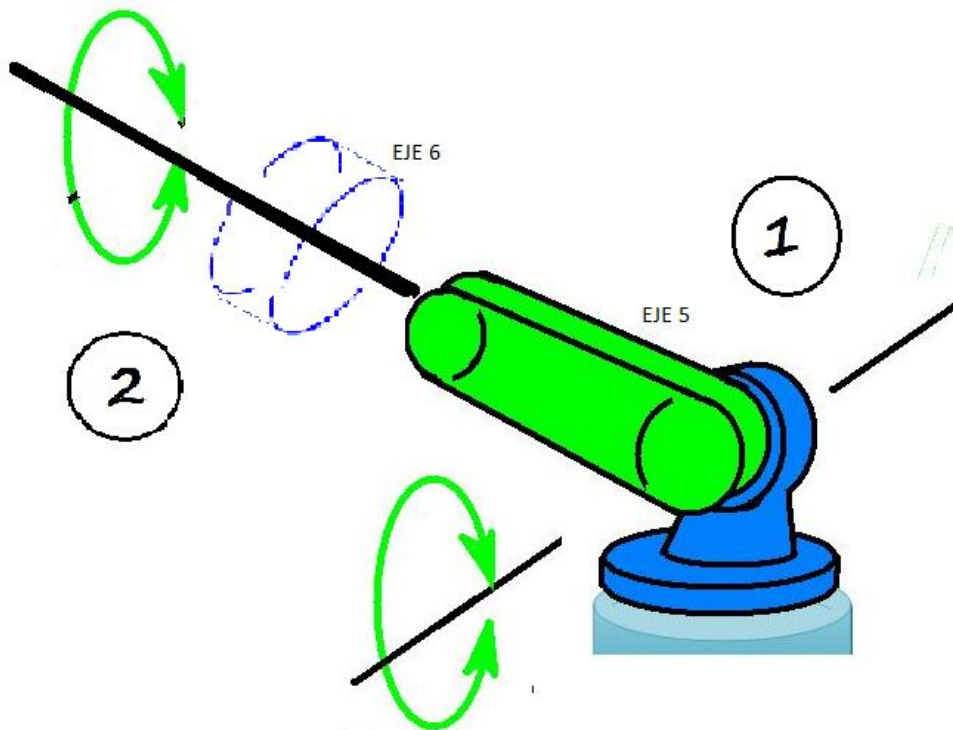


Ilustración 224 - Movimientos de la muñeca durante la maniobra de búsqueda de la posición inicial.

1.3.3.- PROBLEMA DE DESCONEXIÓN DEL MICROCONTROLADOR.

Durante el proceso de programación de la maniobra se producía un fallo constante de desconexión del microcontrolador al ejecutarse la interrupción del sensor inductivo del eje 5. Este problema se debe a los picos de corriente asociados a la etapa de potencia del robot puesto que el microcontrolador es muy sensible a este tipo de alteraciones cuyo efecto inmediato es una desconexión de todo el sistema de control, dejando al usuario sin posibilidades de controlar el movimiento de la muñeca.

Para solucionar este problema se implementaron las subrutinas de la maniobra de modo que las interrupciones solo están habilitadas en los momentos justos en que se necesitan, como se describió en el punto anterior, Al inhabilitar las interrupciones se acotan las posibles fuentes de error que conduzcan a la desconexión del microcontrolador. Además, se programaron dos órdenes desde la interfaz gráfica que permiten al usuario activar o desactivar las interrupciones cuando lo necesite.

1.3.4.- PROBLEMA DE RECONOCIMIENTO DE LA INTERRUPTIÓN.

En el proceso de llevar a la práctica la idea teórica de programación de la maniobra se observó que las señales de los sensores llevadas a través de circuitos inversores al microcontrolador no eran reconocidas. El problema se debe a un conflicto de rebote de señales que le impide al microcontrolador recibir correctamente la señal de control.

Para solucionar este problema se conectó un condensador en paralelo con la entrada de la señal al circuito inversor. El condensador se carga al recibir la señal e impide que se presente el problema de rebote de la misma. El circuito implementado se describe con detenimiento en el Capítulo II Sección III del presente trabajo.

1.4.- DELIMITACIÓN DEL ESPACIO DE TRABAJO. SENSORES FINALES DE CARRERA.

El espacio de trabajo de la muñeca del robot debió de delimitarse por cuestiones de seguridad atendiendo principalmente al movimiento del eje 5. Este movimiento se restringió a 180° mediante la instalación de sensores finales de carrera a ambos lados del eje como se indica en el Capítulo II Sección III del presente trabajo.

Las señales de los sensores finales de carrera son tratadas dentro del esquema del sistema de control diseñado como interrupciones que llegan al microcontrolador a través de los circuitos gestores de interrupciones 8259. Cada sensor final de carrera activa una interrupción. En el caso del sensor final de carrera del lado derecho (mirado desde el lado A convenido, ver Capítulo II Sección I) la subrutina asociada detiene los motores e inicia un movimiento del eje 5 hacia la izquierda con el objetivo de desactivar el sensor y dejar al sistema en un punto operativo.

1.4.1.- PROBLEMA DE RECONOCIMIENTO DE LA INTERRUPTIÓN.

Al igual que en el caso de los sensores inductivos, se tuvo problemas de rebote de la señal al conectar directamente la salida del sensor al microcontrolador. La solución aplicada fue la misma que en el caso anterior: conectar un condensador en paralelo con la señal del sensor que evite el rebote de la señal. El circuito se describe en el Capítulo II Sección III.

1.5.- PROGRAMACIÓN DE MOVIMIENTOS AUTOMÁTICOS DE LA MUÑECA.

Una vez diseñado e implementado correctamente el subsistema de control de la posición debió de establecerse una estructura de programación de maniobras automáticas del robot. Como el subsistema de control de la posición se basa en el conteo de los pulsos de los encoders la idea que se tuvo para la programación de movimientos automáticos de la muñeca gira en torno al mismo principio.

Para realizar un movimiento controlada de la muñeca se realizaron numerosas pruebas de contaje del número de pulsos que constituye un movimiento de 180° del eje 5 y un giro de 360° del eje 6. Una vez conocido el número de pulsos de cada maniobra pueden realizarse asociaciones del número de pulsos de movimientos acotados.

La estructura de programación de maniobras automáticas ideada se basa en recargar los contadores independientemente con el número de pulsos de la maniobra a realizar. Posteriormente se daría una orden de arranque de los motores y se esperaría a la señal de "out" de los circuitos 8253 conectada a una señal de interrupción gestionada por los 8259, que indica que se han contado el número de pulsos indicados en la recarga. De esta forma se pueden programar maniobras complejas a través de procesos que involucran continuas recargas de los contadores y órdenes de espera de interrupciones.

2.- SUBSISTEMA DE CONTROL DE VELOCIDAD.

Este subsistema de control compone, junto con el subsistema de control de la posición y el sistema de control del encendido de la etapa de potencia, el sistema de control de la muñeca del robot diseñado. El objetivo de este subsistema de control de la velocidad es el de generar la consigna de velocidad para introducirla dentro del esquema de control original de la etapa de potencia (ver Capítulo III Sección I). La consigna de velocidad de la etapa de control original era una tensión variable entre $-10V$ y $+10V$ que le indica tanto velocidad como sentido de giro. $+10V$ indica la máxima velocidad en un sentido de giro.

Para diseñar este subsistema de control se debió empezar por corregir un fallo que impedía que los motores funcionasen al darles alimentación. Estudiando los planos eléctricos originales del robot y los planos de la circuitería electrónica de control se logró descubrir que el fallo provenía de una mala conexión de la señal proveniente del tacómetro. Se llegó a la conclusión de que para un correcto funcionamiento del sistema la polaridad del generador de corriente continua (ver Capítulo II Sección III) deberá estar acorde a la polaridad de la alimentación de los motores.

Una vez solventado este fallo se procedió al planteamiento del sistema a utilizar para generar la consigna de velocidad. El sistema ideado consta de un convertidor digital – analógico DAC0800 y una etapa amplificadora implementada mediante el circuito integrado TL074. De esta forma se logró convertir un byte (que consta de ocho bits de información) de información en una tensión variable entre $-10V$ y $+10V$ obtenida a la salida del circuito. Al implementar el circuito original se observaron una serie de fallos debidos a malas conexiones en la placa principal diseñada (ver Capítulo IV Secciones I y II). Debido a la gravedad de los fallos fue necesario rehacer la parte de la placa en la que se implementó la gestión de los motores, es decir, el subsistema de control de la velocidad (ver Capítulo IV Sección III).

La placa de gestión de motores tiene una relación intrínseca con la interfaz gráfica de control. Mediante una serie de botones de comando el usuario puede ejecutar movimiento con cualquiera de los dos ejes controlando su velocidad y pudiendo detenerlos en cualquier momento.

La detención de los motores es compleja debido a que el modo de funcionamiento programado en el DAC0800 (ver Capítulo IV Sección II) no permite obtener una tensión nula a la salida. La mínima tensión que puede obtenerse es de 0,040 mV que es suficiente para mover al motor debido a las bobinas que se conectan en serie con él (ver Capítulo V Sección I). Para solucionar este problema se implementó la etapa amplificadora mencionada anteriormente. La tensión nula a la salida se obtiene conectando a masa la referencia de la etapa amplificadora que normalmente se conecta a 5V. Sin embargo, al implementar el sistema de obtención de tensión nula descrito se observó que el eje 5 (ver Capítulo II Sección I) no se detenía por completo. Se llegó a la conclusión de que el movimiento se debía a una tensión remanente de la etapa de potencia. La solución final vino de la mano de la programación del software de control. Se estudió qué tensión compensaba la tensión remanente y se programó la maniobra de parada de modo que enviara esta tensión y no el cero absoluto a la salida de los convertidores. La tensión necesaria para compensar la tensión remanente que ocasionaba el movimiento del eje 5 es la equivalente al número 126 que en binario es el 1111110.

Por último, se presentó un problema de pérdida del control de los motores al reiniciar el sistema mediante la desconexión de la alimentación de las placas de control del sistema diseñado (ver Capítulo IV). Al no recibir una entrada clara los convertidores digitales analógicos no producen una tensión válida a la salida y los motores responden adquiriendo su velocidad máxima. La solución ideada fue la conexión de la salida del convertidor a la entrada de la tensión de consigna a través de un relé que conmuta a masa cuando se produce la desconexión de la electrónica de control.

2.1.- FALLO EN LA CONEXIÓN DEL TACÓMETRO GENERADOR DE CC.

La primera tarea a realizar antes incluso de pensar acerca del sistema de control a diseñar era la de solucionar el fallo que provocaba que los motores no se moviesen al darles alimentación. El fallo se derivó del hecho de que la muñeca del robot fue trasladada desde el aula original donde estaba el robot hasta el laboratorio, donde se reconectaron los cables.

Para solucionar este problema se recurrió al estudio de los manuales eléctricos del robot original y de los planos de las placas de control de la etapa de potencia (ver Capítulo III Sección III). Se llegó a la solución del problema realizando numerosas pruebas de ensayo y error considerando diversas posibles causas. Tras numerosos intentos de solucionar el problema se estudió con detenimiento la conexión del tacómetro generador de corriente continua dentro del sistema de control de la etapa de potencia. Los cables de conexión no se habían tocado así que se cambió su polaridad y se chequeó el funcionamiento de los motores. El problema está resuelto.

Una vez solucionado el fallo se concluyó que es esencial que la polaridad de la conexión de la señal de salida de los tacómetros coincida con la polaridad de la alimentación de los motores. De esta forma la señal de realimentación del bucle de control original de la velocidad de los motores (ver Capítulo III Sección I) es interpretada de forma correcta.

2.2.- SUBSISTEMA DE CONTROL DE LA VELOCIDAD DISEÑADO.

Como se dijo anteriormente, el subsistema de control de velocidad diseñado tiene por objetivo la creación de una tensión entre -10V y +10V que sirva de tensión de consigna dentro del bucle de control de la velocidad original de la etapa de potencia. Para crear esta tensión se planteó un sistema basado en un convertor digital – analógico DAC0800 (ver Capítulo IV Sección II) y el envío de un byte de información a través de la interfaz gráfica de control (ver último punto de este Capítulo).

El convertor DAC0800 posee diversas configuraciones de funcionamiento que permiten que el circuito integrado se adapte a las necesidades de cada sistema. En este caso la configuración utilizada fue la del offset simétrico en la que el convertor tiene una salida de tensión variables entre -9,96V y +9,96V, valores simétricos correspondientes a 0 y 255 en binario como se ve en la imagen inferior. El envío del byte de entrada del convertor se realiza desde la interfaz gráfica mediante la comunicación serie entre el ordenador y el microcontrolador. Físicamente el byte es enviado por el microcontrolador al convertor a través del bus de datos (puerto D del microcontrolador) y de una báscula 74LS373 al puerto A del circuito 8255 – 7 (ver Capítulo IV Sección II) ubicado en la placa auxiliar 8255 – 8253 (ver Capítulo IV Sección III). El envío de información se realiza una vez habilitado el circuito receptor mediante el “chip select” del circuito conectado directamente al bus de datos de la placa principal (al igual que el “chip select” de todos los seis circuitos de la placa 8255 – 8253).

	B1	B2	B3	B4	B5	B6	B7	B8	E _o
Pos. Full Scale	1	1	1	1	1	1	1	1	+9.960
Pos. Full Scale-LSB	1	1	1	1	1	1	1	0	+9.880
(+)Zero Scale	1	0	0	0	0	0	0	0	+0.040
(-)Zero Scale	0	1	1	1	1	1	1	1	-0.040
Neg. Full Scale+LSB	0	0	0	0	0	0	0	1	-9.880
Neg. Full Scale	0	0	0	0	0	0	0	0	-9.960

Ilustración 225 - Configuración de funcionamiento del DAC0800.

Como se observa en la imagen anterior los valores más cercanos al cero son de +40mV y -40mV. Esto ocasiona un problema en la detención de los motores puesto que es necesario un cero absoluto a la salida del convertor para detener su movimiento. Además de este percance surgió otro problema derivado de la desconexión del microcontrolador en momentos en los que se está maniobrando con la etapa de potencia cuya consecuencia es la pérdida de control de la muñeca del robot. Ambos problemas y sus respectivas soluciones se detallan en el siguiente punto.

2.3.- PROBLEMAS PRESENTADOS.

2.3.1.- PROBLEMA DE DETENCIÓN DE LOS MOTORES.

Este problema deriva del hecho de que el convertor digital - analógico no puede producir una valor nulo de tensión a la salida necesario para detener los motores. Para solucionar este problema se idearon dos sistemas que se detallan a continuación.

2.3.1.1.- SOLUCIÓN MEDIANTE LA VARIACIÓN DE LA TENSIÓN DE REFERENCIA DEL DAC0800.

Como se mencionó anteriormente, la imposibilidad de generar un cero a la salida del conversor digital analógico no permite detener el motor mediante una orden de usuario. Para solucionar este problema se planteó un sistema en el que la tensión de referencia de los circuitos DAC0800 es puesta a masa mediante software para detener los motores. Además, el conversor se complementa con una etapa amplificadora implementada mediante el circuito TL074 (ver Capítulo IV Sección III) con el objetivo de poder ajustar el valor de la tensión de regencia (de aproximadamente 5V) a las necesidades del sistema.

A través de la interfaz gráfica el usuario puede dar la orden de detener por completo los motores. Esta orden envía 0V al pin del DAC0800 correspondiente a la entrada de la tensión de referencia del circuito. Cabe destacar que se implementaron dos sistemas tipo conversor digital - analógico complementado con la etapa amplificadora, puesto que cada uno de ellos controla un único motor y la muñeca del robot consta de dos motores.

Sin embargo, una vez implementado el sistema descrito anteriormente se observó que el eje 5 no se detenía por completo mientras que el eje 6 sí lo hacía. El eje 5 giraba lentamente en sentido antihorario visto desde el lado A convenido (ver Capítulo II Sección I). Estudiando el fallo se llegó a la conclusión de que se debe a una descompensación de la etapa de potencia que afecta negativamente al sistema de control de velocidad diseñado. La solución ideada para este fallo fue la de compensar la tensión remanente mediante una tensión de igual valor pero de signo contraria proveniente del sistema conversor digital - analógico etapa amplificadora. Esta solución se explica en detalle en el punto siguiente.

En la imagen inferior se observa un esquema del subsistema de control de la velocidad diseñado.

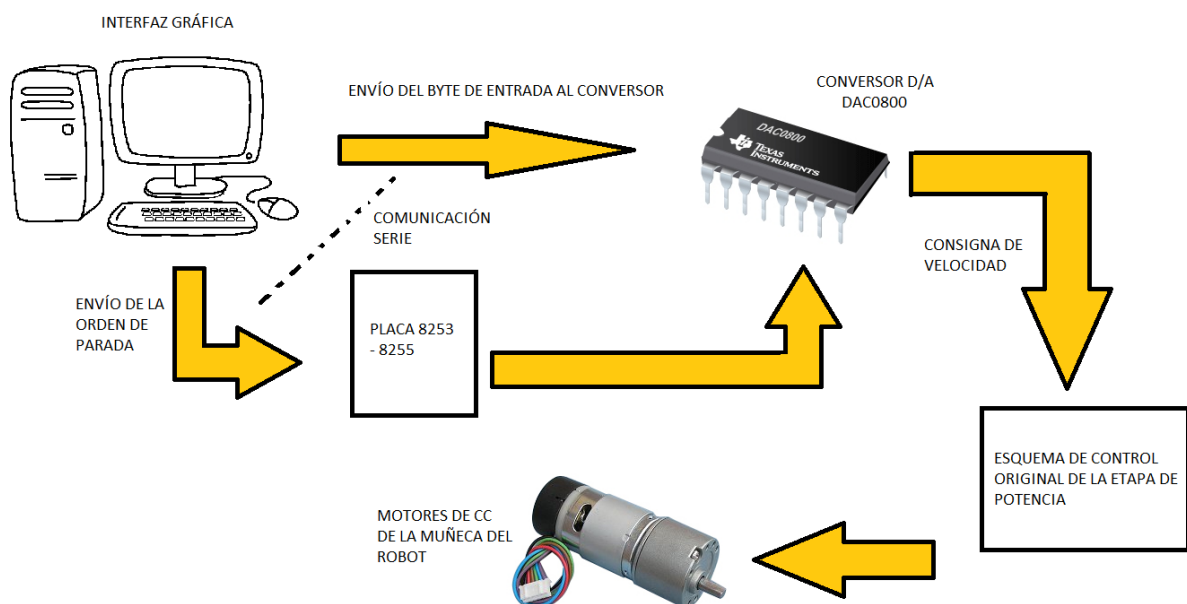


Ilustración 226 - Esquema del subsistema de control de velocidad diseñado.

2.3.1.2.- SOLUCIÓN MEDIANTE COMPENSACIÓN DE TENSIÓN POR SOFTWARE.

Para conocer el valor de tensión a compensar se midió la tensión remanente con un polímetro y realizó el cálculo del valor digital a introducir al convertor para obtener la misma tensión que la medida pero con el sentido contrario. La tensión a compensar es de 11 mV por lo que la tensión de compensación equivale a -11 mV un 126 (1111110 en binario).

Así, la orden de detención del eje 5 proveniente de la interfaz gráfica realiza un envío de un byte que contiene el número 126 en binario al bus de datos del microcontrolador que está conectado con el puerto A del 8255 – 7 (ver capítulo IV Sección III) y a su vez a la entrada del convertor digital – analógico ubicado en la placa auxiliar de gestión de motores (ver Capítulo IV Sección III).

2.3.2.- PROBLEMA DE PÉRDIDA DE CONTROL DE LOS MOTORES.

Este problema se presentó durante las diversas pruebas de funcionamiento del sistema de control al desconectar la alimentación de la electrónica de control diseñada. Al no tener tensiones de alimentación la tensión a la salida del convertor analógica no es clara y el sistema de control de la etapa de potencia original envía la orden de máxima velocidad a los motores.

Para solucionar este problema se ideó un sistema basado en relés que permite conmutar la entrada de la tensión de consigna entre la señal de salida del convertor digital analógico y la masa, tomada de la etapa de potencia. De esta forma, durante el funcionamiento normal del sistema el relé conmuta conectando la señal de salida del convertor con la entrada de la tensión de consigna, y conecta la masa con la entrada de la consigna cuando se desconecta la tensión de alimentación de la electrónica de control. En esta parte del subsistema de control de la velocidad no interviene la interfaz gráfica.

3.- SUBSISTEMA DE CONTROL DE LA ETAPA DE POTENCIA.

Este subsistema de control nació de la necesidad de manipular de forma segura el proceso de encendido de la etapa de potencia. En un principio, todas las maniobras de arranque de la alimentación de la etapa de potencia se realizaban manualmente. Era necesario conectar la alimentación de 230V, activar el contactor trifásico de la alimentación de los motores y finalmente, activar el contactor monofásico de alimentación de frenos (ver Capítulo III Sección II). Una vez se hizo constante la necesidad de realizar estas maniobras durante las numerosas pruebas de funcionamiento del sistema de control diseñado se planteó la necesidad de este subsistema de control.

El subsistema se implementó físicamente mediante tres relés de corriente continua, activos por bajo y que conmutan a 5V. Los relés utilizados admiten el paso de hasta 230V eficaces y 10A. Los tres interruptores de gobierno de los relés se encuentran localizados en un entrenador del laboratorio, como se observa en la imagen inferior.

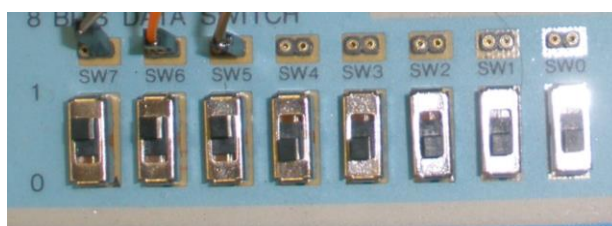


Ilustración 227 - Interruptores accionadores del subsistema de control de la alimentación de la etapa de potencia.

En la imagen inferior se observan los tres relés.

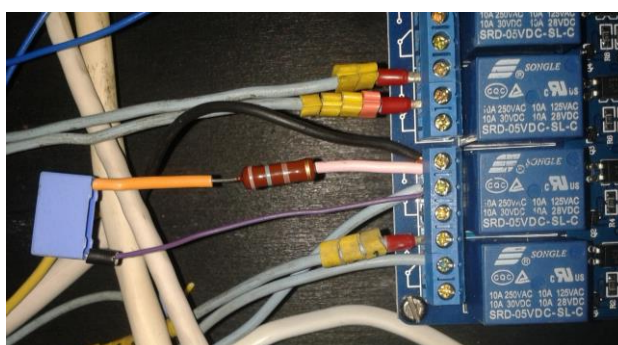


Ilustración 228 - Relés del subsistema de control de la alimentación de la etapa de potencia.

El interruptor de la izquierda activa el relé que alimenta a 230V toda la electrónica de control original de la etapa de potencia. También alimenta los ventiladores de refrigeración por lo que al conmutar el relé se oirá el ruido que producen. El interruptor central activa el relé que alimenta el contactor trifásico de potencia. Mediante este contactor se alimentan los motores a 400V (ver Capítulo III Sección II). A su vez, el contactor trifásico activa el contactor monofásico que alimenta los frenos. En este punto el sistema estaría listo para funcionar. Sin embargo, se instaló un tercer interruptor (el de la derecha) encargado de enviar al sistema de control original de la etapa de potencia la señal de “frenos desactivados” (ver Capítulo III Sección I). De esta forma, así el sistema está listo para funcionar una vez activado el interruptor central, el usuario tiene la orden final para habilitar su funcionamiento a través del interruptor de la derecha. Además, implementando este sistema se evitan numerosos arranques y paradas de la etapa de potencia, ahorrándole un estrés innecesario al sistema.

Sección III

INTERFAZ GRÁFICA

1.- INTRODUCCIÓN

La interfaz gráfica diseñada es el medio de comunicación entre el usuario y el sistema de control diseñado, descrito detalladamente en la sección anterior. Todas las funcionalidades del sistema de control se engloban en las características de la interfaz, constituyéndola como el culmen del control de la muñeca del robot. La interacción entre el usuario y el sistema electrónico se realiza a través de una serie de botones de comando y etiquetas informativas. De esta forma, el usuario envía órdenes al sistema a la vez que recibe información en tiempo real de los procesos realizados. Todas las características de los subsistemas de posición y de velocidad se complementan con la interfaz gráfica, permitiendo un control efectivo y en cierta medida sencillo de un sistema complejo.

Este software de control se diseñó pensando en la sencillez y la practicidad de manejo que requieren los sistemas de control de robots de alta gama. En estos casos, el control electrónico es en gran medida complejo. Sin embargo, las interfaces de control son sencillas, estableciendo una relación directa con el usuario que no necesita conocer en su totalidad el sistema de control para poder manipular el robot.

La interfaz se compone de una serie de módulos independientes: el gobierno de los parámetros de la comunicación serie, la gestión de la velocidad de los motores, la lectura y recarga de los registro contadores de los encoder y los botones de comando para acciones específicas. Cada uno de estos módulos se explicará en detalle en el siguiente punto.

El software de programación utilizado para crear la interfaz fue Labview de National Instruments. La programación en Labview es gráfica y permite al programador diseñar el código tal y como lo estructura en su cabeza, abstrayéndose de los lenguajes de programación convencionales a la vez que proporciona las mismas facilidades. Se escogió este programa debido a su capacidad de amoldarse a las necesidades de programación de muchos tipos de interfaces y al conocimiento previo que se tenía de él. En la imagen inferior se observa la interfaz.

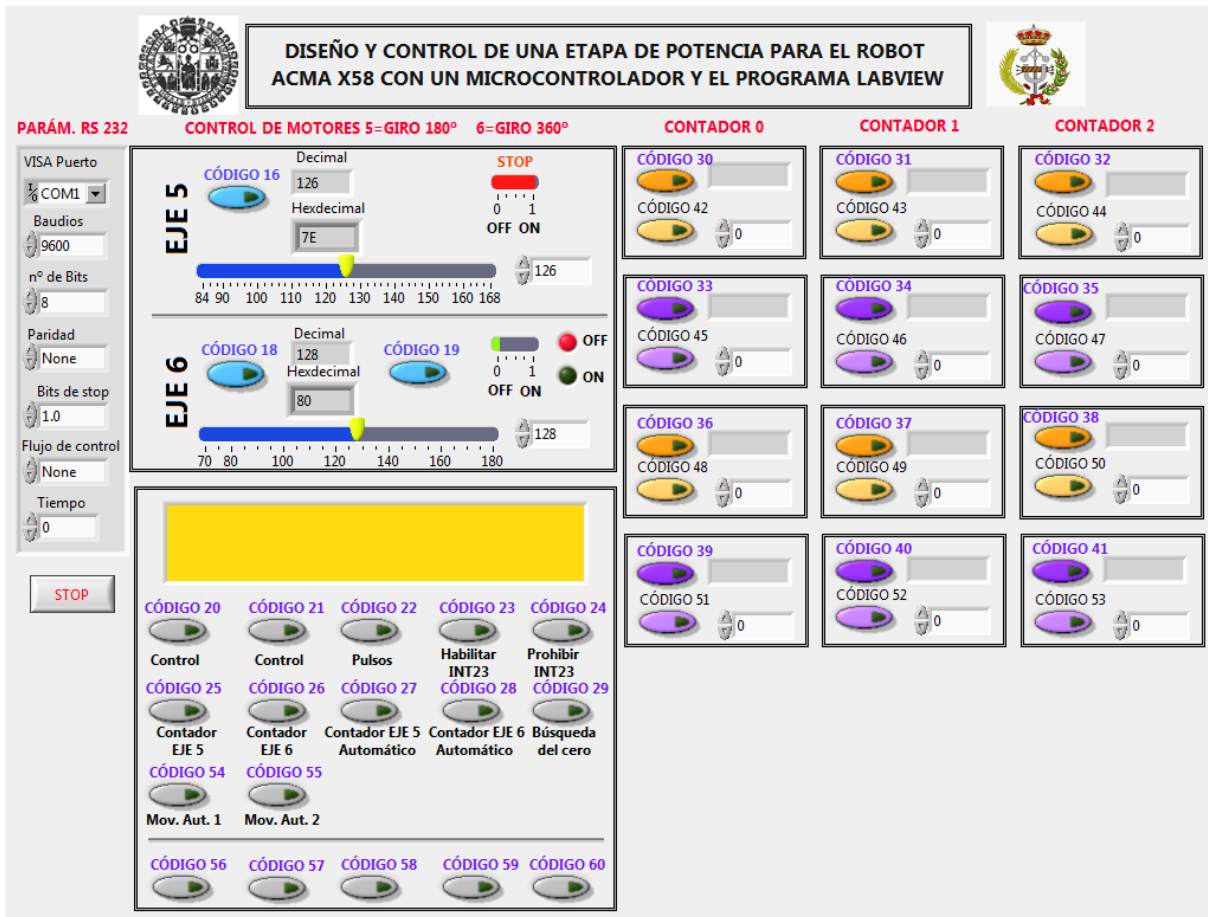


Ilustración 229 - Interfaz gráfica diseñada en Labview.

2.- DESCRIPCIÓN

Como se mencionó en la introducción, la interfaz gráfica se diseñó en módulos que gestionan diversas características del sistema de control diseñado. A continuación se describe cada uno de estos módulos.

2.1.- CONTROL DE LOS PARÁMETROS DE LA COMUNICACIÓN SERIE.

El control de los parámetros de la comunicación serie se realiza desde el módulo ubicado en la parte izquierda de la pantalla. Los parámetros de la comunicación serie con el ordenador que el usuario puede ajustar son: el puerto serie por el que se realizará la comunicación, la velocidad de la comunicación, el número de bits envueltos, el tipo de paridad de los bits, los bits de parada que indica la finalización de la transmisión, el flujo de control y el tiempo.

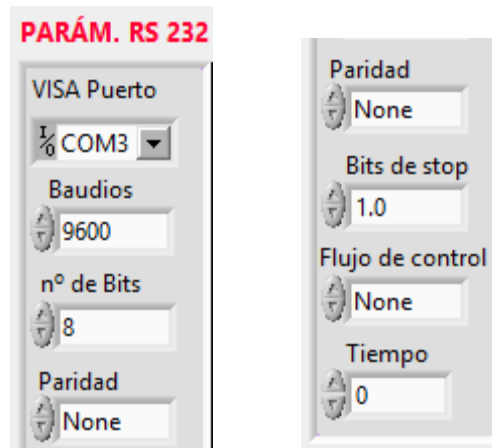


Ilustración 230 - Módulo de gestión de la comunicación serie de la interfaz gráfica.

2.2.- CONTROL DE LA VELOCIDAD DE LOS MOTORES.

El módulo de control de la velocidad de los motores se observa en la siguiente imagen.

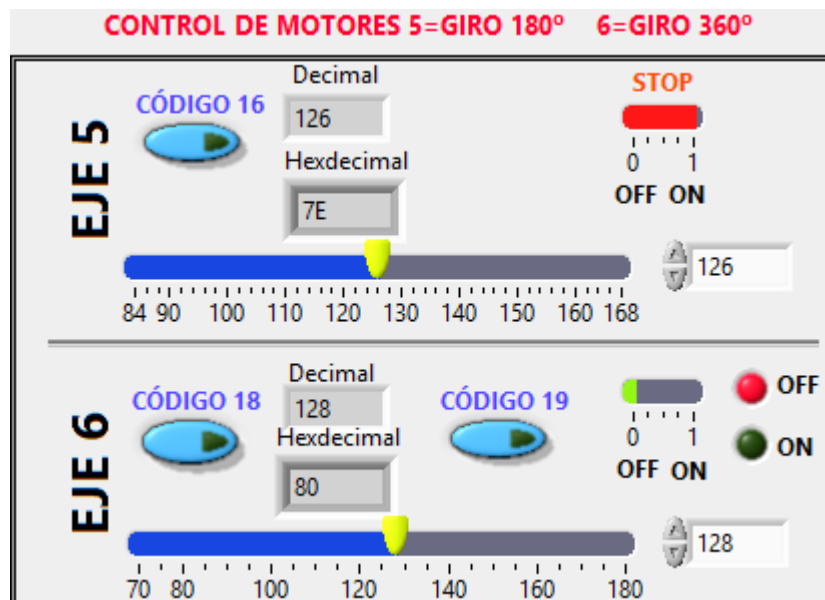


Ilustración 231 - Módulo de control de la velocidad de los motores. Interfaz gráfica.

La parte superior es la encargada de gestionar manualmente el eje 5 que posee un rango de movimiento de 180°. El botón azul con la etiqueta de “código 16” envía al robot los parámetros actualizados de velocidad. El usuario modifica la velocidad con la barra deslizante. En el centro de la barra el eje está parado, hacia la izquierda de mueve en sentido antihorario (desde el lado A convenido, ver Capítulo II Sección I) y hacia la derecha se mueve en sentido horario. La rueda de selección roja y verde permite manipular la velocidad (en la selección de “VARIA” o detener el eje, “DESAC”). Por último, el botón azul de “código 17” envía el valor de la barra verde (0 o 1) a los botones de ON/OFF. Este botón permite detener el eje.

La parte inferior gobierna el movimiento manual del eje 6. El funcionamiento es análogo al anterior exceptuando la ausencia de la rueda de selección. Esto se debe a que para detener el eje 6 sólo es necesario quitar la referencia de tensión (ver sección anterior de este Capítulo).

2.3.- LECTURA Y RECARGA DE REGISTROS CONTADORES.

El módulo de lectura y recarga de los registros contadores de las señales de los encoder se observa en la imagen inferior.

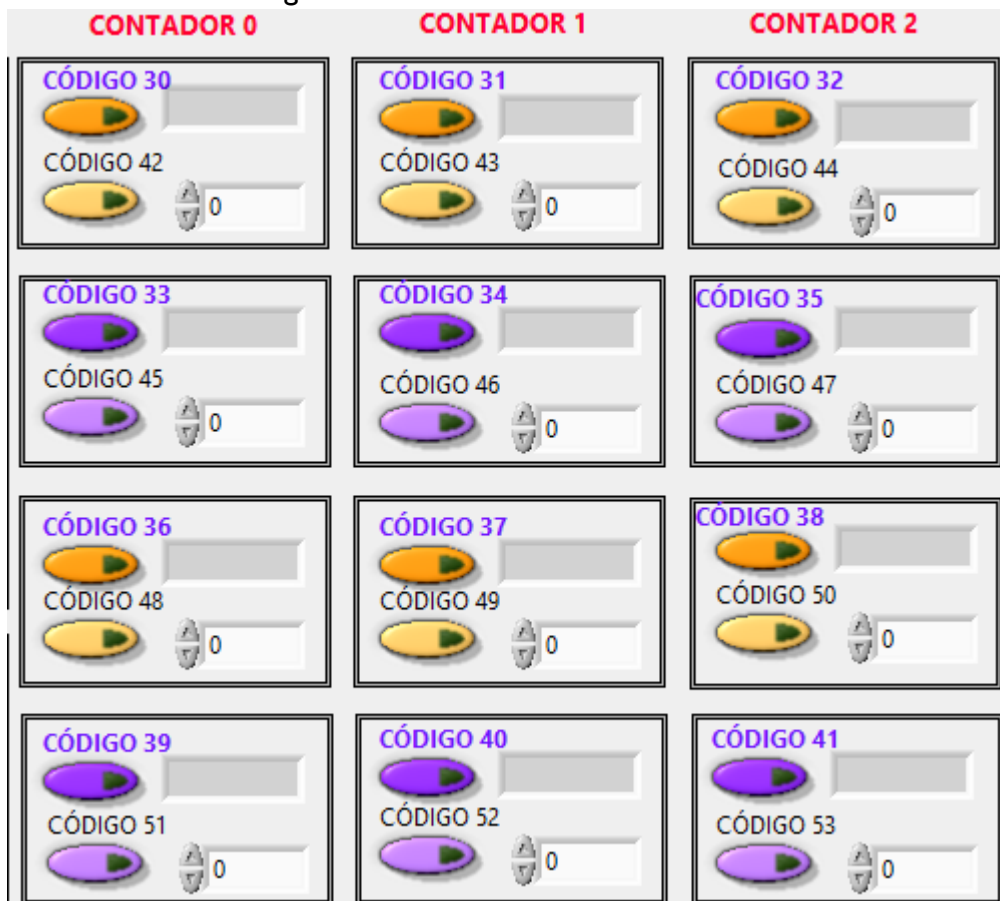


Ilustración 232 - Lectura y recarga de los registros contadores. Interfaz gráfica.

Los botones de color morado oscura permiten la lectura de los tres registros contadores del eje 5 y los tres registros del eje 6 (señales S1, S2 y TOP de los encoder de izquierda a derecha). Los botones de color morado claro permiten la recarga de los contadores al valor introducido por el usuario en la etiqueta numérica ubicada a la derecha del botón. Los botones de color naranja gestionan los contadores encadenados (ver sección anterior de este Capítulo), tanto su lectura (botones naranja oscura) como su recarga (botones naranja claro). Su funcionamiento es completamente análogo a los botones morados.

2.4.- BOTONES DE ACCIONES ESPECÍFICAS.

El módulo de los botones que realizan acciones específicas se observa en la imagen inferior. Está conformado por los botones y una pantalla informativa en la parte superior que informa al usuario de las acciones que se están realizando en cada momento.

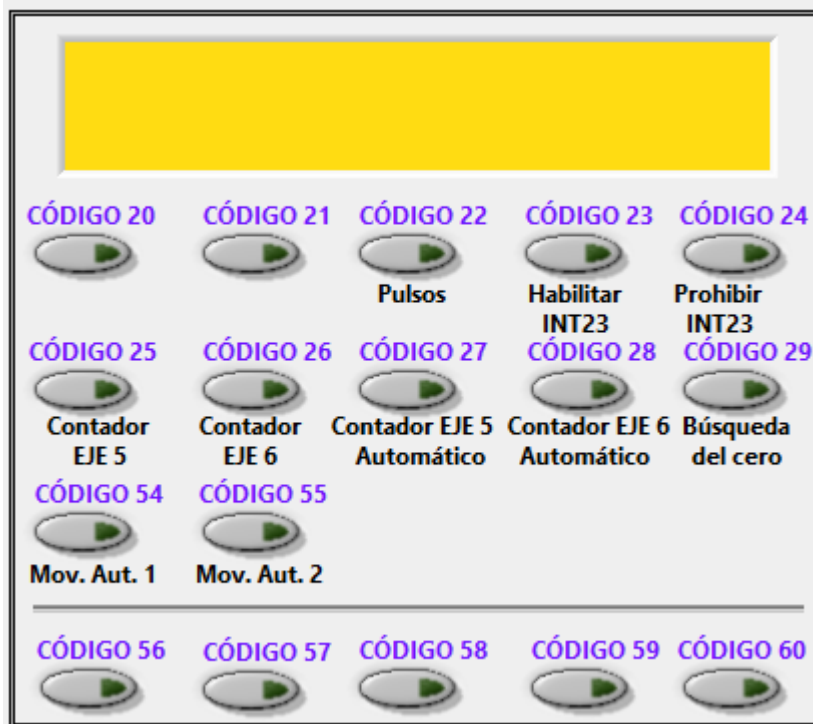


Ilustración 233 - Módulo de botones de acciones específicas. Interfaz gráfica.

Los botones que conforman este módulo realizan el envío de comandos específicos preprogramados dentro de la memoria de código del microcontrolador. En la pantalla amarilla se muestran mensajes informativos al usuario del proceso que se está realizando en cada momento. En la tabla inferior se resumen las acciones que realiza cada uno.

CÓDIGO DEL BOTÓN	DESCRIPCIÓN	ACCIÓN
20	-	Envía "Código 20" a la pantalla. Al enviarse este código se chequea que se tenga una correcta comunicación con el microcontrolador.
21	-	Envía "Código 21" a la pantalla. Al enviarse este código se chequea que se tenga una correcta comunicación con el microcontrolador.
22	Pulsos	Cada vez que se presiona el botón se envía un pulso manual a la entrada de los contadores principales. De esta forma puede verificarse que los contadores cuenten correctamente antes de ejecutar un conteo de pulsos del encoder.
23	HabilitaINT23	Este botón habilita manualmente las interrupciones 2 y 3 correspondientes a las señales de los

		sensores inductivos.
24	ProhibeINT23	Este botón prohíbe manualmente las interrupciones 2 y 3 correspondientes a las señales de los sensores inductivos.
25	Contador 5	Este botón habilita el contaje de pulsos del encoder del eje 6.
26	Contador 6	Este botón habilita el contaje de pulsos del encoder del eje 6.
27	Contador 5 Automático	Este botón ejecuta una rutina que cuenta los pulsos de un movimiento de 180° del eje 5. Para ejecutar la rutina correctamente es necesario llevar el eje manualmente en sentido horario (visto desde el lado convenido A) hasta la posición máxima detectada por el sensor final de carrera.
28	Contador 6 Automático	Este botón ejecuta una rutina que cuenta los pulsos de un giro de 360° del eje 6 utilizando las señales del sensor inductivo de este eje.
29	Búsqueda del cero	Este botón ejecuta la rutina de búsqueda del cero. Para que la muñeca realice correctamente esta rutina debe de llevarse manualmente el eje 5 una posición de menos de 90° con la mesa.
54	Mov. Aut.1	Este botón ejecuta una rutina de movimiento automático que involucra a ambos ejes. Antes de ejecutarla se debe llevar el eje 5 a una posición de menos de 90° con la mesa.
55	Mov. Aut.2	Este botón ejecuta una rutina de movimiento automático que involucra a ambos ejes. Antes de ejecutarla se debe llevar el eje 5 a una posición de menos de 90° con la mesa.

Los botones con códigos 56 al 60 no tienen ninguna acción preprogramada. Sin embargo, están preparados para enviar sus respectivos códigos al microcontrolador y realizar las acciones que se programen dentro de la memoria de código.

3.- PROGRAMACIÓN EN LABVIEW.

Como ya se mencionó anteriormente, el software escogido para programar la interfaz gráfica de control de la muñeca del robot fue Labview. Labview es un software creado por la empresa National Instruments que permite programar aplicaciones a medida a través de un lenguaje de programación gráfico. La programación gráfica permite una asociación directa entre las estructuras mentales del programador y el código a programar. Además, Labview ofrece una tremenda variedad de herramientas gráficas que contribuyen a la creación de

interfaces amoldadas a las necesidades de cada aplicación y de apariencia amigable con el usuario. En la imagen inferior se observa un ejemplo de programación gráfica en Labview.

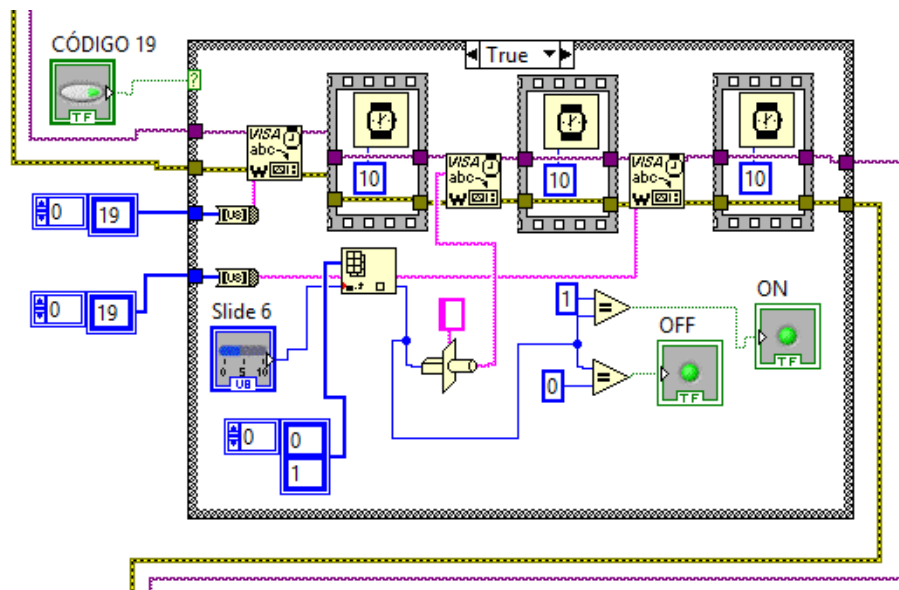


Ilustración 234 - Ejemplo de programación gráfica en Labview.

Dentro de la programación general de la interfaz se destaca la programación del envío y recepción de datos a través de la comunicación serie entre el ordenador y el microcontrolador, base de la interconexión entre los subsistemas de control diseñados. A su vez, se destaca el sistema de control de la pantalla informativa. A continuación se profundiza en la programación de cada uno de estos procesos.

3.1.- PROGRAMACIÓN EN LABVIEW: ENVÍO Y RECEPCIÓN DE DATOS MEDIANTE COMUNICACIÓN SERIE.

La comunicación serie es el proceso de envío de datos en el que se envía un bit a la vez y de forma secuencial por un canal de comunicación. En el marco del sistema de control diseñado la comunicación serie juega un papel fundamental al ser el puente de unión entre la interfaz gráfica y la implementación física de los subsistemas de control diseñados. Para programar la comunicación serie entre ordenador y microcontrolador se utilizaron numerosas funciones de Labview que a continuación se enumeran y explican.

3.1.1.- FUNCIONES UTILIZADAS.

- **Ciclo "While"**: función que ejecuta todas las acciones incluidas en su interior mientras se cumpla la condición que gobierna su funcionamiento (en este caso una contante booleana). Esta función engloba todo el programa para que se ejecute continua y cíclicamente siempre que el usuario así lo desee.

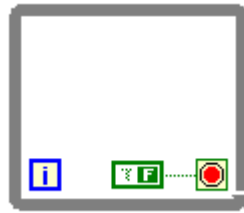


Ilustración 235 - Función ciclo "While" de Labview.

- **Estructura "Case":** esta función ejecuta diversas rutinas según la entrada que reciba. Es la implantación gráfica de la estructura de programación "if".

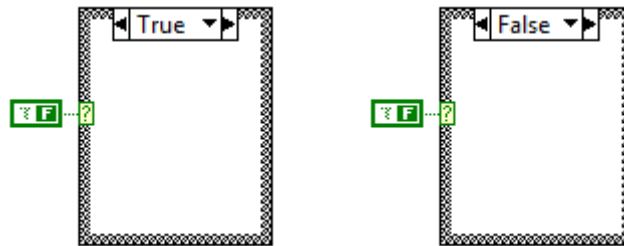


Ilustración 236 - Estructura "case" de Labview.

- **Función "type cast":** esta función transforma un tipo de dato en otro siguiendo un patrón introducido por el programador.



Ilustración 237 - Función "type cast" de Labview.

- **Funciones "byte array to string" y "string to byte array":** la primera convierte una serie de bits que representan caracteres ASCII en un tipo de dato string (letra), la segunda realiza la conversión inversa.



Ilustración 238 - Función "byte array to string" de Labview.



Ilustración 239 - Función "string to byte array" de Labview.

- **Función "array subset":** esta función permite obtener una porción de una matriz. En índice se indica la columna, fila o posición de columna – fila de inicio. En longitud se indica cuántos caracteres de la matriz original deben de incluirse en la matriz resultado.

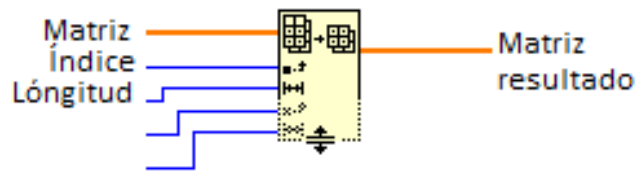


Ilustración 240 - Función "array subset" de Labview.

- Función "wait (ms)": esta función retarda cualquier acción a la que afecte. El usuario introduce el tiempo de retardo que desea en milisegundos. Esta función es de vital importancia pues permite sincronizar los tiempos de los procesos de Labview con los de la comunicación serie. De esta forma se logra una interacción correcta entre la interfaz gráfica y el sistema de control.

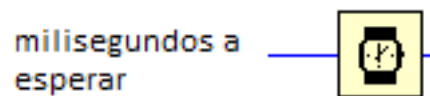


Ilustración 241 - Función "wait (ms)" de Labview.

A continuación se detallan las funciones envueltas directamente en la programación de la comunicación serie.

- **Función "VISA configure serial port"**: esta función configura la comunicación serie a través del puerto indicado en la entrada "visa resource name" (VISA son las siglas en inglés de "arquitectura de software de instrumento virtual"). También configura los parámetros de la configuración de la comunicación serie explicados en el módulo correspondiente en la sección anterior de este Capítulo.

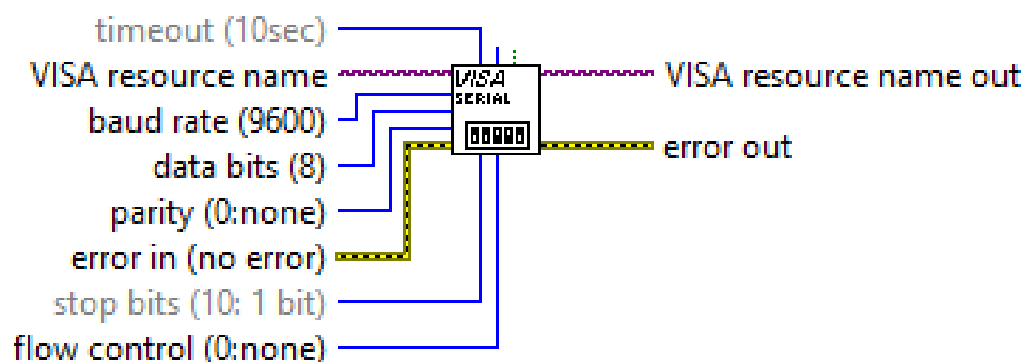


Ilustración 242 - Función "VISA configure serial port" de Labview.

- **Función "VISA write"**: escribe la información ubicada en el buffer de escritura ("write buffer") en el puerto especificado mediante la función "VISA configure serial port".

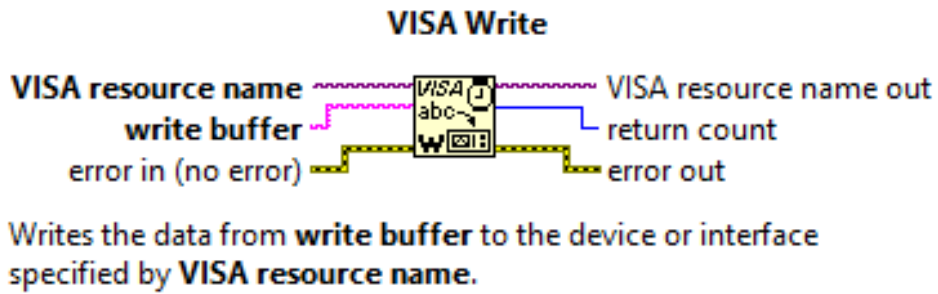


Ilustración 243 - Función "VISA write" de Labview.

- **Función "VISA read"**: esta función lee del puerto especificado mediante la función "VISA serial port configuration" y devuelve la información en el buffer indicado por el usuario en "read buffer".

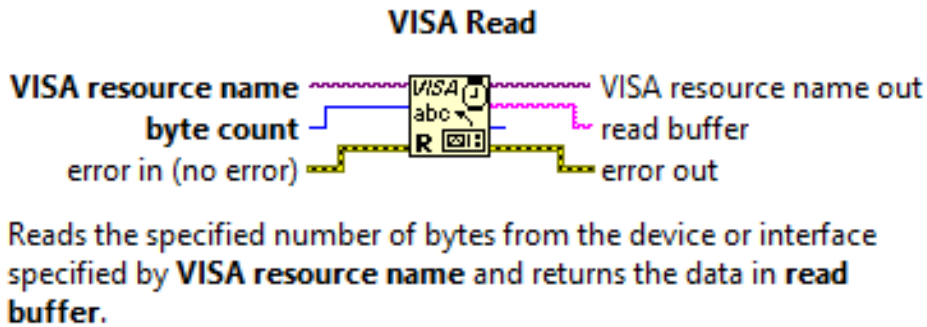


Ilustración 244 - Función "VISA read" de Labview.

- **Función tipo "property node" "bytes at port"**: esta función se conecta a la entrada "byte count" (contaje de bits) de la función VISA read y le informa de cuántos bits a leer se encuentran en el buffer. De esta forma se asegura una lectura correcta de los datos.

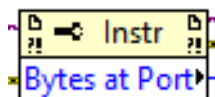


Ilustración 245 - Función "bytes at port" de Labview.

- **Función "VISA Close"**: esta función cierra la sesión del dispositivo o el evento de objeto especificad en la entrada "VISA resource name".

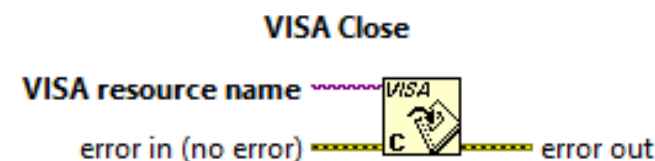


Tabla 23 - Función "VISA Close" de Labview.

3.1.2.- PROGRAMACIÓN DE UNA ESTRUCTURA COMPLETA DE LECTURA Y ESCRITURA.

El envío de comandos al microcontrolador se programa siempre como un proceso de escritura en el buffer de datos. La recepción de información del sistema enviada por el microcontrolador se programa como una lectura de datos del buffer. En la imagen inferior se observa una estructura de programación gráfica del programa en la que se realiza tanto operaciones de escritura como de lectura.

En esta estructura de programación el control booleano “código 32” gobierna la ejecución de la rutina contenida dentro de la opción de “True” (verdadero) de la estructura “case”. Esta rutina realiza una escritura triple en el buffer de datos con tiempos intermedios entre procesos de escritura. La escritura triple es la base del sistema de comunicación serie diseñado. De esta forma, el microcontrolador recibe tres datos que le informan de la dirección de la memoria de código a la cual debe dirigirse y de parámetros a utilizar dentro de la rutina a ejecutar. Una vez finalizado el proceso de escritura se esperan veinte milisegundos a que el microcontrolador responda. Al transcurrir el tiempo de espera se realiza una operación de lectura del buffer de datos para recolectar la información enviada por el microcontrolador en respuesta a la rutina ejecutada. La información se muestra visible en una etiqueta en la interfaz de usuario (cuyo nombre es “string 30”).

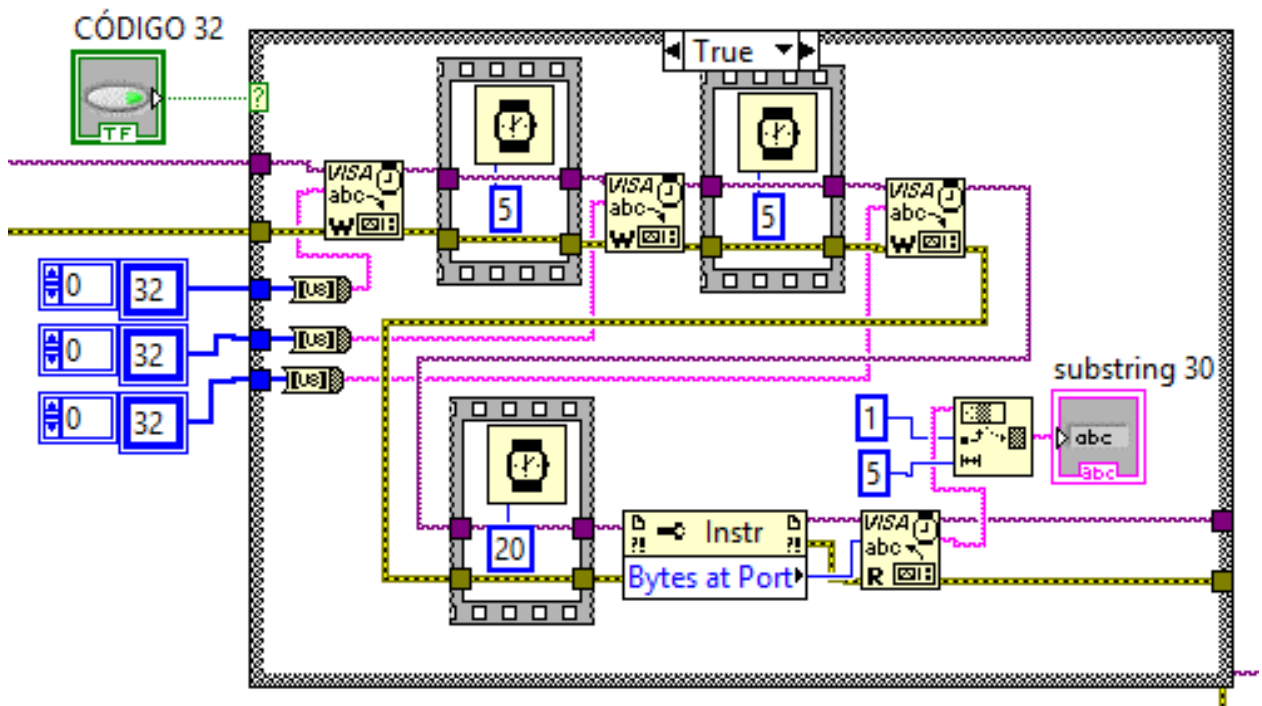


Ilustración 246 - Ejemplo de programación de una estructura completa de lectura y escritura en el buffer de datos.

3.2.- PROGRAMACIÓN EN LABVIEW: ESTRUCTURA DE CONTROL DE LA PANTALLA INFORMATIVA.

La pantalla informativa es de color amarillo y se encuentra ubicada en la parte inferior izquierda de la interfaz gráfica. Su objetivo es informar al usuario de la actividad que se está realizando en cada momento.

La programación de los mensajes que aparecen en la pantalla se realizó pensando en que la información ofrecida por el usuario debe ser claro reflejo del estado del proceso de comunicación con el microcontrolador. Se evitó a toda costa incurrir en procesos que pudiesen llevar a informar falsamente al usuario.

La programación del sistema inicia con una lectura de un código enviado por el microcontrolador que identifica el proceso que se está realizando en cada momento. El proceso de escritura en la pantalla solo se realiza en el caso de que el envío de este código sea correcto. Así se asegura que la información que recibe el usuario acerca del proceso es verídica. El código enviado por el microcontrolador es leído del buffer de datos y gobierna una estructura "case" que rellena la pantalla con el mensaje programado para cada código. En la imagen inferior se observa la programación descrita. En este caso los tres códigos enviados son 32 en decimal.

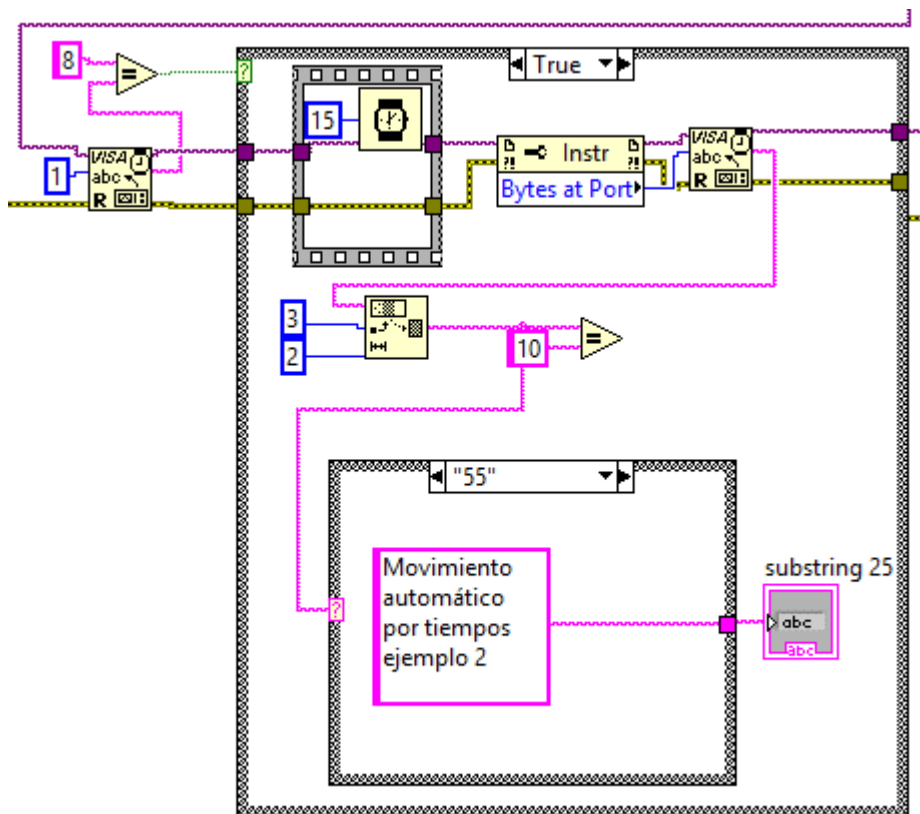


Ilustración 247 - Programación de la estructura de control de la pantalla informativa.

4.- MAX 232

El MAX232 es un circuito integrado que convierte las señales de un puerto serie RS-232 a señales compatibles con los niveles TTL de circuitos lógicos. El MAX232 sirve como interfaz de transmisión y recepción para las señales RXD y TXD de la comunicación serie.

El interface estándar RS-232 está definida como la interface entre un equipo terminal de datos (DTE) y un equipo de comunicación de datos (DCTE) empleando un intercambio en modo serie de datos binarios. La velocidad máxima de transmisión de datos es de 19200 bps y la capacidad máxima de la línea es de 2500 pF, equivalente a una longitud máxima de entre 15 y 20 metros de cable. En esta interfaz de comunicación se utiliza una línea de transmisión y otra para recepción referidas ambas a una misma línea de masa. Las señales son transportadas como una simple señal de tensión referida a una tierra común. Estas características eléctricas derivan en un alto índice de ruido en la transmisión. En cuanto a las características mecánicas se utiliza un conector de nueve pines. Los niveles lógicos de tensión se detallan a continuación.

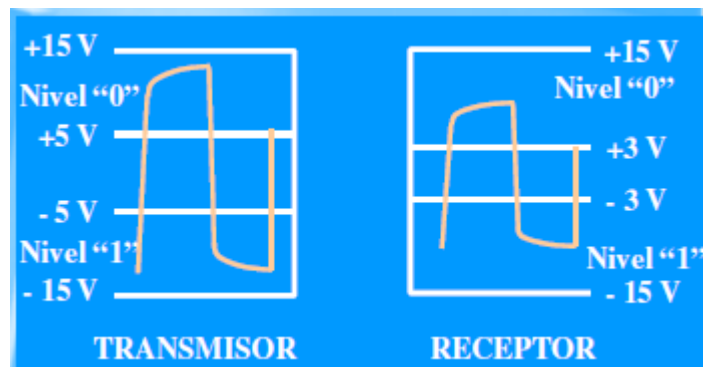


Ilustración 248 - Niveles lógicos de tensión de la interface RS - 232.

En la imagen inferior se observan los pines de un conector serie estándar y su significado.

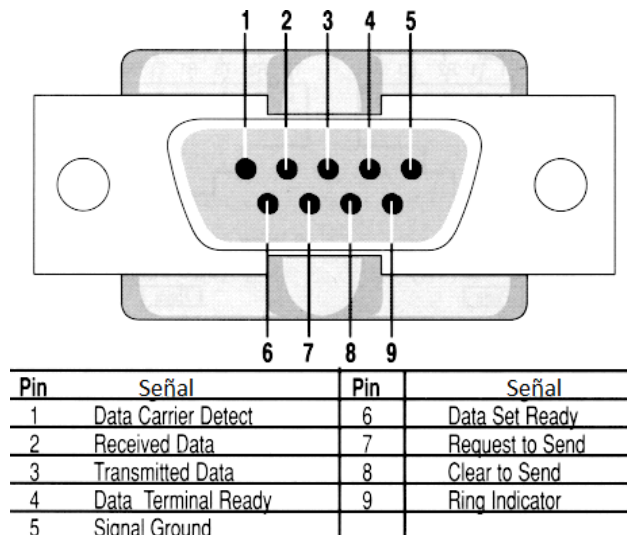


Ilustración 249 - Pines del conector DB9 de la comunicación serie.

En la siguiente tabla se detalla el funcionamiento de cada una de las señales de los nueve pines. Los pines utilizados por el MAX 232 se encuentran en negrita.

Pin	Nombre	Descripción	Función
1	DCD	Data Carrier Detect	Detección portadora módem DTE←DCTE
2	RXD	Receive Data	Línea de recepción de datos DTE (recibe)←DCTE
3	TXD	Transmit Data	Línea de transmisión de datos DTE (transmite) →DCTE
4	DTR	Data terminal Ready	Terminal de datos listo DTE→DCTE
5	GND o SG	System Ground	Masa (línea común de referencia)
6	DSR	Data Set Ready	Equipo del circuito de datos (módem) listo DTE←DCTE
7	RTS	Request to Send	Solicitud de envío DTE→DCTE
8	CTS	Clear to Send	Libre para el envío DTE (habilitación de transmisor) ←DCTE
9	RI	Ring Indicator	Detección de llamada telefónica DTE←DCTE

Tabla 24 - Interface RS-232 características funcionales.

Las entradas de recepción de RS-232 (las cuales pueden llegar a ± 15 V), se convierten al nivel estándar de 5V de la lógica TTL (del inglés “transistor - transistor logic”). Estos receptores tienen un umbral típico de 1,3 V y una histéresis de 0,5 V. En la imagen inferior se observan los niveles de tensión TTL.

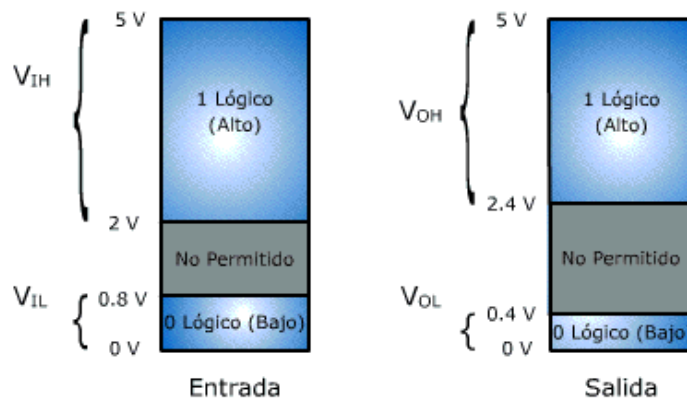


Ilustración 250 - Niveles lógicos de tensión de la tecnología TTL.

El circuito integrado tiene salidas para manejar niveles de tensión del RS-232 (aproximadamente ± 7.5 V) producidos a partir de una tensión de alimentación de + 5V utilizando multiplicadores internos de tensión con la adición de condensadores externos. Esto es de mucha utilidad para la implementación de puertos serie RS-232 en dispositivos que tengan una alimentación simple de + 5V, como lo son los microcontroladores. El circuito interno del MAX 232 se observa en la imagen inferior. Posee alimentación proveniente del puerto serie y una alimentación a +5V independiente.

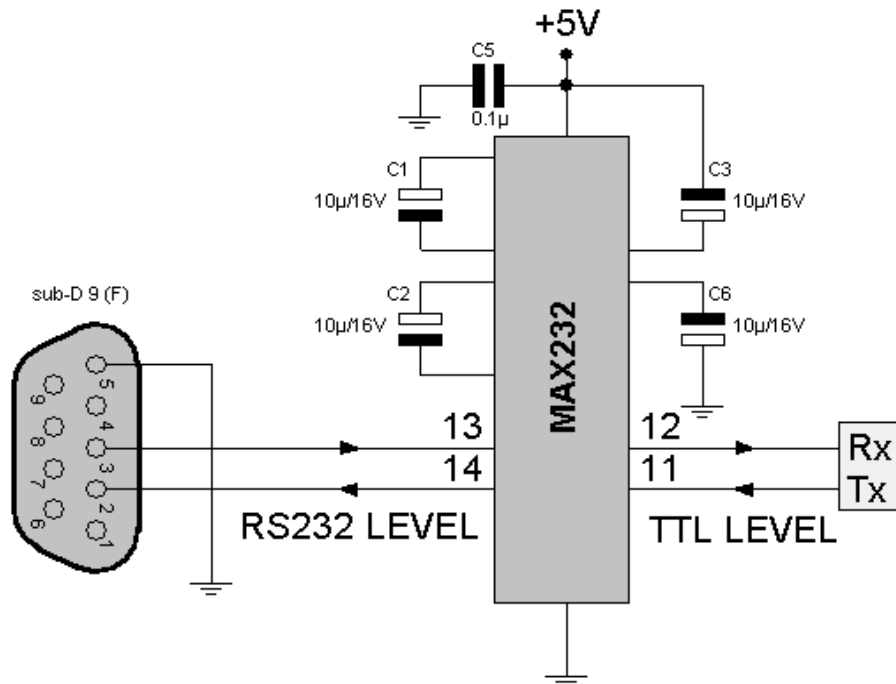


Ilustración 251 - Conexión del circuito MAX 232 con el puerto serie del ordenador.

El tipo de conexión realizada se conoce como XON - XOFF e involucra únicamente tres hilos, como se observa en la imagen inferior.

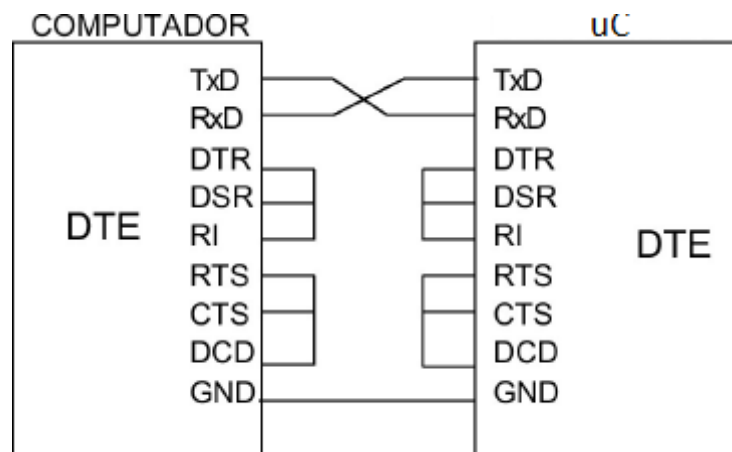


Ilustración 252 - Conexión tipo XON - XOFF entre el ordenador y el microcontrolador.

La línea RX se encuentra implementada de fábrica en el pin 26 correspondiente al bit 7 del puerto C (RC7) del microcontrolador PIC16F877A. La línea TX se encuentra en el pin 25 correspondiente al bit 6 del puerto C (RC6). En la imagen inferior se observa el circuito real.

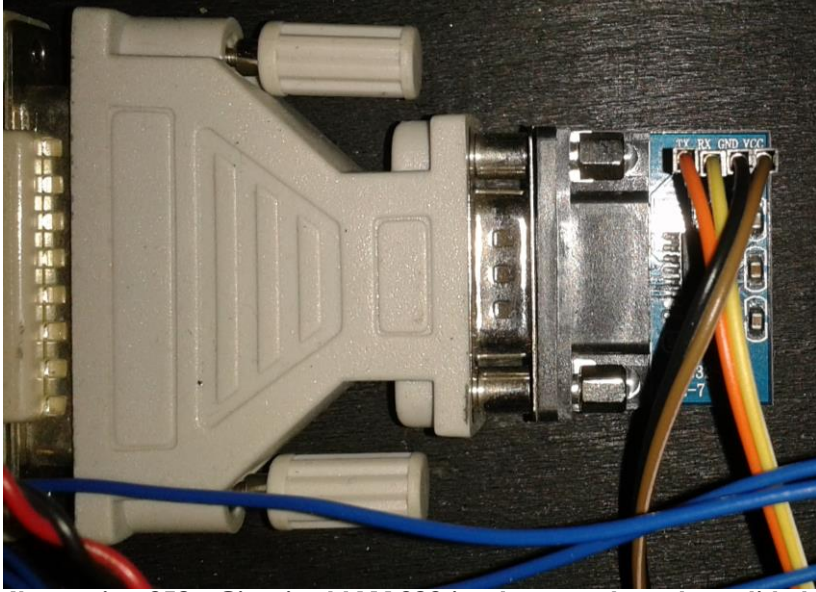


Ilustración 253 - Circuito MAX 232 implementado en la realidad.

Capítulo VI

Programación en ensamblador

```

;*****
; Microcontrolador: 16F877A
; Frecuencia:      20MHz
; Nombre: Versión Final
;
;
; Fecha: 19-06-2015
; Autor: Valeria Ospina y José Torreblanca
;
;
;*****
LIST                P=16F877A
INCLUDE             <P16F877A.inc>
__CONFIG__CP_OFF & __WDT_OFF & __BODEN_OFF & __PWRTE_OFF & __HS_OSC &
__WRT_OFF & __LVP_OFF & __CPD_OFF
;***** REGISTROS *****
;
CONTA20             equ            0x20            ;
CONTA21             equ            0x21            ;
CONTA22             equ            0x22            ;
CONTA23             equ            0x23            ;
CONTA24             equ            0x24            ;
CONTA25             equ            0x25            ;
CONTA26             equ            0x26            ;
CONTA27             equ            0x27            ;
;
CONTAE0             equ            0x28            ;Contador para escribir el
0
CONTAE1             equ            0x29            ;Contador para escribir el
1
CONTAE2             equ            0x2A            ;Contador para escribir el
2
CONTALO             equ            0x2B            ;Contador para leer el 0
CONTAL1             equ            0x2C            ;Contador para leer el 1
CONTAL2             equ            0x2D            ;Contador para leer el 2
CONTADORB           equ            0x2E            ;Valor contador bajo
CONTADORA           equ            0x2F            ;Valor contador alto
;
CONTAS11B           equ            0x30            ;Conta 8259 pulsos S1B
MOTOR1
CONTAS11A           equ            0x31            ;Conta 8259 pulsos S1A
CONTAS21B           equ            0x32            ;Conta 8259 pulsos S2B
CONTAS21A           equ            0x33            ;Conta 8259 pulsos S2A
CONTAT1B           equ            0x34            ;Conta 8259 pulsos TOP1B

```

```

CONTAT1A          equ          0x35          ;Conta 8259 pulsos TOP1A
CONTA36           equ          0x36          ;-----
CONTA37           equ          0x37          ;-----
;
CONTAS12B         equ          0x38          ;Conta 8259 pulsos S1B
                                     MOTOR2
CONTAS12A         equ          0x39          ;Conta 8259 pulsos S1A
CONTAS22B         equ          0x3A          ;Conta 8259 pulsos S2B
CONTAS22A         equ          0x3B          ;Conta 8259 pulsos S2A
CONTAT2B          equ          0x3C          ;Conta 8259 pulsos TOP1B
CONTAT2A          equ          0x3D          ;Conta 8259 pulsos TOP1A
IDSERIE           equ          0x3E          ;-----
PULSOS            equ          0x3F          ;-----
;
DATO8259         equ          0x40          ;-----
BINARIO           equ          0x41          ;-----
CONTA42           equ          0x42          ;-----
CONTA43           equ          0x43          ;-----
CONTA44           equ          0x44          ;Contador 3 datos serie
TXDATO1           equ          0x45          ;Primer dato serie
TXDATO2           equ          0x46          ;Segundo dato serie
TXDATO3           equ          0x47          ;Terce dato serie
CONTA48           equ          0x48          ;-----
CONTA49           equ          0x49          ;-----
CONTA4A           equ          0x4A          ;Conta 8253 pulsos S1B
                                     MOTOR1
CONTA4B           equ          0x4B          ;Conta 8253 pulsos S1A
CONTA4C           equ          0x4C          ;Conta 8253 pulsos S2B
CONTA4D           equ          0x4D          ;Conta 8253 pulsos S2A
CONTA4E           equ          0x4E          ;Conta 8253 pulsos TOP1B
CONTA4F           equ          0x4F          ;Conta 8253 pulsos TOP1A
;
CONTA50           equ          0x50          ;Conta 8253 pulsos S1B
                                     MOTOR2
CONTA51           equ          0x51          ;Conta 8253 pulsos S1A
CONTA52           equ          0x52          ;Conta 8253 pulsos S2B
CONTA53           equ          0x53          ;Conta 8253 pulsos S2A
CONTA54           equ          0x54          ;Conta 8253 pulsos TOP1B
CONTA55           equ          0x55          ;Conta 8253 pulsos TOP1A
;
GCONTA01          equ          0x58          ;Guarda Valor contador1
GCONTA02          equ          0x59          ;Guarda Valor contador2
GCONTA03          equ          0x5A          ;Guarda Valor contador3
GPCLATH           equ          0x5B          ;Guarda Valor PCLATH
GSTATUS           equ          0x5C          ;Guarda Valor STATU
ACUMULA           equ          0x5D          ;Guarda Valor Acumulador

```

```

CONTA5E          equ          0x5E          ;-----
CONTA5F          equ          0x5F          ;-----
;
MULTIPLICANDO_B equ          0x60          ;Multiplicando bajo
                                   (ARRIBA)
MULTIPLICANDO_A equ          0x61          ;Multiplicando alto
                                   (ARRIBA)
MULTIPLICADOR_B  equ          0x62          ;Multiplicador bajo
                                   (ABAJO)
MULTIPLICADOR_A  equ          0x63          ;Multiplicador alto
                                   (ABAJO)
RESULTADOP_B     equ          0x64          ;Resultado bajo
RESULTADOP_BM    equ          0x65          ;Resultado bajo medio
RESULTADOP_MA    equ          0x66          ;Resultado medio alto
RESULTADOP_A     equ          0x67          ;Resultado alto
CONTADOR_BITS    equ          0x68          ;Contador bits=16
NUMERADOR_B      equ          0x69          ;Numerador BAJO
                                   (ARRIBA)
NUMERADOR_M      equ          0x6A          ;Numerador MEDIO
                                   (ARRIBA)
NUMERADOR_A      equ          0x6B          ;Numerador ALTO
                                   (ARRIBA)
DENOMINADOR_B    equ          0x6C          ;Denominador BAJO
                                   (ABAJO)
DENOMINADOR_A    equ          0x6D          ;Denominador ALTO
                                   (ABAJO)
RESULTADOD_B     equ          0x6E          ;resultado BAJO
RESULTADOD_M     equ          0x6F          ;resultado MEDIO
RESULTADOD_A     equ          0x70          ;resultado ALTO
RESTODIVISION_B  equ          0x71          ;resto BAJO
RESTODIVISION_A  equ          0x72          ;resto ALTO
UNIDADES         equ          0x73          ;Unidades
DECENAS          equ          0x74          ;Decenas
CENTENAS         equ          0x75          ;Centenas
UNIDADMIL        equ          0x76          ;Unidades de mil
DECENAMIL        equ          0x77          ;Decenas de mil
PROGRAMACION     equ          0x78          ;Comando para la LCD
CARACTER         equ          0x79          ;Caracter parala LCD
CONTA01          equ          0x7A          ;contador retardo
CONTA02          equ          0x7B          ;contador retardo
CONTA03          equ          0x7C          ;contador retardo
;
;-----
#define CS138     PORTC,0          ;cS Decodificador 74LS138
#define A0138     PORTC,4          ;Decodificador 74LS138
#define A1138     PORTC,5          ;Decodificador 74LS138
#define A2138     PORTC,3          ;Decodificador 74LS138
#define ELCD      PORTC,1          ;Línea E de la LCD
    
```

```

#define RSLCD PORTC,2 ;Línea RS de la LCD
#define INT PORTB,0 ;Interrupció
n
#define INTA PORTB,1 ;Leer interrupción
#define DIR245 PORTB,2 ;DIR del 74LS245
#define A1 PORTA,5 ;Línea A1 del 8253 8255
#define A0 PORTE,0 ;Línea A0 del 8259 8253
8255
#define WWRR PORTE,1 ;Línea WR del 8259 8253
8255
#define RRDD PORTE,2 ;Línea RD del 8259 8253
8255
; ;-----
;=====PROGRAMA===== ;-----
ORG 0x00 ;-----
GOTO INICIO ;-----
;=====INTERRUPCIONES===== ;-----
ORG 0x04 ;-----
GOTO RSI ;-----
RSI MOVWF ACUMULA ;guardamos los registros
MOVF STATUS,0 ;-----
MOVWF GSTATUS ;-----
MOVF CONTA01,0 ;W,CONTA01,CONTA02,CO
NTA03
MOVWF GCONTA01 ;PCLATH y STATUS
MOVF CONTA02,0 ;-----
MOVWF GCONTA02 ;-----
MOVF CONTA02,0 ;-----
MOVWF GCONTA02 ;-----
MOVF PCLATH,0 ;-----
MOVWF GPCLATH ;-----
; ;-----
BCF STATUS,RP0 ;BANCO 0
BTFSC PIR1,5 ;Interrupcion puerto serie
GOTO DATORC ;RCIF se pone a 1 al
completarse la recepción
BTFSC INTCON,1 ;Interrupcion RB0/INT
GOTO INTRB0 ;INTF se pone a 1 con nivel
activo en RB0
BTFSC PIR1,0 ;Interrupcion TMR1
GOTO TMR001 ;TMR1IF se pone a al
desbordarse
; BTFSC INTCON,0 ;Interrupcion RBX
; GOTO INTRBX ;-----
; BTFSC INTCON,2 ;Interrupcion TMR0
; GOTO TMR000 ;-----

```

```

;          BTFSC      PIR1,4          ;(TXIF) Comprobamos si la
;          GOTO       DATOTX          ;interrupción ha sido
;                                     ;producida por la
;                                     ;obtención de un dato.Ç
;                                     ;-----
;          MOVF       GCONTA01,0      ;-----
;          MOVWF      CONTA01         ;-----
;          MOVF       GCONTA02,0      ;-----
;          MOVWF      CONTA02         ;-----
;          MOVF       GCONTA03,0      ;-----
;          MOVWF      CONTA03         ;-----
;          MOVF       GPCLATH,0       ;-----
;          MOVWF      PCLATH          ;-----
;          MOVF       GSTATUS,0       ;-----
;          MOVWF      STATUS          ;-----
;          MOVF       ACUMULA,0       ;-----
;                                     ;-----
;          RETFIE                    ;de nuevo y volvemos.
;=====PROGRAMA DE INICIALIZACIÓN=====
INICIO    BSF        STATUS,RP0      ;BANCO 1
          MOVLW     b'10000110'     ;Puerto A en digital
          MOVWF     ADCON1           ;Que entrarán al conversor
          ;                                     ;A/D
          MOVLW     b'00000000'     ;Puerto A en salida
          MOVWF     TRISA            ;-----
          MOVLW     b'00000001'     ;Pin RB0=INT en entrada
          MOVWF     TRISB            ;-----
          MOVLW     b'10000000'     ;Pin RC7=RX en entrada
          MOVWF     TRISC            ;-----
          CLRF      TRISD            ;puerta D en salida
          CLRF      TRISE            ;puerta E en salida
;=====PROGRAMACION INTERRUPTACIONES=====
          MOVLW     b'01000000'     ;Flanco ascendente
          ;                                     ;Interrupción
          MOVWF     OPTION_REG      ;
          MOVLW     b'01010000'     ;Activamos periféricos e
          ;                                     ;INT/RB0
          MOVWF     INTCON           ;-----
          MOVLW     b'00100001'     ;Permiso puerto serie RX y
          ;                                     ;TMR1
          MOVWF     PIE1             ;-----
          BCF       STATUS,RP0      ;BANCO 0
;=====PROGRAMACION TMR1 SERIE 232=====
          MOVLW     b'00110000'     ;TMR1 contando 1:8
          MOVWF     T1CON            ;Asignamos TMR1
          CLRF      TMR1L            ;-----

```

```

                CLRF      TMR1H      ;-----
;=====PROGRAMACION REGISTROS=====
                MOVLW    b'00000011' ;-----
                                     ;Para leer 3 datos puerto
                                     ;serie RX
                MOVWF    CONTA44     ;-----
                CLRF      TMRO       ;-----
;=====PROGRAMACION INICIAL LINEAS=====
                MOVLW    b'00000000' ;-----
                MOVWF    PORTA       ;-----
                MOVWF    PORTB       ;-----
                MOVWF    PORTC       ;-----
                MOVWF    PORTD       ;-----
                MOVWF    PORTE       ;-----
                BCF      CS138        ;Desactivación 74LS138
                BCF      A0138        ;Desactivación 74LS138
                BCF      A1138        ;Desactivación 74LS138
                BCF      A2138        ;Desactivación 74LS138
                BSF      RRDD         ;No lectura
                BSF      WWRR         ;No
                                     ;escritura
                BCF      A0           ;A cero
                BCF      A1           ;A cero
                BSF      INTA         ;No aceptación
                                     ;interrupción
                MOVLW    b'00110100' ;Escribir
                MOVWF    CONTAEO      ;Contador 0 8253
                MOVLW    b'01110100' ;Escribir
                MOVWF    CONTAE1      ;Contador 1 8253
                MOVLW    b'10110100' ;Escribir
                MOVWF    CONTAE2      ;Contador 2 8253
                MOVLW    b'00000000' ;Leer
                MOVWF    CONTALO      ;Contador 0 8253
                MOVLW    b'01000000' ;Leer
                MOVWF    CONTAL1      ;Contador 1 8253
                MOVLW    b'10000000' ;Leer
                MOVWF    CONTAL2      ;Contador 2 8253
                MOVLW    b'00000001' ;Mantiene el relé del
                                     ;motor
                MOVWF    CONTA42      ;del EJE 5 siempre a 1
                MOVLW    b'00000000' ;-----
                MOVWF    CONTA43      ;-----
                MOVLW    b'00000001' ;-----
                MOVWF    CONTA20      ;-----
                CLRF      CONTA50     ;-----
                CLRF      CONTA51     ;-----
                CLRF      CONTA52     ;-----

```

```

        CLRF      CONTA53                ;-----
        MOVLW    b'00110000'           ;-----
        MOVWF    DECENAMIL              ;-----
        MOVWF    UNIDADMIL              ;-----
        MOVWF    CENTENAS               ;-----
        MOVWF    DECENAS                 ;-----
        MOVWF    UNIDADES                ;-----
;=====PANTALLA INICIO=====
MENU00      BSF      PCLATH,3           ;Página 1
            BCF      PCLATH,4           ;
            CALL     RETA0               ;-----
            CALL     PANT1               ;-----
            CALL     TEXTO9              ;-----
            CALL     TEXTO10             ;-----
            BCF      PCLATH,3           ;Página 0
            BCF      PCLATH,4           ;
;=====programación 82557=====
            CALL     PR82557             ;-----
            CALL     PR82558             ;-----
            CALL     PR8259M             ;-----
            CALL     PR8259E             ;-----
            MOVLW    b'111111100'       ;Prohíbo INT del esclavo
            CALL     OCW1M               ;-----
            MOVLW    b'10100000'       ;Rotación interrupciones
            CALL     OCW2M               ;-----
;=====Desactivación de motores=====
            CALL     DESAC                ;-----
;=====Inicialización de contadores=====
            CALL     INIC                 ;-----
;=====PROGRAMACION SERIE RS232=====
            BSF      STATUS,RP0          ;BANCO 1
            MOVLW    b'10000001'       ;81h=Velocidad 20MHz
                                         con modo rápido 9600b.
            MOVWF    SPBRG               ;81h=velocidad 20MHz
                                         con modo lento 2400
                                         baudios
            MOVLW    b'10000100'       ;Seleccionamos 9600
                                         baudios
            MOVWF    TXSTA                ;Programo el registro de
                                         enviar datos
            BCF      STATUS,RP0          ;BANCO 0
            MOVLW    b'10010000'       ;-----
            MOVWF    RCSTA                ;Programo el registro de
                                         recibir datos.
            BSF      STATUS,RP0          ;BANCO 1
            BSF      TXSTA,TXEN          ;Habilito transmisión de

```



```

GOTO LEIDO2 ;-----
BCF T1CON,0 ;Desactivamos TMR1
BCF STATUS,RP0 ;Dato recibido correcto.
MOVF RCREG,W ;Envío el dato recibido al
acumulador.
MOVWF TXDATO3 ;RECC Cargo el dato
recibido en un registro.
BSF INTCON,GIE ;Habilitamos las
interrupciones de nuevo.
MOVLW b'00000011' ;Para leer dos datos
puerto serie TX
MOVWF CONTA44 ;-----
LEIDO1 GOTO LEIDO3 ;-----
BCF STATUS,RP0 ;Dato recibido correcto.
MOVF RCREG,W ;Envío el dato recibido al
acumulador.
MOVWF TXDATO1 ;RECC Cargo el dato
recibido en un registro.
BSF INTCON,GIE ;Habilitamos las
interrupciones de nuevo.
DECFSZ CONTA44,F ;-----
GOTO TMR10 ;-----
LEIDO2 RETURN ;-----
BCF STATUS,RP0 ;Dato recibido correcto.
MOVF RCREG,W ;Envío el dato recibido al
acumulador.
MOVWF TXDATO2 ;RECC Cargo el dato
recibido en un registro.
BSF INTCON,GIE ;Habilitamos las
interrupciones de nuevo.
DECFSZ CONTA44,F ;-----
RETURN ;-----
;=====Temporizador TMR1=52.430,00us =====
TMR10 MOVLW b'00000000' ;Predivisor de 1:8
MOVWF TMR1L ;-----
MOVLW b'10000000' ;-----
MOVWF TMR1H ;Ponemos a cero TMR1
BSF ;Activamos TMR1
T1CON,0
ESPE1 BTFSZ T1CON,0 ;Comprobamos si termina
GOTO ;-----
ESPE1 RETURN ;-----
TMR001 BCF PIR1,TMR1IF ;Habilitamos señalizador
TMR1
BSF INTCON,GIE ;Habilitamos
interrupciones de nuevo.

```

```

                BCF    T1CON,0                ;Desactivamos TMR1
                MOVLW  b'00000011'          ;Para leer dos datos
                                                puerto serie TX
                MOVWF  CONTA44                ;-----
                RETURN                        ;-----
;                                                    ;-----
;=====Volvemos a esperar=====            ;-----
INTRBO         CALL   LE8259                 ;-----
                MOVLW  b'00000000' ;        ;-----
                SUBWF  DATO8259,W ;         ;-----
                BTFSC  STATUS,Z ;          ;-----
                GOTO   INT0                  ;-----
;                                                    ;-----
                MOVLW  b'00000100' ;        ;-----
                SUBWF  DATO8259,W ;         ;-----
                BTFSC  STATUS,Z ;          ;-----
                GOTO   INT1                  ;-----
;                                                    ;-----
                MOVLW  b'00001000' ;        ;-----
                SUBWF  DATO8259,W ;         ;-----
                BTFSC  STATUS,Z ;          ;-----
                GOTO   INT2                  ;-----
;                                                    ;-----
                MOVLW  b'00001100' ;        ;-----
                SUBWF  DATO8259,W ;         ;-----
                BTFSC  STATUS,Z ;          ;-----
                GOTO   INT3                  ;-----
;                                                    ;-----
                BSF    INTCON,7              ;-----
                RETURN                        ;-----
;                                                    ;-----
INTO           CALL   DESAC                  ;-----
                CLRF  PULSOS                 ;-----
                CALL  PULSOA                 ;-----
                CALL  CS557A                 ;-----
                MOVLW  b'01111000'          ;120d=78h Motor 5
                                                velocidad
                MOVWF PORTD                 ;-----
                BCF   A0                     ;-----
                BCF   A1                     ;-----
                BCF   WWRR                   ;-----
                NOP                            ;-----
                BSF   WWRR                   ;-----
                CALL  CS5355D                ;-----
                CALL  CS557A                 ;-----
                MOVLW  b'00000001'          ;motores.

```

```

MOVWF    PORTD                ;-----
BCF      A0                    ;-----
BSF      A1                    ;-----
BCF      WWRR                  ;-----
NOP      ;                     ;-----
BSF      WWRR                  ;-----
CALL     CS5355D               ;-----
BSF      PCLATH,3              ;Página 1
BCF      PCLATH,4              ;-----
CALL     RETA0                  ;-----
CALL     RETA0                  ;-----
CALL     RETA0                  ;-----
BCF      PCLATH,3              ;Página 0
BCF      PCLATH,4              ;-----
CALL     DESAC                  ;-----
MOVLW   b'00110001'           ;Código 12
MOVWF   DECENAS                ;-----
MOVLW   b'00110010'           ;-----
MOVWF   UNIDADES              ;-----
CALL    EOI                    ;-----
BSF     INTCON,7               ;-----
BSF     PCLATH,3               ;Página 1
BCF     PCLATH,4               ;-----
GOTO    DATTX1                 ;-----
;                                     ;-----
INT1    CALL    DESAC           ;-----
        CLRF   PULSOS          ;-----
        CALL   PULSOA          ;-----
        CALL   CS557A          ;-----
        MOVLW b'10000100'      ;132d=84h Motor 5
                                     velocidad
MOVWF   PORTD                ;-----
BCF     A0                    ;-----
BCF     A1                    ;-----
BCF     WWRR                  ;-----
NOP     ;                     ;-----
BSF     WWRR                  ;-----
CALL    CS5355D               ;-----
BSF     PCLATH,3              ;Página 1
BCF     PCLATH,4              ;-----
CALL    RETA0                  ;-----
CALL    RETA0                  ;-----
BCF     PCLATH,3              ;Página 0
BCF     PCLATH,4              ;-----
CALL    DESAC                  ;-----
MOVLW   b'00110001'           ;Código 13

```

```

MOVWF DECENAS ;-----
MOVLW b'00110011' ;-----
MOVWF UNIDADES ;-----
CALL EOI ;-----
BSF INTCON,7 ;-----
BSF PCLATH,3 ;Página 1
BCF PCLATH,4 ;-----
GOTO DATTX1 ;-----
;
INT2 CALL DESAC ;-----
MOVLW b'11110100' ;Prohíbo INT del esclavo
CALL OCW1M ;-----
CALL CS557A ;-----
MOVLW b'10010101' ;150d=96h Motor 6
; velocidad
MOVWF PORTD ;-----
BSF A0 ;-----
BCF A1 ;-----
BCF WWRR ;-----
NOP ;-----
BSF WWRR ;-----
CALL CS5355D ;-----
CALL CS557A ;-----
MOVLW b'00000011' ;motores.
MOVWF PORTD ;-----
BCF A0 ;-----
BSF A1 ;-----
BCF WWRR ;-----
NOP ;-----
BSF WWRR ;-----
CALL CS5355D ;-----
CALL EOI ;-----
BSF INTCON,7 ;-----
MOVLW b'00110001' ;Código 14
MOVWF DECENAS ;-----
MOVLW b'00110100' ;-----
MOVWF UNIDADES ;-----
BSF PCLATH,3 ;Página 1
BCF PCLATH,4 ;-----
GOTO DATTX1 ;-----
;
INT3 CALL DESAC ;-----
CLRF PULSOS ;-----
CALL PULSOA ;-----
MOVLW b'11111100' ;Prohibo INT del esclavo
CALL OCW1M ;-----

```

```

CALL      EOI                      ;-----
BSF       INTCON,7                 ;-----
MOVLW    b'00110001'              ;Código 15
MOVWF    DECENAS                   ;
MOVLW    b'00110101'              ;-----
MOVWF    UNIDADES                  ;-----
BSF      PCLATH,3                  ;Página 1
BCF      PCLATH,4                  ;-----
GOTO     DATTX1                    ;-----
;                                     ;-----
;-----
LEIDO3    BSF      PCLATH,3         ;Página 1
          BCF      PCLATH,4         ;-----
          CALL     PRUEBA           ;-----
;                                     ;-----
          MOVLW    b'00010000'      ;Dato
          ;                                     ;
          SUBWF    TXDATO1,W        ;-----
          BTFSC   STATUS,Z          ;-----
          GOTO     CONV16           ;-----
;                                     ;-----
          MOVLW    b'00010001'      ;Dato
          ;                                     ;
          SUBWF    TXDATO1,W        ;-----
          BTFSC   STATUS,Z          ;-----
          GOTO     CONV17           ;-----
;                                     ;-----
          MOVLW    b'00010010'      ;Dato
          ;                                     ;
          SUBWF    TXDATO1,W        ;-----
          BTFSC   STATUS,Z          ;-----
          GOTO     CONV18           ;-----
;                                     ;-----
          MOVLW    b'00010011'      ;Dato
          ;                                     ;
          SUBWF    TXDATO1,W        ;-----
          BTFSC   STATUS,Z          ;-----
          GOTO     CONV19           ;-----
;                                     ;-----
          MOVLW    b'00010100'      ;Dato
          ;                                     ;
          SUBWF    TXDATO1,W        ;-----
          BTFSC   STATUS,Z          ;-----
          GOTO     CONVE20          ;-----
;                                     ;-----
          MOVLW    b'00010101'      ;Dato

```

```

;
SUBWF    TXDATO1,W    ;----- 21d=15h
BTFSC   STATUS,Z     ;-----
GOTO    CONVE21      ;-----
;
MOVLW   b'00010110'  ;Dato
;----- 22d=16h
SUBWF    TXDATO1,W    ;-----
BTFSC   STATUS,Z     ;-----
GOTO    CONVE22      ;-----
;
MOVLW   b'00010111'  ;Dato
;----- 23d=17h
SUBWF    TXDATO1,W    ;-----
BTFSC   STATUS,Z     ;-----
GOTO    CONVE23      ;-----
;
MOVLW   b'00011000'  ;Dato
;----- 24d=18h
SUBWF    TXDATO1,W    ;-----
BTFSC   STATUS,Z     ;-----
GOTO    CONVE24      ;-----
;
MOVLW   b'00011001'  ;Dato
;----- 25d=19h
SUBWF    TXDATO1,W    ;-----
BTFSC   STATUS,Z     ;-----
GOTO    CONVE25      ;-----
;
MOVLW   b'00011010'  ;Dato
;----- 26d=1Ah
SUBWF    TXDATO1,W    ;-----
BTFSC   STATUS,Z     ;-----
GOTO    CONVE26      ;-----
;
MOVLW   b'00011011'  ;Dato
;----- 27d=1Bh
SUBWF    TXDATO1,W    ;-----
BTFSC   STATUS,Z     ;-----
GOTO    CONVE27      ;-----
;
MOVLW   b'00011100'  ;Dato
;----- 28d=1Ch
SUBWF    TXDATO1,W    ;-----
BTFSC   STATUS,Z     ;-----
GOTO    CONV28       ;-----

```

```

;
MOV LW    b'00011101'    ;-----
SUB WF    TXDATO1,W      ;Dato 29d=1Dh
BT FSC    STATUS,Z       ;-----
GO TO     CONV29         ;-----
;
MOV LW    b'00011110'    ;-----
SUB WF    TXDATO1,W      ;Dato
BT FSC    STATUS,Z       ;30d=1Eh
GO TO     CONV30         ;8253-1
;
MOV LW    b'00011111'    ;-----
SUB WF    TXDATO1,W      ;Lectura contador 0
BT FSC    STATUS,Z       ;-----
GO TO     CONV30         ;-----
;
MOV LW    b'00011111'    ;Dato
SUB WF    TXDATO1,W      ;31d=1Fh
BT FSC    STATUS,Z       ;8253-1
GO TO     CONV31         ;Lectura contador 1
;
MOV LW    b'00100000'    ;-----
SUB WF    TXDATO1,W      ;Dato
BT FSC    STATUS,Z       ;32d=20h
GO TO     CONV32         ;8253-1
;
MOV LW    b'00100001'    ;-----
SUB WF    TXDATO1,W      ;Lectura contador 2
BT FSC    STATUS,Z       ;-----
GO TO     CONV32         ;-----
;
MOV LW    b'00100001'    ;Dato
SUB WF    TXDATO1,W      ;33d=21h
BT FSC    STATUS,Z       ;8253-2
GO TO     CONV33         ;Lectura contador 0
;
MOV LW    b'00100010'    ;-----
SUB WF    TXDATO1,W      ;Dato
BT FSC    STATUS,Z       ;34d=22h
GO TO     CONV34         ;8253-2
;
MOV LW    b'00100011'    ;-----
SUB WF    TXDATO1,W      ;Lectura contador 1
BT FSC    STATUS,Z       ;-----
GO TO     CONV34         ;-----
;
MOV LW    b'00100011'    ;Dato
SUB WF    TXDATO1,W      ;35d=23h
BT FSC    STATUS,Z       ;8253-2
GO TO     CONV35         ;Lectura contador 2
;
MOV LW    b'00100100'    ;-----
SUB WF    TXDATO1,W      ;Dato
BT FSC    STATUS,Z       ;36d=24h
GO TO     CONV35         ;8253-3
;
MOV LW    b'00100100'    ;-----
SUB WF    TXDATO1,W      ;Lectura contador 0
BT FSC    STATUS,Z       ;-----

```



```

SUBWF    TXDATO1,W           ;8253-1
BTFS    STATUS,Z           ;Recarga contador 2
GOTO     CONV44             ;-----
;
;
MOVLW    b'00101101'        ;Dato 45d=2Dh
SUBWF    TXDATO1,W           ;8253-2
BTFS    STATUS,Z           ;Recarga contador 0
GOTO     CONV45             ;-----
;
;
MOVLW    b'00101110'        ;Dato
;46d=2Eh
SUBWF    TXDATO1,W           ;8253-2
BTFS    STATUS,Z           ;Recarga contador 1
GOTO     CONV46             ;-----
;
;
MOVLW    b'00101111'        ;Dato
;47d=2Fh
SUBWF    TXDATO1,W           ;8253-2
BTFS    STATUS,Z           ;Recarga contador 2
GOTO     CONV47             ;-----
;
;
MOVLW    b'00110000'        ;Dato
;48d=30h
SUBWF    TXDATO1,W           ;8253-3
BTFS    STATUS,Z           ;Recarga contador 0
GOTO     CONV48             ;-----
;
;
MOVLW    b'00110001'        ;Dato
;49d=31h
SUBWF    TXDATO1,W           ;8253-3
BTFS    STATUS,Z           ;Recarga contador 1
GOTO     CONV49             ;-----
;
;
MOVLW    b'00110010'        ;Dato
;50d=32h
SUBWF    TXDATO1,W           ;8253-3
BTFS    STATUS,Z           ;Recarga contador 2
GOTO     CONV50             ;-----
;
;
MOVLW    b'00110011'        ;Dato
;51d=33h
SUBWF    TXDATO1,W           ;8253-4
BTFS    STATUS,Z           ;Recarga contador 0
GOTO     CONV51             ;-----
;
;
MOVLW    b'00110100'        ;Dato

```

```

;
SUBWF    TXDATO1,W           52d=34h
BTFSC   STATUS,Z           ;8253-4
GOTO    CONV52             ;Recarga contador 1
;-----
;-----
MOVLW   b'00110101'        ;Dato
;-----
SUBWF    TXDATO1,W           53d=35h
BTFSC   STATUS,Z           ;8253-4
GOTO    CONV53             ;Recarga contador 2
;-----
;-----
MOVLW   b'00110110'        ;Dato
;-----
SUBWF    TXDATO1,W           54d=36h
BTFSC   STATUS,Z           ;-----
GOTO    CONV54             ;-----
;-----
;-----
MOVLW   b'00110111'        ;Dato
;-----
SUBWF    TXDATO1,W           55d=37h
BTFSC   STATUS,Z           ;-----
GOTO    CONV55             ;-----
;-----
;-----
MOVLW   b'00111000'        ;Dato
;-----
SUBWF    TXDATO1,W           56d=38h
BTFSC   STATUS,Z           ;-----
GOTO    CONVE56            ;-----
;-----
;-----
MOVLW   b'00111001'        ;Dato
;-----
SUBWF    TXDATO1,W           57d=39h
BTFSC   STATUS,Z           ;-----
GOTO    CONVE57            ;-----
;-----
;-----
MOVLW   b'001111010'      ;Dato
;-----
SUBWF    TXDATO1,W           58d=3Ah
BTFSC   STATUS,Z           ;-----
GOTO    CONVE58            ;-----
;-----
;-----
MOVLW   b'00111011'        ;Dato
;-----
SUBWF    TXDATO1,W           59d=3Bh
BTFSC   STATUS,Z           ;-----
GOTO    CONVE59            ;-----

```

```

;
;          MOVLW    b'00111100'          ;-----
;          SUBWF   TXDATO1,W             ;-----
;          BTFSC   STATUS,Z              ;-----
;          GOTO    CONVE60               ;-----
;
;          RETURN                         ;-----
;=====
CONVE20    MOVLW    b'00010100'          ;Dato
;          SUBWF   TXDATO2,W             ;-----
;          BTFSC   STATUS,Z              ;MOTOR A DERECHAS
;          GOTO    CONV20                ;-----
;
;          MOVLW    b'00010101'          ;Dato
;          SUBWF   TXDATO2,W             ;-----
;          BTFSC   STATUS,Z              ;MOTOR A DERECHAS
;          GOTO    CONV21                ;-----
;
;          MOVLW    b'00010110'          ;Dato
;          SUBWF   TXDATO2,W             ;-----
;          BTFSC   STATUS,Z              ;MOTOR A DERECHAS
;          GOTO    CONV22                ;-----
;
;          MOVLW    b'00010111'          ;Dato
;          SUBWF   TXDATO2,W             ;-----
;          BTFSC   STATUS,Z              ;MOTOR A DERECHAS
;          GOTO    CONV23                ;-----
;
;          MOVLW    b'00011000'          ;Dato
;          SUBWF   TXDATO2,W             ;-----
;          BTFSC   STATUS,Z              ;MOTOR A DERECHAS
;          GOTO    CONV24                ;-----
;
;          MOVLW    b'00011001'          ;Dato
;          SUBWF   TXDATO2,W             ;-----
;          BTFSC   STATUS,Z              ;MOTOR A DERECHAS
;          GOTO    CONV25                ;-----
;
;          MOVLW    b'00011010'          ;Dato

```

```

                26d=1Ah
                SUBWF    TXDATO2,W    ;-----
                BTFSC    STATUS,Z     ;MOTOR A DERECHAS
                GOTO     CONV26       ;-----
;
CONVE27        MOVLW    b'00011011'  ;Dato
                27d=1Bh
                SUBWF    TXDATO2,W    ;-----
                BTFSC    STATUS,Z     ;MOTOR A DERECHAS
                GOTO     CONV27       ;-----
;
CONVE56        MOVLW    b'00111000'  ;Dato
                56d=38h
                SUBWF    TXDATO1,W    ;-----
                BTFSC    STATUS,Z     ;MOTOR A DERECHAS
                GOTO     CONV56       ;-----
;
CONVE57        MOVLW    b'00111001'  ;Dato
                57d=39h
                SUBWF    TXDATO1,W    ;-----
                BTFSC    STATUS,Z     ;MOTOR A DERECHAS
                GOTO     CONV57       ;-----
;
CONVE58        MOVLW    b'00111010'  ;Dato
                58d=3Ah
                SUBWF    TXDATO1,W    ;-----
                BTFSC    STATUS,Z     ;MOTOR A DERECHAS
                GOTO     CONV58       ;-----
;
CONVE59        MOVLW    b'00111011'  ;Dato
                59d=3Bh
                SUBWF    TXDATO1,W    ;-----
                BTFSC    STATUS,Z     ;MOTOR A DERECHAS
                GOTO     CONV59       ;-----
;
CONVE60        MOVLW    b'00111100'  ;Dato
                60d=3Ch
                SUBWF    TXDATO1,W    ;-----
                BTFSC    STATUS,Z     ;MOTOR A DERECHAS
                GOTO     CONV60       ;-----
;
                RETURN                ;-----
;
CONV16         MOVLW    b'00110001'  ;Código 16
                MOVWF   DECENAS       ;-----

```

```

MOV LW    b'00110110'    ;-----
MOV WF    UNIDADES      ;-----
CALL      CS557A         ;-----
MOV F     TXDATO2,W     ;MOTOR 1 VELOCIDAD
MOV WF    PORTD         ;-----
BCF       A0            ;-----
BCF       A1            ;-----
BCF       WWRR          ;-----
NOP       ;-----
BSF       WWRR          ;-----
CALL      CS5355D       ;-----
BSF       PCLATH,3     ;Página 1
BCF       PCLATH,4     ;-----
GOTO      DATTX1       ;-----
CONV17    MOV LW    b'00110001' ;Código 17
          MOV WF    DECENAS    ;-----
          MOV LW    b'00110111' ;-----
          MOV WF    UNIDADES    ;-----
          MOV LW    b'00000001' ;Dato
          ;27d=1Bh
          SUB WF    TXDATO2,W   ;-----
          BT FSC    STATUS,Z    ;MOTOR A DERECHAS
          GOTO      MOT5ON      ;-----
          MOV LW    b'00000010' ;-----
          AND WF    CONTA42,F   ;-----
          GOTO      MOTOR5      ;-----
MOT5ON    MOV LW    b'00000011' ;-----
          AND WF    CONTA42,F   ;-----
          MOV LW    b'00000001' ;-----
          IOR WF    CONTA42,F   ;-----
MOTOR5    CALL      CS557A     ;-----
          MOV F     CONTA42,W   ;-----
          MOV WF    PORTD     ;-----
          BCF       A0         ;-----
          BSF       A1         ;-----
          BCF       WWRR      ;-----
          NOP       ;-----
          BSF       WWRR      ;-----
          CALL      CS5355D    ;-----
          BSF       PCLATH,3   ;Página 1
          BCF       PCLATH,4   ;-----
          GOTO      DATTX1     ;-----
CONV18    MOV LW    b'00110001' ;Código 18
          MOV WF    DECENAS    ;-----
          MOV LW    b'00111000' ;-----
          MOV WF    UNIDADES    ;-----

```

```

                CALL    CS557A                ;-----
                MOVF   TXDATO2,W             ;MOTOR 2 VELOCIDAD
                MOVWF  PORTD                ;-----
                BSF    A0                    ;-----
                BCF    A1                    ;-----
                BCF    WWRR                  ;-----
                NOP                                     ;-----
                BSF    WWRR                  ;-----
                CALL   CS5355D               ;-----
                BSF    PCLATH,3              ;Página 1
                BCF    PCLATH,4              ;-----
                GOTO   DATTX1                ;-----
CONV19          MOVLW  b'00110001'          ;Código 19
                MOVWF  DECENAS               ;-----
                MOVLW  b'00111001'          ;-----
                MOVWF  UNIDADES              ;-----
                MOVLW  b'00000001'          ;Dato
                                                27d=1Bh
                SUBWF  TXDATO2,W             ;-----
                BTFSC  STATUS,Z              ;MOTOR A DERECHAS
                GOTO   MOT6ON                ;-----
                MOVLW  b'000000001'         ;-----
                ANDWF  CONTA42,F             ;-----
                GOTO   MOTOR6                ;-----
MOT6ON         MOVLW  b'00000011'          ;-----
                ANDWF  CONTA42,F             ;-----
                MOVLW  b'00000010'          ;-----
                IORWF  CONTA42,F             ;-----
MOTOR6         CALL   CS557A                ;-----
                MOVF   CONTA42,W             ;-----
                MOVWF  PORTD                ;-----
                BCF    A0                    ;-----
                BSF    A1                    ;-----
                BCF    WWRR                  ;-----
                NOP                                     ;-----
                BSF    WWRR                  ;-----
                CALL   CS5355D               ;-----
                BSF    PCLATH,3              ;Página 1
                BCF    PCLATH,4              ;-----
                GOTO   DATTX1                ;-----
;
CONV20         MOVLW  b'00110010'          ;Código 20
                MOVWF  DECENAS               ;-----
                MOVLW  b'00110000'          ;-----
                MOVWF  UNIDADES              ;-----
                BSF    PCLATH,3              ;Página 1

```

```

BCF      PCLATH,4      ;-----
GOTO     DATTX1        ;-----
;
CONV21   MOVLW    b'00110010' ;Código 21
          MOVWF   DECENAS      ;-----
          MOVLW    b'00110001' ;-----
          MOVWF   UNIDADES     ;-----
          BSF     PCLATH,3      ;Página 1
          BCF     PCLATH,4      ;-----
          GOTO     DATTX1        ;-----
;
CONV22   MOVLW    b'00111111' ;-----
          MOVWF   PULSOS       ;-----
          CALL    PULSOA        ;-----
          CLRF    PULSOS       ;-----
          CALL    PULSOA        ;-----
          MOVLW    b'00110010' ;Código 22
          MOVWF   DECENAS      ;-----
          MOVLW    b'00110010' ;-----
          MOVWF   UNIDADES     ;-----
          BSF     PCLATH,3      ;Página 1
          BCF     PCLATH,4      ;-----
          GOTO     DATTX1        ;-----
;
CONV23   MOVLW    b'00110010' ;Código 23
          MOVWF   DECENAS      ;-----
          MOVLW    b'00110011' ;-----
          MOVWF   UNIDADES     ;-----
          MOVLW    b'11110000' ;Prohíbo INT del esclavo
          CALL    OCW1M        ;-----
          BSF     PCLATH,3      ;Página 1
          BCF     PCLATH,4      ;-----
          GOTO     DATTX1        ;-----
;
CONV24   MOVLW    b'00110010' ;Código 24
          MOVWF   DECENAS      ;-----
          MOVLW    b'00110100' ;-----
          MOVWF   UNIDADES     ;-----
          MOVLW    b'11111100' ;Prohíbo INT del esclavo
          CALL    OCW1M        ;-----
          BSF     PCLATH,3      ;Página 1
          BCF     PCLATH,4      ;-----
          GOTO     DATTX1        ;-----
;
CONV25   MOVLW    b'01000000' ;-----

```

```

MOVWF PULSOS ;-----
CALL PULSOA ;-----
MOVLW b'00110010' ;Código 25
MOVWF DECENAS ;-----
MOVLW b'00110101' ;-----
MOVWF UNIDADES ;-----
BSF PCLATH,3 ;Página 1
BCF PCLATH,4 ;-----
GOTO DATTX1 ;-----
; ;-----
CONV26 MOVLW b'10000000' ;-----
MOVWF PULSOS ;-----
CALL PULSOA ;-----
MOVLW b'00110010' ;Código 26
MOVWF DECENAS ;-----
MOVLW b'00110110' ;-----
MOVWF UNIDADES ;-----
BSF PCLATH,3 ;Página 1
BCF PCLATH,4 ;-----
GOTO DATTX1 ;-----
; ;-----
CONV27 MOVLW b'00110010' ;Código 27
MOVWF DECENAS ;-----
MOVLW b'00110111' ;-----
MOVWF UNIDADES ;-----
MOVLW b'01000000' ;-----
MOVWF PULSOS ;-----
CALL PULSOA ;-----
CALL CS557A ;-----
MOVLW b'01011111' ;Motor 5 velocidad alta
95d=5Fh
MOVWF PORTD ;-----
BCF A0 ;-----
BCF A1 ;-----
BCF WWRR ;-----
NOP ;-----
BSF WWRR ;-----
CALL CS5355D ;-----
BSF PCLATH,3 ;Página 1
BCF PCLATH,4 ;-----
GOTO DATTX1 ;-----
; ;-----
CONV28 MOVLW b'00110010' ;Código 28
MOVWF DECENAS ;-----
MOVLW b'00111000' ;-----
MOVWF UNIDADES ;-----

```



```

MOV LW    b'10000000'      ;-----
MOV WF    PULSOS           ;-----
CALL      PULSOA           ;-----
CALL      CS557A           ;-----
MOV LW    b'10100000'      ;160d=A0h Motor 6
                                velocidad
MOV WF    PORTD           ;-----
BSF       A0               ;-----
BCF       A1               ;-----
BCF       WWRR            ;-----
NOP                               ;-----
BSF       WWRR            ;-----
CALL      CS5355D         ;-----
;                               ;-----
CALL      CS557A           ;-----
MOV LW    b'00000011'      ;motores.
MOV WF    PORTD           ;-----
BCF       A0               ;-----
BSF       A1               ;-----
BCF       WWRR            ;-----
NOP                               ;-----
BSF       WWRR            ;-----
CALL      CS5355D         ;-----
MOV LW    b'11110100'      ;Prohíbo INT2 del eje 5
CALL      OCW1M           ;-----
BSF       PCLATH,3        ;Página 1
BCF       PCLATH,4        ;-----
GOTO     DATTX1           ;-----
;                               ;-----
CONV29   CALL      CS557A           ;-----
MOV LW    b'01101110'      ;Motor 5 velocidad
                                mín.110d=6Eh
MOV WF    PORTD           ;-----
BCF       A0               ;-----
BCF       A1               ;-----
BCF       WWRR            ;-----
NOP                               ;-----
BSF       WWRR            ;-----
CALL      CS5355D         ;-----
;                               ;-----
MOV LW    b'11110000'      ;Habilita INT 2 y 3
CALL      OCW1M           ;-----
MOV LW    b'00110010'      ;Código 29
MOV WF    DECENAS         ;-----
MOV LW    b'00111001'      ;-----
MOV WF    UNIDADES        ;-----

```

```

                BSF      PCLATH,3          ;Página 1
                BCF      PCLATH,4          ;-----
                GOTO     DATTX1           ;-----
;
;=====
;=====LECTURA
CONTADORES=====
;=====
CONV30          CALL     CS531A           ;Contador 0 82531
                MOVF    CONTAL0,W        ;Lectura conta 0
                CALL    CONEL            ;-----
                NOP     ;-----
                CALL    CS531A           ;-----
                BCF     A0                ;-----
                BCF     A1                ;-----
                CALL    LEECON1          ;-----
                NOP     ;-----
                CALL    CS531A           ;-----
                CALL    LEECON2          ;-----
                NOP     ;-----
                BSF     PCLATH,3          ;Página 1
                BCF     PCLATH,4          ;-----
                GOTO     BCD16           ;-----
;
CONV31          CALL     CS531A           ;Contador 1 82531
                MOVF    CONTAL1,W        ;Lectura conta 1
                CALL    CONEL            ;-----
                NOP     ;-----
                CALL    CS531A           ;-----
                BSF     A0                ;-----
                BCF     A1                ;-----
                CALL    LEECON1          ;-----
                NOP     ;-----
                CALL    CS531A           ;-----
                CALL    LEECON2          ;-----
                NOP     ;-----
                BSF     PCLATH,3          ;Página 1
                BCF     PCLATH,4          ;-----
                GOTO     BCD16           ;-----
;
CONV32          CALL     CS531A           ;Contador 2 82531
                MOVF    CONTAL2,W        ;Lectura conta 2
                CALL    CONEL            ;-----
                NOP     ;-----
                CALL    CS531A           ;-----
                BCF     A0                ;-----

```

```

BSF      A1                ;-----
CALL     LEECON1          ;-----
NOP      ;-----
CALL     CS531A           ;-----
CALL     LEECON2          ;-----
NOP      ;-----
BSF      PCLATH,3         ;Página 1
BCF      PCLATH,4         ;-----
GOTO     BCD16            ;-----
;-----
;
CONV33   CALL     CS532A           ;Contador 0 82532
        MOVF     CONTAL0,W        ;Lectura conta 0
        CALL     CONEL            ;-----
        NOP      ;-----
        CALL     CS532A           ;-----
        BCF      A0                ;-----
        BCF      A1                ;-----
        CALL     LEECON1          ;-----
        NOP      ;-----
        CALL     CS532A           ;-----
        CALL     LEECON2          ;-----
        NOP      ;-----
        BSF      PCLATH,3         ;Página 1
        BCF      PCLATH,4         ;-----
        GOTO     BCD16            ;-----
;-----
;
CONV34   CALL     CS532A           ;Contador 1 82532
        MOVF     CONTAL1,W        ;Lectura conta 1
        CALL     CONEL            ;-----
        NOP      ;-----
        CALL     CS532A           ;-----
        BSF      A0                ;-----
        BCF      A1                ;-----
        CALL     LEECON1          ;-----
        NOP      ;-----
        CALL     CS532A           ;-----
        CALL     LEECON2          ;-----
        NOP      ;-----
        BSF      PCLATH,3         ;Página 1
        BCF      PCLATH,4         ;-----
        GOTO     BCD16            ;-----
;-----
;
CONV35   CALL     CS532A           ;Contador 2 82532
        MOVF     CONTAL2,W        ;Lectura conta 2
        CALL     CONEL            ;-----

```

```

NOP                                ;-----
CALL    CS532A                      ;-----
BCF     A0                          ;-----
BSF     A1                          ;-----
CALL    LEECON1                     ;-----
NOP                                ;-----
CALL    CS532A                      ;-----
CALL    LEECON2                     ;-----
NOP                                ;-----
BSF     PCLATH,3                    ;Página 1
BCF     PCLATH,4                    ;-----
GOTO    BCD16                       ;-----
;
CONV36  CALL    CS533A                ;Contador 0 82533
        MOVF   CONTAL0,W             ;Lectura conta 0
        CALL   CONEL                 ;-----
        NOP                                ;-----
        CALL   CS533A                ;-----
        BCF    A0                    ;-----
        BCF    A1                    ;-----
        CALL   LEECON1               ;-----
        NOP                                ;-----
        CALL   CS533A                ;-----
        CALL   LEECON2               ;-----
        NOP                                ;-----
        BSF    PCLATH,3              ;Página 1
        BCF    PCLATH,4              ;-----
        GOTO   BCD16                 ;-----
;
CONV37  CALL    CS533A                ;Contador 1 82533
        MOVF   CONTAL1,W             ;Lectura conta 1
        CALL   CONEL                 ;-----
        NOP                                ;-----
        CALL   CS533A                ;-----
        BSF    A0                    ;-----
        BCF    A1                    ;-----
        CALL   LEECON1               ;-----
        NOP                                ;-----
        CALL   CS533A                ;-----
        CALL   LEECON2               ;-----
        NOP                                ;-----
        BSF    PCLATH,3              ;Página 1
        BCF    PCLATH,4              ;-----
        GOTO   BCD16                 ;-----
;

```

```

CONV38      CALL      CS533A      ;Contador 2 82533
            MOVF      CONTAL2,W  ;Lectura conta 2
            CALL      CONEL      ;-----
            NOP      ;-----
            CALL      CS533A      ;-----
            BCF      A0      ;-----
            BSF      A1      ;-----
            CALL      LEECON1     ;-----
            NOP      ;-----
            CALL      CS533A      ;-----
            CALL      LEECON2     ;-----
            NOP      ;-----
            BSF      PCLATH,3    ;Página 1
            BCF      PCLATH,4    ;-----
            GOTO     BCD16      ;-----
;
CONV39      CALL      CS534A      ;Contador 0 82534
            MOVF      CONTAL0,W  ;Lectura conta 0
            CALL      CONEL      ;-----
            NOP      ;-----
            CALL      CS534A      ;-----
            BCF      A0      ;-----
            BCF      A1      ;-----
            CALL      LEECON1     ;-----
            NOP      ;-----
            CALL      CS534A      ;-----
            CALL      LEECON2     ;-----
            NOP      ;-----
            BSF      PCLATH,3    ;Página 1
            BCF      PCLATH,4    ;-----
            GOTO     BCD16      ;-----
;
CONV40      CALL      CS534A      ;Contador 1 82534
            MOVF      CONTAL1,W  ;Lectura conta 1
            CALL      CONEL      ;-----
            NOP      ;-----
            CALL      CS534A      ;-----
            BSF      A0      ;-----
            BCF      A1      ;-----
            CALL      LEECON1     ;-----
            NOP      ;-----
            CALL      CS534A      ;-----
            CALL      LEECON2     ;-----
            NOP      ;-----
            BSF      PCLATH,3    ;Página 1
    
```

```

                BCF      PCLATH,4          ;-----
                GOTO    BCD16             ;-----
;
CONV41          CALL    CS534A           ;Contador 2 82534
                MOVF   CONTAL2,W        ;Lectura conta 2
                CALL   CONEL            ;-----
                NOP                      ;-----
                CALL   CS534A           ;-----
                BCF    A0                ;-----
                BSF    A1                ;-----
                CALL   LEECON1          ;-----
                NOP                      ;-----
                CALL   CS534A           ;-----
                CALL   LEECON2          ;-----
                NOP                      ;-----
                BSF    PCLATH,3         ;Página 1
                BCF    PCLATH,4         ;-----
                GOTO    BCD16           ;-----
;=====
=====
;=====RECARGA
CONTADORES=====
;=====
=====
CONV42          MOVF   TXDATO2,W        ;Contador 0 82531
                MOVWF  CONTADORB       ;Re-programación conta 0
                MOVF   TXDATO3,W        ;-----
                MOVWF  CONTADORA        ;-----
                CALL   CS531A           ;-----
                MOVF   CONTAEO,W        ;-----
                CALL   CONEL            ;-----
                NOP                      ;-----
                CALL   CS531A           ;-----
                BCF    A0                ;-----
                BCF    A1                ;-----
                CALL   BYTEB            ;-----
                NOP                      ;-----
                CALL   CS531A           ;-----
                CALL   BYTEA            ;-----
                NOP                      ;-----
                MOVLW  b'00000001'     ;-----
                MOVWF  PULSOS           ;-----
                CALL   PULSOA           ;-----
                CLRF   PULSOS           ;-----
                CALL   PULSOA           ;-----
                MOVLW  b'00000001'     ;-----

```

```

MOVWF PULSOS ;-----
CALL PULSOA ;-----
CLRF PULSOS ;-----
CALL PULSOA ;-----
RETURN ;-----
; ;-----
CONV43 MOVF TXDATO2,W ;Contador 1 82531
MOVWF CONTADORB ;Re-programación conta 1
MOVF TXDATO3,W ;-----
MOVWF CONTADORA ;-----
CALL CS531A ;-----
MOVF CONTAE1,W ;-----
CALL CONEL ;-----
NOP ;-----
CALL CS531A ;-----
BSF A0 ;-----
BCF A1 ;-----
CALL BYTEB ;-----
NOP ;-----
CALL CS531A ;-----
CALL BYTEA ;-----
NOP ;-----
MOVLW b'00000010' ;-----
MOVWF PULSOS ;-----
CALL PULSOA ;-----
CLRF PULSOS ;-----
CALL PULSOA ;-----
MOVLW b'00000010' ;-----
MOVWF PULSOS ;-----
CALL PULSOA ;-----
CLRF PULSOS ;-----
CALL PULSOA ;-----
RETURN ;-----
; ;-----
CONV44 MOVF TXDATO2,W ;Contador 2 82531
MOVWF CONTADORB ;Re-programación conta 2
MOVF TXDATO3,W ;-----
MOVWF CONTADORA ;-----
CALL CS531A ;-----
MOVF CONTAE2,W ;-----
CALL CONEL ;-----
NOP ;-----
CALL CS531A ;-----
BCF A0 ;-----
BSF A1 ;-----

```

```

CALL      BYTEB          ;-----
NOP
CALL      CS531A        ;-----
CALL      BYTEA          ;-----
NOP
MOVLW    b'00000100'    ;-----
MOVWF    PULSOS         ;-----
CALL     PULSOA         ;-----
CLRF     PULSOS         ;-----
CALL     PULSOA         ;-----
MOVLW    b'00000100'    ;-----
MOVWF    PULSOS         ;-----
CALL     PULSOA         ;-----
CLRF     PULSOS         ;-----
CALL     PULSOA         ;-----
RETURN   ;-----
;
CONV45   MOVF     TXDATO2,W      ;Contador 0 82532
MOVWF    CONTADORB           ;Re-programación conta 0
MOVF     TXDATO3,W          ;
MOVWF    CONTADORA          ;
CALL     CS532A              ;
MOVF     CONTAE0,W          ;
CALL     CONEL              ;
NOP
CALL     CS532A              ;
BCF     A0                  ;
BCF     A1                  ;
CALL     BYTEB              ;
NOP
CALL     CS532A              ;
CALL     BYTEA              ;
NOP
MOVLW    b'00001000'        ;
MOVWF    PULSOS             ;
CALL     PULSOA             ;
CLRF     PULSOS             ;
CALL     PULSOA             ;
MOVLW    b'00001000'        ;
MOVWF    PULSOS             ;
CALL     PULSOA             ;
CLRF     PULSOS             ;
CALL     PULSOA             ;
RETURN
;

```



```

CONV46      MOVF      TXDATO2,W           ;Contador 1 82532
            MOVWF     CONTADORB        ;Re-programación conta 1
            MOVF      TXDATO3,W           ;-----
            MOVWF     CONTADORA        ;-----
            CALL      CS532A           ;-----
            MOVF      CONTAE1,W         ;-----
            CALL      CONEL            ;-----
            NOP                          ;-----
            CALL      CS532A           ;-----
            BSF       A0                ;-----
            BCF       A1                ;-----
            CALL      BYTEB            ;-----
            NOP                          ;-----
            CALL      CS532A           ;-----
            CALL      BYTEA            ;-----
            NOP                          ;-----
            MOVLW     b'00010000'      ;-----
            MOVWF     PULSOS           ;-----
            CALL      PULSOA           ;-----
            CLRF      PULSOS           ;-----
            CALL      PULSOA           ;-----
            MOVLW     b'00010000'      ;-----
            MOVWF     PULSOS           ;-----
            CALL      PULSOA           ;-----
            CLRF      PULSOS           ;-----
            CALL      PULSOA           ;-----
            RETURN                       ;-----
;
CONV47      MOVF      TXDATO2,W           ;Contador 2 82532
            MOVWF     CONTADORB        ;Re-programación conta 0
            MOVF      TXDATO3,W           ;-----
            MOVWF     CONTADORA        ;-----
            CALL      CS532A           ;-----
            MOVF      CONTAE2,W         ;-----
            CALL      CONEL            ;-----
            NOP                          ;-----
            CALL      CS532A           ;-----
            BCF       A0                ;-----
            BSF       A1                ;-----
            CALL      BYTEB            ;-----
            NOP                          ;-----
            CALL      CS532A           ;-----
            CALL      BYTEA            ;-----
            NOP                          ;-----
            MOVLW     b'00100000'      ;-----

```

```

MOVWF PULSOS ;-----
CALL PULSOA ;-----
CLRF PULSOS ;-----
CALL PULSOA ;-----
MOVLW b'00100000' ;-----
MOVWF PULSOS ;-----
CALL PULSOA ;-----
CLRF PULSOS ;-----
CALL PULSOA ;-----
RETURN ;-----
;
CONV48 MOVF TXDATO2,W ;Contador 0 82533
MOVWF CONTADORB ;Re-programación conta 0
MOVF TXDATO3,W ;-----
MOVWF CONTADORA ;-----
CALL CS533A ;-----
MOVF CONTAE0,W ;-----
CALL CONEL ;-----
NOP ;-----
CALL CS533A ;-----
BCF A0 ;-----
BCF A1 ;-----
CALL BYTEB ;-----
NOP ;-----
CALL CS533A ;-----
CALL BYTEA ;-----
RETURN ;-----
;
CONV49 MOVF TXDATO2,W ;Contador 1 82533
MOVWF CONTADORB ;Re-programación conta 1
MOVF TXDATO3,W ;-----
MOVWF CONTADORA ;-----
CALL CS533A ;-----
MOVF CONTAE1,W ;-----
CALL CONEL ;-----
NOP ;-----
CALL CS533A ;-----
BSF A0 ;-----
BCF A1 ;-----
CALL BYTEB ;-----
NOP ;-----
CALL CS533A ;-----
CALL BYTEA ;-----
RETURN ;-----
;

```

```

CONV50      MOVF      TXDATO2,W           ;Contador 2 82533
            MOVWF     CONTADORB        ;Re-programación conta 2
            MOVF      TXDATO3,W           ;-----
            MOVWF     CONTADORA        ;-----
            CALL      CS533A           ;-----
            MOVF      CONTAE2,W         ;-----
            CALL      CONEL            ;-----
            NOP                          ;-----
            CALL      CS533A           ;-----
            BCF       A0                ;-----
            BSF       A1                ;-----
            CALL      BYTEB            ;-----
            NOP                          ;-----
            CALL      CS533A           ;-----
            CALL      BYTEA            ;-----
            RETURN                       ;-----
;
CONV51      MOVF      TXDATO2,W           ;Contador 0 82534
            MOVWF     CONTADORB        ;Re-programación conta 0
            MOVF      TXDATO3,W           ;-----
            MOVWF     CONTADORA        ;-----
            CALL      CS534A           ;-----
            MOVF      CONTAE0,W         ;-----
            CALL      CONEL            ;-----
            NOP                          ;-----
            CALL      CS534A           ;-----
            BCF       A0                ;-----
            BCF       A1                ;-----
            CALL      BYTEB            ;-----
            NOP                          ;-----
            CALL      CS534A           ;-----
            CALL      BYTEA            ;-----
            RETURN                       ;-----
;
CONV52      MOVF      TXDATO2,W           ;Contador 1 82534
            MOVWF     CONTADORB        ;Re-programación conta 1
            MOVF      TXDATO3,W           ;-----
            MOVWF     CONTADORA        ;-----
            CALL      CS534A           ;-----
            MOVF      CONTAE1,W         ;-----
            CALL      CONEL            ;-----
            NOP                          ;-----
            CALL      CS534A           ;-----
            BSF       A0                ;-----
            BCF       A1                ;-----

```

```

CALL      BYTEB      ;-----
NOP      ;-----
CALL      CS534A     ;-----
CALL      BYTEA     ;-----
RETURN   ;-----
;
CONV53    MOVF        TXDATO2,W      ;Contador 2 82534
MOVWF    CONTADORB ;Re-programacion conta 2
MOVF     TXDATO3,W      ;-----
MOVWF    CONTADORA    ;-----
CALL     CS534A        ;-----
MOVF     CONTAE2,W     ;-----
CALL     CONEL        ;-----
NOP      ;-----
CALL     CS534A        ;-----
BCF      A0           ;-----
BSF      A1           ;-----
CALL     BYTEB        ;-----
NOP      ;-----
CALL     CS534A        ;-----
CALL     BYTEA        ;-----
RETURN   ;-----
;
CONV54    MOVLW      b'00110101'    ;Código 54
MOVWF    DECENAS     ;-----
MOVLW      b'00110100'    ;-----
MOVWF    UNIDADES    ;-----
BSF      PCLATH,3     ;Página 1
BCF      PCLATH,4     ;-----
CALL     DATTX1      ;-----
;
CALL     CS557A      ;-----
MOVLW      b'01111000'    ;Motor 5 velocidad
                               mín.120d=78h
MOVWF    PORTD      ;-----
BCF      A0         ;-----
BCF      A1         ;-----
BCF      WWRR      ;-----
NOP      ;-----
BSF      WWRR      ;-----
CALL     CS5355D    ;-----
;
BSF      PCLATH,3   ;Página 1
BCF      PCLATH,4   ;-----
CALL     RETA0     ;-----
CALL     RETA0     ;-----

```

```

CALL    RETA0    ;-----
CALL    RETA0    ;-----
CALL    RETA0    ;-----
BCF     PCLATH,3 ;Página 0
BCF     PCLATH,4 ;-----
;
CALL    DESAC    ;-----
;
CALL    CS557A   ;-----
MOVLW  b'10001111' ;143d=8Fh Motor 6
                                velocidad mínima
MOVWF   PORTD    ;-----
BSF     A0        ;-----
BCF     A1        ;-----
BCF     WWRR      ;-----
NOP     ;-----
BSF     WWRR      ;-----
CALL    CS5355D  ;-----
;
CALL    CS557A   ;-----
MOVLW  b'00000011' ; Desactivamos motores.
MOVWF   PORTD    ;-----
BCF     A0        ;-----
BSF     A1        ;-----
BCF     WWRR      ;-----
NOP     ;-----
BSF     WWRR      ;-----
CALL    CS5355D  ;-----
;
BSF     PCLATH,3 ;Página 1
BCF     PCLATH,4 ;-----
CALL    RETA0    ;-----
CALL    RETA0    ;-----
CALL    RETA0    ;-----
CALL    RETA0    ;-----
CALL    RETA0    ;-----
BCF     PCLATH,3 ;Página 0
BCF     PCLATH,4 ;-----
;
CALL    DESAC    ;-----
RETURN  ;-----
;
CONV55  MOVLW  b'00110101' ;Código 55
MOVWF   DECENAS  ;-----
MOVLW  b'00110101' ;-----
MOVWF   UNIDADES ;-----

```

```

        BSF      PCLATH,3      ;Página 1
        BCF      PCLATH,4      ;-----
        CALL     DATTX1       ;-----
;
        CALL     CS557A        ;-----
        MOVLW    b'10010001'   ;Motor 5 velocidad
                                mín.145d=91h
        MOVWF    PORTD        ;-----
        BCF      A0            ;-----
        BCF      A1            ;-----
        BCF      WWRR         ;-----
        NOP      ;-----
        BSF      WWRR         ;-----
        CALL     CS5355D       ;-----
;
        BSF      PCLATH,3      ;Página 1
        BCF      PCLATH,4      ;-----
        CALL     RETA0         ;-----
        CALL     RETA0         ;-----
        CALL     RETA0         ;-----
        BCF      PCLATH,3      ;Página 0
        BCF      PCLATH,4      ;-----
;
        CALL     DESAC         ;-----
;
        CALL     CS557A        ;-----
        MOVLW    b'10001111'   ;143d=8Fh Motor 6
                                velocidad mínima
        MOVWF    PORTD        ;-----
        BSF      A0            ;-----
        BCF      A1            ;-----
        BCF      WWRR         ;-----
        NOP      ;-----
        BSF      WWRR         ;-----
        CALL     CS5355D       ;-----
;
        CALL     CS557A        ;-----
        MOVLW    b'00000011'   ;Desactivamos motores.
        MOVWF    PORTD        ;-----
        BCF      A0            ;-----
        BSF      A1            ;-----
        BCF      WWRR         ;-----
        NOP      ;-----
        BSF      WWRR         ;-----
        CALL     CS5355D       ;-----
;

```

```

BSF      PCLATH,3      ;Página 1
BCF      PCLATH,4      ;-----
CALL     RETA0         ;-----
CALL     RETA0         ;-----
CALL     RETA0         ;-----
BCF      PCLATH,3      ;Página 0
BCF      PCLATH,4      ;-----
;
CALL     DESAC         ;-----
;
CALL     CS557A        ;-----
MOVLW   b'01101010'   ;Motor 5 velocidad
                        mín.106d=6Ah
MOVWF   PORTD         ;-----
BCF     A0             ;-----
BCF     A1             ;-----
BCF     WWRR          ;-----
NOP     ;-----
BSF     WWRR          ;-----
CALL    CS5355D       ;-----
;
BSF     PCLATH,3      ;Página 1
BCF     PCLATH,4      ;-----
CALL    RETA0         ;-----
CALL    RETA0         ;-----
BCF     PCLATH,3      ;Página 0
BCF     PCLATH,4      ;-----
;
CALL    DESAC         ;-----
;
CALL    CS557A        ;-----
MOVLW   b'01101010'   ;106d=6Ah Motor 6
                        velocidad mínima
MOVWF   PORTD         ;-----
BSF     A0             ;-----
BCF     A1             ;-----
BCF     WWRR          ;-----
NOP     ;-----
BSF     WWRR          ;-----
CALL    CS5355D       ;-----
;
CALL    CS557A        ;-----
MOVLW   b'00000011'   ;Desactivamos motores.
MOVWF   PORTD         ;-----
BCF     A0             ;-----
BSF     A1             ;-----

```

```

        BCF      WWRR          ;-----
        NOP                      ;-----
        BSF      WWRR          ;-----
        CALL     CS5355D       ;-----
;
        BSF      PCLATH,3      ;Página 1
        BCF      PCLATH,4      ;-----
        CALL     RETA0         ;-----
        CALL     RETA0         ;-----
        CALL     RETA0         ;-----
        BCF      PCLATH,3      ;Página 0
        BCF      PCLATH,4      ;-----
;
        CALL     DESAC         ;-----
        RETURN                    ;-----
;
CONV56      RETURN            ;-----
CONV57      RETURN            ;-----
CONV58      RETURN            ;-----
CONV59      RETURN            ;-----
CONV60      RETURN            ;-----
;-----
;===== SUBRUTINAS =====
;-----
;=====programación interrupciones 8259M=====
;-----
PR8259M      MOVLW    b'00010101' ;ICW1 8259
              MOVWF   PORTD        ;-----
              BCF     A0            ;Necesita palabra ICW4
              CALL    CS59MA        ;Modo cascada=0
              UNICO=1
              BCF     WWRR          ;Intervalo direcciones 4
              NOP                      ;CALL      RETA52
              BSF     WWRR          ;Modo disparo flanco
              CALL    CS59D        ;-----
              NOP                      ;-----
              MOVLW   b'00000000'   ;ICW2 8259
              MOVWF   PORTD        ;-----
              BSF     A0            ;-----
              CALL    CS59MA        ;-----
              BCF     WWRR          ;-----
              NOP                      ;CALL      RETA52
              BSF     WWRR          ;-----
              CALL    CS59D        ;-----
              NOP                      ;-----
              MOVLW   b'10000000'   ;ICW3 8259
              MOVWF   PORTD        ;-----

```



```

BSF      A0                ;-----
CALL     CS59MA           ;-----
BCF      WWRR            ;-----
NOP                      ;CALL      RETA52
BSF      WWRR            ;-----
CALL     CS59D           ;-----
NOP                      ;-----
MOVLW   b'00000100'     ;ICW4 8259
MOVWF   PORTD           ;-----
BSF      A0              ;Micro 8085
CALL     CS59MA         ;Modo EOI automático
BCF      WWRR            ;Maestro
NOP                      ;CALL      RETA52
BSF      WWRR            ;No bufered
CALL     CS59D           ;Modo no especial
RETURN   ;-----
;=====Programación interrupciones 8259E=====
PR8259E  MOVLW   b'10010101' ;ICW1 8259
          MOVWF   PORTD     ;-----
          BCF     A0         ;Necesita palabra ICW4
          CALL    CS59EA     ;Modo cascada=0
                               UNICO=1
          BCF     WWRR       ;Intervalo direcciones 4
          NOP                      ;CALL      RETA52
          BSF     WWRR       ;Modo disparo flanco
          CALL    CS59D     ;-----
          NOP                      ;-----
          MOVLW   b'00000000' ;ICW2 8259
          MOVWF   PORTD     ;-----
          BSF     A0         ;-----
          CALL    CS59EA     ;-----
          BCF     WWRR       ;-----
          NOP                      ;CALL      RETA52
          BSF     WWRR       ;-----
          CALL    CS59D     ;-----
          NOP                      ;-----
          MOVLW   b'00000111' ;ICW3 8259
          MOVWF   PORTD     ;-----
          BSF     A0         ;-----
          CALL    CS59EA     ;-----
          BCF     WWRR       ;-----
          NOP                      ;CALL      RETA52
          BSF     WWRR       ;-----
          CALL    CS59D     ;-----
          NOP                      ;-----
          MOVLW   b'00000100' ;ICW4 8259
    
```

```

MOVWF PORTD ;-----
BSF A0 ;Micro 8085
CALL CS59EA ;Modo EOI automático
BCF WWRR ;Maestro
NOP ;CALL RETA52
BSF WWRR ;No bufered
CALL CS59D ;Modo no especial
RETURN ;-----
; ;-----
;=====Lectura interrupciones 8259M===== ;-----
LE8259 CALL PUDE ;-----
BCF INTA ;-----
NOP ;CALL RETA52
BSF INTA ;-----
NOP ;-----
BCF INTA ;-----
NOP ;CALL RETA52
MOVF PORTD,0 ;-----
MOVWF DATO8259 ;-----
BSF INTA ;-----
NOP ;-----
BCF INTA ;-----
NOP ;CALL RETA52
BSF INTA ;-----
CALL PUDS ;-----
RETURN ;-----
; ;-----
;=====Comando EOI para 8259M y 8259E===== ;-----
EOI MOVLW b'00100000' ;OCW2 comando EOI
MOVWF PORTD ;-----
BCF A0 ;Micro 8085
CALL CS59MA ;Modo EOI automático
BCF WWRR ;Maestro
NOP ;CALL RETA52
BSF WWRR ;No bufered
CALL CS59D ;Modo no especial
NOP ;-----
MOVLW b'00100000' ;OCW2 comando EOI
MOVWF PORTD ;-----
BCF A0 ;Micro 8085
CALL CS59EA ;Modo EOI automático
BCF WWRR ;Maestro
NOP ;CALL RETA52
BSF WWRR ;No bufered
CALL CS59D ;Modo no especial

```



```

MOVWF    PORTD                ;-----
BSF      A0                    ;-----
BSF      A1                    ;-----
BCF      WWRR                  ;-----
NOP                               ;CALL    RETA52
BSF      WWRR                  ;-----
CALL     CS5355D               ;-----
RETURN   ;-----
;=====programación 82558===== ;-----
PR82558  CALL     CS558A        ;-----
          MOVLW   b'10000000'   ;Todo en salida
          MOVWF   PORTD         ;-----
          BSF     A0             ;-----
          BSF     A1             ;-----
          BCF     WWRR          ;-----
          NOP                               ;CALL    RETA52
          BSF     WWRR          ;-----
          CALL    CS5355D       ;-----
          RETURN  ;-----
;=====CS activación 8255-7===== ;-----
CS557A   MOVLW   b'11111011'   ;-----
          MOVWF   PORTD         ;-----
          BSF     A0138         ;Activación
                               CS
          BCF     A1138         ;-----
          BCF     A2138         ;-----
          BSF     CS138         ;-----
          BCF     CS138         ;-----
          RETURN  ;-----
;=====CS activación 8255-8===== ;-----
CS558A   MOVLW   b'11111110'   ;-----
          MOVWF   PORTD         ;-----
          BSF     A0138         ;Activación
                               CS
          BCF     A1138         ;-----
          BCF     A2138         ;-----
          BSF     CS138         ;-----
          BCF     CS138         ;-----
          RETURN  ;-----
;=====CS activación 8253-1===== ;-----
CS531A   MOVLW   b'11110111'   ;-----
          MOVWF   PORTD         ;-----
          BSF     A0138         ;Activación
                               CS
          BCF     A1138         ;-----
          BCF     A2138         ;-----

```

```

        BSF      CS138      ;-----
        BCF      CS138      ;-----
        RETURN      ;-----
;=====CS activación 8253-2=====
CS532A  MOVLW    b'01111111' ;-----
        MOVWF   PORTD      ;-----
        BSF     A0138      ;Activación
                           CS
        BCF     A1138      ;-----
        BCF     A2138      ;-----
        BSF     CS138      ;-----
        BCF     CS138      ;-----
        RETURN      ;-----
;=====CS activación 8253-3=====
CS533A  MOVLW    b'10111111' ;-----
        MOVWF   PORTD      ;-----
        BSF     A0138      ;Activación
                           CS
        BCF     A1138      ;-----
        BCF     A2138      ;-----
        BSF     CS138      ;-----
        BCF     CS138      ;-----
        RETURN      ;-----
;=====CS activación 8253-
4=====
CS534A  MOVLW    b'11011111' ;-----
        MOVWF   PORTD      ;-----
        BSF     A0138      ;Activación
                           CS
        BCF     A1138      ;-----
        BCF     A2138      ;-----
        BSF     CS138      ;-----
        BCF     CS138      ;-----
        RETURN      ;-----
;=====CS desactivación 8253 y 8255=====
CS535D  MOVLW    b'11111111' ;-----
        MOVWF   PORTD      ;-----
        BSF     A0138      ;Desactivación CS
        BCF     A1138      ;-----
        BCF     A2138      ;-----
        BSF     CS138      ;-----
        BCF     CS138      ;-----
        RETURN      ;-----
;=====Control del bus de datos=====
PUDE    BSF      STATUS,5    ;Bus en entrada
        MOVLW   b'11111111' ;-----

```

```

MOVWF TRISD ;-----
BCF STATUS,5 ;-----
RETURN ;-----
; ;-----
PUDS BSF STATUS,5 ;Bus en
salida
MOVLW b'00000000' ;-----
MOVWF TRISD ;-----
BCF STATUS,5 ;-----
RETURN ;-----
; ;-----
;=====Programación escritura/lectura contadores===== ;-----
CONEL MOVWF PORTD ;-----
BSF A0 ;-----
BSF A1 ;-----
BCF WWRR ;-----
NOP ;-----
NOP ;-----
NOP ;-----
BSF WWRR ;-----
CALL CS5355D ;-----
RETURN ;-----
; ;-----
BYTEB MOVF CONTADORB,
W ;-----
MOVWF PORTD ;-----
BCF WWRR ;-----
NOP ;-----
NOP ;-----
NOP ;-----
BSF WWRR ;-----
CALL CS5355D ;-----
RETURN ;-----
BYTEA MOVF CONTADORA,
W ;-----
MOVWF PORTD ;-----
BCF WWRR ;-----
NOP ;-----
NOP ;-----
NOP ;-----
BSF WWRR ;-----
CALL CS5355D ;-----
RETURN ;-----
; ;-----
LEECON1 CALL PUDE ;-----
BCF RRDD ;-----

```

```

NOP                                     ;-----
NOP                                     ;-----
NOP                                     ;-----
MOVF      PORTD,0                       ;-----
MOVWF     RESULTADOD_A                  ;-----
BSF       RRDD                          ;-----
CALL      PUDS                          ;-----
CALL      CS5355D                       ;-----
RETURN                                         ;-----
;
LEECON2   CALL      PUDE                  ;-----
          BCF       RRDD                  ;-----
          NOP                                     ;-----
          NOP                                     ;-----
          NOP                                     ;-----
          MOVF      PORTD,0               ;-----
          MOVWF     RESULTADOD_B          ;-----
          BSF       RRDD                  ;-----
          CALL      PUDS                  ;-----
          CALL      CS5355D               ;-----
          RETURN                                         ;-----
;
;=====INICIALIZACION DE CONTADORES=====
;=====PULSO S1 Y S2 DEL CONTADOR 0 Y 1 DEL 82531=====
PULSOA   CALL      CS558A                 ;-----
          MOVF      PULSOS,W              ;-----
          MOVWF     PORTD                  ;-----
          BCF       A0                     ;-----
          BCF       A1                     ;-----
          BCF       WWRR                   ;-----
          NOP                                     ;-----
          BSF       WWRR                   ;-----
          CALL      CS5355D               ;-----
          BSF       PCLATH,3               ;Página 1
          BCF       PCLATH,4               ;-----
          CALL      RETA52                 ;-----
          BCF       PCLATH,3               ;Página 0
          BCF       PCLATH,4               ;-----
          RETURN                                         ;-----
;
;=====Desactivación de motores=====
DESAC    CALL      CS557A                 ;-----
          MOVLW     b'01111110'           ;Motor 5 velocidad
                                          mín.126d=7Eh
          MOVWF     PORTD                  ;-----
          BCF       A0                     ;-----

```

```

        BCF      A1                ;-----
        BCF      WWRR              ;-----
        NOP                      ;-----
        BSF      WWRR              ;-----
        CALL     CS5355D           ;-----
;                                           ;-----
        CALL     CS557A            ;-----
        MOVLW    b'10000000'      ;Motor 6 velocidad mínima
        MOVWF    PORTD            ;-----
        BSF      A0                ;-----
        BCF      A1                ;-----
        BCF      WWRR              ;-----
        NOP                      ;-----
        BSF      WWRR              ;-----
        CALL     CS5355D           ;-----
        CALL     CS558A            ;-----
;                                           ;-----
        CALL     CS557A            ;-----
        MOVLW    b'00000001'      ;Desactivamos motores.
        MOVWF    CONTA42          ;-----
        MOVWF    PORTD            ;-----
        BCF      A0                ;-----
        BSF      A1                ;-----
        BCF      WWRR              ;-----
        NOP                      ;-----
        BSF      WWRR              ;-----
        CALL     CS5355D           ;-----
;                                           ;-----
        MOVLW    b'00000000'      ;Desactivación de
                                           ;contadores
        MOVWF    PORTD            ;-----
        BCF      A0                ;-----
        BCF      A1                ;-----
        BCF      WWRR              ;-----
        NOP                      ;-----
        BSF      WWRR              ;-----
        CALL     CS5355D           ;-----
        RETURN                    ;-----
;=====PROGRAMACION CONTADORESES =====
INIC    MOVLW    b'00000010'      ;valor 00002d=0002h
        MOVWF    TXDATO2          ;-----
        MOVLW    b'00000000'      ;-----
        MOVWF    TXDATO3          ;-----
        CALL     CONV42            ;-----
        CALL     CONV43            ;-----
        CALL     CONV44            ;-----

```



```

CALL      CONV45      ;-----
CALL      CONV46      ;-----
CALL      CONV47      ;-----
MOVLW    b'11111111'  ;valor 255d=00FFh
MOVWF    TXDATO2      ;-----
MOVLW    b'00000000'  ;-----
MOVWF    TXDATO3      ;-----
CALL      CONV48      ;-----
CALL      CONV49      ;-----
CALL      CONV50      ;-----
CALL      CONV51      ;-----
CALL      CONV52      ;-----
CALL      CONV53      ;-----
;=====INICIALIZA CONTADORES 3 Y 4=====
MOVLW    b'00111111'  ;-----
MOVWF    PULSOS       ;-----
CALL     PULSOA       ;-----
CLRF     PULSOS       ;-----
CALL     PULSOA       ;-----
MOVLW    b'00111111'  ;-----
MOVWF    PULSOS       ;-----
CALL     PULSOA       ;-----
CLRF     PULSOS       ;-----
CALL     PULSOA       ;-----
MOVLW    b'00111111'  ;-----
MOVWF    PULSOS       ;-----
CALL     PULSOA       ;-----
CLRF     PULSOS       ;-----
CALL     PULSOA       ;-----
MOVLW    b'00111111'  ;-----
MOVWF    PULSOS       ;-----
CALL     PULSOA       ;-----
CLRF     PULSOS       ;-----
CALL     PULSOA       ;-----
;
MOVLW    b'11101000'  ;valor 65000d=FDE8h
MOVWF    TXDATO2      ;-----
MOVLW    b'11111101'  ;-----
MOVWF    TXDATO3      ;-----
CALL     CONV42       ;-----
CALL     CONV43       ;-----
CALL     CONV44       ;-----
CALL     CONV45       ;-----
CALL     CONV46       ;-----
CALL     CONV47       ;-----

```

```

                RETURN                ;-----
;
                ORG                    ;-----
                0X800                ;-----
;
;=====INCIÓ PANTALLA LCD=====
PANT1          CALL                    ;Retardo 980.000us
                RETA0
                MOVLW                  ;Bus 8 bits 4 líneas
                b'00111000'
                CALL                    ;-----
                COMANDO
                MOVLW                  ;cursor activo sin
                b'00001111'           ;parpadeo
;-----
BORRA          CALL                    ;-----
                COMANDO
                MOVLW                  ;borra la pantalla
                b'00000001'
                CALL                    ;-----
                COMANDO
                RETURN                ;-----
;=====ESCRITURA COMANDO=====
COMANDO        BCF                    ;Banco 0
                STATUS,5
                CLRF                   ;-----
                PORTD
                BSF                    ;-----
                CS138
                BCF                    ;-----
                A0138
                BCF                    ;-----
                A1138
                BCF                    ;-----
                A2138
                MOVWF                  ;guarda el comando en
                PROGRAMACION          ;memoria
;-----
                CALL                   ;-----
                RETA1
                BCF                    ;-----
                RSLCD
                BSF                    ;-----
                ELCD
                MOVF                   ;lleva el comando al W
                PROGRAMACION,0       ;lo saca por el bus de
                MOVWF                  ;datos
;-----
                BCF                    ;-----
                ELCD
                BCF                    ;-----
                CS138
                CLRF                   ;-----
                PORTD
                RETURN                ;-----
;=====ESCRITURA LETRAS=====
LETRA          BCF                    ;-----
                STATUS,5
                CLRF                   ;-----
                PORTD
                BSF                    ;-----
                CS138
                BCF                    ;-----
                A0138
                BCF                    ;-----
                A1138
                BCF                    ;-----
                A2138
                MOVWF                  ;guarda el comando en
                CARACTER              ;memoria
;-----
                CALL                   ;-----
                RETA52
                BSF                    ;-----
                RSLCD
                BSF                    ;-----
                ELCD
                MOVF                   ;lleva el carácter al W
                CARACTER,0

```

```

MOVWF    PORTD                ;lo saca por el bus de
                                datos
BCF      ELCD                  ;-----
BCF      CS138                 ;-----
CLRF     PORTD                 ;-----
RETURN   ;-----
TEXT09   MOVLW    b'10000000'  ;posición cursor dirección
                                00
CALL     COMANDO                ;-----
MOVLW   'P'                    ;-----
CALL    LETRA                   ;1
MOVLW   'r'                    ;-----
CALL    LETRA                   ;2
MOVLW   'o'                    ;-----
CALL    LETRA                   ;3
MOVLW   'g'                    ;-----
CALL    LETRA                   ;4
MOVLW   'r'                    ;-----
CALL    LETRA                   ;5
MOVLW   'a'                    ;-----
CALL    LETRA                   ;6
MOVLW   'm'                    ;-----
CALL    LETRA                   ;7
MOVLW   'a'                    ;-----
CALL    LETRA                   ;8
MOVLW   'd'                    ;-----
CALL    LETRA                   ;9
MOVLW   'o'                    ;-----
CALL    LETRA                   ;10
MOVLW   's'                    ;-----
CALL    LETRA                   ;11
MOVLW   ''                    ;-----
CALL    LETRA                   ;12
MOVLW   't'                    ;-----
CALL    LETRA                   ;13
MOVLW   'o'                    ;-----
CALL    LETRA                   ;14
MOVLW   'd'                    ;-----
CALL    LETRA                   ;15
MOVLW   'o'                    ;-----
CALL    LETRA                   ;16
MOVLW   's'                    ;-----
CALL    LETRA                   ;17
MOVLW   ''                    ;-----
CALL    LETRA                   ;18
MOVLW   ''                    ;-----

```

```

                CALL     LETRA                ;19
                RETURN                ;-----
;=====TEXTOS DE FUNCIONAMIENTO=====
;=====ADELANTE=====
TEXTO10        MOVLW    b'11000000'         ;posición cursor dirección
                                                40
                CALL     COMANDO            ;-----
                MOVLW    'l'                ;-----
                CALL     LETRA              ;1
                MOVLW    'o'                ;-----
                CALL     LETRA              ;2
                MOVLW    's'                ;-----
                CALL     LETRA              ;3
                MOVLW    '.'                ;-----
                CALL     LETRA              ;4
                MOVLW    'c'                ;-----
                CALL     LETRA              ;5
                MOVLW    'i'                ;-----
                CALL     LETRA              ;6
                MOVLW    'r'                ;-----
                CALL     LETRA              ;7
                MOVLW    'c'                ;-----
                CALL     LETRA              ;8
                MOVLW    'u'                ;-----
                CALL     LETRA              ;9
                MOVLW    'i'                ;-----
                CALL     LETRA              ;10
                MOVLW    't'                ;-----
                CALL     LETRA              ;11
                MOVLW    'o'                ;-----
                CALL     LETRA              ;12
                MOVLW    's'                ;-----
                CALL     LETRA              ;13
                MOVLW    '.'                ;-----
                CALL     LETRA              ;14
                MOVLW    '.'                ;-----
                CALL     LETRA              ;15
                MOVLW    '.'                ;-----
                CALL     LETRA              ;16
                RETURN                ;-----
;=====
=====
TEXTO11        MOVLW    'o'                ;-----
                CALL     LETRA              ;1
                MOVLW    'k'                ;-----
                CALL     LETRA              ;2

```

```

                RETURN                                ;-----
;=====INICIO 980.495,60us=====
RETA0          MOVLW    b'00011001'                  ;980.495,6us en 20MHz
                MOVWF   CONTA01                      ;guardamos valor en
                ;                                       CONTA01
UNO0           MOVLW    b'11111111'                  ;ponemos el contador a 01
                MOVWF   CONTA02                      ;guardamos valor en
                ;                                       CONTA01
DOS0           MOVLW    b'11111111'                  ;ponemos el contador a 8F
                MOVWF   CONTA03                      ;guardamos valor en
                ;                                       CONTA01
TRES0          DECFSZ   CONTA03,1                    ;decrementamos conta01,
                ;                                       si es cero salta
                GOTO    TRES0                          ;vuelve
                DECFSZ   CONTA02,1                    ;decrementamos conta02,
                ;                                       si es cero salta
                GOTO    DOS0                            ;volvemos a recargar
                ;                                       conta01
                DECFSZ   CONTA01,1                    ;decrementamos conta02,
                ;                                       si es cero salta
                GOTO    UNO0                            ;volvemos a recargar
                ;                                       conta01
                RETURN                                ;volver del retardo
;=====COMANDO 1.703,40us=====
RETA1          MOVLW    b'00000001'                  ;1.703,40us en 20MHz
                MOVWF   CONTA01                      ;guardamos valor en
                ;                                       CONTA01
UNO1           MOVLW    b'00001100'                  ;ponemos el contador a 01
                MOVWF   CONTA02                      ;guardamos valor en
                ;                                       CONTA01
DOS1           MOVLW    b'11101011'                  ;ponemos el contador a 8F
                MOVWF   CONTA03                      ;guardamos valor en
                ;                                       CONTA01
TRES1          DECFSZ   CONTA03,1                    ;decrementamos conta01,
                ;                                       si es cero salta
                GOTO    TRES1                          ;vuelve
                DECFSZ   CONTA02,1                    ;decrementamos conta02,
                ;                                       si es cero salta
                GOTO    DOS1                            ;volvemos a recargar
                ;                                       conta01
                DECFSZ   CONTA01,1                    ;decrementamos conta02,
                ;                                       si es cero salta
                GOTO    UNO1                            ;volvemos a recargar
                ;                                       conta01
                RETURN                                ;volver del retardo
;=====SERIE Y PALABRA 52us=====
RETA52         MOVLW    b'00000001'                  ;52us 9600 baudios
                MOVWF   CONTA01                      ;guardamos valor en

```

```

UNO52      MOVLW    b'00000001'      CONTA01
           MOVWF    CONTA02          ;ponemos el contador a 01
           ;guardamos valor en
           CONTA01

DOS52      MOVLW    b'01010010'      ;ponemos el contador a 8F
           MOVWF    CONTA03          ;guardamos valor en
           CONTA01

TRES52     DECFSZ   CONTA03,1        ;decrementamos conta01,
           ;si es cero salta
           GOTO     TRES52           ;vuelve
           DECFSZ   CONTA02,1        ;decrementamos conta02,
           ;si es cero salta
           GOTO     DOS52           ;volvemos a recargar
           ;conta01
           DECFSZ   CONTA01,1        ;decrementamos conta02,
           ;si es cero salta
           GOTO     UNO52           ;volvemos a recargar
           ;conta01
           NOP
           RETURN
;=====PASO DE BINARIO A BCD 16bits=====
BCD16     CLRF     DECENAMIL         ;-----
           CLRF     UNIDADMIL        ;-----
           CLRF     CENTENAS         ;-----
           CLRF     DECENAS          ;-----
           CLRF     UNIDADES         ;-----
           SWAPF    RESULTADOD_B,W   ;-----
           ANDLW    b'00001111'      ;-----
           ADDLW    b'11110000'      ;-----
           MOVWF    UNIDADMIL        ;-----
           ADDWF    UNIDADMIL,F       ;-----
           ADDLW    b'11100010'      ;-----
           MOVWF    CENTENAS         ;movwf    Centena
           ADDLW    b'00110010'      ;addlw    0x32
           MOVWF    UNIDADES         ;movwf    Unidad
           MOVF     RESULTADOD_B,W   ;movf     Binario_b,w
           ANDLW    b'00001111'      ;andlw    0x0F
           ADDWF    CENTENAS,F       ;addwf    Centena,f
           ADDWF    CENTENAS,F       ;addwf    Centena,f
           ADDWF    UNIDADES,F       ;addwf    Unidad,f
           ADDLW    b'11101001'      ;addlw    0xE9
           MOVWF    DECENAS          ;movwf    Decena
           ADDWF    DECENAS,F       ;addwf    Decena,f
           ADDWF    DECENAS,F       ;addwf    Decena,f
           SWAPF    RESULTADOD_A,W   ;swapf    Binario_a,w
           ANDLW    b'00001111'      ;andlw    0x0F

```

```

ADDWF    DECENAS,F           ;addwf    Decena,f
ADDWF    UNIDADES,F         ;addwf    Unidad,f
RLF      DECENAS,F           ;rlf      Decena,f
RLF      UNIDADES,F         ;rlf      Unidad,f
COMF     UNIDADES,F         ;comf     Unidad,f
RLF      UNIDADES,F         ;rlf      Unidad,f
MOVF     RESULTADOD_A,W     ;movf     Binario_a,w
ANDLW    b'00001111'       ;andlw    0x0F
ADDWF    UNIDADES,F         ;addwf    Unidad,f
RLF      UNIDADMIL,F        ;rlf      UnidadMil,f
MOVLW    b'00000111'       ;movlw    0x07
MOVWF    DECENAMIL         ;movwf    DecenaMil
MOVLW    b'00001010'       ;movlw    0x0A
UNI      ADDWF    UNIDADES,F ;addwf    Unidad,f
        DECF     DECENAS,F   ;decf     Decena,f
        BTSS    STATUS,C     ;btss    STATUS,C
        GOTO    UNI          ;goto    Lb1
DEC      ADDWF    DECENAS,F   ;addwf    Decena,f
        DECF     CENTENAS,F  ;decf     Centena,f
        BTSS    STATUS,C     ;btss    STATUS,C
        GOTO    DEC          ;goto    Lb2
CEN      ADDWF    CENTENAS,F  ;addwf    Centena,f
        DECF     UNIDADMIL,F  ;decf     UnidadMil,f
        BTSS    STATUS,C     ;btss    STATUS,C
        GOTO    CEN          ;goto    Lb3
UNIMIL   ADDWF    UNIDADMIL,F ;addwf    UnidadMil,f
        DECF     DECENAMIL,F  ;decf     DecenaMil,f
        BTSS    STATUS,C     ;btss    STATUS,C
        GOTO    UNIMIL       ;goto    Lb4
        MOVLW    b'00110000'  ;-----
        IORWF    UNIDADES,F   ;-----
        IORWF    DECENAS,F    ;-----
        IORWF    CENTENAS,F   ;-----
        IORWF    UNIDADMIL,F  ;-----
        IORWF    DECENAMIL,F  ;-----
;
;=====ENVIAR DATOS=====
DATTX1   BSF      STATUS,RP0  ;Enviar dato.
        BTSS    TXSTA,TRMT    ;Comprobamos si estamos
        GOTO    DATTX1        ;-----
;
        BCF     STATUS,RP0    ;Leemos los 8 bit que
        MOVLW    b'00111000'  ;-----
        MOVWF    TXREG        ;Enviamos el dato por el
    
```

```

DATTX2      BSF      STATUS,RP0      puerto serie.
            BTFS    TXSTA,TRMT      ;Enviar dato.
            GOTO    DATTX2          ;Comprobamos si estamos
;           ;                       enviando algún dato.
            BCF      STATUS,RP0      ;-----
            MOVF    DECENAMIL,0      ;-----
            MOVWF   TXREG            ;Leemos los 8 bit que
            ;                       queremos enviar
            GOTO    DATTX2          ;-----
            ;                       ;-----
            BCF      STATUS,RP0      ;Enviar dato.
            MOVF    UNIDADMIL,0      ;Enviar dato.
            MOVWF   TXREG            ;Comprobamos si estamos
            ;                       enviando algún dato.
            GOTO    DATTX3          ;-----
;           ;                       ;-----
            BCF      STATUS,RP0      ;Leemos los 8 bit que
            ;                       queremos enviar
            MOVF    UNIDADMIL,0      ;-----
            MOVWF   TXREG            ;Enviamos el dato por el
            ;                       puerto serie.
            GOTO    DATTX3          ;Enviar dato.
            ;                       ;Comprobamos si estamos
            ;                       enviando algún dato.
            BCF      STATUS,RP0      ;-----
            MOVF    CENTENAS,0      ;-----
            MOVWF   TXREG            ;Leemos los 8 bit que
            ;                       queremos enviar,
            GOTO    DATTX4          ;-----
;           ;                       ;-----
            BCF      STATUS,RP0      ;Enviar dato.
            MOVF    CENTENAS,0      ;Enviar dato.
            MOVWF   TXREG            ;Comprobamos si estamos
            ;                       enviando algún dato.
            GOTO    DATTX5          ;-----
;           ;                       ;-----
            BCF      STATUS,RP0      ;Leemos los 8 bit que
            ;                       queremos enviar
            MOVF    DECENAS,0      ;-----
            MOVWF   TXREG            ;Enviamos el dato por el
            ;                       puerto serie.
            GOTO    DATTX5          ;Enviar dato.
            ;                       ;Comprobamos si estamos
            ;                       enviando algún dato.
            BCF      STATUS,RP0      ;-----
            MOVF    DECENAS,0      ;-----
            MOVWF   TXREG            ;Leemos los 8 bit que
            ;                       queremos enviar
            GOTO    DATTX6          ;-----
;           ;                       ;-----
            BCF      STATUS,RP0      ;Enviar dato.
            BTFS    TXSTA,TRMT      ;Enviar dato.
            GOTO    DATTX6          ;Comprobamos si estamos
            ;                       enviando algún dato.
            ;                       ;-----
            ;                       ;-----

```



```

BCF STATUS,RP0 ;Leemos los 8 bit que
                queremos enviar
MOVF UNIDADES,0 ;-----
MOVWF TXREG ;Enviamos el dato por el
             puerto serie.
DATTX7 BSF STATUS,RP0 ;Enviar dato.
        BTFSS TXSTA,TRMT ;Comprobamos si estamos
                enviando algún dato.
        GOTO DATTX7 ;-----
; ;-----
BCF STATUS,RP0 ;Leemos los 8 bit que
                queremos enviar
                ;-----
MOV LW b'00111000' ;Enviamos el dato por el
MOVWF TXREG ; puerto serie.
DATTX8 BSF STATUS,RP0 ;Enviar dato.
        BTFSS TXSTA,TRMT ;Comprobamos si estamos
                enviando algún dato.
        GOTO DATTX8 ;-----
        BCF STATUS,RP0 ;-----
        GOTO VOLVER ;-----
; ;-----
VOLVER BCF PCLATH,3 ;Página 0
        BCF PCLATH,4 ;-----
        RETURN ;-----
; ;-----
PRUEBA MOV LW b'10010100' ;posición cursor dirección
                14
        CALL COMANDO ;-----
        MOVF TXDATO1,W ;-----
        MOVWF BINARIO ;-----
        CALL BIT8 ;-----
        MOV LW '' ;-----
        CALL LETRA ;1
        MOVF TXDATO2,W ;-----
        MOVWF BINARIO ;-----
        CALL BIT8 ;-----
        MOV LW b'11010100' ;posición cursor dirección
                54
        CALL COMANDO ;-----
        MOVF TXDATO3,W ;-----
        MOVWF BINARIO ;-----
        CALL BIT8 ;-----
        GOTO VOLVER ;-----
;-----
;=====ESCRIBIR VALORES BINARIOS=====
;-----

```

BIT8	btfsC	BINARIO,7	;-----
	GOTO	REUNO8	;-----
	GOTO	RECER8	;-----
REUNO8	MOVLW	b'00110001'	;-----
	CALL	LETRA	;-----
	GOTO	BIT7	;-----
RECER8	MOVLW	b'00110000'	;-----
	CALL	LETRA	;-----
BIT7	btfsC	BINARIO,6	;-----
	GOTO	REUNO7	;-----
	GOTO	RECER7	;-----
REUNO7	MOVLW	b'00110001'	;-----
	CALL	LETRA	;-----
	GOTO	BIT6	;-----
RECER7	MOVLW	b'00110000'	;-----
	CALL	LETRA	;-----
BIT6	btfsC	BINARIO,5	;-----
	GOTO	REUNO6	;-----
	GOTO	RECER6	;-----
REUNO6	MOVLW	b'00110001'	;-----
	CALL	LETRA	;-----
	GOTO	BIT5	;-----
RECER6	MOVLW	b'00110000'	;-----
	CALL	LETRA	;-----
BIT5	btfsC	BINARIO,4	;-----
	GOTO	REUNO5	;-----
	GOTO	RECER5	;-----
REUNO5	MOVLW	b'00110001'	;-----
	CALL	LETRA	;-----
	GOTO	BIT4	;-----
RECER5	MOVLW	b'00110000'	;-----
	CALL	LETRA	;-----
BIT4	btfsC	BINARIO,3	;-----
	GOTO	REUNO4	;-----
	GOTO	RECER4	;-----
REUNO4	MOVLW	b'00110001'	;-----
	CALL	LETRA	;-----
	GOTO	BIT3	;-----
RECER4	MOVLW	b'00110000'	;-----
	CALL	LETRA	;-----
BIT3	btfsC	BINARIO,2	;-----
	GOTO	REUNO3	;-----
	GOTO	RECER3	;-----
REUNO3	MOVLW	b'00110001'	;-----
	CALL	LETRA	;-----

```

RECE3      GOTO    BIT2                ;-----
            MOVLW  b'00110000'        ;-----
            CALL   LETRA               ;-----
BIT2       btfsC  BINARIO,1           ;-----
            GOTO   REUNO2              ;-----
            GOTO   RECE2               ;-----
REUNO2     MOVLW  b'00110001'        ;-----
            CALL   LETRA               ;-----
            GOTO   BIT1                ;-----
RECE2     MOVLW  b'00110000'        ;-----
            CALL   LETRA               ;-----
BIT1      btfsC  BINARIO,0           ;-----
            GOTO   REUNO1              ;-----
            GOTO   RECE1               ;-----
REUNO1     MOVLW  b'00110001'        ;-----
            CALL   LETRA               ;-----
            GOTO   BIT0                ;-----
RECE1     MOVLW  b'00110000'        ;-----
            CALL   LETRA               ;-----
BIT0      RETURN                      ;-----
            ;-----
            END                        ;-----
    
```

Capítulo VII

Conclusiones

Sección I

CONCLUSIONES GENERALES

Una vez finalizado este trabajo se pueden obtener numerosas conclusiones puesto que se ha cumplido en su totalidad con el objetivo principal planteado al inicio del mismo. Este objetivo consistía en poder controlar los movimientos de los dos ejes que componen la muñeca del robot mediante el diseño e implementación de un sistema de control que trabajase paralelamente con el esquema de control de la etapa original. Además, se pretendió gestionar todas las órdenes de control desde una interfaz gráfica de ordenador programada en Labview. A continuación se detalla cada uno de los objetivos parciales conseguidos.

En primer lugar, todo los fallos iniciales de los elementos de la etapa de potencia y de la muñeca del robot fueron corregidos y todo el material puesto a punto para permitir el diseño y la implementación del nuevo sistema de control. El equipo funciona ahora perfectamente, incluyendo también partes del robot antiguo que no se trataron durante la realización de este trabajo. A su vez, la muñeca del robot fue reubicada empotrándola sobre la mesa, lo que asegura un manejo seguro de la misma, sin posibilidades de accidentes derivados de caídas al suelo de la estructura.

El diseño e implementación del sistema de control a través del microcontrolador para el gobierno de los ejes de la muñeca del robot se realizó en su totalidad. Se solucionaron todos los problemas que surgieron en el proceso de implementar físicamente el sistema, obteniendo un control con más prestaciones y posibilidades de ampliación mediante la incorporación de placas de control auxiliares. El sistema diseñado es confiable y seguro para el usuario a la vez que cumple con todas las exigencias marcadas como objetivo al inicio de este trabajo. De la misma forma, se cumplió con el objetivo de unir el control electrónico original de la etapa de potencia con el nuevo sistema, creando un esquema muy completo que aprovecha las ventajas de ambos sistemas de control, el de la etapa de potencia original y el nuevo. Como resultado de la implementación del sistema de control las magnitudes de posición y velocidad de los ejes de la muñeca son ahora parámetros físicos que se gobiernan a voluntad. A su vez, se diseñó un sistema de control del encendido de la etapa de potencia que permite maniobrar la etapa con total seguridad a la vez que le ahorra un estrés innecesario al sistema proveniente de continuos procesos de encendido y apagado.

La interconexión física entre el sistema de control diseñado y la muñeca del robot funciona perfectamente. Todos los fallos eléctricos generados durante este proceso se solucionaron y el funcionamiento del sistema se produce continuamente sin fallos de reseteo del microcontrolador ni pérdidas de comunicación entre el ordenador y el sistema de control. Sin embargo, el sistema se maniobra teniendo siempre a mano la posibilidad de cortar la alimentación de la etapa de potencia. Los encoder, elementos que presentaron bastantes fallos, permiten ahora el conteo de sus pulsos al solucionarse problemas de ruidos en la señal. En definitiva, los sistemas de potencia y control funcionan paralelamente, comunicándose e influyendo el uno en el otro sin que se produzcan fallos de comunicación, permitiendo un control continuo y seguro de los movimientos de los ejes.

En cuanto a la programación de las maniobras automáticas se programaron dos maniobras que demuestran el nivel de control que se tiene de la muñeca exhibiendo los movimientos que podría realizar en un ambiente industrial. Sin embargo, no llegó a implementarse el sistema de programación de las maniobras por conteo de pulsos descrito en el Capítulo V. Las maniobras se realizan por tiempos: el microcontrolador recibe la orden de realizar un movimiento para después entrar en un bucle del que sale al finalizar un retardo de tiempo. La maniobra automática se programa como la concatenación de varios de estos procesos. Cabe resaltar que todos los elementos necesarios para la programación de las maniobras automáticas por conteo de pulsos se encuentran disponibles y que fue la falta de tiempo ocasionada por fallos inesperados en otras fases del trabajo lo que ocasionó que las maniobras no se programaran así.

Finalmente, el objetivo de programar una interfaz gráfica de control se cumplió a la finalización del proyecto, puesto que se debieron de realizar continuas modificaciones durante la implementación del sistema de control. La interfaz creada es sumamente intuitiva y permite el manejo del robot a través del envío de comandos numéricos al microcontrolador. Con la programación de esta interfaz gráfica se cumplen todos los objetivos planteados al inicio de este trabajo. El sistema de control diseñado se amolda totalmente a las necesidades de la etapa de potencia y de la muñeca del robot, permitiendo ejercer un control total y seguro para el usuario sobre los ejes, que involucra un control de posición, velocidad y del encendido de la etapa de potencia desde la comodidad de una interfaz gráfica de ordenador.

Sección II

POSIBLES LÍNEAS DE CONTINUACIÓN

Debido a que la muñeca del robot va a continuar siendo material de prácticas del laboratorio de tecnología electrónica se describen a continuación cuatro posibles líneas de continuación del trabajo realizado que podrían constituirse en trabajos independientes en un futuro.

La primera línea de continuación es la programación de las maniobras automáticas por el sistema de contaje de pulsos descrito en el Capítulo V que no llegó a implementarse en este trabajo. Al programar dichas maniobras por contaje de pulsos debe obtenerse un sistema de control más versátil, que permita una mayor interacción con el usuario. Debe prestarse especial atención al número de líneas de código puesto que aumentarán considerablemente al introducir el sistema de contaje por pulsos. Se recomienda estructurar el código en las cuatro páginas de la memoria de código del microcontrolador antes de empezar a programar.

La segunda línea de continuación radica en programar un sistema de control de los movimientos más allá de los movimientos automáticos preprogramados. Este sistema se basaría en la introducción de datos por parte del usuario de la posición y velocidad final deseadas. Una vez introducidos estos datos el sistema de control deberá gestionar diversos bucles de programación para realizar la maniobra solicitada, estando siempre habilitadas todas las interrupciones de seguridad y la posibilidad de interrupción de la maniobra sin ocasionar fallos en el control.

La tercera línea de continuación que se propone es medir el valor de la corriente que atraviesa el inducido de los motores mediante los conversores analógicos digitales del microcontrolador. Al medir esta variable del sistema se busca poder desconectar los motores al través del sistema de control diseñado en caso de sobrecarga.

La cuarta línea de continuación sería la de conectar un teclado hexadecimal a la placa principal a través del buffer bidireccional 74LS 245. Se busca de esta forma poder controlar el robot prescindiendo del ordenador al enviar órdenes directamente desde el teclado hexadecimal.

Bibliografía

Para la realización de este trabajo se consultaron los siguientes enlaces web:

Encoder

- <http://facultad.bayamon.inter.edu/arincon/encoderIncrementales.pdf>

Transformador

- http://www.profesormolina.com.ar/electronica/componentes/bobinas_trafos/trafos_trif.htm

Sensor inductivo

- <http://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/Sensores%20magneticos.pdf>
- http://galia.fc.uaslp.mx/~cantocar/automatas/PRESENTACIONES_PLC_PDF_S/24_SENSOR S_INDUCTIVOS.PDF
- <http://www.mes-sigma.net/Cursos/images/Sensores%20Inductivos.pdf>

Salida en alta impedancia

- <https://electrouni.files.wordpress.com/2010/08/salidas-logicas-triestados.pdf>

Etapas de potencia

- http://www.virtual.unal.edu.co/cursos/ingenieria/mecatronica/docs_curso/Anexos/TUTORIALcnc/DOCUMENTOS/TEORIA/ROBOTICA%20INDUSTRIAL.pdf

Tacómetro

- <http://es.slideshare.net/AndresRodriguez18/tacometro>

Sensor de corriente

- http://antares.itmorelia.edu.mx/~talfaro/Materias/Instrumentacion_Virtual/Sensores%20de%20Corriente%20Electrica.pdf
- http://e-archivo.uc3m.es/bitstream/handle/10016/12543/Tesis_Monica_Victoria_Rojas_Moreno.pdf?sequence=1
- <http://electronica.uc3m.es/groble/publicaciones/TID/tinvesrog.pdf>

Módulo LCD

- [file:///C:/Users/valer_000/Downloads/lcd_alfa%20\(1\).pdf](file:///C:/Users/valer_000/Downloads/lcd_alfa%20(1).pdf)

A su vez, se consultaron los siguientes apuntes y manuales:

- Apuntes: Tecnología de producción y fabricación industrial.
- Apuntes: Electrónica Industrial (Tema 6 y Tema 4).
- Apuntes: Instrumentación Electrónica (Tema 5).
- Apuntes: Informática Industrial (Tema 2).
- Manuales eléctricos del Robot ACMA X58.

Por último, se consultaron los siguientes libros.

- Máquinas eléctricas, S. Chapman.

- Modelado dinámico y control de un robot flexible de tres grados de libertad, Andrés García Higuera, José Andrés Somolinos Sánchez, José Andrés Somolinos Sánchez.
- Electrónica de potencia, Daniel W. Hart.

Índice de figuras

ÍNDICE DE FIGURAS.

ILUSTRACIÓN 1 - MUÑECA DEL ROBOT	35
ILUSTRACIÓN 2 - MUÑECA DEL ROBOT ACMA X58	35
ILUSTRACIÓN 3 - SEMEJANZA ENTRE LAS PARTES DE UN ROBOT Y LA ANATOMÍA HUMANA	36
ILUSTRACIÓN 4 - ETIQUETA DEL LADO A CONVENIDO DE LA MUÑECA DEL ROBOT.....	37
ILUSTRACIÓN 5 - ESQUEMA CADENA CINEMÁTICA MUÑECA.....	38
ILUSTRACIÓN 6 - REDUCTOR DE EJE TIPO PLANETARIO.....	39
ILUSTRACIÓN 7 - ESQUEMA BÁSICO DE ENGRANAJES.....	39
ILUSTRACIÓN 8 - CILINDRO NEUMÁTICO DE DOBLE EFECTO	42
ILUSTRACIÓN 9 - ROTOR DE UN MOTOR ASÍNCRONO.....	44
ILUSTRACIÓN 10 - MOTOR PASO A PASO. CONFIGURACIÓN BIPOLAR.....	45
ILUSTRACIÓN 11 - MOTOR PASO A PASO. CONFIGURACIÓN UNIPOLAR.....	46
ILUSTRACIÓN 12 - MOTOR DE CORRIENTE CONTINUA CON EXCITACIÓN EN DERIVACIÓN	48
ILUSTRACIÓN 13 - RESPUESTA DE UN MOTOR DE CORRIENTE CONTINUA CON EXCITACIÓN EN DERIVACIÓN ANTE UN AUMENTO EN EL PAR DE CARGA.....	49
ILUSTRACIÓN 14 - MOTORES DE LOS EJES 5 Y 6.	50
ILUSTRACIÓN 15 - MOTORES DE LOS EJES 5 Y 6. DETALLE DE LA PARTE TRASERA.	50
ILUSTRACIÓN 16 - PLACA DE CARACTERÍSTICAS DE LOS MOTORES.	51
ILUSTRACIÓN 17 - LOCALIZACIÓN DE MOTORES DE LOS EJES 5 Y 6 (MUÑECA).....	52
ILUSTRACIÓN 18 - BOBINA EXTERNA CONECTADA AL INDUCIDO DE LOS MOTORES.	52
ILUSTRACIÓN 19 - ELECTROIMÁN FRENOS. ESQUEMA.....	53
ILUSTRACIÓN 20 - FRENOS MECÁNICOS. ESQUEMA.....	53
ILUSTRACIÓN 21 - CONTACTOR MONOFÁSICO DEL CIRCUITO DE ALIMENTACIÓN DE FRENOS.....	54
ILUSTRACIÓN 22 – TRANSFORMADOR MONOFÁSICO Y RECTIFICADOR PARA ALIMENTACIÓN DE FRENOS. CONDENSADOR DEL FILTRO CAPACITIVO.	54
ILUSTRACIÓN 23 - ESQUEMA ELÉCTRICO DE LA ALIMENTACIÓN DE LOS FRENOS.....	55
ILUSTRACIÓN 24 - TACÓMETRO DE CORRIENTES DE EDDY. ESQUEMA.....	59
ILUSTRACIÓN 25 - TACÓMETRO DE CORRIENTES DE EDDY.....	59
ILUSTRACIÓN 26 - TACÓMETRO CENTRÍFUGO. ESQUEMA.	60
ILUSTRACIÓN 27 - TACÓMETRO ÓPTICO.....	61
ILUSTRACIÓN 28 - TACÓMETRO ESTROBOSCÓPICO.	61
ILUSTRACIÓN 29 - POSICIÓN DE LOS TACÓMETROS ORIGINALES DEL ROBOT.	62
ILUSTRACIÓN 30 - ESQUEMA DE CONTROL DE LOS MOTORES DEL MUÑECA. TACÓMETRO	63
ILUSTRACIÓN 31 - CABLE DE SALIDA DE LA SEÑAL DEL TACÓMETRO.	64
ILUSTRACIÓN 32 - SENSOR DE CORRIENTE RESISTIVO.....	65
ILUSTRACIÓN 33 - ESQUEMA DEMOSTRATIVO DEL EFECTO HALL.	66
ILUSTRACIÓN 34 - CAMPO MAGNÉTICO EN EL INTERIOR DE UNA BOBINA TOROIDAL.....	67
ILUSTRACIÓN 35 - BOBINA DE ROGOWSKI CONECTADA A UN CIRCUITO INTEGRADOR.....	67
ILUSTRACIÓN 36 - MEDICIÓN DE UNA TENSIÓN INDUCIDA EN UNA BOBINA DE ROGOWSKI.....	68
ILUSTRACIÓN 37 - UBICACIÓN FÍSICA DE LOS SENSORES DE CORRIENTE.....	68
ILUSTRACIÓN 38 - DETALLE SENSOR DE CORRIENTE.	68
ILUSTRACIÓN 39 - ESQUEMA DE FUNCIONAMIENTO DEL CODIFICADOR ANGULAR DE POSICIÓN ENCODER.....	69
ILUSTRACIÓN 40 - LOCALIZACIÓN DE LOS ENCODER EN LOS EJES DEL ROBOT ORIGINAL.	71
ILUSTRACIÓN 41 - ESQUEMA DEL ENCODER.	71
ILUSTRACIÓN 42 - ENCODER DE LOS EJES DE LA MUÑECA DEL ROBOT.	71
ILUSTRACIÓN 43 - SEÑALES PRINCIPALES DEL ENCODER.....	72
ILUSTRACIÓN 44 - UBICACIÓN DE LOS ENCODER EN LOS MOTORES Y CABLE POR EL QUE VIAJAN LAS SEÑALES.	72
ILUSTRACIÓN 45 - SENSOR INDUCTIVO. ESQUEMA.....	73
ILUSTRACIÓN 46 - CIRCUITO OSCILADOR BASE DEL SENSOR INDUCTIVO.	73
ILUSTRACIÓN 47 - FUNCIONAMIENTO DE UN CIRCUITO OSCILADOR LC.....	74
ILUSTRACIÓN 48 - CIRCUITO DISPARADOR SCHMITT.....	74

ILUSTRACIÓN 49 - SEÑAL DE SALIDA DEL DISPARADOR SCHMITT.....	75
ILUSTRACIÓN 50 - FUNCIONAMIENTO COMPLETO DEL SENSOR INDUCTIVO.	75
ILUSTRACIÓN 51- DIFERENCIAS MORFOLÓGICAS ENTRE EL SENSOR INDUCTIVO BLINDADO Y EL NO BLINDADO.	76
ILUSTRACIÓN 52 - MONTAJE DEL SENSOR BLINDADO.....	76
ILUSTRACIÓN 53 - MONTAJE DEL SENSOR NO BLINDADO.....	76
ILUSTRACIÓN 54 - "EFECTO EPITELIAL" PRESENTADO EN DIVERSOS METALES.....	77
ILUSTRACIÓN 55 - CURVA DE RESPUESTA DEL SENSOR EN FUNCIÓN DE LA PROXIMIDAD.....	77
ILUSTRACIÓN 56 - MODELOS COMERCIALES DE SENSORES INDUCTIVOS.....	78
ILUSTRACIÓN 57 - UBICACIÓN REAL DE LOS SENSORES INDUCTIVOS.....	79
ILUSTRACIÓN 58 - ESQUEMA DE FUNCIONAMIENTO DEL SENSOR INDUCTIVO.....	79
ILUSTRACIÓN 59 - SENSOR INDUCTIVO REAL DEL EJE 6.....	80
ILUSTRACIÓN 60 - CIRCUITO DE UN SENSOR INDUCTIVO CON CONDENSADOR ANTIREBOTE.....	80
ILUSTRACIÓN 61 - FUENTE DE ALIMENTACIÓN EXTERNA DE LOS SENSORES INDUCTIVOS.....	81
ILUSTRACIÓN 62 - SENSOR FINAL DE CARRERA CON PLETINA DE CONTACTO.....	81
ILUSTRACIÓN 63 - MONTAJE DEL SENSOR FINAL DE CARRERA DE LA MUÑECA DEL ROBOT.....	82
ILUSTRACIÓN 64 - POSICIÓN DE LOS SENSORES FINALES DE CARRERA EN LA MUÑECA DEL ROBOT.....	82
ILUSTRACIÓN 65 - CIRCUITO DE LOS FINALES DE CARRERA CON CONDENSADORES ANTIREBOTE.....	83
ILUSTRACIÓN 66 - RANGO DE MOVIMIENTO DE 180° DEL EJE 5. DERECHA.....	83
ILUSTRACIÓN 67 - RANGO DE MOVIMIENTO DE 180° DEL EJE 5. IZQUIERDA.....	84
ILUSTRACIÓN 68 - ESQUEMA DE CONTROL ORIGINAL DE LA ETAPA DE POTENCIA DE ROBOT.....	89
ILUSTRACIÓN 69 - ESQUEMA TRANSFORMADOR MONOFÁSICO.....	92
ILUSTRACIÓN 70 - TRANSFORMADOR MONOFÁSICO REAL DEL PROYECTO.....	92
ILUSTRACIÓN 71 - CIRCUITO RECTIFICADOR MONOFÁSICO DE ONDA COMPLETA.....	93
ILUSTRACIÓN 72 - RECTIFICADOR MONOFÁSICO DE DOBLE ONDA EN PUENTE. SEMICICLOS.....	93
ILUSTRACIÓN 73 - RECTIFICADOR MONOFÁSICO DE ONDA COMPLETA. CORRIENTE EN LOS SEMICICLOS.....	94
ILUSTRACIÓN 74 - RECTIFICADOR MONOFÁSICO CON FILTRO CAPACITIVO.....	95
ILUSTRACIÓN 75 - CONTACTOR TRIFÁSICO.....	96
ILUSTRACIÓN 76 - TRANSFORMADOR TRIÁSICO DE TIPO ACORAZADO.....	97
ILUSTRACIÓN 77 - CONEXIONADO DE LOS DEVANADOS DEL TRANSFORMADOR TRIFÁSICO ESTRELLA- ESTRELLA.....	97
ILUSTRACIÓN 78 - RELACIONES ENTRE LAS TENSIONES DE LÍNEA Y FASE.....	97
ILUSTRACIÓN 79 - TRANSFORMADOR TRIFÁSICO REAL DE LA ETAPA DE POTENCIA.....	98
ILUSTRACIÓN 80 - TRANSFORMADOR TRIFÁSICO. DETALLE ENTRADA PRIMARIO.....	99
ILUSTRACIÓN 81 - TRANSFORMADOR TRIFÁSICO. DETALLE SALIDA SECUNDARIO.....	99
ILUSTRACIÓN 82 - ESQUEMA ELÉCTRICO TRANSFORMADOR TRIFÁSICO.....	100
ILUSTRACIÓN 83 - CIRCUITO RECTIFICADOR TRIFÁSICO DE ONDA COMPLETA.....	100
ILUSTRACIÓN 84 - CONDENSADORES DE LA ETAPA DE FILTRADO DE LA ALIMENTACIÓN DE LOS MOTORES.....	101
ILUSTRACIÓN 85 - CIRCUITO RESISTENCIA DE DESCARGA DEL FILTRO CAPACITIVO.....	102
ILUSTRACIÓN 86 - RESISTENCIA EN SERIE CON EL FILTRO CAPACITIVO.....	102
ILUSTRACIÓN 87 - RELÉS SRD-05VDC-SL-C PARA EL CONTROL DE LA ALIMENTACIÓN DE LA ETAPA DE POTENCIA. INTERRUPTORES DE CONTROL.....	103
ILUSTRACIÓN 88 - CIRCUITO RELÉS DE LA ETAPA DE POTENCIA.....	103
ILUSTRACIÓN 89 - RED SNUBBER A LA SALIDA DE LOS RELÉS DE POTENCIA.....	104
ILUSTRACIÓN 90 - ARMARIO DE CONTROL.....	106
ILUSTRACIÓN 91 - BARRAS DE ALIMENTACIÓN POSTERIORES DEL ARMARIO DE CONTROL.....	106
ILUSTRACIÓN 92 - OBTENCIÓN DE LA SEÑAL DE MODULACIÓN DE ANCHO DE PULSO.....	109
ILUSTRACIÓN 93 - PLACA TRANSFORMADORES DE IMPULSOS.....	119
ILUSTRACIÓN 94 - ESQUEMA PUENTE EN H.....	120
ILUSTRACIÓN 95 - ESTADOS BÁSICOS DEL CIRCUITO PUENTE EN H.....	120
ILUSTRACIÓN 96 - UBICACIÓN DE LOS DIODOS EN UN PUENTE EN H.....	121
ILUSTRACIÓN 97 - PLACA DE PUENTE EN H.....	121
ILUSTRACIÓN 98 - REVERSO PLACA DEL PUENTE EN H.....	122
ILUSTRACIÓN 99 - PLACA PRINCIPAL TERMINADA.....	126

ILUSTRACIÓN 100 - DISEÑO EN AUTOCAD: CARA DE COMPONENTES	127
ILUSTRACIÓN 101 - ESQUEMA A COLOR: CARA DE COMPONENTES	128
ILUSTRACIÓN 102 - DISEÑO DE LA PLACA: CARA DE PISTAS	129
ILUSTRACIÓN 103 - ESQUEMA A COLOR: CARA DE PISTAS	130
ILUSTRACIÓN 104 - FRESADO: CONFIGURACIÓN INICIAL	131
ILUSTRACIÓN 105 - FRESADO: SELECCIÓN DE TALADROS.....	132
ILUSTRACIÓN 106 - FRESADO: SELECCIÓN DE LA HERRAMIENTA	132
ILUSTRACIÓN 107 - FRESADO: CÁLCULO DE LA TRAYECTORIA.....	133
ILUSTRACIÓN 108 - FRESADO: AVISO DE EMERGENCIA.....	133
ILUSTRACIÓN 109 - FRESADO: SIMULACIÓN DEL RESULTADO FINAL.....	133
ILUSTRACIÓN 110 - FRESADO: GUARDAMOS EL ARCHIVO .NC DEL VCARVE.....	134
ILUSTRACIÓN 111 - FRESADO: ELECCIÓN DE LA HERRAMIENTA	134
ILUSTRACIÓN 112 - FRESADO: MEDICIÓN DE LA ALTURA	135
ILUSTRACIÓN 113 - FRESADO: VENTANA INDICADORA DEL PROCESO.....	135
ILUSTRACIÓN 114 - FRESADO: HERRAMIENTA DEL CORTE	135
ILUSTRACIÓN 115 - FRESADO: CARACTERÍSTICAS DEL CORTE.....	136
ILUSTRACIÓN 116 - FRESADO: ARCHIVO DE CORTE .NC	136
ILUSTRACIÓN 117 - FRESADO: DATOS DE COORDENADAS	137
ILUSTRACIÓN 118 - FRESADO: LÍNEAS A SERIGRAFIAR	137
ILUSTRACIÓN 119 - FRESADO: HERRAMIENTA DE SERIGRAFÍA.....	138
ILUSTRACIÓN 120 - FRESADO: CONFIGURACIÓN DEL SERIGRAFIADO.....	138
ILUSTRACIÓN 121 - FRESADO: MÁQUINA CNC	139
ILUSTRACIÓN 122 - FRESADO: AGUJEROS AUXILIARES.....	139
ILUSTRACIÓN 123 - FRESADO: CONFIGURACIÓN TALADROS AUXILIARES.....	140
ILUSTRACIÓN 124 - FRESADO: PLACA TERMINADA	140
ILUSTRACIÓN 125 - PROGRAMADOR DE MICROCHIP PARA DISPOSITIVOS DE HASTA CUARENTA TERMINALES.	232
ILUSTRACIÓN 126 - TABLA DE VERDAD DEL DECODIFICADOR 74LS138	232
ILUSTRACIÓN 127 - DIAGRAMA LÓGICO DEL DECODIFICADOR 74LS138	233
ILUSTRACIÓN 128 - PATILLAJE DEL 40106	234
ILUSTRACIÓN 129 - PATILLAJE DEL 74LS373.....	234
ILUSTRACIÓN 130 - TABLA DE VERDAD 74LS373	235
ILUSTRACIÓN 131 - DIAGRAMA LÓGICO 74LS373	235
ILUSTRACIÓN 132 - NUMERACIÓN DE LAS BASCULAS 74LS138 DE LA PLACA PRINCIPAL.....	236
ILUSTRACIÓN 133 - DIAGRAMA DEL CONEXIONADO DEL 74LS245	237
ILUSTRACIÓN 134 - TABLA DE VERDAD DEL 74LS245	238
ILUSTRACIÓN 135 - DIAGRAMA LÓGICO DEL 74LS245.....	238
ILUSTRACIÓN 136 - MÉTODO DE INTERRUPCIONES	240
ILUSTRACIÓN 137 - MÉTODO DE SONDEO	240
ILUSTRACIÓN 138 - ESQUEMA GENERAL DE CONEXIÓN DEL 8259 DENTRO DE UN SISTEMA.....	241
ILUSTRACIÓN 139 - DIAGRAMA DETALLADO DE LOS TERMINALES DEL 8259.....	243
ILUSTRACIÓN 140 - DIAGRAMA DE BLOQUES DEL 8259.....	244
ILUSTRACIÓN 141 - ACOPLAMIENTO EN CASCADA DE DOS PIC ESCLAVOS Y UN MAESTRO	245
ILUSTRACIÓN 142 - PALABRA DE INICIACIÓN ICW1.....	249
ILUSTRACIÓN 143 - PALABRA DE INICIACIÓN ICW2.....	250
ILUSTRACIÓN 144 - PALABRA DE INICIACIÓN ICW3.....	251
ILUSTRACIÓN 145 - PALABRA DE COMANDO OCW1.....	252
ILUSTRACIÓN 146 - PALABRA DE COMANDO OCW2	253
ILUSTRACIÓN 147 - PALABRA DE COMANDO =CW3	253
ILUSTRACIÓN 148 - SECUENCIA DE LAS PALABRAS DE INICIACIÓN	255
ILUSTRACIÓN 149 - PALABRA OCW2 Y OCW4	256
ILUSTRACIÓN 150 - PALABRA OCW2	257
ILUSTRACIÓN 151 - PALABRA ICW4	257
ILUSTRACIÓN 152 - PALABRA OCW2	258

ILUSTRACIÓN 153 - PALABRA OCW2.....	258
ILUSTRACIÓN 154 - PALABRA OCW1.....	259
ILUSTRACIÓN 155 - PALABRA OCW3.....	259
ILUSTRACIÓN 156 - PALABRA ICW1.....	260
ILUSTRACIÓN 157 - PALABRA ICW4.....	261
ILUSTRACIÓN 158 - PALABRA ICW4.....	261
ILUSTRACIÓN 159 - ACOPLAMIENTO EN CASCADA DE DOS PIC ESCLAVOS Y UN MAESTRO.....	262
ILUSTRACIÓN 160 - PALABRAS DE COMANDO OCW1 Y OCW3.....	263
ILUSTRACIÓN 161 - LOCALIZACIÓN DE LOS DOS 8259 EN LA PLACA PRINCIPAL.....	264
ILUSTRACIÓN 162 - PINES DE UN MÓDULO LCD ESTÁNDAR.....	265
ILUSTRACIÓN 163 - VISTA TRASERA DEL DISPLAY LCD. CONTROLADOR HD44780 INCRUSTADO.....	265
ILUSTRACIÓN 164 - INSTALACIÓN DE UN POTENCIÓMETRO DE CONTRASTE PARA LA PANTALLA LCD.....	266
ILUSTRACIÓN 165 - MÓDULO LCD DEL PROYECTO.....	269
ILUSTRACIÓN 166 - PLACA 8253-8255: CARA DE COMPONENTES.....	273
ILUSTRACIÓN 167 - PLACA 8253-8255: CARA DE PISTAS.....	273
ILUSTRACIÓN 168 - PLACA 8255 - 8253 TERMINADA.....	274
ILUSTRACIÓN 169 - ESTRUCTURA INTERNA DEL 8253.....	275
ILUSTRACIÓN 170 - 8253: DIAGRAMA DETALLADO DE LOS TERMINALES DEL CIRCUITO.....	276
ILUSTRACIÓN 171 - MODO 0 8253. INTERRUPCIÓN AL FINAL DEL CONTAJE.....	279
ILUSTRACIÓN 172 - MODO 1 8253 MONOESTABLE PROGRAMABLE.....	280
ILUSTRACIÓN 173 - MODO 2 8253 DIVISOR DE FRECUENCIA.....	280
ILUSTRACIÓN 174 - MODO 3. 8253 GENERADOR DE ONDA CUADRADA.....	281
ILUSTRACIÓN 175 - MODO 4 8253 RETARDO ACTIVADO POR SOFTWARE.....	282
ILUSTRACIÓN 176 - MODO 5 8253 RETARDO ACTIVADO POR HARDWARE.....	282
ILUSTRACIÓN 177 - NUMERACIÓN DE LOS CIRCUITOS 8253 DE LA PLACA 8255 - 8253.....	284
ILUSTRACIÓN 178 - 8255: LÍNEAS DE CONTROL.....	285
ILUSTRACIÓN 179- ESTRUCTURA INTERNA DEL 8255.....	286
ILUSTRACIÓN 180 - DIAGRAMA DETALLADO DEL CONEXIONADO DEL 8255, LA LETRA "E" INDICA ENTRADAS AL CIRCUITO Y LA LETRA "S" SALIDAS DEL CIRCUITO.....	287
ILUSTRACIÓN 181 - PALABRA DE PROGRAMACIÓN DEL CIRCUITO 8255.....	289
ILUSTRACIÓN 182 - PALABRA PARA ASIGNAR VALORES DE CONTROL A LOS BITS DEL PUERTO C. EN LOS MODO 1 Y 2 CON ESTA CONFIGURACIÓN SE PUEDE PONER A 1 Ó 0 CIERTOS FLIP FLOPS QUE PERMITEN O PROHÍBEN LA GENERACIÓN DE PETICIONES DE INTERRUPCIÓN.....	289
ILUSTRACIÓN 183 - COMPORTAMIENTO DE LAS LÍNEAS DE LOS DIFERENTES PUERTOS CUANDO FUNCIONAN EN MODO 0.....	290
ILUSTRACIÓN 184 - DIAGRAMAS DE TIEMPOS PARA EL FUNCIONAMIENTO EN MODO 0 CUANDO SE ESTA PRODUCIENDO UNA LECTURA DE DATOS.....	290
ILUSTRACIÓN 185 - DIAGRAMAS DE TIEMPOS PARA EL FUNCIONAMIENTO EN MODO 0 CUANDO SE ESTA PRODUCIENDO UNA ESCRITURA DE DATOS.....	290
ILUSTRACIÓN 186 - CONFIGURACIÓN EN MODO 1 CON EL PUERTO A Y B EN ENTRADA.....	291
ILUSTRACIÓN 187 - CONFIGURACIÓN EN MODO 1 CON EL PUERTO A EN ENTRADA Y EL PUERTO B EN SALIDA.....	291
ILUSTRACIÓN 188 - DIAGRAMA DE TIEMPOS CUANDO EL PUERTO FUNCIONA EN MODO COMUNICACIONES Y COMO ENTRADA.....	292
ILUSTRACIÓN 189 - CONFIGURACIÓN EN MODO 1 CON EL PUERTO A EN SALIDA Y EL PUERTO B EN ENTRADA.....	293
ILUSTRACIÓN 190 - CONFIGURACIÓN EN MODO 1 CON EL PUERTO A Y B EN SALIDA.....	293
ILUSTRACIÓN 191 - DIAGRAMA DE TIEMPOS CUANDO EL PUERTO FUNCIONA EN MODO COMUNICACIONES Y COMO SALIDA.....	293
ILUSTRACIÓN 192 - CONFIGURACIÓN EN MODO 2.....	296
ILUSTRACIÓN 193 - DIAGRAMA DE TIEMPOS CUANDO EL PUERTO FUNCIONA EN MODO COMUNICACIONES BIDIRECCIONALES.....	297
ILUSTRACIÓN 194 - MODO COMUNICACIONES UNIDIRECCIONAL.....	297
ILUSTRACIÓN 195 - MODO COMUNICACIONES BIDIRECCIONAL.....	298

ILUSTRACIÓN 196 - ORGANIGRAMA QUE CORRESPONDE CON EL PROGRAMA DE COMUNICACIONES DESDE UN 8255 HACIA UNA IMPRESORA. ESQUEMA DE CONEXIONES.....	300
ILUSTRACIÓN 197 - NUMERACIÓN DE LOS CIRCUITOS 8255 EN LA PLACA 8255 - 8253.....	301
ILUSTRACIÓN 198 - PLACA MOTORES. LADO DE COMPONENTES.....	303
ILUSTRACIÓN 199 - PLACA MOTORES. LADO DE PISTAS.....	303
ILUSTRACIÓN 200 - PLACA DE GESTIÓN DE MOTORES TERMINADA.....	304
ILUSTRACIÓN 201 - DAC0800 DIAGRAMA DE PINES.....	305
ILUSTRACIÓN 202 - DAC0800 DIAGRAMA DE BLOQUES.....	305
ILUSTRACIÓN 203 - DAC0800 CONFIGURACIÓN BÁSICA UNIPOLAR NEGATIVA.....	306
ILUSTRACIÓN 204 - CONFIGURACIÓN BÁSICA BIPOLAR.....	306
ILUSTRACIÓN 205 - DAC0800 CONFIGURACIÓN DE OFFSET SIMÉTRICO.....	307
ILUSTRACIÓN 206 - TL074 DIAGRAMA DE PINES.....	308
ILUSTRACIÓN 207 - CONEXIÓN DAC088 Y TL074.....	309
ILUSTRACIÓN 208 - SEÑALES DEL ENCODER SIN TRATAMIENTO.....	310
ILUSTRACIÓN 209 - PLACA DE TRATAMIENTO DE SEÑALES. LADO DE COMPONENTES.....	311
ILUSTRACIÓN 210 - PLACA DE TRATAMIENTO DE SEÑALES. LADO DE PISTAS.....	311
ILUSTRACIÓN 211 - PLACA DE TRATAMIENTO DE SEÑALES. SEÑALES DESPUÉS DEL PROCESO DE FILTRADO.....	312
ILUSTRACIÓN 212 - PLACA DE TRATAMIENTO DE SEÑALES TERMINADA.....	312
ILUSTRACIÓN 213 - TL082 DIAGRAMA DE PINES.....	313
ILUSTRACIÓN 214 - CIRCUITO BÁSICO DEL OPTOACOPLADOR CON FOTOTRANSISTOR.....	314
ILUSTRACIÓN 215 - ESQUEMA ELÉCTRICO. PLACA AND - OR.....	314
ILUSTRACIÓN 216 - PLACA AND – OR.....	315
ILUSTRACIÓN 217 – PLACA AND – OR. DETALLE CONEXIÓN SEÑALES ENCODER.....	315
ILUSTRACIÓN 218 - PATILLAJE Y CIRCUITO INTERNO DEL 4081 (IZQUIERDA) Y EL 4071 (DERECHA).....	316
ILUSTRACIÓN 219 - SEÑALES A LA SALIDA DEL ENCODER VISTAS CON OSCILOSCOPIO.....	324
ILUSTRACIÓN 220 - PRUEBAS CON LA PLACA DE TRATAMIENTO DE SEÑALES.....	325
ILUSTRACIÓN 221 - SEÑALES DE LOS ENCODER A LA SALIDA DE LA PLACA DE TRATAMIENTO.....	325
ILUSTRACIÓN 222 - ESQUEMA DE CONTAJE MEDIANTE CONTADORES 8253.....	327
ILUSTRACIÓN 223 - ESQUEMA DEL ENVÍO DE PULSOS MANUALES AL 8253.....	328
ILUSTRACIÓN 224 - MOVIMIENTOS DE LA MUÑECA DURANTE LA MANIOBRA DE BÚSQUEDA DE LA POSICIÓN INICIAL.....	330
ILUSTRACIÓN 225 - CONFIGURACIÓN DE FUNCIONAMIENTO DEL DAC0800.....	334
ILUSTRACIÓN 226 - ESQUEMA DEL SUBSISTEMA DE CONTROL DE VELOCIDAD DISEÑADO.....	335
ILUSTRACIÓN 227 - INTERRUPTORES ACCIONADORES DEL SUBSISTEMA DE CONTROL DE LA ALIMENTACIÓN DE LA ETAPA DE POTENCIA.....	337
ILUSTRACIÓN 228 - RELÉS DEL SUBSISTEMA DE CONTROL DE LA ALIMENTACIÓN DE LA ETAPA DE POTENCIA.....	337
ILUSTRACIÓN 229 - INTERFAZ GRÁFICA DISEÑADA EN LABVIEW.....	340
ILUSTRACIÓN 230 - MÓDULO DE GESTIÓN DE LA COMUNICACIÓN SERIE DE LA INTERFAZ GRÁFICA.....	341
ILUSTRACIÓN 231 - MÓDULO DE CONTROL DE LA VELOCIDAD DE LOS MOTORES. INTERFAZ GRÁFICA.....	341
ILUSTRACIÓN 232 - LECTURA Y RECARGA DE LOS REGISTROS CONTADORES. INTERFAZ GRÁFICA.....	342
ILUSTRACIÓN 233 - MÓDULO DE BOTONES DE ACCIONES ESPECÍFICAS. INTERFAZ GRÁFICA.....	343
ILUSTRACIÓN 234 - EJEMPLO DE PROGRAMACIÓN GRÁFICA EN LABVIEW.....	345
ILUSTRACIÓN 235 - FUNCIÓN CICLO "WHILE" DE LABVIEW.....	346
ILUSTRACIÓN 236 - ESTRUCTURA "CASE" DE LABVIEW.....	346
ILUSTRACIÓN 237 - FUNCIÓN "TYPE CAST" DE LABVIEW.....	346
ILUSTRACIÓN 238 - FUNCIÓN "BYTE ARRAY TO STRING" DE LABVIEW.....	346
ILUSTRACIÓN 239 - FUNCIÓN "STRING TO BYTE ARRAY" DE LABVIEW.....	346
ILUSTRACIÓN 240 - FUNCIÓN "ARRAY SUBSET" DE LABVIEW.....	347
ILUSTRACIÓN 241 - FUNCIÓN "WAIT (MS)" DE LABVIEW.....	347
ILUSTRACIÓN 242 - FUNCIÓN "VISA CONFIGURE SERIAL PORT" DE LABVIEW.....	347
ILUSTRACIÓN 243 - FUNCIÓN "VISA WRITE" DE LABVIEW.....	348
ILUSTRACIÓN 244 - FUNCIÓN "VISA READ" DE LABVIEW.....	348
ILUSTRACIÓN 245 - FUNCIÓN "BYTES AT PORT" DE LABVIEW.....	348

ILUSTRACIÓN 246 - EJEMPLO DE PROGRAMACIÓN DE UNA ESTRUCTURA COMPLETA DE LECTURA Y ESCRITURA EN EL BUFFER DE DATOS.	349
ILUSTRACIÓN 247 - PROGRAMACIÓN DE LA ESTRUCTURA DE CONTROL DE LA PANTALLA INFORMATIVA.	350
ILUSTRACIÓN 248 - NIVELES LÓGICOS DE TENSIÓN DE LA INTERFACE RS - 232.	351
ILUSTRACIÓN 249 - PINES DEL CONECTOR DB9 DE LA COMUNICACIÓN SERIE.	351
ILUSTRACIÓN 250 - NIVELES LÓGICOS DE TENSIÓN DE LA TECNOLOGÍA TTL.	352
ILUSTRACIÓN 251 - CONEXIÓN DEL CIRCUITO MAX 232 CON EL PUERTO SERIE DEL ORDENADOR.	353
ILUSTRACIÓN 252 - CONEXIÓN TIPO XON - XOFF ENTRE EL ORDENADOR Y EL MICROCONTROLADOR.	353
ILUSTRACIÓN 253 - CIRCUITO MAX 232 IMPLEMENTADO EN LA REALIDAD.	354
TABLA 1 - CABLES DE ALIMENTACIÓN DE POTENCIA DE LOS MOTORES Y DE LOS FRENOS.	50
TABLA 2 - CARACTERÍSTICAS SENSORES INDUCTIVOS DE LA MUÑECA DEL ROBOT.	79
TABLA 3 - TABLE COMPARATIVA PIC16F877 / PIC16F877A.	145
TABLA 4 - CONEXIONES DEL MICROCONTROLADOR EN LA PLACA PRINCIPAL.	231
TABLA 5 - ICW 8259 DISPOSITIVO MAESTRO Y DISPOSITIVO ESCLAVO.	246
TABLA 6 - CÓDIGO DE INSTRUCCIÓN CALL.	247
TABLA 7 - BYTE MENOS SIGNIFICATIVO DE LA DIRECCIÓN DE INTERRUPCIÓN.	248
TABLA 8 - BYTE MÁS SIGNIFICATIVO DE LA DIRECCIÓN DE INTERRUPCIÓN.	248
TABLA 9 - PRIMER BYTE QUE ENVÍA EL 8259 AL PRIMER PULSO INTA.	250
TABLA 10 - BYTE MENOS SIGNIFICATIVO DE LA DIRECCIÓN DE LA INTERRUPCIÓN.	250
TABLA 11 - BYTE MÁS SIGNIFICATIVO DE LA DIRECCIÓN DE LA INTERRUPCIÓN.	250
TABLA 12 - BYTE MENOS SIGNIFICATIVO DE LA DIRECCIÓN DE LA INTERRUPCIÓN.	250
TABLA 13 - PALABRAS DE COMANDO.	252
TABLA 14 - PALABRAS DE INICIACIÓN ICW Y DE COMANDO OCW.	254
TABLA 15 - NIVELES DE ROTACIÓN CON IGUAL PRIORIDAD.	258
TABLA 16 - COMANDO SONDEO.	260
TABLA 17 - SEÑALES DE CONTROL DE LOS 8253.	276
TABLA 18 - FORMATO DE LA PALABRA DE PROGRAMACIÓN.	277
TABLA 19 - 8253: SELECCIÓN DE MODOS DE FUNCIONAMIENTO.	278
TABLA 20 - 8253: SELECCIÓN DEL MODO DE ESCRITURA EN LOS CONTADORES.	278
TABLA 21 - 8253: MODO DE FUNCIONAMIENTO DE LOS CONTADORES.	278
TABLA 22 - 8255. CONEXIÓN CON UNA IMPRESORA.	299
TABLA 23 - FUNCIÓN "VISA CLOSE" DE LABVIEW.	348
TABLA 24 - INTERFACE RS-232 CARACTERÍSTICAS FUNCIONALES.	352

PLANOS

ÍNDICE DE PLANOS.

Plano N° 1.....	439
Plano N° 2.....	441
Plano N° 3.....	443
Plano N° 4.....	445
Plano N° 5.....	447
Plano N° 6.....	449
Plano N° 7.....	451
Plano N° 8.....	453
Plano N° 9.....	455
Plano N° 10.....	457
Plano N° 11.....	459
Plano N° 12.....	461
Plano N° 13.....	463
Plano N° 14.....	465
Plano N° 15.....	467

PLANO N° I

PLANO N° 2

PLANO N° 3

PLANO N° 4

PLANO N° 5

PLANO N° 6

PLANO N° 7

PLANO N° 8

PLANO N° 9

PLANO N° 10

PLANO N° II

PLANO N° 12

PLANO N° 13

PLANO N° 14

PLANO N° 15

PRESUPUESTO

El presupuesto detalla los precios de los componentes de todas las placas que componen el sistema de control diseñado. Se incluyen los precios de los componentes en el portal español de venta de componentes electrónicos RS y los precios de venta del portal estadounidense EBay. De esta forma se realiza una comparación entre ambos precios, teniendo en cuenta que normalmente el plazo de entrega de la mercancía al comprar por EBay es mucho mayor.

CAPÍTULO I. COMPONENTES DE LA PLACA PRINCIPAL.						
COMPONENTE	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO RS	PRECIO TOTAL RS	PRECIO UNITARIO EBAY	PRECIO TOTAL EBAY
		(Unidad de producto)	(€)	(€)	(€)	(€)
PIC16F877A	Microcontrolador de la familia de los PIC. Modelo de 40 terminales.	1	5,62	5,62	3,25	3,25
74LS138	Decodificador de tres entradas y ocho salidas.	1	0,62	0,62	1,95	1,95
74LS373	Báscula unidireccional. Activo por bajos.	5	0,67	3,35	0,51	2,55
74LS245	Buffer bidireccional de datos. Activo por altos.	1	0,74	0,74	1,30	1,30
8259	Controlador programable de interrupciones.	2	5,67*	11,34	8,50	17,00
40106	Circuito inversor (incluye seis puertas NOT).	8	0,57	4,57	0,92	7,35

Resistencias de 10K		4	0,015	0,06	0,16	0,64
Resistencias de 2K2		4	0,021	0,084	0,031	0,13
Resistencias de 5K6		8	0,092	0,736	0,035	0,28
Condensadores de 0.1 uF		6	0,114	0,684	0,51	3,06
			<u>SUBTOTAL</u> (€)	27,804	<u>SUBTOTAL</u> (€)	37,51

*Este precio no pudo encontrarse en RS puesto que el circuito está descontinuado. El precio mostrado es de la página web de compras online AliExpress.

CAPÍTULO II. COMPONENTES DE LA PLACA 8255 – 8253.						
COMPONENTE	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO RS	PRECIO TOTAL RS	PRECIO UNITARIO EBAY	PRECIO TOTAL EBAY
		(Unidad de producto)	(€)	(€)	(€)	(€)
8255	Periférico programable de entradas – salidas.	2	1,62**	3,24	2,50	5,00
8253	Periférico de temporizadores – contadores.	4	6,61	26,44	3,50	14,00
			SUBTOTAL (€)	29,68	SUBTOTAL (€)	19,00

**Este precio no pudo encontrarse en RS puesto que el circuito está descontinuado. El precio mostrado es de la página web de compras online AliExpress.

CAPÍTULO III. COMPONENTES DE LA PLACA DE GESTIÓN DE MOTORES.

COMPONENTE	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO RS	PRECIO TOTAL RS	PRECIO UNITARIO EBAY	PRECIO TOTAL EBAY
		(Unidad de producto)	(€)	(€)	(€)	(€)
DAC0800	Convertor digital – analógico.	2	1,37	2,74	1,77	3,55
TL074	Circuito amplificador.	2	0,40	0,80	1,54	3,07
Resistencias de 5K6		8	0,092	0,736	0,035	0,28
Condensadores de 0.1 uF		6	0,114	0,684	0,51	3,06
			<u>SUBTOTAL</u> (€)	4,96	<u>SUBTOTAL</u> (€)	9,96

CAPÍTULO IV. COMPONENTES DE LA PLACA DE ADAPTACIÓN DE SEÑALES.

COMPONENTE	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO RS	PRECIO TOTAL RS	PRECIO UNITARIO EBAY	PRECIO TOTAL EBAY
		(Unidad de producto)	(€)	(€)	(€)	(€)
TL072	Circuito amplificador.	12	0,39	4,68	1,45	17,4
4N25	Optoacoplador.	12	0,34	4,08	1,10	13,1
40106	Circuito inversor.	12	0,40	4,80	1,43	17,16
			SUBTOTAL (€)	1,13	SUBTOTAL (€)	3,98

CAPÍTULO V. COMPONENTES DE LA PLACA AND - OR

COMPONENTE	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO RS	PRECIO TOTAL RS	PRECIO UNITARIO EBAY	PRECIO TOTAL EBAY
		(Unidad de producto)	(€)	(€)	(€)	(€)
4081	Circuito que contiene cuatro puertas lógicas OR.	2	3,70	7,40	1,08	2,16
4071	Circuito que contiene cuatro puertas lógicas AND.	2	0,75	1,50	0,51	1,02
			SUBTOTAL (€)	8,90	SUBTOTAL (€)	3,18

<u>GRAN TOTAL PRECIOS EBAY (€)</u>	<u>73,63</u>
<u>GRAN TOTAL PRECIOS RS (€)</u>	<u>69,74</u>

El gran total del presupuesto realizado del portal EBay es de **73,63 €**.

El gran total del presupuesto con precios del portal RS es de **69,474 €**.

PLIEGO DE
CONDICIONES
GENERALES

1.- OBJETO.

El presente pliego de condiciones tiene por objeto regular el diseño y la implementación del sistema de control diseñado. A su vez, pretende informar al usuario de las precauciones necesarias a la hora de maniobrar con la muñeca del robot.

El proyecto ha sido realizado por Valeria Ospina Bohórquez alumna de la Escuela Técnica Superior Industrial de Béjar a petición de José Torreblanca González, profesor del departamento de Tecnología Electrónica.

2.- DOCUMENTOS.

El presente proyecto integra los siguientes documentos:

- Memoria
- Planos
- Presupuesto
- Pliego de condiciones generales

3.- NÚMERO DE COPIAS AUTORIZADAS.

El número de copias autorizadas del proyecto es de dos inicialmente. Una de la copias se guardará en la biblioteca del centro universitario Escuela Técnica Superior de Ingeniería Industrial de Béjar, mientras que la otra copia la guardará la autora del presente proyecto.

Así mismo, el tutor dispondrá de una copia y además podrá realizar tantas copias y modificaciones como considere conveniente, ya que el proyecto se ha realizado con fines didácticos de cara a las asignaturas de electrónica que imparte el área de Tecnología Electrónica.

4.- PLIEGO DE CONDICIONES GENERALES

4.1.- DISEÑO Y FABRICACIÓN DE LA PLACA.

La placa diseñada se complementa en la realidad con otras cuatro placas auxiliares. Por tanto, el sistema de control se encuentra implementado en:

- Placa principal
- Placa de gestión de motores
- Placa de adaptación de señales
- Placa AND - OR
- Placa 8255 – 8253

La placa principal se diseñó en AutoCAD procurando realizar un esquema adecuado atendiendo al ancho de las pistas, que debe ser suficiente para evitar que la pista desaparezca tras el fresado.

Los programas de asistencia para el fresado utilizados fueron Win PC y V Carve PRO. Para realizar el paso de los documentos en AutoCAD al VCarve PRO es importante que el diseño se haya realizado enteramente utilizando la función de “polilínea” de AutoCAD. De no ser así el programa VCarve PRO no detectará un contorno cerrado y producirá un fallo en la conversión.

Durante el proceso de fresado se debe velar en todo momento por la seguridad personal, evitando acercar las manos a la zona de fresado cuando la máquina esté encendida. Los movimientos que realiza la máquina al fresar no son predecibles y esta acción puede resultar en una grave herida.

Para asegurar un proceso correcto de fresado debe de anclarse la placa virgen con cuatro tornillos a la superficie de madera. Se evitará combar en exceso la placa al presionar los tornillos con demasiada fuerza, Esto produce errores en el fresado pues la superficie a fresar no se encuentra en el mismo plano horizontal. Además, se retirará el exceso de residuos para evitar problemas y poder observar con claridad el avance. Se recomienda utilizar bata pues el proceso de fresado es bastante sucio y no exponerse al ruido de la fresadora durante un tiempo prolongado.

4.2.- ETAPA DE POTENCIA.

Al realizar la conexión del sistema de control diseñado con la etapa de potencia se deben de implementar los circuitos eléctricos que sean necesarios para evitar errores de comunicación entre la etapa de potencia y el microcontrolador. Estos circuitos constan de resistencias y condensadores que evitan fenómenos de rebote de la señal.

Todos los problemas de distancia entre las conexiones se solucionarán alargando cables entre un punto y otro. No se moverá la etapa de potencia de lugar puesto que se encuentra lo más lejos posible de zonas de circulación normales del laboratorio.

Para accionar la alimentación del contactor trifásico de potencia deben de activarse los interruptores magnetotérmicos y diferenciales de 400V del laboratorio de tecnología electrónica. Cuando se encuentran activados se dispone de un sistema de alerta lumínica que indica que la etapa de potencia tiene tensión. Nadie debe acercarse al rack de potencia cuando la bombilla se encuentra encendida.

El manejo de la etapa de potencia se realiza a través de un sistema de encendido seguro basado en relés que evita al usuario el tener que accionar manualmente el contactor trifásico de potencia y la alimentación de 230V de la electrónica de control.

Existe un extintor a menos de cinco metros de la etapa de potencia disponible para usarse en caso de emergencia.

4.3.- INTERFAZ GRÁFICA.

Es de vital importancia comprobar que se tiene una correcta comunicación con el microcontrolador antes de conectar la etapa de potencia con el sistema de control. Para ello se dispone de los botones de “Código 20” y “Código 21” que deben de enviar a la pantalla

informativa de la interfaz dichos caracteres. El correcto funcionamiento de este sistema indica que el microcontrolador se comunica correctamente con la interfaz y que puede procederse con la conexión de la etapa de potencia.

4.4.- MUÑECA DEL ROBOT ACMA X58.

Para maniobrar correctamente con la muñeca del robot ésta debe de estar libre de objetos en un área de aproximadamente 1m² alrededor de la misma. Lo anterior para evitar problemas de choque de los ejes del motor.

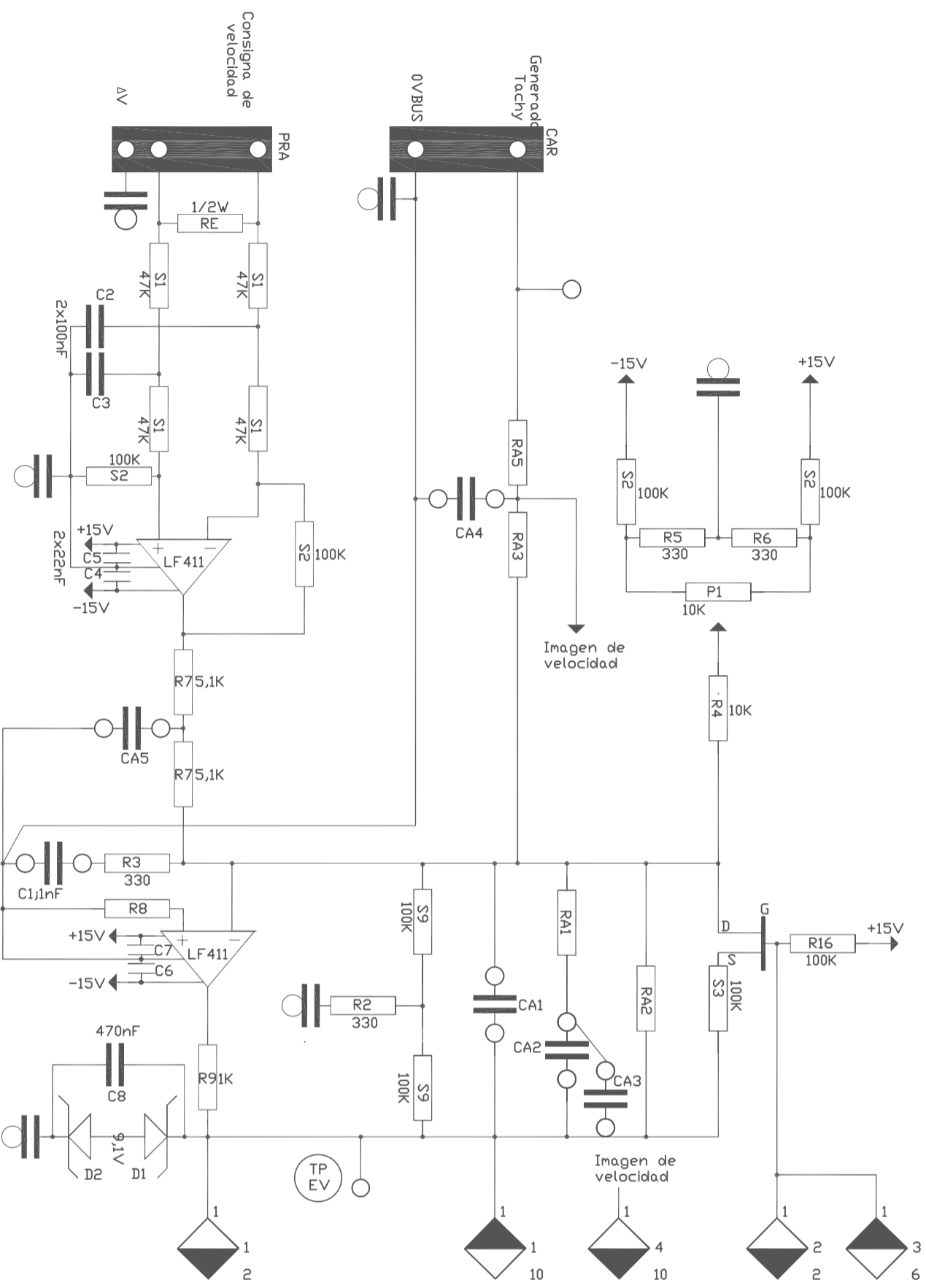
La muñeca se empotró sobre la mesa de trabajo para trabajar con ella de manera segura. De esta forma el eje 5 tiene total libertad de movimientos y se trabaja del lado de la seguridad, evitando que la muñeca impacte con la mesa.


Los sensores finales de carrera son esenciales para el funcionamiento del sistema de control. Debe de verificarse el estado de las gomas que sirven como pletinas antes de trabajar con el sistema de control.

4.5.- PASOS A SEGUIR PARA CONTROLAR LA MUÑECA DEL ROBOT DESDE LA INTERFAZ GRÁFICA.

1. Encender el diferencial y el magnetotérmico de la alimentación de 400V del laboratorio.
2. Encender el entrenador y la fuente de alimentación de los sensores inductivos.
3. Conectar la alimentación de las placas de control. Esta alimentación proviene de una fuente de alimentación de ordenador adaptada para ello.
4. Abrir el programa de la interfaz gráfica en Labview pero no ejecutarlo aún.
5. Encender el primer interruptor (el de la izquierda) que controla la alimentación de la etapa de potencia. Este interruptor alimenta a 230V la electrónica de control de la etapa de potencia. En este momento se escuchará el sonido que emiten los ventiladores del rack.
6. Esperar 5 segundos y encender el segundo interruptor que conecta el contactor trifásico. Se oirá la conmutación del contactor.
7. Ejecutar el programa de la interfaz. Comprobar que se tiene una correcta comunicación con el microcontrolador enviando los códigos 20 y 21.
8. Si se tiene una correcta comunicación encender el tercer interruptor. El sistema está listo para funcionar. En caso contrario apagar el tercer interruptor y desconectar la fuente de alimentación de la placa principal. Empezar de nuevo desde el tercer paso.

9. Cuando se desee desconectar el sistema se debe desconectar la etapa de potencia de derecha a izquierda (interruptores). Posteriormente se desconectarán todas las otras alimentaciones. Finalmente se deben de desactivar el diferencial y el magnetotérmico de la alimentación de 400V del laboratorio.



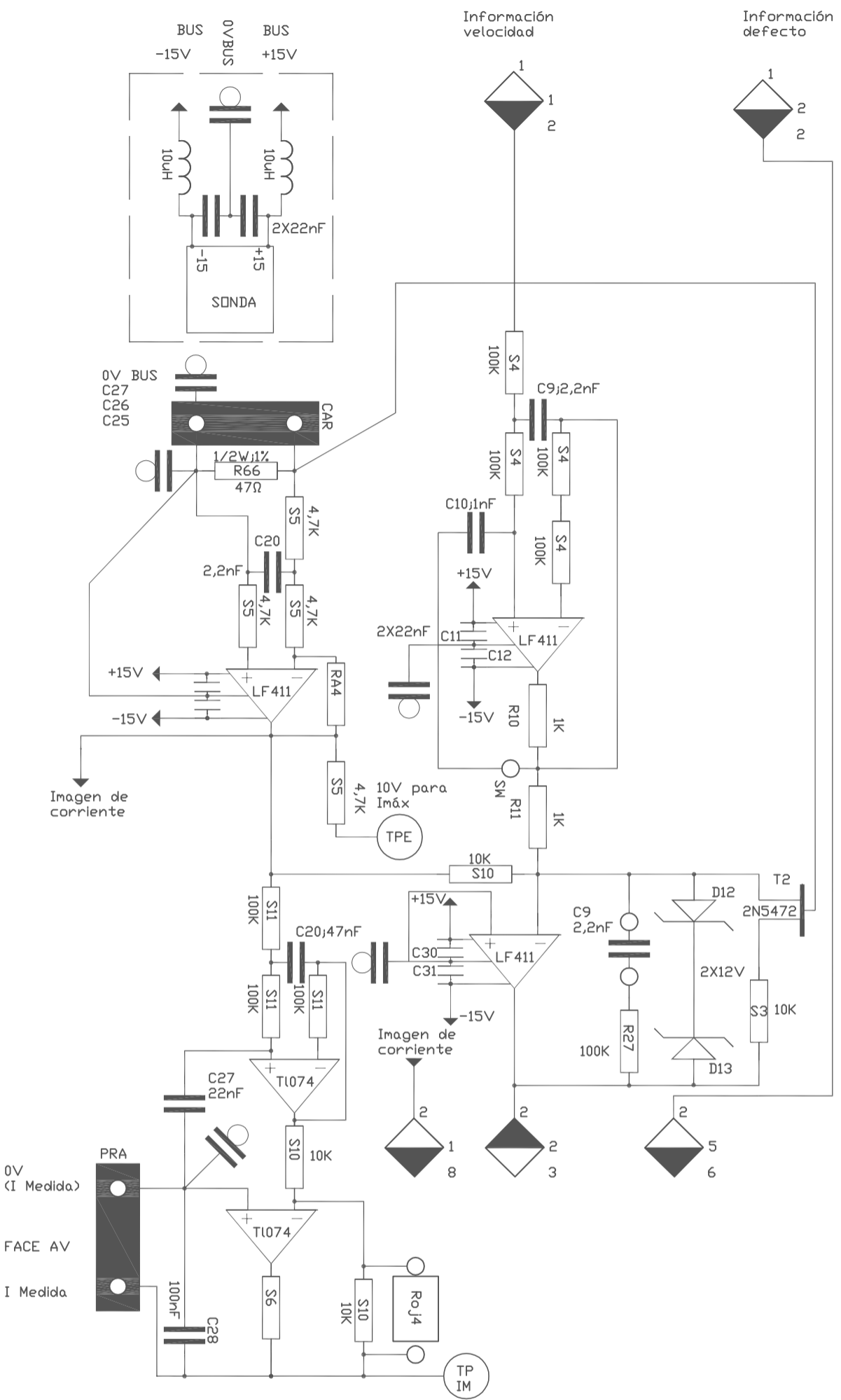
Escala	Fecha	Nombre	 Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW
Compr Obado:	01/04/15	Valeria Espina B.	
Id:snorm:	01/06/15	José Torreblanca	


Esquema eléctrico de control de la etapa de potencia

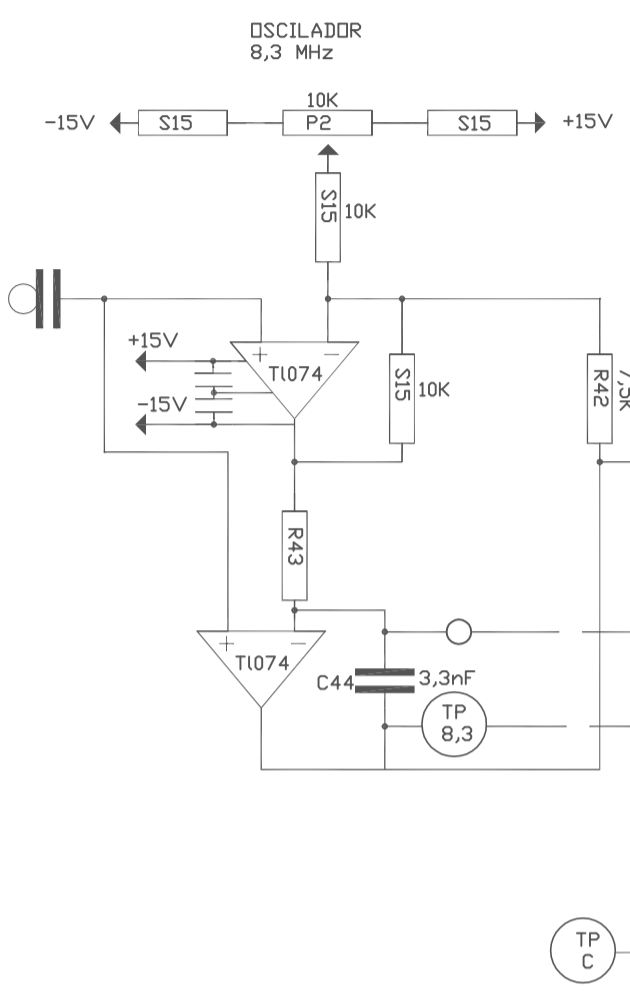
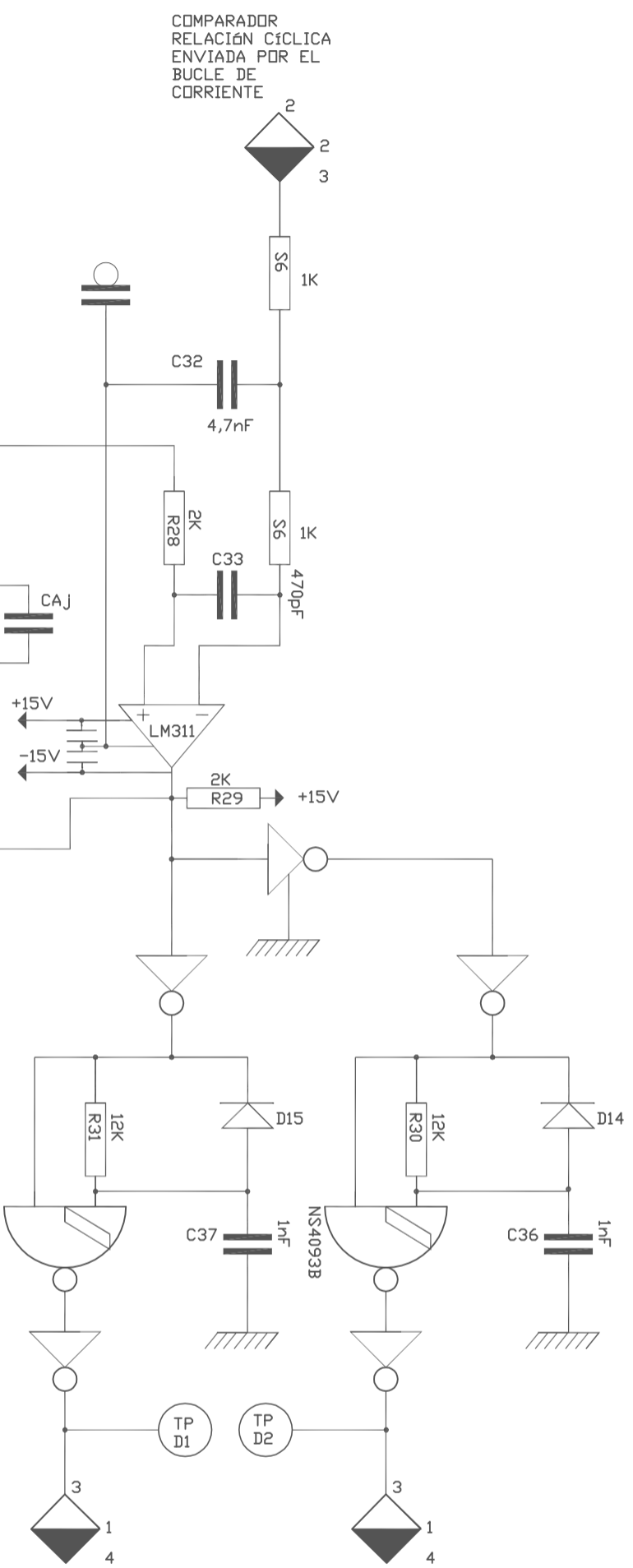
Plano N° 1


Sustituye a:

Sustituido por:



Escala	Fecha	Nombre	 Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW
Comprobado:	01/06/15	Valeria Espina B.	
Dibujado:	01/04/15	Valeria Espina B.	
Esquema eléctrico de control de la etapa de potencia			Plano N° 2
Sustituye a:			
Sustituido por:			



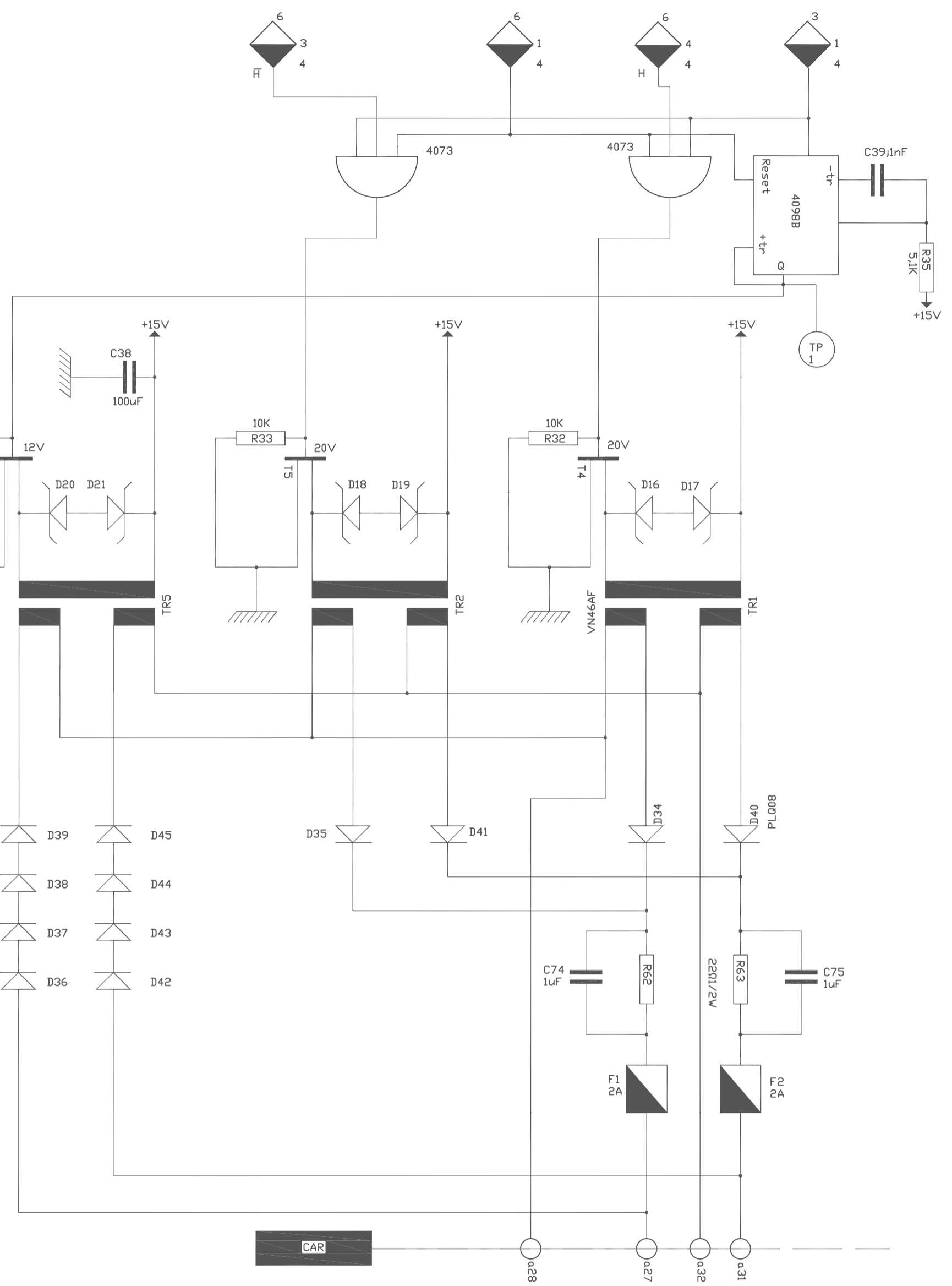
Nombre	Fecha	 Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW
Nombre	Nombre	
Dibujado:	01/04/15	
Comprobado:	01/06/15	José Torreblanca
Idioma:		
Escala		

Esquema eléctrico de control de la etapa de potencia

Plano N° 3

Sustituye a:

Sustituido por:

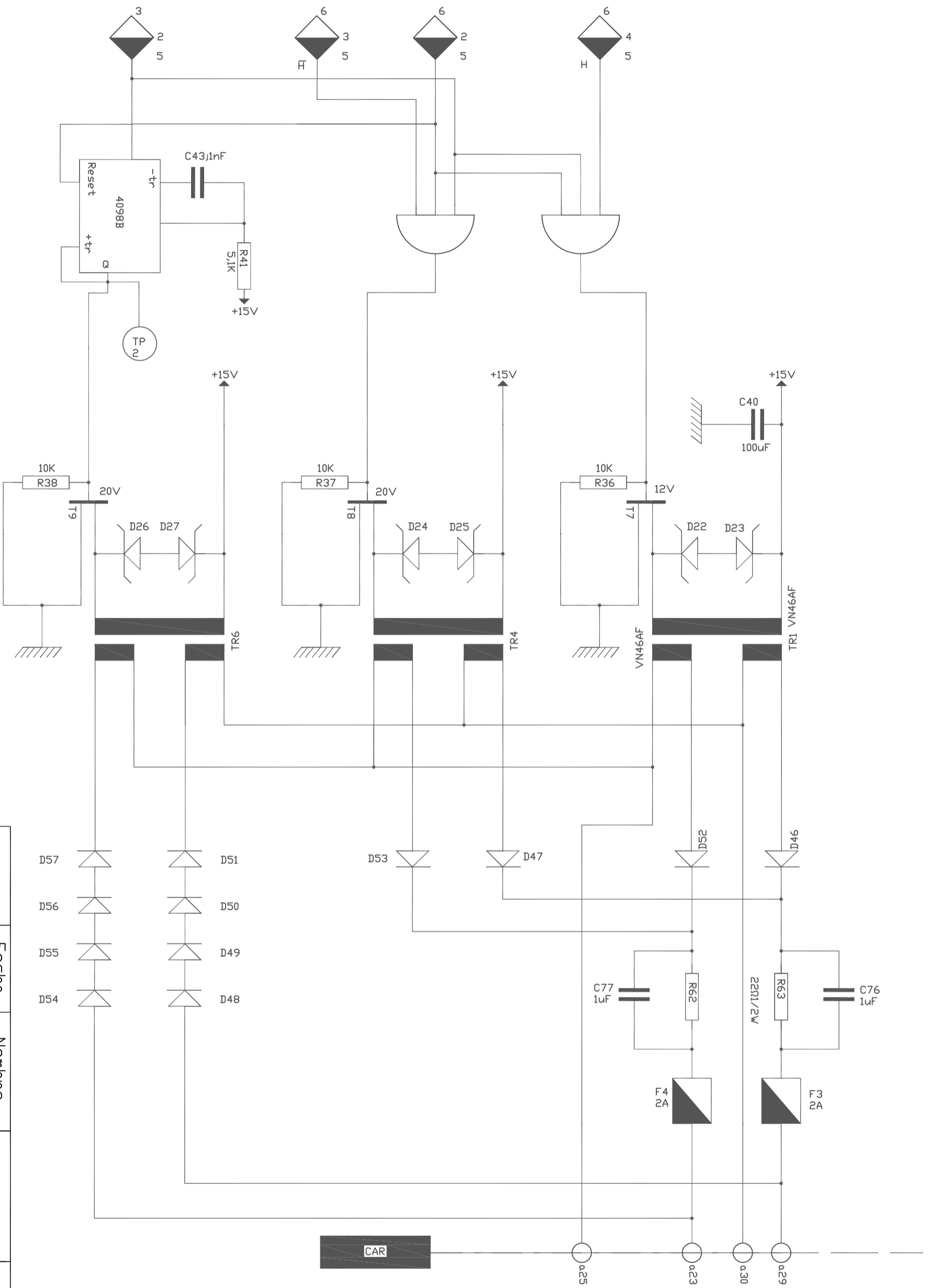



Escalera	Fecha	Nombre
Dibujado:	01/04/15	Valeria Espina B.
Compr.Obado:	01/06/15	José Torreblanca
Id:snorm:		

Esquema eléctrico de control de la etapa de potencia

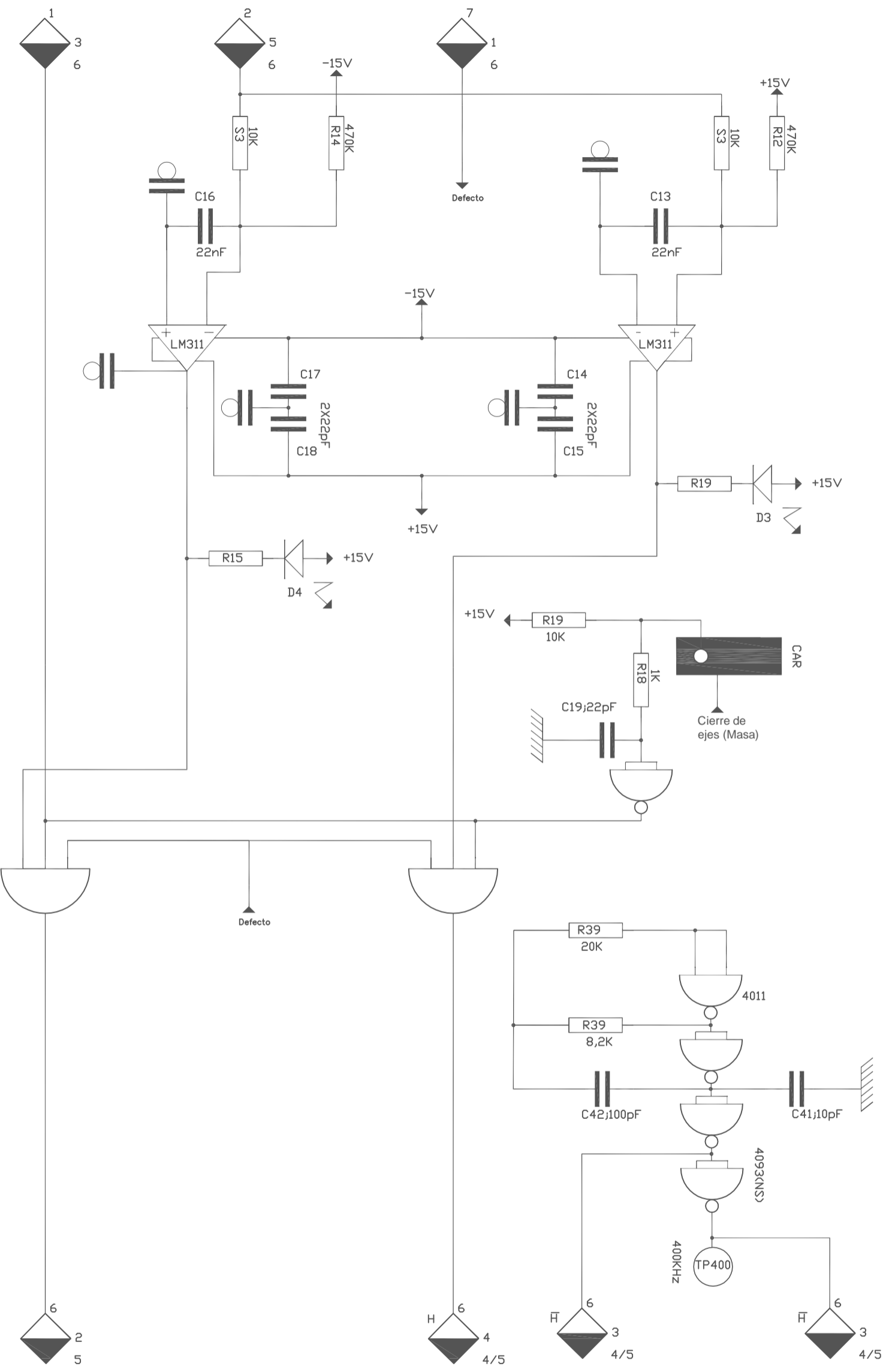
Universidad de Salamanca
 ETSII de Béjar
 DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT ACMA.X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW

Plano N° 4
Sustituye a:
Sustituido por:




Escala		Fecha		Nombre	
Dibujado:		01/04/15	Valeria Espina B.		
Compr Obado:		01/06/15	José Torreblanca		
Id:snorm:					
					
Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW					
Plano N° 5					
Sustituye a:					
Sustituido por:					

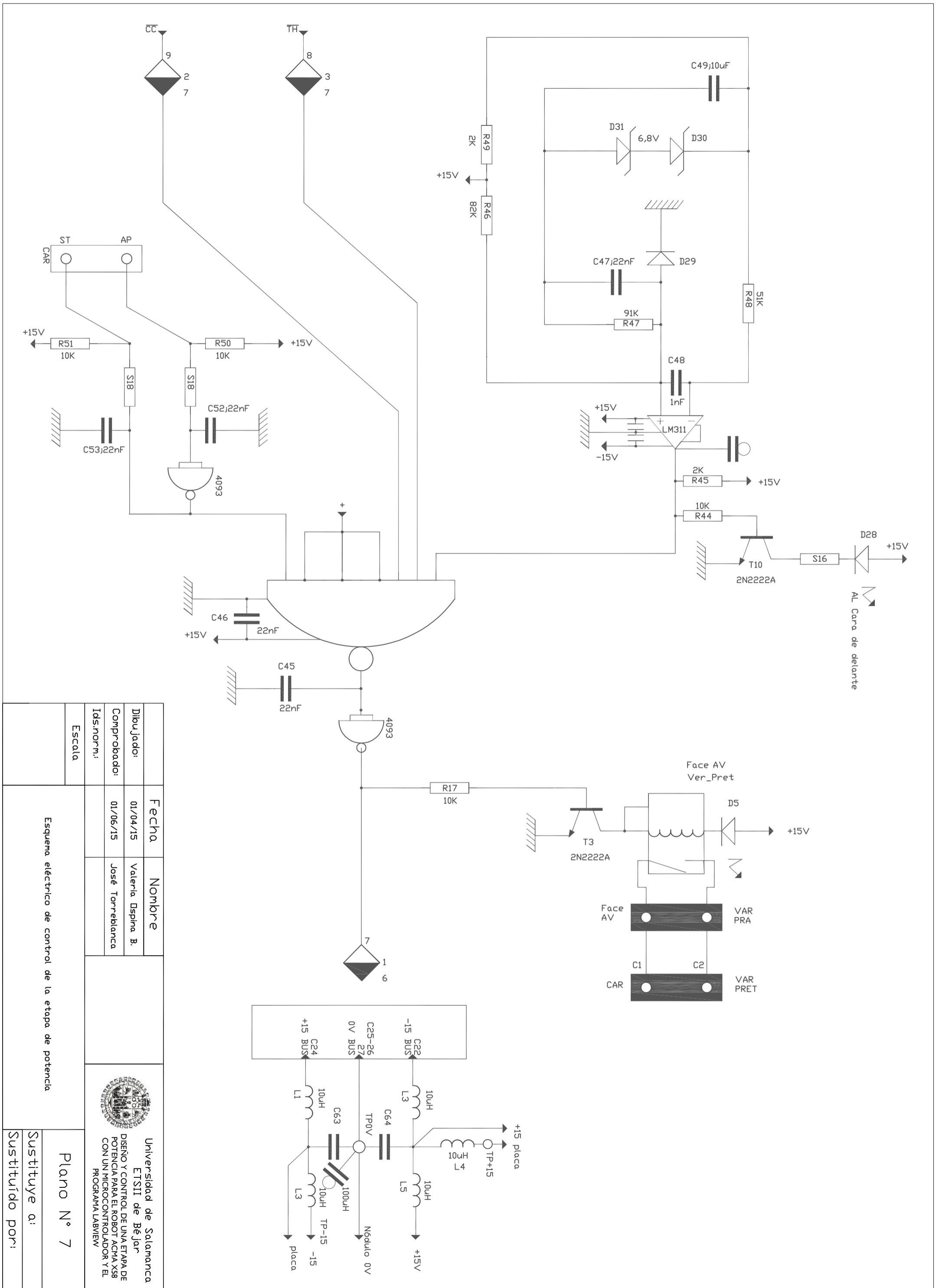
Esquema eléctrico de control de la etapa de potencia



Escala	Fecha	Nombre
	01/04/15	Valeria Espina B.
Comprobado:	01/06/15	José Torreblanca
Id. norm.:		


 Universidad de Salamanca
 ETSII de Béjar
 DISEÑO Y CONTROL DE UNA ETAPA DE
 POTENCIA PARA EL ROBOT ACMA X58
 CON UN MICROCONTROLADOR Y EL
 PROGRAMA LABVIEW

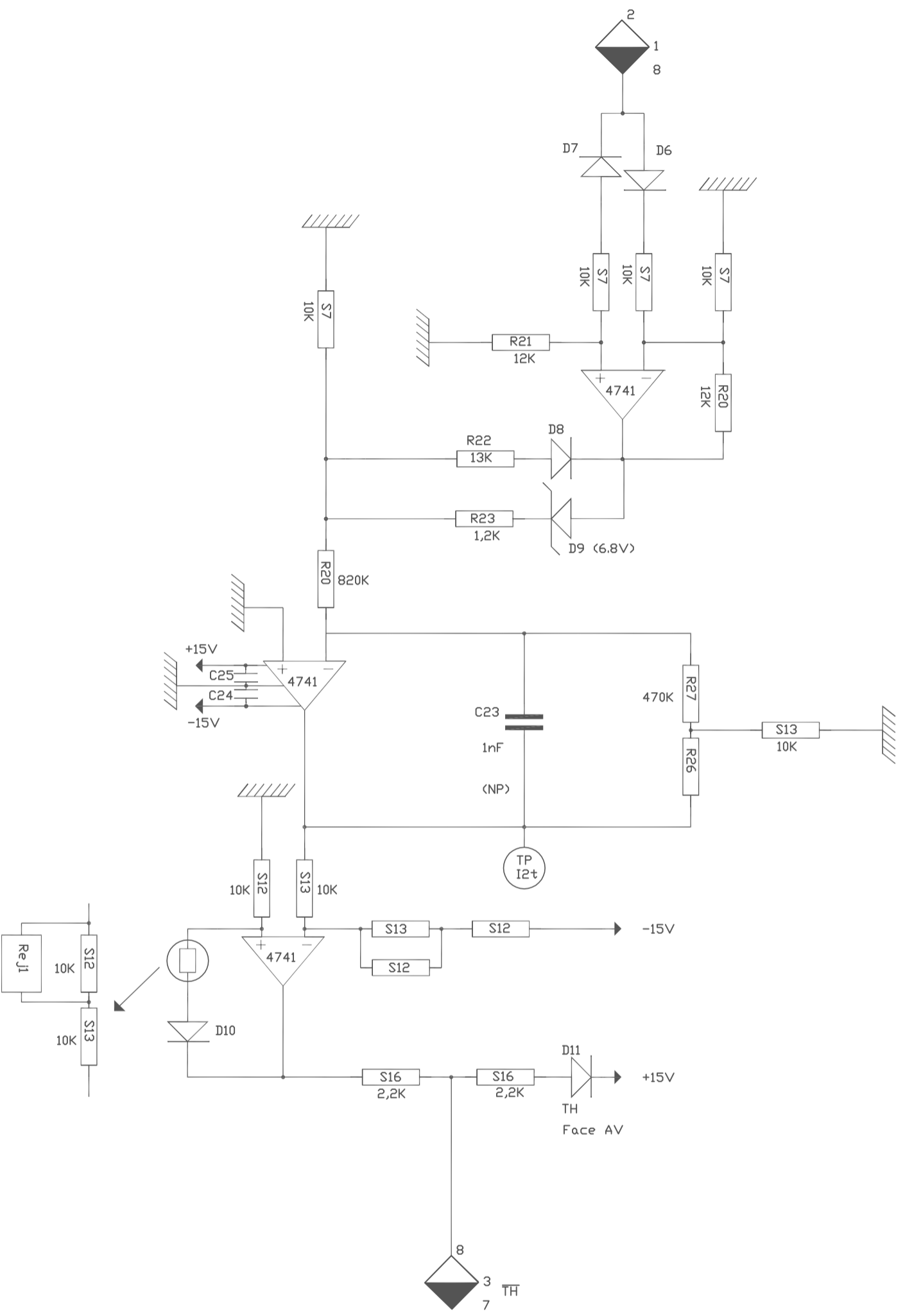
Esquema eléctrico de control de la etapa de potencia
 Plano N.º 6
 Sustituye a:
 Sustituido por:



Escuela		Esquema eléctrico de control de la etapa de potencia	
Id:nom:n:	Jose Torrealanca	Fecha	01/06/15
Comprobado:	01/06/15	Nombre	Valeria Espina B.
Dibujado:	01/04/15		

Universidad de Salamanca
 ETSII de Béjar
 DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT ACMA.XS8 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW

Plano N° 7
 Sustituye a:
 Sustituido por:



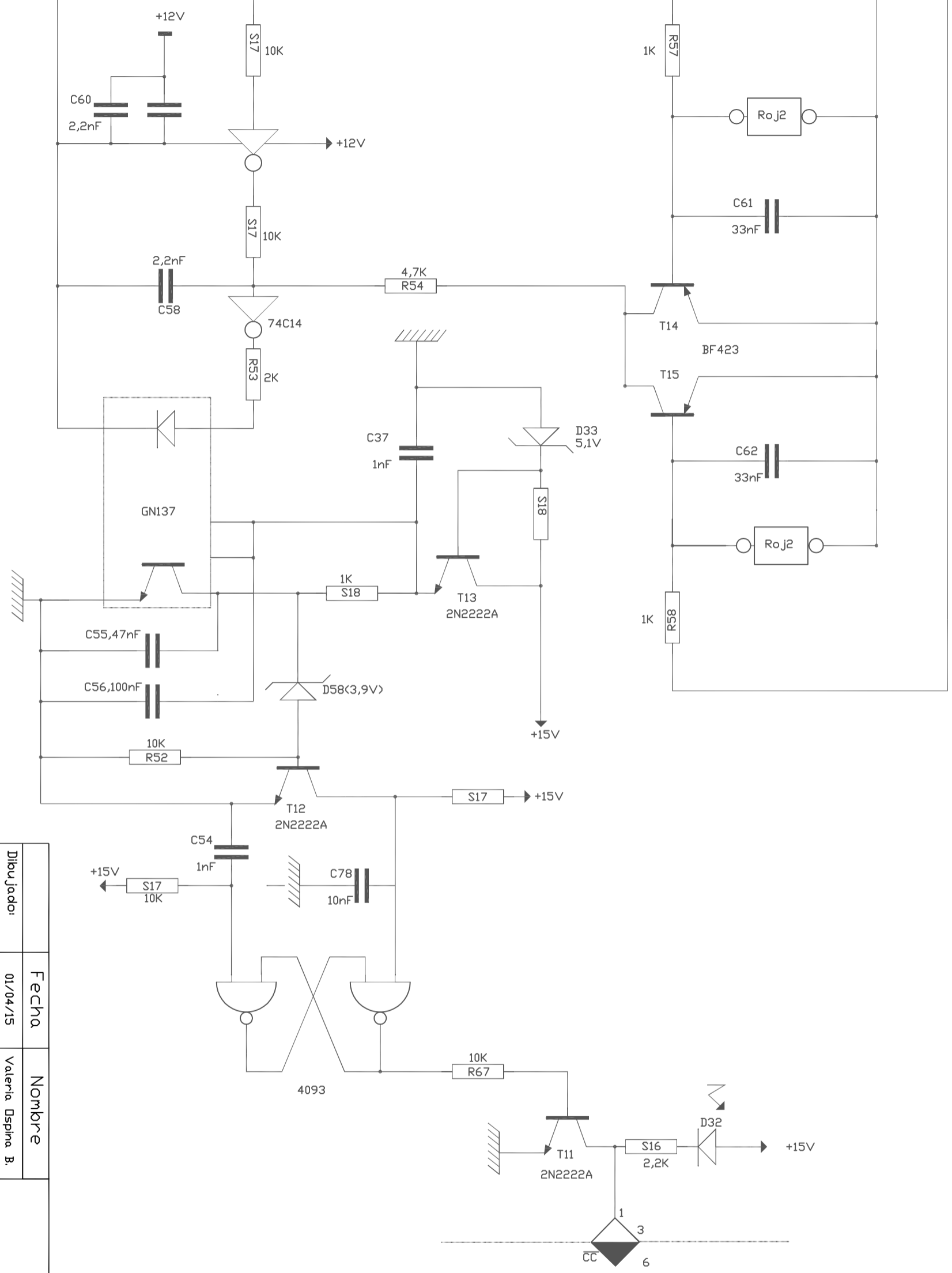
Fecha	01/04/15	Nombre	Valeria Espina B.
Dibujado:	01/06/15	Comprobado:	José Torreblanca
Id:snorm:			
Escala			




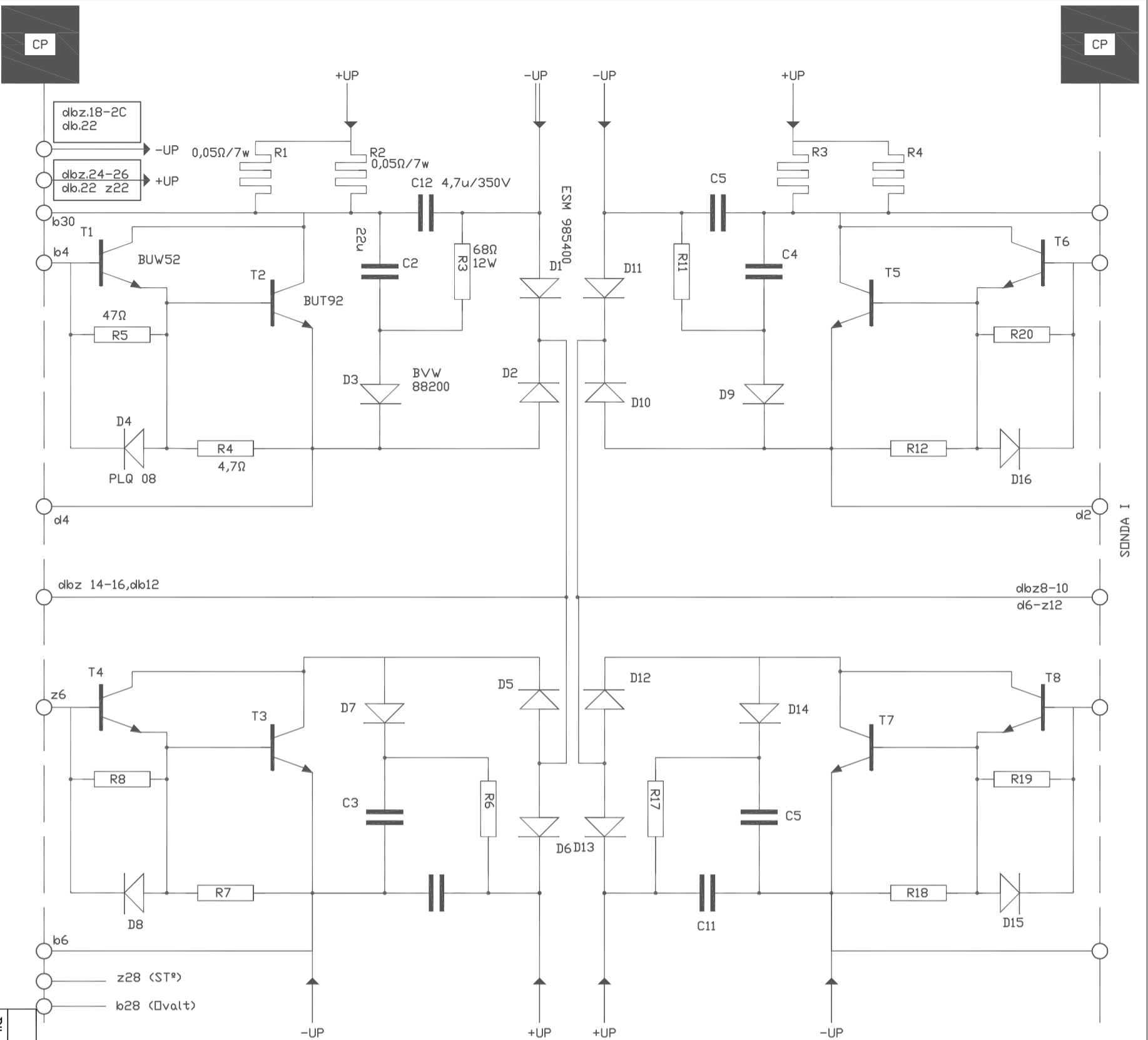
Universidad de Salamanca
 ETSII de Béjar
 DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW


Esquema eléctrico de control de la etapa de potencia	
Plano N° 8	
Sustituye a:	
Sustituido por:	

CAR C19 C17 C32 C18 C31



Escalafón	Fecha	Nombre	 Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW
Dibujado:	01/04/15	Valeria Espina B.	
Comprobado:	01/06/15	José Torreblanca	
Idioma:			
Esquema eléctrico de control de la etapa de potencia			Plano N° 9
Sustituye a:			
Sustituido por:			



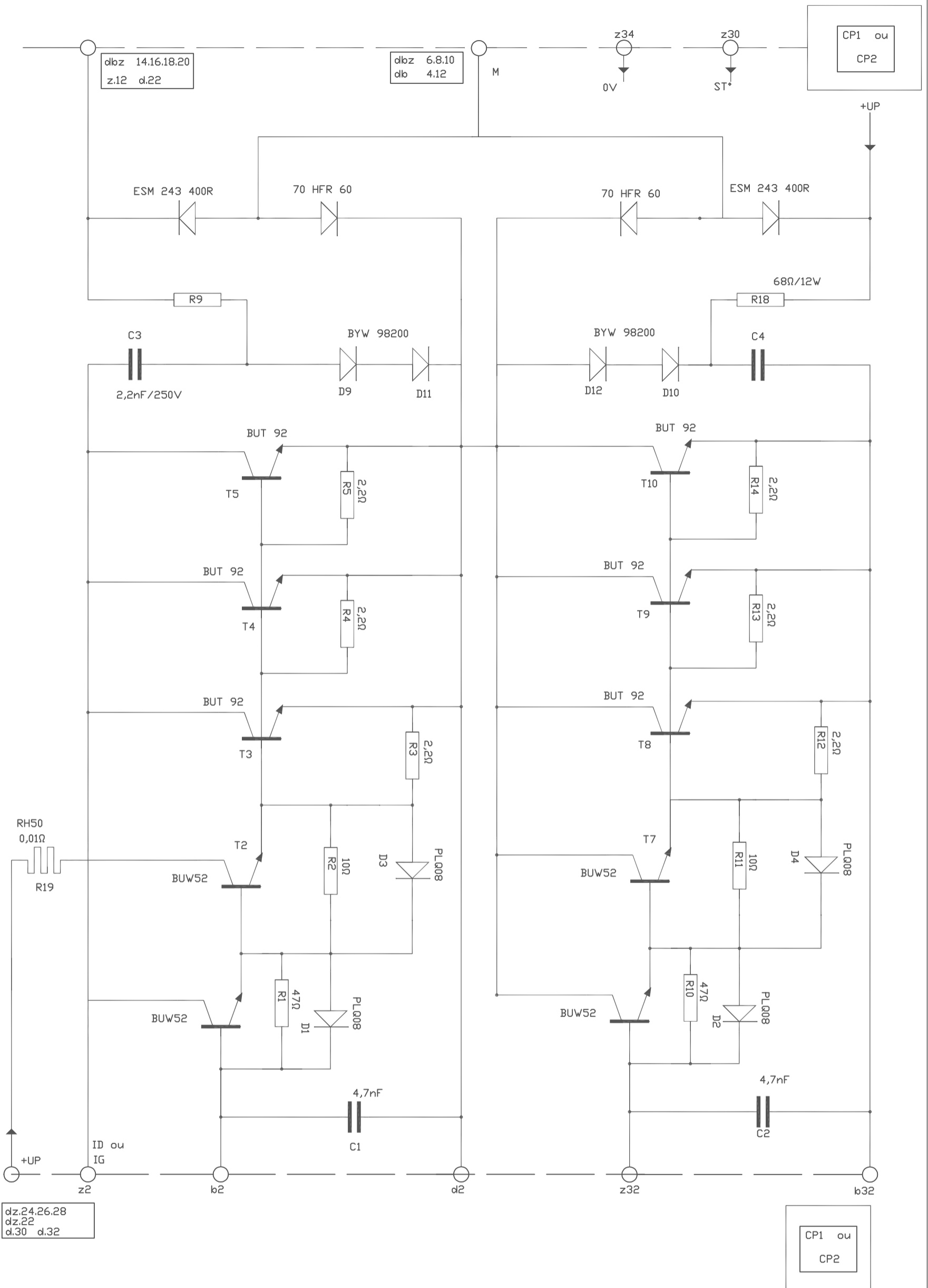
Escala	Fecha	Nombre	 Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW
Dibujado:	01/04/15	Valeria Espina B.	
Comprobado:	01/06/15	José Torreblanca	
Id:snorm:			

Esquema eléctrico de control de la etapa de potencia

Plano N° 10

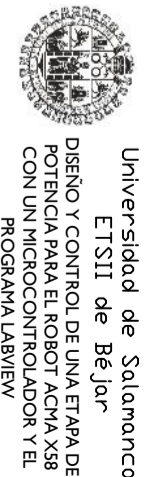
Sustituye a:

Sustituido por:



Dibujado:	01/04/15	Nombre	Valeria Espina B.
Comprobado:	01/06/15	Nombre	José Torreblanca
Id. norm.:			
Escala			

Fecha: 01/04/15
 Nombre: Valeria Espina B.
 Comprobado: 01/06/15
 Nombre: José Torreblanca

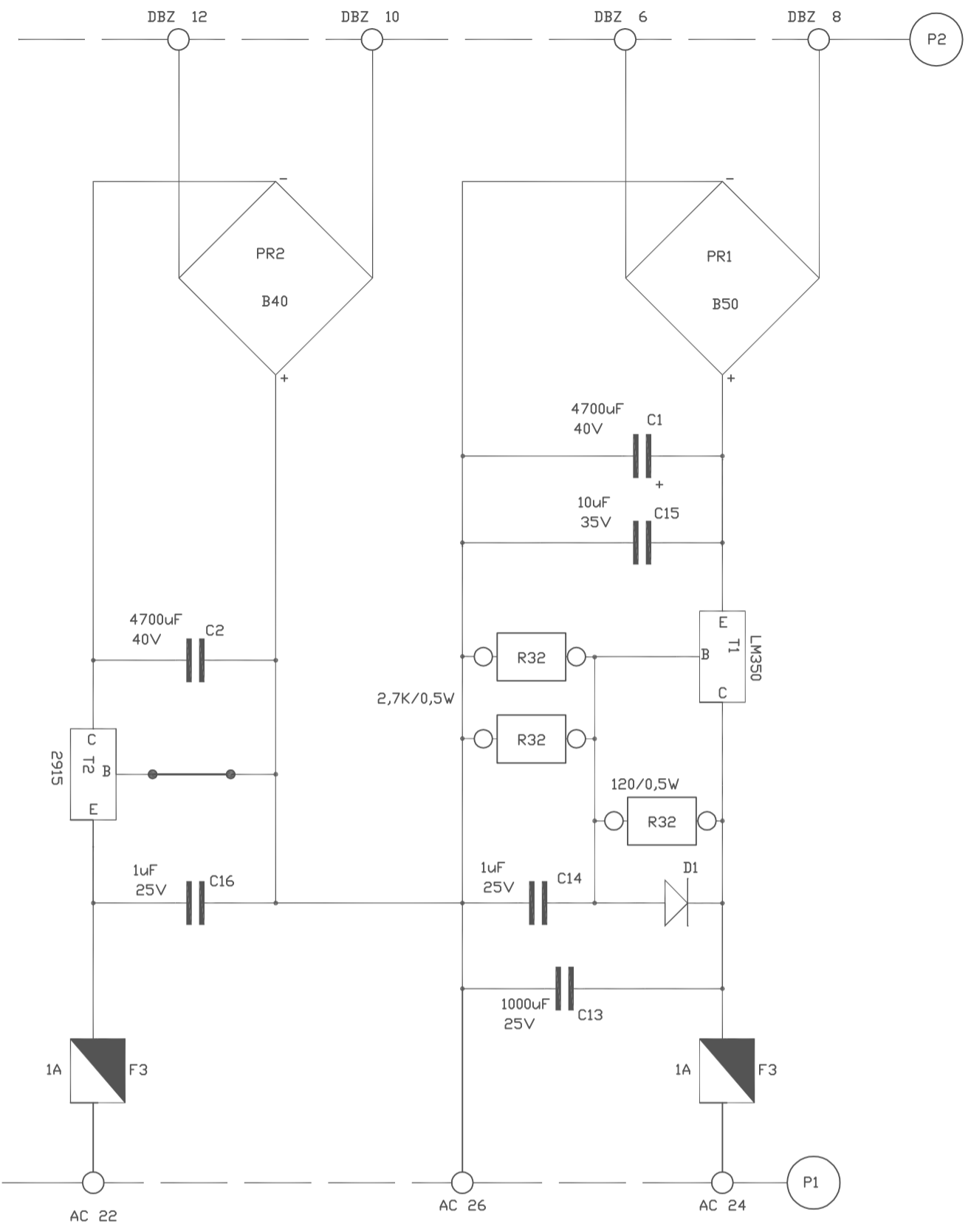



Esquema eléctrico de control de la etapa de potencia

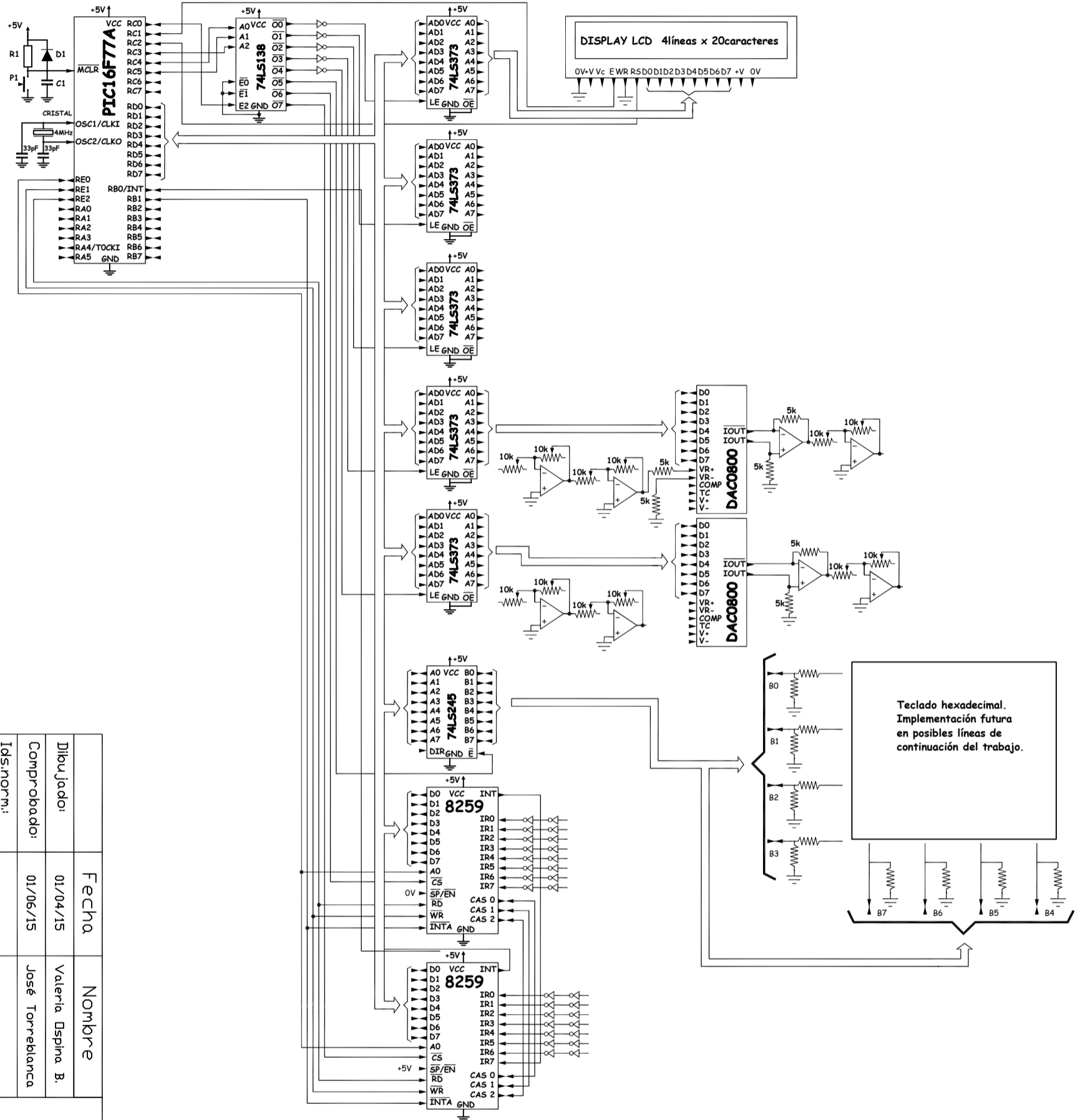
Plano N° 11


Sustituye a:

Sustituido por:

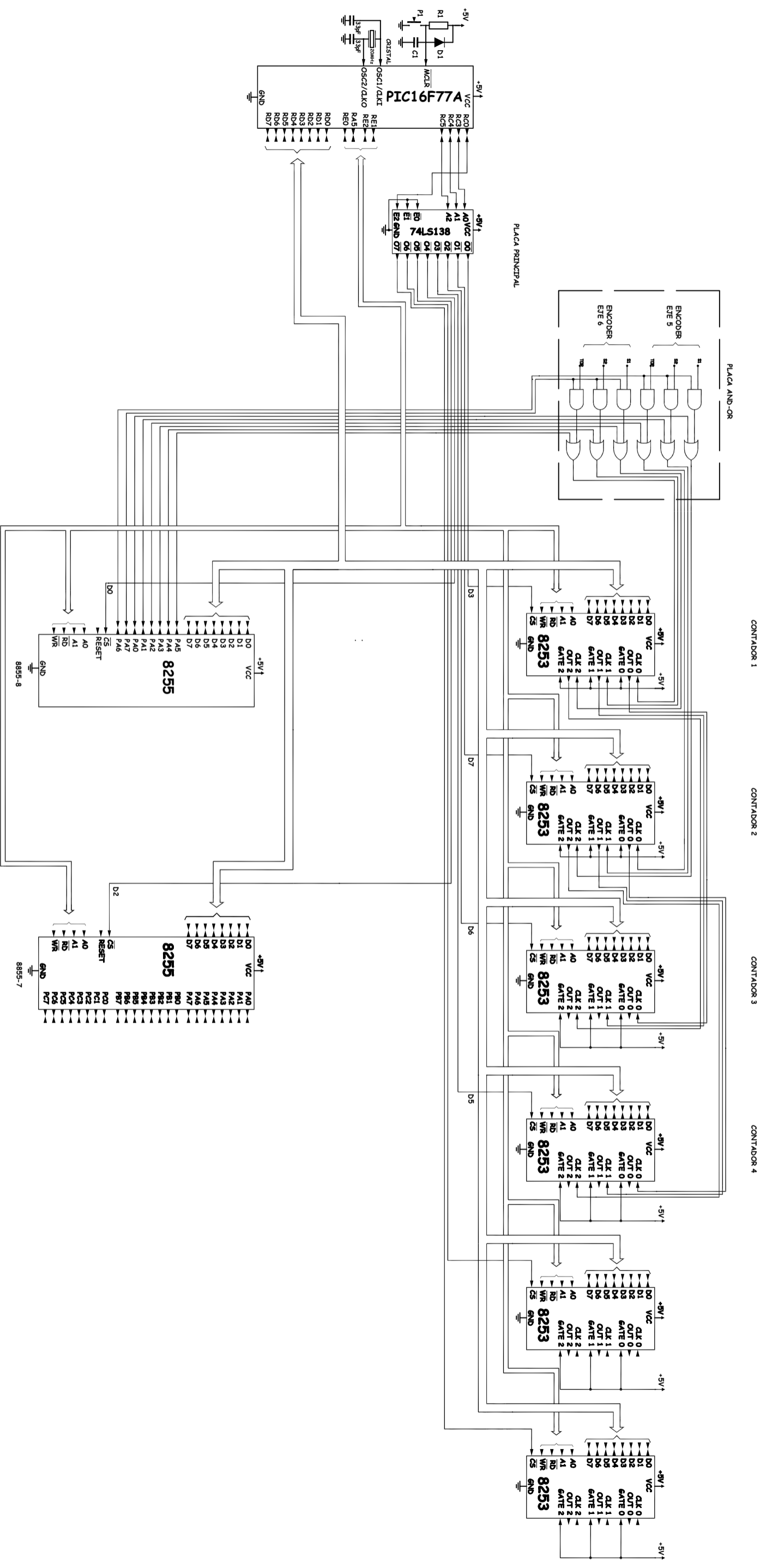


Escala	Fecha	Nombre	 Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW
Comprobado:	01/06/15	Valeria Espina B.	
Dibujado:	01/04/15	Valeria Espina B.	
Esquema eléctrico de control de la etapa de potencia		Plano N° 12	
Sustituye a:			
Sustituido por:			



Escalafón:	Escalafón
Idioma:	Idioma
Comprobado:	01/06/15
Dibujado:	01/04/15
Fecha:	Nombre
Idioma:	Valeria Espina B.
Idioma:	José Torreblanca
 Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT ACMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW	
Plano N° 13	
Sustituye a:	
Sustituido por:	

Esquema eléctrico de la placa principal.



CONTADOR 1


CONTADOR 2

CONTADOR 3

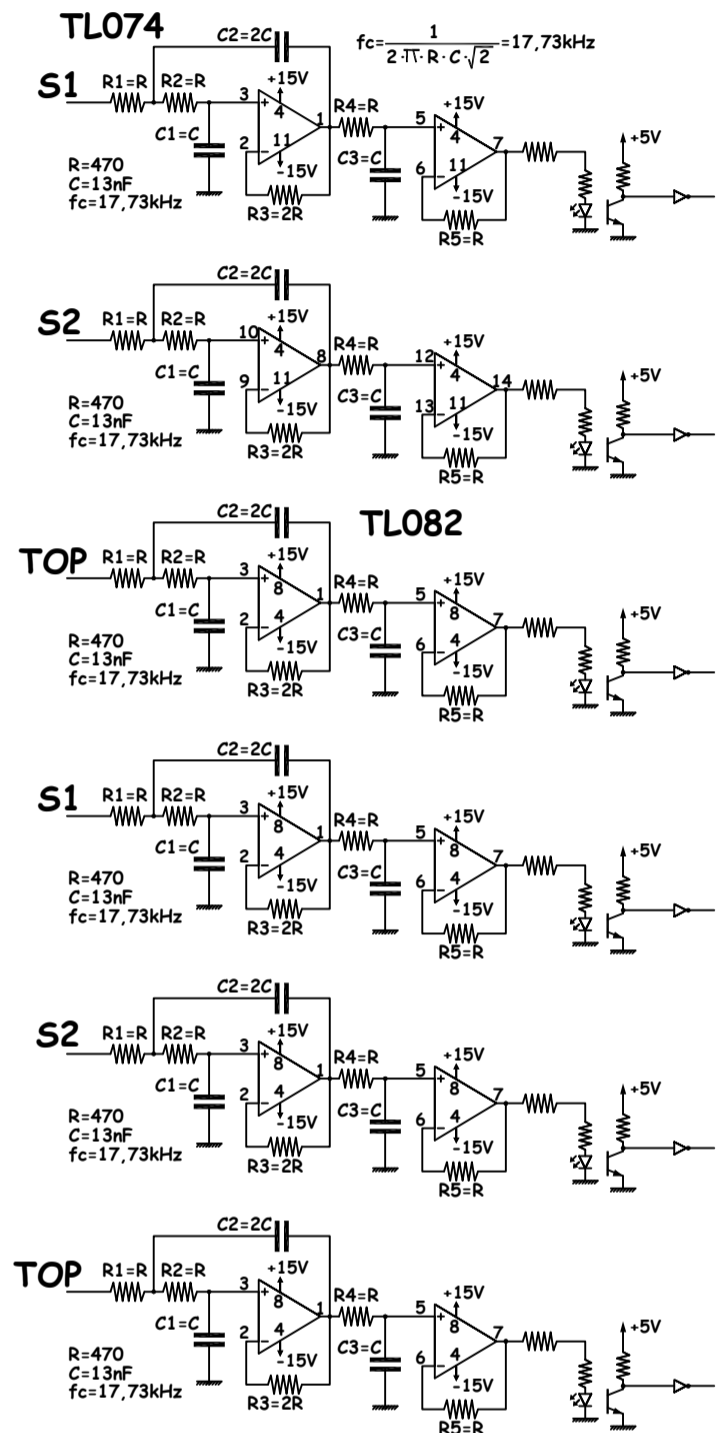
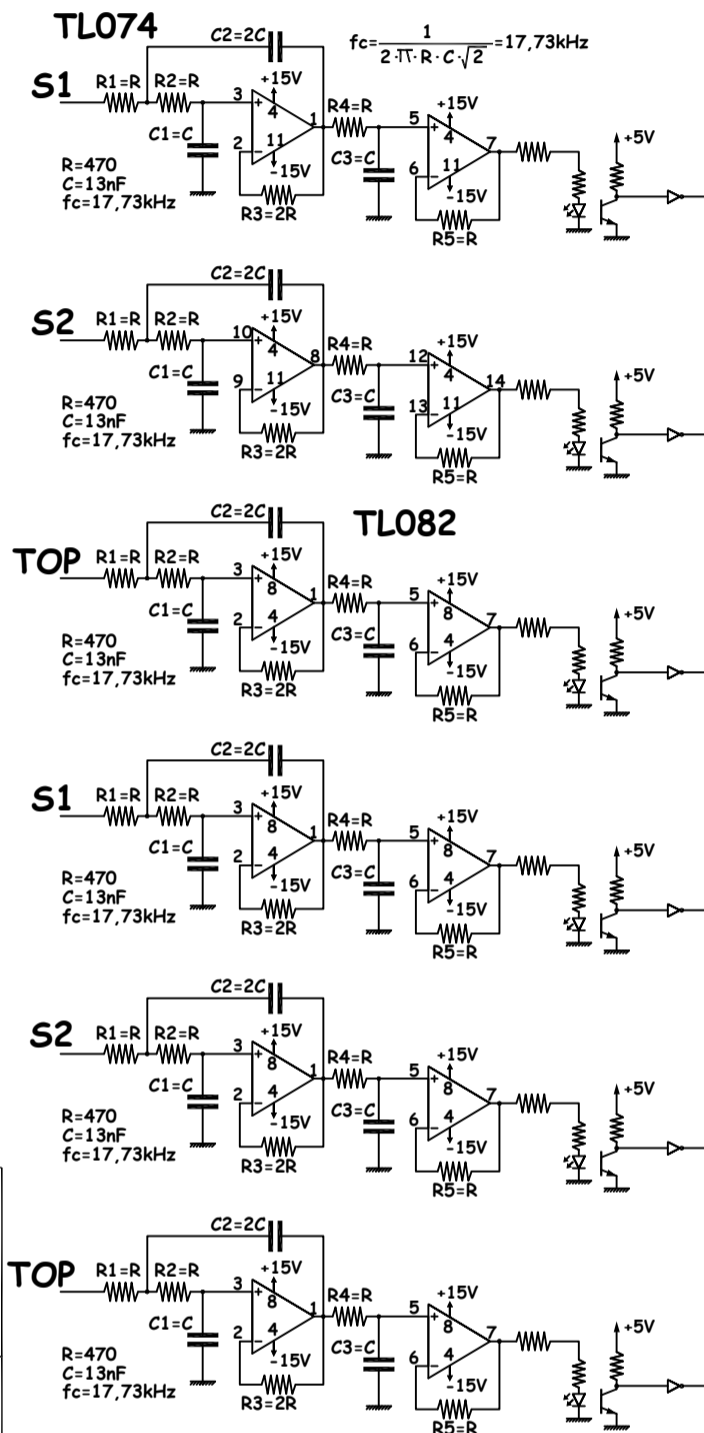
CONTADOR 4


PLACA PRINCIPAL

PLACA AND-OR

Fecha	Nombre	 Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT AGMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW
Dibujado:	Valeria Espina B.	
Comprobado:	José Torreblanca	
Idioma:	Escala	
Esquema eléctrico de la placa auxiliar 8255 - 8253		
Plano N° 14		
Sustituye a:		
Sustituido por:		

Sección de la placa que no se utiliza.



Escala	Fecha	Nombre
	01/04/15	Valeria Espina B.
Comprobado:	01/06/15	José Torrealba
Idioma:		
		
Universidad de Salamanca ETSII de Béjar DISEÑO Y CONTROL DE UNA ETAPA DE POTENCIA PARA EL ROBOT ACMA X58 CON UN MICROCONTROLADOR Y EL PROGRAMA LABVIEW		
Plano N° 15		
Sustituye a:		
Sustituido por:		

Esquema eléctrico de la placa auxiliar de adaptación de señales.