

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

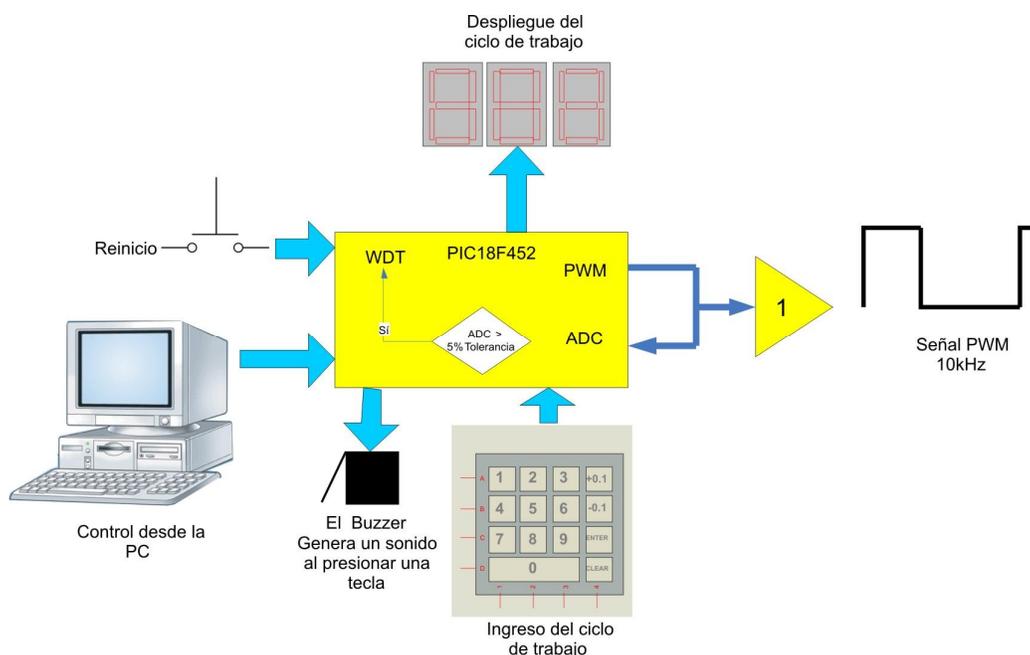


Figura 3. Esquema general del funcionamiento del sistema.

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

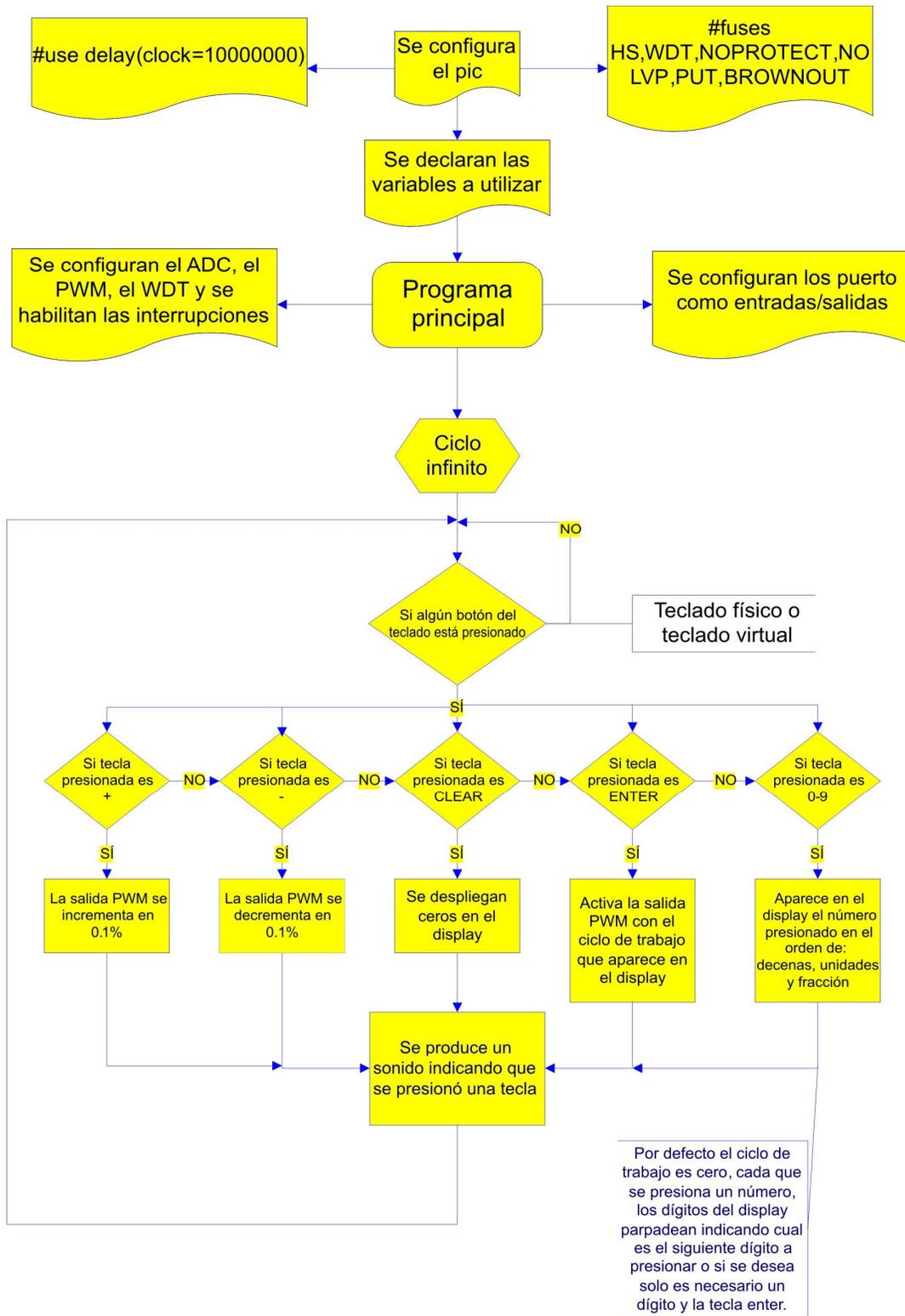


Figura 4. Diagrama de flujo general del sistema.

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.

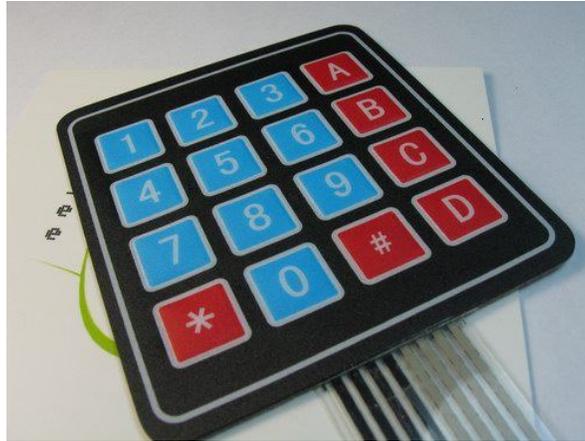


Figura 5. Teclado matricial de 4x4 de membrana.

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.

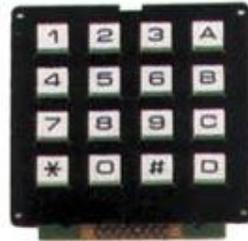


Figura 6. Teclado matricial 4x4 tipo telefónico.

Este es otro teclado que aunque la calidad mejora notoriamente seguimos 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.

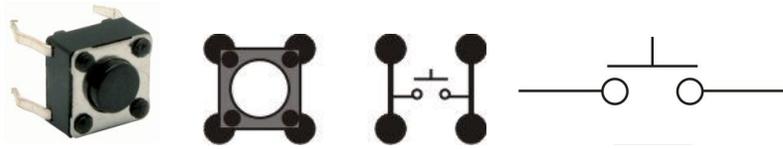


Figura 7. Push button de cuatro terminales tipo SPST.

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.

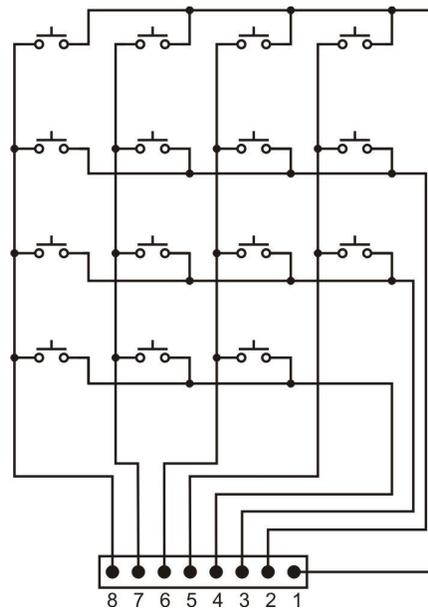


Figura 8. Diseño del teclado matricial.

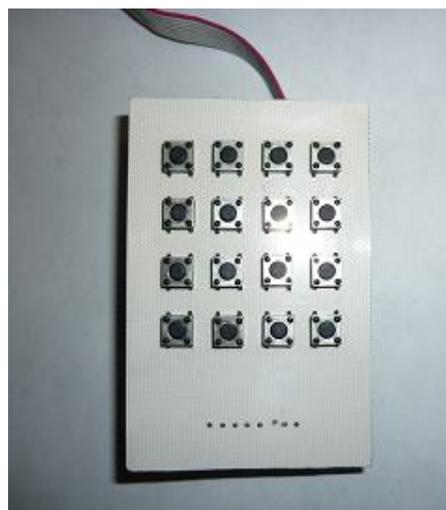


Figura 9. Teclado matricial vista preliminar.

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.

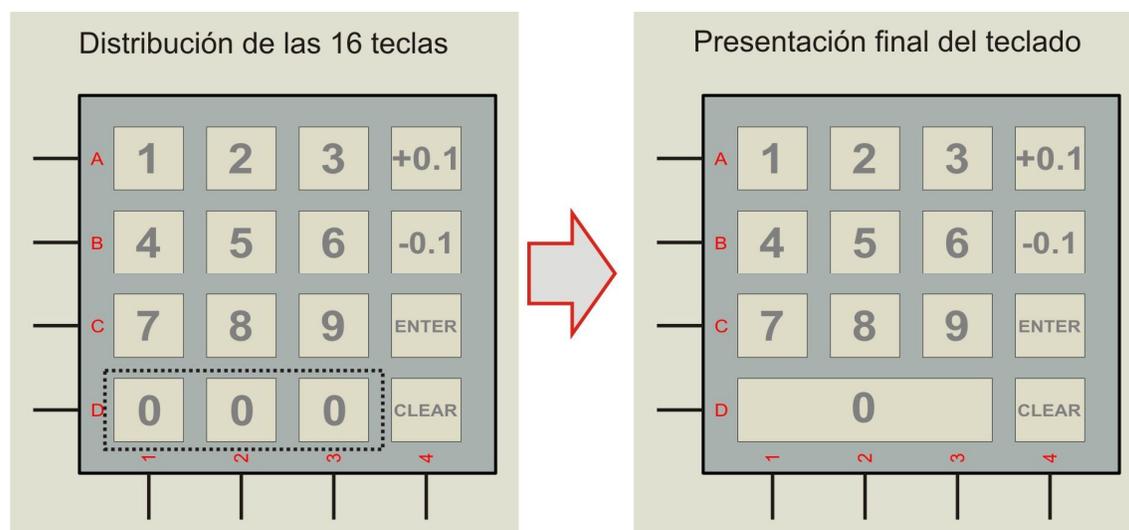


Figura 10. Vista del teclado matricial que estará en el sistema.

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.

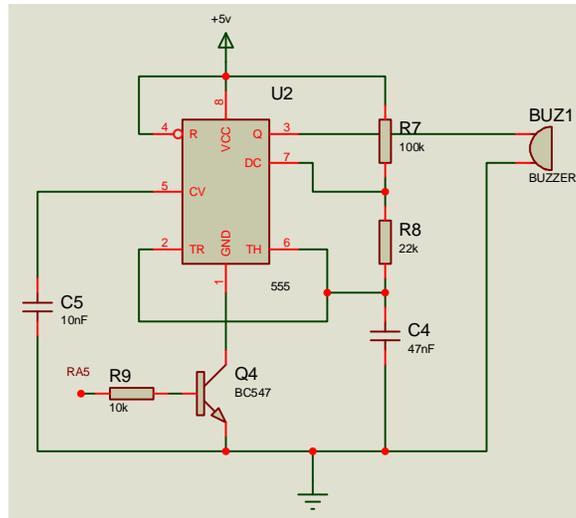


Figura 11. Circuito que crea un sonido al presionar una tecla.

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%.

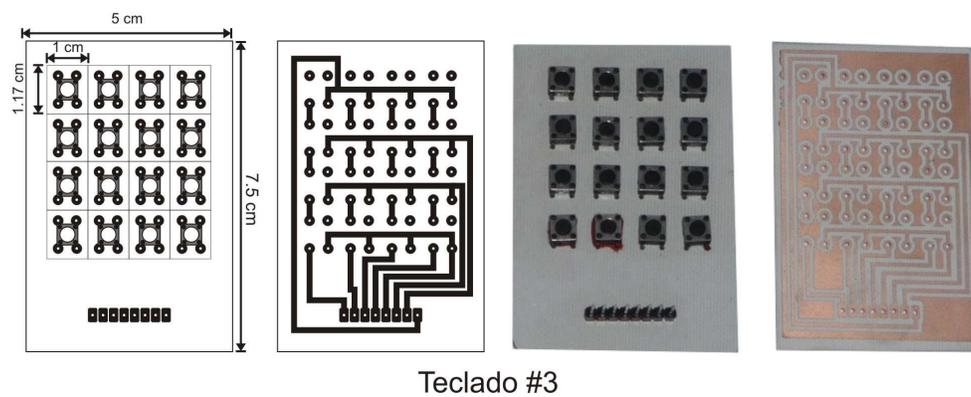
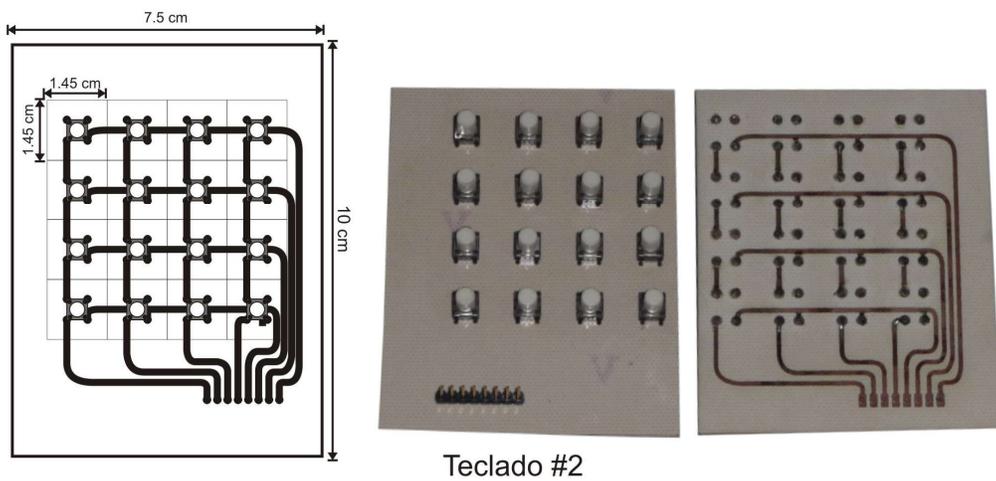
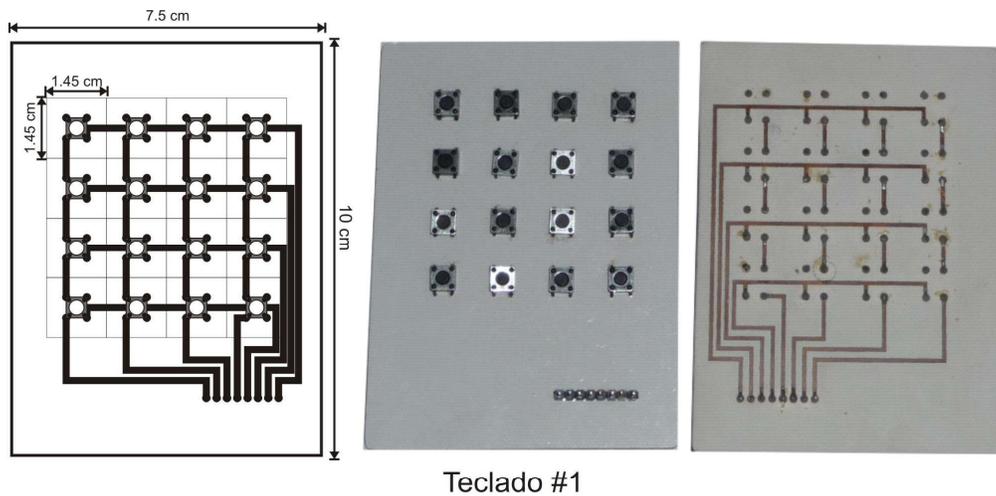


Figura 12. Teclados fabricados.

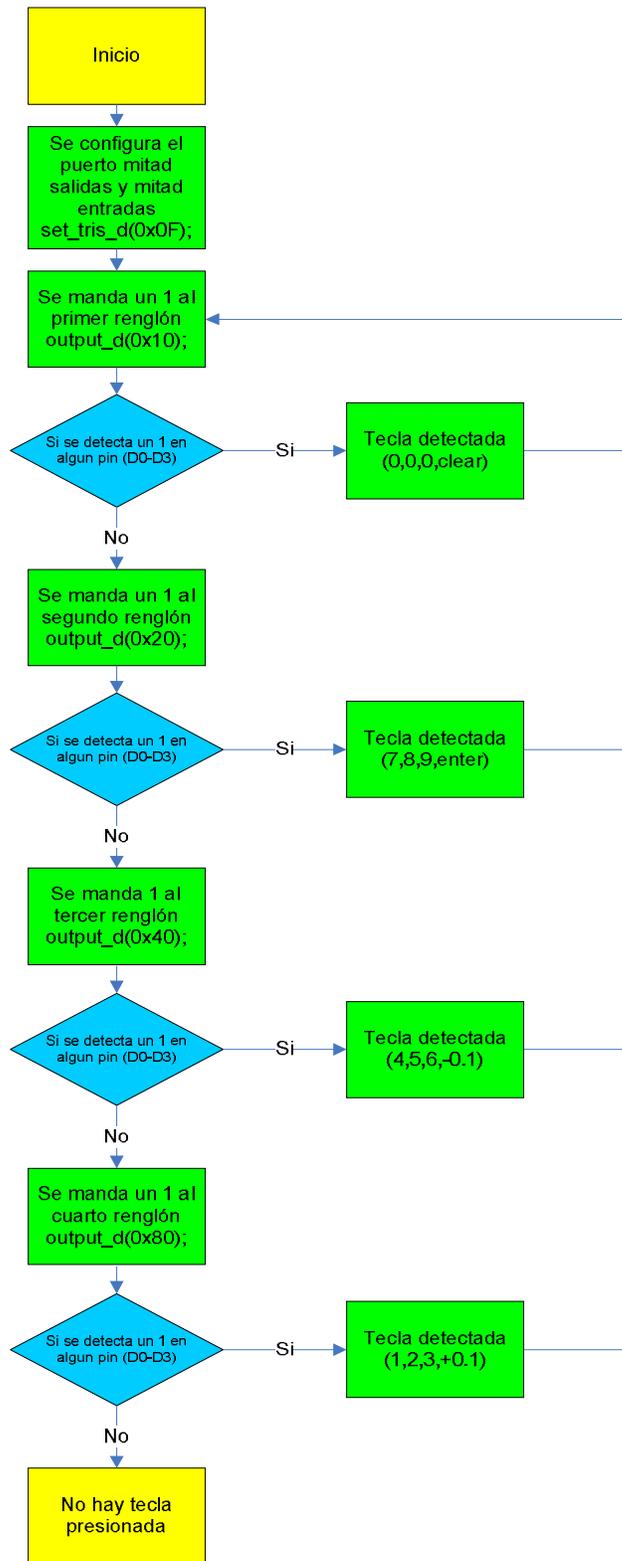
Al concluir las pruebas decidimos que el teclado #1 era el más fácil de usar debido a la separación de sus teclas, además los push button no requieren de tanta presión para activar cada tecla.

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.

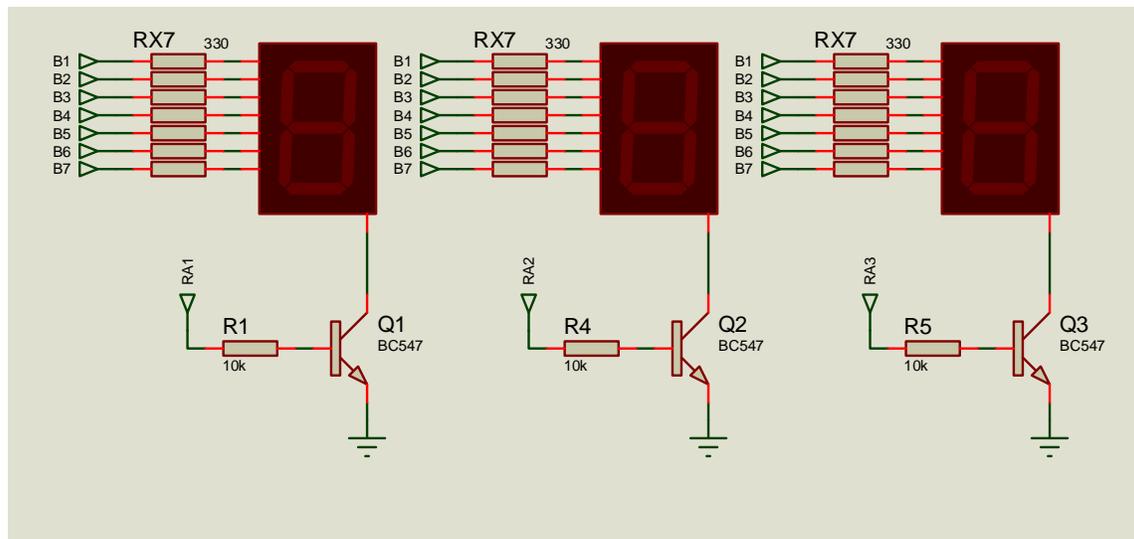


Figura 13. Circuito para el despliegue del ciclo de trabajo.

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.

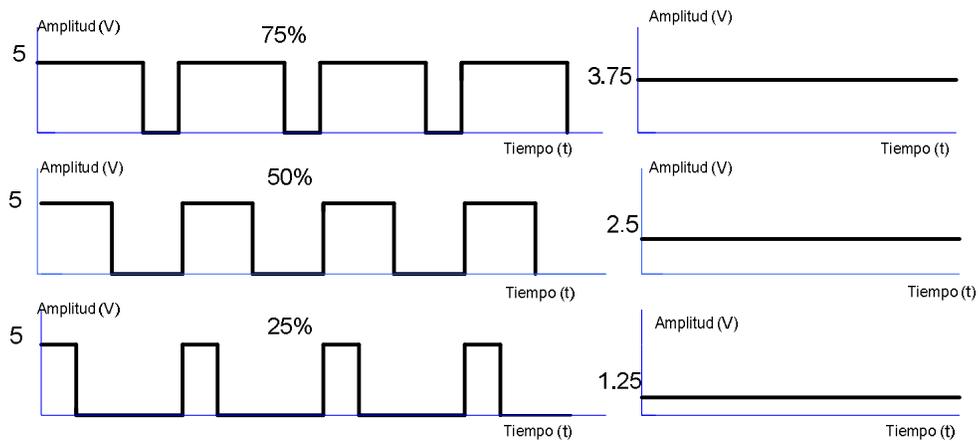


Figura 14. Variación de amplitud respecto al ciclo de trabajo.

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.

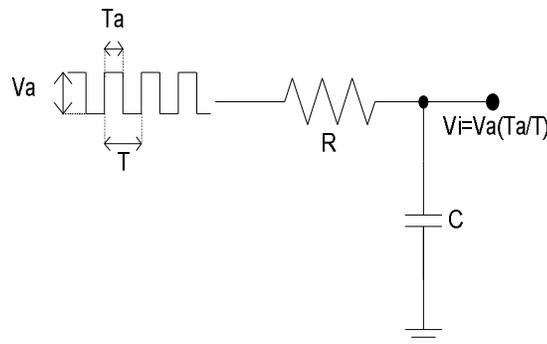


Figura 15. Filtro RC.

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.

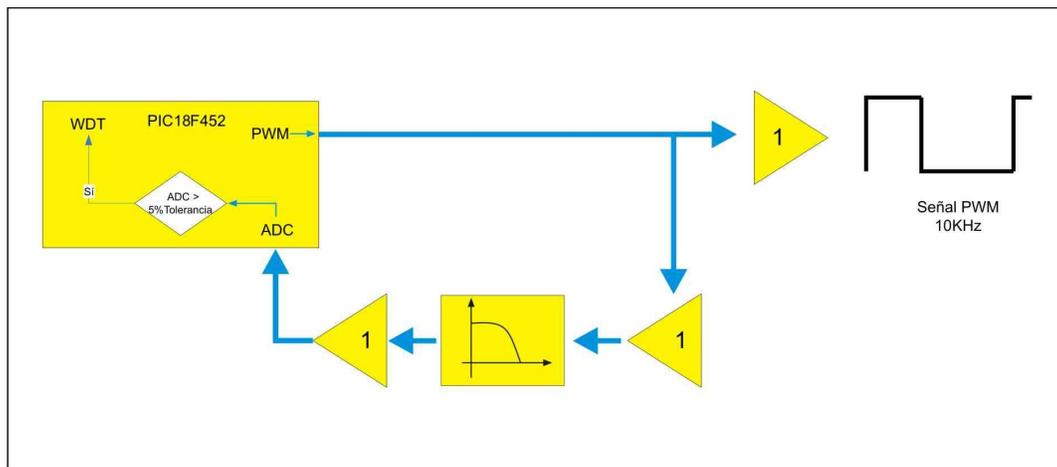


Figura 16. Diagrama del proceso de monitoreo.

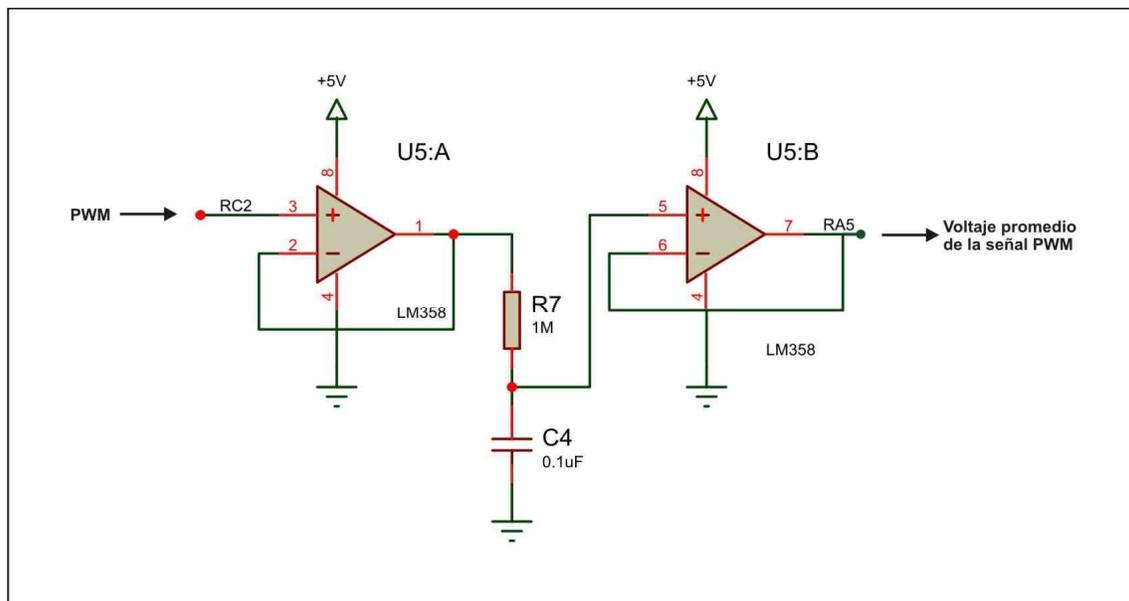


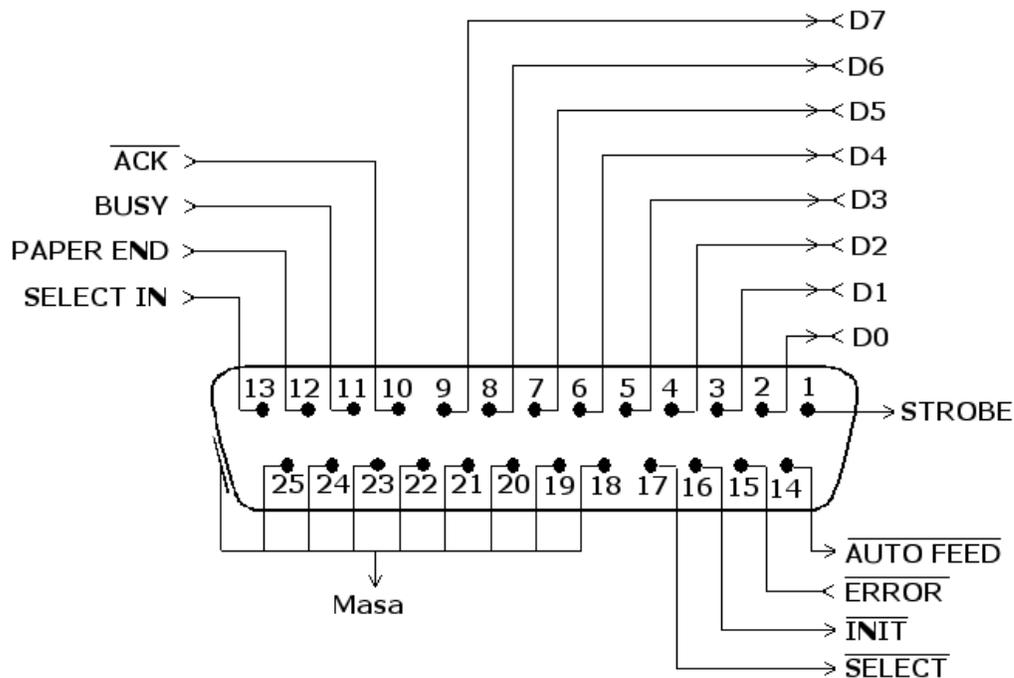
Figura 17. Circuito para realizar el monitoreo.

3.6 Control del ciclo de trabajo mediante la PC

3.6.1 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 lo 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.

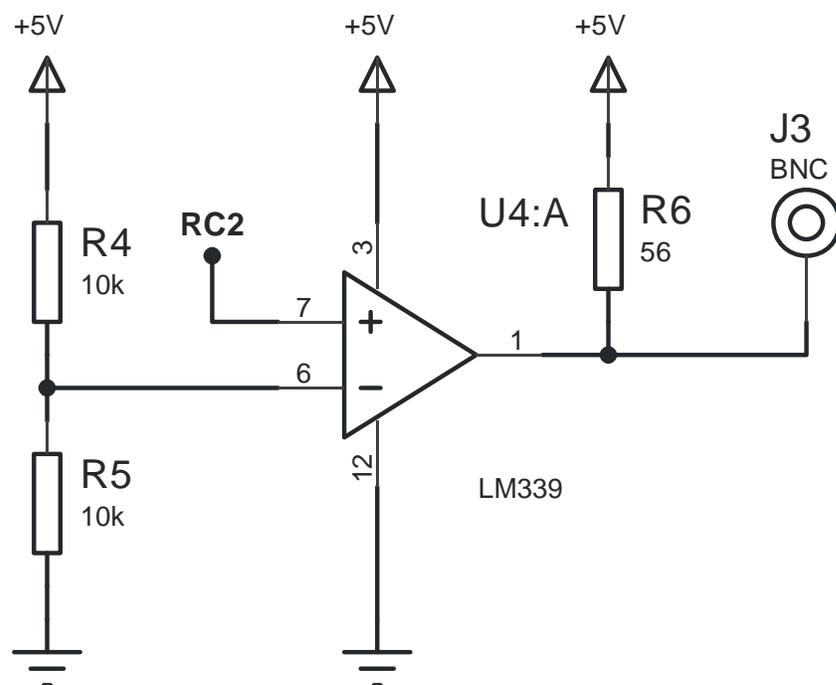


Figura 20. Circuito acondicionador de señal.

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 1.3 μ s y nuestra señal es de 10kHz (100 μ s).

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.

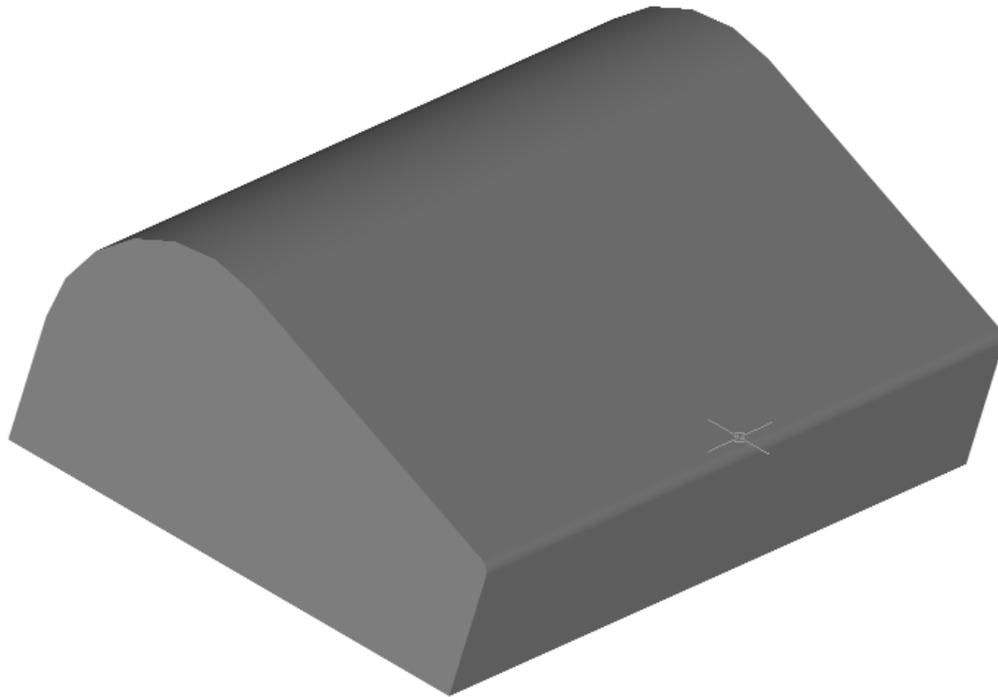


Figura 21. Gabinete de plástico

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 1º 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.

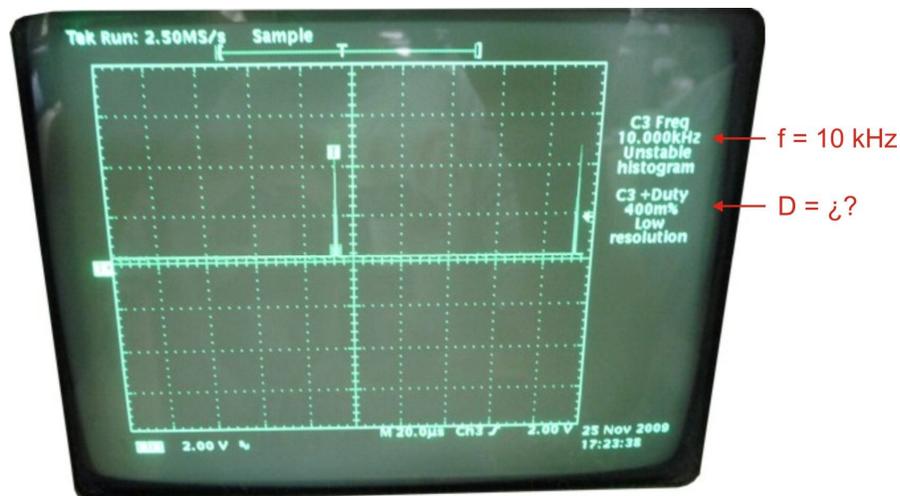


Figura 25. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 0.02 %. Para esta señal el osciloscopio midió una frecuencia de 10 kHz pero no midió su ciclo de trabajo.



Figura 26. Ciclo de trabajo mostrado en el sistema.

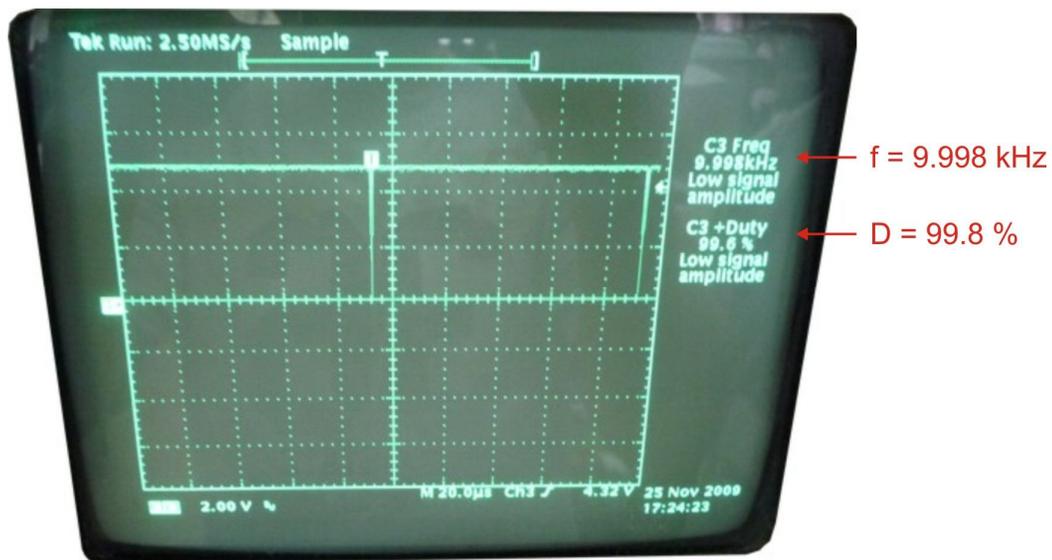


Figura 27. Ciclo de trabajo máximo obtenido. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 99.8 %. Para esta señal el osciloscopio midió una frecuencia de 9.998 kHz y un ciclo de trabajo de 99.8 %.



Figura 28. Ciclo de trabajo mostrado en el sistema.

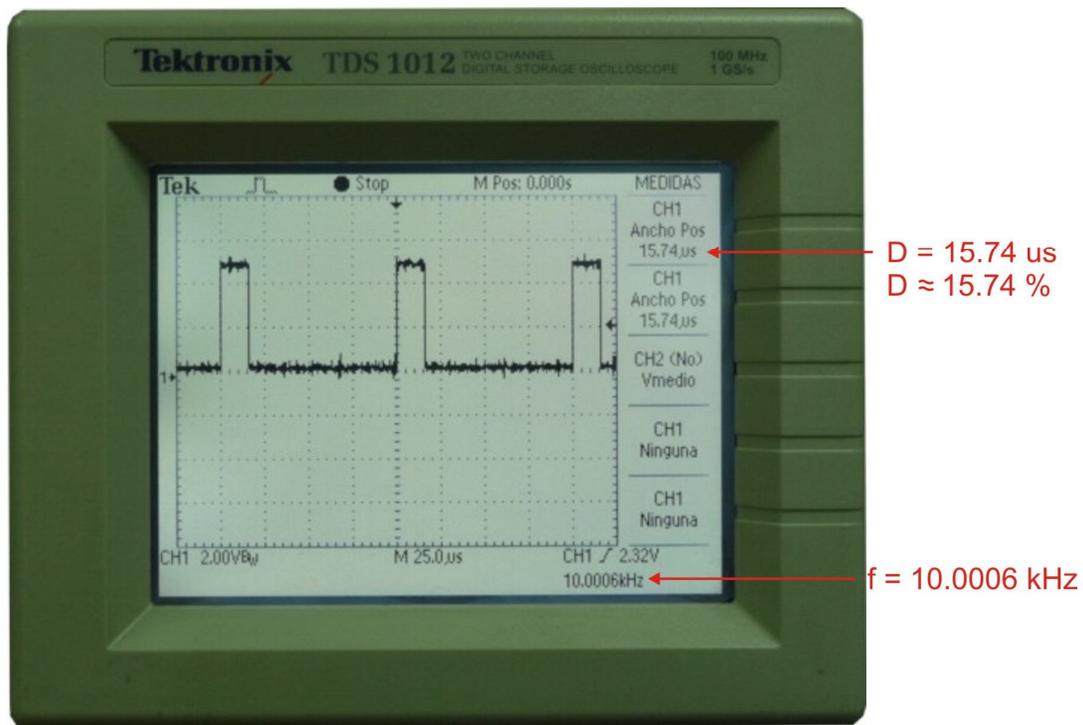


Figura 29. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 15.7 %. Para esta señal el osciloscopio midió una frecuencia de 10.0006 kHz y un ciclo de trabajo de 15.74 µs.



Figura 30. Ciclo de trabajo mostrado en el sistema

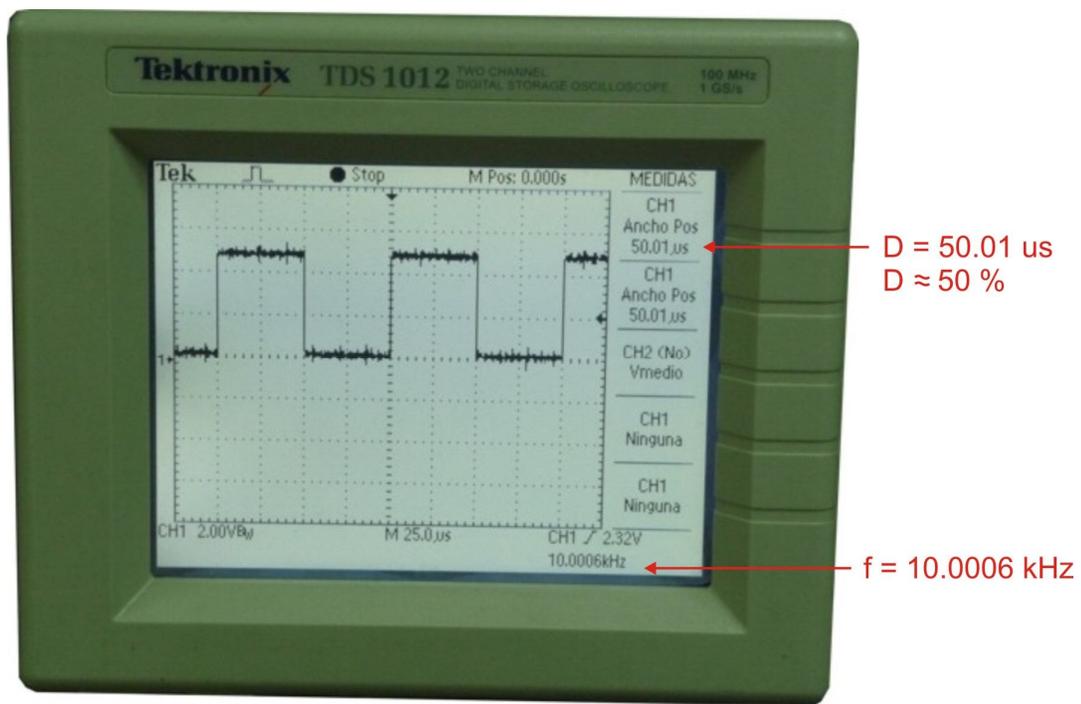


Figura 31. Señal vista en un osciloscopio. Se programó el sistema para generar una señal PWM con ciclo de trabajo de 50 %. Para esta señal el osciloscopio midió una frecuencia de 10.0006 kHz y un ciclo de trabajo de 50 μ s.

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

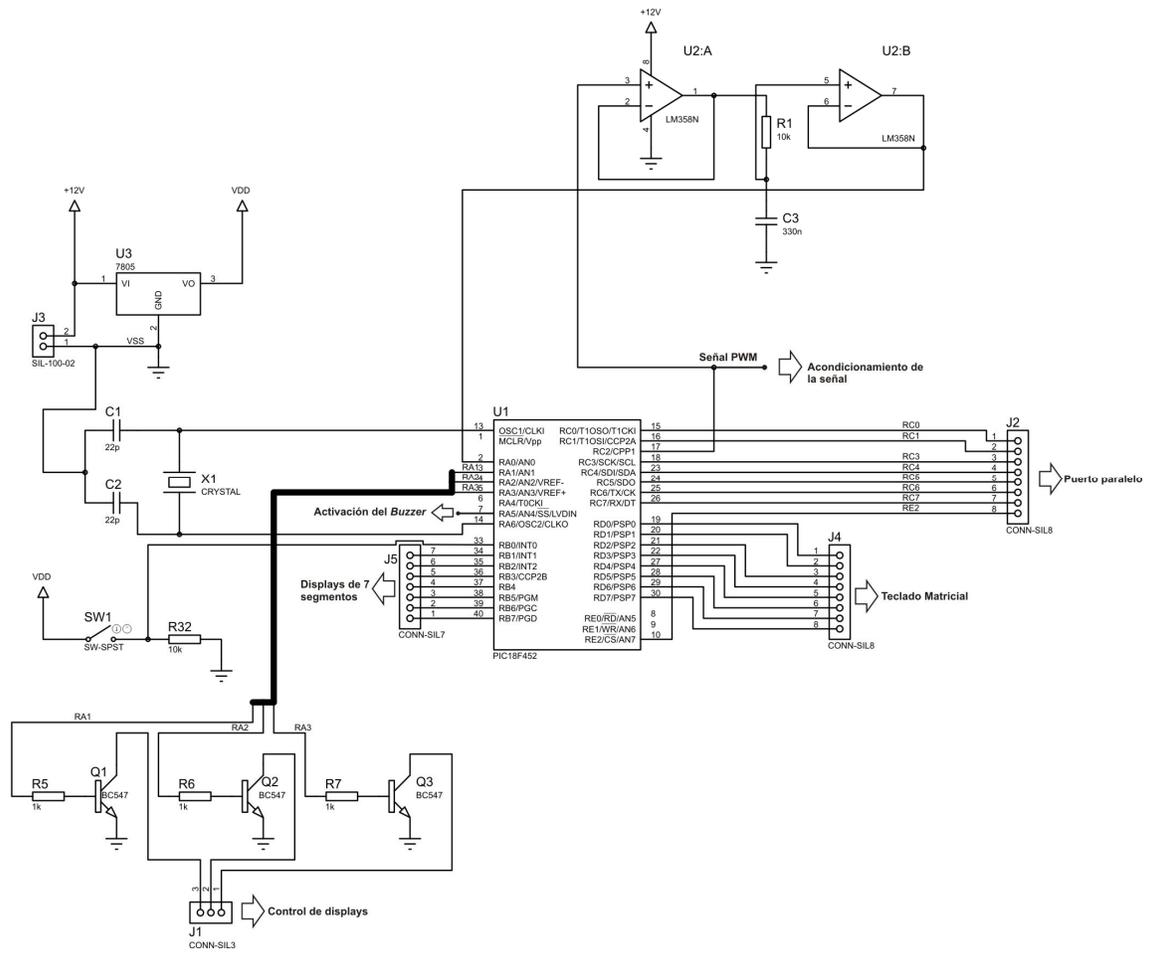


Figura 33. Diagrama eléctrico del circuito impreso principal.

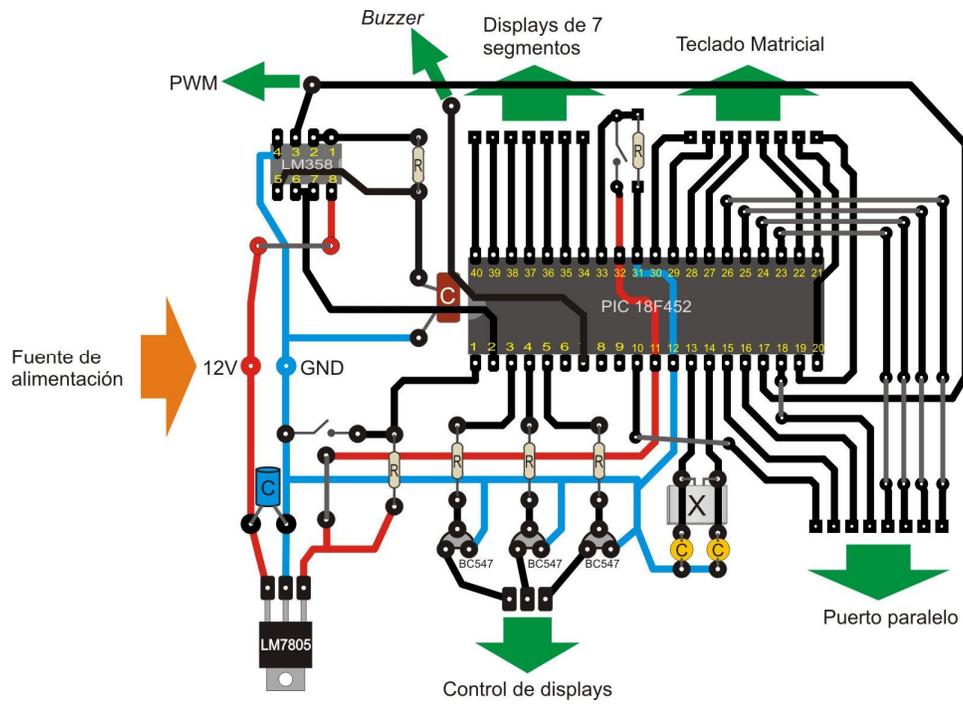


Figura 34. Circuito impreso principal con representación de componentes electrónicos.

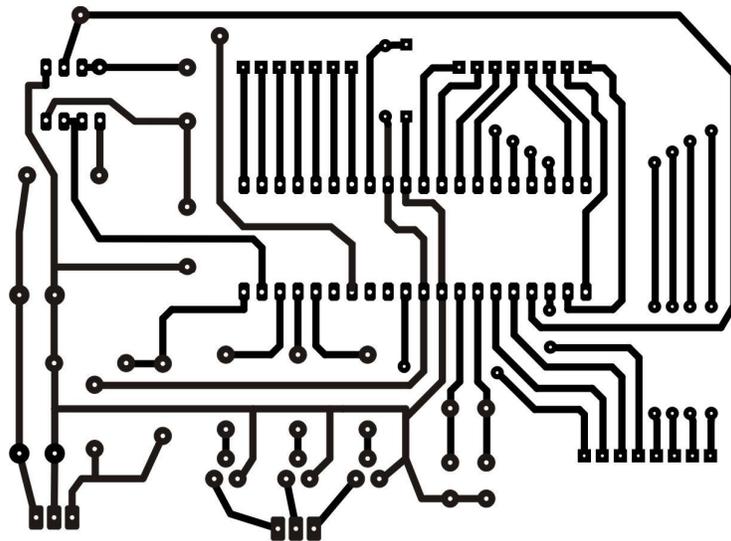


Figura 35. Diseño final del circuito impreso principal.

B) Fuente de alimentación

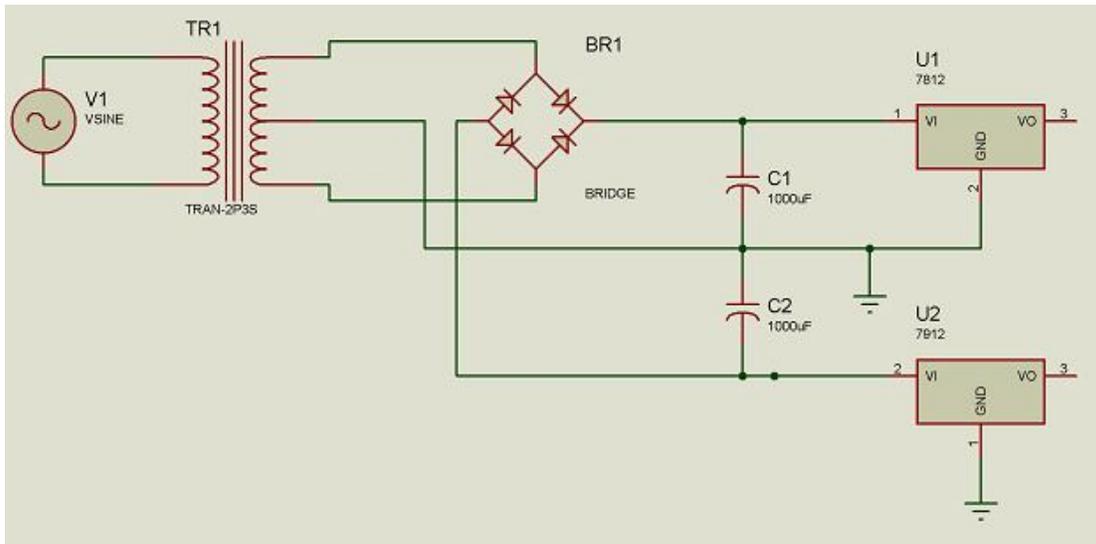


Figura 36. Diagrama eléctrico de la fuente de alimentación.

C) Teclado Matricial

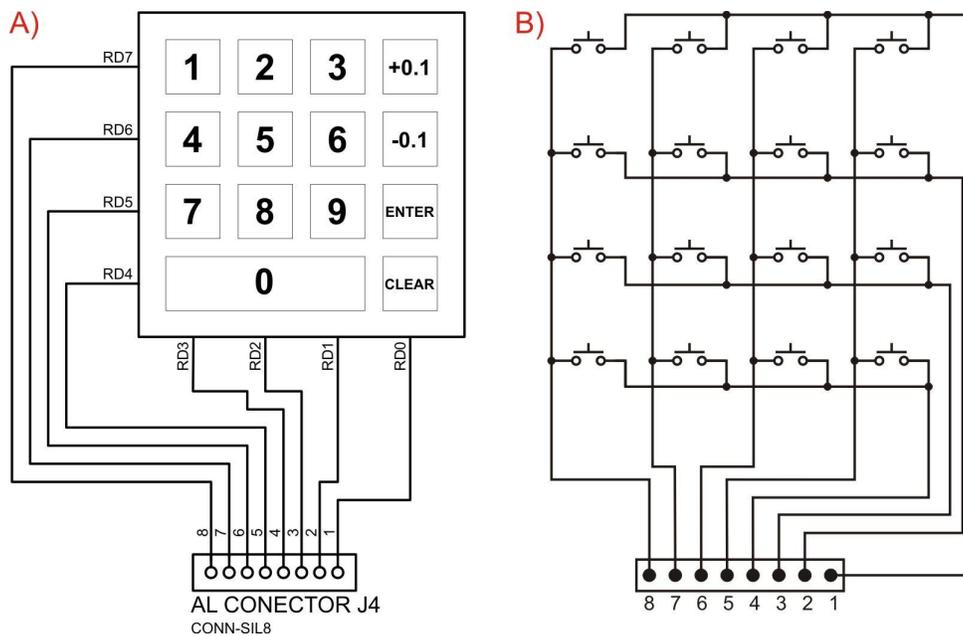


Figura 37. Teclado matricial. A) Distribución de teclas. B) Diagrama eléctrico.

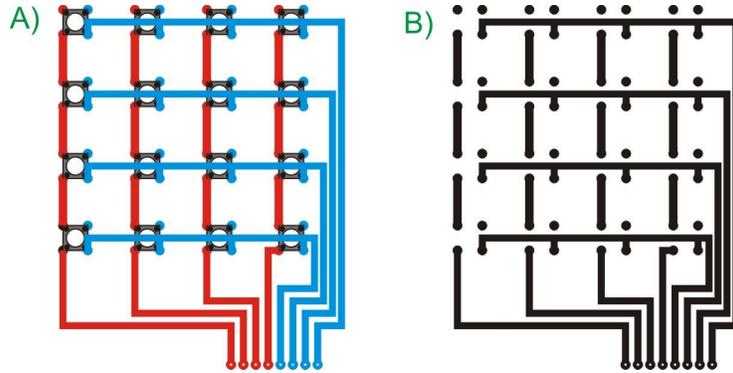


Figura 38. A) Circuito impreso del teclado matricial con representación de componentes electrónicos. B) Diseño final del circuito impreso del teclado matricial.

D) Displays de 7 segmentos

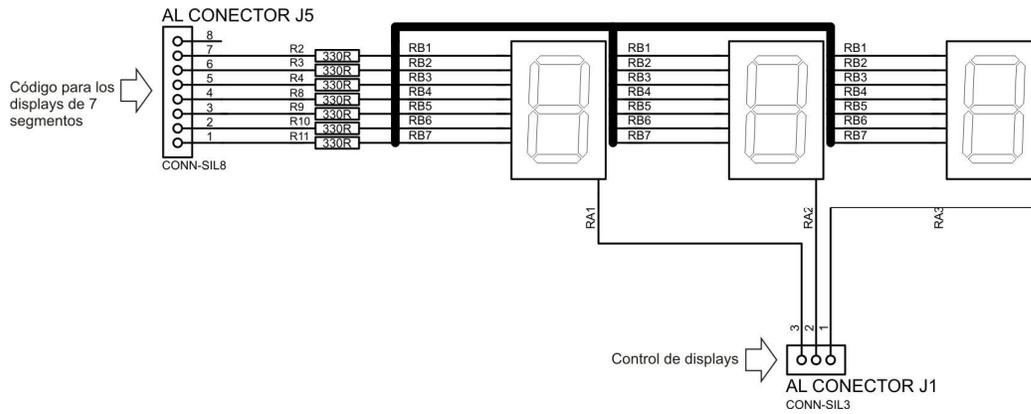


Figura 39. Diagrama eléctrico del módulo de displays.

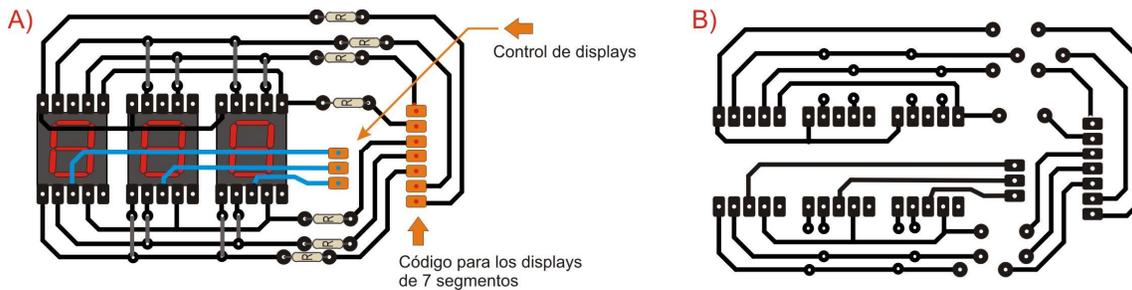


Figura 40. A) Circuito impreso del módulo de displays con representación de componentes electrónicos. B) Diseño final del circuito impreso del módulo de displays.

E) Acondicionamiento de señal y generación de sonido.

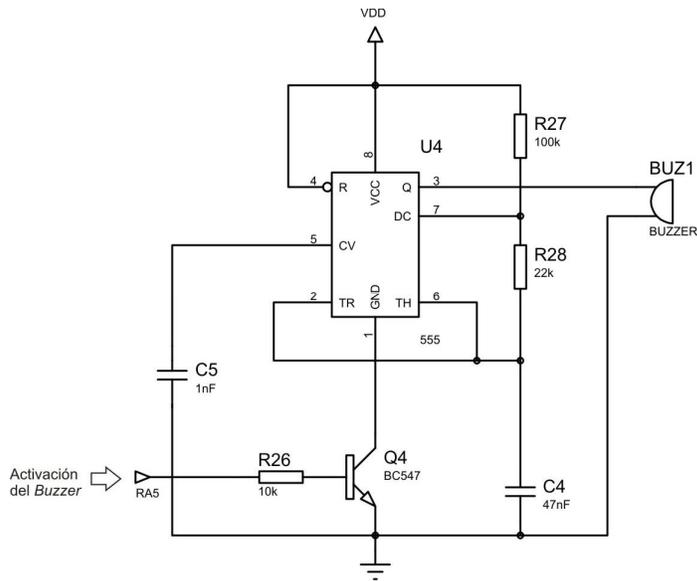


Figura 41. Diagrama eléctrico del módulo de acondicionamiento de señal y generación de sonido.

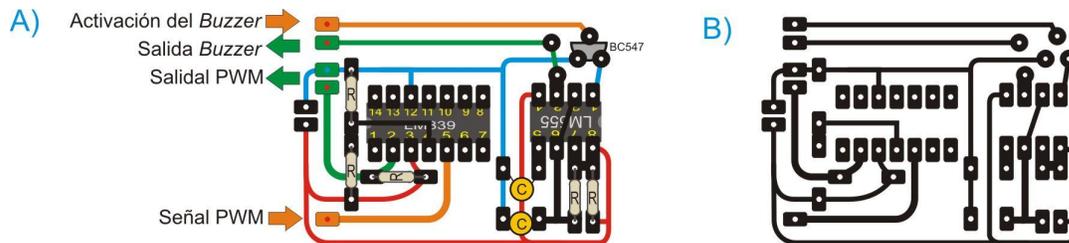


Figura 42. A) Circuito impreso del módulo de acondicionamiento de señal y generación de sonido con representación de componentes electrónicos. B) Diseño final del circuito impreso del módulo de acondicionamiento de señal y generación de sonido.

- 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

SEMICONDUCTORES				
3	BC547	Transistor bipolar NPN	\$ 2.00	\$ 6.00
1	2W02M	Puente rectificador	\$ 5.00	\$ 5.00
RESISTORES				
1	47 Ω	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.10
1	560 Ω	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.10
5	1k Ω	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.50
2	10 k Ω	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.20
1	22 k Ω	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.10
1	100 k Ω	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.20
1	1M Ω	Res. Carbón 1/2 W 5%	\$ 0.10	\$ 0.10
CAPACITORES				
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
OTROS				
1	Transformador con tap 127@12V -0.5 ^a		\$ 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
			TOTAL	\$ 578.80

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.