



UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE TECNOLOGÍA
ELECTRÓNICA

***Sistema microposicionador motorizado en un eje para
caracterización de dispositivos electroópticos***

PROYECTO FIN DE CARRERA
INGENIERÍA TÉCNICA INDUSTRIAL: ELECTRÓNICA INDUSTRIAL

Autor: JORGE GÓMEZ MARCOS
Directora: VIRGINIA URRUCHI DEL POZO

A mis padres, hermano y familia, amigos y compañeros

A mi tutora Virginia y mi amigo Agustín

Gracias

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN	1
1.1 INTRODUCCIÓN.....	2
1.2 SISTEMAS ÓPTICOS DE MEDIDA.....	3
1.3 ESTRUCTURA DEL DOCUMENTO	4
CAPÍTULO 2. MOTIVACIÓN Y OBJETIVOS	7
2.1 INTRODUCCIÓN.....	8
2.2 MOTIVACIÓN.....	9
2.3 OBJETIVOS	10
CAPÍTULO 3. SISTEMA MECÁNICO MOTORIZADO	13
3.1 INTRODUCCIÓN.....	14
3.2 CARACTERÍSTICAS DE LOS MOTORES `PASO A PASO´ .	14
3.3 PRINCIPIO DE FUNCIONAMIENTO DE LOS MOTORES `PASO A PASO´	15
3.4 TIPOS DE MOTORES `PASO A PASO´	16
3.4.1 Motores de imán permanente	16
3.4.2 Motores de reluctancia variable	19
3.4.3 Motores híbridos	20
3.5 MODOS DE FUNCIONAMIENTO DE LOS MOTORES DE IMÁN PERMANENTE	21
3.5.1 Motores unipolares de secuencia normal	21
3.5.2 Motores unipolares de modo onda	22
3.5.3 Motores unipolares de medio paso.....	23
3.6 CONTROL DEL MOTOR	23
3.7 ELECCIÓN DE COMPONENTES MECÁNICOS.....	24
3.7.1 Conjunto vagoneta-carril.....	27
3.7.2 Conjunto completo: vagoneta-carril-motor.....	28

CAPÍTULO 4. HARDWARE DE CONTROL DEL SISTEMA MOTORIZADO	29
4.1 DESCRIPCIÓN DE LOS BLOQUES DEL SISTEMA	30
4.2 BLOQUE CONTROLADOR	31
4.3 SOLUCIONES PARA DESARROLLAR EL BLOQUE CONTROLADOR.....	31
4.3.1 Dispositivo FPGA	32
4.3.2 Dispositivo microprocesador	33
4.3.3 Dispositivo microcontrolador	34
4.4 ELECCIÓN DEL MICROCONTROLADOR PIC18F2550	35
4.4.1 Componentes de un microcontrolador	36
4.4.2 Características principales del microcontrolador PIC18F2550.....	39
4.4.3 Asignación de pines en el microcontrolador PIC18F2550.....	40
4.4.4 Organización de la memoria	43
4.4.5 Líneas de Entrada-Salida	47
4.4.6 Reloj principal.....	47
4.4.7 Recursos auxiliares: Interrupciones y Puertos de comunicación	48
4.4.8 RESET	51
4.4.9 Interfaz de comunicación Universal Serial Bus (USB).....	52
4.5 SELECCIÓN DE COMPONENTES Y CONEXIONADO	54
4.5.1 Alimentación.....	55
4.5.2 Condensadores de desacoplo	56
4.5.3 Regulador de tensión	56
4.5.4 Driver L293.....	57
4.5.5 Circuito de RESET	58
4.5.6 Cristal del oscilador.....	59
4.5.7 Pulsadores	61
4.5.8 Conexión USB.....	65
4.5.9 Pantalla LCD	68
4.5.10 Comunicación con el programador.....	70
4.5.11 Diseño completo del hardware de control.....	71

CAPÍTULO 5. SOFTWARE DE CONTROL DEL SISTEMA MOTORIZADO	75
5.1 DESCRIPCIÓN DE LOS DISPOSITIVOS DEL SISTEMA.....	76
5.2 LENGUAJES DE PROGRAMACIÓN DEL PIC	77
5.3 CONFIGURACIÓN DE LA PROGRAMACIÓN.....	78
5.3.1 Grabación de los PICs	79
5.3.2 Creación de un proyecto con CCS PIC C Compiler	79
5.3.3 Creación de un proyecto con MPLAB.....	81
5.4 DIAGRAMA DE FLUJO DEL CÓDIGO PROGRAMADO	84
5.4.1 Diagrama de flujo de la función “MAIN”	85
5.4.2 Diagrama de flujo de la función “CONFIGURACION”	99
5.4.3 Diagrama de flujo de la función “TEST LCD”.....	101
5.4.4 Diagrama de flujo de la función “POSICIÓN_ACTUAL”	102
5.4.5 Diagrama de flujo de la función “LEER_PULSADORES”	103
5.4.6 Diagrama de flujo de la función “PASO_MOTOR”.....	106
5.4.7 Diagrama de flujo de la función “GUARDA_EEPROM”	108
5.4.8 Diagrama de flujo de la función “POSICION_IZQUIERDA”	109
 CAPÍTULO 6. RESULTADOS EXPERIMENTALES	 111
6.1 INTRODUCCIÓN.....	112
6.2 MONTAJE DE CALIBRACIÓN DEL SISTEMA.....	112
6.2.1 Emisor láser	113
6.2.2 Filtros neutros	114
6.2.3 Regla micrométrica	114
6.2.4 Objetivo óptico	115
6.2.5 Cámara CCD.....	116
6.2.6 Montaje de calibración completo	117
6.3 GUÍA DE FUNCIONAMIENTO DEL POSICIONADOR	118

6.4 CALIBRACIÓN Y MEDIDA DE ESPECIFICACIONES	121
6.4.1 Procedimiento de medida	122
6.4.2 Medida de la resolución del posicionador micrométrico	123
6.4.3 Calibración con corrección del destensado de la correa	124
6.4.4 Medición de la precisión obtenida con una corrección de 80 pasos.....	138
6.5 INICIALIZACIÓN Y CONEXIONES A PUERTOS	143
6.5.1 Verificación de funcionamiento del PROGRAMA de RESET	143
6.5.2 Verificación de funcionamiento de la conexión USB	146
 CAPÍTULO 7. CONCLUSIONES Y MEJORAS FUTURAS	 147
7.1 CONCLUSIONES.....	148
7.2 MEJORAS FUTURAS.....	150
 BIBLIOGRAFÍA	 151
REFERENCIAS BIBLIOGRÁFICAS	152
 ANEXOS	 153
ANEXO 1: Hoja de características del PIC 18F2550	154
ANEXO 2: Hoja de características del driver L293D	155
ANEXO 3: Hoja de características del Regulador LM7805	156
ANEXO 4: Hoja de características de la pantalla LCD 162D.....	157
ANEXO 5: Hoja de características de la vagoneta-carril	158
ANEXO 6: Código fuente del software del microcontrolador ...	159
 PLANOS	 169
PLANO 1: CARA TOP PLACA DEL MOTOR	170
PLANO 2: CARA BOTTOM PLACA DEL MOTOR.....	171
PLANO 3: CARA COMPONENTES PLACA DEL MOTOR	172
PLANO 4: CARA BOTTOM PLACA DE PULSADORES	173
PLANO 5: CARA COMPONENTES PLACA DE PULSADORES..	174
 PLIEGO DE CONDICIONES	 175
HARDWARE	176
SOFTWARE	176

PRESUPUESTO	177
1.1 INTRODUCCIÓN.....	178
1.2 COSTE DEL MATERIAL ELECTRÓNICO Y MECÁNICO	178
1.3 COSTE DEL PERSONAL INTERVINIENTE.....	179
1.4 PRESUPUESTO TOTAL.....	180

ÍNDICE DE FIGURAS

Figura 1.1. Ejemplos de los distintos tipos de microposicionadores comerciales	2
Figura 2.1. Listado ejemplo de posicionadores y precios	9
Figura 3.1. Ejemplo de motor `paso a paso´	14
Figura 3.2. Principio de funcionamiento de un motor `paso a paso´	15
Figura 3.3. Motor `paso a paso´ de imán permanente.....	17
Figura 3.4. Secuencias de movimiento del motor según la alimentación del estator	18
Figura 3.5. Detalle de las bobinas en un motor bipolar.....	19
Figura 3.6. Detalle de las bobinas en un motor unipolar	19
Figura 3.7. Imagen de un motor de reluctancia variable.....	20
Figura 3.8. Imagen de un motor híbrido.....	21
Figura 3.9. Posicionamiento de un motor unipolar según una secuencia de activación normal.....	22
Figura 3.10. Posicionamiento de un motor unipolar operando en modo onda	22
Figura 3.11. Posicionamiento de un motor unipolar de medio paso.....	23
Figura 3.12. Motor utilizado para la implementación del sistema motorizado en un eje	25
Figura 3.13. Circuito de control del motor con integrado unipolar disponible	25
Figura 3.14. Imagen del conjunto vagoneta-carril.....	28
Figura 3.15. Imagen del subsistema mecánico del microposicionador	28
Figura 4.1. Diagrama de bloques del sistema completo.....	30
Figura 4.2. Diagrama interno de una FPGA.....	32
Figura 4.3. Microcontrolador	34
Figura 4.4. Componentes del microcontrolador	36
Figura 4.5. Pin-out PIC18F2550.....	40
Figura 4.6. Arquitectura interna del microcontrolador PIC18F2550.....	42
Figura 4.7. Mapa de memoria del microcontrolador PIC18F2550	44
Figura 4.8. Contador de programa	45
Figura 4.9. Banco de registros del PIC18F2550	46
Figura 4.10. Modos del oscilador para el microcontrolador PIC18F2550.....	48
Figura 4.11. Lógica de control para la generación de la interrupción	50

Figura 4.12. Diagrama de bloques del circuito de RESET	52
Figura 4.13. Perspectiva general de un USB.....	53
Figura 4.14. Asignación de pines de Entradas/Salidas del PIC18F2550	55
Figura 4.15. Alimentación	55
Figura 4.16. Condensadores de desacoplo	56
Figura 4.17. Conexión del regulador de tensión	57
Figura 4.18. Conexión del driver L293	58
Figura 4.19. Circuito de RESET con pulsador	59
Figura 4.20. Esquema de conexión del cristal oscilador.....	60
Figura 4.21. Pulsadores para el control LCD	62
Figura 4.22. Pulsador 2 accionado	63
Figura 4.23. Pulsadores del final de carrera	65
Figura 4.24. Condensador para fijar tensión USB	65
Figura 4.25. Conexión del USB	66
Figura 4.26. USB a 48MHz	67
Figura 4.27. Pantalla LCD	69
Figura 4.28. Esquemático de la LCD con la numeración de pines.....	69
Figura 4.29. Esquemático de la comunicación entre el ICSP y el PIC18F2550.....	71
Figura 4.30. Esquemático del hardware de control. Placa del motor: microcontrolador, driver L293, circuito de RESET, regulador LM7805 y conector	72
Figura 4.31. Esquemático del hardware de control. Placa del motor: motor, USB, ICSP, LCD y fines de carril derecho e izquierdo	73
Figura 4.32. Esquemático del hardware de control. Placa de pulsadores: pulsadores y conector.....	74
Figura 5.1. Comunicación entre dispositivos del sistema motorizado de posicionamiento.....	76
Figura 5.2. Ventana de PIC Wizard para la selección del dispositivo	80
Figura 5.3. Detalle de la ventana de PCW para la selección de librerías.....	80
Figura 5.4. Esquema del conexionado entre el ordenador, el programador y el microcontrolador	81
Figura 5.5. Programador MPLAB ICD2.....	81
Figura 5.6. Menú principal MPLAB.....	82
Figura 5.7. Ventana de selección del dispositivo en MPLAB.....	82
Figura 5.8. Detalle de la ventana para la carga del archivo archivo “.hex”.....	83
Figura 5.9. Selección de programador MPLAB ICD2	83
Figura 5.10. Éxito en la programación del PIC	84
Figura 5.11. Simbología utilizada en los diagramas de flujo.....	85
Figura 5.12. Diagrama de flujo función MAIN: inicializaciones puertos, USB y LCD	86
Figura 5.13. Diagrama de flujo función MAIN: Comprobación RESET, USB conectado	87
Figura 5.14. Diagrama de flujo función MAIN: pulsadores	88
Figura 5.15. Pantalla LCD. Mensaje “USB conectado”.....	88
Figura 5.16. Diagrama de flujo del PULSADOR1	90
Figura 5.17. Pantalla LCD. Mensajes de “Posición actual” y “Posición nueva”	91
Figura 5.18. Diagrama de flujo del PULSADOR4	92

Figura 5.19. Diagrama de flujo del PULSADOR2	93
Figura 5.20. Pantalla LCD. Mensaje “Pulsa izquierda. Final de carril izquierdo”	94
Figura 5.21. Diagrama de flujo del PULSADOR3	95
Figura 5.22. Diagrama de flujo del PULSADOR5	96
Figura 5.23. Situación inicial de desplazamiento a izquierdas	97
Figura 5.24. Situación intermedia de desplazamiento a izquierdas.....	98
Figura 5.25. Situación final de desplazamiento a izquierdas.....	98
Figura 5.26. Diagrama de flujo de la función CONFIGURACION.....	100
Figura 5.27. Diagrama de flujo de la función TEST LCD.....	101
Figura 5.28. Pantalla LCD. Mensaje de inicialización y test de funcionamiento.....	101
Figura 5.29. Diagrama de flujo de la función POSICION_ACTUAL	102
Figura 5.30. Diagrama de flujo de la función LEER_PULSADORES	104
Figura 5.31. Diagrama de flujo de la función PASO_MOTOR.....	106
Figura 5.32. Diagrama de flujo de la función GUARDA EEPROM	108
Figura 5.33. Diagrama de flujo de la función POSICION_IZQUIERDA.....	109
Figura 6.1. Mesa óptica.....	112
Figura 6.2. Láser en el sistema de caracterización.....	113
Figura 6.3. Filtros neutros	114
Figura 6.4. Regla micrométrica de la marca Thorlabs	115
Figura 6.5. Objetivo de 20 aumentos (X20) de la marca Thorlabs	115
Figura 6.6. Cámara CCD de la marca Hamamatsu	116
Figura 6.7. Sistema microposicionador insertado en el montaje de caracterización óptico	117
Figura 6.8. Vista en planta del prototipo implementado.....	118
Figura 6.9. Esquema de los pulsadores del prototipo implementado.....	119
Figura 6.10. Vista lateral del prototipo implementado.....	120
Figura 6.11. Alzado principal del prototipo.....	120
Figura 6.12. Prototipo final del sistema microposicionador en un eje.....	121
Figura 6.13. Reglaje óptico	122
Figura 6.14. Calibración de la resolución de 1 paso del motor:	
Referencia / 5 pasos derecha / 10 pasos derecha	123
Figura 6.15. Posición inicial con la correa tensada “de izquierda a derecha”	124
Figura 6.16. Avance de N pasos (teóricos) en sentido contrario, “de derecha a izquierda”.....	125
Figura 6.17. Avance de N+1 pasos (teóricos) en sentido inicial, “de izquierda a derecha”	126
Figura 6.18. Medida de 1 paso con cambio en el sentido de movimiento.....	126
Figura 6.19. Prueba de tensado de la correa con N = 30 pasos	127
Figura 6.20. Campana de Gauss para N = 30 pasos.	129
Figura 6.21. Prueba de tensado de la correa con N = 40 pasos	129
Figura 6.22. Campana de Gauss para N = 40 pasos	131
Figura 6.23. Prueba de tensado de la correa con N = 50 pasos	131
Figura 6.24. Campana de Gauss para N = 50 pasos	133
Figura 6.25. Prueba de tensado de la correa con N = 80 pasos	133
Figura 6.26. Campana de Gauss para N = 80 pasos	135
Figura 6.27. Prueba de tensado de la correa con N = 100 pasos	135
Figura 6.28. Campana de Gauss para N = 100 pasos	137

Figura 6.29. Campana de Gauss para 30 pruebas de 80 pasos	139
Figura 6.30. Capturas de referencia y de 1 paso.....	140
Figura 6.31. Capturas de 10 y de 25 pasos	141
Figura 6.32. Capturas de 50 y de 100 pasos	141
Figura 6.33. Capturas de 200 y de 300 pasos	142
Figura 6.34. Capturas de 500 y de 1000 pasos	142
Figura 6.35. Sistema mecánico recolocado con el extremo del carril enfrente de la cámara CCD de captura de imágenes	144
Figura 6.36. Referencias para la comprobación de funcionamiento del PROGRAMA de RESET	144
Figura 6.37. Captura del resultado tras el accionamiento del botón de RESET	145
Figura 6.38. Comunicación entre USB y microcontrolador	146

ÍNDICE DE TABLAS

Tabla 3.1. Secuencia de control de giro.....	26
Tabla 3.2. Relación bobina-nombre asignado	27
Tabla 4.1. Ventajas e inconvenientes de utilizar una FPGA.....	33
Tabla 4.2. Principales características del microcontrolador PIC18F2550	39
Tabla 4.3. Asignación de pines para el microcontrolador PIC18F2550 y funciones que realizan.....	40-41
Tabla 4.4. Selección de la capacidad del condensador.....	60
Tabla 4.5. Relación Pulsadores/Tensión/Valor ADC	64
Tabla 4.6. Pines de la LCD	70
Tabla 5.1. Relación Pulsadores / Tensión / Valor ADC	105
Tabla 5.2. Secuencia de control de giro.....	107
Tabla 5.3. Relación Bobinas/Variables /Pines PIC	107
Tabla 6.1. Medida de la resolución del motor con tensado de la correa para N = 30 pasos....	128
Tabla 6.2. Medida de la resolución del motor con tensado de la correa para N = 40 pasos....	130
Tabla 6.3. Medida de la resolución del motor con tensado de la correa para N = 50 pasos....	132
Tabla 6.4. Medida de la resolución del motor con tensado de la correa para N = 80 pasos....	134
Tabla 6.5. Medida de la resolución del motor con tensado de la correa para N = 100 pasos..	136
Tabla 6.6. Resumen de resultados de dispersión respecto de la media en la medida del paso del motor.....	137
Tabla 6.7. Resultados obtenidos en 30 medidas de la resolución del motor en píxeles con prueba de tensado de la correa para N = 80 pasos	138
Tabla 6.8. Datos característicos de calibrado del paso del motor del microposicionador medidos en micrómetros	139
Tabla 7.1. Comparación de un micrómetro comercial con el prototipo diseñado.....	149

CAPÍTULO 1

INTRODUCCIÓN

1.1 SISTEMAS POSICIONADORES

Los sistemas posicionadores, como su nombre indica, se destinan a ubicar piezas o componentes en lugares determinados donde la posición de destino se encuentra prefijada. Los posicionadores en un eje se basan en plataformas de rodadura para movimientos de traslación donde una vagoneta se desliza sobre un carril. Por tanto, los aspectos más importantes en estos sistemas, y que se deben tener en consideración como criterios de diseño, son la robustez de la vagoneta y del sistema de rodamiento, la precisión del movimiento, es decir la repetitividad en la ubicación de posiciones absolutas y la resolución o unidad mínima de movimiento. Todos ellos se abordarán a lo largo del documento completo en el proceso de diseño.

Existen tres formas de realizar el movimiento de traslación en un eje en los dispositivos que se pueden encontrar en el mercado.

En la Figura 1.1 se muestra un ejemplo de cada uno de los tres tipos.

- a) arrastre manual del carril en la dirección del eje considerado
- b) a través sistemas mecánicos que transforman el movimiento manual de giro en movimiento de traslación en el eje considerado
- c) a partir del caso anterior, utilizando un motor como sistema de control.



a) Manual

b) Mecánico

c) Motorizado

Figura 1.1. Ejemplos de los distintos tipos de microposicionadores comerciales

No es habitual encontrar soluciones en el mercado que combinen las ventajas de los sistemas manuales de arrastre con las ventajas de los sistemas motorizados. Por ejemplo, los sistemas manuales de arrastre pueden estar dotados de carriles para desplazamientos de largo recorrido (centímetros), imprescindibles en algunas aplicaciones; sin embargo, si se combinan con un control motorizado, el encarecimiento del precio final es excesivo y difícil de abordar en ocasiones. Por otra parte, los posicionadores motorizados existentes actualmente en el mercado necesitan el uso de algún *software* dedicado de control sobre las plataformas, lo que hace imprescindible la utilización de un ordenador.

El objetivo general del Proyecto es la construcción de un prototipo de microposicionador controlado en un eje a partir de uno de arrastre manual que se integrará como equipo adicional dentro de un sistema óptico de caracterización. Para evitar la necesidad de trabajar siempre a través de un ordenador, se instalarán unos pulsadores a modo de cursores para desplazar la vagoneta sobre el carril y una pantalla LCD para visualizar la posición en la que se encuentra la vagoneta. Además, se podrá trabajar indistintamente con desplazamientos pequeños (decenas de micrómetros, como en algunas plataformas comerciales) y grandes (del orden de centímetros). Se diseñará la electrónica necesaria para controlar el posicionador, incluyendo el diseño de las placas de control sobre el motor, los pulsadores para su desplazamiento y las comunicaciones con un PC mediante una conexión USB. Los objetivos concretos respecto del diseño del control se definirán en profundidad en el siguiente Capítulo.

1.2 SISTEMAS ÓPTICOS DE MEDIDA

En los laboratorios de caracterización electroóptica es frecuente la implementación de sistemas o montajes de caracterización para la medida de magnitudes que están relacionadas con la respuesta electroóptica de los dispositivos de análisis, como por ejemplo, la luminancia, la transmitancia, las franjas de interferencia, etc. A menudo, los montajes incluyen componentes ópticos cuyo posicionamiento y alineamiento respecto del resto de componentes se convierte en una tarea ardua de realizar y que conlleva un innecesario consumo de tiempo. Son estas aplicaciones un ejemplo de uso donde se vuelve crítico el disponer de soportes y plataformas estables que permitan el

manejo de los componentes ópticos con seguridad, es decir, que permitan cargar componentes sin sufrir deformaciones que puedan influir en la medida. Si a los requerimientos previos se añade, además, la necesidad de caracterizar magnitudes donde la variación espacial es crítica en un eje, el diseño de un microposicionador a medida constituye una herramienta de apoyo importante para la consecución de resultados. Como ejemplo, se puede considerar la determinación del punto de enfoque en una muestra (en un eje) o la medida de la homogeneidad superficial de un parámetro sobre la muestra (en el eje perpendicular).

El objetivo general del Proyecto que nos ocupa, desde el punto de vista de la aplicación a la que va destinado el uso del microposicionador, es dar una solución alternativa a las opciones que existen en el mercado mejorando su versatilidad. Se anclará al microposicionador el componente que se desee desplazar en el montaje óptico y se ubicará de forma controlada en el punto adecuado sin necesidad de influir manualmente en el sistema, lo que alteraría el resultado de la medida. Para ello se utilizará un motor 'paso a paso' (PaP) con resolución de decenas de micrómetros. Los objetivos específicos relativos al diseño de los requerimientos del microposicionador dentro del sistema óptico se incluyen en el siguiente Capítulo.

1.3 ESTRUCTURA DEL DOCUMENTO

A lo largo de este documento se describirá el diseño, desarrollo e implementación del sistema microposicionador motorizado en un eje y las consideraciones necesarias para su realización en el Proyecto.

- En el Capítulo 2 se expone la motivación y objetivos para el desarrollo del Proyecto.
- En el Capítulo 3 se enumeran y explican todos los componentes mecánicos necesarios para la construcción y transformación de un microposicionador manual en otro motorizado.
- En el Capítulo 4 se justifica la elección de todos los componentes electrónicos necesarios para el diseño de la electrónica de control del sistema microposicionador motorizado. Se explica detalladamente la

funcionalidad cada uno de ellos y su ubicación en el esquemático del diseño final.

- En el Capítulo 5 se expone la programación del microcontrolador y el *software* para el PC que ayuda a testear el correcto funcionamiento del *hardware*. Mediante sencillos diagramas de flujo se describen las tareas que realizan las funciones implementadas en el programa.
- En el Capítulo 6 se realiza un estudio completo de calibración del sistema microposicionador.
- En el Capítulo 7 se describen las conclusiones obtenidas en el desarrollo de este Proyecto y las ampliaciones futuras, y se valoran sus prestaciones (ventajas e inconvenientes) frente a otros dispositivos comerciales.

Asimismo, se incluyen al final del documento los apartados normalizados: Listado de la bibliografía consultada, Anexos, Planos, Pliego de condiciones y Presupuesto. Se enumera a continuación el contenido de los Anexos y Planos por ser específicos del Proyecto desarrollado.

- Anexos:
 - Hoja de características del PIC18F2550.
 - Hoja de características del Driver L293D.
 - Hoja de características del Regulador LM7805.
 - Hoja de características de la Pantalla LCD 162D.
 - Hoja de características de la vagoneta-carril.
 - Código fuente del código del software del microcontrolador.
- Planos:
 - Circuitos impresos de la placa del motor.
 - Circuitos impresos de la placa de pulsadores.

o

CAPÍTULO 2

MOTIVACIÓN Y OBJETIVOS

2.1 INTRODUCCIÓN

Como se comentó en el Capítulo anterior, el objetivo general del Proyecto es el control del movimiento de una plataforma deslizante sobre un carril mediante el uso de un motor paso a paso (PaP). Esta plataforma deberá poder situarse en posiciones definidas a lo largo de la longitud del carril. Debido a que el motor PaP recorre sólo unas decenas de micrómetros en cada avance, el prototipo se denominará *posicionador micrométrico* o *microposicionador*. La aplicación específica de un microposicionador está asociada a un sistema de caracterización de magnitudes propias en dispositivos electroópticos, permitiendo un adecuado alineamiento y enfoque de los componentes ópticos.

En este sentido, existen en el mercado productos similares al posicionador micrométrico motorizado a diseñar, en los que la característica principal que define las prestaciones de su respuesta es la precisión en el control del movimiento. Una mayor precisión se traducirá en una mayor repetitividad del posicionamiento de forma correcta, es decir, la ubicación en las posiciones definidas irá asociada a una disminución del error. El uso de los posicionadores micrométricos se destina a una amplia gama de tareas: aplicaciones ópticas, aplicaciones biomédicas, fotónica en general.

Como muestra de ejemplo de microposicionadores comerciales, se han seleccionado un conjunto de microposicionadores de la serie T-LSR del fabricante Zaber con el objeto de estudiar sus prestaciones.

En la Figura 2.1, se listan varios ejemplos especificando el tamaño de los carriles (longitud del recorrido de la vagoneta) y el precio. Se observa que existen carriles cuyo tamaño varía desde 75mm hasta 450mm oscilando los precios entre 1750\$ y 2860\$.



T-LSR Series Motorized Linear Slides with Built-in Controllers

- 75, 150, 300 and 450mm travel
- 20kg load capacity
- Up to 80mm/s speed and up to 100N thrust
- Built-in controller; daisy chains with other T-series products

Follow the links below for more details or to generate a quote or online order.

Part Number	Description	Typical Lead Time	*Price without accessories (USD)
T-LSR075A	Motorized Linear Slide, 75mm travel, RS-232 plus manual control	3-5 Days	\$1,750.00 Info / Buy / Quote
T-LSR075B	Motorized Linear Slide, 75mm travel, RS-232 plus manual control	3-5 Days	\$1,750.00 Info / Buy / Quote
T-LSR075D	Motorized Linear Slide, 75mm travel, RS-232 plus manual control	3-5 Days	\$1,750.00 Info / Buy / Quote
T-LSR150A	Motorized Linear Slide, 150mm travel, RS-232 plus manual control	3-5 Days	\$1,970.00 Info / Buy / Quote
T-LSR150B	Motorized Linear Slide, 150mm travel, RS-232 plus manual control	3-5 Days	\$1,970.00 Info / Buy / Quote
T-LSR150D	Motorized Linear Slide, 150mm travel, RS-232 plus manual control	3-5 Days	\$1,970.00 Info / Buy / Quote
T-LSR300A	Motorized Linear Slide, 300mm travel, RS-232 plus manual control	3-5 Days	\$2,410.00 Info / Buy / Quote
T-LSR300B	Motorized Linear Slide, 300mm travel, RS-232 plus manual control	3-5 Days	\$2,410.00 Info / Buy / Quote
T-LSR300D	Motorized Linear Slide, 300mm travel, RS-232 plus manual control	3-5 Days	\$2,410.00 Info / Buy / Quote
T-LSR450A	Motorized Linear Slide, 450mm travel, RS-232 plus manual control	3-5 Days	\$2,860.00 Info / Buy / Quote

Figura 2.1. Listado ejemplo de posicionadores y precios

A través de una comunicación RS-232 y un software dedicado que se proporciona adjunto al dispositivo mecánico, se consigue la colocación correcta de la vagoneta sobre posiciones definidas en el carril. En particular, el error cometido o la desviación que hay entre la medida de la posición “real” y la posición “supuestamente real”, es variable dependiendo del modelo. Para el del carril de 75mm el máximo error cometido es de 12 micrómetros y para el de 450mm de 68 micrómetros.

A la vista de los ejemplos mostrados, se especifica a continuación la motivación y objetivos específicos para el diseño e implementación del sistema posicionador motorizado en un eje, estableciendo comparaciones con los ejemplos comerciales mostrados haciendo hincapié en los aspectos más convenientes.

2.2 MOTIVACIÓN

La principal motivación para el diseño del microposicionador es la necesidad de cubrir la ausencia de sistemas de desplazamiento en un eje con control del avance y largo recorrido, a un coste razonable.

Existen posicionadores mecánicos en los Laboratorios de Electrónica de la Universidad Carlos III de Madrid, pero su longitud de desplazamiento resulta limitada para algunas aplicaciones. Tal es el caso de la captura de las vistas en una aplicación de visión estereoscópica en la cual la desviación angular de las imágenes puede precisar de recorridos de la vagoneta del orden de decenas de centímetros.

Así mismo, se desea dotar de cierta autonomía al sistema, a diferencia de los posicionadores motorizados existentes en el mercado. Todos ellos necesitan el uso de un *software* de control sobre el dispositivo, lo que hace imprescindible la utilización de un ordenador. En este Proyecto se pretende conseguir un dispositivo que funcione con y sin ordenador, siendo el propio usuario el que decida cómo quiere trabajar.

Finalmente, el coste final del dispositivo será un aspecto muy importante en el diseño e implementación. Como se ha mostrado en los ejemplos anteriores, los precios de los posicionadores son bastante elevados en todos los casos. En este Proyecto se pretende construir un posicionador motorizado con prestaciones similares a los que actualmente se comercializan en el mercado y con una precisión del mismo orden de magnitud, pero a un coste inferior.

2.3 OBJETIVOS

Establecido el objetivo general de este Proyecto, se describen a continuación los objetivos específicos que guiarán el desarrollo del mismo. Se ha optado por

plantear estos objetivos en una lista que pretende resumir todos los aspectos considerados en el diseño e implementación del microposicionador:

- ✓ Diseñar un sistema microposicionador basado en un carril y una vagoneta que inicialmente posee movimiento manual.

- ✓ Dotar al sistema de dos modos opcionales de funcionamiento: autónomo y a través de PC. En los sistemas comerciales es habitual el control obligatorio a través de un programa que se controla desde el PC.

- ✓ En el modo autónomo, darle cierta autonomía al sistema. Para conseguirlo es preciso incluir unos pulsadores a modo de cursores para desplazar la vagoneta sobre el carril y una pantalla LCD para visualizar la posición en la que se encuentra. Ofrecer, a través de los pulsadores, la funcionalidad para trabajar con desplazamientos pequeños (del orden de micrómetros) o con desplazamientos grandes (del orden de centímetros). Permitir, a través de la pantalla LCD de 16 × 2 (16 columnas y 2 filas), la visualización en todo momento de la posición actual de la vagoneta en el carril.

- ✓ En el modo a través de PC, incluir en el sistema un puerto USB para establecer la comunicación entre el microcontrolador y el ordenador. De esta manera se pueden variar posiciones desde el teclado del ordenador y éste las transmite al microcontrolador. Así mismo, a través del puerto USB es posible capturar información y trasladarla al ordenador

- ✓ Diseñar las placas electrónicas de circuito impreso del circuito total. Las placas PCB son láminas duras de cierto espesor sobre la que se colocan

componentes electrónicos que están conectados entre sí a través de pistas o rutas de un material conductor. El proceso de grabado fotolitográfico de las placas correrá de cargo del Servicio Técnico con que cuenta la Universidad Carlos III. Se fabricarán dos placas PCB, una para el motor y otra para los pulsadores. La placa del motor controlará el desplazamiento del mismo y éste, a su vez, el de la vagoneta sobre el carril. La placa de pulsadores permitirá desplazarse a las posiciones deseadas.

- ✓ Programar el código de control del posicionador. Desarrollar el código de comunicación entre módulos: placa de control del motor, placa de control de los pulsadores externos de desplazamiento, y comunicaciones con el PC mediante conexión USB.

- ✓ Calibrar el prototipo final del sistema microposicionador motorizado en un eje:
 - Medir su resolución, es decir, la traducción del paso del motor en distancia física.
 - Medir su precisión, es decir, la capacidad de ubicarse en posiciones absolutas sobre la longitud del carril.
 - Comparar ambas medidas con las especificaciones para un dispositivo comercial.

- ✓ Elegir los componentes adecuados para la implementación *hardware* del diseño. Varios de los componentes del diseño se encuentran disponibles en el laboratorio de investigación del Grupo de Displays y Aplicaciones Fotónicas de la Universidad Carlos III: motor PaP, correa-tambor, pantalla LCD y detectores de fin de carril. La disponibilidad de dichos componentes supondrá una reducción significativa de coste del prototipo final.

CAPÍTULO 3

SISTEMA MECÁNICO MOTORIZADO

3.1 INTRODUCCIÓN

Ya se ha comentado que en el Laboratorio de Electrónica de la Universidad Carlos III se encuentra disponible un motor PaP (‘paso a paso’), que ha sido extraído de un escáner obsoleto y en desuso. En este Capítulo se hará un breve análisis sobre los motores paso a paso (su principio de funcionamiento, control, tipos, etc.) que son utilizados en gran número de aplicaciones. Junto a esta descripción, se mostrarán la plataforma deslizante (vagoneta) y el carril escogidos para la implementación del microposicionador en un eje.

En la Figura 3.1 se muestra un motor ‘paso a paso’.



Figura 3.1. Ejemplo de motor ‘paso a paso’

3.2 CARACTERÍSTICAS DE LOS MOTORES ‘PASO A PASO’

Los motores ‘paso a paso’ son adecuados para mecanismos en los que se demanden movimientos de alto grado de precisión, ya que con ellos se consigue un posicionamiento fijo y fiable. Por cada impulso eléctrico se consigue el movimiento del rotor del motor sobre su eje, que se desplaza cierto ángulo también llamado *paso*.

El tamaño del paso puede variar desde grandes movimientos como por ejemplo 90° , donde con 4 pasos se recorre un giro completo, hasta pequeños movimientos de $1,8^\circ$, en los que hacen falta 200 pasos para completar 360° .

Estos motores pueden quedar fijados en posiciones concretas, o bien totalmente libres. Mientras cualquiera de sus bobinas esté energizada, el motor estará ubicado en la posición que corresponda, y estará completamente libre si no circula corriente por ninguna de sus bobinas.

Existen gran cantidad de aplicaciones que utilizan motores paso a paso. Se trata de casos donde es esencial un posicionamiento fijo y preciso de algún elemento. Como ejemplos cabe destacar: posicionadores, impresoras, unidades de disquetera en ordenadores, plataformas inerciales, movimiento de cámaras y antenas en satélites o telescopios, aplicaciones médicas, etc.

3.3. PRINCIPIO DE FUNCIONAMIENTO DE LOS MOTORES 'PASO A PASO'

Estos motores están constituidos por bobinas excitadoras arrolladas independientemente sobre su estator y distintos imanes permanentes situados sobre un rotor. Si se hacen circular impulsos de corriente por una o varias bobinas del estator se crea un campo electromagnético en el rotor (ver Figura 3.2).

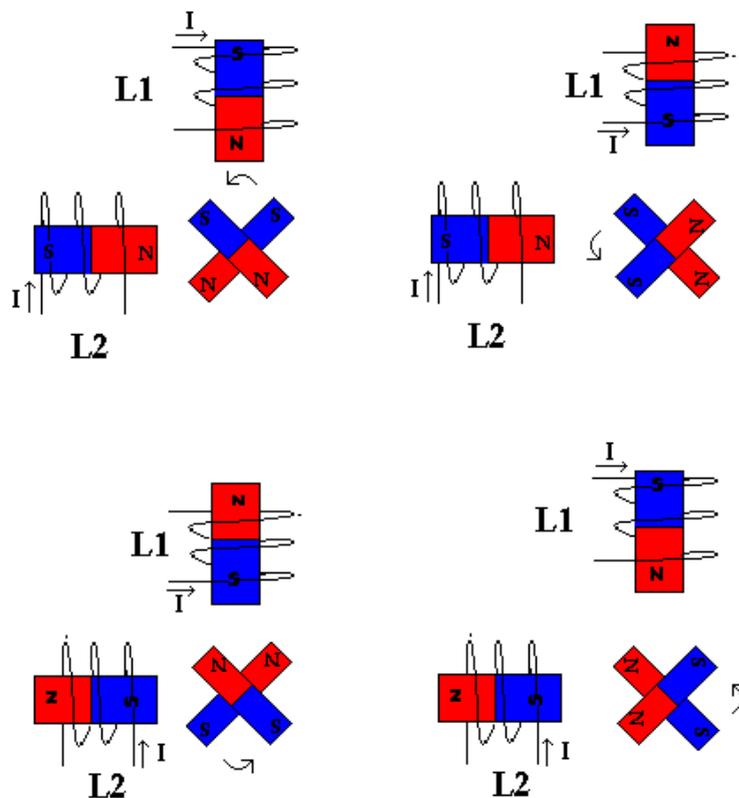


Figura 3.2. Principio de funcionamiento de un motor 'paso a paso'

El rotor tiene su propio campo magnético, con su propia orientación pero al circular corriente por las bobinas del estator, hará que el rotor se oriente buscando la posición de equilibrio magnético dentro del campo del estator. De esta forma el rotor girará sobre su eje a una nueva posición.

Si al alcanzar el rotor la nueva posición de equilibrio el estator cambia la orientación de sus polos, provocará que el rotor se mueva nuevamente en busca de una nueva posición de equilibrio.

Si se mantiene esta forma de actuar de forma continuada, se conseguirá un movimiento giratorio y continuo del rotor y se transformará la energía eléctrica en otra mecánica en forma de movimiento circular.

Como se puede apreciar en la Figura 3.2, si se van excitando las bobinas del estator, se puede hacer girar al eje, en el instante deseado, un ángulo determinado.

3.4 TIPOS DE MOTORES `PASO A PASO`

Desde el punto de vista de su construcción, existen tres tipos básicos de motores paso a paso: de imán permanente, de reluctancia variable e híbridos. Los de imán permanente son, a su vez, de dos tipos: unipolares y bipolares.

3.4.1 Motores de imán permanente

En este tipo de motores el rotor está formado por un imán permanente y, rodeándolo, una pieza mecanizada con forma de dientes, abrazados por bobinas que constituyen las fases (ver Figura 3.3).

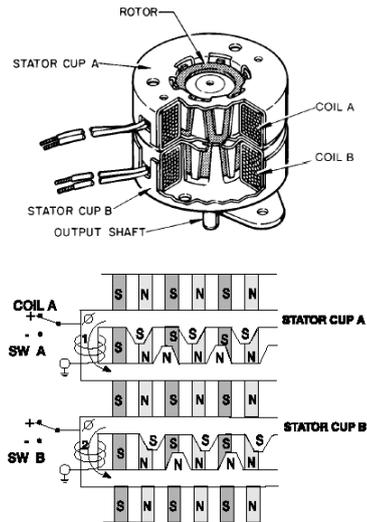


Figura 3.3. Motor `paso a paso´ de imán permanente

Al estar el rotor permanentemente magnetizado, se crea un campo magnético con una orientación determinada. Al mismo tiempo el estator crea otro campo magnético, rodeando al del rotor, con lo que el rotor tendrá que buscar la siguiente posición de equilibrio magnético. De esta forma se consigue el movimiento del rotor hasta una nueva posición.

Según cambie la excitación del estator se producirá movimiento en el rotor y se logrará posicionar el eje del motor con gran precisión. Como se aprecia en la Figura 3.4, al variando la polaridad aparecen fuerzas de atracción-repulsión generadas por el campo magnético. Así el campo fijo del imán permanente del rotor se ve atraído en cada paso a una nueva posición.

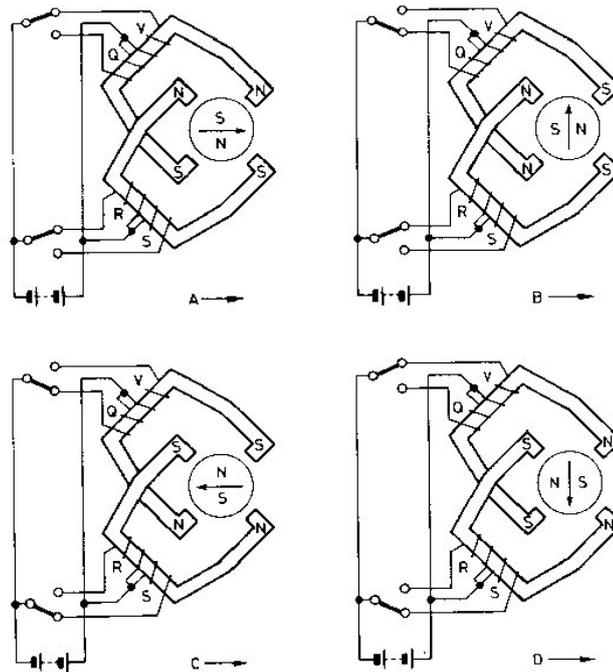


Figura 3.4. Secuencias de movimiento del motor según la alimentación del estator

Una de las ventajas de utilizar este tipo de motores es que garantizan una gran reducción en las pérdidas eléctricas. Esto hace que sea el tipo de motores paso a paso más comunes en áreas como la robótica. Dependiendo del tipo de bobinas que se encuentran devanadas sobre el estator (y, por tanto, del modo de crear el campo giratorio) los motores de imán permanente se pueden dividir en:

- **Bipolares.** Constan de dos bobinas independientes por las que puede circular corriente en ambos sentidos según la secuencia definida. Mientras no circule corriente por las bobinas, no se moverá el motor.

Cambiando el sentido de la corriente se invertirá la polaridad de las bobinas, de ahí el nombre de bipolar. Al producirse un cambio en la polaridad de las bobinas se provoca el movimiento del eje, avanzando éste un paso. Para invertir las corrientes se utiliza un circuito de control del motor, y a través de cuatro cables de salida se controlan las bobinas (ver Figura 3.5).

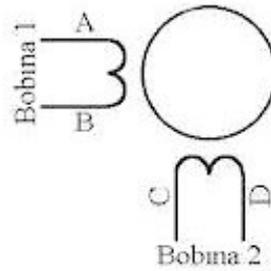


Figura 3.5. Detalle de las bobinas en un motor bipolar

- **Unipolares.** Están compuestos por dos bobinas con una toma intermedia cada una, estando conectada esta toma intermedia a un terminal de alimentación (ver Figura 3.6). De este modo, la intensidad de la corriente siempre circula en el mismo sentido a través de la bobina, de ahí el nombre de unipolar. Una vez alimentada la toma intermedia, se aplica una secuencia de pulsos a un extremo de la bobina y al otro, y se consigue el movimiento del motor. Generalmente tienen 5 ó 6 cables de salida.

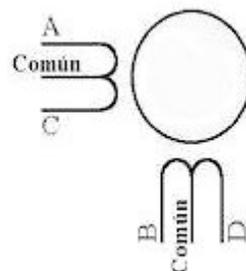


Figura 3.6. Detalle de las bobinas en un motor unipolar

3.4.2 Motores de reluctancia variable

Los motores de reluctancia variable están formados por un estator laminado, y por un rotor sin excitar, de un material ferromagnético de alta permeabilidad. Tanto el estator como el rotor tienen su superficie dentada.

En la Figura 3.7 se muestra un motor de reluctancia variable.



Figura 3.7. Imagen de un motor de reluctancia variable

El rotor se sitúa facilitando el camino del campo magnético generado por las bobinas de estator, buscando el punto de reluctancia mínima. Este punto será el punto de equilibrio entre el estator y el rotor. Si se modifica el campo en el estator, el rotor girará buscando el siguiente punto de equilibrio.

Su principal ventaja es que puede funcionar con pasos más pequeños que los de imán permanente, mientras que su principal inconveniente es que da menos par.

3.4.3 Motores híbridos

Este motor es una mezcla de los dos anteriores, logrando un alto rendimiento a buena velocidad. El rotor está formado por anillos de acero dulce dentado, donde el número de dientes es levemente distinto al del estator; dichos anillos están montados sobre un imán permanente dispuesto axialmente.

En la Figura 3.8 se muestra un motor híbrido.



Figura 3.8. Imagen de un motor híbrido

Los motores híbridos suministran importantes pares de accionamiento, un gran número de pasos por vuelta y una frecuencia de trabajo elevada.

3.5. MODOS DE FUNCIONAMIENTO DE LOS MOTORES DE IMÁN PERMANENTE

En este Apartado se considerarán únicamente los motores de imán permanente, por ser los más comunes y es necesario distinguir, una vez más, entre los unipolares y los bipolares. Los motores bipolares se controlan de forma más simple, ya que sólo requieren que se invierta el sentido de la corriente que circula por sus bobinas. Cada inversión provoca en el eje un movimiento de 1 paso. Los motores unipolares pueden tener tres modos de operación: secuencia normal, modo onda y medio paso.

3.5.1 Motores unipolares de secuencia normal

La secuencia normal es la más usada y la que generalmente recomiendan los fabricantes. En esta secuencia se excitan las bobinas de dos en dos, en cada ocasión. El motor avanza 1 paso por vez y, debido a que siempre hay dos bobinas activadas, se obtiene un alto par en cada paso y retención.

Tiene el inconveniente de que hay que mantener siempre activadas dos bobinas para el desplazamiento de 1 paso y, por tanto, consume mucha corriente.

En la Figura 3.9 se muestran los 4 movimientos que permiten dar la vuelta completa al motor.

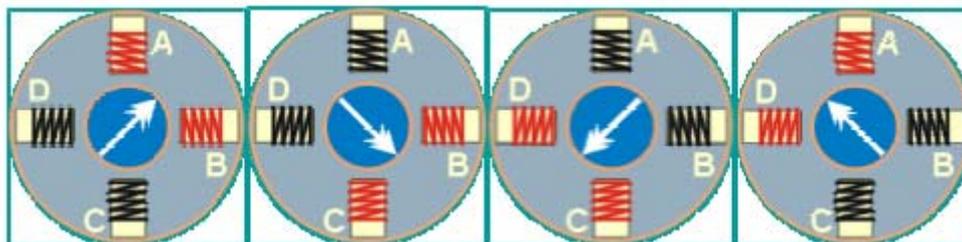


Figura 3.9. Posicionamiento de un motor unipolar según una secuencia de activación normal

3.5.2 Motores unipolares de modo onda

En este caso sólo se activa una bobina en un momento indicado; por tanto, la corriente consumida es la mitad que en modo normal. Al activarse solo una bobina se produce un movimiento más suave y eficiente; sin embargo el par es menor, y al estar el rotor alineado con un único polo, se pierde fiabilidad y precisión en el paso.

Para aplicaciones en las que no se requiera un alto grado de precisión es aconsejable la elección de este modo de funcionamiento, ya que el calentamiento del motor por el paso de corriente es considerablemente menor que en los otros casos.

En la Figura 3.10 se muestran los 4 movimientos que permiten dar la vuelta completa al motor.

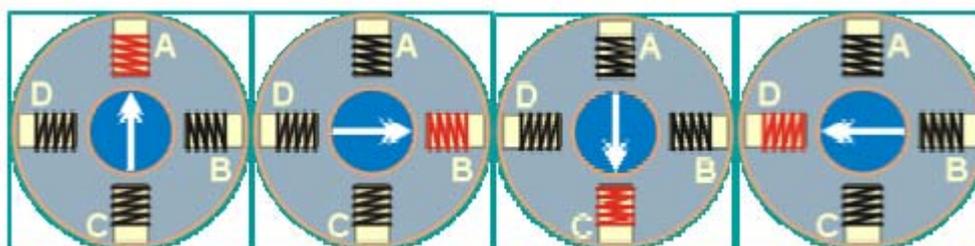


Figura 3.10. Posicionamiento de un motor unipolar operando en modo onda

3.5.3 Motores unipolares de medio paso

En este caso se consigue un movimiento igual a la mitad del paso real. Para ello se excitan primero dos bobinas y luego una sola, y así sucesivamente. Por tanto, para que el motor dé una vuelta completa debe realizar 8 movimientos en lugar de 4, como se muestra en la Figura 3.11.

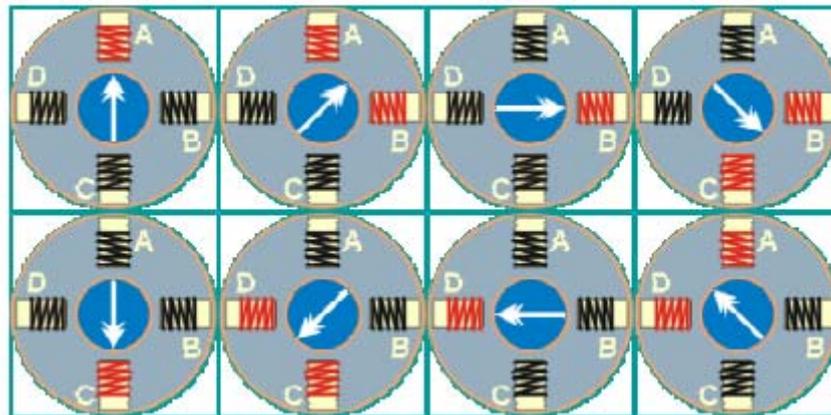


Figura 3.11. Posicionamiento de un motor unipolar de medio paso

Ya que en cada movimiento se excitan una o dos bobinas, el par variará. Esto creará un paso fuerte y otro débil. Con este modo se gana en número de posiciones (para satisfacer la necesidad de incrementos más pequeños), pero se pierde precisión y par con respecto al modo normal.

3.6. CONTROL DEL MOTOR

El control de un motor paso a paso sólo se puede realizar a través de un dispositivo que genere la secuencia de pulsos adecuada en los bobinados. Una forma sencilla sería mediante un PC, una interfaz con salidas de nivel TTL y una etapa de amplificación.

Para el excitado de las bobinas se ha de proporcionar la corriente necesaria, por lo que es necesario diseñar una sencilla etapa de potencia. En este caso se puede optar por el uso de circuitos integrados. Por ejemplo, el integrado L293D, es un driver de corriente formado por transistores en configuración Darlington. Sus salidas se pueden conectar directamente a las fases del motor y proporcionarle los pulsos de corriente.

Otra posibilidad es sustituir el PC por un dispositivo programado a tal efecto. Puede tratarse de una tarjeta de Entrada/Salida, una placa controladora, o simplemente un módulo que contenga una memoria programable y unos canales de salida. Se puede descargar en la memoria del controlador el programa necesario para controlar el motor, consiguiendo más autonomía en el sistema, sin tener que controlar manualmente las órdenes a través del PC.

3.7. ELECCIÓN DE COMPONENTES MECÁNICOS

Debido a que los motores paso a paso son dispositivos mecánicos, deben vencer ciertas inercias, por lo que el tiempo y la frecuencia de los pulsos aplicados son puntos muy importantes a tener en cuenta. Si la frecuencia de pulsos es muy elevada, el motor puede reaccionar no realizando ningún movimiento en absoluto, comenzando a vibrar pero sin llegar a girar, girando erráticamente o incluso puede llegar a girar en sentido contrario. Por tanto, el motor debe dar un paso antes de que comience la siguiente secuencia de pulsos.

Por otra parte, si el control del movimiento del motor se realiza mediante un microcontrolador, es preciso tener en cuenta los ciclos de trabajo de cada uno. El microcontrolador hace muchas operaciones por segundo; sin embargo el motor tarda cierto tiempo desde que recibe la señal de giro hasta que coge la energía suficiente para mover el rotor. Para evitar problemas con esta situación, para la implementación del sistema motorizado que se describe en este proyecto, se ha considerado pertinente añadir un retraso en el movimiento del motor mediante *software*; así, cada vez que termine una secuencia, esperará 5 milisegundos antes de responder a la siguiente secuencia.

En la Figura 3.12 se muestra una imagen del motor que se ha utilizado en la implementación del Proyecto.



Figura 3.12. Motor utilizado para la implementación del sistema motorizado en un eje

Para identificar el tipo de motor disponible en el Laboratorio, se han realizado las comprobaciones pertinentes, descritas anteriormente, y se ha concluido que se trata de un motor 'paso a paso' de tipo unipolar.

En la Figura 3.13 se muestra el circuito de control del motor.

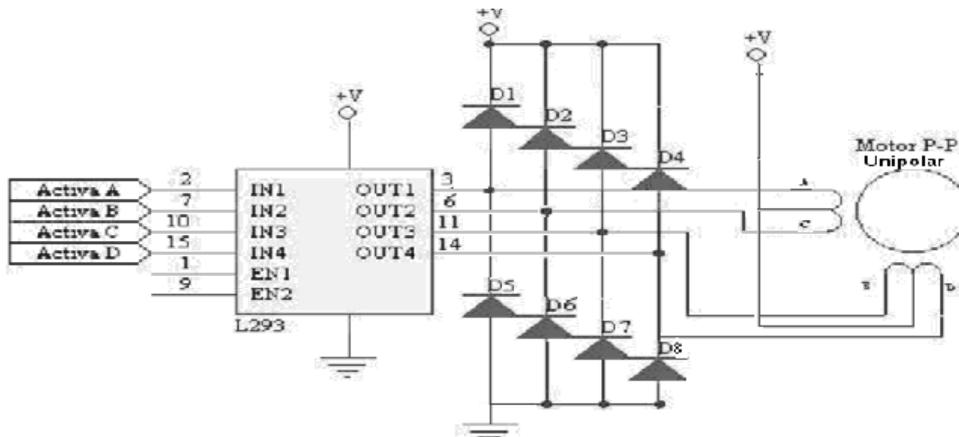


Figura 3.13. Circuito de control del motor con integrado unipolar disponible

La secuencia de giro a programar mediante el microcontrolador se muestra en la Tabla 3.1:

PASO	TERMINALES			
	A	B	C	D
1	ON	ON	OFF	OFF
2	OFF	ON	ON	OFF
3	OFF	OFF	ON	ON
4	ON	OFF	OFF	ON

Tabla 3.1. Secuencia de control de giro

Así mismo, se ha identificado a qué bobina corresponde cada uno de los cinco cables del motor accesibles desde el exterior (cables de colores: rojo, amarillo, negro, marrón y naranja).

El método seguido para dicha identificación se explica a continuación. Primeramente, se ha utilizado un polímetro para medir la resistencia entre pares de cables, ya que el cable común será el único que tenga la mitad del valor de la resistencia que el resto de cables. Se ha obtenido un valor de 53Ω para el cable rojo y de 107Ω para el resto de los cables.

Seguidamente, se selecciona el cable amarillo, al que se denomina “cable A” y se conecta a masa. Manteniendo este cable conectado a masa, se prueba cuál de los tres cables restantes provoca un paso del motor en sentido antihorario al ser conectado también a masa, resultando ser el de color negro, al que se le llama “cable B”.

Realizando la misma operación descrita anteriormente, se descubre que el cable marrón provoca un paso del motor en sentido horario, y se le denomina “cable D”.

El cable restante, de color naranja, debería ser el “cable C” y, para comprobarlo, basta conectarlo a masa y comprobar que no genera movimiento alguno, debido a que es la bobina opuesta a A.

En la Tabla 3.2 se muestra un resumen de los cables con los colores correspondientes y el nombre con que se han asignado al controlador.

Bobina	Color cable	Secuencia
A	Amarillo	A1
B	Negro	A2
C	Naranja	B1
D	Marrón	B2
Común	Rojo	

Tabla 3.2. Relación bobina-nombre asignado

Una vez conocidas las características del motor, se escogió una vagoneta y un carril para construir con el conjunto de las tres piezas el posicionador objeto del Proyecto.

3.7.1. Conjunto vagoneta - carril

Se ha escogido una vagoneta muy robusta, con lo que se evitan posibles deformaciones por el peso que puedan influir en la medida. Otro de los aspectos que ha influido en la elección de la vagoneta es que cuenta con 4 orificios donde poner las muestras y son de métrica M6. En este sentido es compatible con multitud de postes y soportes de óptica disponibles en el Laboratorio de Electrónica de la Universidad Carlos III, en el cual se desarrolla el Proyecto.

Por otra parte, se ha seleccionado un carril que sea compatible con la vagoneta y de longitud 300mm para su uso en aplicaciones de caracterización de dispositivos electroópticos que requieran recorridos grandes.

En la Figura 3.14 se muestra el conjunto vagoneta-carril de la empresa de componentes RS (311-3705 y 311-3682).

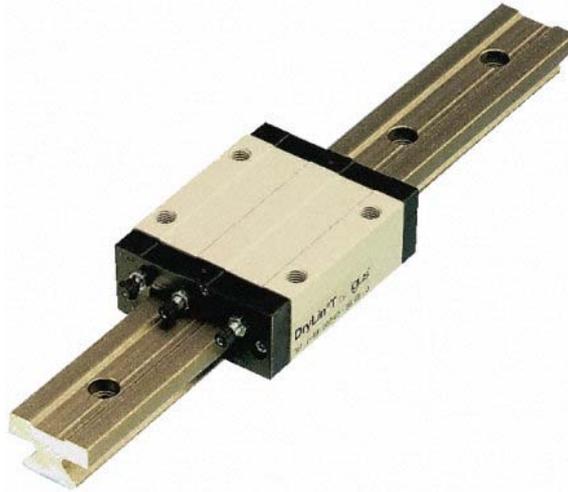


Figura 3.14. Imagen del conjunto vagoneta-carril

3.7.2. Conjunto completo: vagoneta - carril - motor

El conjunto completo vagoneta – carril – motor se dispone sobre dos placas metálicas compradas para este fin. Estas placas presentan la ventaja de que están agujeradas con orificios de métrica M6, además de disponer también de orificios de métrica M10. Estos tamaños coinciden con los de los orificios que tiene la mesa óptica (M6) y con los 4 orificios que tiene el carril (M10).

Por tanto, en los orificios de métrica M6 se ha colocado el motor y la estructura completa se ha fijado a la mesa óptica. En los orificios de métrica M10 se ha fijado la placa metálica al carril y con esto se ha conseguido que siempre esté estirada la correa.

En la Figura 3.15 se muestra el aspecto definitivo del sistema microposicionador., En particular, se muestra el detalle del subsistema mecánico.



Figura 3.15. Imagen del subsistema mecánico del microposicionador

CAPÍTULO 4

HARDWARE DE CONTROL DEL SISTEMA MOTORIZADO

4.1 DESCRIPCIÓN DE LOS BLOQUES DEL SISTEMA

En este Capítulo se dará una visión general de los componentes específicos (*hardware*) seleccionados para la implementación del sistema de posicionamiento lineal.

En la Figura 4.1 se muestra el diagrama de bloques para el sistema completo.

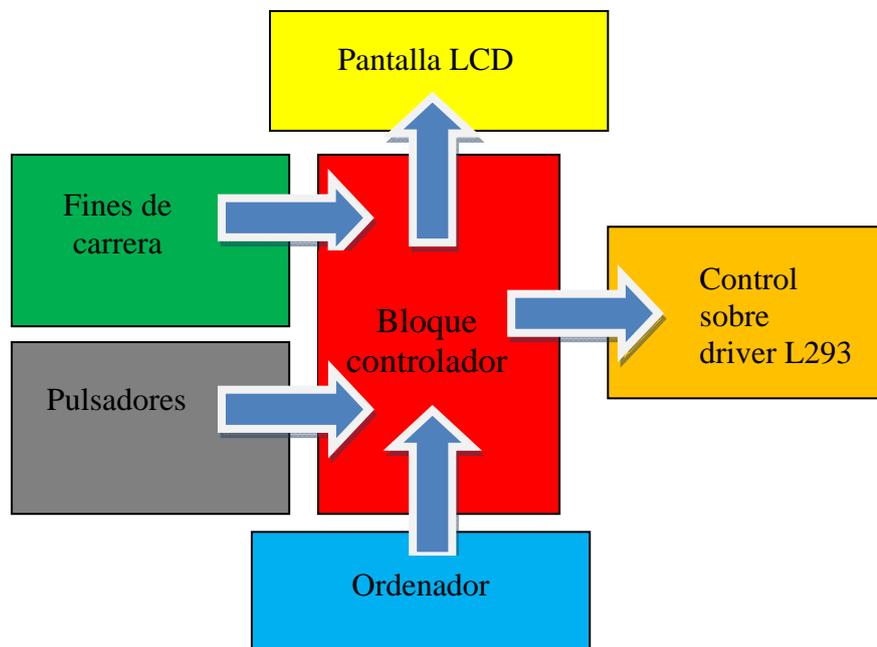


Figura 4.1. Diagrama de bloques del sistema completo

Como ya se comentó en Capítulos anteriores, se ha desarrollado un sistema de posicionamiento a través de una plataforma que desliza sobre un carril. El posicionamiento de la plataforma se puede conseguir mediante el ordenador o a través de la placa de pulsadores. Estos elementos le dan las instrucciones al bloque controlador y éste, a su vez, actuará sobre el driver L293 que pondrá en marcha el motor. Los fines de carrera adosados al carril, al activarse, indicarán al bloque controlador que se ha llegado al final del carril para que se detenga el motor. Además de lo ya mencionado, desde el bloque controlador se mostrará en la pantalla LCD la posición en que se encuentra la vagoneta.

4.2 BLOQUE CONTROLADOR

Un controlador es un dispositivo que controla uno o varios procesos. Antiguamente se construían únicamente con dispositivos de lógica discreta y más tarde se usaron los microprocesadores, a los que se adjuntaban chips de memoria y módulos de Entrada/Salida sobre circuito impreso. En la actualidad, todos estos elementos se incluyen en un único chip, que se llama *microcontrolador*.

Desde la aparición del circuito integrado se ha producido un desarrollo constante de la electrónica digital, dando lugar a dispositivos cada vez más sofisticados, entre ellos, los microprocesadores y los microcontroladores. Si se va a trabajar con aplicaciones sencillas son aconsejables las soluciones no programables que no requieren desarrollo de *software*. Escribir *software* consume mucho tiempo y es, por tanto, más laborioso, por lo que en aplicaciones sencillas se puede utilizar sólo el *hardware*. En cambio, según se incrementa la complejidad del sistema, aumentan las ventajas del uso de sistemas programables. Una de las principales ventajas es que permiten actualizar el funcionamiento del *software* sin tener que volver a diseñar el *hardware*.

Es esta flexibilidad para actualizar con facilidad y economía el *software* la que hace que el bloque controlador esté basado en un sistema reprogramable.

4.3 SOLUCIONES PARA DESARROLLAR EL BLOQUE CONTROLADOR

En las siguientes líneas se describen las posibles soluciones para desarrollar el bloque controlador, siempre en base a sistemas reprogramables, haciendo hincapié en sus principales ventajas e inconvenientes. Se describen someramente los dispositivos FPGA, los microprocesadores y los microcontroladores.

4.3.1 Dispositivo FPGA

Las FPGA's son dispositivos lógicos programables por los usuarios mediante lenguajes de programación como pueden ser VHDL y Verilog, entre otros, compuestos de bloques lógicos comunicados por medio de una matriz de cables y *switches* programables. Mediante la lógica programable se pueden realizar, desde funciones sencillas, como por ejemplo una puerta lógica, hasta sistemas más complejos en un chip.

En la Figura 4.2 se muestra el esquema básico de una FPGA con los bloques lógicos configurables (CLBs), las interconexiones y los bloques de Entrada/Salida (IOB).

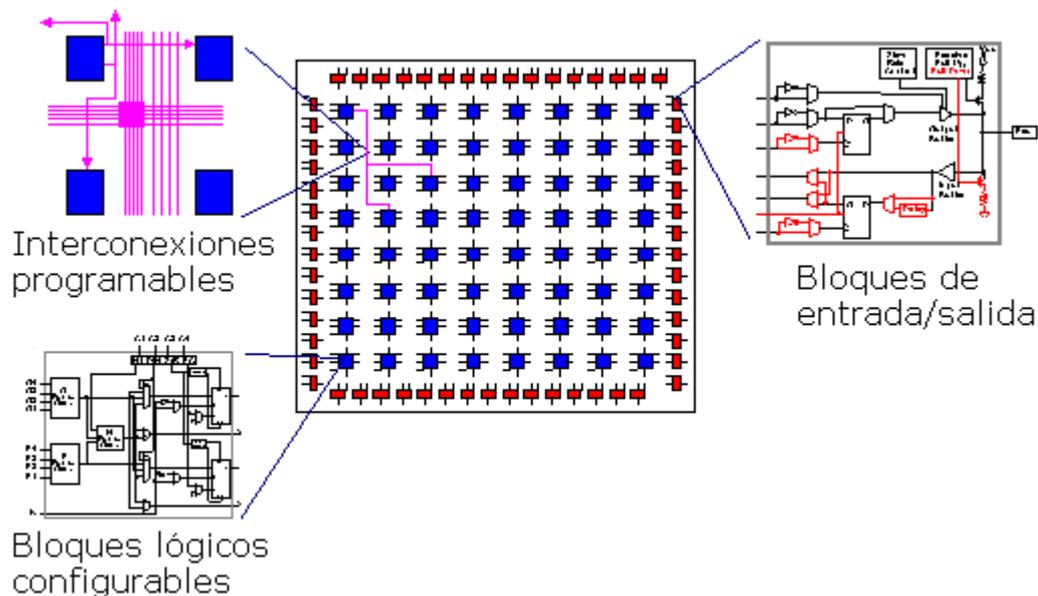


Figura 4.2. Diagrama interno de una FPGA

En la Tabla 4.1 se reflejan las ventajas e inconvenientes de la utilización de este dispositivo como bloque controlador.

Ventajas	Inconvenientes
Bajo coste de las herramientas de programación.	Tamaño del chip, por su elevado número de entradas y salidas.
Verificación efectiva del diseño mediante simuladores o en el chip.	Coste elevado en comparación con otros elementos que pueden realizar la función de controlador.
Permite reemplazar gran cantidad de circuitos digitales integrados con un solo chip y agilizar las etapas de desarrollo, prototipado y aplicación.	Alto consumo de potencia, aunque se han desarrollado algunas familias de bajo consumo.
Funcionan a altas velocidades.	

Tabla 4.1. Ventajas e inconvenientes de utilizar una FPGA

Debido a que es un dispositivo muy potente y con un número muy elevado de Entradas/Salidas, que no son aprovechables para este sistema, no se ha considerado adecuado para desempeñar la función de controlador.

4.3.2 Dispositivo microprocesador

Un microprocesador está compuesto por los siguientes componentes: la *unidad aritmético-lógica* (ALU, siglas en inglés), *registros*, *unidad de control* y *memoria local*.

- La *unidad aritmético-lógica* efectúa operaciones aritméticas como puede ser una suma.
- Los *registros* son memorias especiales para almacenar datos temporalmente.
- La *unidad de control* es el elemento que se encarga de sincronizar cada una de las acciones que realiza la CPU.
- Los *buses* son los encargados de transmitir datos entre los componentes de un ordenador.
- La *memoria local* se emplea para los cálculos realizados en el mismo chip.

Los pines de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, permitiéndole conectarse con las memorias y módulos de Entrada/Salida. La necesidad de realizar estas conexiones es el principal inconveniente, por lo que tampoco se ha elegido este dispositivo. Sin embargo, su tamaño, y también su coste, son reducidos en comparación con las dimensiones y el coste de una FPGA.

4.3.3 Dispositivo microcontrolador

Aunque sus prestaciones son limitadas, un microcontrolador es un sistema completo: microprocesador, memoria y periféricos de entrada salida (ver Figura 4.3).

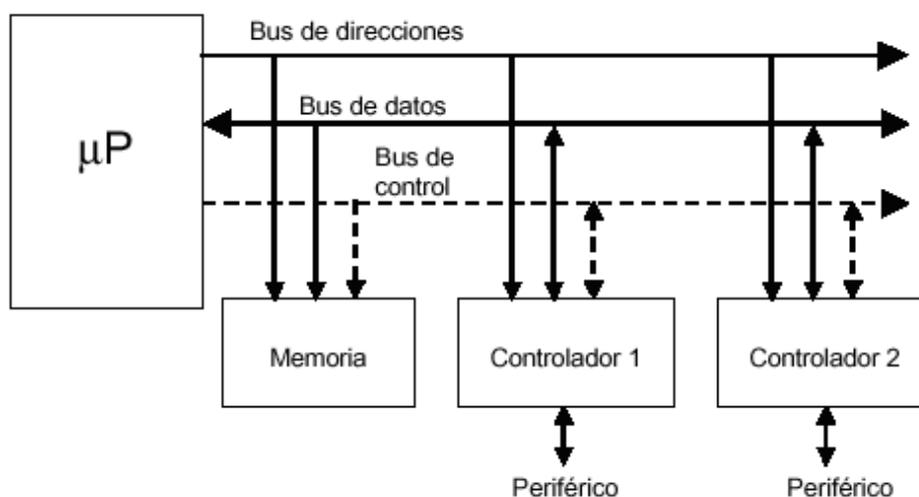


Figura 4.3. Microcontrolador

Los microcontroladores disponen generalmente también de una gran variedad de dispositivos de Entrada/Salida, como convertidores de datos (DAC y ADC), temporizadores, UART y buses de interfaz serie especializados, como I2C y CAN, entre otros. De este dispositivo cabe destacar que tiene una capacidad de almacenamiento alto y un coste reducido, y su tamaño es considerablemente pequeño, siendo éstas algunas de las características que se están buscando para el desarrollo de este Proyecto.

Después de haber estudiado y valorado las distintas opciones de las que se dispone para implementar el controlador, se ha optado por un microcontrolador, por las siguientes razones:

- Para este diseño, cuyo requerimiento en tareas computacionales es pequeño, se descarta el uso de un microprocesador.
- Se requiere que incorpore un puerto USB y pocas líneas de Entrada/Salida, por lo que el uso de la FPGA ya no es factible.
- En relación al Proyecto que se trata en este documento, las necesidades del microprocesador no son muy exigentes, ya que no es necesario realizar múltiples tareas ni operaciones excesivamente complejas.
- Las necesidades mínimas que debe soportar no se basan en el número de instrucciones por segundo que pueda ejecutar la CPU, sino en sus conexiones externas.

4.4 ELECCIÓN DEL MICROCONTROLADOR PIC18F2550

La distinción más importante entre microcontroladores que se puede realizar, se basa en el ancho de palabra que maneja el dispositivo, teniendo microcontroladores de 4, 8, 16 y 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son mucho más elevadas, son los microcontroladores de 8 bits los que realmente dominan el mercado. La razón de esta tendencia es que los microcontroladores de 8 bits son apropiados para la gran mayoría de aplicaciones, lo que hace innecesario emplear micros más potentes y, consecuentemente, más caros y complejos. A la hora de elegir el microcontrolador que se va a utilizar hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y el precio de éstas, el número de fabricantes que lo producen y, por supuesto, las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, puertos de comunicaciones etc.).

Como ya se ha explicado, un microcontrolador es un sistema cerrado de reducido tamaño en el que todos los bloques lógicos que lo componen están contenidos en su interior, siendo éstas las características por las que ha sido elegido este tipo de dispositivo como controlador del sistema que se va a desarrollar.

De todas las posibilidades que ofrece el mercado de microcontroladores se ha optado por la utilización de una arquitectura diseñada en el año 1989, la familia de microcontroladores tipo RISC (del inglés *Reduce Instruction Set Computer*), en concreto un modelo basado en el PIC18F2550, un microcontrolador del fabricante *Microchip Technology* encapsulado de 28 pines. Esta elección se basa en que cumple con todas los requisitos del sistema de comunicación USB, además de disponer en el Laboratorio de todas las herramientas necesarias para su programación (*software* y programador).

4.4.1 Componentes de un microcontrolador

A continuación se describen las características principales constructivas en las que se basa un microcontrolador general y las específicas para el modelo de microcontrolador elegido.

En la Figura 4.4 se muestra un diagrama simplificado de comunicaciones de un microcontrolador con sus diferentes bloques lógicos.

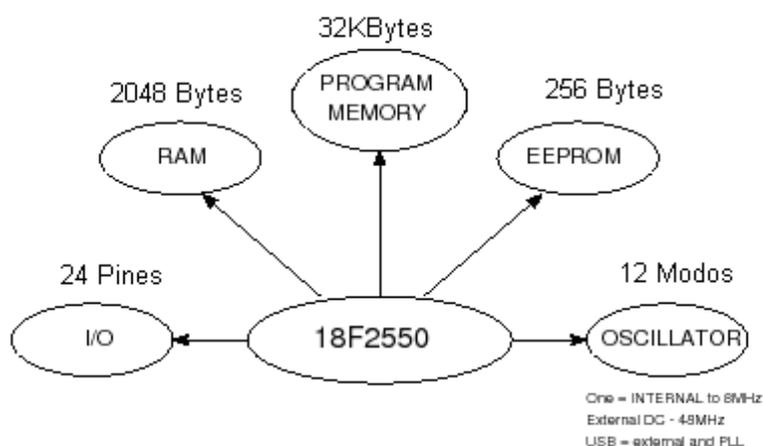


Figura 4.4. Componentes del microcontrolador

- **Procesador.**

Es el componente más significativo en el microcontrolador tanto a nivel *hardware* como *software*. Actúa como Unidad Central del Proceso controlando todas las operaciones. Se encarga de direccionar la memoria de instrucciones, recibe la instrucción en curso, la decodifica y la ejecuta. Una vez realizado esto, almacena el resultado. En este diseño, el microcontrolador es un PIC18F2550 (Controlador de Interfaz Periférico).

- **Memoria de programa.**

Es un bloque lógico funcional donde se almacenan todas las instrucciones del programa de control. Este tipo de memoria es de sólo lectura, aunque programable para poder cargar el nuevo código mediante un *hardware* externo denominado *programador*. Para el modelo en concreto seleccionado para este Proyecto, se dispone de 32kbytes de memoria de programa Flash con una vida de reprogramación de más de 100.000 ciclos.

- **Memoria de datos.**

Los programas varían continuamente los datos con los que trabajan, y esto exige poder almacenarlos de forma temporal. Para el Proyecto que se va a desarrollar se tienen 2kbytes de memoria RAM interna.

- **Líneas de Entrada y Salida.**

Las líneas de entrada y salida (Entrada/Salida) permiten comunicar al procesador con otros dispositivos. Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de Entrada/Salida se destinan a proporcionar el soporte a las señales de entrada, de salida y de control. En concreto para el PIC18F2550 se dispone de 24 líneas de Entrada/Salida programables.

- **Reloj principal.**

Todos los microcontroladores disponen de un circuito oscilador que genera una señal periódica de alta frecuencia, la cual genera los impulsos para la sincronización de todas las operaciones del sistema.

Esta señal de reloj es la responsable de que el programa y los contadores avancen. Como es necesario un procesamiento de señal, el dispositivo controlador trabaja a una frecuencia elevada, particularmente en este sistema, ya que es necesaria la ejecución de un elevado número de instrucciones por segundo. La frecuencia del oscilador es variable desde los 4MHz hasta 48MHz. Aumentar la frecuencia del reloj supone disminuir el tiempo que tardan en ejecutarse las instrucciones.

Dentro del microcontrolador está el circuito de reloj y la determinación de la frecuencia de trabajo se consigue a través de unos componentes externos. Dichos componentes son, o una red R-C, o un resonador cerámico, o bien un cristal de cuarzo junto a dos condensadores.

- **Recursos auxiliares.**

Junto a las características descritas anteriormente se incorporan otras prestaciones que refuerzan la potencia y flexibilidad del dispositivo. Entre los recursos más comunes que son útiles para el buen fin de este Proyecto, se encuentran los siguientes:

- **Interrupciones.** Las interrupciones permiten la ejecución de una parte del código del programa para desarrollar una función en concreto. La interrupción utilizada con mayor frecuencia para el desarrollo de este sistema ha sido la atención a los pulsadores. De esta forma la atención a este evento se realiza en un pequeño intervalo de tiempo, cuando sucede, ejecutándose la mayor parte del tiempo el programa principal.
- **Puertos de comunicación.** Dan al microcontrolador la posibilidad de comunicarse con dispositivos externos que utilicen el mismo protocolo. Para

este Proyecto, el microcontrolador PIC18F2550 dispone de un puerto de comunicaciones USB.

4.4.2 Características principales del microcontrolador PIC18F2550

El microcontrolador que se ha usado en el desarrollo de este Proyecto es un PIC18F2550 de *Microchip Technology* con arquitectura interna de 16 bits. Se ha elegido este modelo principalmente debido a que se dispone de todas las herramientas, tanto de compilación como de programación. Su arquitectura de 16 bits es suficiente para desarrollar las operaciones que requiere este Proyecto y dispone de un transmisor-receptor USB para la conexión con un PC.

En la Tabla 4.2 se resumen las principales características del microcontrolador PIC18F2550.

Descripción	Tamaño
Memoria de programa FLASH	32KByte
Memoria para datos RAM	2KByte
Líneas E/S	24
Timer/Contadores	4
USB	48MHz
Fuentes de interrupción externa	3
Comparador analógico	2

Tabla 4.2. Principales características del microcontrolador PIC18F2550

4.4.3 Asignación de pines en el microcontrolador PIC18F2550

En la Figura 4.5 se encuentran representados los 28 pines del microcontrolador PIC18F2550. La descripción de las funciones correspondientes a cada uno de estos 28 pines se muestra resumida en la Tabla 4.3.

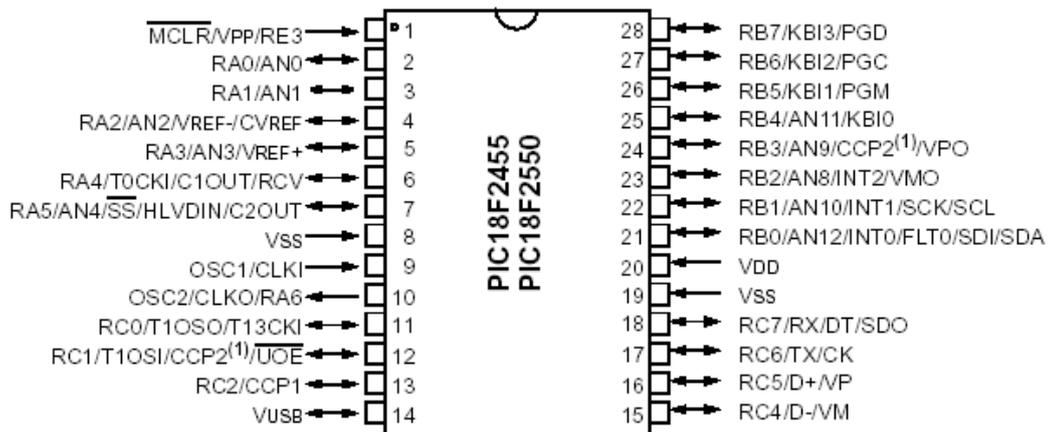


Figura 4.5. Pin-out PIC18F2550

Pin	Descripción
MCLR/VPP	RESET externo. Por este pin se aplica también la tensión / VPP usada en la grabación del programa.
RA0/AN0	E/S digital o entrada analógica.
RA1/AN1	E/S digital o entrada analógica.
RA2/AN2/VREF/CVREF	E/S digital o entrada analógica. Tensión baja de referencia convertor A/D o salida analógica del comparador Tensión de referencia.
RA3/AN3/ VREF	E/S digital o entrada analógica. Tensión alta de referencia convertor A/D.
RA4/TOCKI/C1OUT/RCV	E/S digital o entrada del reloj para TMR0.Salida 1 del comparador o USB externo RCV entrada.
RA5/AN4/SS/HLVDIN/C2OUT	E/S digital o entrada analógica. Selección del puerto serie síncrono. Detección de nivel alto o bajo señal de entrada. Salida 2 del comparador

VSS	Tierra
OSC1/CLKIN	Entrada al cristal cuarzo o reloj externo.
OSC2/CLKOUT/RA6	Salida del cristal externo. En modo R-C por este pin sale ¼ FOSC1. E/S propósito general.
RC0/T1OSO/T13CL1	E/S digital del Puerto C. Conexión del oscilador externo para el temporizador TMR1 o entrada de reloj para el TMR1/TMR3.
RC1/T1OSI/CCP2: E/S	Pin de Entrada/Salida. Entrada oscilador TMR1. Entrada modulo Captura2/Salida comparador 2/Salida PWM2. Externa USB
RC2/CCP1	E/S digital del Puerto C. Conexión del oscilador externo para TMR1 o salida del modulo 2 de captura/comparación.
VUSB	Regulador de tensión interna USB
RC4/V-/VM	Entrada digital. Línea diferencial de datos USB. Entrada VM USB.
RC5/D+/VP	Entrada digital. Línea diferencial de datos USB. Salida VP USB.
RC6/TX/CK	E/S digital. Transmisión serie asíncrona. Entrada de reloj para comunicación serie síncrona.
RC7/RX/DT	E/S digital. Recepción serie asíncrona. Línea de datos en la comunicación serie síncrona.
VDD	Entrada del positivo de la alimentación.
RB0/AN12/INT0/FLT0/SDI/SDA	E/S digital o entrada analógica. Interrupción externa 0. PWM entrada. Datos entrada SPI. I2C datos E/S.
RB1/AN10/INT1/SCK/SCL	E/S digital o entrada analógica. Interrupción externa 1. Entrada reloj serie síncrono/salida modo SPI. Entrada reloj serie síncrono/salida modo I2C.
RB2/AN8/INT2/VMO	E/S digital o entrada analógica. Interrupción externa 2. Salida VMO USB.
RB3/AN9/CCP2/VPO	E/S digital o entrada analógica. Entrada modulo Captura2/Salida comparador2/Salida PWM2. Salida VPO USB.
RB4/AN11/KBI0	E/S digital o entrada analógica. Interrupción de cambio de pin.
RB5/KBI1/PGM	E/S digital. Interrupción de cambio de pin. ICSP programador baja tensión.
RB6/KBI2/PGC	E/S digital. Interrupción de cambio de pin. ICSP reloj.
RB7/KBI3/PGM	E/S digital. Interrupción de cambio de pin. ICSP datos.

Tabla 4.3. Asignación de pines para el microcontrolador PIC18F2550 y funciones que realizan

Una vez explicado el funcionamiento de cada pin del PIC18F2550, en la Figura 4.6 se muestra su arquitectura interna, con el diagrama de bloques de los periféricos y las líneas de entrada y salida.

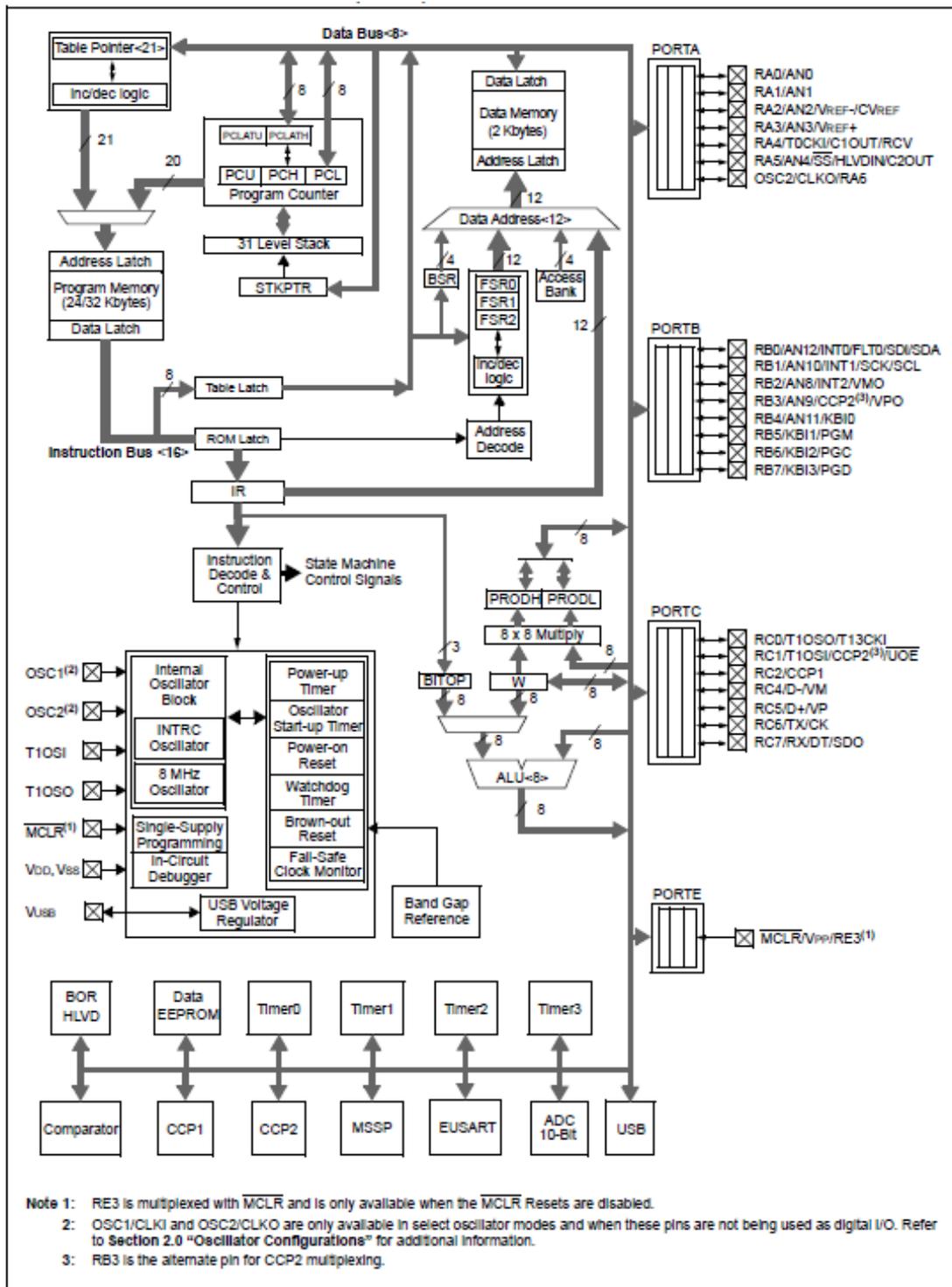


Figura 4.6. Arquitectura interna del microcontrolador PIC18F2550

4.4.4 Organización de la memoria

El PIC18F2550 tiene dos tipos de memoria: memoria de programa y memoria de datos. En la memoria de programa se almacenan todas las instrucciones del programa de control, que debe estar almacenado de forma permanente. En la memoria de datos se almacenan los datos que manejan los programas. Éstos van variando continuamente, lo que exige que sea de lectura y escritura.

- **Memoria de programa.**

Los microcontroladores de la familia PIC18 implementan un contador de programa (CP) de 21-bit, el cual es capaz de direccionar hasta 2Mbytes de memoria de programa. Si se intenta acceder a alguna ubicación entre el límite superior de la memoria implementada (8000h) y los 2Mbytes de memoria totales (20000h) devolverá '0's (instrucción NOP). El PIC18F2550 tiene 32kbytes de memoria Flash, que pueden almacenar hasta 16.384 instrucciones de una palabra. Los PIC18 tienen dos vectores de interrupción: el vector RESET que está en la dirección 0000h y los vectores de interrupción, que están en las direcciones 0008h y 0018h.

En la Figura 4.7 se muestra el mapa de memoria del PIC18F2550.

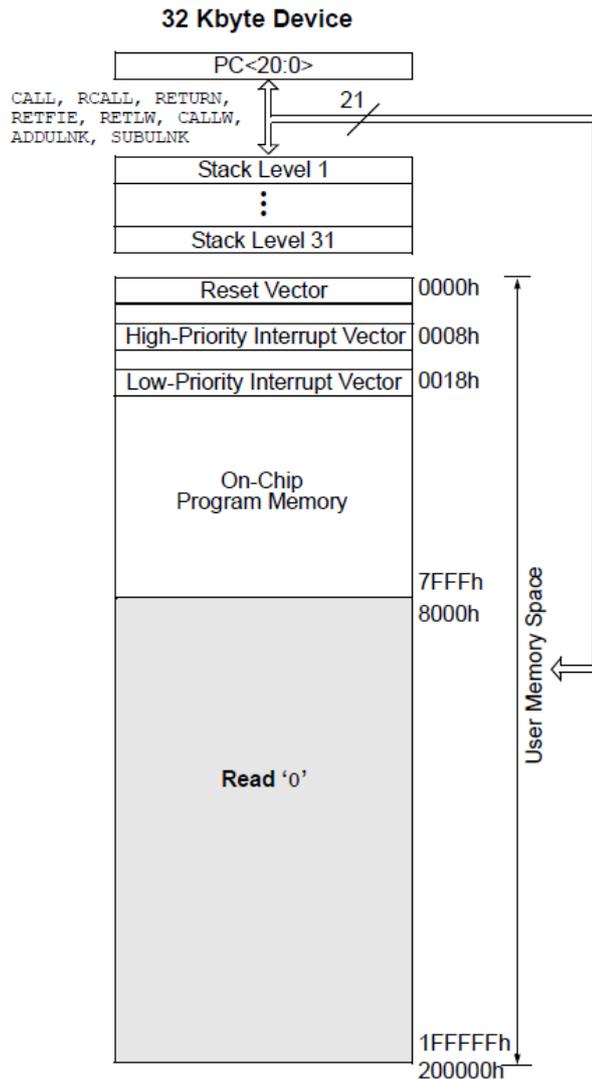


Figura 4.7. Mapa de memoria del microcontrolador PIC18F2550

- **Direccionamiento de la memoria de programa.**

El contador del programa (CP) de 21 bits, está compuesto por tres registros de 8 bits cada uno (ver Figura 4.8). El *byte bajo* del registro, conocido como PCL, puede ser leído y escrito. El *byte intermedio* (PC<15:8>) está alojado en el registro PCLH, y sobre él no se puede leer ni escribir, pero se puede acceder a él indirectamente a través del registro PCLATH. El *byte superior* (PC<20:16>) está alojado en el registro PCLU, y sobre él no se puede leer ni escribir, pero se puede acceder a él indirectamente a través del registro PCLATU.

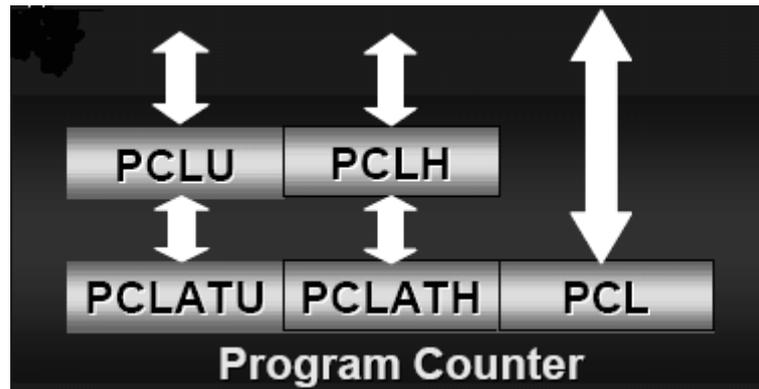


Figura 4.8. Contador de programa

- **Memoria de datos: banco de registros y EEPROM.**

La memoria de datos contiene Registros de Función Especial (SFRs) y Registros de Propósito General (GPRs). Los SFRs son usados para control y estado del controlador y para funciones de los periféricos, mientras que los GPRs son usados para almacenamiento de datos de las aplicaciones del usuario. Cualquier lectura de una localización no implementada debe ser leída como 0's.

Con el conjunto de instrucciones se pueden realizar operaciones en todos los bancos. Existen los siguientes modos de direccionamiento: Directo, Indirecto o Indexado.

La memoria de datos en los PIC18 se implementa como una RAM estática. Cada registro en la memoria de datos tiene 12 bits de direccionamiento, permitiendo hasta 4096 bytes de memoria de datos. El espacio de memoria es dividido en 16 bancos de registros que contienen cada uno 256 bytes. El PIC18F2550 tiene implementados 8 bancos completos para un total de 2048 bytes.

La Figura 4.9 muestra la organización de la memoria de datos para el PIC18F2550.

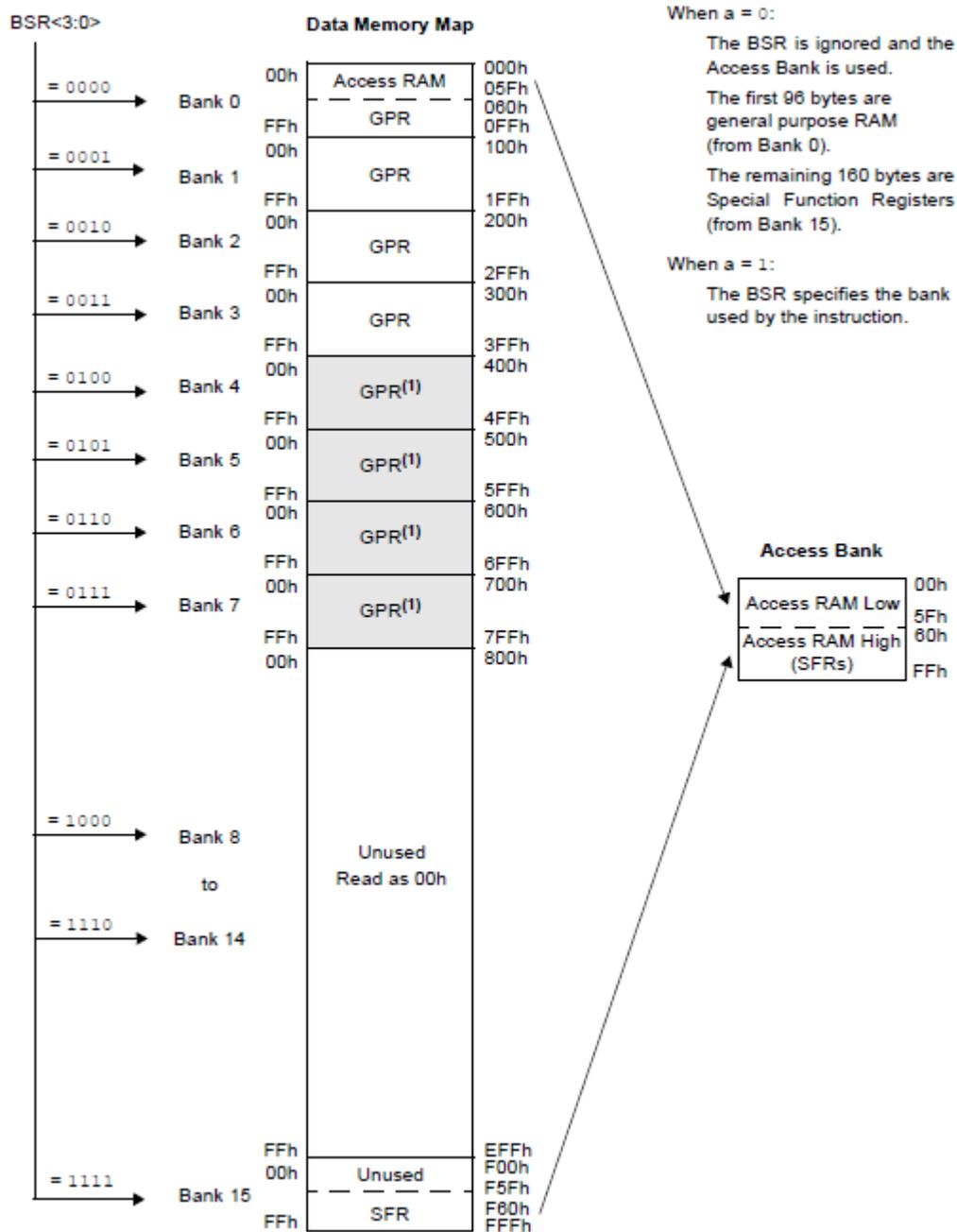


Figura 4.9. Banco de registros del PIC18F2550

La memoria de datos EEPROM de 256 bytes en el PIC18F2550 es una memoria no volátil que almacena a largo plazo cualquier dato que se desee retener cuando se apague la alimentación. Esta memoria es de 8 bits, no forma parte del espacio normal direccionable y sólo es accesible en lectura y escritura a través de dos registros.

4.4.5 Líneas de Entrada-Salida

Hay tres puertos disponibles, llamados A, B y C, siendo cada uno de ellos de 8 bits bidireccionales. Cada puerto tiene 3 registros para su funcionamiento:

- Registro TRIS (datos del registro de dirección).
- Registro PORT (lee los niveles en los terminales del dispositivo).
- Registro LAT (retención de salida).

Mediante los registros TRIS se configuran los puertos como entrada o salida. Si se está conectado con otro dispositivo como por ejemplo un sensor, al configurarlo como entrada, cada vez que se reciba una señal de éste, avisará al microcontrolador para que ejecute la instrucción correspondiente. Si está configurado como salida, será el microcontrolador el encargado de comunicarle al dispositivo las tareas a realizar.

Los pines de cualquier puerto, A, B ó C, pueden funcionar como pines de Entrada/Salida pero además suelen tener otras funciones asociadas, como por ejemplo de reloj, de conversión Analógico/Digital recepción de datos del puerto serie, etc. Con el registro PORT se selecciona cómo se utilizarán en cada caso los puertos.

Con los registros LAT se almacenan y leen los datos de los puertos.

4.4.6 Reloj principal

Es un elemento imprescindible, ya que es el encargado de generar la base de tiempos para la ejecución de cada instrucción. Los PIC18 tienen un oscilador flexible, es decir, el microcontrolador puede estar trabajando a 12MHz y al mismo tiempo a los 48MHz necesarios para el USB. Hay osciladores primarios, secundarios e internos, existiendo en total 12 osciladores.

La Figura 4.10 muestra los diferentes modos del oscilador del microcontrolador PIC18F2550.

1. XT Crystal/Resonator
2. XTPLL Crystal/Resonator with PLL enabled
3. HS High-Speed Crystal/Resonator
4. HSPLL High-Speed Crystal/Resonator with PLL enabled
5. EC External Clock with Fosc/4 output
6. ECIO External Clock with I/O on RA6
7. ECPLL External Clock with PLL enabled and Fosc/4 output on RA6
8. ECPIO External Clock with PLL enabled, I/O on RA6
9. INTHS Internal Oscillator used as microcontroller clock source, HS Oscillator used as USB clock source
10. INTXT Internal Oscillator used as microcontroller clock source, XT Oscillator used as USB clock source
11. INTIO Internal Oscillator used as microcontroller clock source, EC Oscillator used as USB clock source, digital I/O on RA6
12. INTCKO Internal Oscillator used as microcontroller clock source, EC Oscillator used as USB clock source, Fosc/4 output on RA6

Figura 4.10. Modos del oscilador para el microcontrolador PIC18F2550

4.4.7 Recursos auxiliares: Interrupciones y Puertos de comunicación

En esta Apartado se describe una parte muy importante de los microcontroladores, las interrupciones para controlar sucesos esperados y los puertos de comunicación para intercambiar información con otros dispositivos.

- **Interrupciones.**

Una interrupción consiste en la ejecución de un programa secundario dejando parado el programa principal. Una vez acabada la interrupción, la ejecución volverá al programa principal en el punto en que se abandonó. Las causas que originan una interrupción pueden ser tanto externas como internas. Las externas suelen estar asociadas a sensores o actuadores, mientras que las internas se producen, por

ejemplo, al desbordarse un temporizador o un contador. Las interrupciones evitan que el contador de programa (CP), examine continuamente el nivel lógico de un pin o el valor de un contador.

Las prioridades para atender interrupciones son asignadas por el usuario. Los vectores de alta prioridad están en la dirección 000008h y los vectores de baja prioridad están en 000018h. Hay 10 registros para la operación de control de las interrupciones, que se enumeran a continuación:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

Cada fuente de interrupción tiene 3 bits de control y operación. La función de estos bits es:

- El *bit de bandera* indica que se ha producido un evento de interrupción.
- El *bit de habilitación* permite saltar, durante la ejecución de un programa, a la dirección del vector de interrupción cuando el bit de bandera está en *set*.
- El *bit de prioridad* informa si es de alta o baja prioridad.

En la Figura 4.11 se muestra el esquema lógico que controla la generación de la interrupción, cuando aparece un nivel alto en su línea INT de salida.

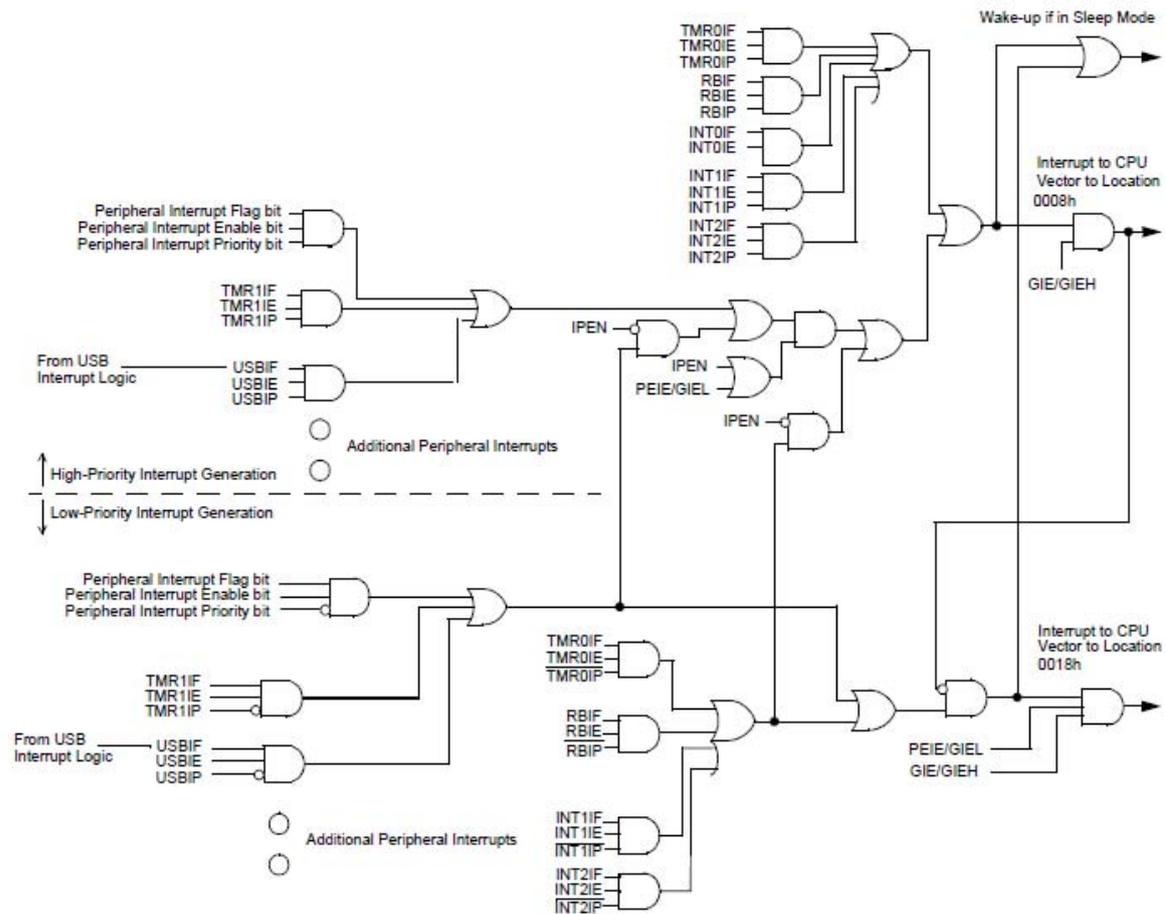


Figura 4.11. Lógica de control para la generación de la interrupción

Cada causa de interrupción está controlada mediante dos líneas o señales: una de ellas actúa como una bandera de señalización que indica si se ha producido o no el evento, mientras que la otra es la que da permiso, o prohíbe, la interrupción en sí.

▪ **Puertos de comunicación.**

Los pines del microcontrolador se agrupan formando puertos. A través de estos puertos se puede establecer comunicación entre el microcontrolador y otros dispositivos para la recepción y/o envío de datos. Las características más destacadas de los periféricos de los PIC18 son:

- Alta corriente de alimentación (25 mA).
- 3 interrupciones externas.
- 4 módulos de Timer (Timer0 a Timer3).

- 2 Módulos de Captura/Comparación/PWM (CCP) :
 - ✓ *Capture* es de 16-bit, máx. resolución 5.2 ns (TCY/16).
 - ✓ *Compare* es de 16-bit, máx. resolución 83.3 ns (TCY).
 - ✓ Salida PWM con resolución de 1 a 10 bits.
- Módulo de Capture/Compare/PWM mejorado (ECCP).
 - ✓ Múltiples modos de salida.
 - ✓ Polaridad seleccionable.
 - ✓ Programación de tiempo muerto.
 - ✓ Auto-apagado y Auto-encendido.
- Módulo USART mejorado:
 - ✓ Soporta bus LIN.
- Módulo *Master Synchronous Serial Port* (MSSP) que soporta 3-cables SPI™ (4 modos) y modo maestro – esclavo de I2C™.
- 10-bit, hasta 13-canales del módulo convertidor Análogo-a-Digital (A/D).
- Doble comparador analógico con entrada multiplexada.

4.4.8 RESET

El *reset* de los microcontroladores puede ser originado por alguna de las siguientes causas:

- 1) Conexión de la alimentación (POR: Power-On-Reset).
- 2) Activación del pin MCLR (Master Clear Reset) durante una operación normal.
- 3) Activación de MCLR en el estado de Reposo o SLEEP.
- 4) Desbordamiento del *perro guardián*.
- 5) Programable Brown-out-reset(BOR)
- 6) Instrucción RESET.
- 7) Reset completo de pila.
- 8) Reset de pila por desbordamiento.

Como se aprecia en el esquema de la Figura 4.12, cualquiera de estas posibilidades descritas anteriormente produce un nivel bajo en la entrada S. Transcurridos

18 milisegundos, origina un flanco ascendente en la salida Q que supone la generación del reset interno.

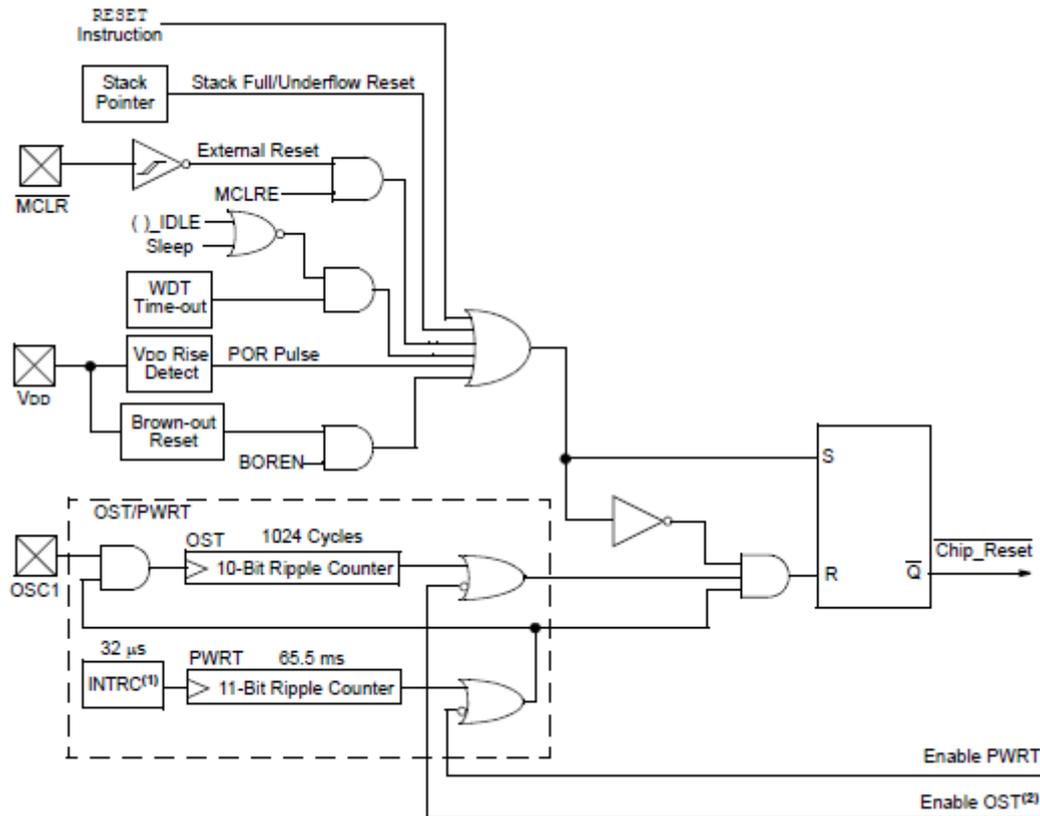


Figura 4.12. Diagrama de bloques del circuito de RESET

4.4.9 Interfaz de comunicación Universal Serial Bus (USB)

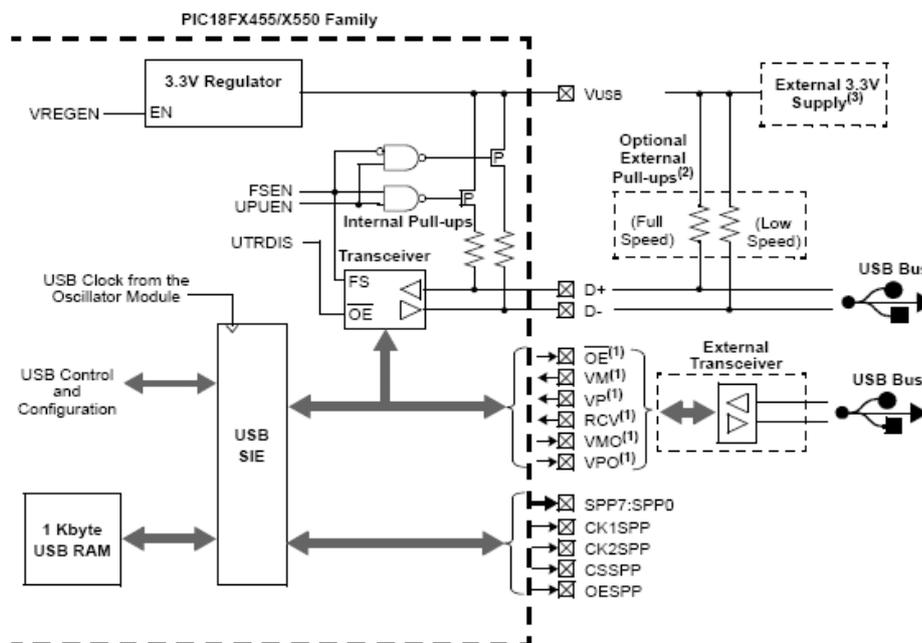
Las características fundamentales del módulo USB (*Universal Serial Bus*) de los PIC18 se enumeran a continuación:

- Cumple con los parámetros USB V2.0.
- Baja velocidad (1.5 Mb/s) y alta velocidad (12 Mb/s).
- Soporta transferencias de lotes de control, interrupción y sincronización.
- Soporta hasta 32 *endpoints* (16 bidireccionales).
- 1-kbyte dual de acceso a RAM para USB.

- Circuito interno USB de transmisión-recepción con regulador de voltaje.
- Interfaz para un circuito externo USB de transmisión-recepción.

El PIC18F2550 incorpora un completo módulo de comunicación compatible con el USB, *Serial Interface Engine* (SIE), tanto para alta velocidad como para baja. Esto permite una rápida comunicación entre cualquier host USB y el microcontrolador PIC18F2550. El SIE puede ser una interfaz directa al puerto USB utilizando el transceptor interno, o a través de un transceptor externo. Un regulador interno de 3.3V también está disponible para alimentar al transceptor interno en aplicaciones que necesiten 5V. Se han incluido algunas características especiales de *hardware* para conseguir un mejor desarrollo. La memoria de doble puerto en el espacio de la memoria de datos (USB RAM) suministra el acceso directo a memoria entre el microcontrolador y el SIE. También se han incluido descriptores de buffer para permitir al usuario programar los *endpoint* usando el espacio de la USB RAM. Un SPP (*Streaming Parallel Port*) provee el soporte para una transferencia ininterrumpida de un gran volumen de datos, tales como datos asíncronos a los buffer externos de memoria.

La Figura 4.13 presenta una perspectiva general del periférico USB.



- Note 1: This signal is only available if the internal transceiver is disabled (UTRDIS = 1).
 Note 2: The internal pull-up resistors should be disabled (UPUEN = 0) if external pull-up resistors are used.
 Note 3: Do not enable the internal regulator when using an external 3.3V supply.

Figura 4.13. Perspectiva general de un USB

Los PIC18F2550 incorporan un módulo de comunicación USB (*Universal Serial Bus*), que cumple con la especificación USB 2.0. Este módulo tolera los tipos de transferencia de datos de baja y alta velocidad.

La operación asíncrona del reloj dual, permite al modulo USB de los PIC18 trabajar a alta frecuencia, mientras que el resto de los microcontroladores sólo tienen un reloj que proviene de un oscilador interno de baja frecuencia. Las operaciones con el módulo USB se configuran y gestionan a través de 3 registros. En total hay 22 registros a través de los cuales se configura la transacción de datos por el USB:

- Registro de control USB (UCON).
- Registro de configuración USB (UCFG).
- Registro estado de transferencia USB (USTAT).
- Registro dirección del dispositivo USB (UADDR).
- Registros de número de estructura (UFRMH:UFRML).
- Registros (del 0 al 15) de habilitación de puntos finales (UEPn).

4.5 SELECCIÓN DE COMPONENTES Y CONEXIONADO

Una vez dadas las principales características de este microcontrolador, se entrará en el detalle de las funcionalidades del mismo que se han utilizado. Las entradas y salidas del PIC18F2550 se conectan siguiendo las indicaciones de la hoja de características del dispositivo y las necesidades de diseño de la aplicación.

La Figura 4.14 muestra la asignación de pines de Entradas/Salidas del PIC18F2550 y en ella se puede ver que cada uno de los pines utilizados lleva una etiqueta que identificará la conexión de ese pin con el resto de elementos que forman el diseño de la placa.

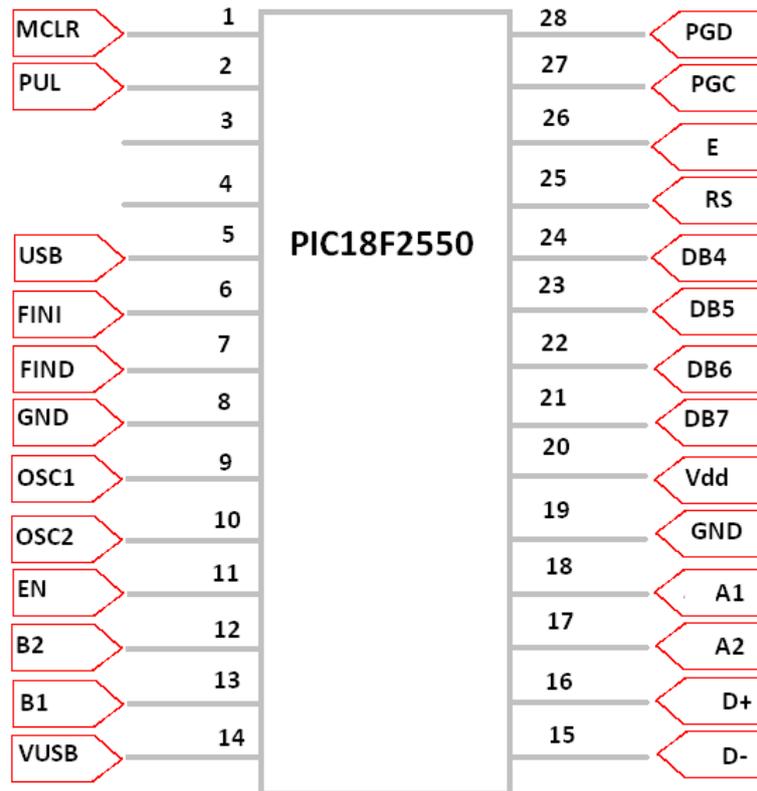


Figura 4.14. Asignación de pines de Entradas/Salidas del PIC18F2550

4.5.1 Alimentación

Se coloca un borne que constituirá el punto de conexión de la placa con la fuente de alimentación (ver Figura 4.15). Físicamente se trata de una simple regleta que por un lado se puede soldar a la placa de circuito impreso y por el otro, se conecta a los cables conductores de la tensión de alimentación.

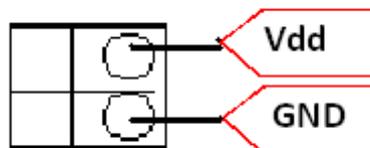


Figura 4.15. Alimentación

Todos los microcontroladores ofertados por *Microchip Technology* poseen una alimentación de 5V. El resto de componentes para el desarrollo del sistema se ha elegido teniendo en cuenta este dato, para evitar el uso de alimentaciones diversas sobre la misma placa. Finalmente se eligió un regulador de voltaje LM7805 cuyo rango de tensión de alimentación de entrada es de 8V y suministra 5V, quedando la placa alimentada con una única línea. Las salidas de los bornes de la alimentación se conectan a la entrada 20 del PIC para V_{DD} y a los pines 8 y 19 para V_{SS} .

4.5.2 Condensadores de desacoplo

El condensador de desacoplo se utiliza para dar una mayor estabilidad a la hora de alimentar el microcontrolador, ya que evita las posibles oscilaciones en la alimentación del dispositivo. Se coloca en paralelo, lo más próximo posible al integrado, entre alimentación y tierra.

En la Figura 4.16 se muestra el condensador de desacoplo.

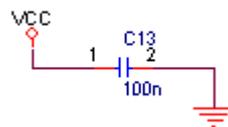


Figura 4.16. Condensadores de desacoplo

4.5.3 Regulador de tensión

Los reguladores de tensión son dispositivos que protegen a los aparatos eléctricos de posibles picos en la tensión de entrada. Permiten la conversión de la tensión de entrada en otra tensión de menor valor a su salida y suficientemente estable.

El regulador elegido es el modelo LM7805 de la marca Fairchild. Es un regulador de bajo consumo, en el que la tensión entre los terminales V_{out} y GND es de valor fijo, no variable, y depende del modelo de regulador que se utilice. En este caso, un regulador de la serie 7805 es de 5V. La tensión de entrada V_{in} deberá ser siempre unos 2 ó 3 V superior a la de V_{out} para asegurar el correcto funcionamiento.

En la Figura 4.17 se muestra el conexionado del LM7805.

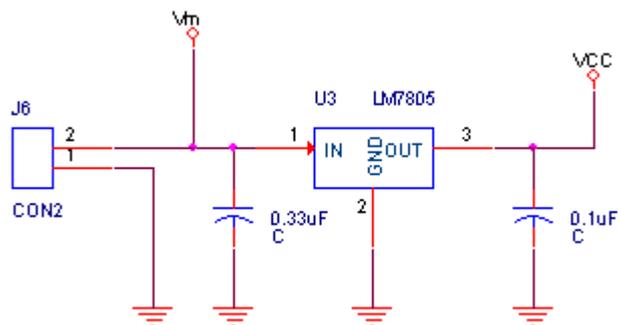


Figura 4.17. Conexión del regulador de tensión

V_m es el voltaje para que funcione el motor, que en este caso vale 8V. Con esta tensión se alimentan el motor, el driver L293 y las líneas de salida del driver hasta que lleguen al motor. V_{cc} es la tensión con la que se alimenta el resto del circuito (microcontrolador, pulsadores, el mismo driver L293 y la pantalla LCD) y vale 5V. El driver L293 utiliza dos fuentes diferentes, la de 5V para alimentar el propio driver y la fuente de 8V para controlar las bobinas del motor. El fabricante recomienda la colocación de dos condensadores en la entrada y salida del LM7805, recomendando para ellos unos valores que mejoran la estabilidad.

4.5.4 Driver L293

Como ya se explicó en el Capítulo 3, el driver L293 se encarga de energizar las bobinas de motor. Es un puente en H que convierte las señales analógicas que recibe, en señales capaces de controlar el movimiento de un motor.

En la Figura 4.18 se puede ver el conexionado.

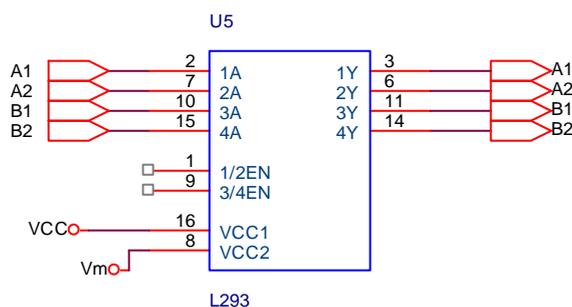


Figura 4.18. Conexión del driver L293

Se aprecia que el *driver* del motor necesita estar alimentado a dos tensiones distintas, una de ellas la tensión de alimentación de toda la placa (Vcc) y la otra, la tensión del motor (Vm).

4.5.5 Circuito de RESET

Cuando se alimenta el microcontrolador se produce un RESET y, por tanto, todos los registros adquieren la configuración que presentaban originalmente. Se puede resetear un microcontrolador si se dan las condiciones explicadas en el Apartado 4.4.8, o a través de un circuito de RESET. Cuando se origina un RESET, se producen lo siguientes eventos:

- El programa se reinicia volviendo a comenzar la ejecución del mismo.
- Los registros modificados vuelven a su estado normal.

Los microcontroladores disponen de un pin \overline{MCLR} que es el pin de RESET y a través de un circuito de RESET se produce la reinicialización del microcontrolador. El RESET puede ser activado de manera externa a través del circuito de RESET o internamente, por el propio PIC18F2550 si se produce alguno de los siguientes sucesos:

- Cada vez que, estando apagado, se alimente a +Vcc.
- Con caídas de tensión de alimentación del PIC18F2550 por debajo de 2,2V.
- Con desbordamientos de WDT (contador que funciona con los impulsos de su propio oscilador).

En la Figura 4.19 se muestra el circuito de RESET externo.

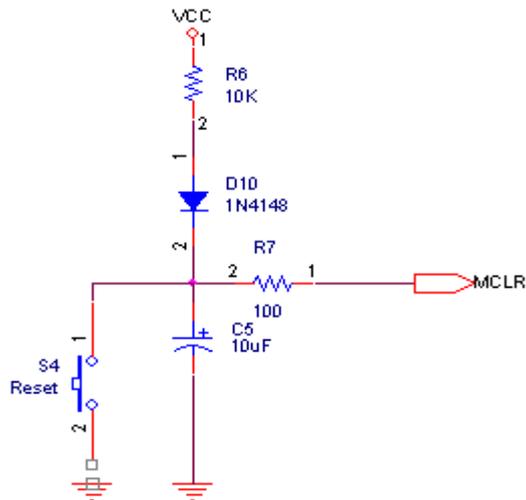


Figura 4.19. Circuito de RESET con pulsador

Mientras no se apriete el pulsador S4, toda la corriente suministrada por la fuente Vcc pasará a través de la resistencia R6 y del diodo D10. Como hay 3 posibles salidas en paralelo (un condensador C5, un interruptor abierto S4, o una resistencia R7), la corriente estará obligada a atravesar la resistencia R7 y llegar al pin \overline{MCLR} . Por el condensador no pasa corriente, ya que en continua se comporta como circuito abierto, y mientras no se pulse el interruptor S4, estará también en circuito abierto. Por el contrario, si se pulsa el interruptor S4 toda la corriente a la salida del diodo D10 pasará directamente a través del interruptor S4 a masa. En este caso, el nivel de tensión que alcanza el pin \overline{MCLR} es suficientemente bajo como para que el microcontrolador se resetee. Algunas reglas básicas a tener en cuenta para realizar este circuito son:

- Es recomendable que $R6 < 40 \text{ k}\Omega$.
- Se recomienda que $R7 < 1 \text{ k}\Omega$.

4.5.6 Cristal del oscilador

Es el elemento que se encarga de generar la base de tiempo para que el microcontrolador ejecute el programa, es decir, el cristal es el encargado de proporcionar la frecuencia de reloj al PIC18F2550. Este tema ya se trató en el Apartado 4.4.6 de este mismo Capítulo. El USB sólo puede tomar la señal del reloj del oscilador primario, mientras que el resto de módulos del microcontrolador pueden utilizar cualquier otra fuente.

En este diseño se utiliza un dispositivo USB de alta velocidad; por tanto, son necesarios 48MHz que se tomarán del oscilador primario. Esos 48MHz se conseguirán a partir de un cristal de 20MHz, de la forma que se explicará en detalle en el Apartado 4.5.8. Según se indica en la documentación del microcontrolador elegido, el PIC18F2550, es necesario instalar dos condensadores encargados de estabilizar la frecuencia de salida del cristal. Los condensadores irán conectados entre cada patilla del cristal y tierra (ver Figura 4.20).

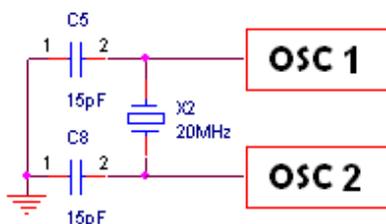


Figura 4.20. Esquema de conexión del cristal oscilador

Las salidas del cristal de cuarzo serán entradas del PIC18F2550 sobre los pines 9 (OSC1) y 10 (OSC2), preparados para recibir las frecuencias del cristal y tratar la información.

Para conocer los valores de los condensadores se debe acudir a la hoja de características del PIC18F2550, comprobando la frecuencia del cristal. Para la implementación particular en este diseño los condensadores valen 15pF, como se puede apreciar en la Tabla 4.4.

TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	C2
XT	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

Tabla 4.4. Selección de la capacidad del condensador

Ambos circuitos, condensadores de desacoplo y cristal, deben estar próximos al PIC18F2550. Ésta es una de las ideas a tener en cuenta a la hora de empezar con el Layout.

Hasta ahora se han comentado los elementos auxiliares que siempre necesita todo microcontrolador, cristal, alimentación, circuito de reset y condensadores de desacoplo. A continuación, se detalla el diseño de los periféricos de entrada y salida necesarios para la aplicación que se desea implementar.

4.5.7 Pulsadores

Se ha diseñado, en una placa adicional, un circuito con 5 pulsadores para el control de la LCD y el movimiento del microposicionador. Para implementar un cursor capaz de controlar la posición de la vagoneta sobre el carril se necesitan 5 pulsadores, cuatro de ellos para el desplazamiento en los dos sentidos y un quinto como tecla INTRO.

La funcionalidad de los pulsadores de *izquierda* y *derecha* se ha diseñado para conseguir desplazamientos cortos de la vagoneta, mientras que para los pulsadores de *arriba* y *abajo* se ha pensado en desplazamientos largos. Se han conectado los cinco pulsadores en un circuito divisor de tensión y una única salida PUL conectada a un único pin del microcontrolador (ver Figura 4.21). De esta forma, dependiendo de qué pulsador se presione, habrá una tensión diferente en la salida PUL y, mediante el conversor analógico-digital, el microcontrolador sabrá qué pulsador se está accionando de los cinco posibles. Así, con un solo pin del microcontrolador se controlan cinco interrupciones; para este diseño se ha elegido el pin2 del PIC18F2550.

La conversión Analógica a Digital que realiza el PIC18F2550 consiste en transformar un valor de tensión, conectado a un pin habilitado para ello, en un valor numérico proporcional a dicha tensión, que se puede guardar en una variable y ser utilizado por el programa del PIC18F2550.

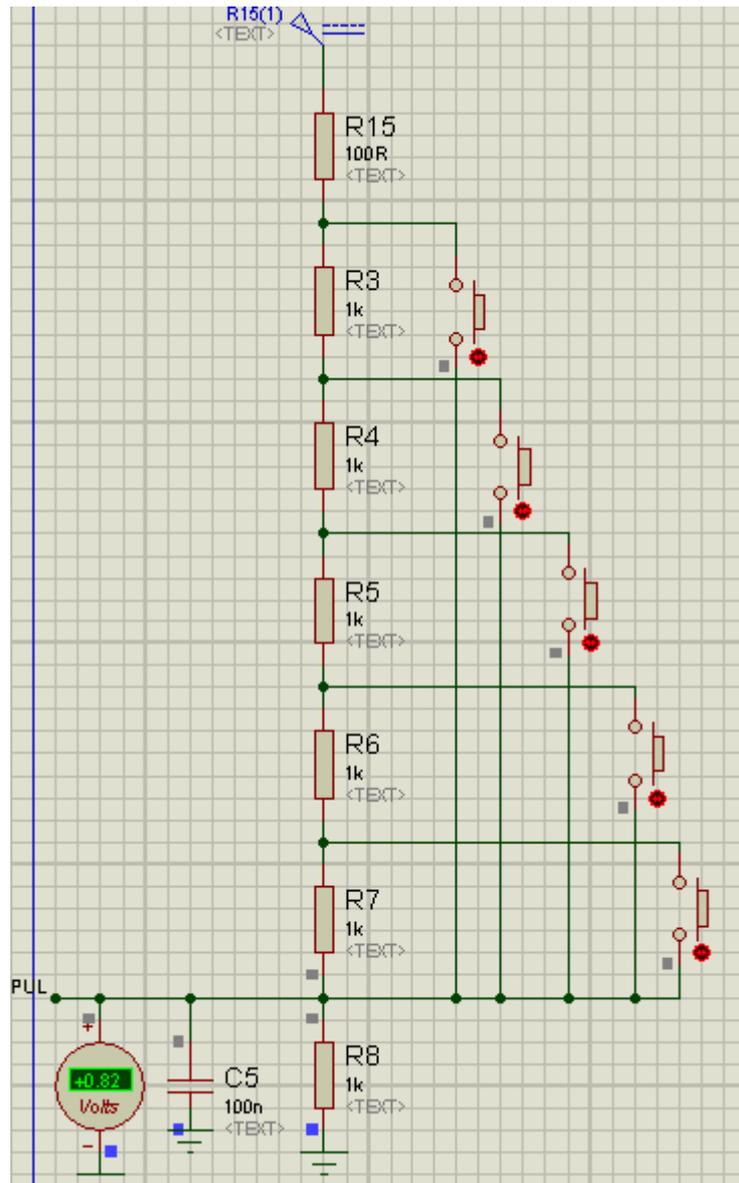


Figura 4.21. Pulsadores para el control LCD

Es decir, si los pines del PIC18F2550 aceptan tensiones entre 0V y 5V, respecto de GND, y se realiza una conversión Analógico - Digital de dicha tensión utilizando 8 bits, el número de niveles discretos que el PIC18F2550 puede “leer” en el pin es de 256 ($2^8 = 256$). Si los niveles se numeran del 0 al 255, el nivel 0 se corresponde con 0V y el nivel 255 con 5V, con lo que el paso de cuantificación vale 0,02V.

Como se aprecia en la Figura 4.21, según se vayan seleccionando los diferentes pulsadores, habrá diferentes tensiones a la salida del divisor de tensión resistivo. Cuando no se presiona ningún pulsador, la salida del divisor de tensión será de 0,82V.

En el caso de que se apriete el pulsador más cercano a la fuente, toda la corriente circulará directamente hacia la salida PUL, por tanto, la tensión será aproximadamente de 5V. Si se aprieta el segundo pulsador, la corriente procedente de la fuente pasará por la primera y segunda resistencia y saldrá por PUL. Para calcular el nuevo valor de tensión se aplica el divisor de tensión entre las dos primeras resistencias y la última de todas, obteniéndose 2,38V, como se aprecia en la Figura 4.22.

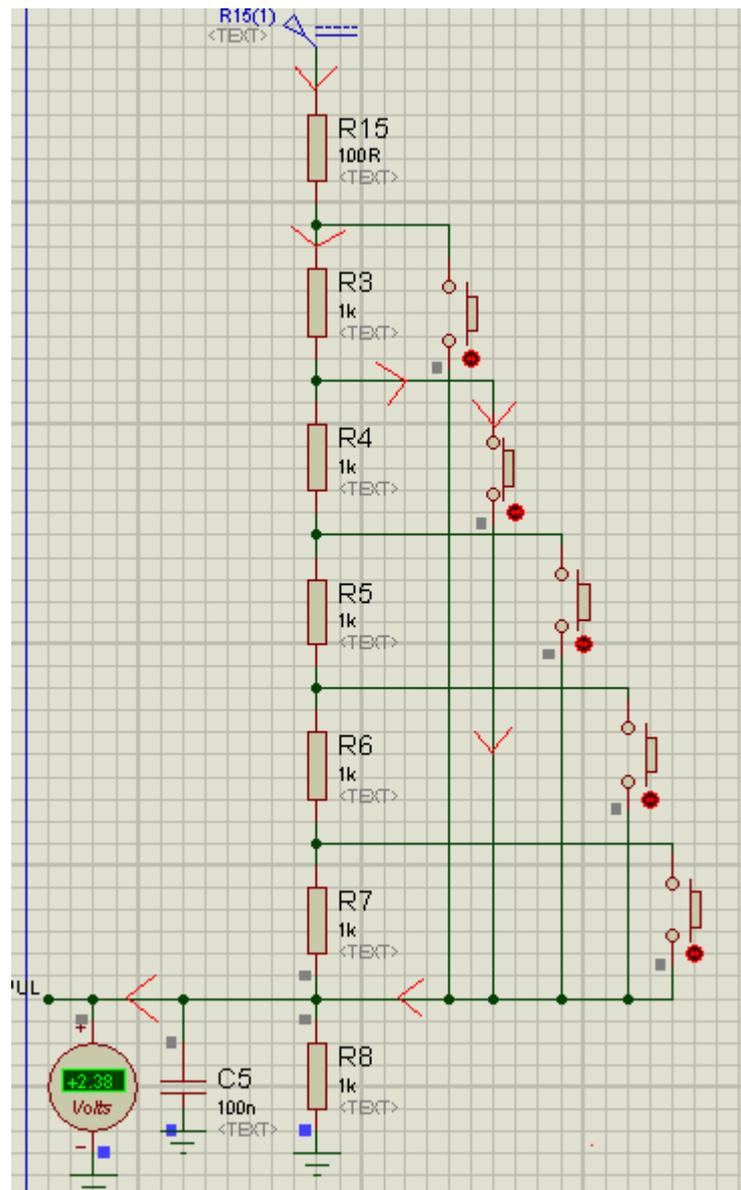


Figura 4.22. Pulsador 2 accionado

De la misma manera se deducen todos los valores de tensión en PUL. Resumiendo, los valores de la tensión para los diferentes pulsadores se indican en la Tabla 4.5.

Pulsador	Tensión (V)	Valor ADC
1	4,55	232
2	2,38	121
3	1,61	82
4	1,22	62
5	0,98	49

Tabla 4.5. Relación Pulsadores/Tensión/ValorADC.

Además de los pulsadores se han utilizado unos detectores de final de carrera. Funcionan igual que un pulsador: mientras no se accionan, el programa sigue funcionando y cuando se aprietan avisan al microcontrolador de que la vagoneta ha llegado al final del carril. El aviso evitará que el motor siga en marcha al detectar el final del carril y se queme. Se han instalado dos detectores, uno en cada extremo del carril, para comunicar con el PIC18F2550 ambos eventos. En el diseño, los interruptores están conectados para ser activos a nivel alto. Se tendrá esto presente a la hora de programarlos en el diseño de la aplicación.

Los finales de carrera están conectados a las entradas digitales que presenta el PIC18F2550 en los pines 6 y 7, correspondientes con Entradas/Salidas digitales del Puerto A. Como se aprecia en la Figura 4.23, mientras no se active el final de carrera representado en la Figura como un pulsador (S2), la tensión Vcc pasa directamente a la salida FIND. Una vez se accione el final de carrera, toda la tensión se derivará a tierra.

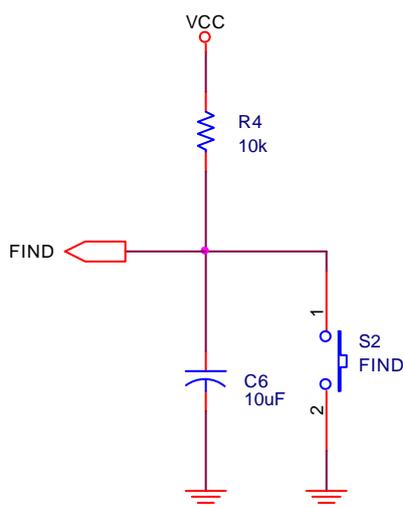


Figura 4.23. Pulsadores del final de carrera

4.5.8 Conexión USB

El PIC18F2550 permite la comunicación con un ordenador a través de la conexión USB (Universal Serial Bus). Para ello debe haber una tensión de 3,3V en el pin14 del PIC18F2550. Se conectará este pin a tierra a través de un condensador de desacoplo para dejar fijada dicha tensión y que no presente problemas en las entradas y salidas cercanas. Según la especificación del fabricante no debe ser mayor de 10µF; por lo que se ha elegido uno de 4,7µF, que es un valor comercial (verFigura 4.24).

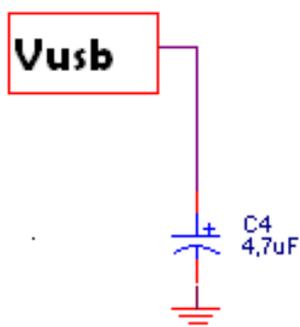


Figura 4.24. Condensador para fijar tensión USB

El PIC18F2550 soporta una comunicación con un puerto USB 2.0, a velocidades por tanto de 12 Mbit/s.

En la Figura 4.25 se puede apreciar el esquema de conexión del conector USB a un pin del microcontrolador denominado USB.

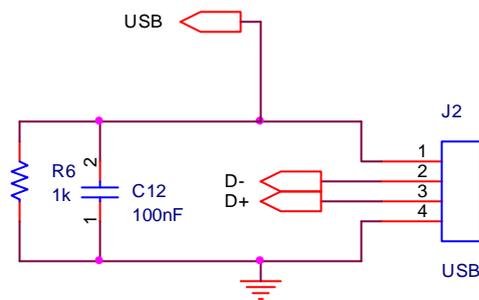


Figura 4.25. Conexión del USB

La línea de alimentación del pin1 del USB debería ser +Vcc pero se ha conectado al pin5 del microcontrolador para que cada vez que se conecte el USB se produzca una interrupción y el microcontrolador la atienda. Tiene una tensión de 5V y la corriente está limitada entre 3 y 5 amperios por razones de seguridad. Los terminales D+ y D- se utilizan para el envío y recepción de datos. Con el condensador puesto entre los terminales 1 y 4 se consigue filtrar todas las tensiones altas procedentes del ordenador. La resistencia en paralelo con el condensador se ha añadido para que el condensador se descargue más rápidamente una vez apagada la alimentación. Para esta placa se ha escogido un conector hembra USB de tipo B.

A continuación se detalla la configuración del microcontrolador para que funcione a 48MHz (necesarios para el USB) a partir de 20MHz. El detalle se ilustra en la Figura 4.26.

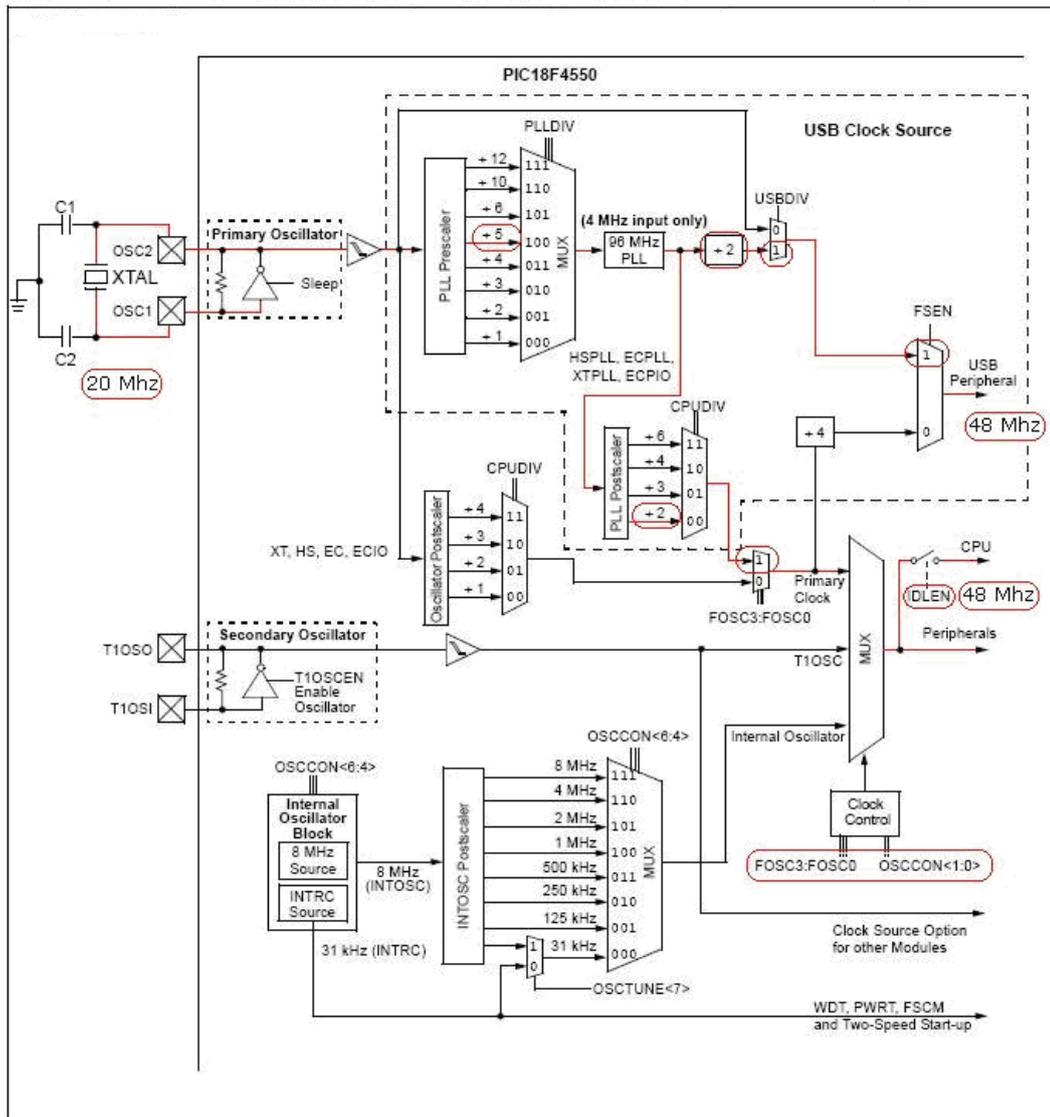


Figura 4.26. USB a 48MHz

Como se comentó anteriormente, en OSC1 y OSC2 se conecta el cristal, pudiéndose conectar cualquiera de los siguientes cristales: 4MHz, 8MHz, 12MHz, 16MHz, 20MHz, 24MHz, 40MHz ó 48MHz, que son los cristales válidos compatibles para este PIC18F2550.

Existen varias formas de conseguir 48MHz necesarios para el USB 2.0 a partir del cristal de 20MHz. El esquema se muestra en la Figura 4.26. A partir del cristal se accede al Primary Oscillator y en ese punto hay 3 rutas posibles.

- La línea superior acaba directamente en el *switch* USBDIV que (seleccionando su “entrada 0”) comunica directamente la frecuencia del oscilador de entrada con la salida. Este caso es el resultado de conectar directamente un cristal de 48MHz en el oscilador inicial.
- La segunda ruta obedece a la conexión en el oscilador de partida de un cristal compatible, pero de frecuencia diferente a 48MHz. La frecuencia de oscilación, al llegar al PLL Prescaler, se divide entre 1, 2, 3, 4, 5, 6, 10 ó 12, de manera que a la salida del multiplexor PLLDIV siempre haya 4MHz. Estos se convertirán en 96MHz a la salida de PLL. El *switch* USBDIV permite el paso de la frecuencia dividida entre 2 (48MHz) seleccionando su “entrada 1”. En particular, en este caso en que se parte de un cristal de 20MHz, la frecuencia se divide entre 5, y da lugar a 4MHz a la salida del multiplexor.
Una ruta alternativa a la salida del 96 MHz PLL es dirigirse hacia el PLL Postscaler. Allí, la frecuencia de oscilación de 96MHz se puede dividir entre 2, 3, 4 ó 6, obteniéndose frecuencias de reloj de 16MHz, 24MHz, 32MHz ó 48MHz.
- Por la tercera ruta, la frecuencia del cristal llega al Oscilator Postcaler. En este nuevo divisor los cocientes pueden ser 1, 2, 3 ó 4. Una vez dividida la frecuencia, la señal accede hasta el *switch* FOSC3:FOSC0. Finalmente, las frecuencias resultantes se usarán como reloj para el funcionamiento del programa.

4.5.9 Pantalla LCD

La pantalla de cristal líquido ó LCD (*Liquid Crystal Display*) es un dispositivo microcontrolado de visualización gráfica para la presentación de caracteres o símbolos. La LCD utilizada para la implementación del dispositivo dispone de 2 filas de 16 caracteres cada una.

En la Figura 4.27 se muestra una fotografía de la pantalla LCD.

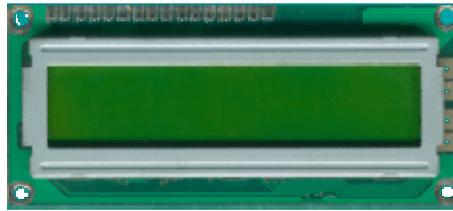


Figura 4.27. Pantalla LCD

Sus características principales se enumeran a continuación:

- Presentar caracteres ASCII y caracteres de alfabeto tradicional.
- Permitir desplazamiento de los caracteres hacia la izquierda o la derecha.
- Direccional de manera simple la posición de visualización de los caracteres.
- Recibir y mostrar caracteres personalizados.
- Movimiento del cursor y cambio de su aspecto.

Se ha escogido esta pantalla debido a que estaba disponible en el Laboratorio por haberse utilizado en trabajos previos de investigación, con lo cual se ahorran costes y, además su manejo es sencillo.

En la Figura 4.28 se muestra el esquemático con el patillaje de la pantalla LCD.

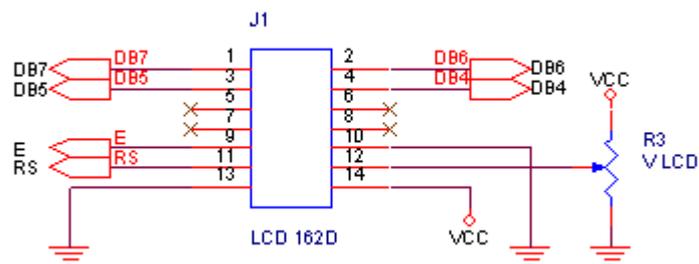


Figura 4.28. Esquemático de la LCD con la numeración de pines

Se observa que no es necesario tener conectados los 14 pines, ya que con las señales DB7 a DB4 se puede hacer el control completo de la LCD.

En la Tabla 4.6 se describen todos los pines de la LCD.

Pin NO.	Symbol	Level	Description
1	DB7	H/L	Data bit 7
2	DB6	H/L	Data bit 6
3	DB5	H/L	Data bit 5
4	DB4	H/L	Data bit 4
5	DB3	H/L	Data bit 3
6	DB2	H/L	Data bit 2
7	DB1	H/L	Data bit 1
8	DB0	H/L	Data bit 0
9	E	H, H → L	Chip enable signal
10	R/W	H/L	H : Read mode, L : Write mode
11	RS	H/L	H : Data signal, L : Instruction signal
12	VO	---	Input voltage for LCD
13	VSS	0V	Ground
14	VDD	5.0V	Supply voltage for logic
15	LED+	---	Backlight anode
16	LED-	---	Backlight cathode

Tabla 4.6. Pines de la LCD

4.5.10 Comunicación con el programador

Finalmente, a través del ICSP se consigue la programación del microcontrolador. Primeramente se compilan los programas y, con el puerto de comunicaciones ICSP, se traslada el código al PIC18F2550.

En la Figura 4.29 se muestran las conexiones entre el puerto y el microcontrolador, especificándose únicamente los pines que intervienen en la comunicación.

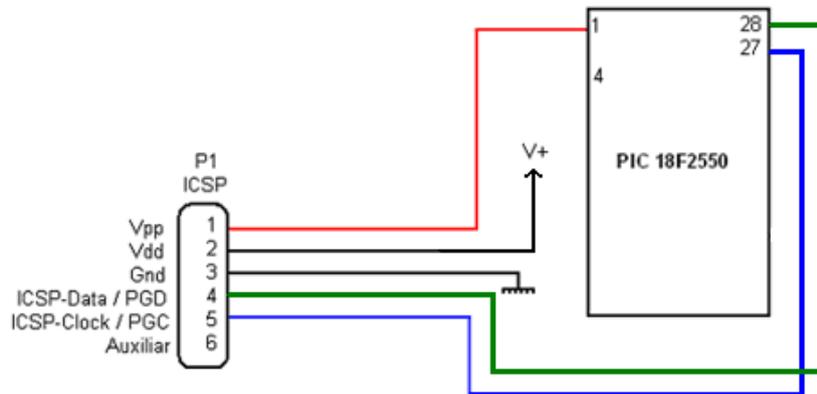


Figura 4.29. Esquemático de la comunicación entre el ICSP y el PIC18F2550

4.5.11 Diseño completo del hardware de control

En este Apartado se incluyen los esquemas eléctricos del circuito implementado completo. Se ha simulado el funcionamiento del circuito completo con el *software Proteus 7*, para comprobar su viabilidad y se ha demostrado el funcionamiento correcto con los componentes propuestos. A partir de los esquemáticos se han diseñado las placas de circuito impreso, es decir, se han rutado pistas adecuadas para la implementación práctica con el *software Orcad 10.0*.

Se han agrupado los componentes en dos placas. En la primera placa, la placa del motor, se incluyen el PIC18F2550, el driver L293, la pantalla LCD, el conector USB, el puerto ICSP, el regulador LM7805 y los detectores de fin de carril. Los esquemáticos se pueden ver en las Figuras 4.30 y 4.31.

En la Figura 4.32 se puede ver el esquemático completo de la segunda placa, la placa de pulsadores, donde están instalados los pulsadores de control y un conector para comunicación con la primera placa.

Los fotolitos de las dos placas se pueden encontrar en los Anexos, al final de este documento.

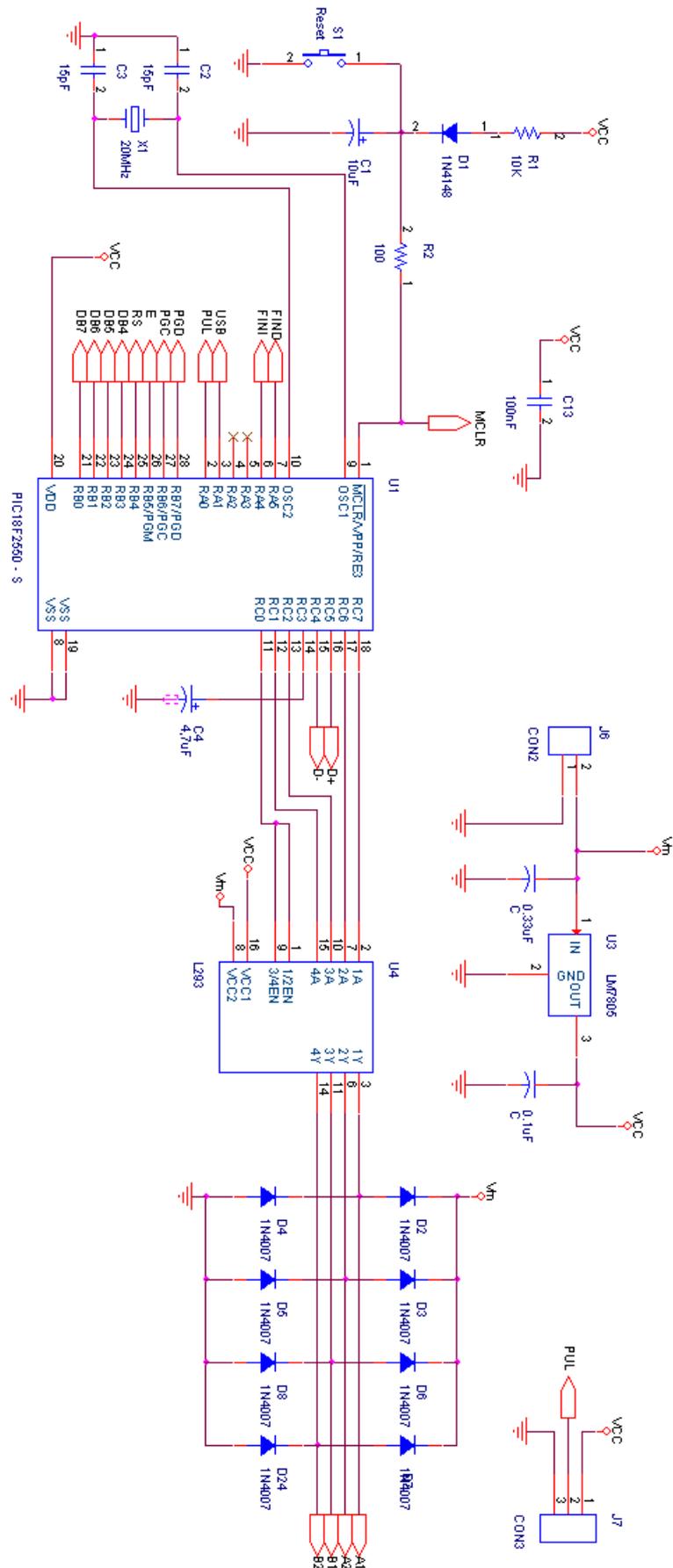


Figura 4.30. Esquemático del hardware de control. Placa del motor: microcontrolador, driver L293, circuito de RESET, regulador LM7805 y conector

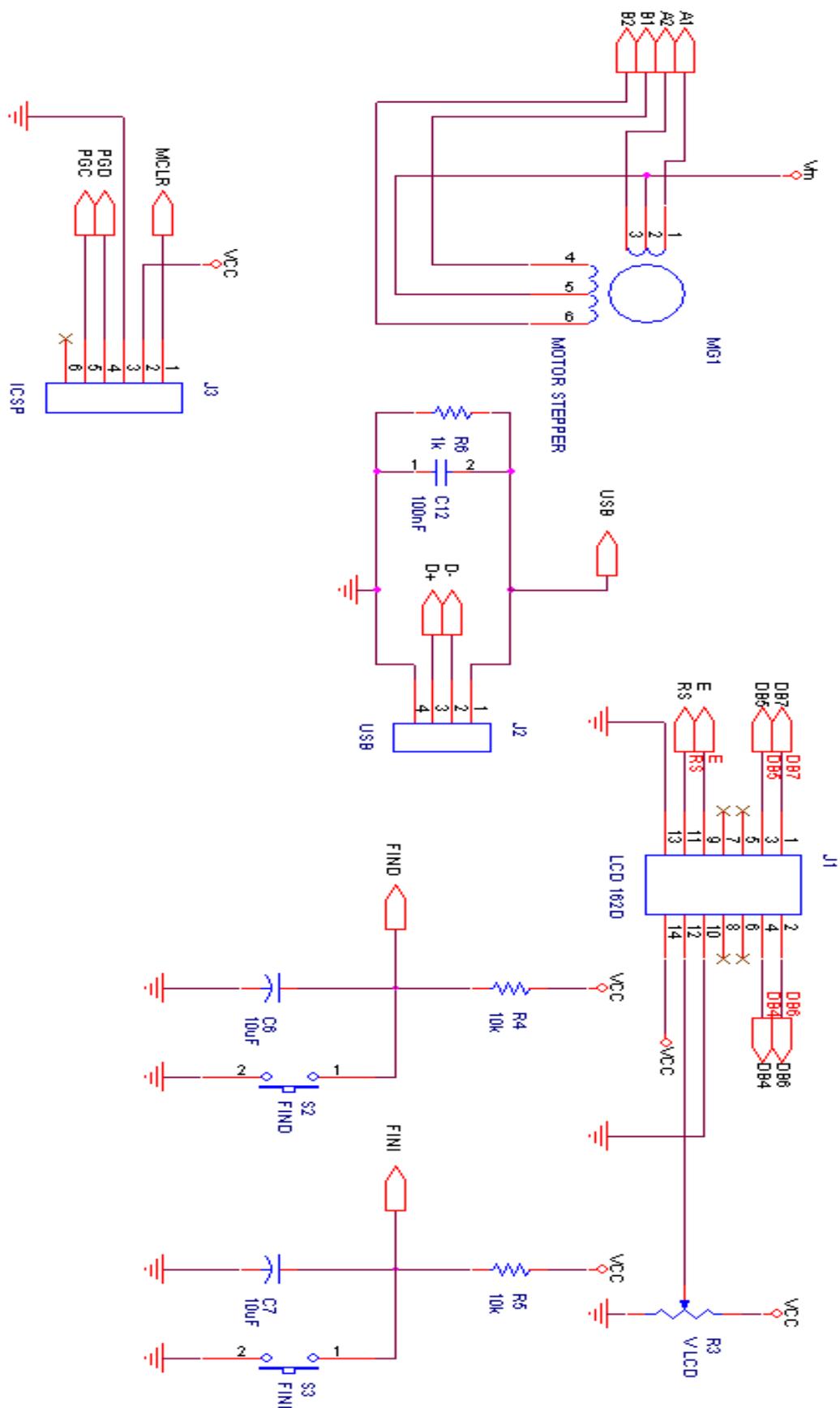


Figura 4.31. Esquemático del hardware de control. Placa del motor: motor, USB, ICSP, LCD y fines de carril derecho e izquierdo

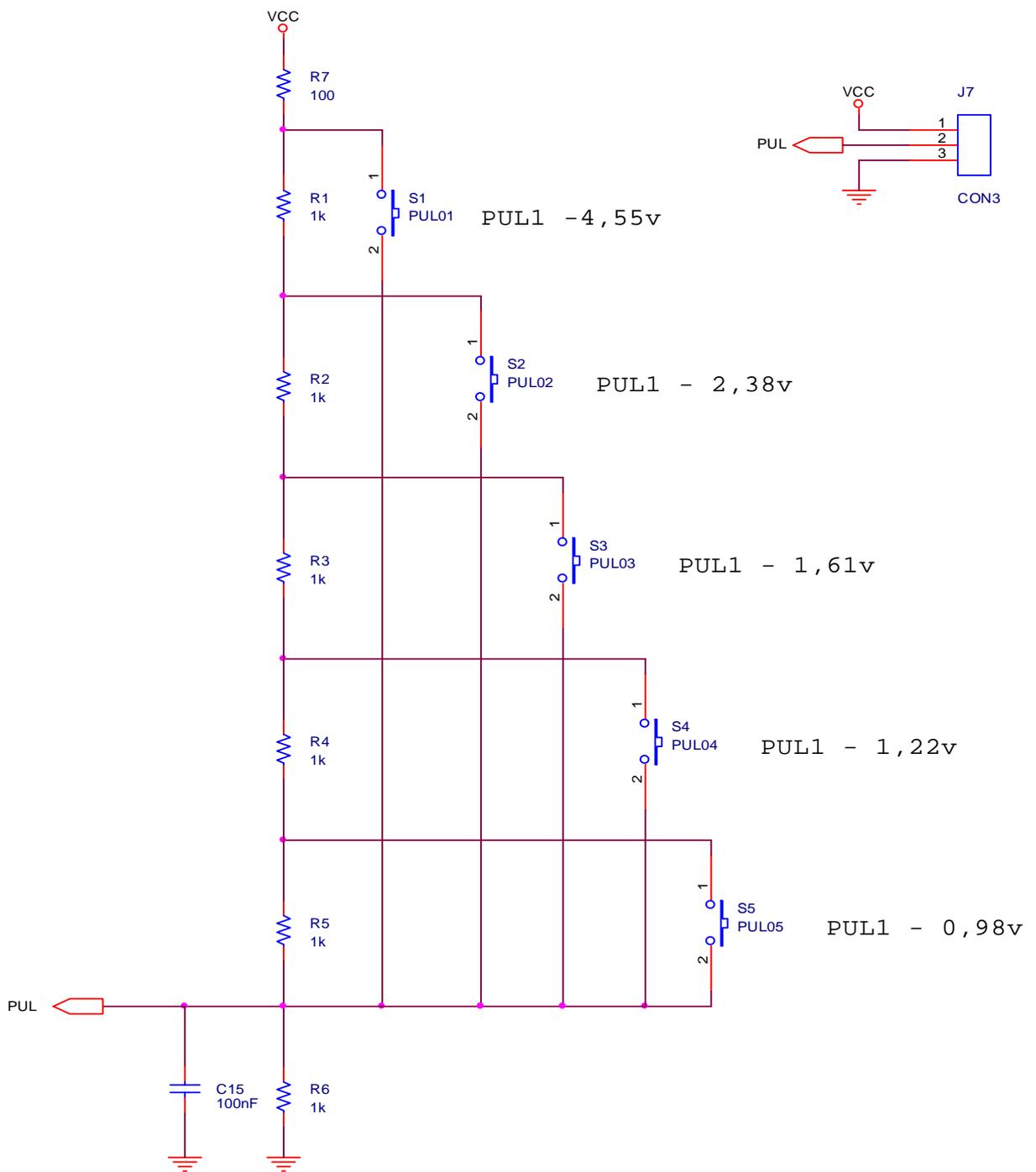


Figura 4.32. Esquemático del hardware de control. Placa de pulsadores: pulsadores y conector

CAPÍTULO 5

SOFTWARE DE CONTROL DEL SISTEMA MOTORIZADO

5.1 DESCRIPCIÓN DE LOS DISPOSITIVOS DEL SISTEMA

En este Capítulo se ahondará en el diseño del código de control del sistema posicionador completo. En la Figura 5.1 se representan, a modo de bloques, los elementos del sistema que precisan de la elaboración de cierto código de control para comunicarse con el resto.

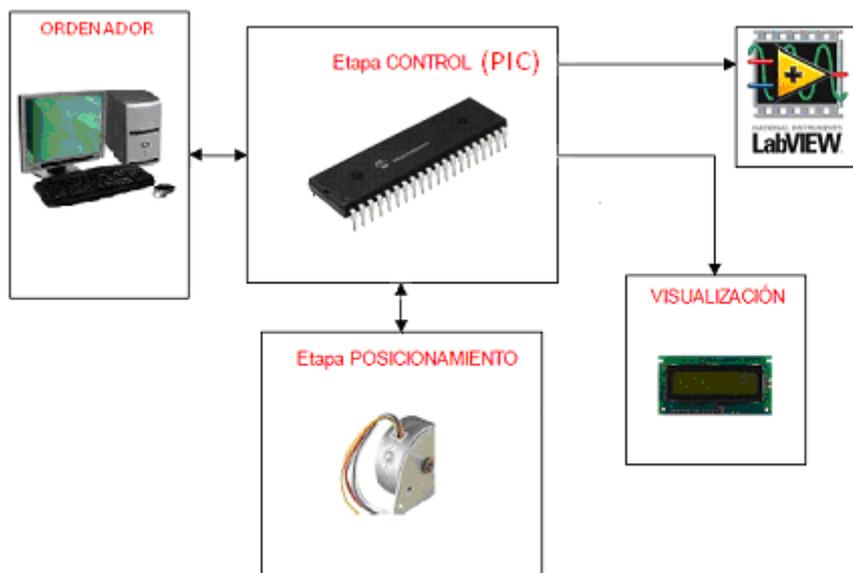


Figura 5.1. Comunicación entre dispositivos del sistema motorizado de posicionamiento.

Se observa en el diagrama de bloques que, desde el ordenador, se transmite el código de programación al PIC. Este PIC controlará la posición concreta en la que se desea que se sitúe el posicionador. Este mismo controlador mostrará en la pantalla LCD la posición en la que está y ofrecerá la posibilidad de poder desplazarse a otras posiciones. Existe además comunicación en sentido contrario entre el sistema de posicionamiento y el PIC, cuando le informa sobre el estado los pulsadores o los finales de carrera.

Simultáneamente a este Proyecto, se está desarrollando un sistema para la comunicación y monitorización de la posición del microposicionador con el tratamiento

de datos mediante LabView. En este sentido, existe flujo de información y control del PIC hacia el PC cuando, en “modo con PC”, el microcontrolador notifica al ordenador la posición actual del motor a través del Hiperterminal.

Para realizar las tareas de programación especificadas, en este Capítulo se justificará el lenguaje de programación escogido, se describirán los programas necesarios para compilar el código y se desglosarán los diagramas de flujo del mismo para cada una de las funciones implementadas. A continuación se describe brevemente cada una de estos aspectos.

- Los lenguajes de programación son el conjunto de órdenes o instrucciones dadas para que la máquina entienda la tarea que debe realizar. Están formados por conjuntos de símbolos y reglas que ordenados convenientemente hacen que las instrucciones tengan significado pleno. Al proceso por el cual se diseña, se escribe, se prueba, se compila y se depuran posibles errores, se le llama *programación*.
- Un programa es un conjunto de componentes lógicos de instrucciones que ha de ejecutar un ordenador para realizar tareas específicas.
- Un diagrama de flujo es una representación gráfica de un algoritmo o proceso. Se utiliza en diferentes ámbitos, como la programación, ya que a través de unos símbolos sencillos se representa el flujo de ejecución de un proceso utilizando para ello flechas que conectan los puntos de principio y de fin.

5.2 LENGUAJES DE PROGRAMACIÓN DEL PIC

Para escribir programas para un microcontrolador se puede escoger entre varios lenguajes de programación: de bajo nivel (Ensamblador) y de alto nivel (BASIC ó C).

El lenguaje Ensamblador es un lenguaje de bajo nivel donde una colección de símbolos representa las instrucciones del código máquina. El ordenador entiende el

lenguaje binario, y el lenguaje Ensamblador es el que más se acerca al lenguaje binario. Es un lenguaje difícil de depurar y requiere más instrucciones para realizar el mismo proceso que un lenguaje de alto nivel, aunque ocupa menos espacio.

El lenguaje BASIC se caracteriza por su sencillez, siendo por lo tanto fácil de aprender. Su mayor defecto es que se puede saltar a cualquier parte del programa usando la instrucción GOTO, lo que dificulta la comprensión del código.

El lenguaje C es uno de los lenguajes más potentes para la programación de alto nivel en microcontroladores. Con respecto a BASIC, posee la ventaja de ser estructurado y además tiene gran cantidad de librerías. También permite introducir código en ensamblador dentro del programa. Existen varios compiladores que poseen abundantes librerías para todo tipo de dispositivos. Su principal desventaja es la implícita a cualquier lenguaje de alto nivel: poco control sobre el código generado, lo que no permite optimizar el código. Además, lo hace más difícil de depurar.

En conclusión, el Ensamblador es el lenguaje óptimo para programas de poca entidad, con PICs pequeños, donde se busque la rapidez y sencillez. Así mismo permite depurar el código línea por línea. Si se necesita realizar programas más complejos, se podría empezar con BASIC, ya que es más sencillo de aprender, pero conlleva dificultades la depuración y programación del código. Debido a que ya se poseen conocimientos mínimos de programación en C, y a la facilidad que ofrecen sus instrucciones, se ha escogido para este desarrollo el lenguaje C.

Una vez decidido el lenguaje de programación, el siguiente paso es escoger las instrucciones para programar el código y el posterior traslado al PIC.

5.3 CONFIGURACIÓN DE LA PROGRAMACIÓN

En este Apartado se explicarán con detalle los pasos a seguir para la creación de un nuevo proyecto: elección del dispositivo, compilaciones y traslado del archivo ejecutable al programador.

Todos estos pasos son necesarios para el correcto funcionamiento tanto del código como de los programas, y se asegura un traslado sin errores del código al PIC.

5.3.1 Grabación de los PICs

Una vez depurado de los posibles errores, el código se traslada del ordenador al PIC18F2550 a través de un dispositivo llamado programador. La mayoría de los PICs de la empresa Microchip incorporan un programador ICSP (*In Circuit Serial Programming*, programación serie incorporada) o LVP (*Low Voltage Programming*, programación a bajo voltaje), que permiten el traslado fácil del código. Una de las ventajas del ICSP y del LVP es que se puede transferir el código en el mismo circuito de destino, sin necesidad de extraer el microcontrolador de la placa y programarlo fuera.

5.3.2 Creación de un proyecto con CCS PIC C Compiler

Una vez abierto el programa, para crear el Proyecto se selecciona "PIC Wizard" del menú "Project". Tras indicar el nombre y la dirección del Proyecto, se selecciona en la pestaña "General", el PIC18F2550, la frecuencia de reloj de 48000000Hz y Cristal Osc \leq 4MHz (ver Figura 5.2.).

Por último, en la sección "Communications" se deshabilita el RS-232. El programa crea un archivo ".C" y un archivo ".hex".

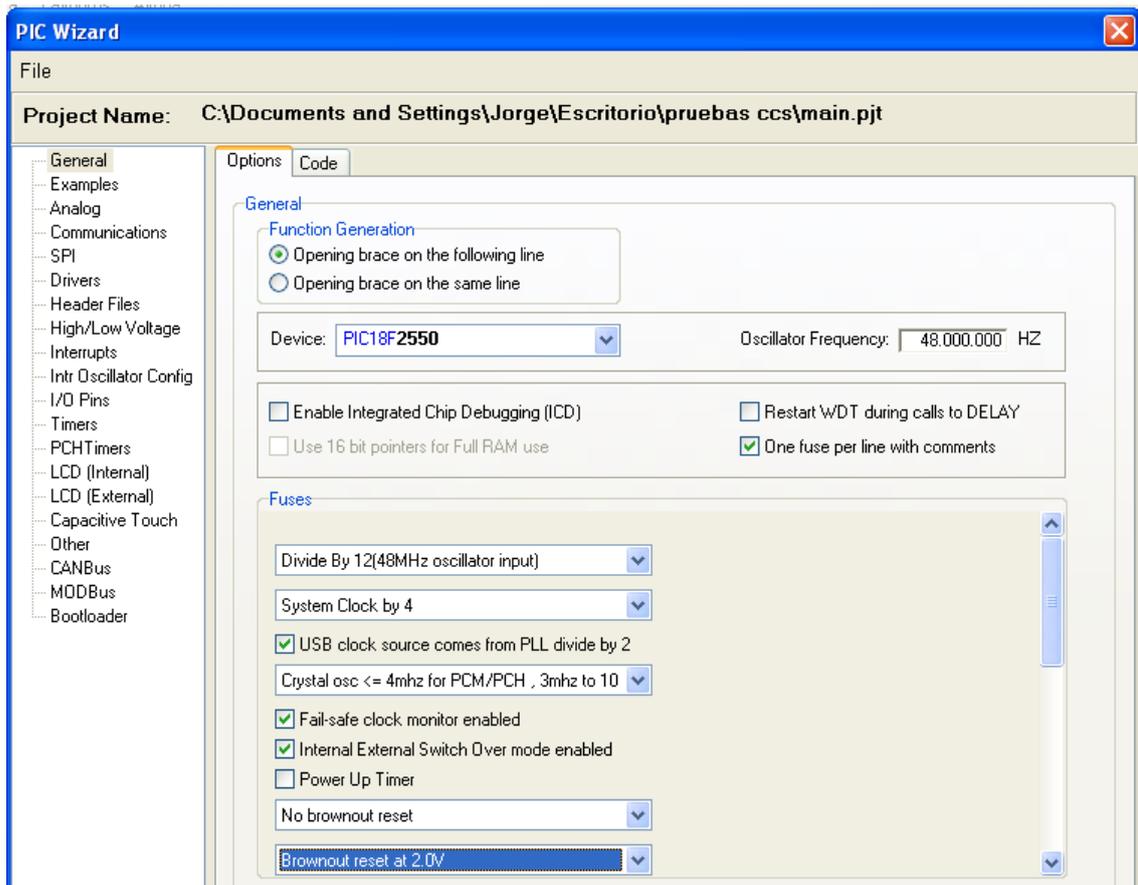


Figura 5.2. Ventana de PIC Wizard para la selección del dispositivo

A continuación, se introducen las librerías de la LCD, USB y el PIC18F2550 en la carpeta del proyecto y también se incluyen en la cabecera del programa, como se puede observar en la Figura 5.3.

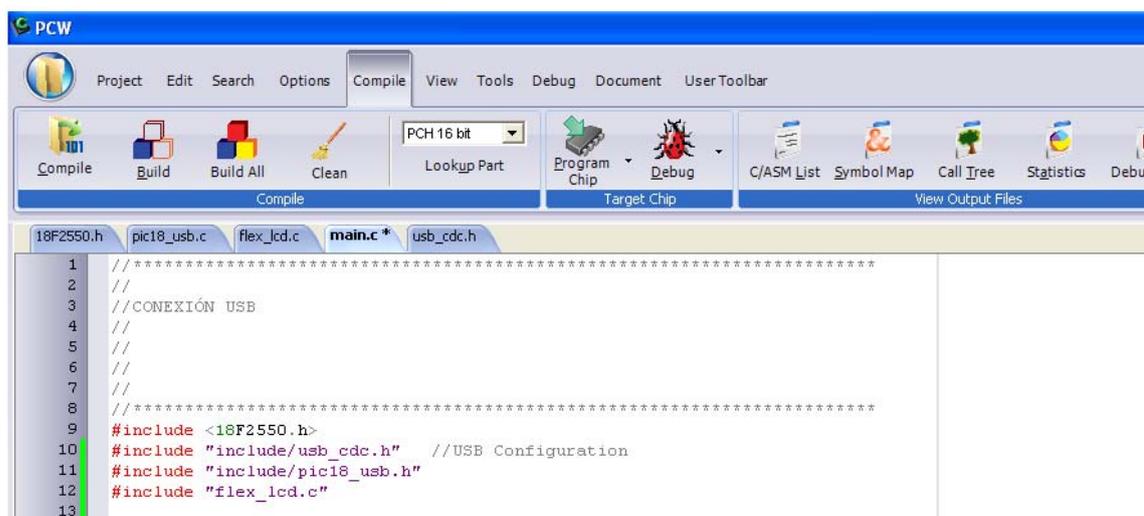


Figura 5.3. Detalle de la ventana de PCW para la selección de librerías

Se escribe el programa y se compila mediante la opción “Compile” del menú “Compile”. Con la opción “Build All”, el programa crea el archivo “.hex” que se grabará al microcontrolador.

5.3.3 Creación de un proyecto con MPLAB

Antes de abrir el programa MPLAB es necesario que MPLAB ICD2 esté conectado de la forma que se indica en la Figura 5.4. Se ha de poner en contacto el ordenador con el PIC, previamente al envío del archivo al microcontrolador. El circuito completo se alimenta a +8V.



Figura 5.4. Esquema del conexionado entre el ordenador, el programador y el microcontrolador

El programador MPLAB ICD2 tiene en uno de los extremos una toma USB para conectarlo al ordenador y en el otro extremo posee una salida cableada que termina en un conector RJ11 macho a insertar en un conector hembra habilitado para ello en la placa (ver Figura 5.5).



Figura 5.5. Programador MPLAB ICD2

A la salida del RJ11 hembra se añade una tira de conectores de 6 pines dirigida al puerto de comunicaciones ICSP. La configuración del conector en el puerto del microcontrolador se ha mostrado en el Capítulo anterior (véase Apartado 4.5.10).

Una vez abierto el programa, se selecciona el PIC; para ello se entra en “Configure” y, en el menú desplegable, se elige “Select Device”, como se muestra en la Figura 5.6.

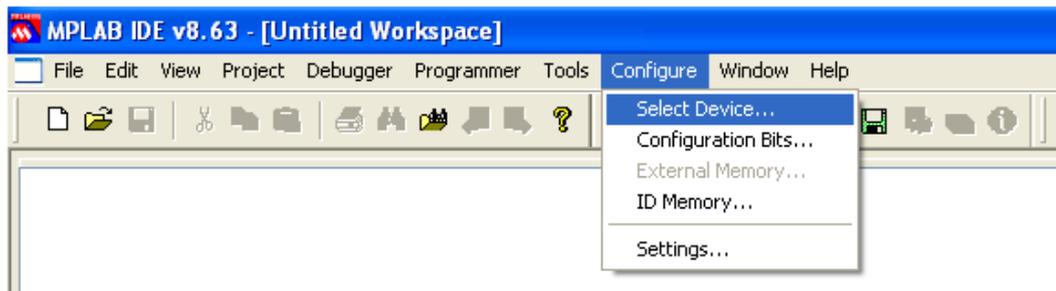


Figura 5.6. Menú principal MPLAB

Aparecerá el menú que se muestra en la Figura 5.7, se selecciona el dispositivo y se pulsa “OK”.

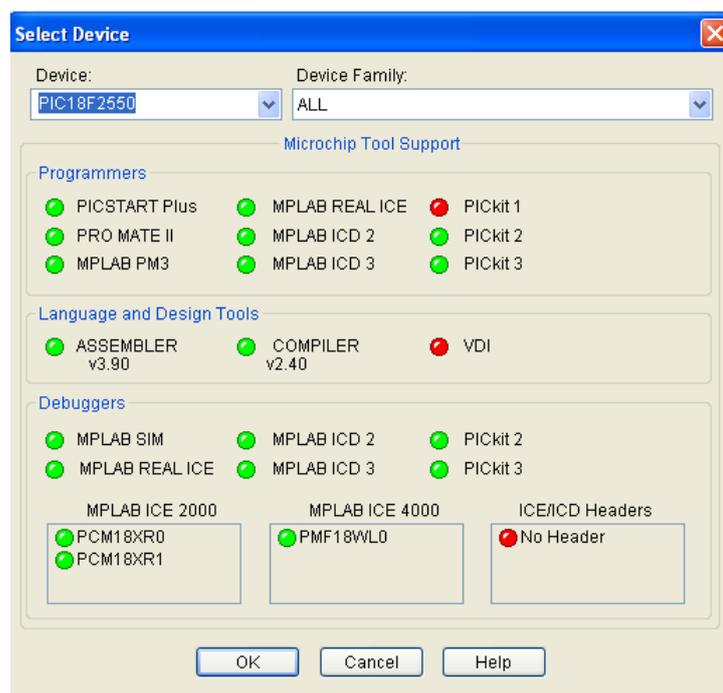


Figura 5.7. Ventana de selección del dispositivo en MPLAB

Una vez seleccionado el dispositivo, el siguiente paso es coger el archivo “.hex” generado. Para ello se selecciona “File”, y en el menú desplegable que aparecerá se pincha en “Import...”. Allí se encuentra el archivo con extensión “.hex”, a importar por el microcontrolador (ver Figura 5.8).

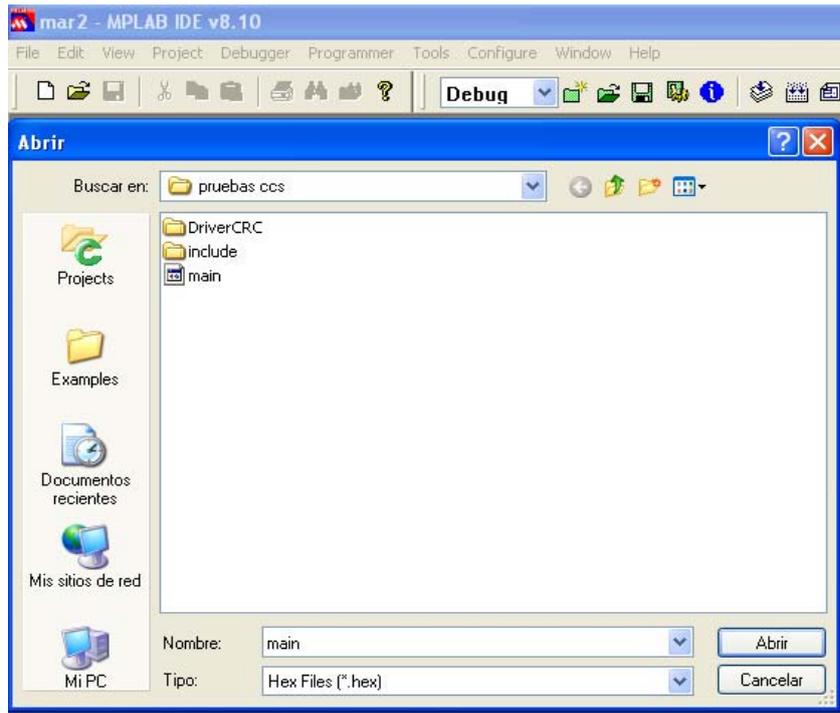


Figura 5.8. Detalle de la ventana para la carga del archivo “.hex”.

El siguiente paso es programar el microcontrolador seleccionando “Programmer”. En el menú desplegable que aparecerá, se selecciona “Select Programmer” y después se elige “MPLAB ICD2” (ver Figura 5.9).

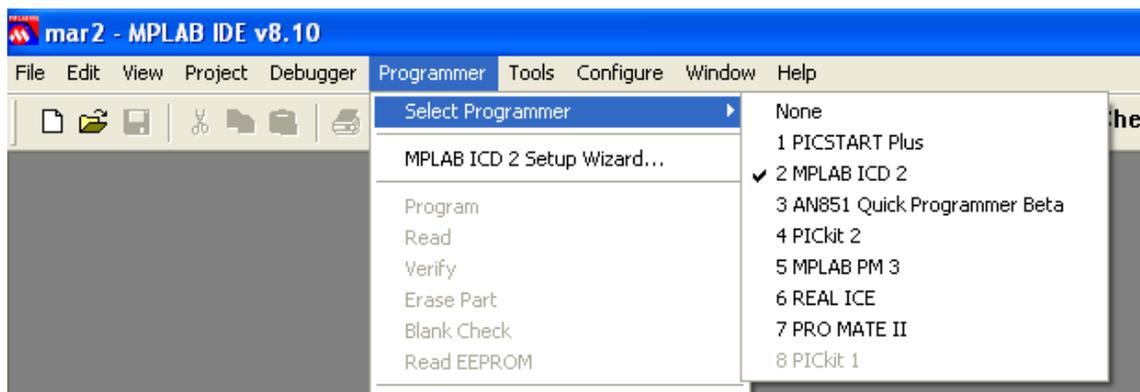


Figura 5.9. Selección de programador MPLAB ICD2

Tras realizar estos pasos, ya se puede transferir el programa. Para ello, de nuevo en “Programmer” y en el menú desplegable en “Connect” se controla la comunicación entre ordenador y PIC18F2550. Una vez hecho esto, en el propio menú “Programmer”

se pinchará en “Program”. Finalmente, si todo ha salido correctamente aparecerá una pantalla similar a la de la Figura 5.10, que confirma el éxito en la introducción del código necesario al PIC.

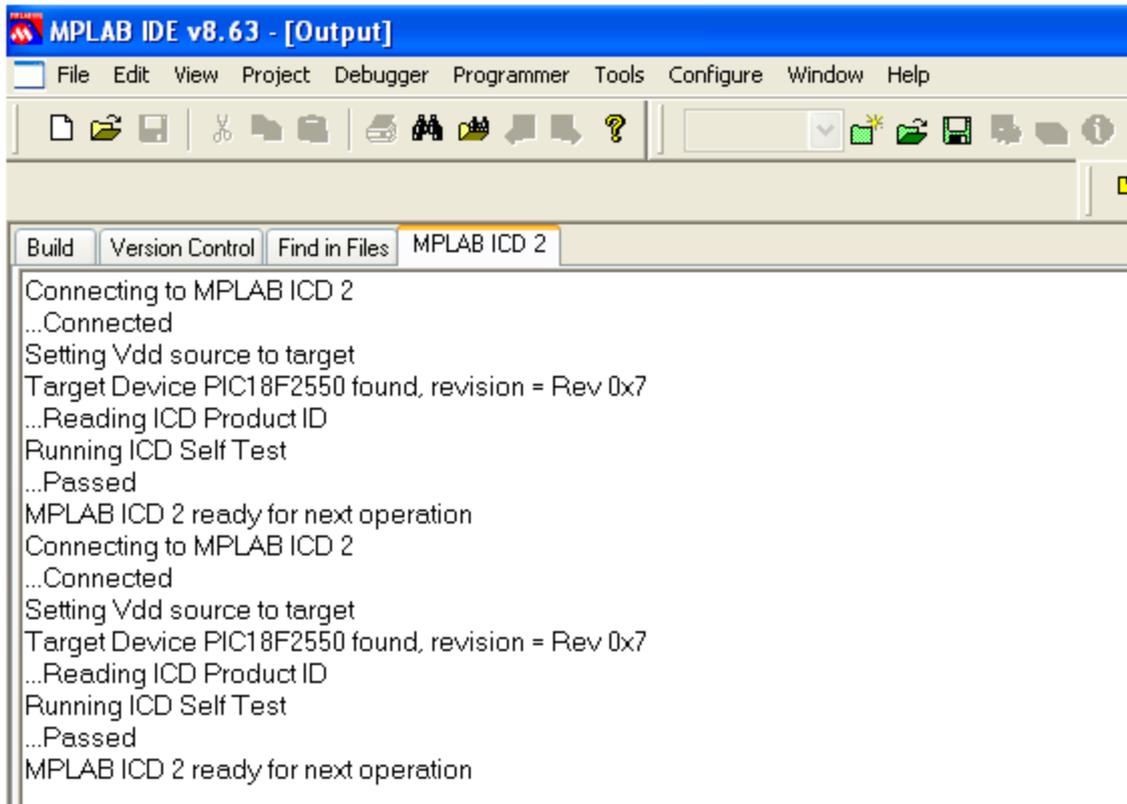


Figura 5.10. Éxito en la programación del PIC

Una vez descritos los programas de ayuda a la programación hardware del microcontrolador y los modos de empleo, a continuación se profundiza en los diagramas de flujo del código.

5.4 DIAGRAMA DE FLUJO DEL CÓDIGO PROGRAMADO

En este Apartado se detalla específicamente el diseño del programa de control para el sistema microposicionador. Se van a analizar, mediante diagramas de flujo, todas las funciones realizadas para conseguir el correcto funcionamiento del dispositivo.

Se han creado las siguientes funciones: "Main", "Configuración", "Test LCD", "Posición_actual", "Leer_pulsadores", "Paso_Motor", "Posición_izquierda" y "Guarda_eeeprom". Dentro de cada diagrama de flujo aparecen 5 tipos de símbolos (ver Figura 5.11), cuyo significado es el siguiente:

- el primero (inicial), de color rojo, es el de inicio de función.
- el rectángulo de color azul oscuro, es una llamada a una función.
- el rectángulo azul claro, es una instrucción.
- el rombo verde, es un condicional.
- el último (final), de color granate, es la vuelta a la función principal.



Figura 5.11. Simbología utilizada en los diagramas de flujo

5.4.1 Diagrama de flujo de la función "MAIN"

Debido a que el diagrama de flujo completo, incluyendo la parte relativa a los pulsadores, es muy extenso, se ha descompuesto su parte final en diagramas de flujo más pequeños, correspondientes a cada pulsador, para que sea más fácil su entendimiento y comprensión.

El desglose de los diagramas de flujo se representa en varias figuras. En la Figura 5.12 se muestra el diagrama de flujo de función MAIN con las inicializaciones de puertos, USB y LCD. En la Figura 5.13 se muestra el diagrama de flujo de la función MAIN con la comprobación de RESET y USB conectado. Por último, en la Figura 5.14 se muestra el diagrama de flujo de la función MAIN y los pulsadores.

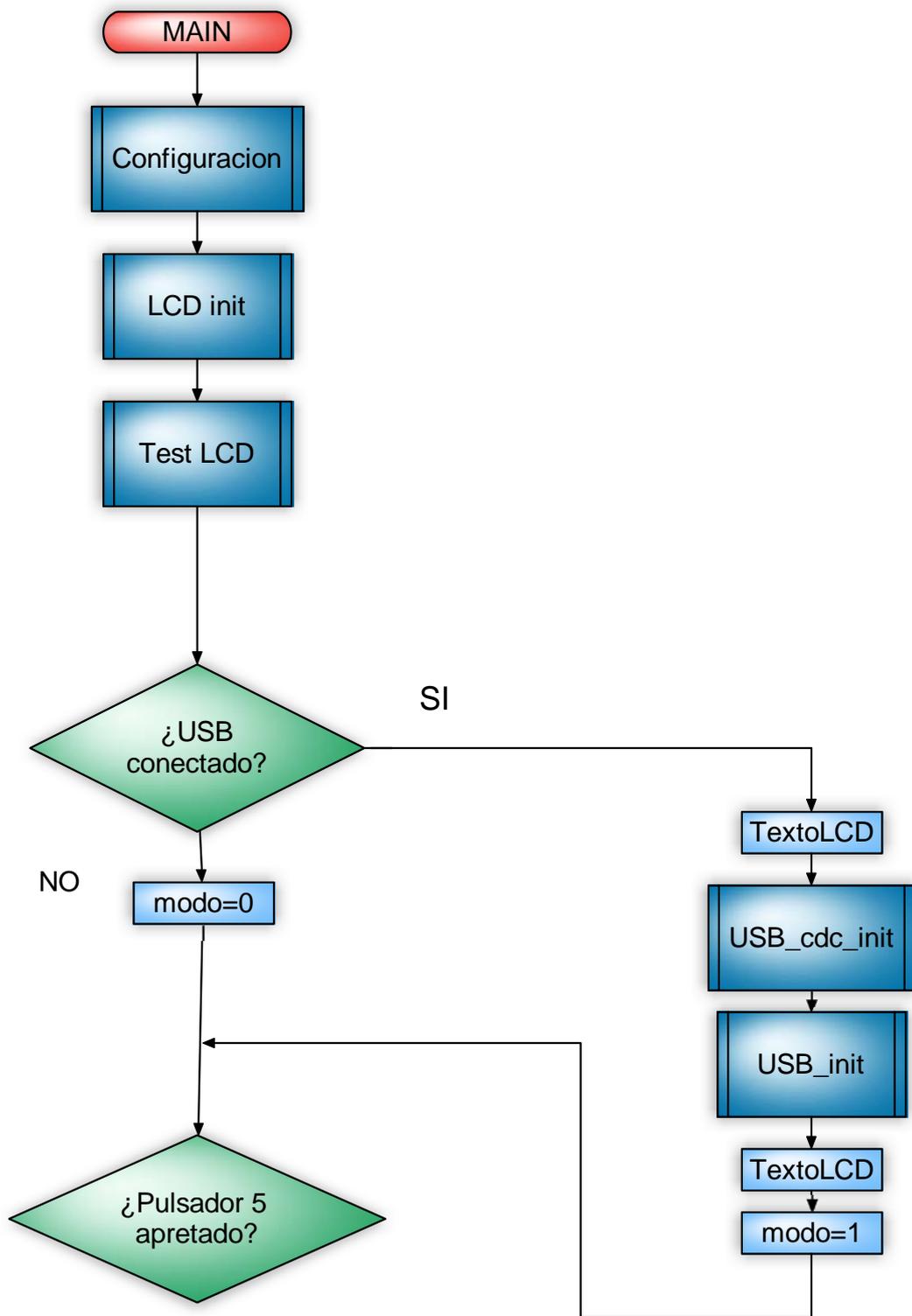


Figura 5.12. Diagrama de flujo función MAIN: inicializaciones puertos, USB y LCD

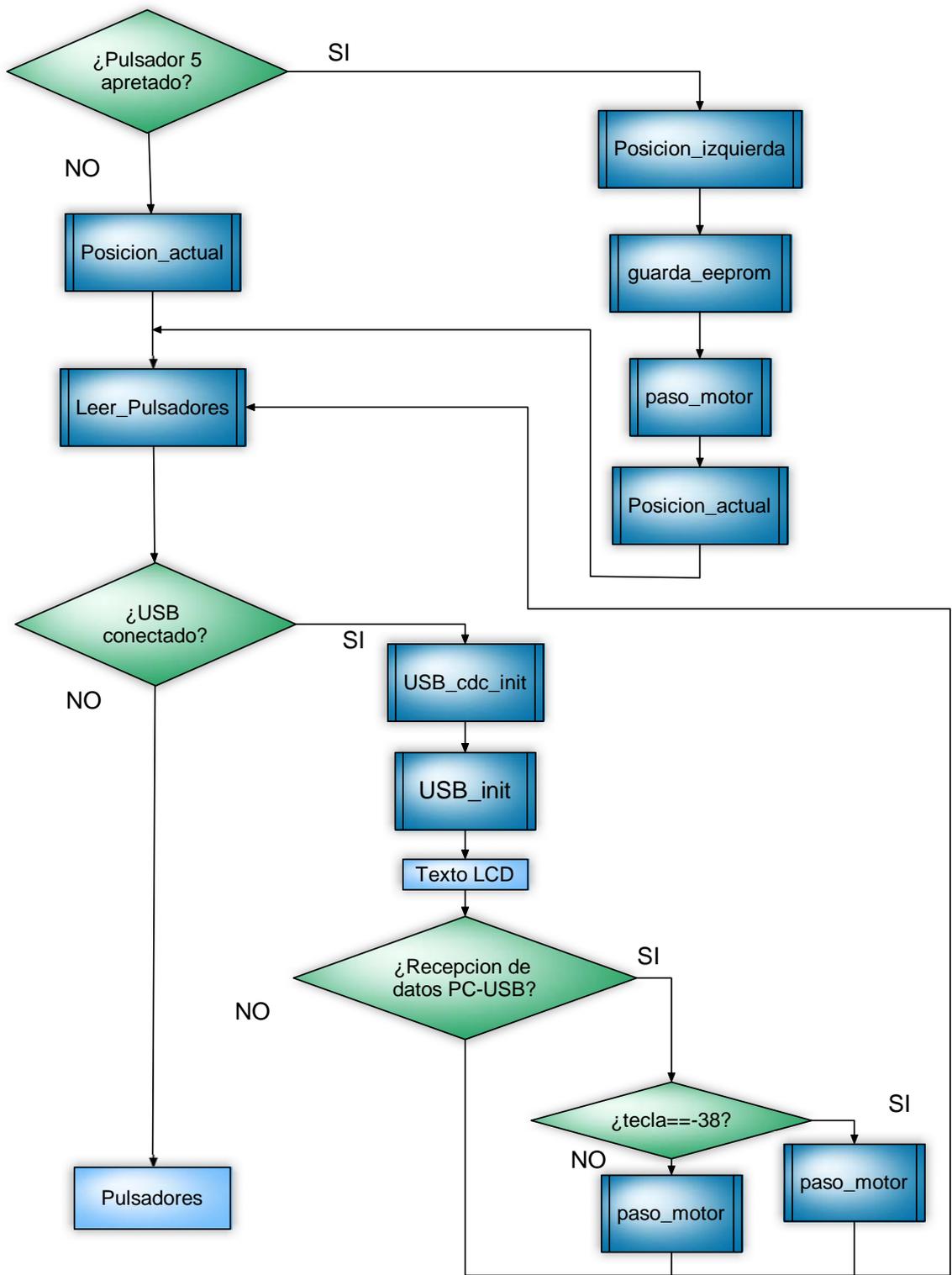


Figura 5.13. Diagrama de flujo función MAIN: Comprobación RESET, USB conectado

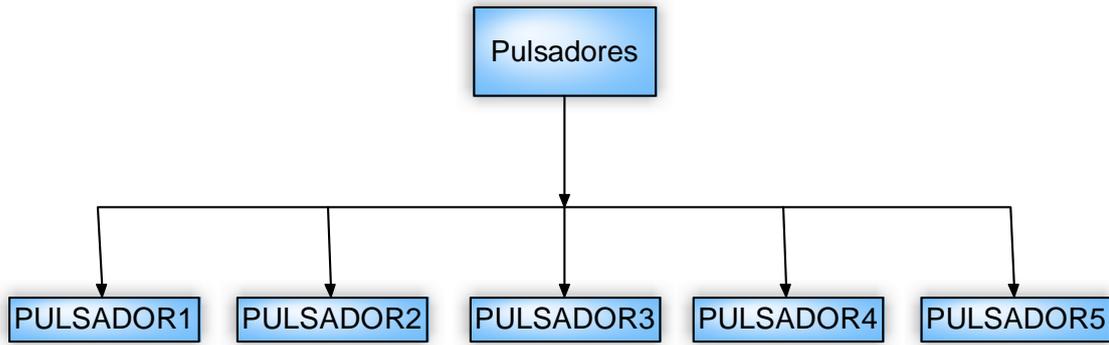


Figura 5.14. Diagrama de flujo función MAIN: pulsadores

Como se puede apreciar en el diagrama, “Main” comienza haciendo unas llamadas a las funciones “Configuración”, “LCD init“ y ”Test LCD”, que son inicializaciones y configuraciones tanto de la LCD como de los puertos del microcontrolador.

Realizadas las inicializaciones, se comprueba si el cable USB está conectado; si es así, se detectará la tensión en el pin 5 del microcontrolador y aparecerá el texto en la LCD: “USB conectado” (ver Figura 5.15). Una vez sacado el texto por la LCD, comenzará con la inicialización y configuración del USB mediante las funciones “USB_init” y “USB_cdc_init”.

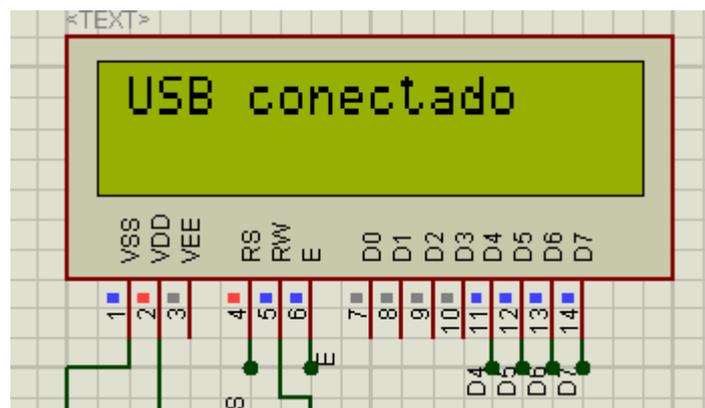


Figura 5.15. Pantalla LCD. Mensaje “USB conectado”

Mientras el USB esté conectado, se guardará en una variable como *modo=1*. Si el cable USB está desconectado y se intenta enviar información al Hyperterminal del PC, el PIC18F2550 se queda ejecutando infinitamente esta instrucción hasta que, de nuevo, el USB esté conectado. Por esta razón, cada vez que el microcontrolador quiera enviar un mensaje al Hyperterminal confirmará que *modo=1* y la enviará.

Una vez realizada la comprobación del USB, avanzará el programa y confirmará si está presionado el Pulsador 5, PROGRAMA de RESET. La funcionalidad que realiza el PROGRAMA DE RESET es, en primer lugar, desplazar la vagoneta hasta el final del carril izquierdo. Seguidamente, y cuando la vagoneta está en dicho extremo, avanzar hasta la posición 2500, aproximadamente la mitad del recorrido total. De este modo se consigue situar rápidamente la vagoneta a la mitad del carril, lo que es útil en los casos en que esta posición sea preferida.

A continuación, tanto si se ha seleccionado el Pulsador 5 como si no, se llama a la función "Posicion_actual". "Posicion_actual" será explicada más adelante en detalle junto con su diagrama de flujo; a través de esta función se conoce siempre la posición de la vagoneta dentro del carril. Una vez hecho esto, el programa principal se mete en un bucle infinito donde comprueba si se ha pulsado alguno de los pulsadores (modo de funcionamiento autónomo) o se ha conectado el USB (modo de funcionamiento a través de PC), en caso de que no se hubiera conectado antes. Mientras no se pulse nada ni se conecte nada, mostrará por pantalla la posición actual.

Si se conecta el cable USB, el microcontrolador comenzará con la inicialización y configuración del USB mediante las funciones "USB_init" y "USB_cdc_init". Terminadas las inicializaciones y configuraciones del USB, aparecerá un mensaje en la LCD avisando al usuario para que abra el Hyperterminal del ordenador. Una vez hecho esto, comprobará si hay recepción de datos desde el ordenador al PIC18F2550. Los datos se envían a través de la información de las teclas pulsadas desde el teclado del ordenador. Se ha comprobado que al presionar cualquier tecla del teclado del ordenador y recoger el dato, éste viene dado en código ASCII. Se han elegido las teclas "1" e "INTRO" para desplazar la vagoneta en cada uno de los dos sentidos. Por tanto, si se pulsa cualquiera de las dos teclas, la vagoneta se desplazará. Si la tecla apretada no fuera ninguna de estas dos, se volverá a "Leer_pulsadores" esperando a

que el usuario presione cualquier pulsador (modo autónomo) o que se pulse el teclado del ordenador (modo desde PC).

En el caso de seleccionar cualquier pulsador, la función “Leer_pulsadores” devolverá un valor de 1 a 5, que se corresponde con cada uno de los 5 pulsadores (cursores más INTRO). Esta función, junto con su diagrama de flujo, será explicada más adelante. Dependiendo del pulsador presionado, hará una tarea diferente.

A continuación se detallan las funcionalidades del sistema según se presione uno u otro pulsador en modo autónomo.

En la Figura 5.16 se muestra el diagrama de bloques del Pulsador 1.

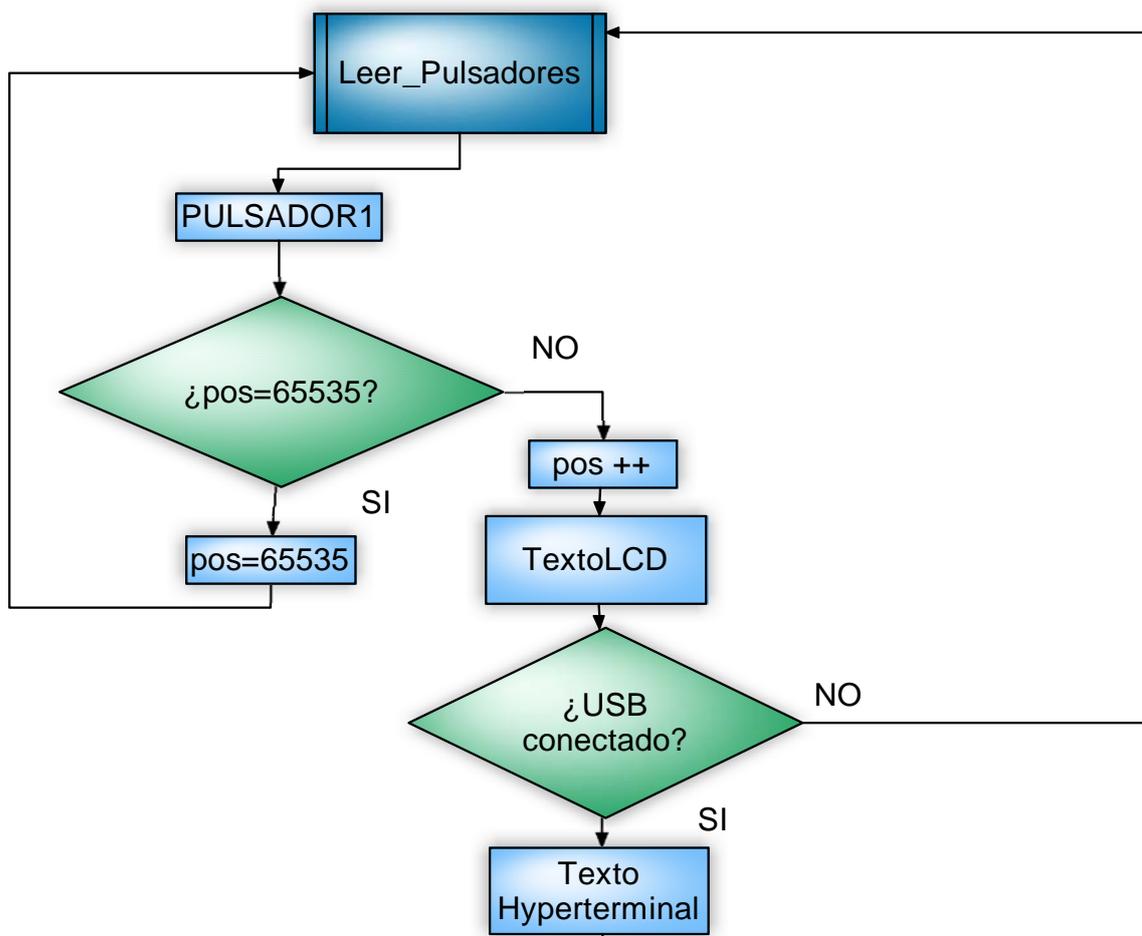


Figura 5.16. Diagrama de flujo del PULSADOR1

Antes de explicar el funcionamiento del Pulsador 1 conviene aclarar que se ha llamado “aux2” a la posición real en la que se encuentra y “pos” a la posición a la que se le quiere enviar.

En el caso de que se haya apretado el Pulsador 1, comprobará si $pos = 65535$, ya que esto significaría que la vagoneta ha llegado al final del carril por la derecha y no puede continuar. Mientras no se llegue a este valor, se puede incrementar el valor de la variable pos . Mostrará en la pantalla LCD los mensajes de posición actual y posición donde queremos ir (ver Figura 5.17) y, en caso de estar conectado el USB, mandará el mismo mensaje al ordenador.

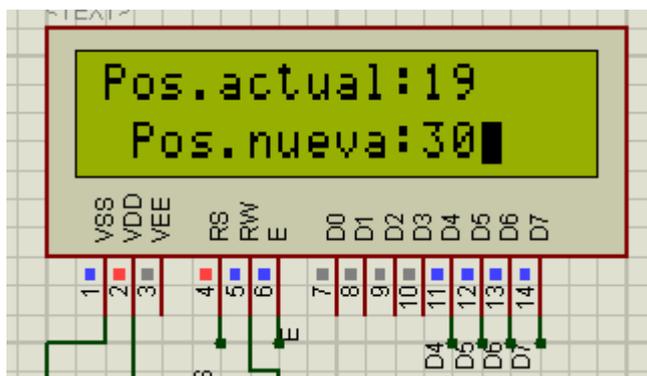


Figura 5.17. Pantalla LCD. Mensajes de “Posición actual” y “Posición nueva”

Por ejemplo, si se encuentra en la posición real 19, al apretar este pulsador aparecerá la posición actual en la primera fila de la pantalla LCD y en la segunda fila aparecerá el valor 19 incrementado en tantas unidades como veces se haya pulsado el Pulsador 1 ó se deje apretado el pulsador. La pantalla muestra la posición actual y el posible destino, pero la vagoneta no se desplazará a la nueva posición hasta que se accione el Pulsador 5 (“intro”). En el pulsador 5 se hace el INTRO y en el mismo pulsador 5 se implementa el programa de RESET. El Pulsador 1 está diseñado para desplazamientos largos a derechas, por tanto el contador pos se incrementará a alta velocidad.

Seguidamente, se describe la función realizada por el Pulsador número 4. En el caso de apretarse el Pulsador 4, ocurre lo mismo que si se accionara el Pulsador 1 pero con movimientos de la vagoneta hacia la izquierda y, en este caso, disminuirá la variable “pos”. Se puede observar la similitud entre los diagramas de flujo del Pulsador 1 y el 4 (ver Figura 5.18).

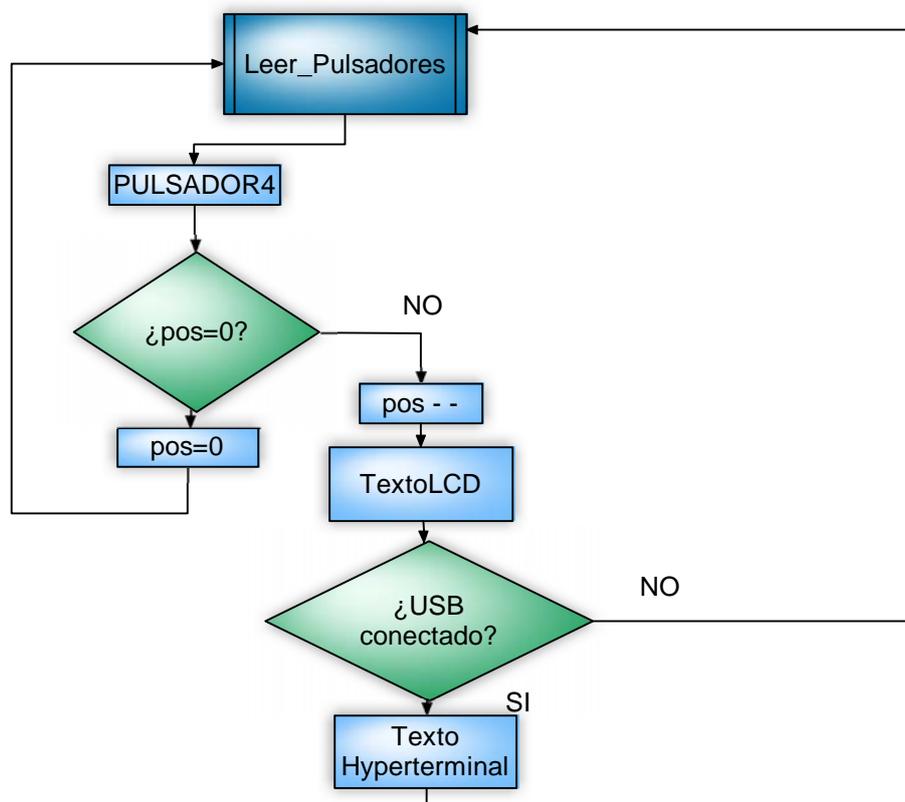


Figura 5.18. Diagrama de flujo del PULSADOR4

Seguidamente, se describe la función realizada por el Pulsador número 2. En la Figura 5.19 se incluye el flujo que sigue el programa al accionar el Pulsador 2.

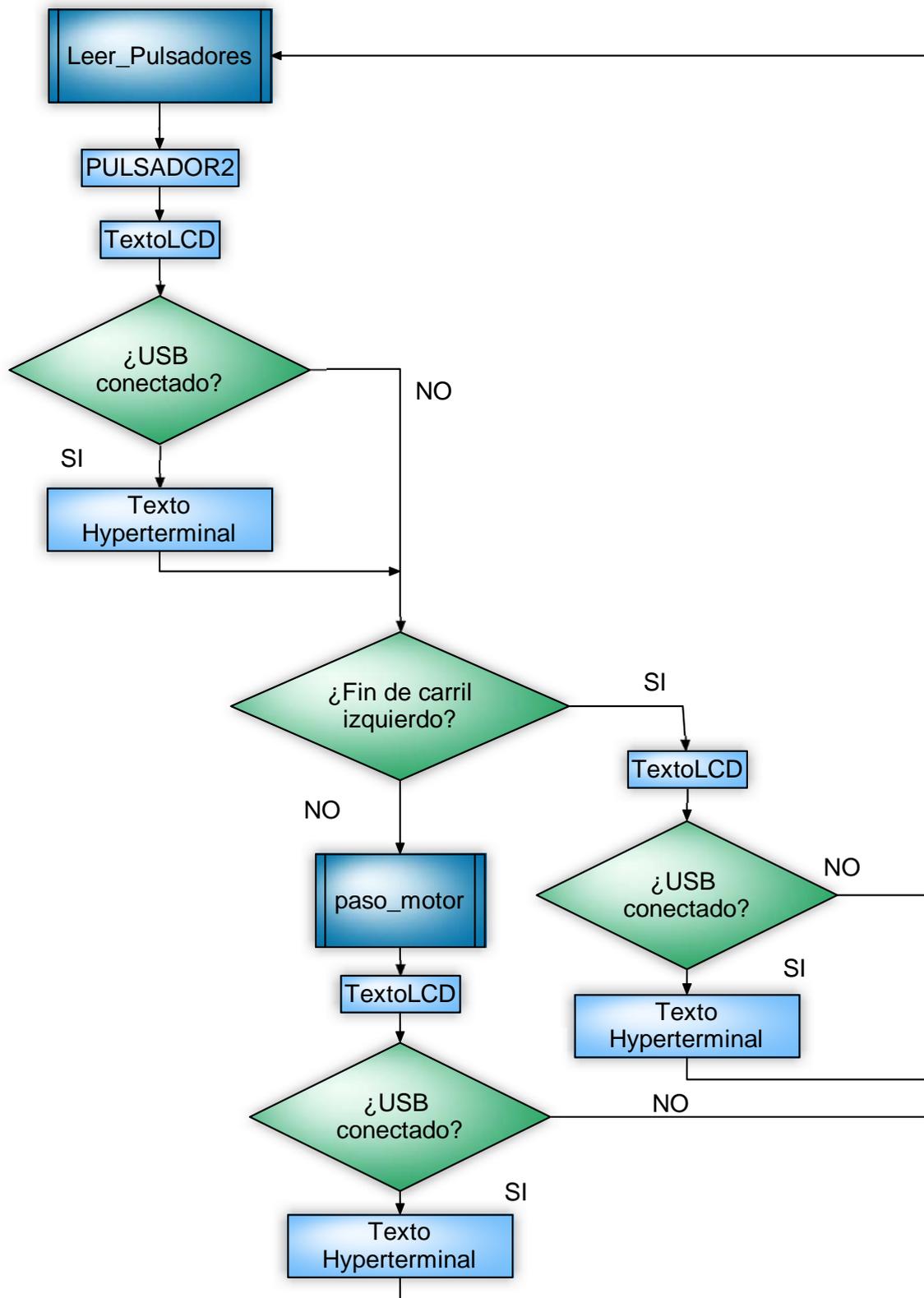


Figura 5.19. Diagrama de flujo del PULSADOR2

El Pulsador 2 se ha diseñado para desplazamientos cortos, por tanto cada vez que se pulse moverá un paso la vagoneta a la izquierda. Se le ha introducido un retardo al pulsador para evitar que, si se deja pulsado mucho tiempo, se acumulen muchos desplazamientos. En caso de apretarse el Pulsador 2, aparecerá en la pantalla LCD un texto avisando de que se ha pulsado el pulsador izquierdo. Comprobará si se ha llegado al final del carril izquierdo y, si no es así, se llamará a la función “paso_motor”. Esta función será explicada más adelante junto con su diagrama de flujo, pero se puede adelantar que se encarga de mover el motor.

Después, mostrará por la pantalla LCD la posición actual y, en caso de estar conectado con el Hyperterminal, lo mostrará en el ordenador. Si se ha llegado al final del carril, mostrará el aviso de fin de carril por la pantalla LCD (ver Figura 5.20) y, si está conectado el Hyperterminal, lo mostrará en el ordenador.

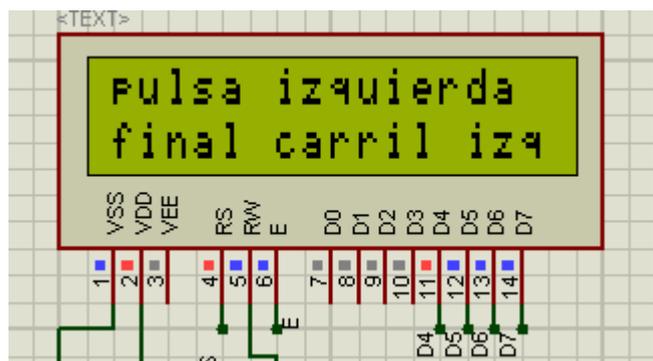


Figura 5.20. Pantalla LCD. Mensaje “Pulsa izquierda. Final de carril izquierdo”

En caso de apretarse el Pulsador 3, ocurre lo mismo que en el caso anterior pero en sentido contrario, moviéndose por tanto a la derecha, como se aprecia en el diagrama de flujo de la Figura 5.21. De nuevo, se observa la similitud entre los diagramas de flujo de los Pulsadores 2 y 3.

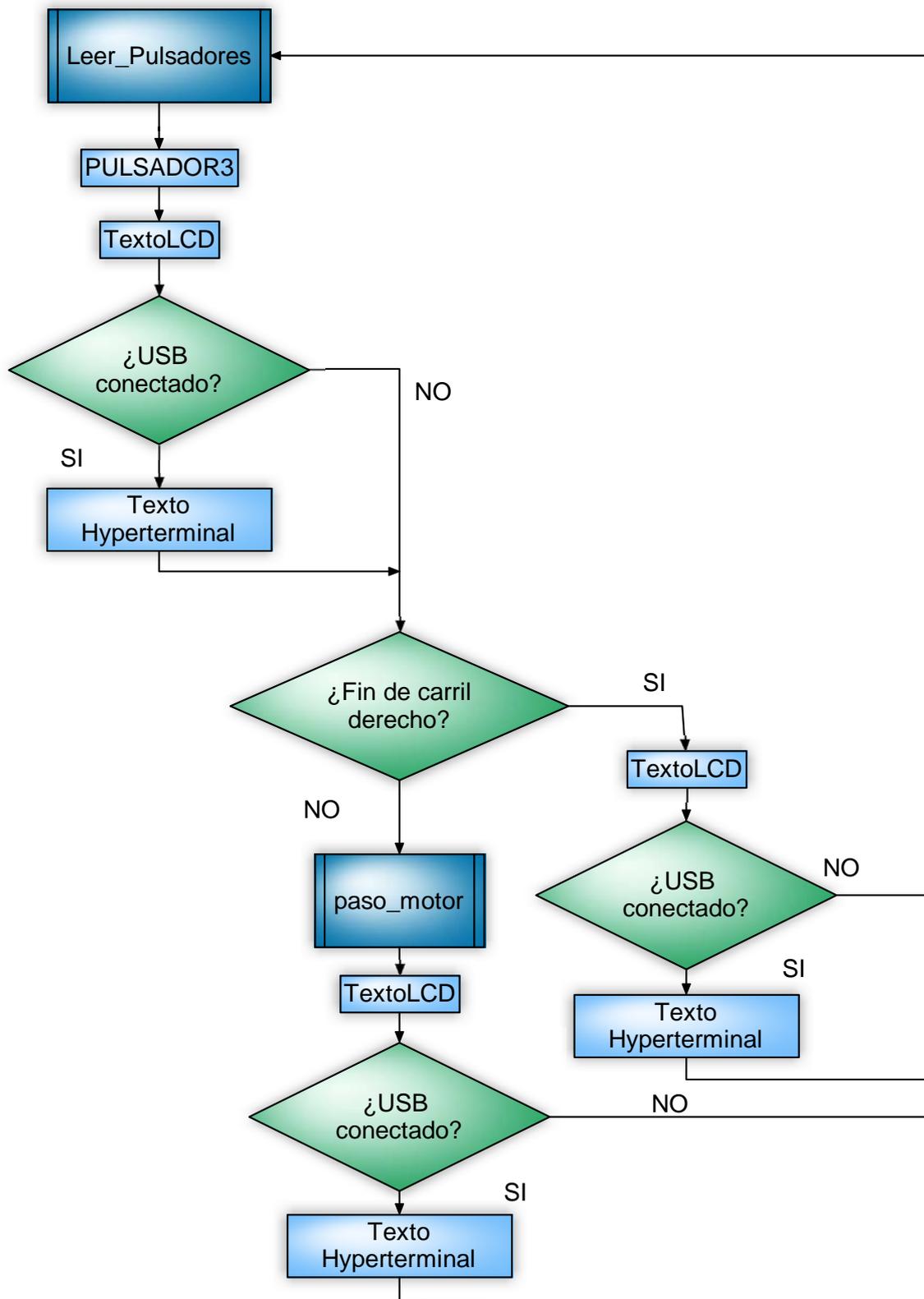


Figura 5.21. Diagrama de flujo del PULSADOR3

Finalmente, para acabar con los pulsadores, se muestra a continuación en la Figura 5.22 el diagrama de bloques del Pulsador 5. Se recuerda que se ha fijado como referencia el extremo izquierdo, que es la posición 0 y, por tanto, el extremo derecho es la posición 5190. Además “aux2” es la posición actual y “pos” es la posición destino a donde se moverá la vagoneta.

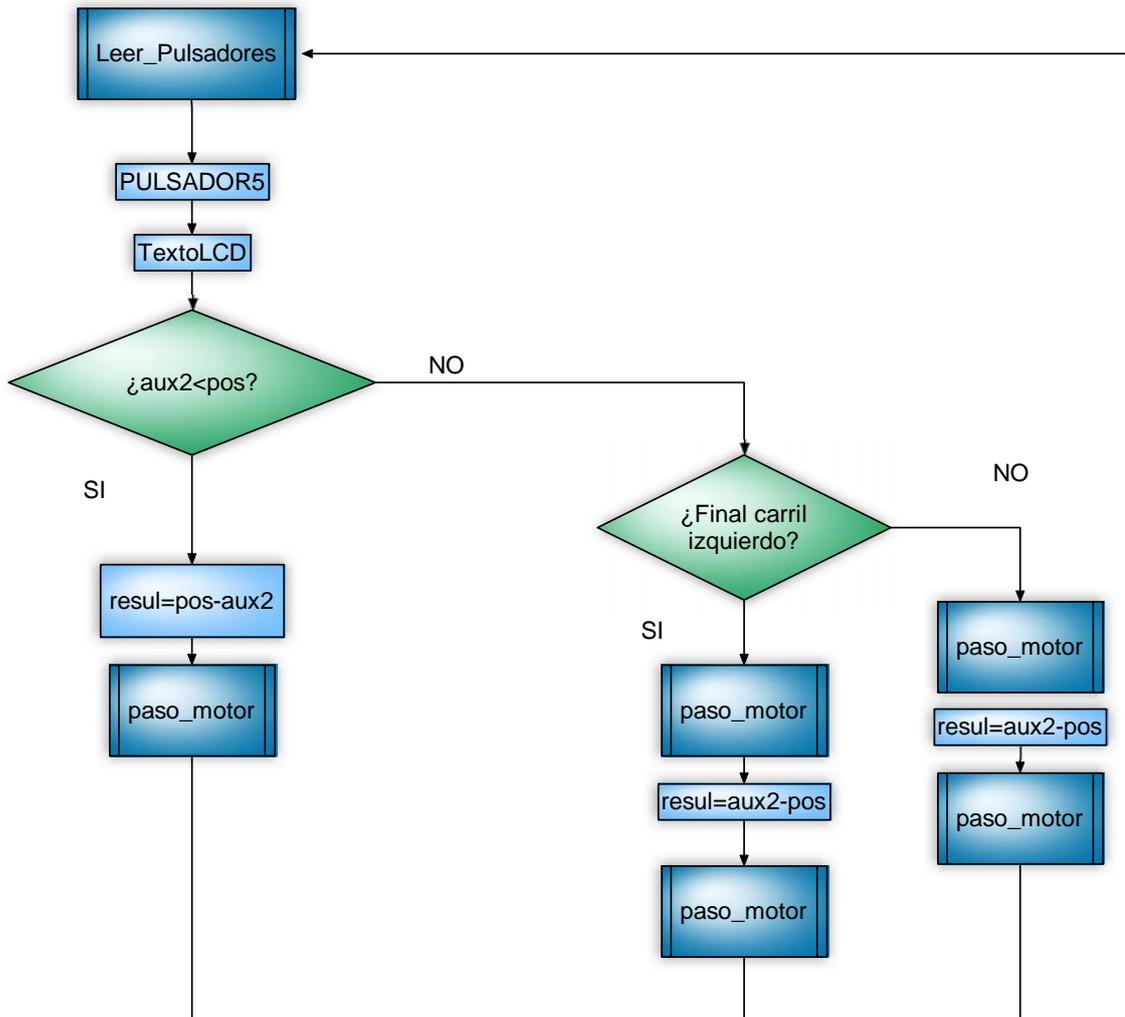


Figura 5.22. Diagrama de flujo del PULSADOR5

Como ya se explicó anteriormente, mediante los Pulsadores 1 y 4 se ha incrementado o decrementado la variable *pos*, pero sólo visualmente en la pantalla LCD sin mover el motor. Será al apretar el Pulsador 5, cuando se pondrá en marcha el motor y desplazará la vagoneta hasta la posición fijada por el usuario.

A continuación se explica su diagrama de flujo. Primeramente, tomará el valor de “pos” cargado por el usuario, lo comparará con la posición actual, y calculará cuál es mayor de los dos. En caso de que “pos” sea mayor que “aux2”, esto significa que el desplazamiento es a la derecha y en caso contrario, a la izquierda. Si es un avance a derechas, calcula la diferencia desde donde está hasta la posición fijada por el usuario, y avanza hacia la derecha. Los movimientos hacia la izquierda son análogos, pero invirtiendo el sentido.

Conviene advertir aquí que, realizando las pruebas experimentales en el laboratorio, se observó que cada vez que se programaba un cambio de sentido en el movimiento, se producía un destensamiento de la correa accionada por el motor. El destensado de la correa se traducía en giros del motor para volver a tensar la correa pero sin producirse avance físico de la vagoneta. En el Capítulo siguiente se explican con detalle las comprobaciones realizadas y las medidas correctoras adoptadas para solucionar este problema. De forma resumida, la solución adoptada ha consistido en dejar siempre la correa tensada desde la derecha y añadir, para desplazamientos del motor hacia la izquierda, la siguiente secuencia de acciones: “80 pasos en el sentido del movimiento” + “distancia hasta posición fijada” + “80 pasos en el sentido contrario al movimiento”. Con esta forma de actuar siempre se está dejando tensada la correa desde la derecha.

En la Figura 5.23 y siguientes se puede seguir la secuencia de movimientos a realizar.

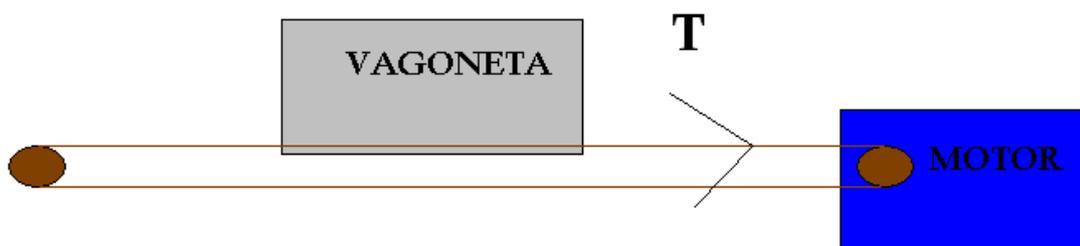


Figura 5.23. Situación inicial de desplazamiento a izquierdas

Inicialmente, se puede suponer que la vagoneta está situada en la posición 2500, que es aproximadamente la mitad del carril. Se puede observar que se parte con la correa tensada desde el extremo derecho y, suponiendo que se quiere ir a la posición 2000, hay que retroceder 500 posiciones más el ajuste por destensado de la correa.

Lo primero que se hace, es un desplazamiento de 80 posiciones en el sentido en que se quiere avanzar. Como ya se ha comentado, esas 80 posiciones no constituyen en realidad un desplazamiento de esa magnitud, ya que la correa no está tensada en esa dirección. Una vez realizado ese desplazamiento hacia la izquierda, se produce el desplazamiento, esta vez real, de 500 pasos (ver Figura 5.24).

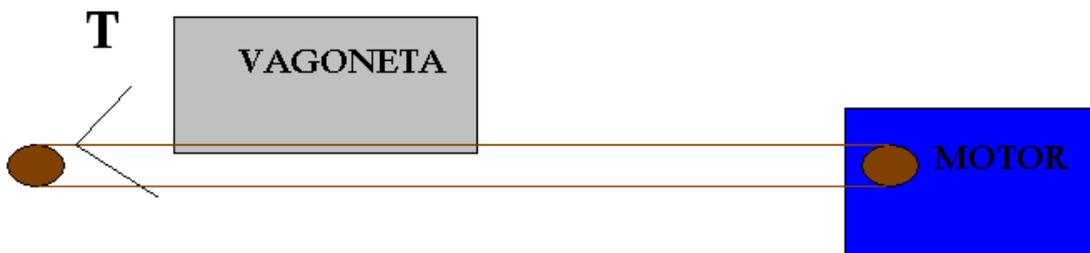


Figura 5.24. Situación intermedia de desplazamiento a izquierdas

Como se puede observar, la correa está tensada hacia la izquierda. En ese momento, se realiza un movimiento de 80 pasos en sentido contrario al anterior y con él se contrarrestan los 80 pasos que se dieron inicialmente, gastando también en este caso una parte de ellos en tensar hacia la derecha, con lo que finalmente se consigue un desplazamiento total de 500 pasos y que la correa quede tensada desde el lado derecho, como se encontraba al principio (ver Figura 5.25).

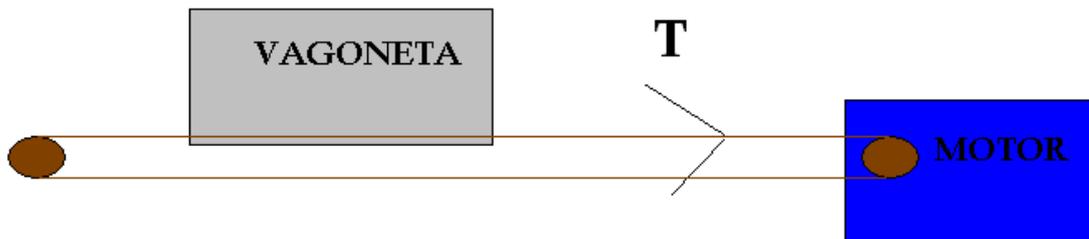


Figura 5.25. Situación final de desplazamiento a izquierdas

La decisión de considerar incrementos de 80 pasos es el resultado de adoptar la solución óptima de compromiso para resolver los problemas debidos al destensado de la correa sin añadir movimientos excesivamente grandes. En el Capítulo siguiente se comparan en detalle los resultados obtenidos con diversos incrementos (número de pasos), llegando a la conclusión de que 80 pasos proporcionan resultados suficientemente satisfactorios y no constituyen un desplazamiento adicional demasiado grande.

Por tanto, para hacer un desplazamiento a izquierdas, el desplazamiento completo consiste en desplazarse 80 pasos a la izquierda + distancia deseada hacia la izquierda + 80 pasos a la derecha.

La corrección indicada vale para cualquier desplazamiento de la vagoneta por el carril, excepto cuando la vagoneta está situada muy cerca del extremo izquierdo. Por ejemplo, si la vagoneta estuviera situada en la posición 80 y se la quisiera desplazar 1 paso hacia la izquierda, a la posición 79, no sería posible un primer desplazamiento de 81 posiciones hacia la izquierda desde la posición 80. En este caso, en el que la posición de destino final es inferior a 80, la vagoneta se desplazará hasta el extremo izquierdo y, cuando se tope con el final del carril, cambiará de sentido y avanzará en sentido contrario hasta la posición definitiva fijada por el usuario.

Para desplazamientos a la derecha no es necesario ningún tipo de corrección, ya que la correa siempre está tensada en esa dirección.

5.4.2 Diagrama de flujo de la función “CONFIGURACION”

En la Figura 5.26 se representa el diagrama de flujo de la función “Configuración”. En esta función se establece la inicialización todos los puertos, temporizadores, contadores, etc. Se han asignado los pines 1, 2, 5, 6 y 7 del puerto A como entrada, siendo el resto de pines, de salida. El pin 1 corresponde al MCLR, el pin 2 a los pulsadores PUL y el pin 5 al USB. Los pines 6 y 7 corresponden a los fines de carrera adosados al carril y, en caso de pulsarse, avisarán al PIC de que el movimiento del

motor ha llegado a su fin. El resto de pines del puerto A es de salida, lo que también ocurre con todos los pines de los puertos B y C.

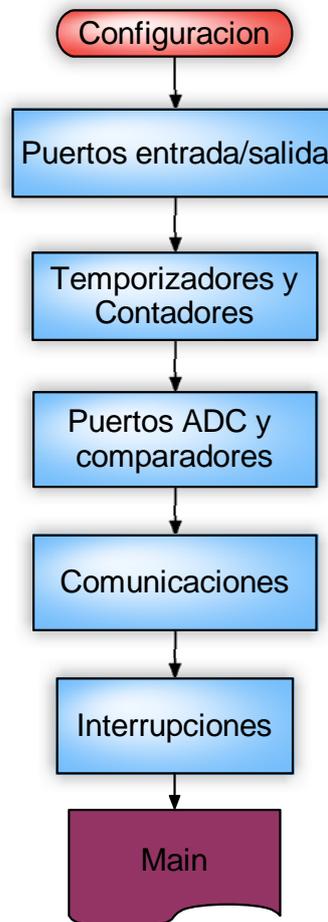


Figura 5.26. Diagrama de flujo de la función CONFIGURACION

El pin 2 del microcontrolador, al cual llamamos PUL, se habilita como convertor ADC. Recepciona datos como canal analógico y se le aplica un retardo en la conversión analógico-digital para que dé tiempo suficiente a leer el dato PUL, convirtiendo este dato en digital.

Los temporizadores y contadores se deshabilitan, ya que no son necesarios. Se deshabilitan las comunicaciones SPI, ya que no se va a utilizar otro PIC que se comunique con el existente como maestro-esclavo. Se deshabilitan las interrupciones globales, ya que no hace falta que el microcontrolador atienda ninguna interrupción.

5.4.3 Diagrama de flujo de la función “TEST LCD”

En la Figura 5.27 se muestra el diagrama de flujo de la función Test LCD.

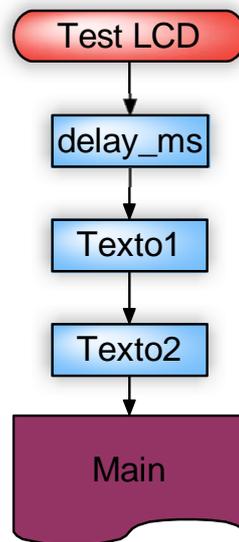


Figura 5.27. Diagrama de flujo de la función TEST LCD

Esta función se encarga de poner dos textos en la pantalla LCD en la inicialización del programa, sacando cada uno de ellos en una línea de la LCD (ver Figura 5.28). Como se comentó en el capítulo anterior, la LCD es del tipo 16x2 (16 columnas y 2 filas). Por tanto, en la inicialización se comprueba que funcionan las 2 filas. Se utiliza un retardo “delay_ms”, ya que se necesita un tiempo de configuración antes de poder sacar un texto al inicializar.

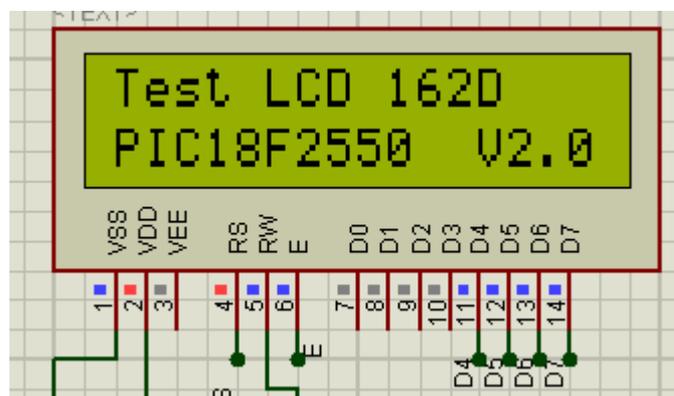


Figura 5.28. Pantalla LCD. Mensaje de inicialización y test de funcionamiento

5.4.4 Diagrama de flujo de la función “POSICIÓN_ACTUAL”

En la Figura 5.29 se muestra el diagrama de flujo de la función “Posición_actual”.

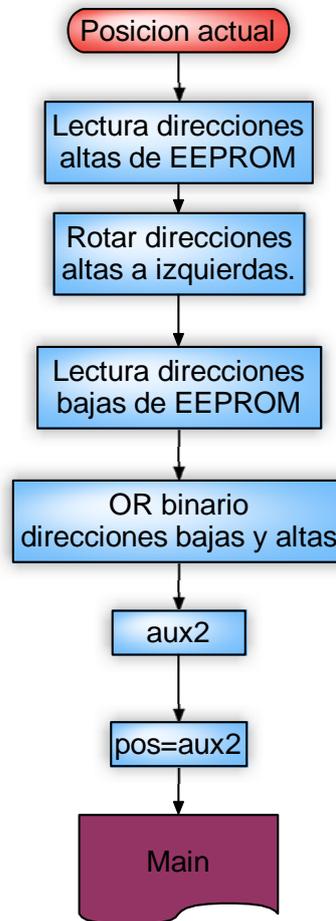


Figura 5.29. Diagrama de flujo de la función POSICION_ACTUAL

Esta función es la encargada de leer en la memoria EEPROM el dato de la posición actual, que consta de 4 cifras en sistema decimal. Ese número que indica la posición actual se descompone en dos números de dos cifras, a los que se identifica como dirección alta y dirección baja.

Primeramente se leen, con una variable auxiliar, las direcciones altas y, en otra variable auxiliar de doble tamaño que la anterior, se coloca este valor en las posiciones altas utilizando para ello la instrucción *rotar*. Utilizando de nuevo la primera variable auxiliar, se leen las direcciones bajas y se guardan en las posiciones bajas utilizando una variable de doble tamaño. Mediante un OR binario, también conocido como “suma

lógica”, se guarda la posición completa, con las direcciones altas y bajas, en una variable auxiliar de doble tamaño.

La tabla correspondiente a OR binario es:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

Resumiendo, el resultado arrojado será siempre 1 si, al menos, una de las variables tiene por valor 1.

Si, por ejemplo, la posición actual es la 2483, este número se descompone en 24 y 83 que, convertidos a sistema binario, son:

$$24 \rightarrow 00011000 \qquad 83 \rightarrow 01010011$$

Trasladando la dirección alta a la variable auxiliar de doble tamaño resulta:

XXXXXXXX00011000

y al rotar este valor a las direcciones altas, se obtiene:

00011000XXXXXXXX.

Si ahora se suman las bajas, que son:

XXXXXXXX01010011

se obtiene:

0001100001010011.

De esta manera, se puede saber en qué posición se está en cada momento.

5.4.5 Diagrama de flujo de la función “LEER_PULSADORES”

En la Figura 5.30 se muestra el diagrama de flujo de la función “Leer_pulsadores”.

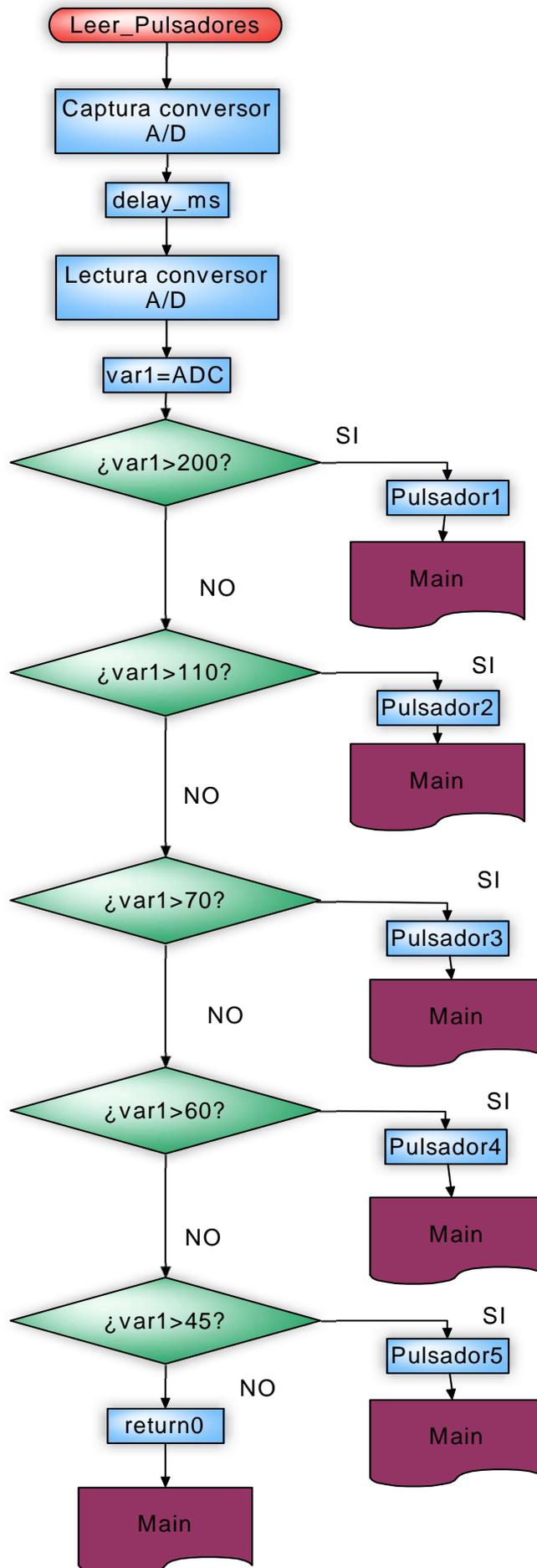


Figura 5.30. Diagrama de flujo de la función LEER_PULSADORES

Esta función “Leer_pulsadores”, toma las diferentes tensiones que se obtienen del conversor analógico-digital. Como ya se explicó en el Capítulo anterior, dependiendo del pulsador que detecte, tendrá una tensión asociada y por tanto un valor ADC.

En la Tabla 5.1 se muestra un resumen con los valores correspondientes:

Pulsador	Tensión	Valor ADC
1	4,55	232
2	2,38	121
3	1,61	82
4	1,22	62
5	0,98	49

Tabla 5.1. Relación Pulsadores / Tensión / Valor ADC

Cabe destacar que la tensión de salida se guarda en una variable de 8 bits pudiéndose representar hasta $2^8 = 256$ niveles diferentes. Si la tensión máxima son 5V, 0V se corresponderían con el nivel ADC 0 y 5V con el nivel ADC 255. Debido a que es posible que haya fluctuaciones en la tensión, se han fijado valores ADC más bajos que los de la Tabla 5.1 en el código para garantizar que, en caso de producirse fluctuaciones, se detecte siempre el pulsador presionado.

Si, por ejemplo, se presiona el Pulsador 1, el programa captura el valor del dato ADC, que es 232. Como es posible que se produzcan fluctuaciones, puede resultar que el valor capturado sea mayor o menor que 232. Al indicar por código que la captura debe ser superior a 200, existe la seguridad de haber capturado correctamente el Pulsador 1.

Es necesario introducir un retardo entre la captura y lectura del ADC para que le dé tiempo al microcontrolador a hacer la conversión correctamente. Una vez capturado ya puede leerlo, lo guarda en la variable auxiliar *var1* y, a partir de ahí, comprueba si coincide con alguno de los valores asignados.

Si coincide con alguno de los valores asignados, sabrá qué pulsador ha sido pulsado y le dará el aviso a la función "Main". Si corresponde al Pulsador 1 (arriba), al Pulsador 2 (izquierda), al Pulsador 3 (derecha), al Pulsador 4 (abajo) ó al Pulsador 5 (intro), la función "Leer_pulsadores" devolverá al "Main" un 1, un 2, un 3, un 4 ó un 5, respectivamente.

En caso de que no coincida con ninguno de los valores fijados, le devolverá un 0.

5.4.6 Diagrama de flujo de la función "PASO_MOTOR"

En la Figura 5.31 se muestra el diagrama de flujo de la función "Paso_motor"

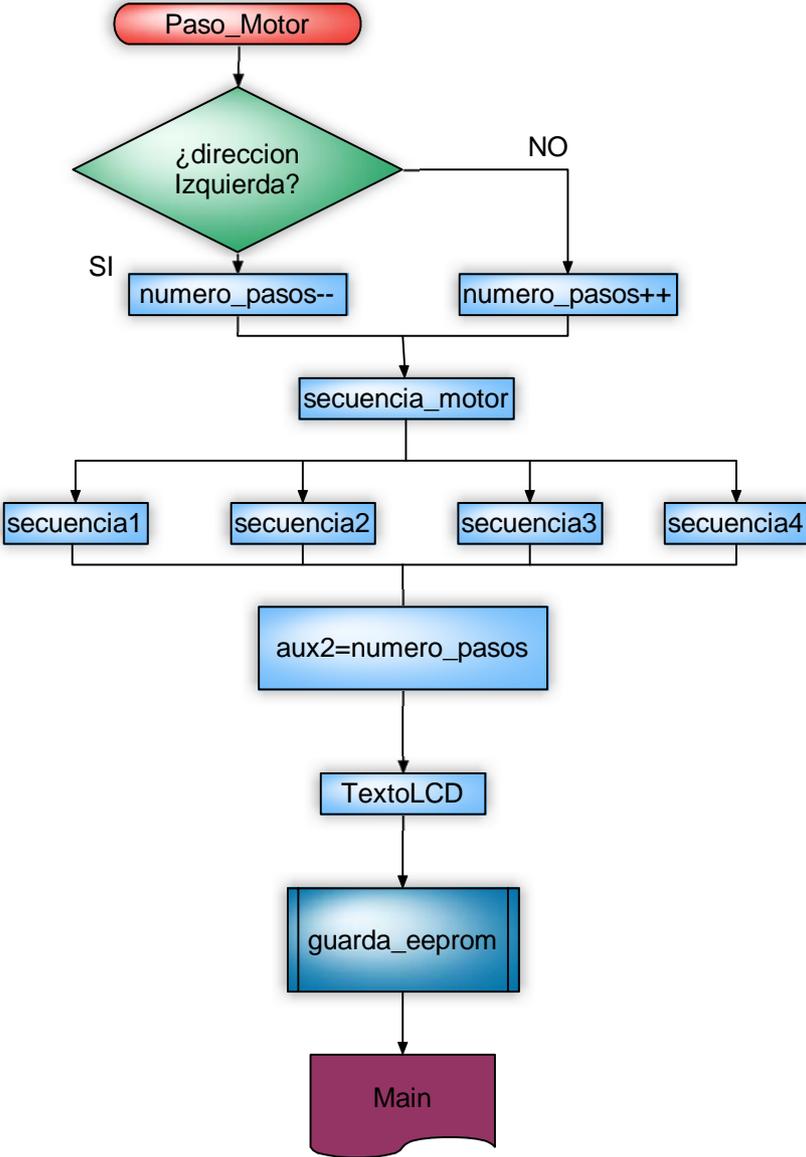


Figura 5.31. Diagrama de flujo de la función PASO_MOTOR

Lo primero que tiene que realizar esta función es comprobar si la dirección de movimiento es a derechas o a izquierdas. Si es a izquierdas, disminuirá el número de pasos, mientras que si es a derechas aumentará el número de pasos.

Como ya se justificó en el Capítulo 3, se ha escogido la secuencia “normal giro”.

En la Tabla 5.2 se muestra de nuevo la secuencia.

PASO	TERMINALES			
	A	B	C	D
1	ON	ON	OFF	OFF
2	OFF	ON	ON	OFF
3	OFF	OFF	ON	ON
4	ON	OFF	OFF	ON

Tabla 5.2. Secuencia de control de giro

Se observa que el control del motor paso a paso unipolar se consigue mediante 4 secuencias que irán activando únicamente 2 bobinas a la vez.

En la Tabla 5.3 se muestra un esquema con los pines utilizados para controlar los terminales del motor.

Bobina	Variables	Pines PIC
A	A1	18
B	A2	17
C	B1	13
D	B2	12

Tabla 5.3. Relación Bobinas/Variables/Pines PIC

Si se quisiera poner en marcha el motor, por ejemplo con la configuración del primer paso de la secuencia (secuencia1), se han de poner a nivel alto las pines 18 y 17 y a nivel bajo las pines 13 y 12. Una vez hecha esta aclaración, el motor sabrá que si quiere avanzar tendrá que empezar desde el primer paso de la secuencia (secuencia1), continuar con el segundo paso (secuencia2) y así sucesivamente, y al llegar al cuarto paso (secuencia4) volver a empezar por el primer paso (secuencia1). Lo mismo ocurre, pero en sentido contrario, si se quiere retroceder: habrá que ir configurando las bobinas del motor siguiendo una secuencia descendente y cíclica en los pasos a seguir hasta llegar al primer paso(secuencia1) y empezar de nuevo en el último(secuencia4). Cada vez que haga una secuencia de giro lo grabará en la función “Guarda_eeprom”, que será explicada a continuación junto con su diagrama de flujo. Una vez guardada la posición, retornará al programa principal “Main”.

5.4.7 Diagrama de flujo de la función “GUARDA_EEPROM”

En la Figura 5.32 se muestra el diagrama de flujo de la función “Guarda_eeprom”.

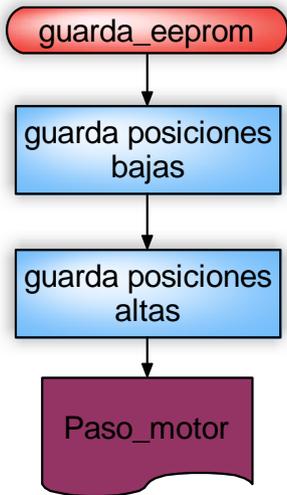


Figura 5.32. Diagrama de flujo de la función GUARDA_EEPROM

Todas las veces que se mueva el motor ha de grabar su posición en la memoria EEPROM. Con esto, el PIC18F2550 indicará en qué posición está y así se evitará que el motor se ponga en marcha cuando haya llegado al final del carril. También permite controlar los movimientos y que, una vez apagado y puesto en marcha otra vez, siga en la misma posición en la que estaba cuando fue apagado.

Esta función, al contrario que las demás, es llamada desde otra función que no es el programa principal "Main". Primeramente guardará las posiciones altas; después, guardará las bajas y volverá a la función "Paso_motor" para que continúe ejecutándose.

5.4.8 Diagrama de flujo de la función "POSICION_IZQUIERDA"

Finalmente, en este último Apartado se describe el flujo de programa de la función "Posición_izquierda".

En la Figura 5.33 se muestra el diagrama de flujo.

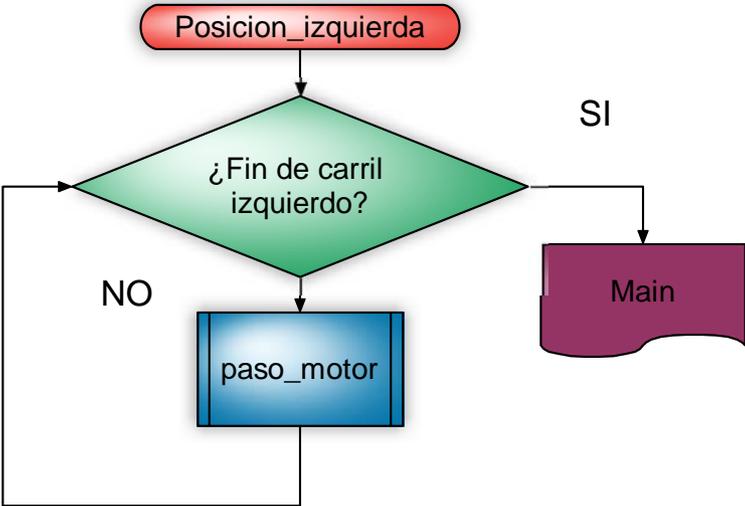


Figura 5.33. Diagrama de flujo de la función POSICION_IZQUIERDA

Esta función se encarga de desplazar la vagoneta hasta el extremo izquierdo. Hasta que no llegue al final del carril no se detendrá; cuando se active el detector de fin de carril, se detendrá y volverá al "Main" para continuar con el resto de instrucciones.

CAPÍTULO 6

RESULTADOS EXPERIMENTALES

6.1 INTRODUCCIÓN

Este Capítulo, que se ocupa de los resultados experimentales, constituye un aporte esencial al desarrollo del Proyecto desde dos puntos de vista diferentes. Por una parte, permitirá establecer las condiciones de calibración necesarias para un uso satisfactorio del sistema diseñado. Por otra, se verificará el correcto funcionamiento del programa, en particular, en los aspectos relacionados con la inicialización del sistema y la conexión a puertos.

Superada la fase de pruebas y mediciones sobre el sistema, los resultados experimentales permitirán hacer una valoración de los objetivos conseguidos en el diseño del posicionador micrométrico en un eje y plantear las posibles mejoras a introducir.

En la primera parte del Capítulo se describe el montaje utilizado para la calibración del sistema. Está basado en varios componentes que forman parte de un montaje de caracterización de dispositivos electroópticos que se utiliza en el Laboratorio del Grupo de Displays y Aplicaciones Fotónicas de la Universidad Carlos III de Madrid. En los sucesivos Apartados se describen las pruebas realizadas y los resultados y conclusiones obtenidas. Para completar el trabajo, se ha considerado interesante incluir en un Apartado una sencilla guía de utilización del prototipo implementado, para facilitar su manejo a cualquier usuario del sistema.

6.2 MONTAJE DE CALIBRACIÓN DEL SISTEMA

Para la realización de las pruebas experimentales del sistema se han instalado un conjunto de elementos sobre una mesa óptica. (ver Figura 6.1).



Figura 6.1. Mesa óptica

La mesa óptica posee la propiedad de absorber las vibraciones procedentes del entorno en el que se ubica, ya sean procedentes de otros instrumentos de medida o las propias de la circulación del personal por el recinto. Para llevar a cabo esta función, se encuentra instalado un compresor anexo a la mesa óptica que le permite recuperar la posición horizontal nivelada tras haberla perdido. Su superficie está llena de orificios roscados (métrica M6, en este caso) sobre los que se instalan los diferentes componentes. Se han ubicado un láser, filtros neutros, una regla micrométrica, un objetivo óptico y una cámara CCD. A continuación se explican con detalle cada uno de los componentes que forman parte del montaje.

6.2.1 Emisor láser

Un emisor láser (*Ligth Amplification by Stimulated Emisión of Radiation*) es un emisor óptico coherente cuya luz emitida es intensa y está colimada en un haz estrecho de spot milimétrico. A diferencia de otros emisores, como pueden ser los diodos LED, los láseres emiten en una única longitud de onda, por lo que su uso está indicado en determinadas aplicaciones donde este aspecto es relevante. En el caso que nos ocupa, se ha utilizado un láser como fuente de luz para el sistema de caracterización, aprovechando su buena característica de emisión. La distancia entre el láser y el resto de componentes del sistema óptico es de 80 cm, aproximadamente. Para la calibración del sistema es suficiente con un láser de 1mW de potencia. Este bajo valor permite el trabajo de los operarios con seguridad evitando, por ejemplo, el daño que producen en los ojos los láseres de potencias superiores.

En la Figura 6.2 se muestra el láser utilizado sobre un soporte adaptado.

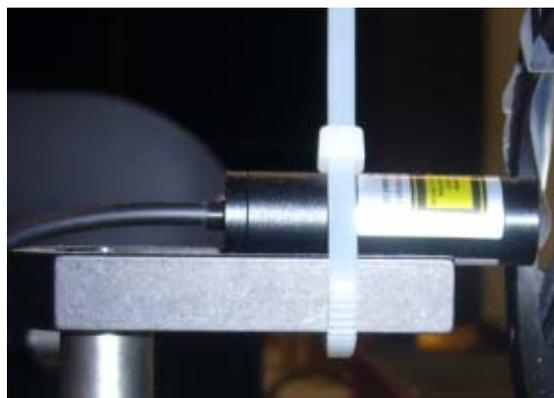


Figura 6.2. Láser en el sistema de caracterización

6.2.2 Filtros neutros

Los filtros de densidad neutra (ND) reducen la cantidad de luz que pasa a través de la lente. Están fabricados con vidrio óptico de gran calidad que filtran todo el espectro visible, permitiendo la reducción de la intensidad de la luz sin que se altere el color o el contraste. En nuestro caso se disponen dos filtros, uno ND16 y otro ND4. El ND16 deja pasar el 6.2% de la intensidad lumínica y el filtro ND4 deja pasar el 25% de ese 6.2%; por tanto, sólo deja pasar el 1,55% de intensidad lumínica generada por el láser. Con esta intensidad lumínica se consigue que no se sature la cámara CCD.

En la Figura 6.3 se pueden observar los dos filtros neutros, colocados muy cercanos de la salida del láser.

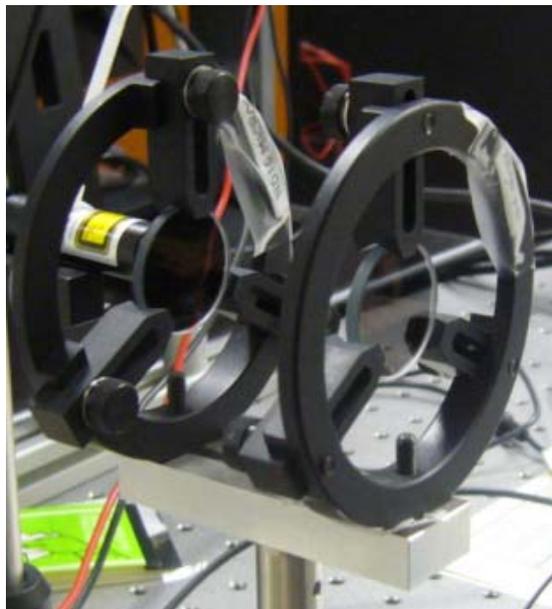


Figura 6.3. Filtros neutros

6.2.3 Regla micrométrica

Una regla micrométrica es un elemento de medición cuya especificación es el tamaño total de la regla. Para la calibración del Proyecto se ha escogido una regla de 1mm, de la marca Thorlabs. La longitud de 1mm está dividida en 10 partes, siendo cada una de ellas de 100 micrómetros. Cada una de estas partes está dividida, a su vez, en 10 partes, por lo que la resolución de la regla es de 10 micrómetros. Esta regla está

grabada sobre un vidrio de dimensiones 1" x 3", como se muestra en la Figura 6.4. Este tamaño permite su colocación sobre un portaobjetos de medidas estándar.



Figura 6.4. Regla micrométrica de la marca Thorlabs

6.2.4 Objetivo óptico

El objetivo óptico se encarga de aumentar el tamaño de la imagen para su posible captura. Se ha utilizado un objetivo de 20 aumentos de la marca Thorlabs; por tanto, 1mm a la entrada del objetivo se convierte en 20mm tras el objetivo. El conjunto de lentes que forman el objetivo hacen que sea preciso colocarlo muy próximo a la muestra a enfocar. La muestra a enfocar será la regla de medida.

En la Figura 6.5 se muestra el objetivo utilizado.



Figura 6.5. Objetivo de 20 aumentos (X20) de la marca Thorlabs

6.2.5 Cámara CCD

Las cámaras CCD (*Charge-coupled device*) son cámaras de gran calidad de imagen. Las imágenes tomadas se representan en la pantalla y su calidad depende del número de píxeles. A mayor número de píxeles, mayor calidad de imagen o, lo que es lo mismo, mayor resolución.

En el Laboratorio se dispone de una cámara CCD C4742, de la marca Hamamatsu. Entre sus características se destaca una resolución de 1,37 millones de píxeles para la visualización de imágenes (ver Figura 6.6.).



Figura 6.6. Cámara CCD de la marca Hamamatsu

La cámara no dispone de ningún *display* en el que poder visualizar las imágenes, por lo que siempre ha de trabajar a través de un ordenador y visualizar las imágenes en la pantalla del mismo. La cámara dispone de un *software* de control para la captura de imágenes y ofrece la posibilidad de poder hacer medidas de distancias sobre las imágenes. El detalle del procedimiento utilizado para la toma de medidas será explicado más adelante.

6.2.6 Montaje de calibración completo

En la Figura 6.7 se muestra el montaje final del conjunto sobre la mesa óptica: subsistema óptico, parte mecánica y subsistema electrónico.

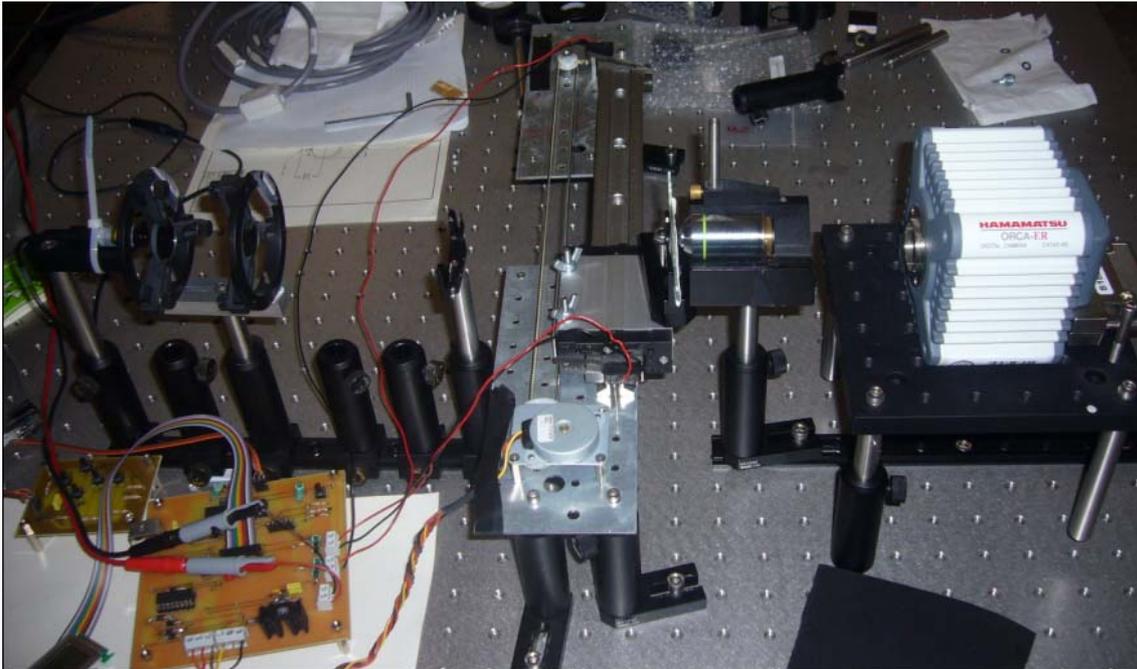


Figura 6.7. Sistema microposicionador insertado en el montaje de caracterización óptico

En el subsistema óptico se observa que, en primer lugar, está colocado el láser; a continuación, se encuentran los dos filtros neutros y, después, el posicionador micrométrico. Sobre la vagoneta se ha colocado la regla micrométrica y junto a ella se encuentra el objetivo del microscopio. En el extremo opuesto del montaje se ubica la cámara CCD.

También se puede apreciar en la imagen que el carril del microposicionador se ha dispuesto en la dirección perpendicular al circuito óptico. Se ha optado por disponer su eje en esta posición para poder medir y calibrar el desplazamiento de la vagoneta. Se ha usado como referencia la regla micrométrica y se ha empleado el software de captura de imágenes de la cámara CCD, HCLImage de Hamamatsu, para medir las distancias. La posición de origen del carril se define a la izquierda del mismo, observándolo desde la posición del láser. La posición final del carril es el extremo derecho, y se llega a ella tras haber recorrido el motor 5200 pasos.

En la parte inferior izquierda de la imagen, se pueden ver también los siguientes elementos del subsistema eléctrico: la placa de pulsadores, la placa de control sobre el motor y la pantalla LCD.

6.3 GUÍA DE FUNCIONAMIENTO DEL POSICIONADOR

El sistema electrónico de control del prototipo del microposicionador en un eje implementado se encuentra insertado en una única caja. En el panel frontal de la misma se ubican la pantalla LCD y 5 pulsadores de color negro (ver Figura 6.8.). Además, hay un orificio pintado de color gris que se corresponde con el circuito de RESET del microcontrolador. Cada vez que se inserte un objeto de punta fina o un destornillador fino en el orificio de RESET, se activa el botón de RESET del microcontrolador.

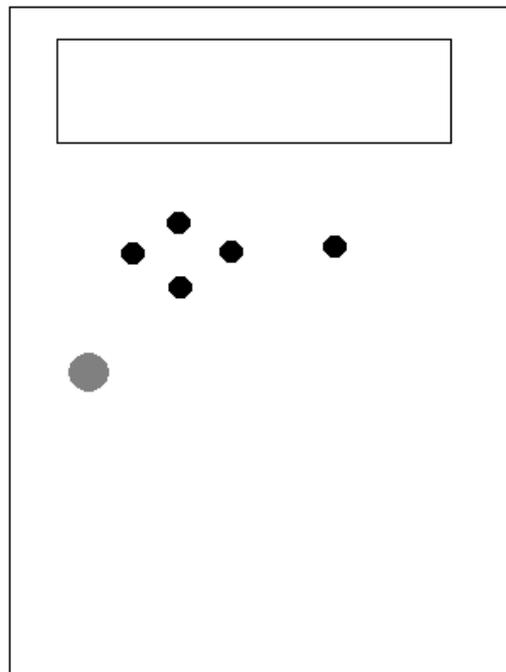


Figura 6.8. Vista en planta del prototipo implementado

Cada uno de los pulsadores realiza una función diferente que se describe a continuación. Los 4 pulsadores más próximos entre sí sirven para realizar desplazamientos de la vagoneta sobre el carril, largos o cortos, tanto a derecha como a

izquierda (ver Figura 6.9.). El pulsador que se encuentra separado del resto es el botón de INTRO.

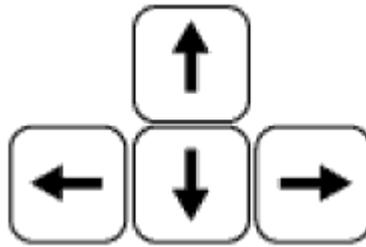


Figura 6.9. Esquema de los pulsadores del prototipo implementado

La función de los pulsadores con flechas *derecha* e *izquierda* es la siguiente: cada vez que se pulsan , se produce el movimiento de un paso del motor en la dirección que indica la flecha. En este modo de funcionamiento, la velocidad de ejecución es baja en cada movimiento..

La función de los pulsadores con flechas *arriba* y *abajo* difiere del caso anterior en que está pensada para grandes desplazamientos. Con el botón con flecha-*arriba*, se puede fijar una posición mayor que la ocupada actualmente; permitiendo, por tanto, desplazamientos grandes hacia la derecha. El botón con flecha-*abajo* es el inverso que el de *arriba*, estando indicado, por tanto, para desplazamientos grandes hacia la izquierda. Estos desplazamientos no se producirán hasta que se pulse el botón INTRO (a diferencia de los pulsadores *izquierda* y *derecha*) que son obedecidos en el momento en que se pulsan.

Por ejemplo, si la vagoneta se encuentra en la posición 1000 y se quiere despalzar hasta la posición 2000, se ha de pulsar el botón de *arriba*. En caso de que se produzca un error y se fije la posición 2003, se tiene la posibilidad de, o bien ir a esa posición y descontar después 3 posiciones, o bien descontar previamente esas 3 posiciones tocando el botón de *abajo* y pulsar después el botón de INTRO para desplazar la vagoneta.

Si al inicializar se deja pulsado el botón de INTRO, el sistema se pondrá en marcha buscando el extremo izquierdo y, una vez situado allí, avanzará hacia la derecha hasta la posición 2500 (aproximadamente en el centro del carril).

En la Figura 6.10, se muestra la vista lateral del prototipo.

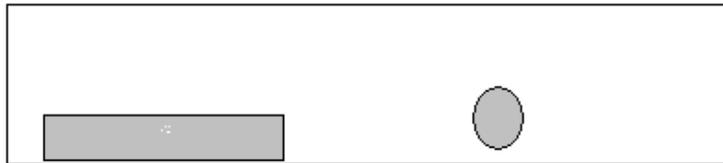


Figura 6.10. Vista lateral del prototipo implementado

En este lado, el prototipo tiene dos orificios, el rectangular donde se insertan los cables del motor y el circular, que corresponde a la alimentación del sistema. Por el orificio de la alimentación se accede a dos cables, uno rojo y otro negro: rojo, para la señal de alimentación y negro, para la tierra. Para el correcto funcionamiento de todo el circuito es necesario que se alimente con una tensión de +8V.

En la Figura 6.11 se muestra el alzado principal del prototipo, donde se aloja un cable USB para comunicación con el PC. Cada vez que se conecte el cable USB al microcontrolador, éste lo detectará y mostrará mensajes, tanto en la LCD como en la pantalla del PC.

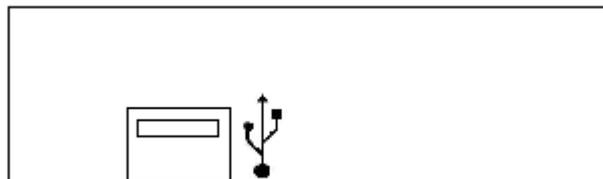


Figura 6.11. Alzado principal del prototipo

Finalmente, en la Figura 6.12 se presenta el prototipo del microposicionador en un eje en su versión final.

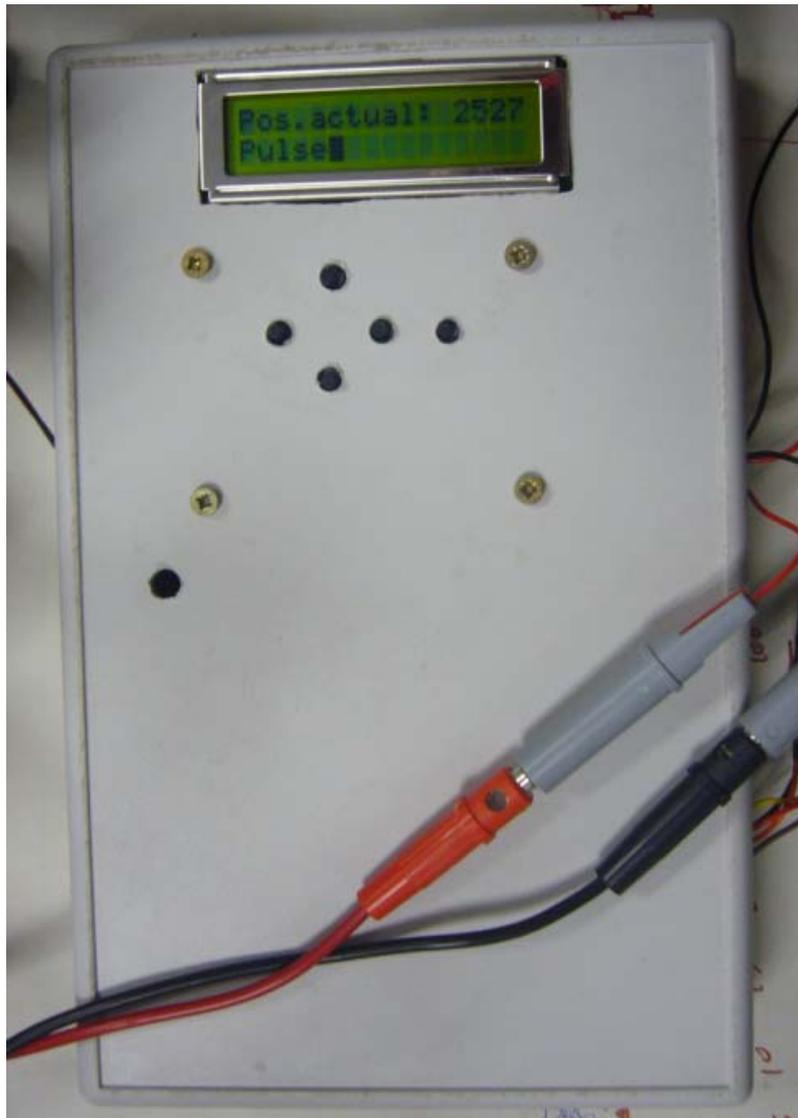


Figura 6.12. Prototipo final del sistema microposicionador en un eje

6.4 CALIBRACIÓN Y MEDIDA DE ESPECIFICACIONES

Una vez enumerados los componentes del montaje de calibración completo, se explicará detalladamente la forma de realizar las capturas de las imágenes y los pasos seguidos para la calibración. Después de abierto el programa de captura HCIImage, la captura de imágenes comienza seleccionado la pestaña *Live*. Una vez que se visualiza en el ordenador lo que está capturando la cámara, es preciso alinear el sistema y enfocar la regla con el objetivo de microscopio de manera que se visualice nivelada y permita realizar medidas de distancias a partir de ella.

6.4.1 Procedimiento de medida

Como ya se ha indicado, entre dos cifras consecutivas de la regla, por ejemplo 0 y 1, hay 10 marcas transversales, cada una de ellas de 10 micrómetros. Hay, por tanto, 100 micrómetros entre esos dos valores. Una vez enfocada la regla micrométrica, y con ayuda del programa de la CCD se puede conocer la equivalencia entre 100 micrómetros y la misma distancia expresada en número de píxeles. Para ello se hace uso de “la regla del programa” definiendo su tamaño en función de la prueba a realizar. Se debe mantener la misma regla mientras no cambien las condiciones de medida.

En la Figura 6.13 se muestran simultáneamente las dos reglas: la imagen de la regla micrométrica de Thorlabs y “la regla del programa”, que mide en píxeles.

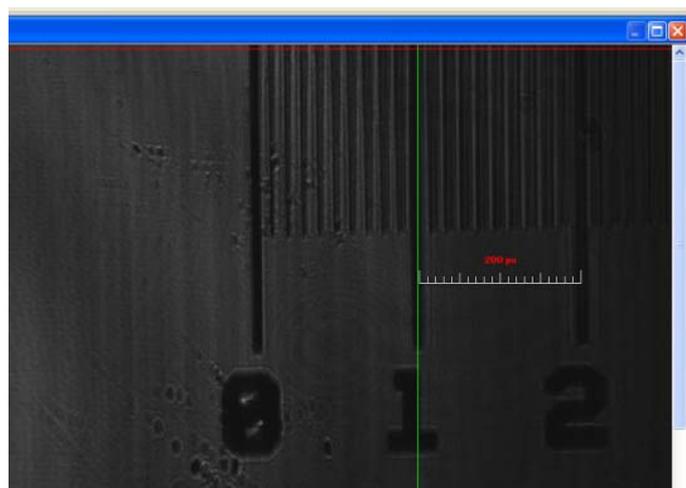


Figura 6.13. Reglaje óptico

En el ejemplo de la figura, se puede apreciar que 100 micrómetros de la regla equivalen a 200 píxeles. En las pruebas realizadas, se ha trabajado con tamaños de “la regla del programa” próximos a 200 píxeles. Este valor se modifica ligeramente con el ajuste de la posición de los componentes en cada medida, dependiendo fundamentalmente de la posición de la cámara respecto del objetivo.

6.4.2 Medida de la resolución del posicionador micrométrico

La primera prueba que se ha realizado ha sido comprobar el desplazamiento que se produce en cada paso del motor, es decir, se ha medido la resolución del motor. El programa dispone de unas líneas de referencia, vertical y horizontal, que pueden servir de ayuda, pero sólo ha sido necesario utilizar la referencia vertical. Dejando la referencia vertical en una posición fija coincidente con una marca de la regla micrométrica se ha podido medir el avance del motor en cada paso.

Se ha realizando el experimento de medir la distancia que recorre la vagoneta sobre el carril en 1 paso del motor, en 2 pasos, en 3 pasos, etc. hasta 10 pasos, desde una posición fija conocida. Se ha comprobado que en recorridos sin cambio de sentido el avance medio del motor en un paso es de, aproximadamente, 40 micrómetros.

En la Figura 6.14 pueden verse algunas capturas de la cámara utilizadas para esta deducción.

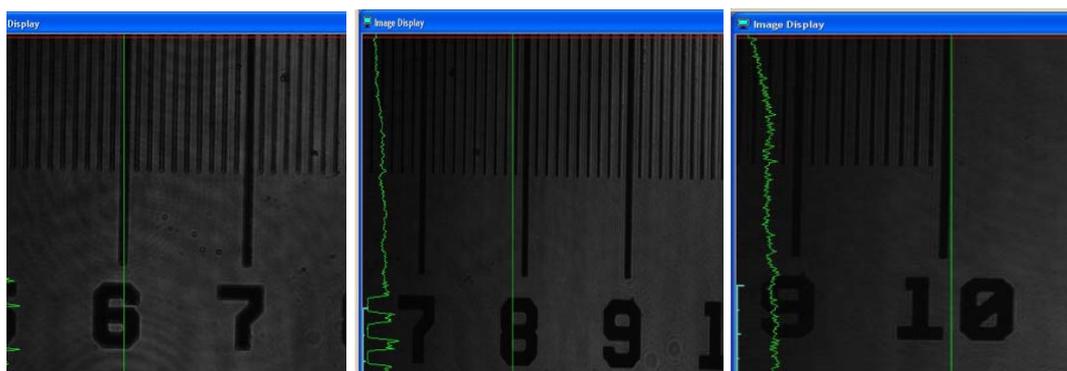


Figura 6.14. Calibración de la resolución de 1 paso del motor: Referencia / 5 pasos derecha / 10 pasos derecha

En la figura 6.14. izquierda se muestra la posición fija, 6, de partida y en las dos figuras restantes se muestra el resultado tras avanzar 5 pasos a la derecha (imagen central) y 10 pasos a la derecha (imagen derecha).

6.4.3 Calibración con corrección del destensado de la correa

En el proceso de calibración del sistema se detectó un error en el posicionamiento de la vagoneta que se producía cuando después de desplazarla en un sentido, el movimiento siguiente se hacía en sentido contrario. Se observó que la vagoneta no se mueve inmediatamente en el sentido contrario, sino que necesita tensar la correa antes de empezar el nuevo movimiento. A continuación se explican las pruebas realizadas para medir este desfase y poder rectificarlo por *software*.

Primero se describe la secuencia que se ha implementado en el control del movimiento del motor, para que se mueva un paso “de izquierda a derecha”. Este proceso es obligatorio si se quiere controlar el movimiento del motor en un sentido y hacerlo independiente del sentido de movimiento que hubiera llevado previamente. La prueba completa se realiza en tres pasos:

- i) Inicialmente, se tiene la correa tensada en el sentido de avance, “*de izquierda a derecha*”
- ii) se avanzan N pasos en sentido contrario “*de derecha a izquierda*” y,
- iii) finalmente, se avanzan N+1 pasos en el sentido inicial, “*de derecha a izquierda*”. La correa queda de nuevo tensada “*de izquierda a derecha*”.

El detalle del proceso se muestra a continuación. Al inicio de cada avance la correa está tensada en el sentido de avance. Si, por ejemplo, el sentido de avance fuera “*de izquierda a derecha*”, la correa debería estar tensada en ese sentido.

En la Figura 6.15 se muestra la posición inicial con la correa tensada.

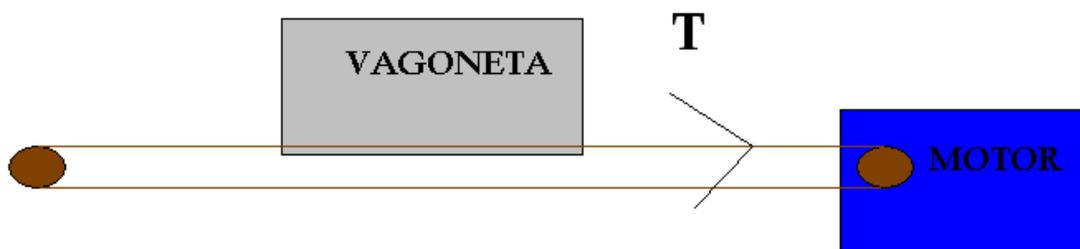


Figura 6.15. Posición inicial con la correa tensada “*de izquierda a derecha*”

En el caso de que la correa no estuviera tensada en el sentido del movimiento que se le solicita al motor, el número de pasos reales que daría al recibir la orden no sería correcto, ya que el motor consumiría algunos de dichos pasos en tensar la correa y, entre tanto, no se produciría avance de la vagoneta.

Para solucionar este problema, es preciso ordenar el avance de N pasos en sentido contrario, “*de derecha a izquierda*”, y después en sentido directo, “*de izquierda a derecha*”, para asegurar el tensado de la correa en la dirección de avance. El número de pasos, N , es un número intrascendente desde el punto de vista del avance deseado de la vagoneta y, en principio, N es desconocido.

La primera parte de estos N pasos “*de derecha a izquierda*” servirá para que se tense la correa y, una vez tensada el número restante de pasos hasta N será un remanente que provocará un avance no deseado de la vagoneta.

En la Figura 6.16 se muestra el esquema de este desplazamiento completo.

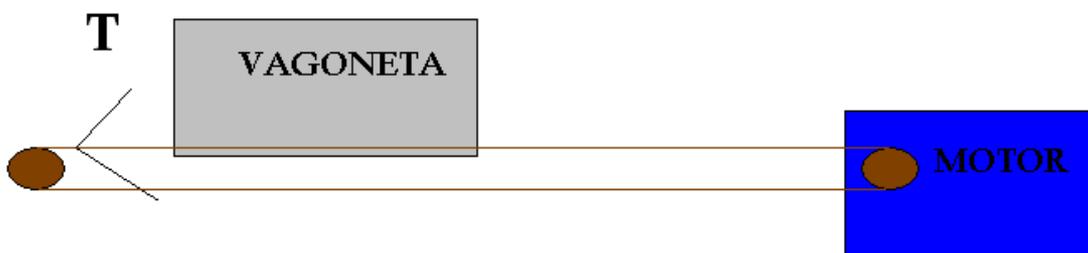


Figura 6.16. Avance de N pasos (teóricos) en sentido contrario, “*de derecha a izquierda*”

Una vez terminado este avance, la correa estará tensada “*de izquierda a derecha*”. Es ahora cuando se debe cumplir la orden de realizar un avance de $N+1$ pasos en el sentido inicial: N pasos para compensar los anteriores y 1 paso más, que es el avance real deseado para la vagoneta. Al igual que en el proceso anterior, se producirá previamente el tensado de la correa y el avance de la vagoneta un número de pasos no deseado, pero el resultado final implicará el desplazamiento de 1 paso “*de izquierda a derecha*” desde la posición inicial, quedando de nuevo la correa tensada “*de izquierda a derecha*” (ver Figura 6.17).

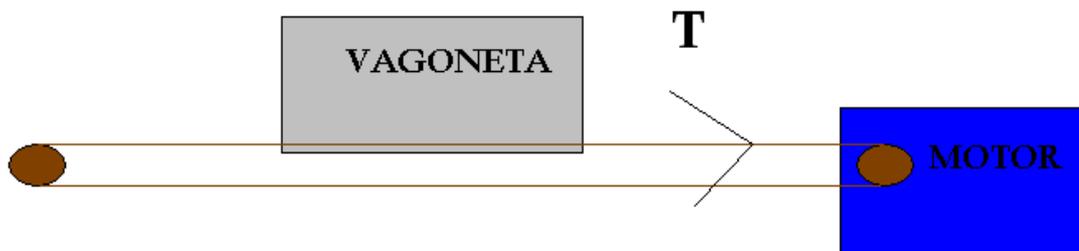


Figura 6.17. Avance de N+1 pasos (teóricos) en sentido inicial, “de izquierda a derecha”

Por tanto, si antes de iniciar el movimiento se ajusta en un punto la regla vertical del programa de la cámara (línea verde de la Figura 6.18), y al final del movimiento se comprueba dónde ha quedado la regla vertical, la diferencia será el desplazamiento de 1 paso. En la Figura se observa que la diferencia son 79 píxeles (79 px) que corresponden con 40 micrómetros, aproximadamente, es decir, coincide con el tamaño del paso o resolución del motor caracterizada en el Apartado anterior, cuando no existía cambio en el sentido del movimiento.

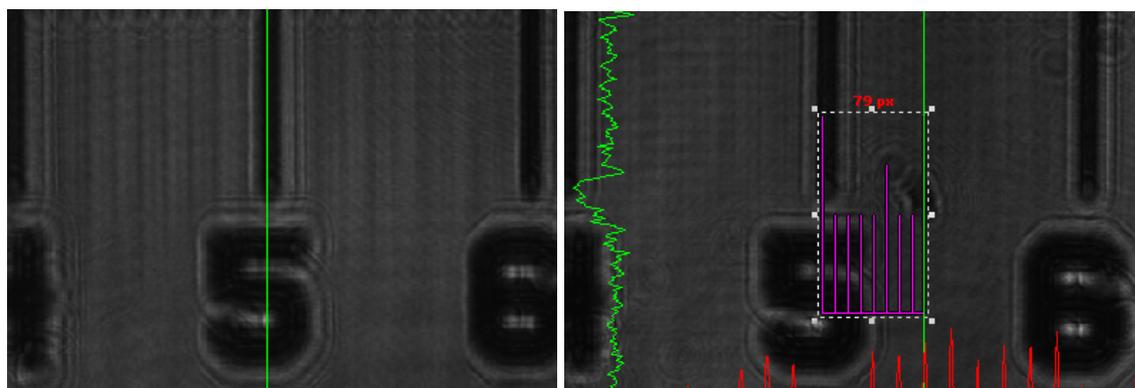


Figura 6.18. Medida de 1 paso con cambio en el sentido de movimiento

Para poder corregir por *software* el destensado de la correa, hay que decidir cuál es el número de pasos N más conveniente para realizar este calibrado. Este número debe ser suficientemente grande para garantizar el tensado de la correa, pero no demasiado grande, para que no resulten incómodos los desplazamientos añadidos.

Se ha realizado un conjunto repetitivo de experimentos en los que se ha utilizado N con diferentes valores: 30, 40, 50, 80 y 100 pasos. Para cada valor de N, y con objeto de elegir la solución que menos se aleje de la media, se ha realizado 10 veces la prueba descrita anteriormente. Este número es suficiente para elegir el número N.

- Prueba de tensado del motor con N= 30 pasos

La primera prueba se ha realizado con 30 pasos, un número suficiente de pasos para comenzar el experimento. Con este número de pasos parece seguro que se ha conseguido tensar la correa y avanzar. En la Figura 6.19 se pueden observar los resultados obtenidos para la 1ª captura, 5ª captura y 10ª captura. Se parte de la regla en la que 210 píxeles equivalen a 100 micrómetros. Para la 1ª captura (imagen de la izquierda) se ha puesto la referencia en la raya del 3 y se dan 30 pasos hacia la izquierda y 31 hacia la derecha. Para la 5ª captura (imagen central) se ha puesto la referencia en la raya del 4 y se dan 30 pasos hacia la izquierda y 31 hacia la derecha. Y así sucesivamente, hasta la décima captura (imagen de la derecha) donde se ha puesto la referencia en la raya del 6 y se dan 30 pasos hacia la izquierda y 31 hacia la derecha.

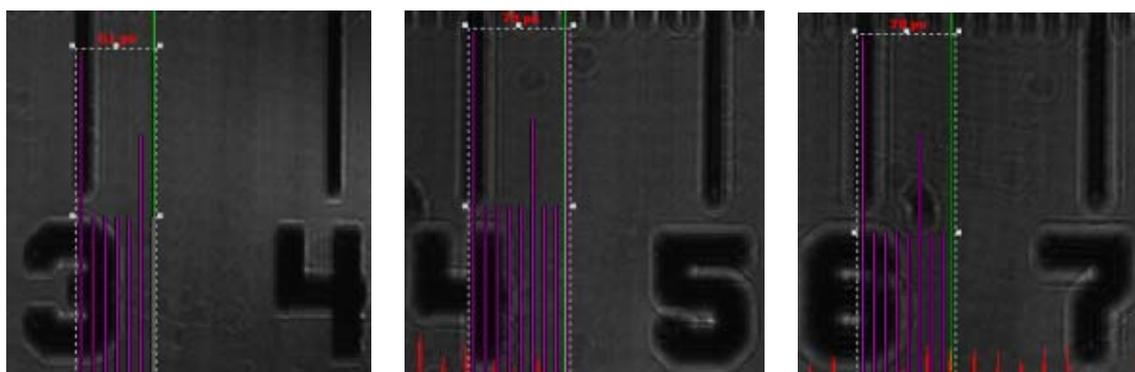


Figura 6.19. Prueba de tensado de la correa con N = 30 pasos

En la Tabla 6.1 se muestran los resultados para la medida del ajuste del tensado de la correa con N = 30 pasos.

210 px		100 μm
Tensado 30 pasos, movimiento 1 paso		
Captura	Medida (px)	Medidas (μm)
1ª	61	29,05
2ª	95	45,24
3ª	81	38,57
4ª	95	45,24
5ª	79	37,62
6ª	80	38,10
7ª	97	46,19
8ª	81	38,57
9ª	89	42,38
10ª	78	37,14
Media	82,20	39,14
Max	97	46,19
Min	61	29,05
+	14,80	
-	21,20	

Tabla 6.1. Medida de la resolución del motor con tensado de la correa para N = 30 pasos.

Cabe destacar, como conclusión de este ensayo con N = 30 pasos, que la media de las distancias es 82,20 píxeles pero, teniendo en cuenta que 100 micrómetros equivalen a 210 píxeles, la media de 1 paso es de 39,14 micrómetros. Hay una diferencia grande entre la media de los 10 valores expresada en píxeles y el valor más grande y más pequeño, que son 14 y 21 píxeles, por lo que es conveniente utilizar más pasos para tensar.

En la Figura 6.20 se puede ver la campana de Gauss correspondiente que muestra de forma muy visual la desviación de las medidas de la resolución del paso del motor en micrómetros respecto de la media.

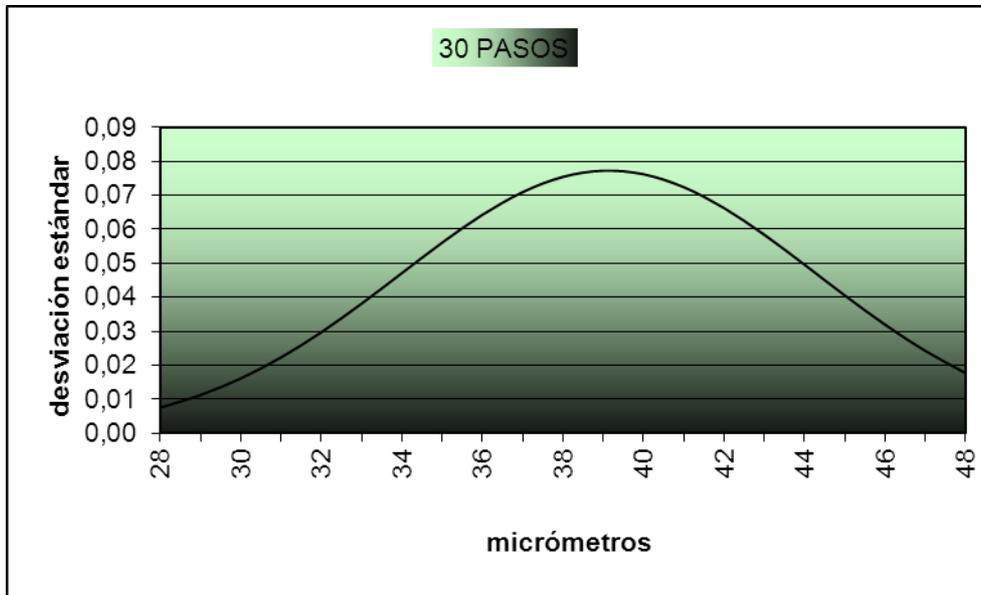


Figura 6.20. Campana de Gauss para N = 30 pasos.

- Prueba de tensado del motor con N= 40 pasos

La siguiente prueba se realizó con 40 pasos en un sentido y 41 en el sentido contrario. Se parte también en este caso de la regla en la que 210 píxeles equivalen a 100 micrómetros. De nuevo, para cada una de las sucesivas capturas, se ha puesto como referencia una raya de la regla micrométrica. En este caso, para la primera captura se ha puesto la referencia en el valor 7.

En la Figura 6.21 se pueden observar los resultados obtenidos para la 1ª captura, 5ª captura y 10ª captura.

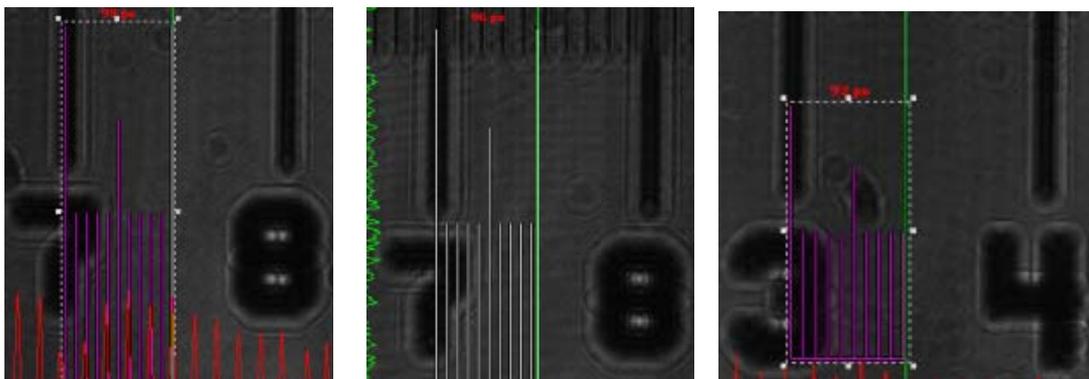


Figura 6.21. Prueba de tensado de la correa con N = 40 pasos

En la Tabla 6.2 se muestran los resultados para la medida del ajuste del tensado de la correa con N = 40 pasos.

	210 px	100 μ m
Tensado 30 pasos, movimiento 1 paso		
Captura	Medida (px)	Medidas (μ m)
1ª	99	47,14
2ª	75	35,71
3ª	94	44,76
4ª	80	38,10
5ª	96	45,71
6ª	94	44,76
7ª	82	39,05
8ª	82	39,05
9ª	83	39,52
10ª	91	43,33
Media	86,91	41,39
Max	99	47,14
Min	75	35,71
+	12,09	
-	11,91	

Tabla 6.2. Medida de la resolución del motor con tensado de la correa para N = 40 pasos.

La media para 1 paso es de 41,39 micrómetros y en este caso los valores máximo y mínimo están más cercanos a la media que en la prueba anterior, pero conviene hacer pruebas con valores mayores de N.

En la Figura 6.22 se muestra la campana de Gauss correspondiente.

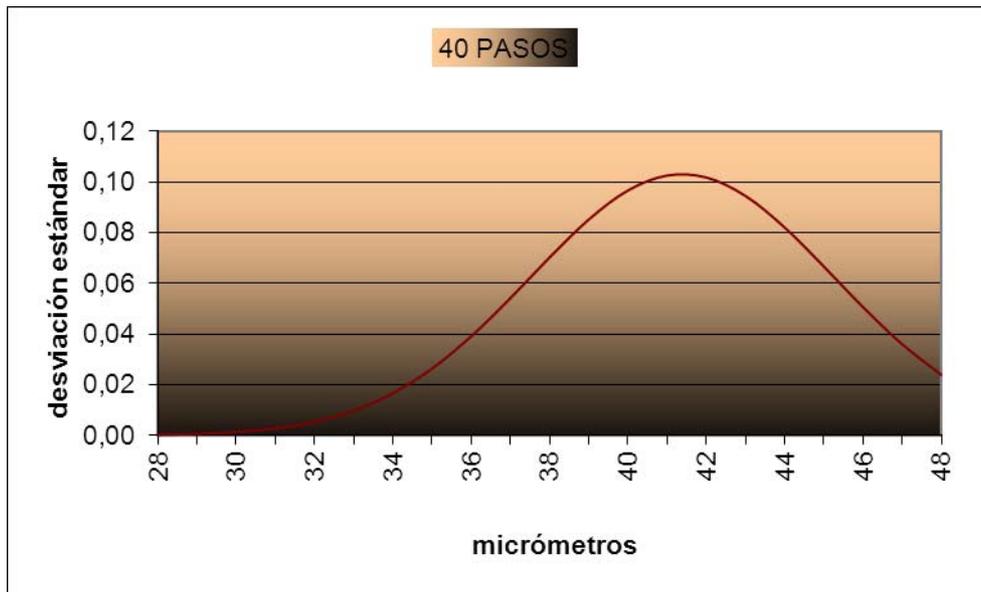


Figura 6.22. Campana de Gauss para N = 40 pasos

- Prueba de tensado del motor con N= 50 pasos

La siguiente prueba se ha realizado con 50 pasos. Al igual que en el caso anterior, se dieron 50 pasos en un sentido con la correa destensada y 51 en el sentido contrario. Se parte de la regla donde 210 píxeles equivalen a 100 micrómetros.

En la Figura 6.23 se pueden observar los resultados obtenidos para la 1ª captura, 5ª captura y 10ª captura.

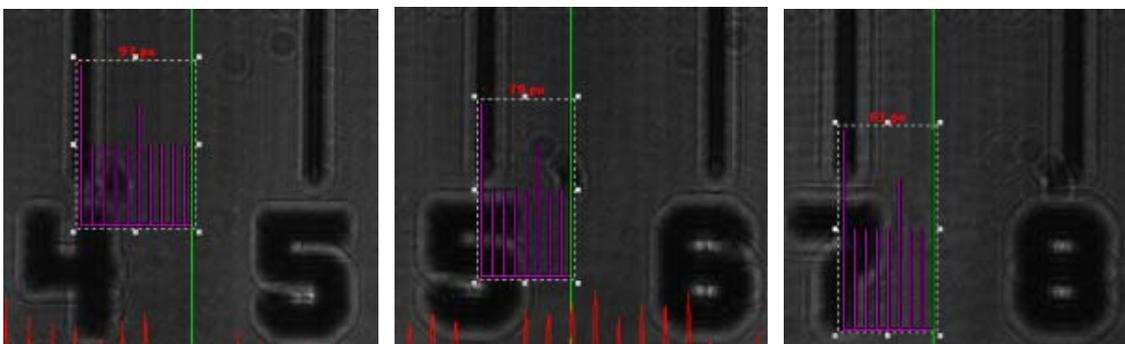


Figura 6.23. Prueba de tensado de la correa con N = 50 pasos

En la Tabla 6.3 se muestran los resultados para la medida del ajuste del tensado de la correa con N =50 pasos.

210 px		100 μ m
Tensado 30 pasos, movimiento 1 paso		
Captura	Medida (px)	Medidas (μ m)
1ª	97	46,19
2ª	81	38,57
3ª	85	40,48
4ª	85	40,48
5ª	79	37,62
6ª	93	44,29
7ª	94	44,76
8ª	80	38,10
9ª	96	45,71
10ª	81	38,57
Media	86,58	41,23
Max	97	46,19
Min	79	37,62
+	10,42	
-	7,58	

Tabla 6.3. Medida de la resolución del motor con tensado de la correa para N = 50 pasos.

La media para 1 paso es de 41,23 micrómetros y en este caso los valores máximo y mínimo están más cercanos a la media que en los casos anteriores, por lo que conviene aumentar el número de pasos hasta llegar al valor óptimo.

En la Figura 6.24 se muestra la campana de Gauss correspondiente.

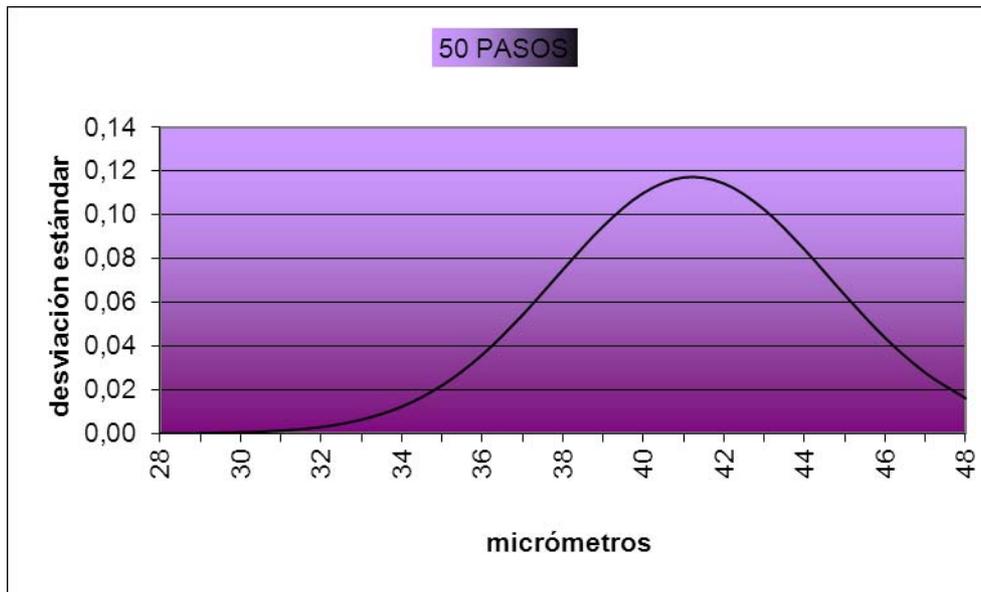


Figura 6.24. Campana de Gauss para N = 50 pasos

- Prueba de tensado del motor con N= 80 pasos

Visto que, a medida que aumentan los pasos, los valores máximo y mínimo se van acercando más a la media, la siguiente prueba se realiza con 80 pasos. Se darán, por tanto, 80 pasos en un sentido y 81 en el sentido contrario. Se parte de la regla donde 202 píxeles equivalen a 100 micrómetros.

En la Figura 6.25 se pueden observar los resultados obtenidos para la 1ª captura, 5ª captura y 10ª captura.

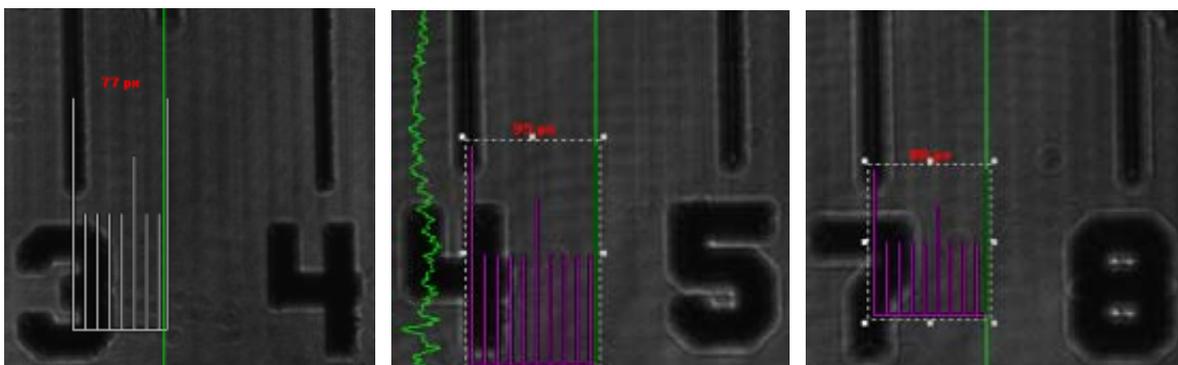


Figura 6.25. Prueba de tensado de la correa con N = 80 pasos

En la Tabla 6.4 se muestran los resultados para la medida del ajuste del tensado de la correa con N = 80 pasos.

	202 px	100µm
Tensado 30 pasos, movimiento 1 paso		
Captura	Medida (px)	Medidas (µm)
1ª	77	38,12
2ª	96	47,52
3ª	79	39,11
4ª	92	45,54
5ª	95	47,03
6ª	81	40,10
7ª	85	42,08
8ª	92	45,54
9ª	88	43,56
10ª	88	43,56
Media	86,83	42,99
Max	96	47,52
Min	77	38,12
+	9,17	
-	9,83	

Tabla 6.4. Medida de la resolución del motor con tensado de la correa para N = 80 pasos.

La media para 1 paso es de 42,99 micrómetros y en este caso los valores máximo y mínimo están ya suficientemente cercanos a la media, pero es conveniente realizar otra prueba con más pasos para comprobar si la desviación disminuye al aumentar el número por encima de los 80 pasos.

En la Figura 6.26 se muestra la campana de Gauss correspondiente a 80 pasos.

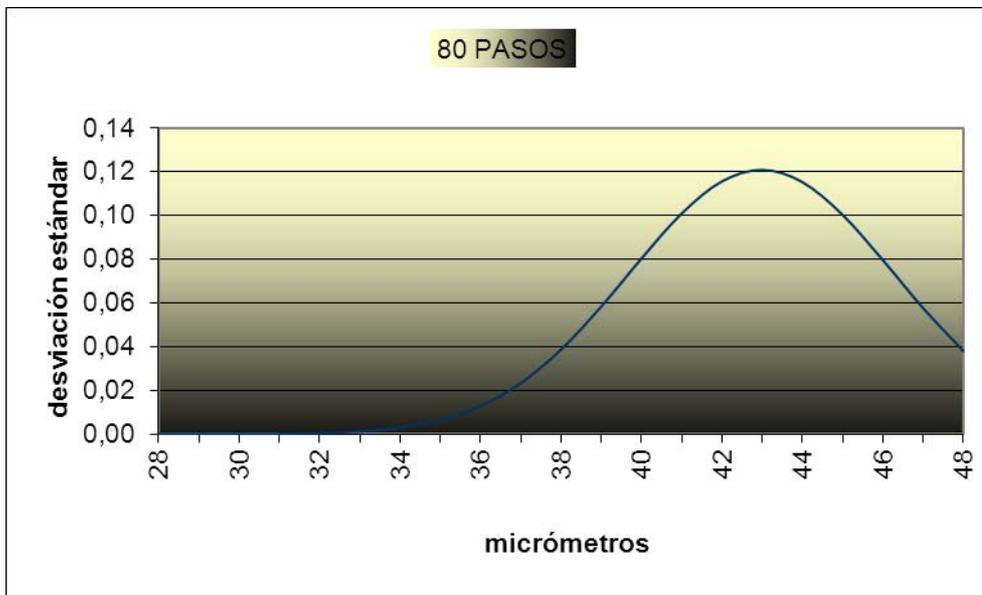


Figura 6.26. Campana de Gauss para N = 80 pasos

- Prueba de tensado del motor con N= 100 pasos

Por último, se ha realizado una prueba de 100 pasos en un sentido y 101 en el sentido contrario. Se parte de la regla donde 202 píxeles son 100 micrómetros.

En la Figura 6.27 se pueden observar los resultados obtenidos para la 1ª captura, 5ª captura y 10ª captura.

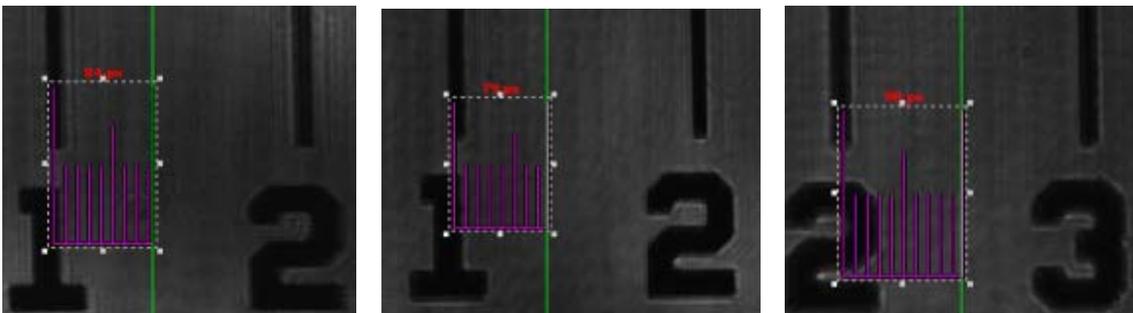


Figura 6.27. Prueba de tensado de la correa con N = 100 pasos

En la Tabla 6.5 se muestran los resultados para la medida del ajuste del tensado de la correa con N = 100 pasos.

202 px		100 μ m
Tensado 30 pasos, movimiento 1 paso		
Captura	Medida (px)	Medidas (μ m)
1 ^a	84	41,58
2 ^a	94	46,53
3 ^a	77	38,12
4 ^a	97	48,02
5 ^a	79	39,11
6 ^a	95	47,03
7 ^a	76	37,62
8 ^a	97	48,02
9 ^a	77	38,12
10 ^a	97	48,02
Media	86,37	42,76
Max	97	48,02
Min	76	37,62
+	10,63	
-	10,37	

Tabla 6.5. Medida de la resolución del motor con tensado de la correa para N = 100 pasos.

La media para 1 paso es de 42,76 micrómetros y en este caso los valores máximo y mínimo vuelven a alejarse de la media, por lo que se decide no incrementar el número N.

En la Figura 6.28 se muestra la campana de Gauss correspondiente.

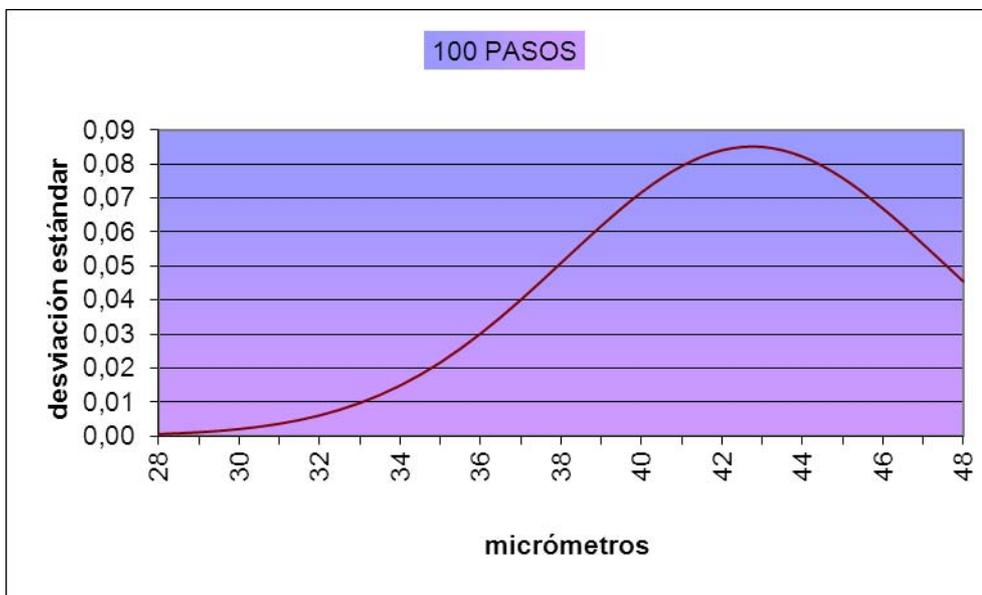


Figura 6.28. Campana de Gauss para N = 100 pasos

El resumen de los resultados obtenidos aparece reflejado en la Tabla 6.6.

Nº de pasos	Media (px)	Dif. máx (px)	Dif. mín (px)	Paso (μm)
30	82,20	14,80	21,20	39,14
40	86,91	12,09	11,91	41,39
50	86,58	10,42	7,58	41,23
80	86,83	9,17	9,83	42,99
100	86,37	10,63	10,37	42,76

Tabla 6.6. Resumen de resultados de dispersión respecto de la media en la medida del paso del motor

La desviación estándar, que mide cuánto se alejan los datos de la media aritmética, es muy pequeña en todos los casos, por lo que la corrección parece adecuada con cualquiera de los números N ensayados. Sin embargo, cuando se dan N = 80 pasos la diferencia entre los valores máximo y mínimo es menor de 10 píxeles, por lo que esta cifra parece ser la más aconsejable.

6.4.4 Medición de la precisión obtenida con una corrección de 80 pasos

A la vista de los resultados anteriores, se ha elegido la cantidad $N = 80$ pasos como la cifra más conveniente para corregir el tensado de la correa. Así pues, cada vez que tenga que desplazarse el posicionador en un sentido cualquiera, se desplazará la cantidad deseada incrementada en 80 pasos y, cuando haya llegado al término de su recorrido, se desplazará 80 pasos en sentido contrario, quedando situado en la posición deseada.

Para poder realizar un análisis estadístico riguroso de una distribución normal, o campana de Gauss, es necesario disponer de una muestra constituida, al menos, por 30 elementos. Por tanto, se han realizado 20 pruebas más dando $N = 80$ pasos en un sentido y retrocediendo $N+1 = 81$ pasos, en las mismas condiciones en que se hicieron las 10 pruebas anteriores, es decir, que 202 píxeles equivalgan a 100 micrómetros. Así, se obtiene una muestra total de 30 resultados.

En la Tabla 6.7 se muestran los resultados obtenidos en las 30 pruebas realizadas.

Capturas 1ª a 30ª		
77	90	79
96	67	81
79	87	79
92	84	86
95	82	80
81	88	76
85	84	85
92	84	84
88	83	83
88	88	86

Tabla 6.7. Resultados obtenidos en 30 medidas de la resolución del motor en píxeles con prueba de tensado de la correa para $N = 80$ pasos.

En la Tabla 6.8 se indican los valores estadísticos obtenidos: media, desviación estándar y diferencias de los valores máximo y mínimo con la media.

Media (μm)	41,80
Desv. Estdr.	2,89
MAX (μm)	47,52
MIN (μm)	33,50
Diferencia +	5,72
Diferencia -	8,30

Tabla 6.8. Datos característicos de calibrado del paso del motor del microposicionador medidos en micrómetros

Como se puede apreciar la media es más baja que la que se había calculado con 10 datos, siendo ahora de 41,80 micrómetros en vez de 42,99 micrómetros. Además, los valores máximo y mínimo se desvían muy poco de la media; por tanto, la precisión del movimiento será mayor que la propuesta anteriormente.

En la Figura 6.29 se muestra la campaña de Gauss correspondiente.

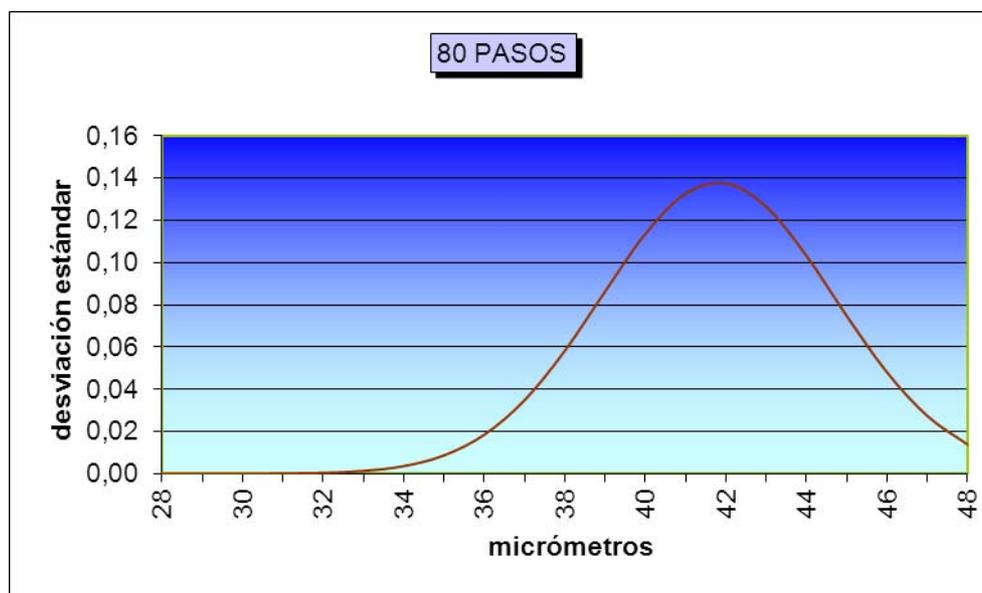


Figura 6.29. Campana de Gauss para 30 pruebas de 80 pasos

A continuación, y para comprobar la robustez de y repetitividad en la ubicación del motor en posiciones definidas, es decir, se ha comprobado la precisión que adquiere el sistema con esta corrección en el tensado de la correa. Para ello se ha fijado una posición de referencia y se han realizado diferentes desplazamientos, consistentes

todos ellos en avanzar y retroceder el mismo número de pasos (con la corrección implícita en cada caso) y comprobar si la vagoneta con la regla micrométrica calibrada vuelve al punto de partida. Se han realizado, para ello, desplazamientos de 1 paso, 10 pasos, 25 pasos, 50 pasos, 100 pasos, 200 pasos, 300 pasos, 500 pasos y 1000 pasos.

En la Figura 6.30 izquierda se puede ver la captura de la posición inicial de referencia en la marca del número 5. En las Figuras 6.30 derecha y 6.31 a 6.34 se muestran las capturas tras realizar cada una de las secuencias de desplazamientos indicadas anteriormente. Por ejemplo, en la Figura 6.32 izquierda se muestra la captura final tras haberse movido el motor 50 pasos reales a la derecha y 50 pasos reales a la izquierda (con las correcciones por retensado de la correa implícitas).

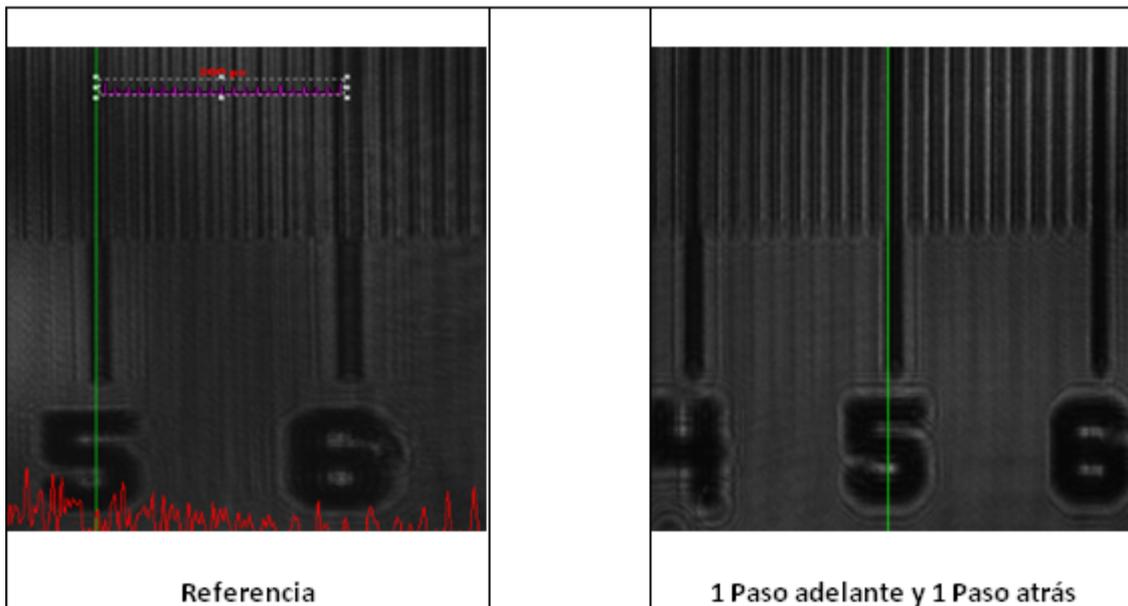


Figura 6.30. Capturas de referencia y de 1 paso

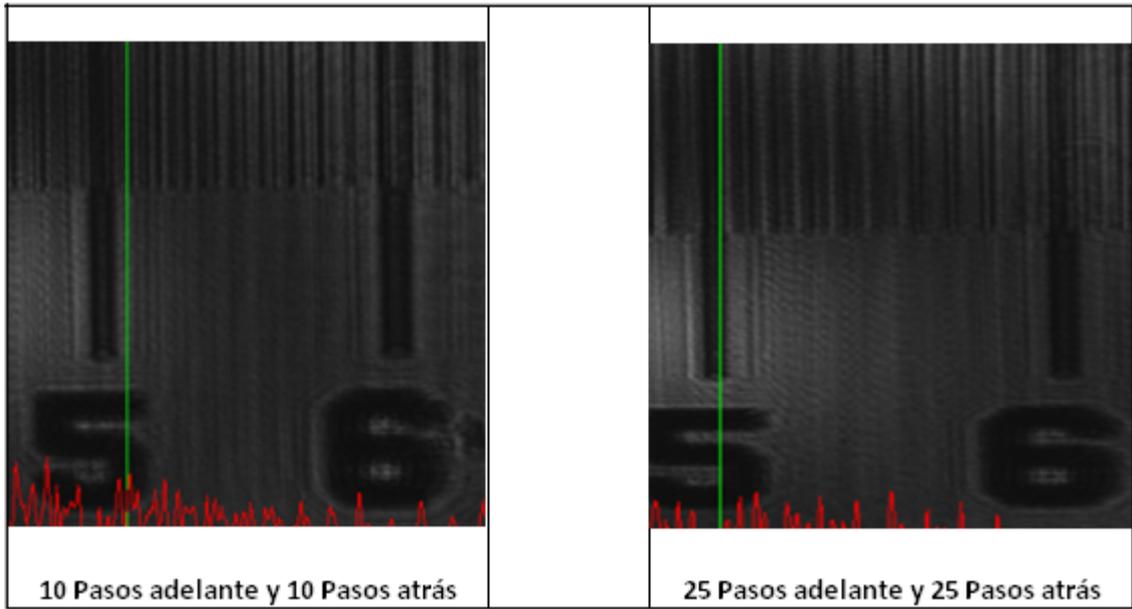


Figura 6.31. Capturas de 10 y de 25 pasos

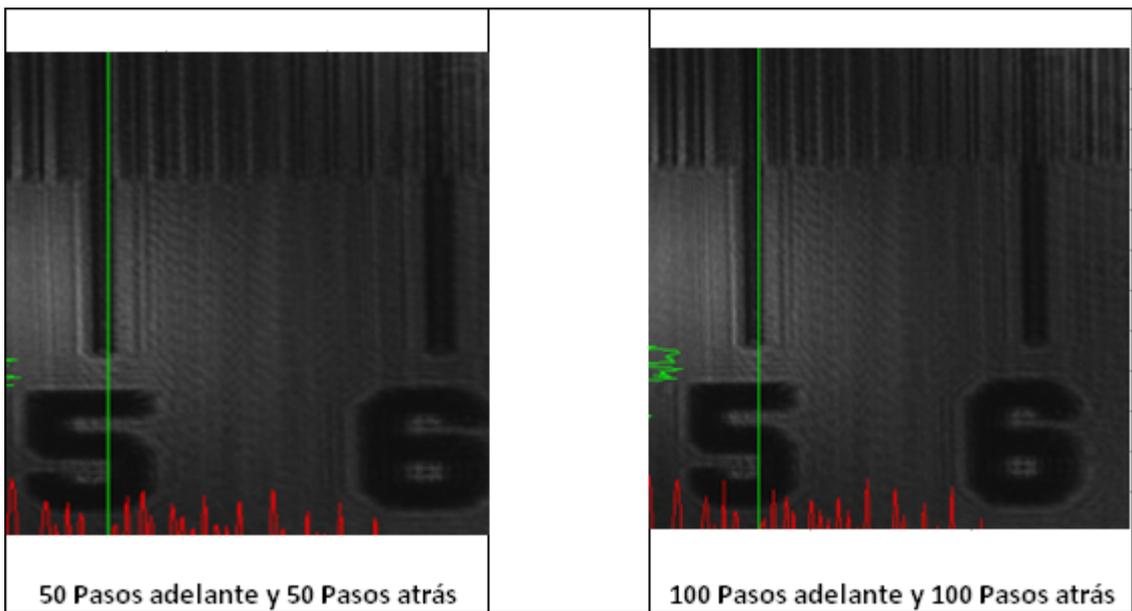


Figura 6.32. Capturas de 50 y de 100 pasos

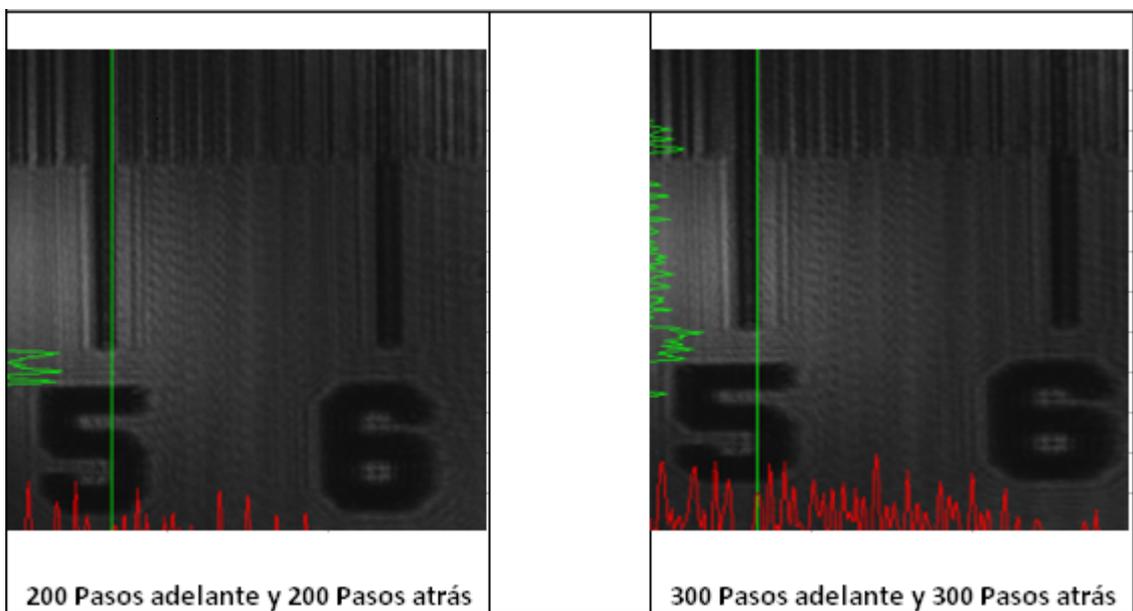


Figura 6.33. Capturas de 200 y de 300 pasos

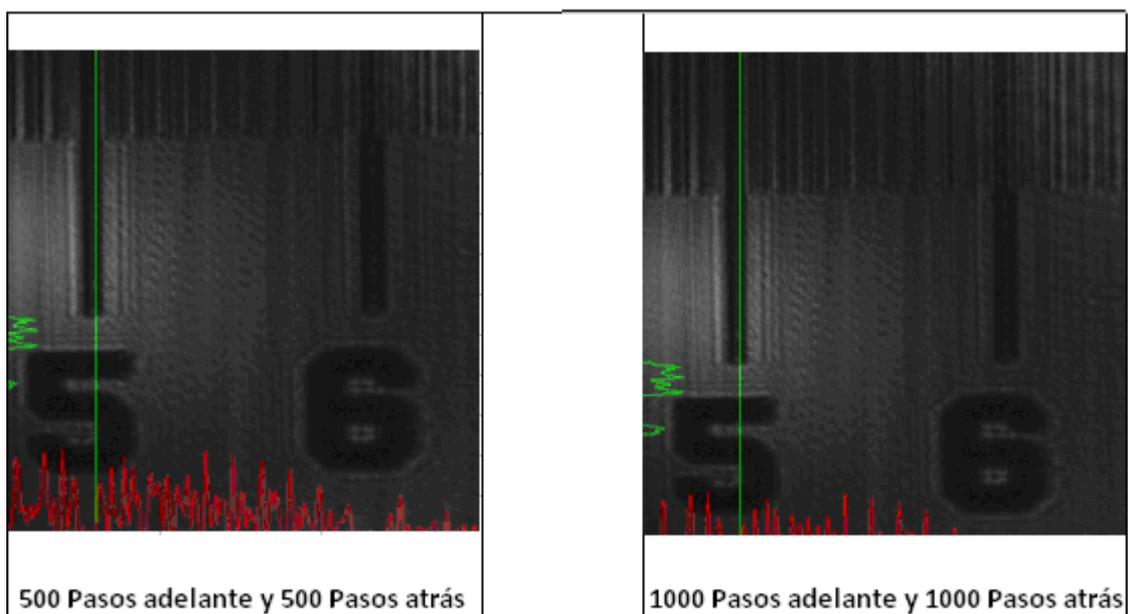


Figura 6.34. Capturas de 500 y de 1000 pasos

Como se puede apreciar en las imágenes, el movimiento es muy preciso en todos los desplazamientos, aunque cabe destacar que se aprecia mayor precisión en los desplazamientos largos que en los cortos.

Debido a la pequeña magnitud del error, fue imposible medirlo con el material disponible, ya que la regla que se utiliza como referencia de la cámara no es suficientemente sensible para realizar esta medición.

Para no tener que ajustar el destensado de la correa en todos los desplazamientos (sean a derechas o a izquierdas), se ha decidido que el sistema quede siempre tensado a la derecha, por lo que no será necesario tener en cuenta el destensado cuando el movimiento a realizar sea hacia la derecha. Para ello, cuando el movimiento a realizar sea hacia la izquierda se realizará primero el movimiento a izquierdas (de amplitud $N+80$) y después se realizará el movimiento a derechas (de amplitud 80), con lo que el movimiento resultante será de N pasos hacia la izquierda y la correa quedará tensada hacia la derecha, a la espera de un nuevo desplazamiento.

6.5. INICIALIZACIÓN Y CONEXIONES A PUERTOS

Una vez comprobado el correcto funcionamiento del microposicionador con la introducción de la corrección, los siguientes pasos a realizar consisten en comprobar el PROGRAMA de RESET y la comunicación con el Hyperterminal a través del puerto USB.

6.5.1 Verificación de funcionamiento del PROGRAMA de RESET

Es preciso probar el PROGRAMA de RESET, cuya funcionalidad se describió en Capítulos anteriores. El accionamiento del RESET provoca el posicionamiento de la vagoneta en el extremo izquierdo del carril y su reubicación en una posición fija, la 2500, definida en el código del programa.

Para poder verificar el funcionamiento correcto del RESET, se desplazó el conjunto del microposicionador hacia la izquierda todo lo posible, dejando fijo el resto del montaje, como se aprecia en la Figura 6.35.

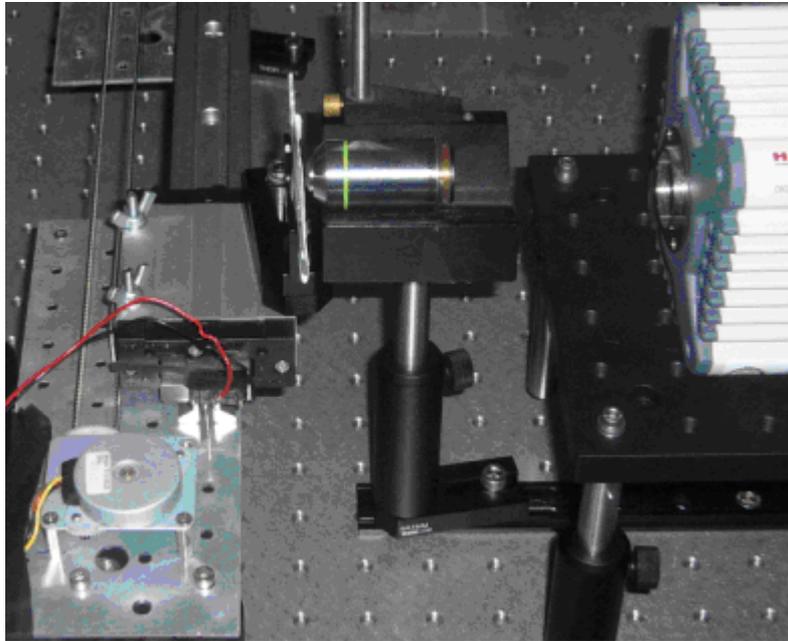


Figura 6.35. Sistema mecánico recolocado con el extremo del carril enfrente de la cámara CCD de captura de imágenes

Para ello se introdujo una modificación en el código, consistente en cambiar la posición 2500 por la posición 5005. Se consideró que cuanto mayor fuera la distancia que tuviera que recorrer el microposicionador, más fácil sería poder medir los posibles errores. Se dejó el microposicionador en esa posición y, con la cámara, se colocaron las referencias, como se muestra en la Figura 6.36.

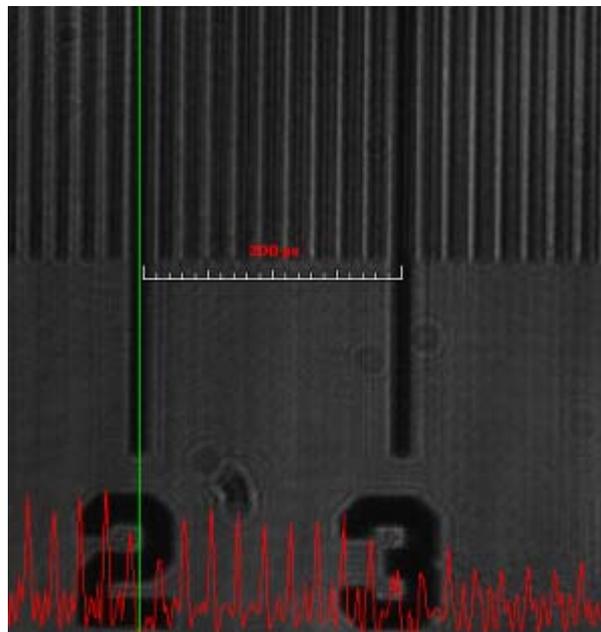


Figura 6.36. Referencias para la comprobación de funcionamiento del PROGRAMA de RESET

Una vez hecho esto, se reseteó el microcontrolador y se dejó accionado el pulsador de INTRO del panel frontal del sistema de control. Después de los mensajes iniciales de pantalla, la vagoneta se desplazó hasta el extremo izquierdo del carril avanzando hasta la posición 5005.

En la Figura 6.37 se muestran los resultados obtenidos al realizar este ensayo 10 veces.

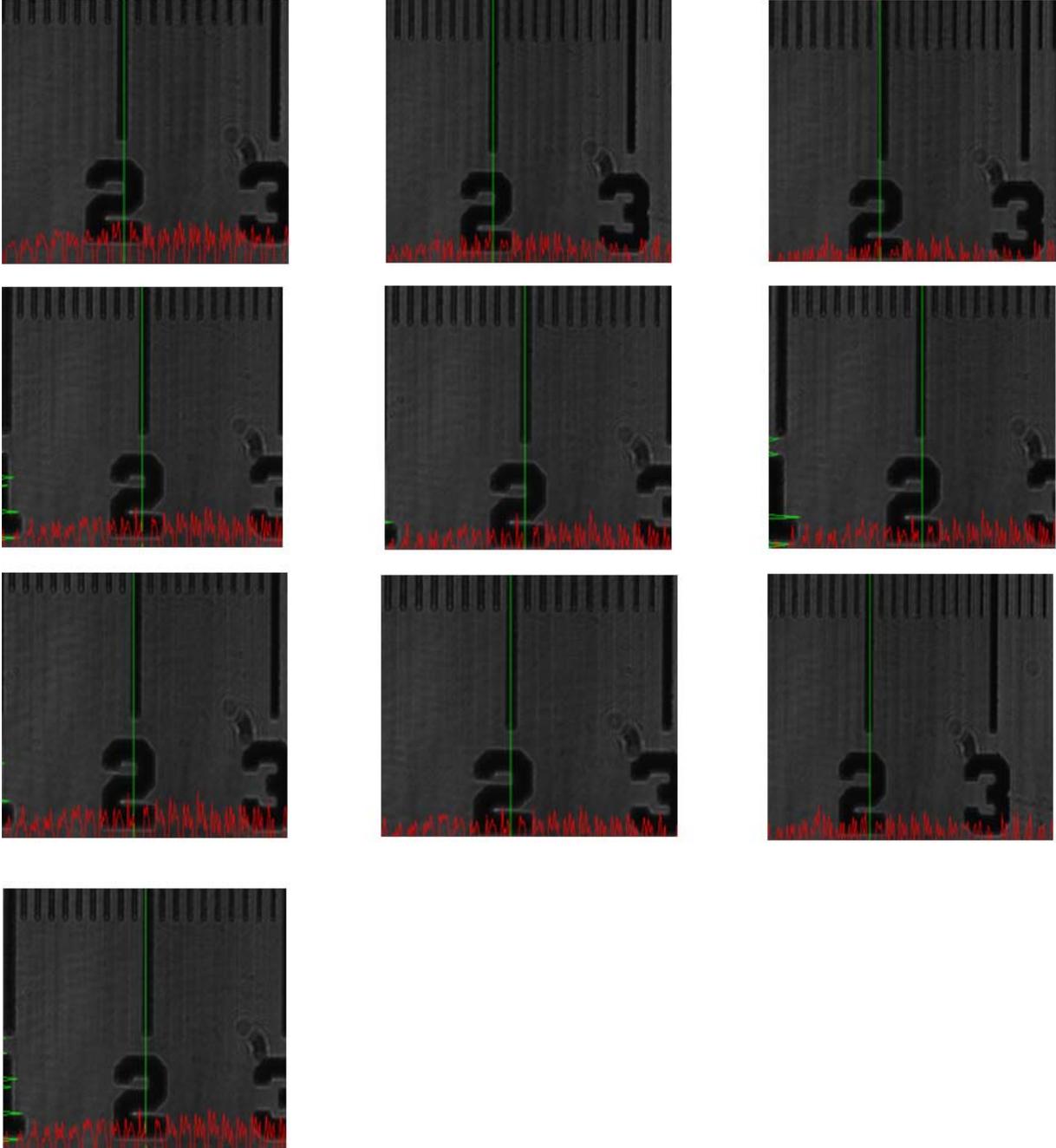


Figura 6.37. Captura del resultado tras el accionamiento del botón de RESET

Cabe destacar que los resultados son muy parecidos en todos los casos, pudiéndose afirmar que el movimiento es muy preciso. Al igual que en el caso anterior, se aprecia que existe un error, pero es muy pequeño y resulta imposible cuantificarlo por falta de medios suficientemente sensibles.

6.5.2 Verificación de funcionamiento de la conexión USB

Habiendo comprobado el correcto funcionamiento de la placa, se han realizado finalmente pruebas de control sobre el posicionador a través del ordenador. Una vez colocado el cable USB entre el ordenador y la placa, se ha abierto el programa Hyperterminal en el ordenador a través de la ruta Inicio/ Programas/ Accesorios/ Comunicaciones/ Hyperterminal.

Una vez puesto todo en marcha se ha comprobado cómo la misma información que se muestra por la pantalla LCD es mostrada en la pantalla del ordenador.

A través de las teclas 1 ó INTRO del ordenador se consigue el desplazamiento del microposicionador en los dos sentidos.

En la Figura 6.38 se puede observar que la comunicación es correcta.

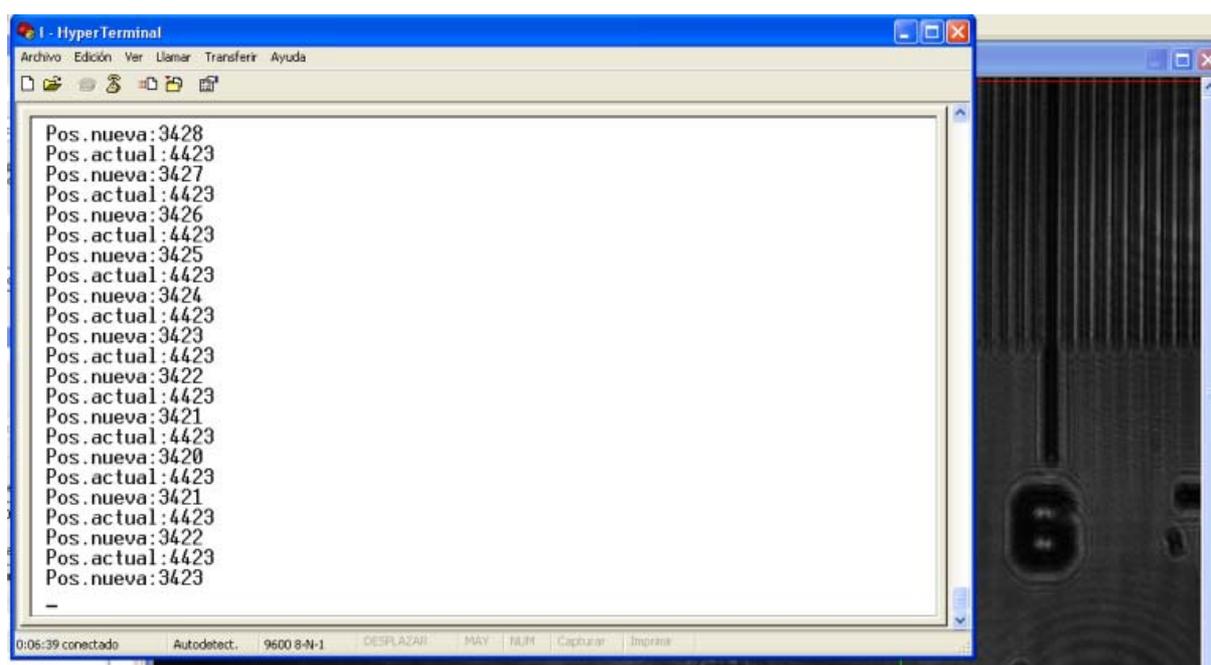


Figura 6.38. Comunicación entre USB y microcontrolador

CAPÍTULO 7

CONCLUSIONES Y MEJORAS FUTURAS

7.1 CONCLUSIONES

Una vez realizado el Proyecto y hechas las mediciones correspondientes, se ha llegado a las siguientes conclusiones:

- ✓ Se ha diseñado un sistema microposicionador basado en un carril y una vagoneta que inicialmente poseían movimiento manual.
- ✓ Se ha desarrollado un microposicionador autónomo que no necesita estar conectado a un ordenador para trabajar. Es suficiente alimentar el motor para poner en marcha el microposicionador. Gracias a la LCD, en todo momento se conoce la posición que ocupa el microposicionador. A través de los pulsadores se marcan las posiciones para trabajar con desplazamientos pequeños (del orden de micrómetros) o con desplazamientos grandes (del orden de centímetros).
- ✓ Se ha dispuesto una comunicación USB para poder visualizar en un ordenador la posición del microposicionador y, a su vez, poder controlar el microposicionador desde el ordenador.
- ✓ Se han diseñado e implementado las placas electrónicas de circuito impreso del circuito total. Se han fabricado dos placas PCB, una para el motor y otra para los pulsadores. La placa del motor controlará el desplazamiento del mismo y éste, a su vez, el de la vagoneta sobre el carril. La placa de pulsadores servirá para desplazarse a las posiciones deseadas.
- ✓ Se ha programado el código de control del microposicionador. Se ha desarrollado el código de comunicaciones entre módulos: placa de control del motor, placa de control de los pulsadores externos de desplazamiento y comunicaciones con el PC mediante conexión USB.
- ✓ Se ha calibrado el conjunto final del sistema microposicionador en un eje, corrigiendo el destensado de la correa que se producía con cada cambio de sentido del movimiento.

- ✓ Se ha conseguido un microposicionador de largas distancias más barato que cualquiera del mercado y con un error de la misma magnitud que los que actualmente se comercializan

A continuación se muestra la Tabla 7.1, que resume las características principales del microposicionador comercial de la marca Zaber T-LSR300D comparándolas con las del prototipo diseñado, ambos con la misma longitud de carril.

Especificaciones	Comercial	Prototipo
Avance en cada paso	1,98 μm	41,80 μm
Máxima desviación recorriendo el carril completo	$\pm 20 \mu\text{m}$	$< 10 \mu\text{m}$
Repetitividad	$< 3 \mu\text{m}$	$< 10 \mu\text{m}$
Error cambio de sentido	$< 20 \mu\text{m}$	$< 10 \mu\text{m}$
Precio	1.721,42€	1118,25€

Tabla 7.1. Comparación de un micrómetro comercial con el prototipo diseñado

Cabe destacar que, debido a la ausencia de medios suficientemente sensibles el mínimo error medible es de 10 μm , por lo que sólo se puede afirmar que el error es inferior a este valor, aunque por las pruebas realizadas se puede aventurar que el prototipo es más preciso que el micrómetro comercial.

El precio que se ha puesto para el prototipo es el obtenido en el Apartado PRESUPUESTO, al final de este Documento, donde se explica la forma de obtenerlo.

Una vez finalizados el diseño e implementación del sistema, y comprobado su correcto funcionamiento, se puede afirmar que se han cumplido satisfactoriamente los objetivos marcados.

7.2 MEJORAS FUTURAS

A partir del diseño creado, se pueden desarrollar mejoras futuras en el dispositivo, como las que se indican a continuación.

- 1.- Debido a que no se realizaban correctamente los cambios de sentido por culpa del destensamiento de la correa, ha sido necesario introducir por software una corrección de 80 pasos en los desplazamientos.

Una posibilidad alternativa de conseguir que los movimientos sean correctos, sería poner unos detectores de infrarrojos sobre el carril y así, según vaya avanzando la vagoneta, irá pasando por los detectores y se podrá saber la posición que ocupa.

Otra posibilidad consistiría en disponer un tornillo sin fin en lugar de la correa, ya que con él no se vería afectado el cambio de sentido en los movimientos.

- 2.- Debido a que los desplazamientos son muy grandes con el motor unipolar, éste podría ser sustituido por un motor híbrido. Los motores híbridos realizan un gran número de pasos por vuelta y, a cambio, no se encarecería demasiado el producto final.
- 3.- Una vez conseguido un posicionador que se desplaza en un eje, sobre la vagoneta se podría colocar otra plataforma que se desplace perpendicularmente a ese eje, y así obtener un posicionador sobre dos ejes ortogonales, X e Y.
- 4.- Ya que se ha realizado una conexión mediante USB que sirve de comunicación con el PC se podría desarrollar un software de control del motor con un programa de entorno gráfico como es el Labview, para control de instrumentación que permita definir ciclos de movimiento en los pasos del motor, por ejemplo

BIBLIOGRAFÍA

REFERENCIAS BIBLIOGRÁFICAS

1. "Curso Avanzado de Microcontroladores PIC". Edison Duque Cardona. Ed. Cedit (2000).
2. "PIC Robotics". Iovine, John. Ed. McGraw Hill. E.U. A., 290 p.p. (2004)
3. "Microcontroladores PIC. Diseño práctico de aplicaciones 16F84 PA I". Angulo Martínez, José María. Ed. McGraw Hill.(2003)
4. Tutorial microcontroladores. Miguel Torres Torriti http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf (2007)
5. Tutorial motores paso a paso. Copyright © 2000-2010 TodoRobot. <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm> motores pap
6. Descarga compilador CCS C Compiler. Copyright © 2011. <http://www.ccsinfo.com>
7. Manual PIC 18F2550 © 2009 Microchip Technology Inc <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>
8. Tutorial USB-PIC18. Copyright © 2001-2011 Migsantiago.com http://www.migsantiago.com/index.php?option=com_content&view=article&id=9:pic-usb-para-principiantes&catid=1:tutorial&Itemid=23
9. Tutorial A/D-Pantalla LCD-PIC18 <http://picmania.garcia-cuervo.net/conceptos.php>

ANEXOS

ANEXO 1:HOJA CARACTERÍSTICAS

PIC 18F2550



PIC18F2455/2550/4455/4550

Data Sheet

28/40/44-Pin, High-Performance,
Enhanced Flash, USB Microcontrollers
with nanoWatt Technology

ANEXO 2:HOJA CARACTERÍSTICAS

DRIVER L293D



L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

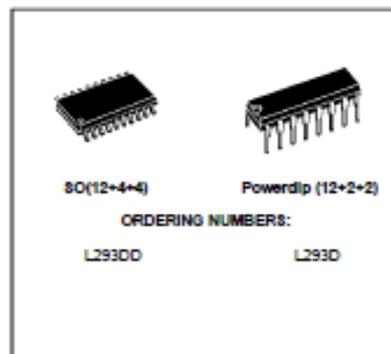
- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

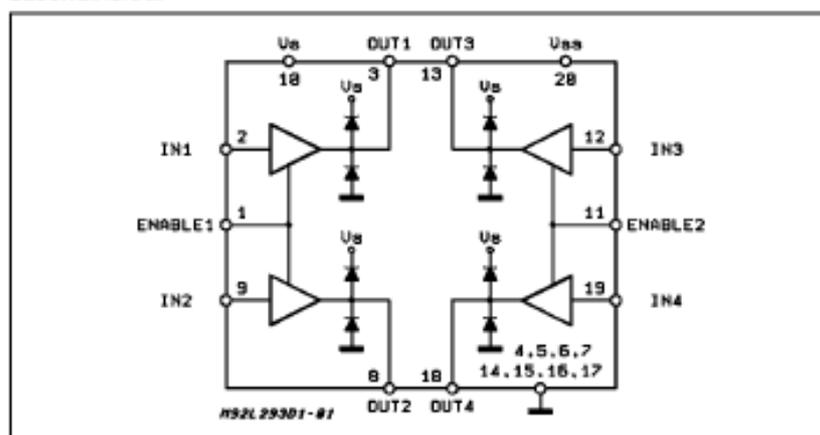
This device is suitable for use in switching applications at frequencies up to 5 kHz.



The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking.

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

BLOCK DIAGRAM



June 1996

1/7

ANEXO 3:HOJA CARACTERÍSTICAS

REGULADOR LM7805



www.fairchildsemi.com

MC78XX/LM78XX/MC78XXA

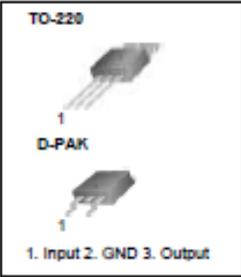
3-Terminal 1A Positive Voltage Regulator

Features

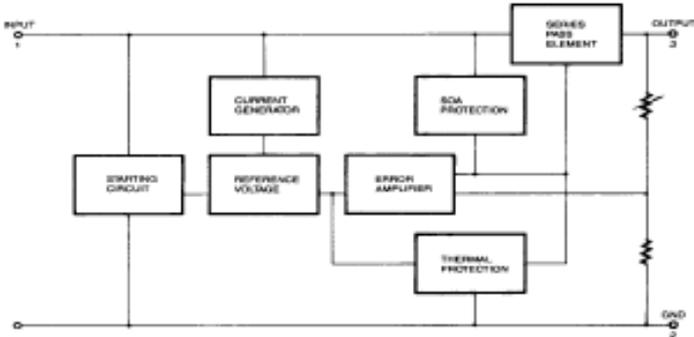
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



Rev. 1.0.1

©2001 Fairchild Semiconductor Corporation

ANEXO 4:HOJA CARACTERÍSTICAS

PANTALLA LCD 162D

Displaytech Ltd LCD MODULE 162D SERIES Version : 2.0 P1 of 19

PRODUCT SPECIFICATIONS

- PHYSICAL DATA
- EXTERNAL DIMENSIONS
- BLOCK DIAGRAM
- ABSOLUTE MAXIMUM RATINGS
- ELECTRICAL CHARACTERISTICS
- OPERATING PRINCIPLES & METHODS
- DISPLAY DATA RAM ADDRESS MAP
- ELECTRO-OPTICAL CHARACTERISTICS
- INTERFACE PIN CONNECTIONS
- RELIABILITY
- QUALITY GUARANTEE
- INSPECTION CRITERIA
- PRECAUTIONS FOR USING LCD MODULES
- USING LCD MODULES

ANEXO 5:HOJA CARACTERÍSTICAS VAGONETA-CARRIL

DryLin® T

Phone +44 (0) 16 04 - 67 72 40
Fax +44 (0) 16 04 - 67 72 45

igus® UK Ltd

Internet www.igus.co.uk
E-mail sales_uk@igus.co.uk

61.2



DryLin® T | Linear Guide System

DryLin® T linear guide systems were originally developed for applications in both automation and materials handling. The goal was to create a high performance, maintenance-free linear guide for use in the most diverse, even extreme environments.

Technical Data

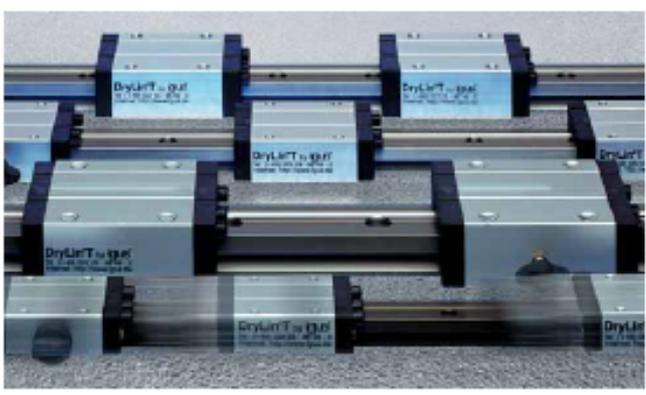
Sliding elements:
Maintenance-free

Material:
Iglidur® J

Max. surface speed:
15 m/s

Temperature range:
-40 °C to +90 °C

* Other materials upon request





DryLin® linear guide system in a treatment machine



Valve machining produces an extreme environment



DryLin® T linear guide system in pneumatic doors of tool changers



- ❶ Slide rails and base structure of the carriages manufactured from aluminium Al Mg Si 0.5
The rail is hard anodized, the aluminium carriage housing is clear anodized
- ❷ 6 sliding Iglidur® J elements act as guide bearings, which are set in opposing pairs and act as three guide bearings altogether
- ❸ Each of the three guide bearings can be adjusted manually or automatically
- ❹ All steel parts are galvanized or stainless steel
- ❺ The end plate is solid plastic

Technical Data

Guide rails	
Material	Aluminium, extruded section
Substance	Al Mg Si 0,5
Coating	Hard anodized aluminium, 50 µm
Hardness	500 HV
Sliding carriage	
Base structure	Aluminium, extruded section
Material	Al Mg Si 0,5
Coating	Anodized aluminium
Sliding elements	Maintenance-free plain bearing Iglidur® J
Bolts, springs	Stainless steel, galvanized steel
Cover	Plastic
Max. surface speed	15 m/s
Temperature range	-40 °C to +90 °C

Lifetime calculation, CAD files and much more support ► www.igus.co.uk/en/DryLinT

ANEXO 6: CÓDIGO FUENTE DEL SOFTWARE DEL MICROCONTROLADOR

```
//*****  
***  
//  
//CONTROL MICROPOSICIONADOR  
//  
//  
//  
//*****  
***  
#include <18F2550.h>  
#device ADC=8  
#fuses HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN  
#use delay(clock=48000000)  
#include "include/usb_cdc.h" //USB Configuration  
#include "flex_lcd.c"  
  
#define EN PIN_C0  
#define FIND PIN_A5  
#define FINI PIN_A4  
  
#define A1 PIN_C7  
#define A2 PIN_C6  
#define B1 PIN_C2  
#define B2 PIN_C1  
//Variables globales  
  
int8 test1[]="Test LCD 162D ";  
int8 test2[]="PIC18F2550 V3.3";  
int8 test3[]="USB conectado ";  
  
int paso[4]={0b1100, 0b0110, 0b0011, 0b1001};  
unsigned int16 numero_pasos=0;  
unsigned int16 aux2;  
unsigned int16 pos;  
int8 tecla;  
  
int8 modo=0; //Nos indica si el USB se encuentra conectado  
  
//SUBROUTINA DE ALMACENAMIENTO DATO EN EEPROM  
//  
void guarda_eeprom(int address, int data){
```

```

    write_eeprom (address, data);
}

void posicion_actual(void){
    unsigned int8 aux;

    aux2 = 0x0000;
    aux =read_eeprom (0x02);
    aux2 = aux<<8;
    aux = read_eeprom (0x01);
    aux2 = aux2 | aux;
    numero_pasos=aux2;
    pos=aux2;
}
//SUBROUTINA DE CONTROL DEL MOTOR PASO A PASO
//
void paso_motor(BYTE speed, BYTE dir, BYTE steps) {
    static BYTE stepper_state = 0;
    BYTE i;

    for(i=0; i<steps; ++i) {
        delay_ms(speed);

        if(dir!='L'){ //movimientos derecha
            stepper_state=(stepper_state+1)&(sizeof(paso)-1);
            if(numero_pasos == 0xFFFF){
                numero_pasos == 0x00;
            }
            else
                numero_pasos++;
        }
        else{ //movimientos izquierda
            stepper_state=(stepper_state-1)&(sizeof(paso)-1);
            if(numero_pasos == 0x0){
                numero_pasos == 0xFFFF;
            }
            else
                numero_pasos--;
        }
    }

    switch(stepper_state){
        case 0:
            output_high(A1);
            output_high(A2);
            output_low(B1);
            output_low(B2);;break;
        case 1:
            output_low(A1);
            output_high(A2);
            output_high(B1);
            output_low(B2);;break;
        case 2:
            output_low(A1);
            output_low(A2);
            output_high(B1);
            output_high(B2);;break;
        case 3:
            output_high(A1);

```

```

        output_low(A2);
        output_low(B1);
        output_high(B2);;break;
    }
    aux2 = numero_pasos;
    lcd_gotoxy(1,1);
    printf(lcd_putc, "Pos.actual=%ld", numero_pasos);
    //Escritura en la EEPROM posición actual
    guarda_eeprom(0x01, numero_pasos);
    guarda_eeprom(0x02, (numero_pasos>>8));
}
}

//SUBROUTINA QUE MUEVE EL DISPOSITIVO HASTA EL FNAL DE CARRERA DE LA
//DERECHA
void posicion_izquierda(){
    while(input(FINI)){
        paso_motor(5, 'L', 1);
    }
}

//SUBROUTINA QUE MUEVE EL DISPOSITIVO HASTA EL FNAL DE CARRERA DE LA
//IZQUIERDA
void posicion_derecha(){
    while(input(FIND)){
        paso_motor(5, 'R', 1);
    }
}

void test_lcd(void){
    int i;
    lcd_putc("\f");
    lcd_gotoxy(1,1);
    lcd_command(BLINK_ON);
    delay_ms(2000);
    lcd_command(BLINK_OFF);

    for(i=0;i<sizeof(test1);i++){
        lcd_gotoxy(i+1,1);
        delay_ms(50);
        printf(lcd_putc, "%c", test1[i]);
    }
    delay_ms(1000);
    for(i=0;i<sizeof(test2);i++){
        lcd_gotoxy(i+1,2);
        delay_ms(50);
        printf(lcd_putc, "%c", test2[i]);
    }
    delay_ms(1000);
    lcd_putc("\f");
    lcd_command(BLINK_ON);
}

//Subrutina de configuración del uC
void configuracion (void){

    //Puertos
    set_tris_a(0b01110111); //1 input; 0 output
    set_tris_b(0b00000000);
}

```

```

set_tris_c(0b00000000);

//Temporizadores y contadores
setup_counters(RTCC_INTERNAL,RTCC_DIV_256);
setup_timer_0(RTCC_INTERNAL);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
setup_wdt(WDT_OFF);

//Puertos ADC y comparadores
setup_adc_ports(AN0_TO_AN1_ANALOG|VSS_VDD);
setup_adc(ADC_CLOCK_DIV_64);
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);

//Comunicaciones
setup_spi(SPI_SS_DISABLED);

//Interrupciones
disable_interrupts(int_ext);
disable_interrupts(INT_RTCC);
disable_interrupts(GLOBAL); //deshabilita interrupciones
}

int leer_pulsadores(void){
    int var1;

    set_adc_channel(0); //Asi Captura el valor teniendo como referencia
Vcc y Gnd o sea de 0 a 5v
    delay_ms(10);
    var1 = read_adc();

    lcd_putc("\f");

    if(var1>200){
        return 2;
    }
    else
        if(var1>110){
            return 1;
        }
        else
            if(var1>70){
                return 4;
            }
            else
                if(var1>60){
                    return 3;
                }
                else
                    if(var1>45){
                        return 5;
                    }
                    else{
                        return 0;
                    }
}

```

```

    }

}

//Programa principal
void main(){
    int i;
    int pulsador;
    unsigned int16 var2;
    unsigned int16 resul,result1;

    configuracion();

    lcd_init();
    test_lcd();

    //Preguntamos si se encuentra conectado el USB mediante
    //la lectura de la entrada analógica

    set_adc_channel(1);
    delay_ms(1);
    var2 = read_adc();
    if ( var2 > 240){
        lcd_putc("\f");
        lcd_gotoxy(1,1);
        printf(lcd_putc,"USB conectado");
        lcd_gotoxy(1,2);
        printf(lcd_putc,"conecte PC");

        modo = 1;
        delay_ms(300);
        usb_cdc_init();
        usb_init();          //inicializamos el USB
        while(!usb_cdc_connected()) {}
        for(i=0;i<sizeof(test3);i++){          //imprimir texto3
            lcd_gotoxy(i+1,1);
            //delay_ms(50);
            printf(lcd_putc,"%c",test3[i]);
        }
    }
    else
        modo = 0;

    //Si se encuentra pulsado el pulsador 5, entramos en la calibración
    //del carril
    if(leer_pulsadores() == 5){
        //Nos desplazamos al lado izquierdo hasta alcanzar el fin de
//carril
        posicion_izquierda();
        //Inicialización de variables
        guarda_eeprom(0x01,0x00);
        guarda_eeprom(0x02,0x00);
        for(i=0;i<10;i++){

            paso_motor(5, 'R', 250);
        }

        posicion_actual();

```

```

}
else
    posicion_actual();

//Inicialización de puertos
output_high(PIN_C0);

delay_ms(1000);
lcd_putc("\f");
lcd_command(BLINK_ON);

while(TRUE){

    pulsador = leer_pulsadores();

set_adc_channel(1);
delay_ms(1);
var2 = read_adc();
if (var2 < 240){
    modo = 0;
}
else{
    if(modo == 0){
        lcd_putc("\f");
        lcd_gotoxy(1,1);
        printf(lcd_putc,"USB conectado");
        lcd_gotoxy(1,2);
        printf(lcd_putc,"conecte PC");

        modo = 1;//envio de datos al PC desde el micro
        delay_ms(300);
        usb_cdc_init();
        usb_init();          //inicializamos el USB
        while(!usb_cdc_connected()) {}
        for(i=0;i<sizeof(test3);i++){          //imprimir texto3
            lcd_gotoxy(i+1,1);
            //delay_ms(50);
            printf(lcd_putc,"%c",test3[i]);
        }
    }
}

if((usb_cdc_kbhit()) && (modo=1)){
    //recepción de datos en el micro enviados desde el PC
    tecla = usb_cdc_getc();
    printf(usb_cdc_putc, "TECLA;%d",tecla);

    if(tecla== -38){
        paso_motor(5, 'R', 1);
    }
    else
        paso_motor(5, 'L', 1);

}
lcd_putc("\f");

```

```

lcd_gotoxy(1,1);

switch(pulsador){
  case 1:
    if(pos==65535)
      pos = 65535;
    else
      pos++;
      delay_ms(125);
    lcd_putc("\f");
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Pos.actual:%lu", aux2);
    if(modo==1)
      printf(usb_cdc_putc,"Pos.actual:%lu\n\r ",aux2);
    lcd_gotoxy(1,2);
    printf(lcd_putc," Pos.nueva:%lu", pos);
    if(modo==1)
      printf(usb_cdc_putc,"Pos.nueva:%lu\n\r ",pos);
    delay_ms(25);
  break;
  case 2:
    lcd_gotoxy(1,2);
    printf(lcd_putc,"pulsa izquierda");
    if(modo==1)
      printf(usb_cdc_putc,"pulsa izquierda ");
      //si no hemos llegado a la izquierda del todo movemos
      //el motor a la izda
    if((input(FINI)) && (aux2!=0)){
      paso_motor(5, 'L', 1);
      pos=aux2;
      //Indicar al PC que se ha movido un paso a la izquierda

      //printf(usb_cdc_putc,"PASO=%ld\n\r",numero_pasos);
      delay_ms(150);
    }
    //fin de carrera Izda
    else{
    //final carril izquierdo
      lcd_gotoxy(1,2);
      printf(lcd_putc,"final carril izq");
      if(modo==1)
        printf(usb_cdc_putc,"final carril izq\n\r");
      //delay_ms(250);
    }

    delay_ms(150);
    break;
  case 3:
    lcd_gotoxy(1,2);
    printf(lcd_putc,"pulsa derecha");
    if(modo==1)
      printf(usb_cdc_putc,"pulsa derecha ");
      //movemos el motor a la derecha si no estamos en la
      //posición final
    if((input(FIND)) && (aux2 != 65535)){
      paso_motor(5, 'R', 1);
      pos=aux2;
      //printf(usb_cdc_putc,"PASO=%ld\n\r",numero_pasos);
      delay_ms(150);
    }
}

```

```

else{
//final carril derecho
    lcd_gotoxy(1,2);
    printf(lcd_putc,"final carril dch");
    //printf(usb_cdc_putc,"final carril dch\n\r");
    //delay_ms(250);
}

delay_ms(150);
//Indicar al PC que se ha movido un paso a la derecha
break;
case 4:
    if(pos==0)
        pos = 0;
    else
        pos--;
        delay_ms(125);
    lcd_putc("\f");
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Pos.actual:%lu", aux2);
    if(modos==1)
        printf(usb_cdc_putc,"Pos.actual:%lu\n\r ",aux2);
    lcd_gotoxy(1,2);
    printf(lcd_putc," Pos.nueva:%lu", pos);
    if(modos==1)
        printf(usb_cdc_putc,"Pos.nueva:%lu\n\r ",pos);
    delay_ms(25);
break;
case 5:
    lcd_putc("\f");
    //lcd_gotoxy(1,1);
    //printf(lcd_putc,"Pos.actual: %lu", aux2);

//en aux2 está la posición actual
//en pos está la posición destino
//la posición actual es menor que posición destino, hay que
//realizar el movimiento a la derecha

// dejamos la correa tensa para movimientos a derechas

    if(aux2<pos){
        //la posición derecha del todo es 5200.
//podemos acercarnos todo lo queramos al extremo derecho sin ningún
// tipo de correccion
        if((input(FIND)) && (pos < 5200)){

            resul=(pos-aux2);
            resul1=resul;
            resul=resul/250;
            resul1=resul1%250;

            paso_motor(5, 'R', resul1);

            for(i=0;i<resul;i++){
                paso_motor(5, 'R', 250);
            }

```

```

        pos=aux2;
    }
}
//no es un movimiento a la derecha
else{
//movimiento a la izda
    if(aux2>pos){
//si la posición destino está, por lo menos, a más de
//80 posiciones del principio del recorrido
//añadimos 80 pasos en el mismo sentido para después moverlos
//en el otro
        if((input(FINI)) && (pos>80)){

            resul=(aux2-pos)+80;
            resul1=resul;
            resul=resul/250;
            resul1=resul1%250;

            paso_motor(5, 'L', resul1);
            for(i=0;i<resul;i++){
                paso_motor(5, 'L', 250);
            }
            paso_motor(5, 'R', 80);
            pos=aux2;

        }
//si la posición destino está dentro de las 80 primeras
//posiciones, nos ponemos a la izquierda del todo
// y avanzamos hasta la posición solicitada
        else{

            posicion_izquierda();
            paso_motor(5, 'R', pos);

            pos=aux2;

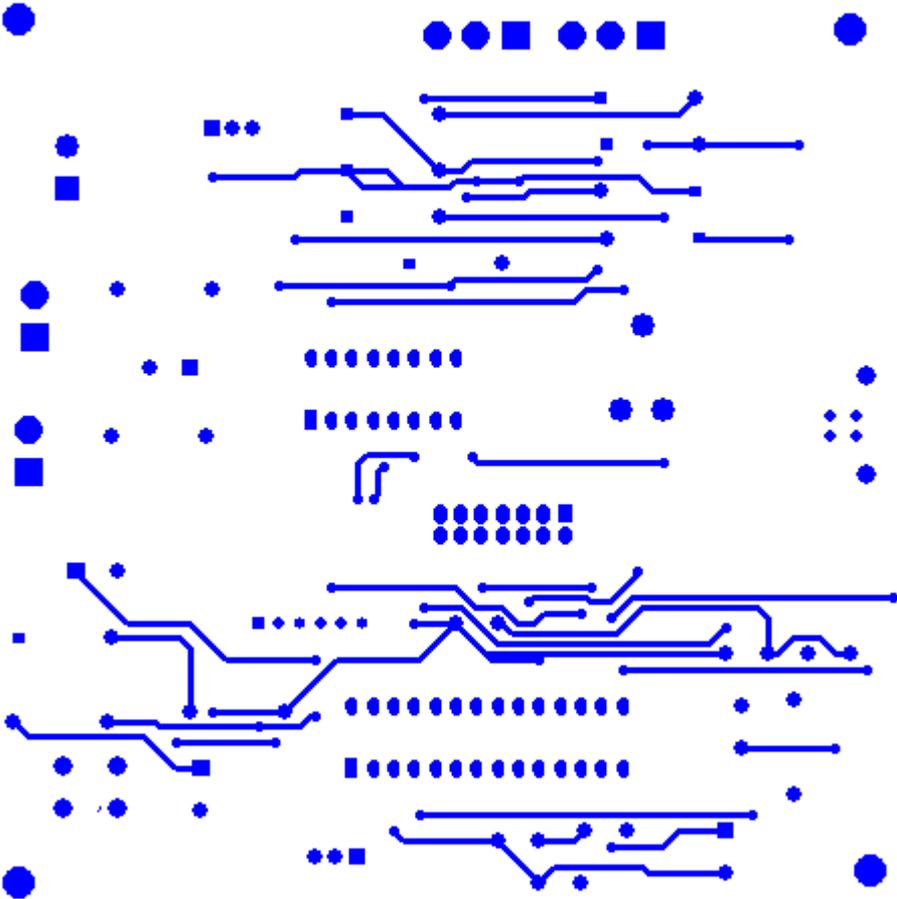
        }
    }
//si la posición destino es la misma en la que está
//actualmente, no movemos el motor
else{
    lcd_gotoxy(1,2);
    printf(lcd_putc,"OK");
}
}

break;
//por defecto mostramos la posición actual
default:
    lcd_putc("\f");
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Pos.actual: %lu", aux2);
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Pulse");
break;
}
}
}

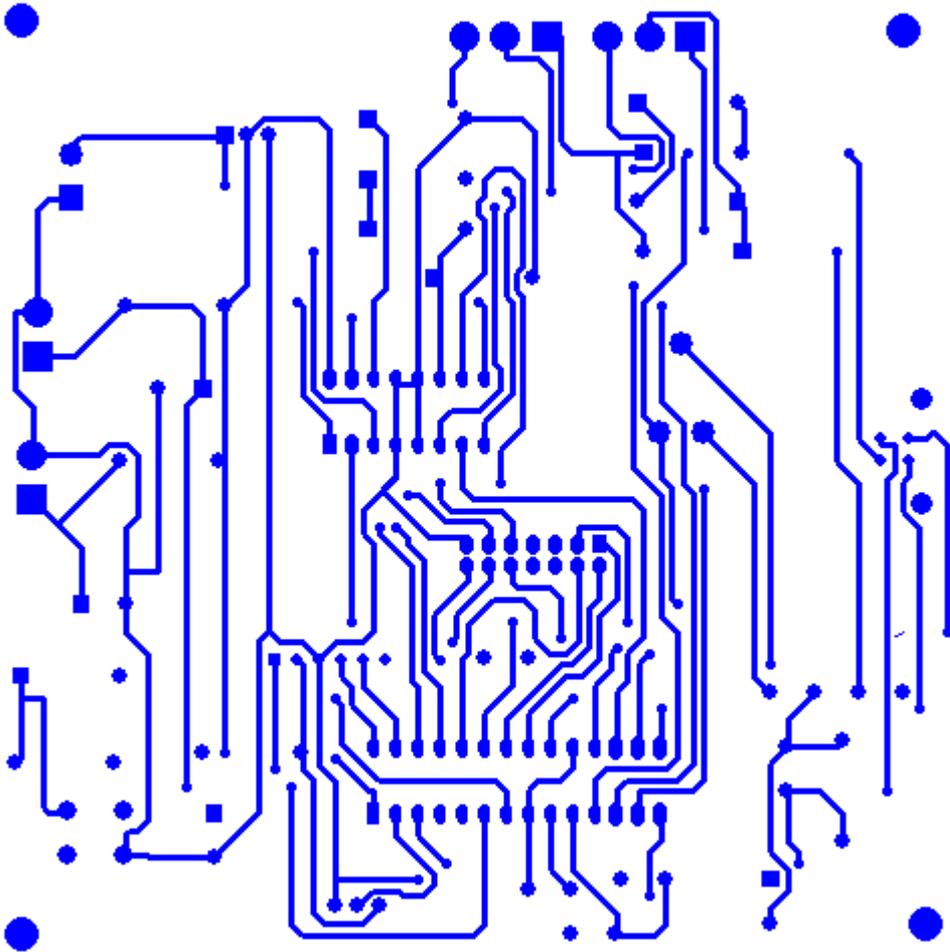
```


PLANOS

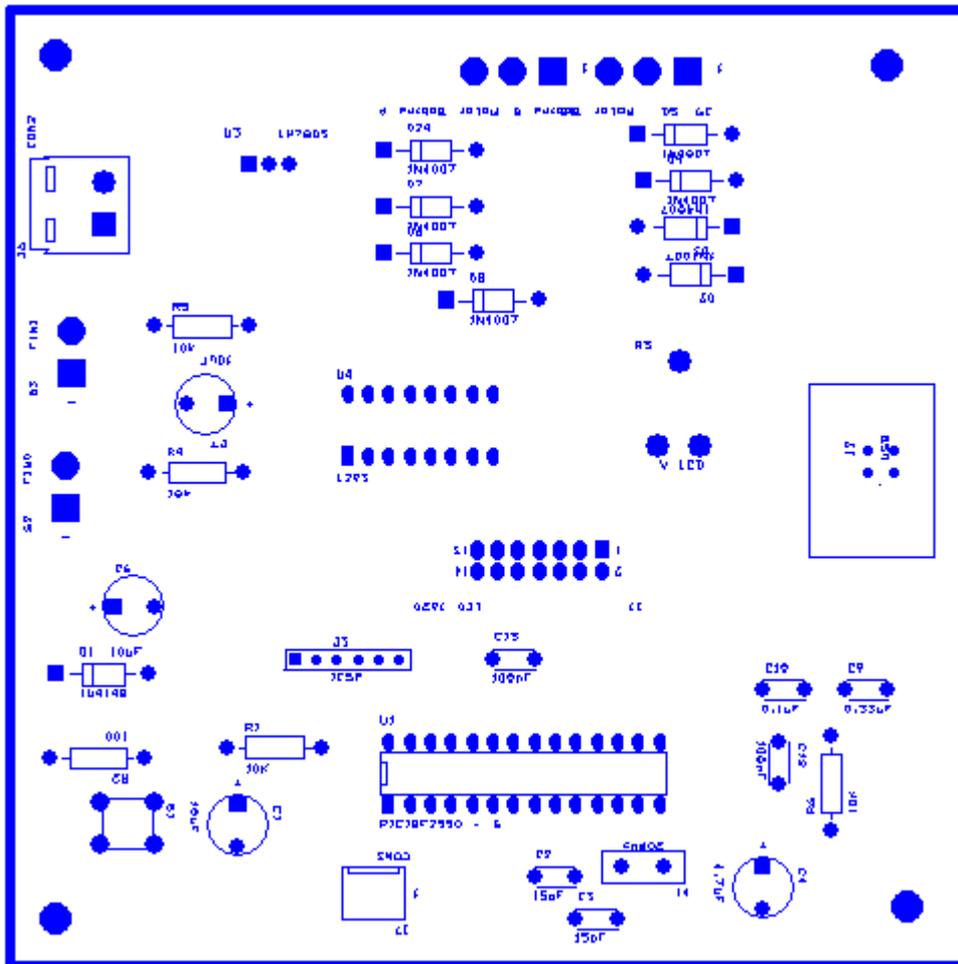
PLANO 1: CARA TOP PLACA DEL MOTOR



PLANO 2: CARA BOTTOM PLACA DEL MOTOR

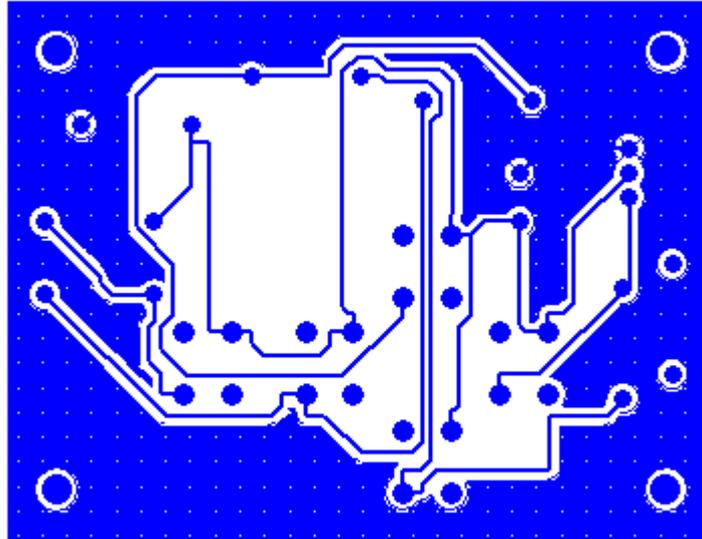


PLANO 3: CARA COMPONENTES PLACA DEL MOTOR

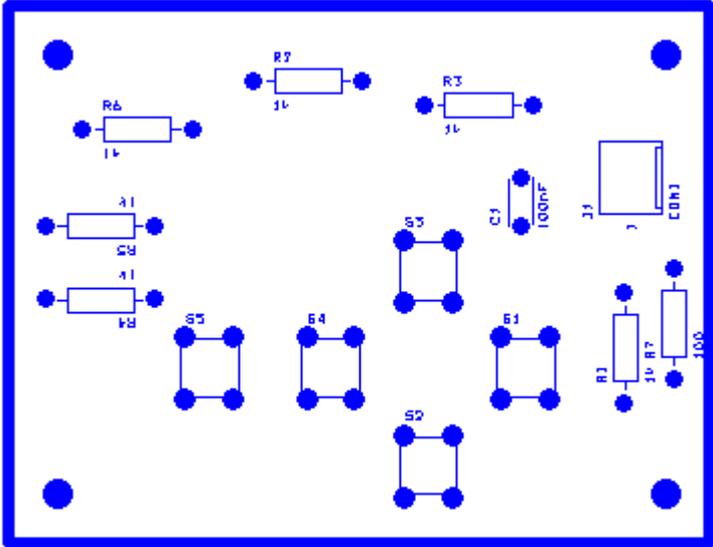


Dimensiones del PCB: 120x120mm

PLANO 4: CARA BOTTOM PLACA DE PULSADORES



PLANO 5: CARA COMPONENTES PLACA DE PULSADORES



Dimensiones del PCB: 80x60mm

PLIEGO DE CONDICIONES

HARDWARE

Los dispositivos que se han utilizado para desarrollar este proyecto son los siguientes:

- Fuente de alimentación de corriente continua, TEKTRONIX EX752M MULTI-MODE PSU 75V/50V 300W
- Osciloscopio TEKTRONIX TDS 2010 Digital storage Osciloscop.
- Programador MPLAB
- Ordenador personal

SOFTWARE

El software utilizado en el desarrollo de este Proyecto ha sido:

- MPLAB
- CCS
- ORCAD Capture V10.0 para desarrollar el circuito electrónico.
- ORCAD Layout Plus para el diseño del circuito impreso PCB.

PRESUPUESTO

1.1 INTRODUCCIÓN

En este presupuesto se muestra una estimación de los gastos que ha conllevado la realización de este proyecto, quedando reflejados los costes del material tanto electrónico como mecánico, así como los relativos al personal que ha participado en el desarrollo.

1.2 COSTE DEL MATERIAL ELECTRÓNICO Y MECÁNICO

Estos costes son los referentes a las componentes y materiales utilizados para el desarrollo de las placas de circuito impreso y del conjunto vagoneta-carril para la construcción de un microposicionador.

Material electrónico:

Componente	Unidades	Precio Unitario	Precio Total
PIC18F2550	1	4,81 €	4,81 €
L293D	1	3,43 €	3,43 €
LM7805	1	0,61 €	0,61 €
162D	1	6,81 €	6,81 €
USB	1	0,75 €	0,75 €
XTAL 20MHz	1	0,54 €	0,54 €
Zócalo 28vías	1	0,45 €	0,45 €
Zócalo 16 vías	1	0,45 €	0,45 €
10uF	3	0,08 €	0,24 €
15pF	2	0,35 €	0,70 €
4,7uF	1	0,13 €	0,13 €
0,33uF	1	0,08 €	0,08 €
100nF	4	0,12 €	0,48 €
100Ω (1/4w)	1	0,07 €	0,07 €
1KΩ (1/4w)	6	0,07 €	0,42 €
10kΩ (1/4w)	3	0,07 €	0,21 €
1N4148	1	0,07 €	0,07 €
1N4007	8	0,07 €	0,56 €
PULSADORES	6	0,05 €	0,30 €
POTENCIOMETRO	1	0,05 €	0,05 €
Conector Vcc PCB	6	0,39 €	2,34 €
Placa PCB	2	50 €	100 €
Caja	1	6,71 €	6,71 €
		TOTAL:	130,21 €

Material mecánico:

Componente	Unidades	Precio Unitario	Precio Total
Vagoneta	1	24,80 €	24,80 €
Carril	1	56,50 €	56,50 €
Detectores fin de carril	2	3,25 €	6,50 €
Planchas metálicas	2	6,49 €	12,98 €
TOTAL:			100,78 €

1.3 COSTE DEL PERSONAL INTERVINIENTE

Los costes que se van a detallar a continuación comprenden los costes desprendidos de las etapas de diseño y desarrollo, los costes de montaje y pruebas (desarrolladas por un Técnico) y los costes de la redacción de la documentación relativa al proyecto.

Para la cuantificación del coste se ha considerado que el precio/hora del ingeniero junior es de 14 €/hora, considerando que no dispone de experiencia en el desarrollo de proyectos.

El número de horas invertidas en las tareas anteriores son:

Concepto	Número de Horas	Coste/Hora	Total
Diseño y desarrollo	20 h	14 €	280 €
Montaje y pruebas del hardware	20 h	14 €	280 €
Desarrollo y depuración del firmware del microcontrolador	50 h	14 €	700 €
Desarrollo del software para PC	100 h	14 €	1.400 €
TOTAL:			2.660 €

1.4 PRESUPUESTO TOTAL

El presupuesto de ejecución material de este proyecto asciende a:

Concepto	Coste (€)
Costes de material	230,99 €
Costes de personal	2.660,00 €
TOTAL:	2.890,99 €
Precio de venta*	1.118,25 €

(IVA incluido)

Asciende el presupuesto de ejecución material de este proyecto a la cantidad de DOS MIL OCHOCIENTOS NOVENTA EUROS CON NOVENTA Y NUEVE CÉNTIMOS. En esta cantidad está incluido el IVA exigido por Ley, con su valor actualizado.

* Nota:

Para establecer un posible precio de venta del microposiconador elaborado, se ha considerado el precio de los materiales utilizados (electrónicos y mecánicos) y se ha considerado el 10 % de los costes de personal. Por tanto si se fabricasen 10 equipos, se reparte el coste de personal entre estos 10 equipos para que sea rentable. A esta cantidad se le ha añadido un 50% de beneficio para el fabricante y al precio resultante se le ha añadido un 50% de beneficio para el comerciante.

Costes de material=	230,99 Euros
Repercusión costes de personal: 10% sobre 2660=	266,00 Euros
TOTAL =	496,99 Euros

Beneficio del fabricante: 50% sobre 496,99 =	248,50 Euros
Precio de suministro al comerciante =	745,50 Euros

Beneficio del comerciante: 50% sobre 745,50 =	372,75 Euros
Total precio de venta al público =	1.118,25 Euros

Por tanto el precio de venta estimado al público asciende a 1.118,25 € por unidad. Esta cantidad es la que aparece reflejada en la Tabla 7.1 en la que se ha comparado el prototipo diseñado con uno existente en el mercado