



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**FACULTAD DE INGENIERÍA**



## **“Diseño de un sistema de control de intensidad de laser”**

**Tesis**

**Que presentan**

**Orozco Arias Jesús**

**Segoviano Plata Jorge Alberto**

Para obtener el título de

**INGENIERO ELÉCTRICO-ELECTRÓNICO**

Director de Tesis

M.I. Miguel Ángel Bañuelos Saucedo

México D.F, septiembre 2010



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## Índice

<b>1</b>	<b>Introducción</b> .....	6
1.1	Definición del problema .....	6
1.2	Objetivo .....	7
<b>2</b>	<b>El microcontrolador (PIC)</b> .....	8
2.1	¿Qué es el microcontrolador? .....	8
2.2	El procesador .....	9
2.3	Memoria.....	9
2.4	Puertos de entrada y salida.....	9
2.5	Reloj principal.....	10
2.6	Recursos especiales .....	10-11
2.7	El PIC18f452 .....	12
<b>3</b>	<b>Implementación del sistema</b> .....	13
3.1	Funcionamiento general del sistema.....	13-16
3.2	Generación de la señal PWM.....	17
3.3	Lectura del teclado .....	17-24
3.4	Despliegue del ciclo de trabajo.....	25
3.5	Monitoreo de la señal PWM.....	26-27
3.6	Control del ciclo de trabajo mediante la PC.....	28-31
3.7	Salida PWM.....	32
3.8	Diseño exterior del sistema .....	33-34
3.9	Pruebas realizadas al sistema.....	35-38
3.10	Diseño interior del sistema .....	39-46
3.11	Lista de precios y componentes del sistema .....	46-47
3.12	Programación del PIC .....	48

<b>4 Conclusiones</b> .....	49
<b>5 Apéndice</b> .....	50
5.1 Manual de usuario.....	51-55
5.2 Código fuente del PIC .....	56-62
<b>6 Bibliografía</b> .....	63

### Índice de figuras

Figura 1. Esquema de la arquitectura Von Neumann. ....	8
Figura 2. Esquema de la arquitectura Harvard.....	9
Figura 3. Esquema general del funcionamiento del sistema. ....	13
Figura 4. Diagrama de flujo general del sistema. ....	15
Figura 5. Teclado matricial de 4x4 de membrana.....	18
Figura 6. Teclado matricial 4x4 tipo telefónico.....	18
Figura 7. Push button de cuatro terminales tipo SPST.....	19
Figura 8. Diseño del teclado matricial.....	19
Figura 9. Teclado matricial vista preliminar. ....	19
Figura 10. Vista del teclado matricial que estará en el sistema. ....	20
Figura 11. Circuito que crea un sonido al presionar una tecla.....	21
Figura 12. Teclados fabricados. ....	22
Figura 13. Circuito para el despliegue del ciclo de trabajo. ....	25
Figura 14. Variación de amplitud respecto al ciclo de trabajo.....	26
Figura 15. Filtro RC. ....	26
Figura 16. Diagrama del proceso de monitoreo.....	27
Figura 17. Circuito para realizar el monitoreo.....	27
Figura 18. Vista del teclado virtual.....	30
Figura 19. Código generado para la interfaz con el puerto paralelo. ....	31
Figura 20. Circuito acondicionador de señal.....	32
Figura 21. Gabinete de plástico.....	33
Figura 22. Vista frontal y vista posterior del sistema 1° versión.....	34
Figura 23. Versión definitiva. ....	34

Figura 24. Ciclo de trabajo mostrado en el sistema.....	35
Figura 25. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 0.02 %. Para esta señal el osciloscopio midió una frecuencia de 10 kHz pero no midió su ciclo de trabajo.....	35
Figura 26. Ciclo de trabajo mostrado en el sistema.....	36
Figura 27. Ciclo de trabajo máximo obtenido. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 99.8 %. Para esta señal el osciloscopio midió una frecuencia de 9.998 kHz y un ciclo de trabajo de 99.8 %. .....	36
Figura 28. Ciclo de trabajo mostrado en el sistema.....	37
Figura 29. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 15.7 %. Para esta señal el osciloscopio midió una frecuencia de 10.0006 kHz y un ciclo de trabajo de 15.74 $\mu$ s.....	37
Figura 30. Ciclo de trabajo mostrado en el sistema.....	38
Figura 31. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 50 %. Para esta señal el osciloscopio midió una frecuencia de 10.0006 kHz y un ciclo de trabajo de 50 $\mu$ s.....	38
Figura 32. Vista interior del sistema.....	39
Figura 33. Diagrama eléctrico del circuito impreso principal.....	40
Figura 34. Circuito impreso principal con representación de componentes electrónicos. ....	41
Figura 35. Diseño final del circuito impreso principal.....	41
Figura 36. Diagrama eléctrico de la fuente de alimentación.....	42
Figura 37. Teclado matricial. A) Distribución de teclas. B) Diagrama eléctrico. ....	42
Figura 38. A) Circuito impreso del teclado matricial con representación de componentes electrónicos. B) Diseño final del circuito impreso del teclado matricial.....	43
Figura 39. Diagrama eléctrico del módulo de displays. ....	43

Figura 40. A) Circuito impreso del módulo de displays con representación de componentes electrónicos. B) Diseño final del circuito impreso del módulo de displays. ....	43
Figura 41. Diagrama eléctrico del módulo de acondicionamiento de señal y generación de sonido.....	43
Figura 42. A) Circuito impreso del módulo de acondicionamiento de señal y generación de sonido con representación de componentes electrónicos. B) Diseño final del circuito impreso del módulo de acondicionamiento de señal y generación de sonido. ....	44
Figura 43. Distribución de pines del PIC18F452. ....	45
Figura 44. Vista posterior del módulo. ....	52
Figura 45. Vista frontal del módulo. ....	54
Figura 46 Teclado virtual. ....	55

## **1 Introducción**

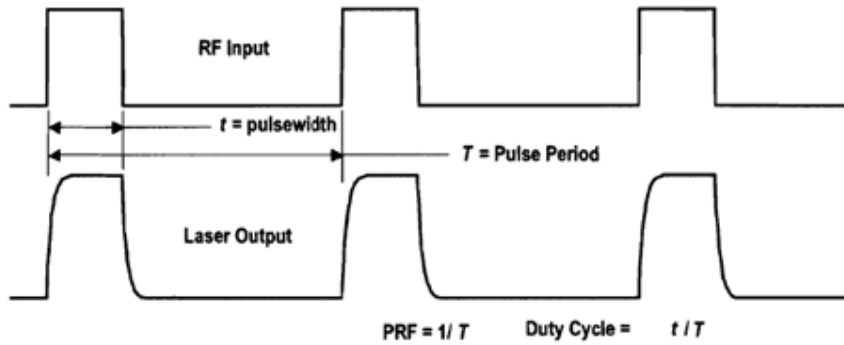
En el Instituto Nacional de Cardiología (INC) se diseñan y fabrican prótesis de válvulas cardíacas a partir de tejido animal, el cual consiste básicamente de válvulas porcinas y tejido pericardio bovino. En el caso del tejido pericardio, es necesario efectuar un corte del tejido, para darle la forma adecuada antes de fijarlo a un anillo metálico y agregarle un refuerzo de dacrón.

Actualmente el proceso de corte se realiza manualmente, sin embargo es deseable automatizarlo para estandarizar las características de la válvula. Para ello se desarrolló un sistema de corte basado en un láser GEM-30 de CO<sub>2</sub> de 30 W, del fabricante Coherent, una platina controlada por computadora y un control de intensidad del laser.

### **1.1 Definición del problema**

A solicitud del Instituto Nacional de Cardiología (INC), diferentes grupos del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET) participaron en el desarrollo de un sistema de corte láser para tejidos, el cual se utilizará para la fabricación de bioprótesis valvulares cardíacas.

Como primera etapa, se desarrolló un sistema para activar el láser GEM-30 enfriado por líquido del fabricante Coherent. El láser cuenta con dos modos básicos de operación: onda continua (Continuous Wave: CW) y onda pulsada (Gated Wave: GW). Una forma de variar la potencia entregada por el láser es utilizar la onda pulsada, la cual consiste en un tren de pulsos con ciclo de trabajo variable. Las restricciones para la señal de control son que no existan pulsos menores a 1  $\mu$ s y que la frecuencia (PRF: Pulse Repetition Frequency) no exceda los 25 kHz.



El sistema que realiza estas funciones está basado en el temporizador LM555, pero tenía un problema con el ajuste del ciclo de trabajo, ya que era muy inestable. Por esta razón, se decidió hacer una segunda versión en la cual se pudiera tener un mejor control y una mejor estabilidad en el ciclo de trabajo.

## 1.2 Objetivo

Diseñar un sistema de control de intensidad para un láser utilizado en la fabricación de bioprótesis cardiacas, que sustituya al ya existente.



## 2 EL microcontrolador

### 2.1 ¿Qué es un microcontrolador?

Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos para desempeñar una tarea específica. Dicha tarea debe ser programada por el usuario a través de un lenguaje de programación.

Dentro de los microcontroladores existen dos arquitecturas básicas de hardware.

#### a) Von Neumann

Se caracteriza por tener una memoria única para los datos y las instrucciones del programa. A dicha memoria se accede a través de un sistema de buses único (control, direcciones y datos).

### ARQUITECTURA VON NEUMANN

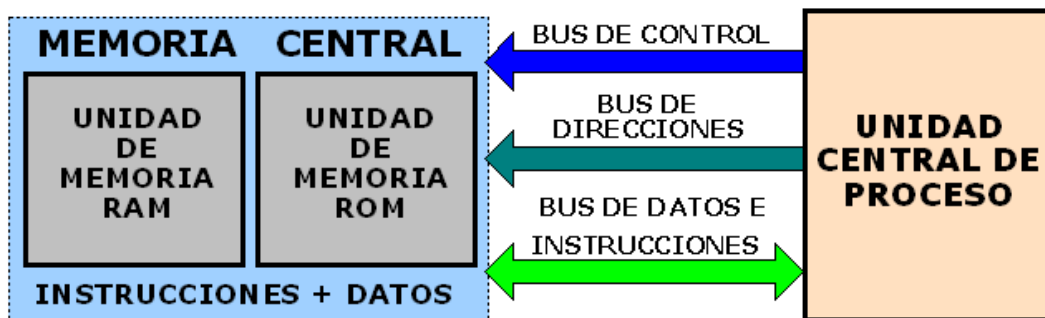


Figura 1. Esquema de la arquitectura Von Neumann.

#### b) Harvard

Este modelo tiene la unidad central de proceso (CPU) conectada a dos memorias, una con las instrucciones y otra con los datos, por medio de dos buses diferentes. Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y la otra sólo almacena datos (Memoria de Datos).

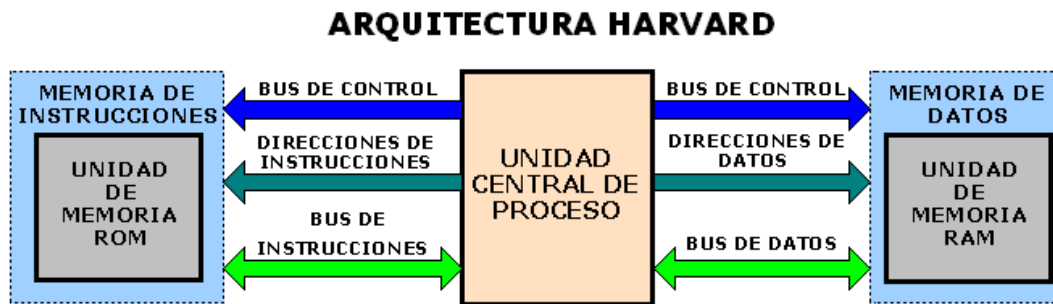


Figura 2. Esquema de la arquitectura Harvard.

## 2.2 El procesador

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software. Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

## 2.3 Memoria

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

## 2.4 Puertos de Entrada y Salida

Los puertos de Entrada y Salida (E/S) permiten comunicar al procesador con el mundo exterior, a través de interfaces, o con otros dispositivos. Estos puertos son la principal utilidad de las patas o pines de un microprocesador.

## 2.5 Reloj principal

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Esta señal del reloj es el motor del sistema y la que hace que el programa y los contadores avancen.

## 2.6 Recursos especiales

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software. Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o **Timers**. Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).
- Perro guardián o **Watchdog**. Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicia el sistema.
- Protección ante fallo de alimentación o **Brownout**. Se trata de un circuito que reinicia al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo (brownout). Esto es muy útil para evitar datos erróneos por transiciones y ruidos en la línea de alimentación.
- Estado de reposo o de bajo consumo (**Sleep mode**). Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en

funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos.

- Conversor A/D (Analógico ->Digital). Los microcontroladores que incorporan un Convertidor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones.
- Conversor D/A (Digital ->Analógico). Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patillas del chip.
- Comparador analógico. Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.
- Modulador de anchura de impulsos o PWM (**Pulse Wide Modulation**). Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.
- Puertos de E/S digitales. Todos los microcontroladores destinan parte de su patillaje a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos. Las líneas digitales de los puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

## 2.7 EI PIC18F452

El Pic18F452 es un microcontrolador de arquitectura Harvard fabricado por Microchip. Algunas de sus características son las siguientes:

- a) La arquitectura de este microcontrolador está optimizada para el uso de compiladores de lenguaje C.
- b) Puede realizar hasta 10 MIPs (millones de instrucciones por segundo).
- c) Puede trabajar con un reloj de hasta 40 MHz activando el PLL.

Dentro de sus características cuenta con:

- d) Niveles de prioridad para el manejo de interrupciones.
- e) Pines con salida de alta corriente 25mA.
- f) 3 pines para el uso de interrupciones externas.
- g) 4 temporizadores.
- h) 2 módulos de captura/comparación/PWM con resolución de 10 bits si se trabaja hasta 39 kHz y resolución de 8 bits si se trabaja hasta 139 kHz.
- i) Convertidor analógico/digital de 10 bits de resolución.
- j) Memoria flash de 32k y memoria EEPROM 256 bytes.
- k) Watchdog.

### 3 Implementación del sistema

#### 3.1 Funcionamiento general del sistema

Para que el laser GEM-30 pueda funcionar es necesario cumplir ciertas características que fueron mencionadas anteriormente, ante esto decidimos estar dentro de estas especificaciones por lo que elegimos las siguientes características:

- Señal de salida: PWM
- Amplitud de la señal: 5V
- Frecuencia de la señal: 10 kHz
- Variación del ciclo de trabajo: 0% a 99.9%
- Resolución del ciclo de trabajo: 0.1%
- Ingreso de datos por teclado
- Ingreso de datos por PC (vía el puerto paralelo)
- Despliegue del ciclo de trabajo por medio de displays de 7 segmentos
- Monitoreo de la señal de salida

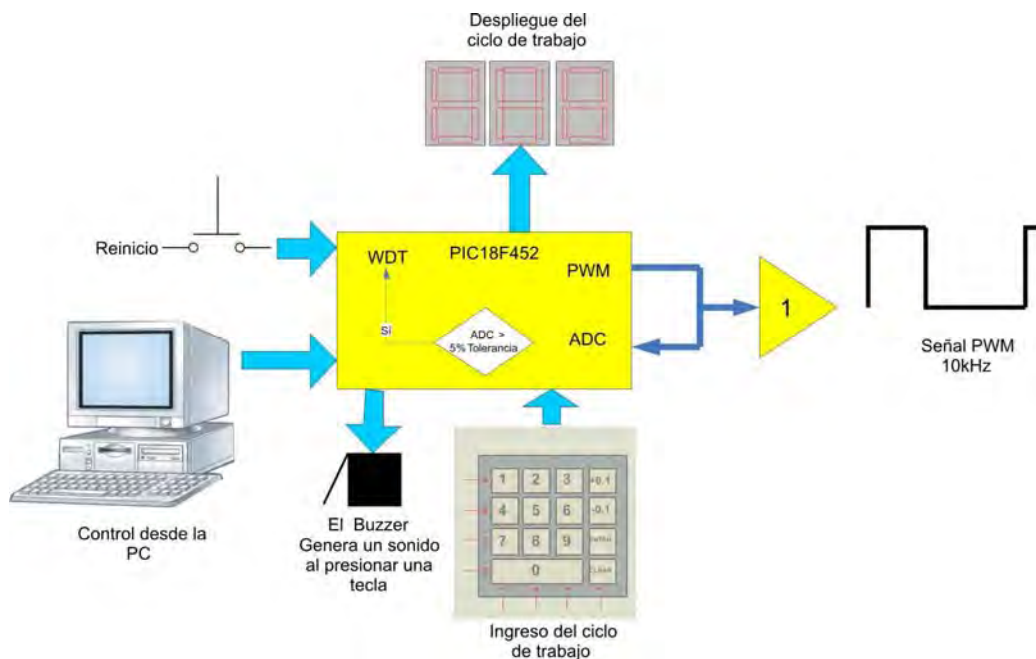


Figura 3. Esquema general del funcionamiento del sistema.

La frecuencia se eligió de tal manera que estuviera dentro de los requisitos del laser GEM-30, el fabricante sugiere que no se excedan los 25 kHz.

Algunas de las características que se eligieron fueron en cierta medida para la comodidad del usuario lo que es el caso del despliegue de datos por medio de displays de 7 segmentos, el teclado y el ciclo de trabajo variable.

En el caso del despliegue se decidió así debido a la facilidad visual que es mucho mayor comparándola con la visibilidad que tiene un LCD.

En el caso del teclado se eligió así debido a que para el usuario es más cómodo introducir los dígitos del ciclo de trabajo deseado ya que asemeja el manejo de un teléfono.

En el caso del ciclo de trabajo variable se propuso una resolución del 0.1% con el fin de obtener precisión en el control de la potencia del láser.

El monitoreo de la señal es para evitar que la salida del sistema sea diferente a lo que muestra en el display, logrando tener un control sobre esta.

El control del sistema por PC es para tener mayor seguridad, debido a que el usuario puede controlar el sistema desde un lugar lejano al sistema por medio de la PC.

El funcionamiento del sistema se puede consultar en el manual de usuario, por el momento se puede resumir con el diagrama de flujo siguiente:

## Diagrama de flujo del funcionamiento general del sistema

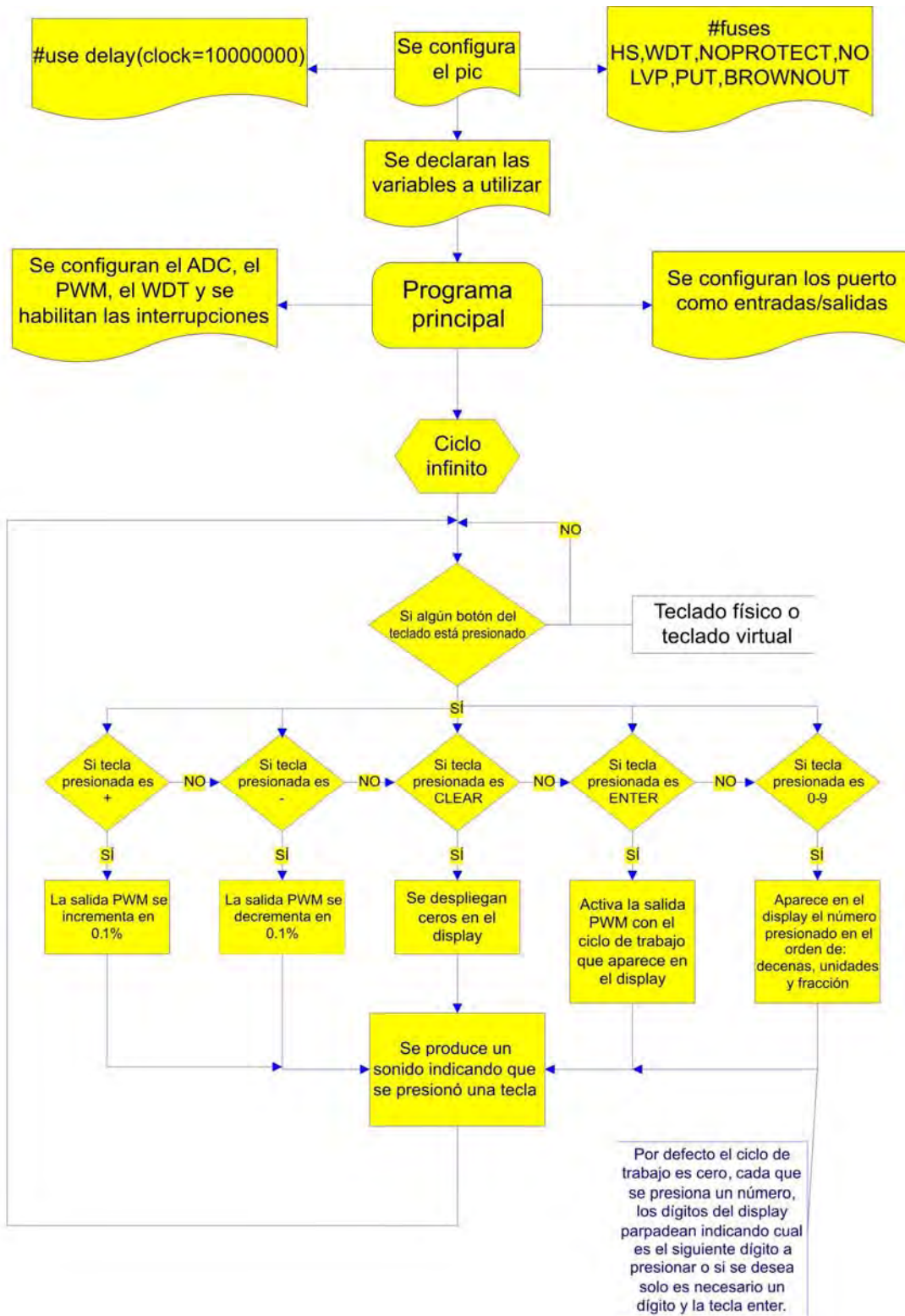


Figura 4. Diagrama de flujo general del sistema.



El módulo de inicio comenzará desplegando ceros en los displays, esperando a que el usuario introduzca el ciclo de trabajo deseado, una vez que esto sucede el programa revisa por cuál de los dos teclados fue ingresado el ciclo de trabajo (teclado físico o virtual), además de identificar cuál fue el número que ingresó el usuario e ir mostrando en el display el número decodificado.

Tiene otras teclas que no son números las cuales son +0.1,-0.1, y *clear*, cuyas funciones son principalmente incrementar y decrementar el ciclo de trabajo o en caso de equivocarse al ingresar el ciclo de trabajo la tecla *clear* que coloca ceros en los displays.

Una vez que el usuario decidió cuál será el ciclo de trabajo deseado, el módulo estará en espera de que el usuario oprima la tecla enter, cuando esto ocurre se calcula cuál fue el ciclo de trabajo deseado para finalmente activar la salida PWM.

### 3.2 Generación de la señal PWM

El PIC18F452 cuenta con dos módulos de captura/comparación/PWM. En el modo de modulación por ancho de pulsos (PWM), el pin CCP1 genera una salida PWM con una resolución de hasta 10 bits. El periodo de la señal PWM se especifica en el registro PR2 y puede ser calculado utilizando la siguiente fórmula:

$$\text{Periodo PWM} = [(PR2) + 1] * 4 * TOSC * (\text{valor de prescalador TMR2})$$

La frecuencia PWM se define como  $1 / [\text{periodo PWM}]$

El ciclo de trabajo de la señal PWM se especifica en el registro CCPR1L y en los bits 4-5 del registro CCP1CON. La siguiente fórmula es utilizada para calcular el ciclo de trabajo:

$$\text{Ciclo de trabajo PWM} = (\text{CCPR1L:CCP1CON}_{<5-4}) * TOSC * (\text{valor de prescalador TMR2})$$

A partir de estas fórmulas calculamos los parámetros para obtener una señal PWM de 10 kHz de frecuencia, y los parámetros para tener cambios de 0.1% del ciclo de trabajo.

### 3.3 Lectura del teclado

Para realizar la lectura de los datos introducidos por el usuario, decidimos utilizar un teclado matricial, debido a la facilidad visual y facilidad de manejo.

Dentro del mercado se encuentra una gran variedad de teclados matriciales pero nos teníamos que adaptar tanto a la codificación que ya traen integrada como al tamaño de los teclados y la distribución de las teclas. Es por ello que decidimos diseñar un teclado para satisfacer nuestras necesidades tanto de codificación, distribución y tamaño.

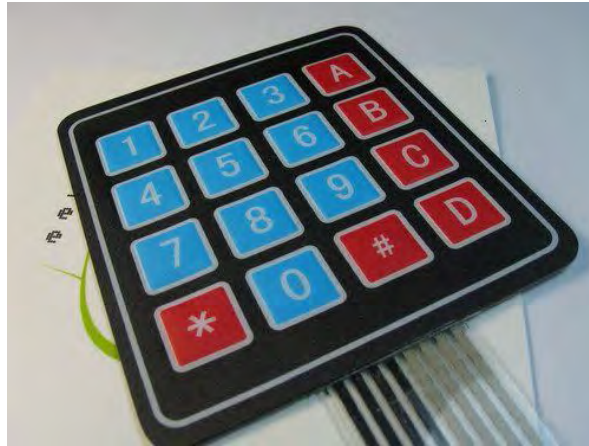


Figura 5. Teclado matricial de 4x4 de membrana.

Como un ejemplo de esto tenemos el teclado que se muestra en la figura 5, podemos observar que el tamaño, la distribución de las teclas y la forma de codificación es fijo, por otra parte podemos mencionar que son de calidad muy pobre ya que el contacto de las teclas no es del todo bueno, además de tener un costo relativamente elevado.



Figura 6. Teclado matricial 4x4 tipo telefónico.

Este es otro teclado que aunque la calidad mejora notoriamente seguíamos teniendo el problema del tamaño, la distribución de las teclas y la codificación, además de que el costo de este tipo de teclado es elevado lo que le añade otra desventaja.

A continuación les mostraremos cómo se realizó el teclado que diseñamos. El principio de funcionamiento es muy sencillo, utilizamos los switches del tipo DPST como se ve en la figura.

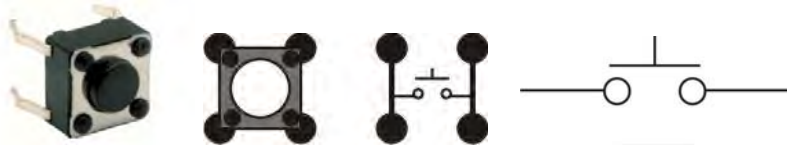


Figura 7. Push button de cuatro terminales tipo SPST.

Elegimos este tipo de switch por que tiene grandes ventajas tanto en tamaño, costo, además de su calidad y vida útil, la cual el fabricante garantiza unas 100,000 pulsaciones.

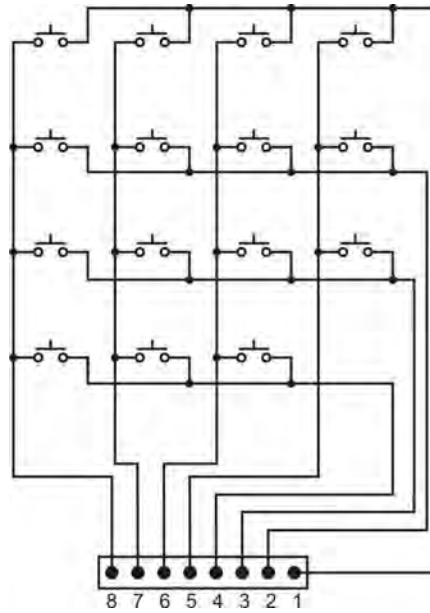


Figura 8. Diseño del teclado matricial.



Figura 9. Teclado matricial vista preliminar.

Con este tipo de teclado solucionamos los problemas mencionados anteriormente debido a que podemos modificar el tamaño, la distribución de las teclas y el tipo de código a utilizar, además de añadirle algunas ventajas como es el costo y la durabilidad del teclado.

Haciendo esto finalmente tenemos la vista definitiva que queremos para el módulo la cual es la siguiente:

El teclado cuenta con 16 teclas de las cuales 12 se utilizaron para los números y 4 para funciones especiales.

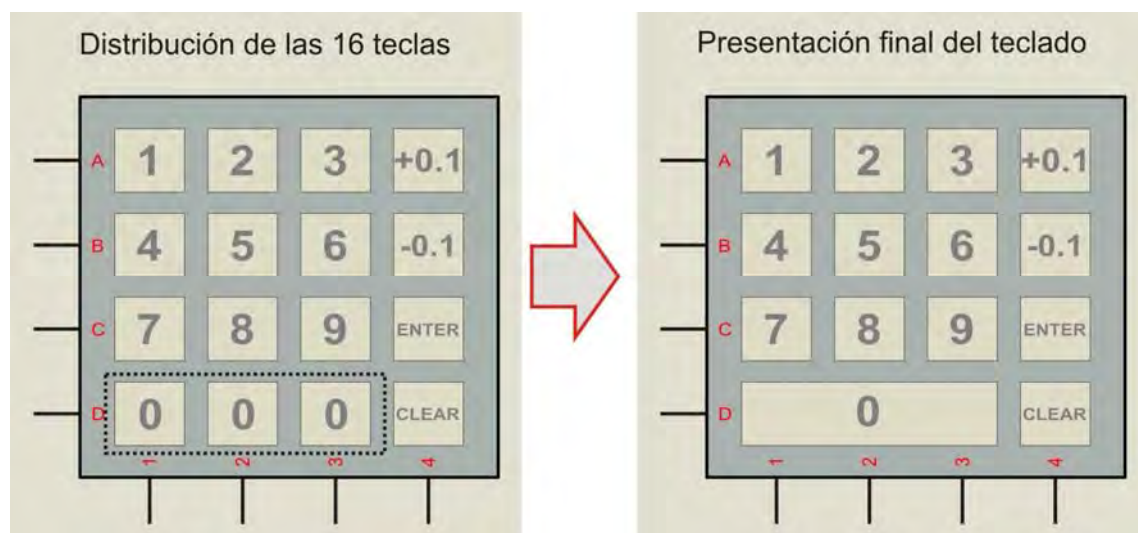


Figura 10. Vista del teclado matricial que estará en el sistema.

A partir de este diseño fabricamos tres teclados diferentes para probarlos físicamente y verificar si cumplían con nuestros requisitos. Durante las pruebas nos dimos cuenta de que sería conveniente tener una señal audible que se activara cuando se presionaba alguna tecla, por lo que agregamos un buzzer al módulo del teclado.

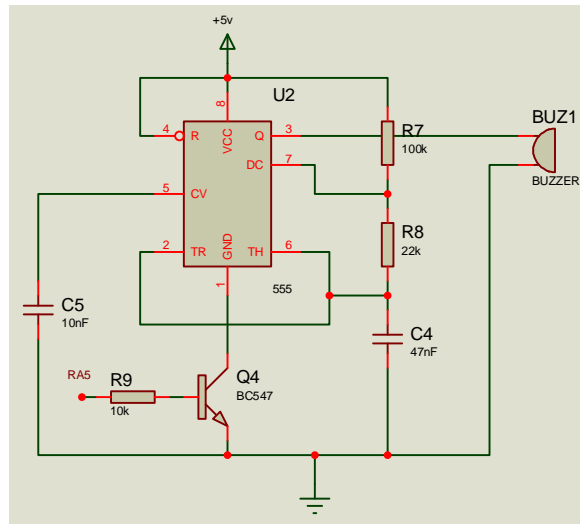


Figura 11. Circuito que crea un sonido al presionar una tecla.

El funcionamiento del circuito de la figura 11 es el siguiente: Cuando el microcontrolador detecta que una de las teclas ha sido presionada éste envía una señal por el pin RA5 la cual sirve para activar un circuito integrado LM555 cuya función es generar una frecuencia que es enviada al Buzzer.

El circuito integrado LM555 opera en el modo de multivibrador astable cuya frecuencia se calcula de la siguiente manera:

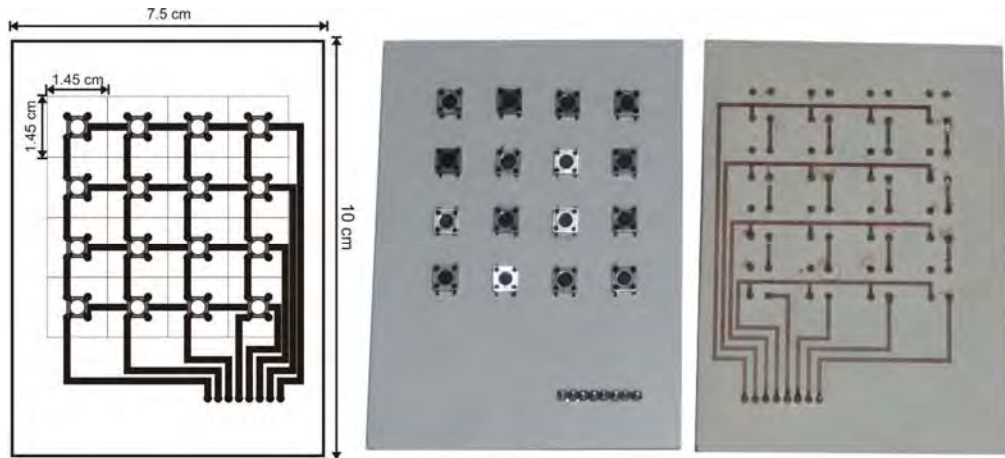
$$f = \frac{1}{0.693 \cdot C \cdot (R1 + 2R2)}$$

El ciclo de trabajo puede calcularse con la siguiente fórmula:

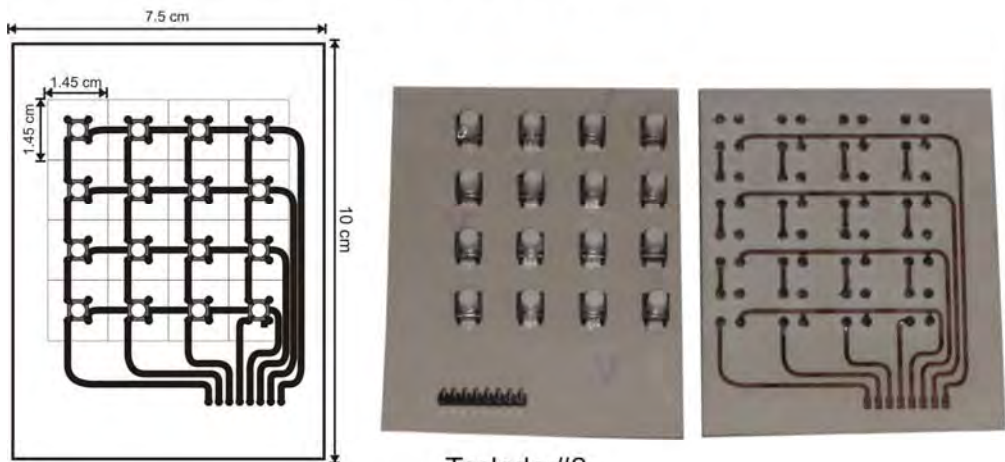
$$D = \frac{R1 + R2}{R1 + 2R2}$$

La frecuencia se eligió de tal manera que al presionar una tecla no fuera molesta para el oído.

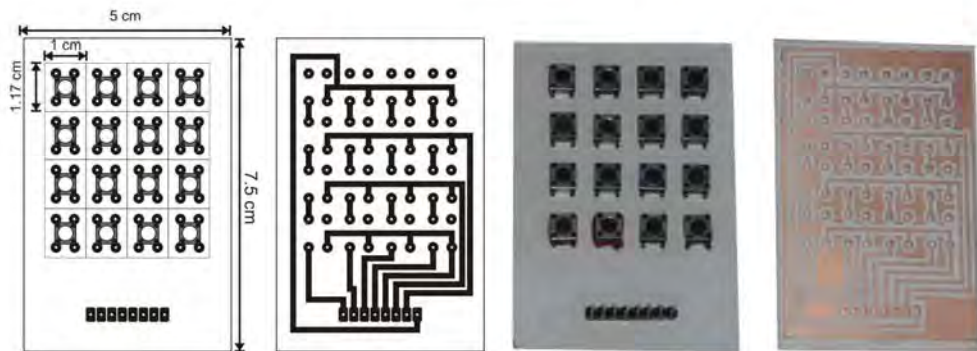
Con los valores elegidos se tiene una frecuencia aproximada de 213 [Hz] con un ciclo de trabajo del 84%.



Teclado #1



Teclado #2



Teclado #3

Figura 12. Teclados fabricados.

Al concluir las pruebas decidimos que el teclado #1 era el más fácil de usar debido a la separación de sus teclas, además los push button no requieren de tanta presión para activar cada tecla.

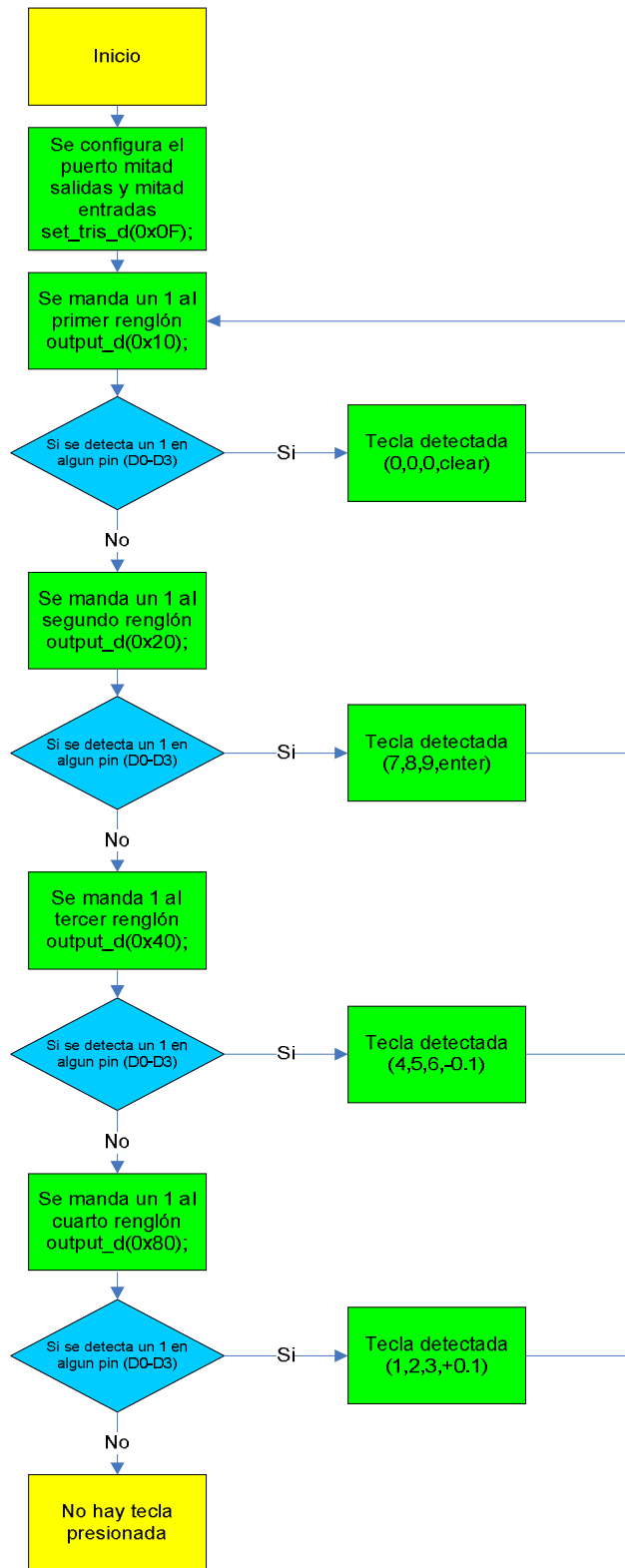
Los teclados #2 y #3 no eran los adecuados para la aplicación debido a que el teclado #2 tenía switches con un bastón lo que hacía que el teclado no cupiera en el espacio destinado.

El teclado #3 era muy pequeño lo que hacía que los switches estén muy juntos y al querer presionar una tecla se presione otra por accidente.

Para saber qué tecla ha sido presionada se realizó una rutina para la lectura del teclado matricial, cuyo diagrama de flujo sería el siguiente:



## Rutina para realizar la lectura del teclado



### 3.4 Despliegue del ciclo de trabajo

Para el despliegue del ciclo de trabajo se decidió usar displays de 7 segmentos debido a la facilidad visual.

Usar tres displays de 7 segmentos implicaría tener disponibles 21 pines del microcontrolador lo cual para fines prácticos sería un desperdicio, por lo que se optó por utilizar el multiplexaje de las señales para con esto lograr solo tener que usar 10 pines del microcontrolador.

Para realizar dicha operación utilizamos transistores tal como se ve en la figura.

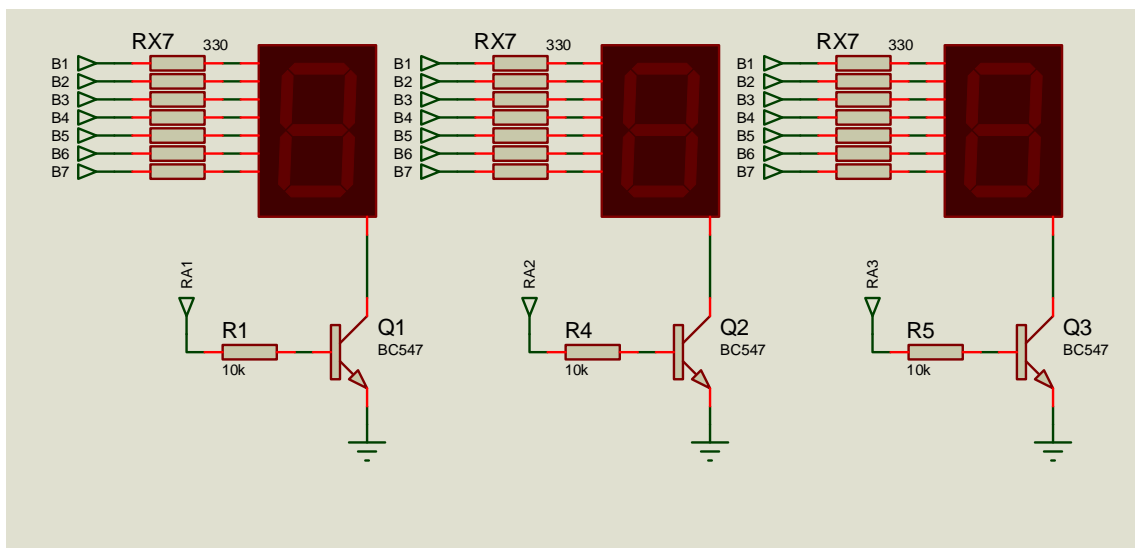


Figura 13. Circuito para el despliegue del ciclo de trabajo.

El transistor opera en la región de corte y saturación de tal manera que solo opera un display a la vez por medio de una señal de control o habilitadora, que es proporcionada por el microcontrolador.

Para que el multiplexaje funcione debemos tener una frecuencia tal que no se note un parpadeo y que no sobrepase las limitaciones del transistor.

Para generar dicha frecuencia de multiplexaje utilizamos una interrupción la cual se realizaba cada 5ms aproximadamente, con esto logramos un despliegue prácticamente continuo para el ojo humano.

### 3.5 Monitoreo de la señal PWM

Una vez que el microcontrolador ha generado la señal PWM se debe tener un monitoreo de dicha señal para garantizar su correcto funcionamiento. Para ello decidimos tomar una muestra de la señal de salida y, mediante un filtro paso-bajas, obtener su voltaje promedio para después relacionarlo con su ciclo de trabajo correspondiente, por ejemplo, si el ciclo de trabajo es de 100% entonces el voltaje promedio debe ser 5V, y si el ciclo de trabajo es de 50% el voltaje promedio debe ser 2.5V.

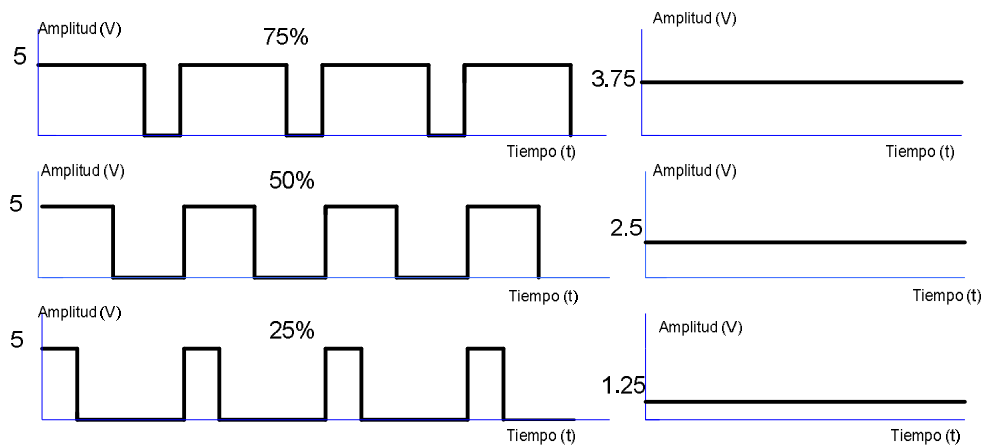


Figura 14. Variación de amplitud respecto al ciclo de trabajo.

Para obtener el voltaje promedio utilizamos un filtro paso-bajas RC. Si la señal PWM tiene una frecuencia lo suficientemente alta respecto a la frecuencia de corte de un filtro paso-bajas el voltaje de salida del filtro sería prácticamente constante y su valor sería la componente de directa de la señal.

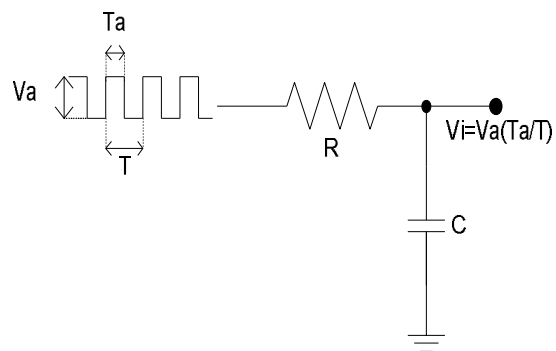


Figura 15. Filtro RC.

Después de obtener el voltaje promedio de la señal PWM realizamos una retroalimentación con el microcontrolador para estar monitoreando la señal, esto se hizo mediante el convertidor analógico digital. Después de esto el microcontrolador compara la lectura del convertidor A/D con su PWM correspondiente y decide si está dentro de la tolerancia.

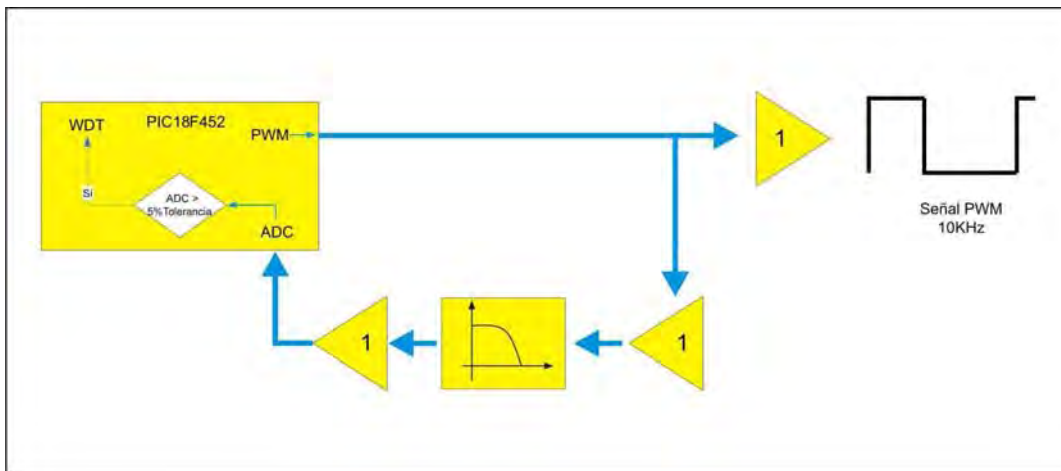


Figura 16. Diagrama del proceso de monitoreo.

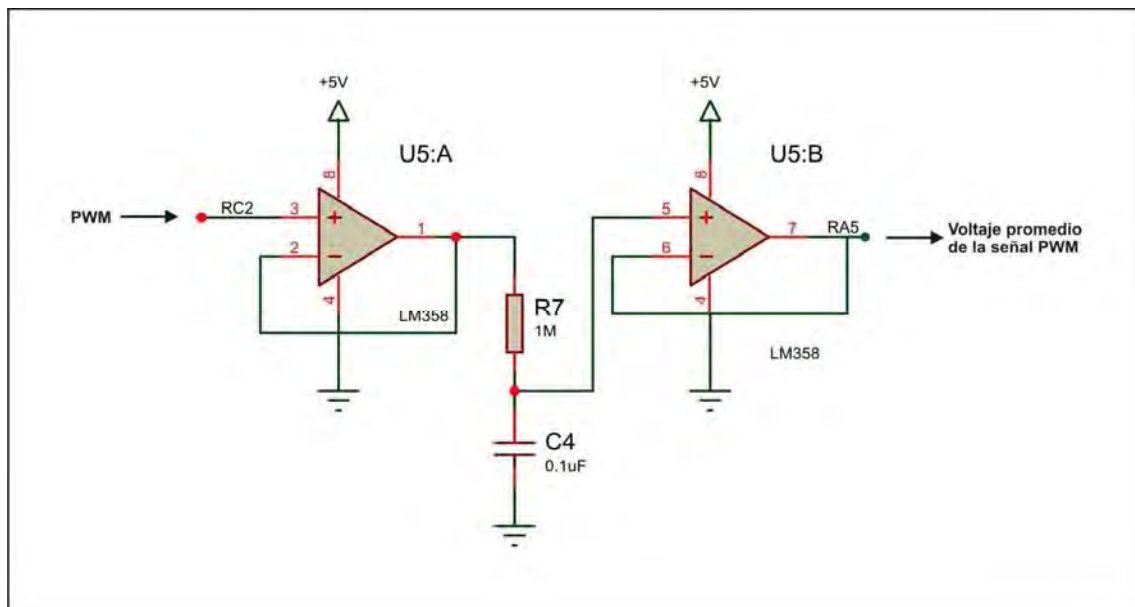


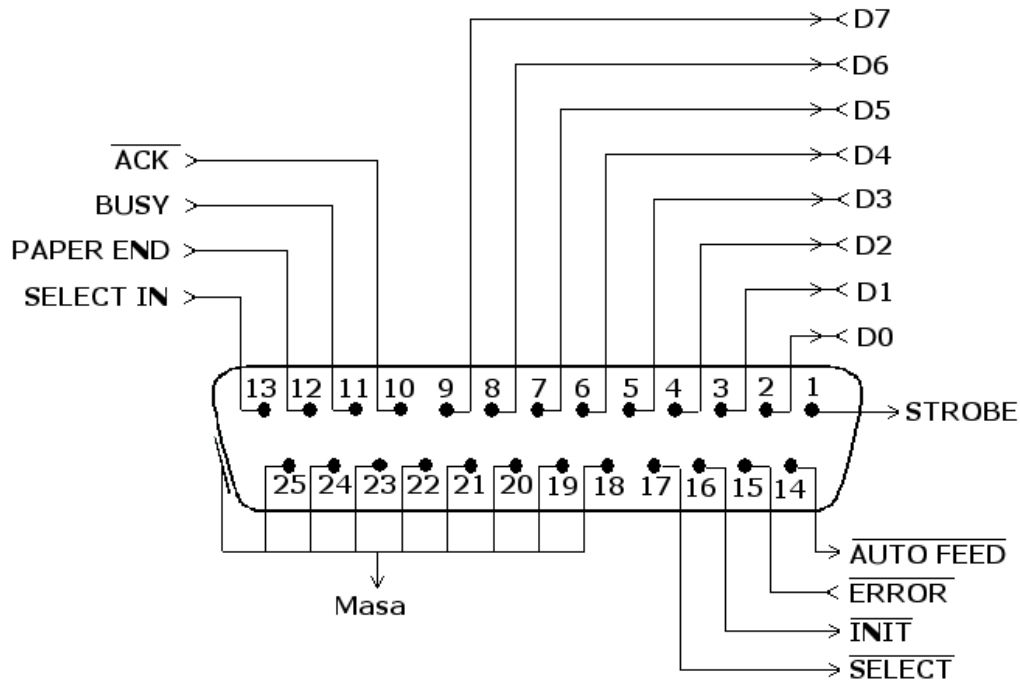
Figura 17. Circuito para realizar el monitoreo.

### 3.6 Control del ciclo de trabajo mediante la PC

#### 3.61 Breve descripción del puerto paralelo

El puerto paralelo consta de tres registros (datos, estado y control) de 8 bits cada uno, que ocupan tres direcciones de E/S. Cada puerto requiere tres direcciones de memoria consecutivas del espacio de E/S (entrada-salida) del procesador para seleccionar todas sus posibilidades. La dirección base del puerto LPT es la 378h ó 888.

PUERTO	REGISTROS			
	DATOS	ESTADO	CONTROL	NOMBRE
	378h	379h	37Ah	LPT1



- - 8 líneas (pines) son para salida de datos (bits de DATOS). Sus valores son únicamente modificables a través de software, y van del pin 2 (dato 0, D0) al pin 9 (dato 7, D7).

- - 5 líneas son de entrada de datos (bits de ESTADO), únicamente modificables a través del hardware externo. Estos pines son: 11, 10, 12, 13 y 15, del más al menos significativo.
- - 4 líneas son de control (bits de CONTROL), numerados del más significativo al menos: 17, 16, 14 y 1. Habitualmente son salidas, aunque se pueden utilizar también como entradas y, por tanto, se pueden modificar tanto por software como por hardware.
- - Las líneas de la 18 a la 25 son la tierra.

Cabe mencionar que en Windows XP no se tiene el acceso al puerto paralelo directamente. Sin embargo hay algunos controladores que se pueden descargar de la red, como el que elegimos nosotros, que es la librería INPOUT32.DLL con la cual podemos tener acceso al puerto paralelo por medio de algún software, en nuestro caso Visual Basic.

Por otra parte nosotros únicamente utilizamos el puerto de datos del puerto paralelo, ya que únicamente enviamos datos desde la PC al microcontrolador.

### **3.62 Teclado virtual**

Por medio de una interfaz creada en Visual Basic se puede simular un teclado muy similar al que se muestra en el módulo salvo por algunas teclas adicionales como se muestra en la figura 18.



Figura 18. Vista del teclado virtual.

En este teclado se puede apreciar que tiene dos teclas adicionales que no contiene el teclado instalado en el módulo debido a que se pretende tener un control total del equipo desde la PC ya que con estas teclas se puede tener un paro de emergencia así como también poder deshabilitar o habilitar los teclados para que el equipo no opere hasta que se este seguro de que se quiere trabajar. Por otra parte posee las mismas características que el teclado físico y el modo de operación es igual para que no exista confusión entre uno y otro.

Para comunicarnos por medio del puerto paralelo se creó una codificación, la cual es útil para poder enviársela al microcontrolador y poder así después decodificarla para que la interprete el módulo. Esto se hizo así debido a que se debían cumplir con ciertas especificaciones como son:

- A) Tener un bit para paro de emergencia
- B) Tener un bit para habilitar o deshabilitar los teclados

Por lo que sólo usamos 4 bits para mandar los dígitos del ciclo de trabajo.

Puerto de datos	D7	D6	D5	D4	D3	D2	D1	D0
Designación	MSB	-----	-----	LSB	Identificación de tecla presionada	-----	Stop	Habilitación
Tecla '0' (0x09)	0	0	0	0	1	0	0	1
Tecla '1' (0x19)	0	0	0	1	1	0	0	1
Tecla '2' (0x29)	0	0	1	0	1	0	0	1
Tecla '3' (0x39)	0	0	1	1	1	0	0	1
Tecla '4' (0x49)	0	1	0	0	1	0	0	1
Tecla '5' (0x59)	0	1	0	1	1	0	0	1
Tecla '6' (0x69)	0	1	1	0	1	0	0	1
Tecla '7' (0x79)	0	1	1	1	1	0	0	1
Tecla '8' (0x89)	1	0	0	0	1	0	0	1
Tecla '9' (0x99)	1	0	0	1	1	0	0	1
Tecla '+0.1' (0xA9)	1	0	1	0	1	0	0	1
Tecla '-0.1' (0xB9)	1	0	1	1	1	0	0	1
Tecla 'enter' (0xC9)	1	1	0	0	1	0	0	1
Tecla 'clear' (0xD9)	1	1	0	1	1	0	0	1
Tecla 'stop' (0x03)	0	0	0	0	0	0	1	1
Tecla 'habilitar' (0x00)-(0x01)	0	0	0	0	0	0	0	0/1

Figura 19. Código generado para la interfaz con el puerto paralelo.



### 3.7 Salida PWM

Para entregar la señal con ciclo de trabajo variable, el sistema cuenta con un conector tipo BNC para poder conectarse con el acondicionamiento de señal para que pueda funcionar el láser.

Siendo así para mostrar el ciclo de trabajo utilizamos un comparador para tener una señal lo más cuadrada posible.

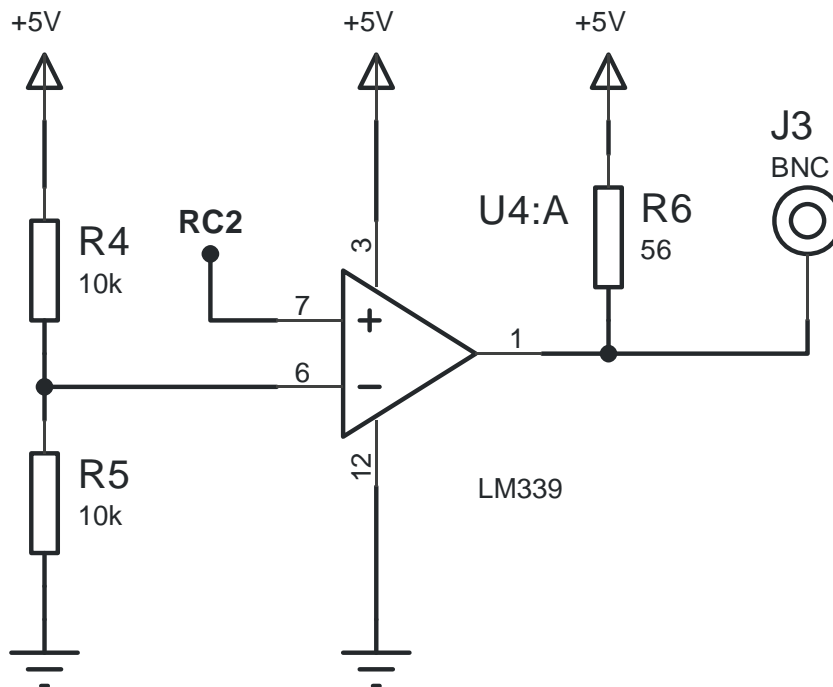


Figura 20. Circuito acondicionador de señal.

Al realizar este circuito tuvimos ciertas complicaciones con la resistencia de pull-up debido a que típicamente se utiliza una resistencia de 10k, sumándole el efecto capacitivo del cable BNC a dicha resistencia provocaba que la salida PWM no fuera del todo cuadrada sino que tuviera un efecto de carga y descarga, es por ello que utilizamos una resistencia de 56 ohms con la cual se reducía el problema.

Elegimos el circuito integrado LM339 debido a que es un comparador de alta precisión y su tiempo de respuesta es el adecuado para nuestra aplicación ya que es de 1.3µs y nuestra señal es de 10kHz (100µs).

### 3.8 Diseño exterior del sistema

Una parte importante en el desarrollo de este sistema fue el aspecto físico que tendría el módulo. No estábamos interesados en crear un producto estético, más bien pensamos en un producto funcional que se adaptara a las necesidades del usuario.

Para ensamblar el producto final decidimos utilizar un gabinete de plástico del tamaño justo para contener todos los componentes del sistema. Una vez escogido el gabinete el siguiente paso fue definir la distribución de cada uno de los componentes: fuente de alimentación, circuitos impresos, teclado, conector DB25 para puerto paralelo, conector coaxial para la salida PWM, conector para cable de alimentación, teclado, displays, buzzer, botón de paro y switch de encendido/apagado.

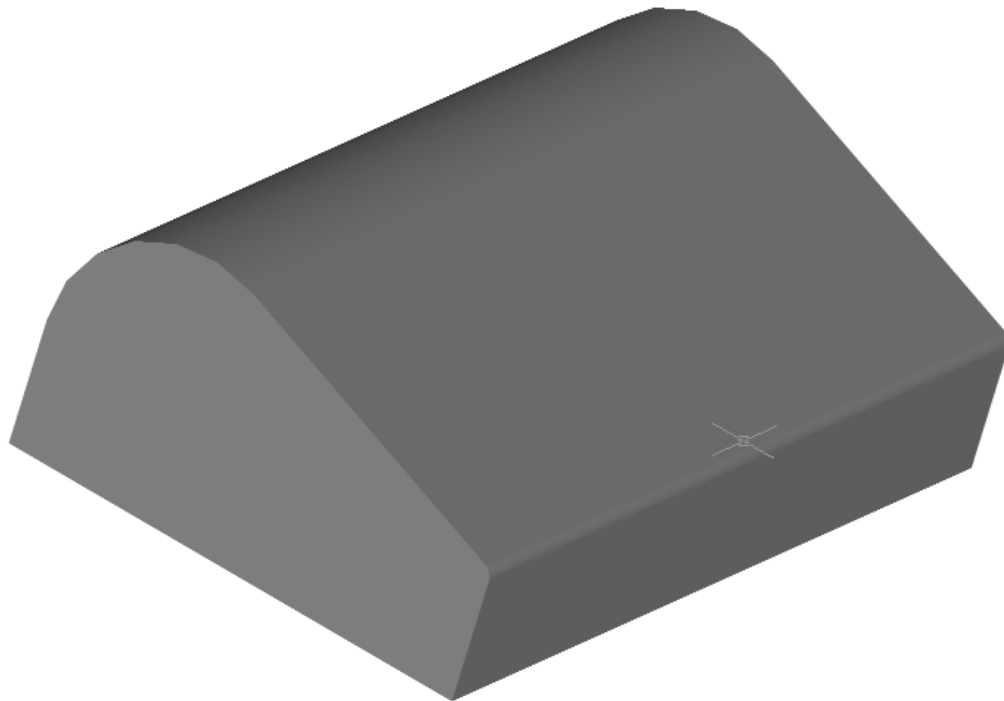


Figura 21. Gabinete de plástico

El display, el teclado y el botón de paro se ubican en la parte superior del gabinete. El teclado se encuentra a la derecha del display porque está diseñado para operarse con la mano derecha, de esta manera se puede ver cómodamente el display mientras se oprimen las teclas. El botón de paro se encuentra en un lugar de fácil acceso pensando en la necesidad de detener rápidamente la señal PWM. La carátula del sistema es de tipo membrana y está diseñada para que sea de fácil limpieza y para evitar la entrada y acumulación de polvo.



Figura 22. Vista frontal y vista posterior del sistema 1° versión.



Figura 23. Versión definitiva.

### 3.9 Pruebas realizadas al sistema

Después de haber realizado algunas mediciones al sistema en dos equipos de medición (osciloscopios) distintos, llegamos a la conclusión de que para que el sistema comience a trabajar, el ciclo de trabajo mínimo es de 0.2% como se ve en la figura, por otra parte el ciclo de trabajo máximo es de 99.8% ya que a partir de 99.9% se ve una señal prácticamente constante. Sin embargo, estando dentro de este intervalo, obtuvimos los resultados deseados que consistían en tener un ciclo de trabajo variable con una resolución de 0.1% a una frecuencia de 10 kHz y a una amplitud de 5V.



Figura 24. Ciclo de trabajo mostrado en el sistema.

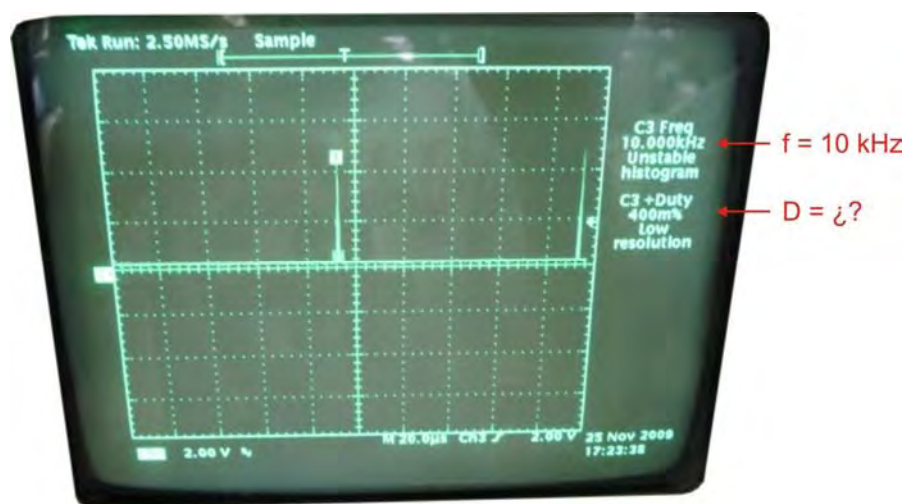


Figura 25. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 0.02 %. Para esta señal el osciloscopio midió una frecuencia de 10 kHz pero no midió su ciclo de trabajo.



Figura 26. Ciclo de trabajo mostrado en el sistema.

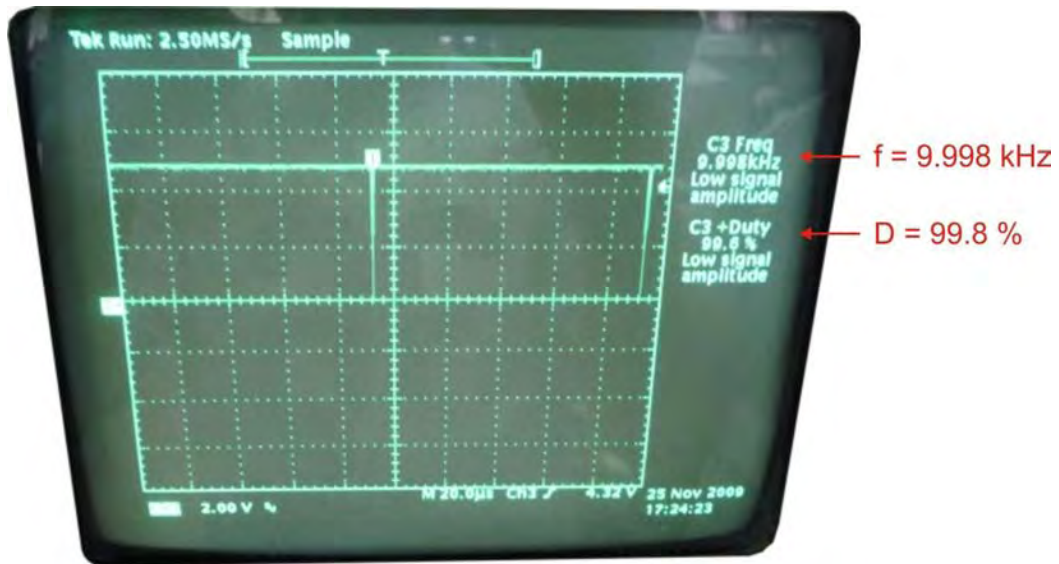


Figura 27. Ciclo de trabajo máximo obtenido. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 99.8 %. Para esta señal el osciloscopio midió una frecuencia de 9.998 kHz y un ciclo de trabajo de 99.8 %.



Figura 28. Ciclo de trabajo mostrado en el sistema.

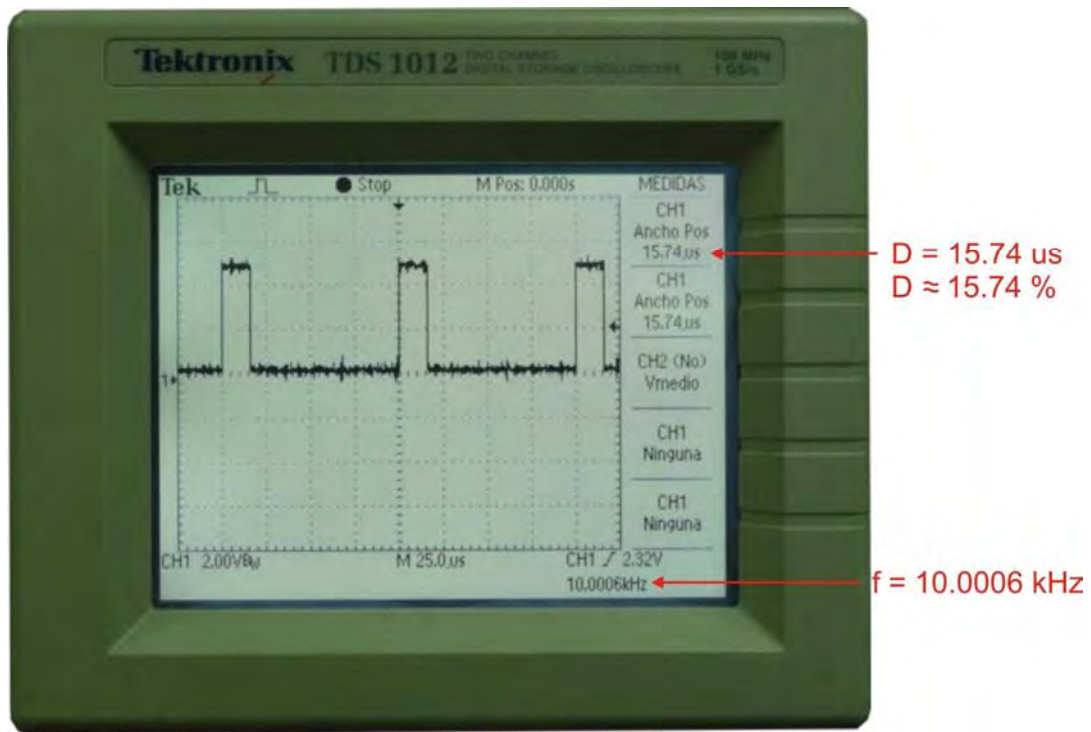


Figura 29. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 15.7 %. Para esta señal el osciloscopio midió una frecuencia de 10.0006 kHz y un ciclo de trabajo de 15.74  $\mu$ s.



Figura 30. Ciclo de trabajo mostrado en el sistema

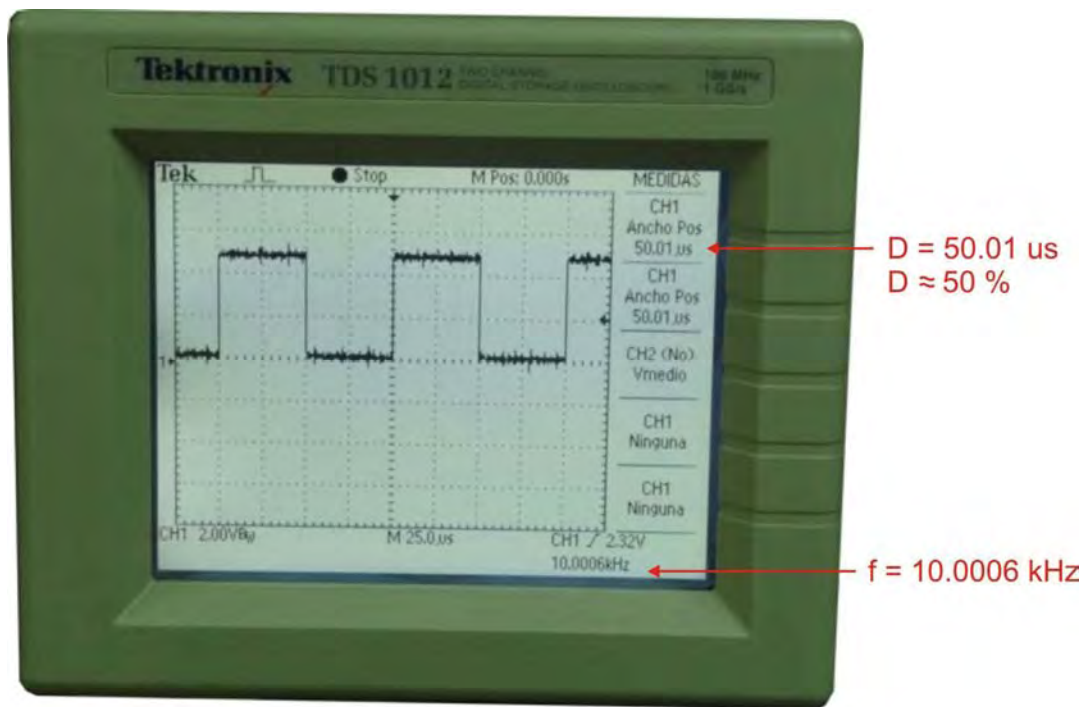


Figura 31. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 50 %. Para esta señal el osciloscopio midió una frecuencia de 10.0006 kHz y un ciclo de trabajo de 50  $\mu$ s.

### 3.10 Diseño interior del sistema

Dado que las dimensiones del gabinete no eran adecuadas para hacer solo un circuito impreso, nos vimos en la necesidad de realizar varios módulos para poder así tener un mayor aprovechamiento del espacio.

Además de que facilita el mantenimiento, dado que al ser módulos independientes estos pueden ser retirados para su reparación o cambio de estos en caso de ser necesario.

Las interconexiones entre los módulos se realizaron con conectores tipo molex debido a que su conexión es única, por lo que no hay forma alguna de conectarlo incorrectamente.



Figura 32. Vista interior del sistema.



A) Circuito impreso principal

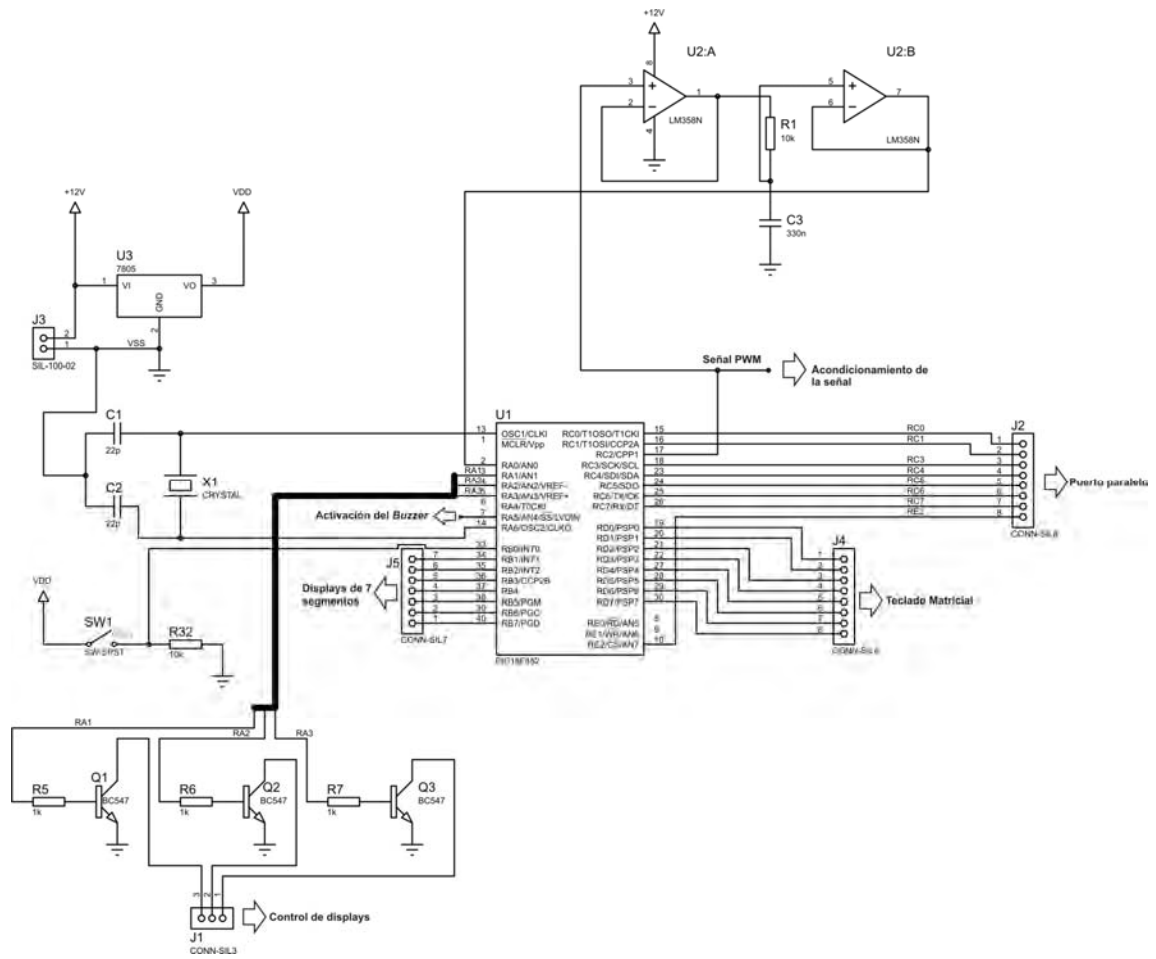


Figura 33. Diagrama eléctrico del circuito impreso principal.

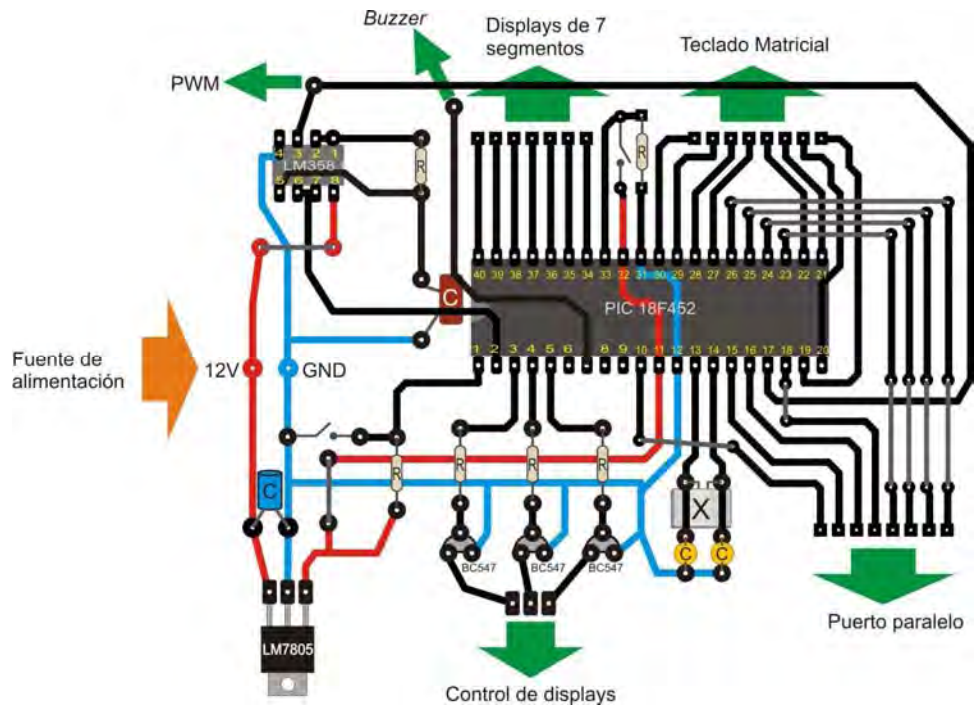


Figura 34. Circuito impreso principal con representación de componentes electrónicos.

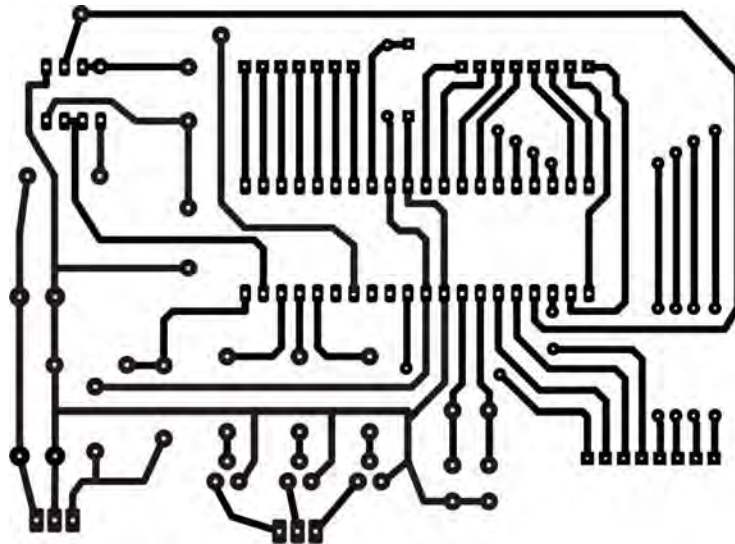


Figura 35. Diseño final del circuito impreso principal.

B) Fuente de alimentación

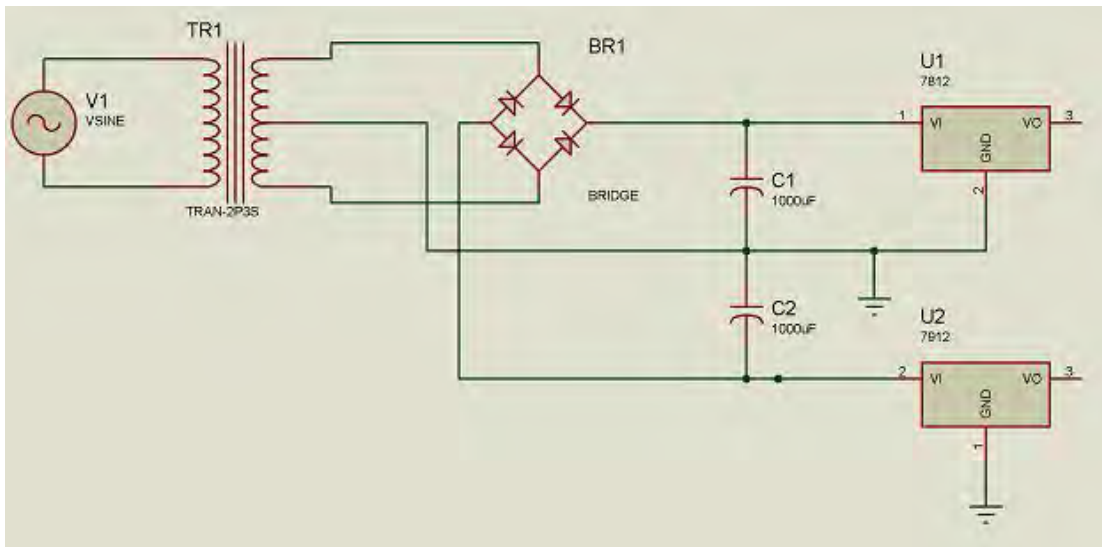


Figura 36. Diagrama eléctrico de la fuente de alimentación.

C) Teclado Matricial

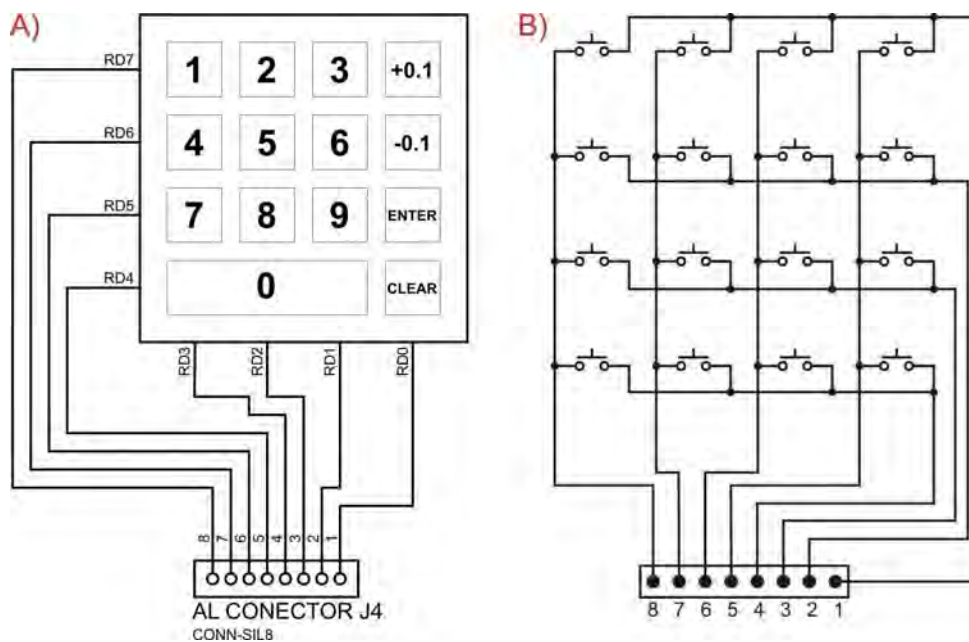


Figura 37. Teclado matricial. A) Distribución de teclas. B) Diagrama eléctrico.

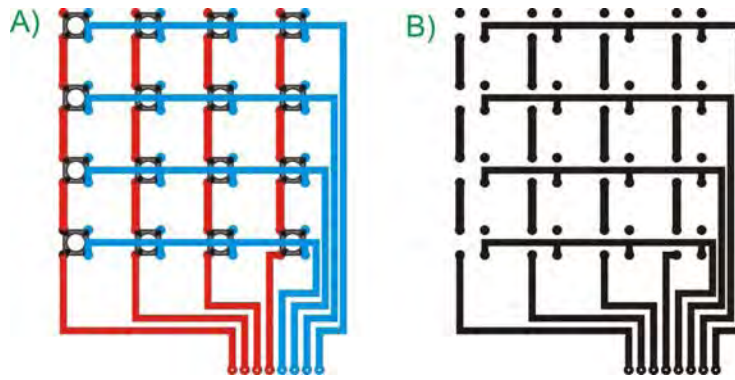


Figura 38. A) Circuito impreso del teclado matricial con representación de componentes electrónicos. B) Diseño final del circuito impreso del teclado matricial.

D) Displays de 7 segmentos

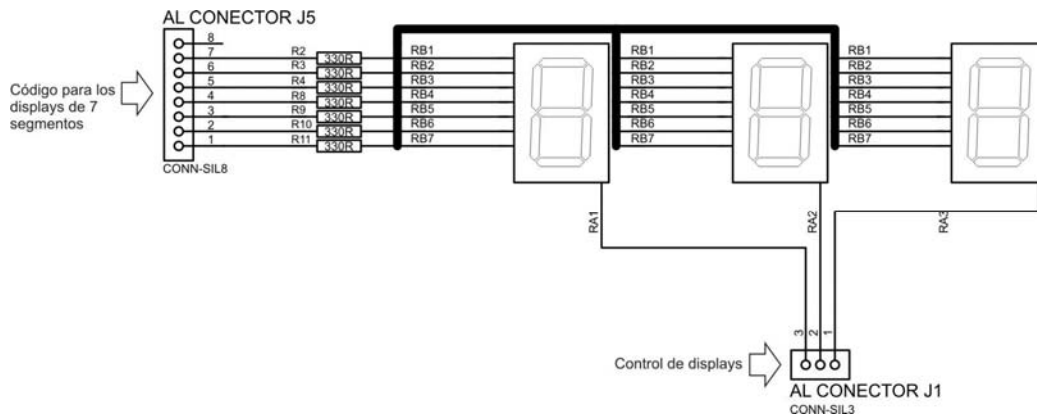


Figura 39. Diagrama eléctrico del módulo de displays.

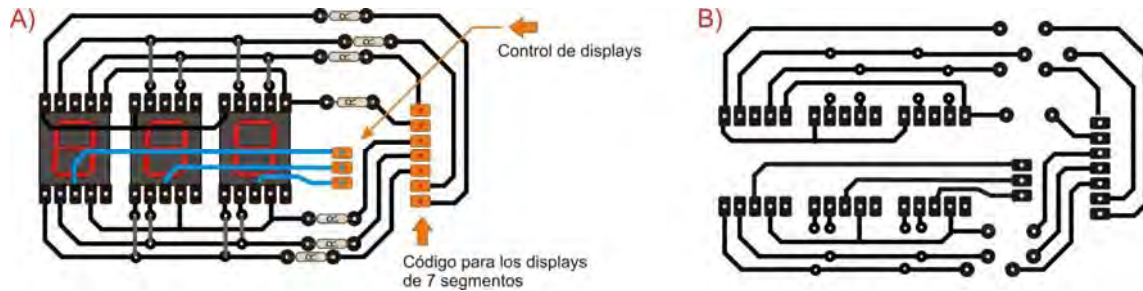


Figura 40. A) Circuito impreso del módulo de displays con representación de componentes electrónicos. B) Diseño final del circuito impreso del módulo de displays.

E) Acondicionamiento de señal y generación de sonido.

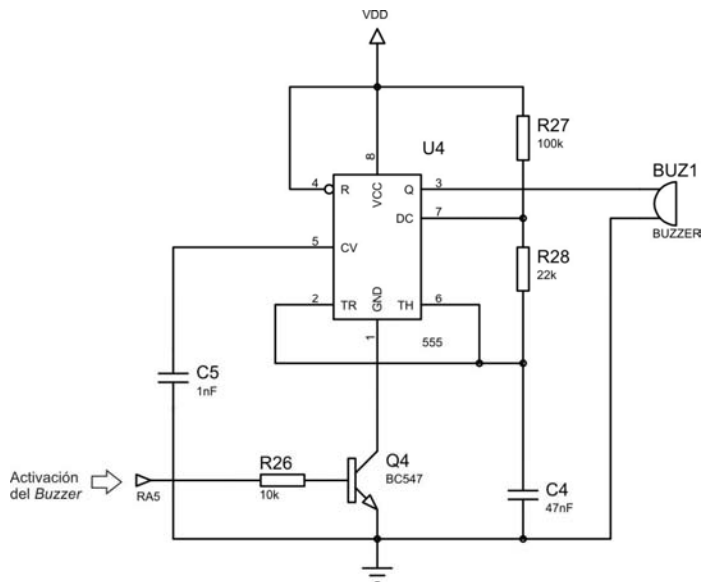


Figura 41. Diagrama eléctrico del módulo de acondicionamiento de señal y generación de sonido.

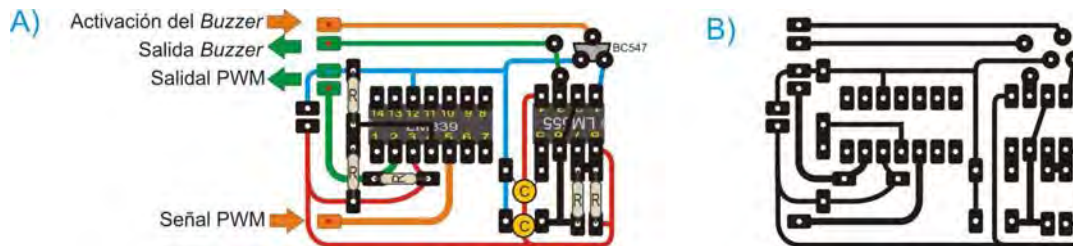


Figura 42. A) Circuito impreso del módulo de acondicionamiento de señal y generación de sonido con representación de componentes electrónicos. B) Diseño final del circuito impreso del módulo de acondicionamiento de señal y generación de sonido.

## Distribución de los pines del PIC18F452

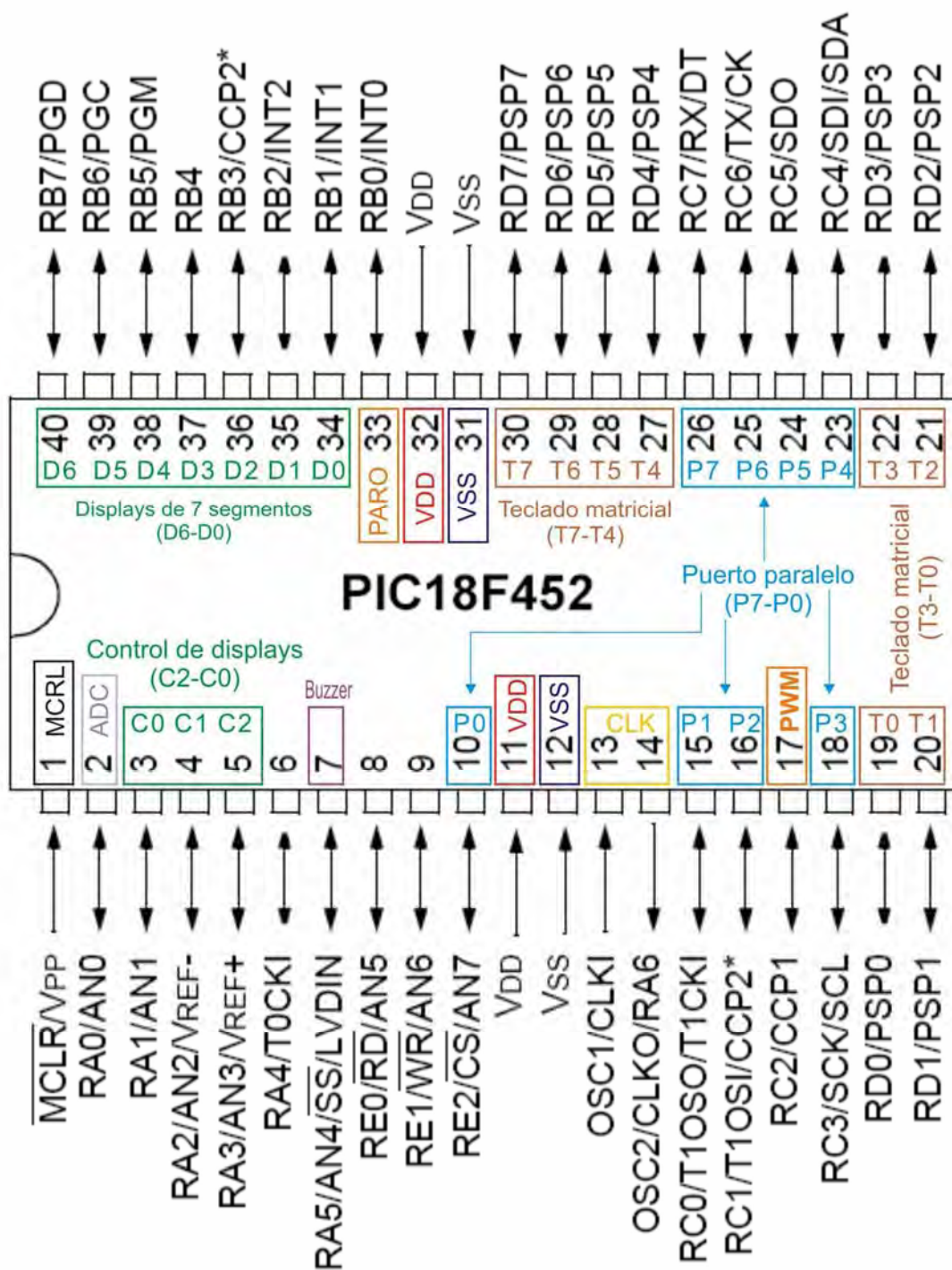


Figura 43. Distribución de pines del PIC18F452.

- Pin 1: MCLR o master clear (reset), pin de reinicio.

- Pin 2: Buzzer o zumbador, activa una señal audible cuando se oprime alguna tecla.
- Pines 3-5: Control de displays. Estos pines son los encargados de multiplexar los 3 displays de 7 segmentos.
- Pin 7: ADC, convertidor analógico-digital.
- Pines 10, 15, 16, 18 y 23-26: Pines destinados a la comunicación con la PC por puerto paralelo.
- Pines 11, 12, 31 y 32: Alimentación del PIC.
- Pines 13 y 14: Señal de reloj del PIC.
- Pin 17: Salida de la señal PWM.
- Pines 19-22 y 27-30: Pines para la lectura del teclado matricial.
- Pin 33: Pin de interrupción externa, está conectado al botón de paro del sistema.
- Pines 34-40: Señal de salida hacia los displays de 7 segmentos.

### 3.11 Lista de componentes y precios del sistema

#### LISTA DE PRECIOS DEL SISTEMA DE CONTROL PARA LA INTENSIDAD DE UN LASER

Circuitos integrados				
CANTIDAD		TIPO	C/U	TOTAL
1	LM358	Amplificador operacional	\$ 5.00	\$ 5.00
1	LM339	Comparador	\$ 5.00	\$ 5.00
1	LM7912	Regulador de voltaje negativo	\$ 5.00	\$ 5.00
	-12V			
1	LM7812	Regulador de voltaje positivo	\$ 5.00	\$ 5.00
	12V			
1	NE555	Temporizador	\$ 5.00	\$ 5.00
1	LM7805	Regulador de voltaje negativo	\$ 5.00	\$ 5.00
	5V			
1	PIC18F452	Microcontrolador	\$ 75.00	\$ 75.00

<b>SEMICONDUCTORES</b>				
3	BC547	Transistor bipolar NPN	\$ 2.00	\$ 6.00
1	2W02M	Puente rectificador	\$ 5.00	\$ 5.00
<b>RESISTORES</b>				
1	47 $\Omega$	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.10
1	560 $\Omega$	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.10
5	1k $\Omega$	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.50
2	10 k $\Omega$	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.20
1	22 k $\Omega$	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.10
1	100 k $\Omega$	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.20
1	1M $\Omega$	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.10
<b>CAPACITORES</b>				
1	10 nF	Capacitor Cerámico	\$ 0.50	\$ 0.50
2	22 pF	Capacitor Cerámico	\$ 0.50	\$ 1.00
1	100 uF @25V	Cap. Electrolítico Radial	\$ 3.00	\$ 3.00
2	220 uF @25V	Cap. Electrolítico Radial	\$ 3.00	\$ 6.00
1	1000 uF @25V	Cap. Electrolítico Radial	\$ 5.00	\$ 5.00
1	0.1 uF	Cap. Poliéster	\$ 4.00	\$ 4.00
1	4.7 nF	Cap. Poliéster	\$ 3.00	\$ 3.00
<b>OTROS</b>				
1	Transformador con tap 127@12V -0.5 <sup>a</sup>		\$ 70.00	\$ 70.00
1	Terminal con 2 tornillos para circuito impreso		\$ 5.00	\$ 5.00
3	Display 7 segmentos C-C		\$ 7.50	\$ 22.50
1	Jack DB25 para cable plano		\$ 29.00	\$ 29.00
17	Microswitch, de push, 4 terminales		\$ 1.00	\$ 17.00
1	Switch de push, de botón cuadrado, normalmente abierto		\$ 7.00	\$ 7.00
1	Receptáculo de alimentación		\$ 10.00	\$ 10.00
1	Cable de alimentación		\$ 25.00	\$ 25.00
1	Disipadores de calor TO-220C/C		\$ 5.00	\$ 5.00
1	Buzzer de 2.9 kHz de 3 a 20 VCC		\$ 15.00	\$ 15.00
1	Switch de balancín, de 1 polo, 1 tiro, 2 posiciones con piloto		\$ 8.50	\$ 8.50
1	Bases p/ circuito impreso 40 pines. Para soldar.		\$ 3.00	\$ 3.00
1	Portafusible de cartucho tipo europeo		\$ 6.00	\$ 6.00
1	Fusible europeo 5 A @250 V		\$ 1.00	\$ 1.00
1	Conector BNC para chasis		\$ 6.00	\$ 6.00
1	Placa fenólica 20x20 cms.		\$ 29.00	\$ 29.00
1	Cristal de cuarzo de 10 MHz		\$ 10.00	\$ 10.00
1	Gabinete de plástico		\$150.00	\$150.00
1	Carátula		\$ 20.00	\$ 20.00
			<b>TOTAL</b>	<b>\$ 578.80</b>



### 3.12 Programación del microcontrolador PIC

Para la programación del microcontrolador PIC nosotros decidimos utilizar un compilador de lenguaje C, debido a las grandes ventajas que nos proporciona un lenguaje de alto nivel como son:

- a) Facilidad de programación. Debido a que el lenguaje C dispone de una serie de comandos, operadores y datos nos permite desarrollar programas en un tiempo mucho menor que en el lenguaje ensamblador.
- b) Programación estructurada. Debido a la organización del lenguaje C se puede tener un completo orden de la estructura del programa, teniendo una organización de espacios para las cabeceras, para las funciones y para el programa principal a diferencia del lenguaje ensamblador que es una programación no estructurada.
- c) Compatibilidad entre plataformas. Teniendo un algoritmo implementado en C puede ser adaptado a cualquier compilador de lenguaje C que se maneje
- d) Fácil mantenimiento de los programas. Debido a que se tiene una estructura de programación, la corrección de problemas puede ser más rápido y fácil.

Sin embargo no se puede descartar por completo el lenguaje ensamblador ya que al estar mas en contacto con el código de máquina, se pueden realizar rutinas más eficientes, además de tener una mayor velocidad de programa.

## 4 Conclusiones

En el presente trabajo mostramos el diseño y construcción de un equipo que servirá para la fabricación de bioprótesis cardiacas. Debido a que este proyecto es una segunda versión se le añadieron mejoras considerando los detalles que tenía la primera versión.

De acuerdo a los resultados mostrados previamente podemos decir que el sistema cumple satisfactoriamente con todas las características y requisitos de diseño propuestos. La señal PWM cumple con las condiciones de trabajo deseadas, dado que tenemos un ciclo de trabajo variable de 0-99% con una resolución de 0.1% con una tolerancia del 5%, si sobrepasa esta tolerancia el sistema se reiniciará. Además la amplitud de la señal es de 5V y su frecuencia, de 10kHz, se mantiene prácticamente constante.

Sin embargo cabe mencionar que aparentemente el sistema comienza a trabajar después de 0.2% y antes de 99.9%, esto podría deberse a que el osciloscopio mide los límites cercanos, es decir, en lugar de 0.1% mide 0% y en lugar de 99.9% mide 100%.

Por otra parte, al tratar de acondicionar la señal, tuvimos un problema al colocar un comparador de colector abierto, debido a que el cable con el que se conecta el equipo es del tipo BNC y éste presenta una capacitancia. Al conectar el cable BNC a la salida del comparador se forma un circuito RC serie, debido a la resistencia del comparador y la capacitancia del cable, que produce un efecto de filtro paso bajas, el cual nos hace una señal un tanto curva en lugar de la señal cuadrada que se esperaba. Es por ello que se tuvo que sustituir la resistencia del comparador por una más pequeña para minimizar este efecto.

El siguiente paso consistiría en probar este sistema junto con el láser GEM-30 y con los demás módulos que componen el proyecto para evaluar su funcionalidad, y en caso de ser necesario, efectuar correcciones y proponer mejoras para desarrollar una nueva versión.

## 5. Apéndice

## 5.1 manual de usuario



**MANUAL PARA LA INTERFAZ DE USUARIO PARA EL SISTEMA DE CONTROL  
DE INTENSIDAD DE UN LASER UTILIZADO EN LA FABRICACIÓN DE  
BIOPRÓTESIS CARDIACAS**



**CENTRO DE CIENCIAS APLICADAS Y  
DESARROLLO TECNOLÓGICO**

Laboratorio de electrónica

# OPERACIÓN DEL SISTEMA

- 1.- El usuario deberá conectar el sistema a la computadora por medio de un cable DB25 al puerto paralelo. (**nota: de no hacer esto el sistema estará bloqueado marcando ceros y haciendo un sonido**)
- 2.- Una vez que esté conectado el sistema deberá abrir la interfaz proporcionada cuyas indicaciones de uso se mostrarán posteriormente.
- 3.- El sistema puede trabajar de dos formas seleccionándose indistintamente:
  - a) Por medio de un teclado que se encuentra en el sistema y
  - b) Por medio de la PC utilizando el mouse y el teclado que se muestra en el software.



Figura 44. Vista posterior del módulo.

## OPERACIÓN DEL SISTEMA MEDIANTE EL TECLADO QUE SE ENCUENTRA EN EL MÓDULO

El sistema inicialmente estará en espera de que el usuario introduzca el ciclo de trabajo deseado por medio del teclado (mostrará ceros hasta que se introduzca un número), una vez que el usuario haya introducido dicho ciclo de trabajo deberá oprimir la tecla "ENTER" para que el sistema comience a trabajar.

En el caso de haberse equivocado al introducir el ciclo de trabajo se tienen dos opciones:

- a) Oprimir la tecla "CLEAR" la cual hará que aparezcan ceros en el display sin modificar el ciclo de trabajo actual.
- b) Reingresar los números pertenecientes al ciclo de trabajo deseado, cabe mencionar que sólo se obtendrá la salida con el ciclo de trabajo mostrado en los displays hasta que se presione la tecla "ENTER".

El orden en el que se introduce el ciclo de trabajo es: Decenas, unidades y parte fraccionaria.

En el caso de necesitar un ciclo de trabajo entero múltiplo de 10 es decir 10%,20%, etc. Basta con introducir el número que pertenece a las decenas y oprimir la tecla "ENTER".

El sistema tiene una resolución de 0.1% el cual se puede modificar por medio de las 2 teclas "+0.1" y "-0.1".

El sistema cuenta con un zumbador el cual suena al oprimir una tecla.

En la parte inferior donde se ubican el despliegue de los números hay un botón el cual sirve como paro en caso de que el sistema tenga que ser detenido.



Figura 45. Vista frontal del módulo.

## OPERACIÓN DEL SISTEMA MEDIANTE LA PC

Por medio del software Visual Basic se creó una interfaz con el sistema por medio del puerto paralelo.

El teclado que se muestra en dicha interfaz tiene las mismas características que el teclado que esta en el módulo, con la finalidad de que se pueda trabajar de la misma manera.

A diferencia del teclado físico, este teclado tiene dos teclas adicionales las cuales son:

- a) **Habilitar:** cuya función es habilitar ambos teclados. Una vez que se oprime esta tecla el usuario podrá activar la salida PWM por medio de la tecla "ENTER".
- b) **Deshabilitar:** cuya función es totalmente contraria a la tecla anterior. Una vez que se oprima dicha tecla, el teclado físico seguirá funcionando pero **deshabilita la función de la tecla "ENTER"**, esto es, que no habrá salida PWM.



Figura 46 Teclado virtual.

### POSIBLES FALLAS Y POSIBLES SOLUCIONES

Falla	Solución
El sistema no enciende.	Verificar que la luz del botón de encendido este prendida, en caso contrario verificar que el fusible funcione, en el caso de que esto este bien verificar que haya energía eléctrica en el tomacorriente.
El display marca ceros y hace un sonido.	Verificar que este conectado el cable paralelo.
No hay salida PWM.	Verificar que la tecla enter haya sido oprimida, si esto persiste verificar que la tecla habilitar haya sido oprimida.
El teclado del sistema no funciona	Verificar que el conector no se haya salido de su lugar o que algún cable este desconectado.



## 5.2 Código fuente del PIC

```
#include <18f452.h>
#device ADC=10
#fuses HS,WDT,NOPROTECT,NOLVP,PUT,BROWNOUT,
#use delay(clock=1000000)
#use standard_io(c)
#use standard_io(e)

byte          const          LED_MAP[16]          =
{0x7e,0x0d,0xb6,0x9e,0xcc,0xda,0xfa,0x0e,0xfe,0xce,0x7e,0x7e,0x7e,0x7e,0x7e,0x7e};

short pcen = false;
short pdec = false;
short puni = false;
short boton = false;
short conversion;
short habilitado = false;
int cont=1;
int tempo;
int tecla;
int tec;
int centenas, decenas, unidades;
int n = 0;
int p = 0;
int c = 0;
long valor;
long duty;
long tol;

int teclado(void){

output_d(0x10);
if (input(pin_d3)){
    tecla = 14;
}

if (input(pin_d2)){
    tecla = 0;
}

if (input(pin_d1)){
    tecla = 15;
}

if (input(pin_d0)){
    tecla = 13;
}

output_d(0x20);
if (input(pin_d3)){
    tecla = 7;
}

if (input(pin_d2)){
    tecla = 8;
}

if (input(pin_d1)){
```

```

        tecla = 9;
    }

    if (input(pin_d0)){
        tecla = 12;
    }

    output_d(0x40);
    if (input(pin_d3)){
        tecla = 4;
    }

    if (input(pin_d2)){
        tecla = 5;
    }

    if (input(pin_d1)){
        tecla = 6;
    }

    if (input(pin_d0)){
        tecla = 11;
    }

    output_d(0x80);
    if (input(pin_d3)){
        tecla = 1;
    }

    if (input(pin_d2)){
        tecla = 2;
    }

    if (input(pin_d1)){
        tecla = 3;
    }

    if (input(pin_d0)){
        tecla = 10;
    }
    return tecla;
}

int tecladovb(void){
    tecla = 0;
    if (input(pin_c4)){
        tecla = tecla + 1;
    }
    if (input(pin_c5)){
        tecla = tecla + 2;
    }
    if (input(pin_c6)){
        tecla = tecla + 4;
    }
    if (input(pin_c7)){
        tecla = tecla + 8;
    }
    return tecla;
}

```

```

void reset(void){
    set_pwm1_duty(0);
    unidades =0;
    decenas =0;
    centenas=0;
}

#INT_TIMER0
void toma_adc(void){ // Lectura del canal 0
    disable_interrupts(INT_TIMER0);
    conversion = adc_done();
    while(!conversion) {
        conversion = adc_done();
    }
    valor = read_adc();

    if (duty <= valor){
        tol = valor - duty;
    }else{
        tol = duty - valor;
    }
    set_timer0(0);
    enable_interrupts(INT_TIMER0);
}

void enter(void){
    if(input(pin_e2)){
        duty = _mul(centenas,10) + decenas;
        duty = _mul(duty,1000)/100;
        duty = duty + unidades;

        if (duty <= 1 ){
            duty = 0;
        }
        set_pwm1_duty(duty);
        pcen = false;
        pdec = false;
        puni = false;
        n = 0;
    }
}

#INT_EXT
void paro_de_emergencia(){
    ext_int_edge(0,L_TO_H);
    if(input(pin_b0)){
        reset();
        pcen = false;
        pdec = false;
        puni = false;
    }
}

#INT_TIMER1
void adquisicion(void){

    if(cont==1){

```

```

output_low(pin_a1);
output_low(pin_a2);
    output_b(LED_MAP[centenas]);
    if (p <=50 || pcent == false){
        output_high(pin_a3);
    }else{
        output_low(pin_a3);
    }
}
if(cont==2){
    output_low(pin_a1);
    output_low(pin_a3);
    output_b(LED_MAP[decenas]);
    if (p <=50 || pdec == false){
        output_high(pin_a2);
    }else{
        output_low(pin_a2);
    }
}
if(cont==3){
    output_low(pin_a2);
    output_low(pin_a3);
    output_b(LED_MAP[unidades]);
    if (p <=50 || puni == false){
        output_high(pin_a1);
    }else{
        output_low(pin_a1);
    }
    cont=0;
}
cont=cont+1;
p = p + 1;
if (p == 100){
    p = 0;
}

set_timer1(65000);
enable_interrupts(INT_TIMER1);
}

#INT_TIMER3
void P_paralelo(void){ // Lectura del canal 0
    if (input(pin_c0)){ //bit de paro
        set_pwm1_duty(0);
        unidades =0;
        decenas =0;
        centenas=0;
    }
    if (input(pin_e2)){
        habilitado = true;
    }else{
        habilitado = false;
    }
    set_timer3(65435); //40 us
}

void main() {
    set_tris_b(0xfe);

```

```

    set_tris_d(0x0F);
    set_tris_a(0xF0);
    set_tris_e(0xf0);

    setup_wdt(WDT_ON);
tempo=249;
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(0);
    setup_adc_ports(RA0_analog);
    setup_ccp1(CCP_PWM); // Configura CCP1 como PWM
    setup_timer_2(T2_DIV_BY_1,tempo,1); // f
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_2);//
    setup_timer_3 (T3_INTERNAL | T3_DIV_BY_1);
    set_timer3(65435); //para un T= 40us
    set_timer1(65000); // para un T=5 ms
    set_timer0(0); //para T=4 us
    enable_interrupts(INT_TIMER3);
    enable_interrupts(INT_TIMER1);
    enable_interrupts(INT_TIMER0);
    enable_interrupts(INT_EXT);
    enable_interrupts(GLOBAL);
    reset();
    duty =0;

do {
    restart_wdt();
    output_d(0xf0);
    if (duty !=0){
        do {
            }while(tol >= 50);
        }

    if (input(pin_d0)||input(pin_d1)||input(pin_d2)||input(pin_d3)||input(pin_c3)){
        delay_ms(10);
        if (boton == false){
            boton = true;
            if (input(pin_c3)){
                tec = tecladovb();
            }else{
                tec = teclado();
            }
            output_high(pin_a5);

            if(tec==10 && duty <=998 ){ //Aumentar 1%
                unidades = unidades + 1;
                if (unidades >=10){
                    unidades = unidades - 10;
                    decenas = decenas + 1;
                    if (decenas ==10){
                        decenas = 0;
                        centenas = centenas + 1;
                    }
                }
            }
            enter();
            output_low(pin_a5);
        }
    }
}

```

```

if(tec==11 && duty >=1){ //Reducir 1%
    if (unidades == 0){
        unidades = 9;
        if (decenas >=1){
            decenas = decenas - 1;
        }else{
            decenas = 9;
            if (centenas >=1){
                centenas = centenas - 1;
            }
        }
    }else{
        unidades = unidades -1;
    }
    enter();
    output_low(pin_a5);
}
if(tec==13){ //clear
    unidades = 0;
    decenas = 0;
    centenas = 0;
    n = 0;
    pcen = true;
    pdec = true;
    puni = true;
}
if (tec == 12){ //enter, C1 es el pin para el bit de inicio
    enter();
}

if( tec <= 9 || tec >= 14 ){ // E y F = 0

    if (n==0){
        centenas = tec;
        pcen = false;
        pdec = true;
        puni = true;
    }
    if (n==1){
        decenas = tec;
        pdec = false;
        puni = true;
    }
    if (n==2){
        unidades = tec;
        puni = false;
    }
    n = n + 1;
    if (n==3){
        n = 0;
    }
}
}else{
    c = c + 1;
    if (c==25){
        boton = false;
        c = 0;
        output_low(pin_a5);
    }
}

```

```
    }  
  }else{  
    if(boton == true){  
      delay_ms(50);  
      boton = false;  
      c = 0;  
    }  
    output_low(pin_a5);  
  }  
}while(true);  
}
```

## 6 Bibliografía

2008 Microchip Technology Inc.  
PIC18F2420/2520/4420/4520 Data Sheet  
<http://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf>

COHERENT, INC.  
GEM-30A, Air-Cooled RF-Excited, OEM Industrial CO2 Laser  
<http://www.coherent.com/downloads/GEM-30A.pdf>

Ing. Diego Alberto Ortega Carreto  
Tutorial Básico Control Puerto Paralelo con Visual Basic 6.0  
<http://www.scribd.com/doc/31339168/Tutorial-Basico-Control-Puerto-Paralelo-con-Visual-Basic-6-0>

monografias.com  
Introducción a los microcontroladores  
<http://www.monografias.com/trabajos12/micrcont/micrcont.shtml>