



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

PROTOTIPO DE PRUEBAS EN CONMUTADORES

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

PARA OBTENER EL TÍTULO PROFESIONAL DE:

INGENIERO EN ELECTRÓNICA

PRESENTA:

RODRIGO HERNÁNDEZ JUÁREZ

NOMBRE DEL ASESOR:

M.C. MARIO EDUARDO LEAL LÓPEZ



APIZACO, TLAX., MARZO 2024



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Apizaco
Electrónica

Apizaco, Tlaxcala, 14/marzo/2023

Departamento: Ingeniería Eléctrica y Electrónica
No. OFICIO IEE/171/2023

Asunto: Liberación de Proyecto
para la Titulación Integral

MARTÍN ROJAS RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

Por este medio me permito informar a usted que ha sido liberado el siguiente proyecto para la Titulación Integral:

Nombre del egresado:	RODRIGO HERNÁNDEZ JUÁREZ
Carrera:	INGENIERÍA ELECTRÓNICA
No. de Control:	14370340
Nombre del proyecto:	"PROTOTIPO DE PRUEBAS EN CONMUTADORES"
Producto:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

Agradeciendo de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados, le envío un cordial saludo.

ATENTAMENTE
*Excelencia en Educación Tecnológica.
Pensar para Servir, Servir para Triunfar.*


MARÍA GUADALUPE MEDINA BARRERA
SUBDIRECTORA ACADÉMICA



MARIO EDUARDO LEAL LÓPEZ	HAYDEE PATRICIA MARTÍNEZ HERNÁNDEZ	RAÚL CORTÉS MALDONADO
		
Nombre y Firma del Asesor	Nombre y Firma del Revisor	Nombre y Firma del Revisor

c.c.p. Expediente



Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec, Municipio de Tzompantepec,
Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010 Ext. 144,
e-mail: electronica@apizaco.tecnm.mx | tecnm.mx | apizaco.tecnm.mx



2023
AÑO DE
Francisco VILA
EL REVOLUCIONARIO DEL NOROCCIDENTE



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Apizaco, Tlax., **12/marzo/2024**
No. OFICIO: DEP-CT/265/2024

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

RODRIGO HERNANDEZ JUAREZ
EGRESADO DE LA CARRERA DE INGENIERÍA ELECTRÓNICA
CON NÚMERO DE CONTROL: 14370340
P R E S E N T E

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** cuyo tema es: **"PROTOTIPO DE PRUEBAS EN CONMUTADORES"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga:

AUTORIZACIÓN DE IMPRESION

Debiendo entregar **3** empastados del mismo, apropiadamente encuadernados, a la brevedad, para presentar su acto de recepción profesional.

Sin otro particular por el momento, le envió un cordial saludo.

ATENTAMENTE

*Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®*


MARTÍN ROJAS RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES



c.p. archivo



Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec, Municipio de Tzompantepec,
Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010 Ext. 304, e-mail:
dep_titulacion@apizaco.tecnm.mx



AGRADECIMIENTOS

A mi madre y a mi padre, por darme todo lo que necesite hasta donde les alcanzo la vida y por dejarme sus costumbres y valores que me ayudaron a no rendirme a pesar de todas las complicaciones que se me presentaron. Hasta donde estén, ¡Gracias!.

A mis hermanas, por ayudarme emocional y económicamente durante todos estos años, sin su aportación, estar en este momento no hubiera sido posible.

A mi prometida, por estar conmigo cuando seriamente pensé en abandonar mi formación académica.

A mis compañeros y amigos del tecnológico, que saben lo complicado que fue la carrera y por los momentos de diversión que vivimos.

Al Ingeniero Emmanuel Paul Camacho, por permitirme realizar mi residencia profesional en su distinguida empresa y asesorarme en todo momento.

Al Instituto Tecnológico de Apizaco, por darme los conocimientos y la experiencia que llevare por el resto de mi vida. Los colores del instituto se quedan grabados por siempre.

ÍNDICE

CAPITULO I INTRODUCCION	1
CAPITULO II DESCRIPCIÓN DE LA EMPRESA	2
CAPITULO III PROBLEMAS A RESOLVER	3
CAPITULO IV OBJETIVOS 4.1 Objetivo general 4.2 Objetivo especifico	4 4 4
CAPITULO V JUSTIFICACIÓN	4
CAPITULO VI MARCO TEÓRICO 6.1 Conmutador 6.2 Microcontrolador 6.3 DTMF 6.4 C 6.5 Compilador CCS 6.6 Proteus Design Suite 6.7 Pickit 2 6.8 PIC16F88 6.9 74LS164 6.10 7805 6.11 Condensador electric 6.12 Resistor 6.13 LCD 16x2 6.14 Relevador	5 5 5 5 5 5 6 6 6 6 7 7 7 8
CAPITULO VII PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS 7.1 Análisis 7.2 Programación de un PIC16F88 que simula el Modulo MT8870 7.3 Programa principal	8 8 9 16

7.4 Diseño en software Proteus	23
7.5 Pruebas en protoboard	25
7.6 Diseño de PCB	27
7.7 Ensamble Final	29
CAPITULO VIII	
RESULTADOS	31
8.1 Programa	31
8.2 Simulación en software Proteus	38
8.3 Prototipo concluido	39
CAPITULO IX	
CONCLUSIONES	40
CAPITULO X	
COMPETENCIAS DESARROLLADAS Y/O APLICADAS	41
CAPITULO XI	
FUENTES DE INFORMACIÓN	42
CAPITULO XII	
ANEXOS	

CAPITULO I

INTRODUCCIÓN

El proyecto “Prototipo de Pruebas en Conmutadores” tomado del banco de proyectos de la empresa Dipag Soluciones, me fue asignado debido a la problemática que hasta antes de concluirlo se enfrentaba la compañía. Y es que era necesario desarrollar un dispositivo capaz de desempeñar las mismas operaciones con el mismo estándar de calidad, eficiencia y eficacia de un equipo que se encuentra en el mercado, pero con un costo de producción mucho menor

En este documento, informo a detalle la metodología que emplee para desarrollar el prototipo. Inicio describiendo a la empresa que me dio la oportunidad de realizar mi residencia profesional, enumero cada uno de los problemas que se presentaron antes, durante y después de concluir el prototipo, defino los objetivos generales y específicos y en la parte de “justificación” menciono la razón del porque se realizó el prototipo y a quien va a beneficiar.

Como Marco Teórico, concentro todos los conceptos teóricos que me permitieron partir de cero en ese trabajo, cito basándome en fuentes bibliográficas y digitales todos los elementos que use.

En el desarrollo describo las actividades que realice y como resultado muestro mi programa, las simulaciones, el diseño del PCB y el prototipo con su imagen final.

Como conclusión menciono las recomendaciones y la experiencia profesional que me dejo trabajar en esto.

En la parte final del documento, cito las fuentes de información que use, así como las competencias desarrolladas.

CAPITULO II

DESCRIPCIÓN DE LA EMPRESA

Dipag Consultores es una empresa ubicada en Prolongación Linda Vista No. 1, Tlatempan, Apetatiltan de Antonio Carvajal, Tlaxcala.

Fundada el 8 de enero del 2012 por el Ingeniero Emmanuel Paul Camacho, quien egreso del Instituto Tecnológico de Apizaco en 2011.



FIG.1 Logotipo de la Empresa Recuperado de:
<https://dipag.com.mx/>

La empresa funge como desarrolladora de proyectos de ingeniería y administrativos, a diferentes clientes, sean particulares o empresas.

Es sabido que la empresa se caracteriza por dar la oportunidad de realizar la residencia profesional a futuros ingenieros de todo el país.

Fui asignado en el área de desarrollo de proyectos y estuve bajo la supervisión directa del Ingeniero Emmanuel Paul Camacho. En esta área, puse en práctica los conocimientos de programación de microcontroladores y electrónica digital que adquirí en el instituto, en el proceso y bajo la asesoría del ingeniero, aprendí a optimizar y minimizar los recursos al usar los elementos necesarios.

CAPITULO III

PROBLEMAS A RESOLVER

1. El problema principal es desarrollar un equipo que sea capaz de probar la condición en la que se encuentran los conmutadores

Si bien en el mercado hay equipos capaces de realizar esta operación, los costos de ellos rebasan la capacidad de adquisición de algunos clientes, como ejemplo puedo mencionar al popular Compa Tempo® que tan solo en el mercado de segunda mano rebasan los siete mil pesos.

2. El prototipo requiere que sea capaz de probar hasta ocho líneas.
3. Se debe realizar con microcontroladores y por cuestiones de miniaturización, se requiere que se emplee el PIC16F88, lo que nos limita a solo dos puertos.
4. Se debe utilizar el módulo MT8870, lo que indica que se debe controlar de forma remota mediante tonos DTMF

CAPITULO IV

OBJETIVOS

4.1 Objetivo General: Diseñar, fabricar e implementar un prototipo probador de conmutadores basado en tecnologías DTMF y microcontroladores, costeable, de calidad y eficiente.

4.2 Objetivo específico: Con microcontroladores, decodificadores de tonos MT8870 como principales elementos electrónicos, se busca diseñar un sistema automático de pruebas en conmutadores, que será capaz de ser controlado a la distancia.

CAPITULO V

JUSTIFICACIÓN

Para quien busque una alternativa de menor costo, pero con la misma calidad que se espera de un dispositivo comercial, existe la necesidad de desarrollar el prototipo de pruebas en conmutadores aplicando tecnologías de microcontroladores, elaborando un software desde cero y empleando hardware eficazmente, el resultado será un equipo capaz de detectar la falla y localización de la misma en el sistema, evitando daños serios que significarían pérdidas económicas importante para el cliente.

CAPITULO VI

MARCO TEÓRICO

Es importante mencionar las terminologías que hacen referencia al proyecto, así como los dispositivos electrónicos que conforman el prototipo y las herramientas digitales utilizadas.

6.1 Conmutador, es un sistema de telecomunicaciones utilizado en la red telefónica pública conmutada o en empresas para interconectar circuitos de teléfonos con el fin de establecer llamadas telefónicas entre los usuarios, o entre otras centrales.

6.2 Microcontrolador, un microcontrolador es un circuito integrado programable que contiene todos los componentes necesarios para controlar el funcionamiento de una tarea determinada, como el control de una lavadora, un teclado de ordenador, una impresora, un sistema de alarma, etc. Palacios, Remiro y López (2009).

6.3 DTMF, (Dual-Tone Multi-Frequency) es usado para la señalización de telecomunicaciones sobre líneas telefónicas analógicas en la banda de frecuencia vocal entre teléfonos u otros equipos de comunicaciones y la central telefónica.

6.4 C, es un lenguaje de programación de propósito general que ofrece como ventajas economía de expresión, control de flujo y estructuras de datos modernos y un rico conjunto de operaciones. Kernighan y Ritchie (1991).

6.5 Compilador CCS, ha sido desarrollado específicamente para PIC MCU, obteniendo la máxima optimización del compilador con estos dispositivos. Dispone de una amplia librería de funciones predefinidas, comandos de preprocesado y ejemplos. García (2008).

6.6 Proteus Design Suite, es un software de automatización de diseño electrónico, desarrollado por Labcenter Electronics Ltd., que consta de los dos programas principales: Ares e Isis, y los módulos VSM y Electra.

6.7 Pickit 2, Programador, Emulador, Debugger, es un programador (también conocido como grabador o quemador) diseñado por Microchip para la programación de una gran variedad de dispositivos.

6.8 PIC16F88, El PIC16F88 tiene un oscilador interno de 8MHz, 256 bytes de memoria de datos EEPROM, una función de captura / comparación / PWM, un USART direccionable, un puerto serie síncrono que se pueden configurar como interfaz de bus de 3 hilos (SPI) o de 2 hilos (I²C), 7 canales de 10 bits (A/D) y 2 que lo hacen ideal para la grabación analógica.



FIG 2. PIC16F88 Recuperado de:
<https://www.microchip.com/wwwproducts/en/PIC16F88>

6.9 74LS164, es un circuito digital secuencial (es decir, que los valores de sus salidas dependen de sus entradas y de los valores anteriores) consistente en una serie de biestables, generalmente de tipo D, conectados en cascada, que basculan de forma sincrónica con la misma señal de reloj. Según las conexiones entre los biestables, se tiene un desplazamiento a la izquierda o a la derecha de la información almacenada. Es de señalar que un desplazamiento a la izquierda de un conjunto de bits, multiplica por 2, mientras que uno a la derecha, divide entre 2. Existen registros de desplazamiento bidireccionales, que pueden funcionar en ambos sentidos.

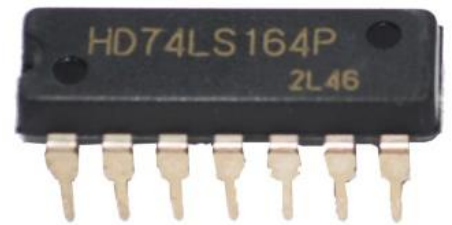


FIG. 3 Integrado 74ls164 Recuperado de:
<http://www.geekbotelectronics.com/producto/74ls164-registro-de-desplazamiento/>

6.10 7805, Es un dispositivo electrónico que tiene la capacidad de regular voltaje positivo de 5V a 1A de corriente.



FIG.4 Integrado 7805 Recuperado de:
<https://www.indiamart.com/proddetail/7805-voltage-regulator-21182453248.html>

6.11 Condensador eléctrico, es un dispositivo pasivo, utilizado en electricidad y electrónica, capaz de almacenar energía sustentando un campo eléctrico. Senner (1994).



FIG 5. Diferentes tipos de condensadores
Recuperado de: <https://www.amateur-radio-wiki.net/capacitor/>

6.12 Resistor diseñado para introducir una resistencia eléctrica determinada entre dos puntos de un circuito eléctrico. En otros casos, como en las planchas, calentadores, etc., se emplean resistencias para producir calor aprovechando el efecto Joule. Es un material formado por carbón y otros elementos resistivos para disminuir la corriente que pasa. Se opone al paso de la corriente. Senner (1994).



FIG. 6 Resistor Recuperado de:
<https://learn.sparkfun.com/tutorials/resistors/all>

6.13 LCD 16x2, se refiere a un pequeño dispositivo con pantalla de cristal líquido que cuenta con dos filas, de dieciséis caracteres cada una, que se utiliza para mostrar información, por lo general alfanumérica. Las capacidades de estos dispositivos son altas, pues se puede mostrar todo tipo de información sin importar qué tipo de símbolos o caracteres sean, el idioma o el lenguaje,



FIG. 7 LCD 16x2 Recuperado de:
<https://www.iberobotics.com/producto/pantalla-lcd-16x2/>

pues el sistema puede mostrar cualquier carácter alfanumérico, símbolos y algunas figuras, el número de píxeles que tiene cada símbolo o carácter varía dependiendo del modelo del dispositivo y cada artefacto está controlado por un microcontrolador que está programado para dirigir el funcionamiento y la imagen mostrada en la pantalla.

6.14 Relevador, es un dispositivo electromagnético. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.



FIG. 8 Relevador Recuperado de:
<https://robotec.mx/products/relevador-5v/>

CAPITULO VII

PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

7.1 Análisis

Identificación del problema, una vez presentada la idea del prototipo, surgieron las primeras preguntas: ¿Qué materiales voy a utilizar?, ¿En qué lenguaje se va a programar el microcontrolador?, por lo que se tuvo una reunión donde se aclararon estas y más dudas que tenía del proyecto. Para hacer un dispositivo de dimensiones pequeñas y sea ergonómico para el usuario, decidimos que los elementos electrónicos empleados debían ser pequeños, pero con funciones suficientes para realizar las tareas propias del proyecto.

Se eligió el PIC16F88 porque cumplía el requisito de ser pequeño, además de poseer las siguientes características:

Arquitectura: Harvard, memoria de código de 14 bits, separada de la memoria de datos de 8 bits.

Tecnología: RISC (reduced instruction set computer), con 35 instrucciones.

Memoria: 4K localidades (14 bits) de FLASH, 368 localidades (8 bits) de RAM, 256 localidades (8 bits) de EEPROM.

Puertos digitales: puerto A de 5 bits, puerto B de 8 bits, un total de 13 bits programables como entradas o como salidas.

Capacidad de salidas: cada bit de salida puede tomar, o generar, hasta 25mA.

Puertos seriales: USART compatible RS232. SSP Puerto serial síncrono con 2 modos de operación: **SPI** (Serial Peripheral Interface, modos Master/Slave) e **I2C** (Integrated, Integrated Circuit. Modo Slave)

Convertidores A/D: 7 canales, con 10 bits de resolución.

Funciones adicionales: power-on reset, brown out reset, power up timer, watch dog, code protection , sleep (bajo consumo).

Temporizadores: 3 temporizadores. Un generador de PWM

Sin embargo, el hecho de solo contar con 18 pines significaría un problema, ya que el uso de una pantalla LCD 16x2 era obligatorio y usarla de modo convencional en el puerto b, agotaría la posibilidad de utilizarlo para las demás operaciones.

La solución fue utilizar un registro de desplazamiento 74LS164, con la ayuda de este dispositivo, únicamente serian usados tres pines para controlar la pantalla LCD 16x2.

El control del proyecto se llevaría a cabo mediante tonos DTMF, por ello utilice el modulo decodificador de tonos MT8870, que integra tanto el filtro divisor de banda y las funciones de decodificación digital.

El uso de este módulo, requiere cuatro pines del microcontrolador lo que limitaba los pines para salidas del prototipo.

Utilice 2 registros de desplazamiento 74LS164 mas para poder ampliar las salidas, por lo que el PIC 16F88 si fue suficiente.

La identificación del problema y el análisis de los elementos a utilizar forman parte de la primera actividad

7.2 Programación de un PIC16F88 que simula el Modulo MT8870

En el software Proteus, no existe hasta la fecha un elemento que pueda simular el decodificador de tonos MT8870, por tal motivo y para fines meramente didácticos, hice un programa que realiza las funciones del módulo real, use el PIC16f88 ya que

tiene los pines suficientes y al también usarlo como microcontrolador principal, lo tenía a la mano para que en caso de ser necesario, hiciera pruebas físicas. Al tratarse de estados lógicos donde cada tecla del teclado matricial representa una combinación binaria, no fue de gran problema hacerlo, lo presento a continuación:

Programa que simula el decodificador de tonos DTMF en Proteus

```
#include <16f88.h>
#fuses xt,nowdt
#use delay(clock=4M)
#define USE_PORTB_KBD
#include <KBD.C>
#use standard_io(a)
#use standard_io(b)
char k;

void tecla1()
{
    output_a(0b0001);
}

void tecla2()
{
    output_a(0b0010);
}

void tecla3()
{
    output_a(0b0011);
}

void tecla4()
{
    output_a(0b0100);
}

void tecla5()
{
    output_a(0b1001);
}

void tecla6()
{
    output_a(0b0110);
}

void tecla7()
{
    output_a(0b0111);
}
```

```

void tecla8()
{
    output_a(0b1000);
}

void tecla9()
{
    output_a(0b1001);
}

void tecla()
{
    output_a(0b1010);
}

void tecla*()
{
    output_a(0b1011);
}

{
    kbd_init();
    while (true)    {
        k=0;
        k=kbd_getc();
        output_a(0b0000);
        while(k=='1')
        {
            tecla1();
        }
        while(k=='2')
        {
            tecla2();
        }
    }
}

```

```

while(k=='3')
{
    tecla3();
}
while(k=='4')
{
    tecla4();
}
while(k=='5')
{
    tecla5();
}
while((k=='6'))
{
    tecla6();
}
while(k=='7')
{
    tecla7();
}
while(k=='8')
{
    tecla8();
}
while(k=='9')
{
    tecla9();
}
while(k=='0')
{
    tecla0();
}
while(k=='*')
{
    tecla*();
}
while(k=='#')
{
    tecla#();
}
}

```

Teniendo ya el MT8870, las simulaciones posteriores serían mucho más eficientes y cercanas a la realidad. Realice una simulación donde hago uso del programa antes mostrado para verificar su correcto funcionamiento.

En la siguiente imagen tomada de Proteus, se observa el módulo MT8870 simulado en el PIC16F88, un teclado matricial, un PIC16F88 encargado de realizar la función principal, que en este caso es encender el Diodo Emisor de Luz al momento que se oprime la tecla correspondiente.

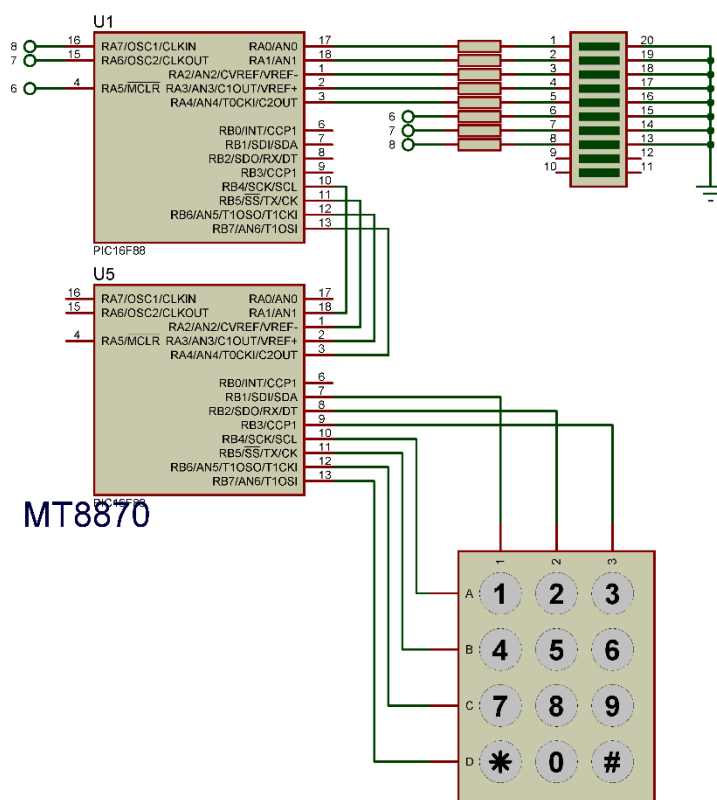


FIG. 9 Simulación del módulo MT8870 Creación propia

La siguiente tabla muestra las diferentes combinaciones posibles del módulo real, posteriormente muestro el programa que realice para su comprobación.

Digit	TOE	INH	ES _t	Q ₄	Q ₃	Q ₂	Q ₁
ANY	L	X	H	Z	Z	Z	Z
1	H	X	H	0	0	0	1
2	H	X	H	0	0	1	0
3	H	X	H	0	0	1	1
4	H	X	H	0	1	0	0
5	H	X	H	0	1	0	1
6	H	X	H	0	1	1	0
7	H	X	H	0	1	1	1
8	H	X	H	1	0	0	0
9	H	X	H	1	0	0	1
0	H	X	H	1	0	1	0
*	H	X	H	1	0	1	1
#	H	X	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0
A	H	H	L	undetected, the output code will remain the same as the previous detected code			
B	H	H	L				
C	H	H	L				
D	H	H	L				

TAB.1 *Tabla de funcionamiento lógico del módulo MT8870* Recuperado de:
https://www.microsemi.com/document-portal/doc_download/127041-mt8870d-datasheet-oct2006

Programa de comprobación del módulo MT8870

```
#include <16F88.h>
#use delay(clock=8000000)
#fuses XT,NOWDT,NOMCLR
#byte puerto_b=06
#byte puerto_a=05

void main() {

set_tris_a(0x00);
set_tris_b(0xF0);
```

```

while(true)
{
if (input(PIN_B4)==false && input(PIN_B5)==false && input(PIN_B6)==false && input(PIN_B7)==false)
{
puerto_a=0b00000000;
}
if (input(PIN_B4)==true && input(PIN_B5)==false&& input(PIN_B6)==false && input(PIN_B7)==false)
{
puerto_a=0b00000001;
}
if (input(PIN_B4)==false && input(PIN_B5)==true&& input(PIN_B6)==false && input(PIN_B7)==false)
{
puerto_a=0b00000010;
}
if (input(PIN_B4)==true && input(PIN_B5)==true&& input(PIN_B6)==false && input(PIN_B7)==false)
{
puerto_a=0b00000100;
}
if (input(PIN_B4)==false && input(PIN_B5)==false&& input(PIN_B6)==true && input(PIN_B7)==false)
{
puerto_a=0b00001000;
}
if (input(PIN_B4)==true && input(PIN_B5)==false&& input(PIN_B6)==true && input(PIN_B7)==false)
{
puerto_a=0b00010000;
}
if (input(PIN_B4)==false && input(PIN_B5)==true&& input(PIN_B6)==true && input(PIN_B7)==false)
{
puerto_a=0b00100000;
}
if (input(PIN_B4)==true && input(PIN_B5)==true&& input(PIN_B6)==true && input(PIN_B7)==false)
{
puerto_a=0b01000000;
}
if (input(PIN_B4)==false && input(PIN_B5)==false && input(PIN_B6)==false && input(PIN_B7)==true)
{
puerto_a=0b10000000;
}
}
}
}

```

7.3 Programa principal

Se comenzó la creación del programa, partiendo de las instrucciones que el prototipo tendría que realizar:

1. Cruzar el par. Debe ser capaz de enviar la instrucción de cruzar el par para verificar su correcto funcionamiento.
2. Enviar tono. Una vez que se haya cruzado el par y si este se encuentra bien, se enviara el tono.
3. Se podrá programar el tiempo de cruce y envío de tono, de 5 hasta 60 segundos.

Como todo programa, comienza con las directivas de cabecera, a continuación, se describen las utilizadas en el proyecto:

```
#include <16F88.h>
```

Esta directiva es usada para indicarle al compilador con que microcontrolador se va a trabajar

```
#use delay(clock=8Mhz,crystal)
```

Para trabajar con un cristal de cuarzo de 8 Megahertz

```
#fuses HS,NOWDT,NOPUT,NOMCLR,NOBROWNOUT,NOLVP
```

Fusibles usados:

HS: Cuando se utiliza un oscilador mayor a 4Mhz

NOWDT: No se utiliza el Watch Dog Timer

NOPUT: No se utiliza el Power Up Timer

NOMCLR: El pin RA5 del PIC puede ser usado como i/o

NOBROWNOUT: Trabaja hasta la tensión mínima de 3v

NOLVP: Sin programación en baja tensión

```
#use standard_io (a)
```

```
#use standard_io (b)
```

Declaración de los puertos como entradas o salidas

```
#define LCD_E    PIN_A2
#define LCD_CK   PIN_A1
#define LCD_DAT  PIN_A0
```

Declaración de los pines A2, A1 y A0 para ser usados por la pantalla LCD 16x2

```
#define clock    PIN_A3
#define data     PIN_A4
```

Declaración de los pines A3 y A4 para los pines de reloj y dato respectivamente del registro de desplazamiento encargado del cruce

```
#define clockdos  PIN_B0
#define datados   PIN_B1
```

Declaracion de los pines B0 Y B1 para los pines de reloj y dato del encapsulado 74LS164 encargado de enviar el tono

```
#include <lcd3.c>
```

Uso de la librería para el control de la pantalla LCD 16x2

```
#byte puerto_B = 0x06
```

Se define el registro PORT B como byte y se sitúa en la posición correspondiente de la memoria RAM.

Posteriormente, se hace un arreglo o una matriz de 8 bits para compararlo con un vector y ser usado por el registro de desplazamiento.

```
byte valor;
byte const bitMask[9] = {128,64,32,16,8,4,2,1,0};
byte const salida[9] = {0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
```

```
byte valortono;
byte const bitMasktono[9] = {128,64,32,16,8,4,2,1,0};
byte const salidatono[9] = {0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
```

Se observan los arreglos tanto para el registro de cruce como para el del envío de tono.

Declaración de variables de tipo entero utilizadas en el programa

```
int a=0,b=0, c=0, ti,t,r=0,f,g;
```

Función de configuración de los registros de desplazamiento 74LS164

Como los registros de desplazamiento son circuitos donde la información de entrada se va desplazando en las salidas cuando se les aplica una señal de sincronismo; si se deja de enviar dicha señal, en la salida se mantiene la información anteriormente guardada, con estas funciones, que posteriormente serán llamadas en otra etapa del programa, se podrán activar los relevadores que se encargan de enviar el tono o la orden de cruce.

```
void comparacion(void)
{
    for(b=0; b<8; b++) // Recorrido de los 8 bits
    {
        output_low(clock); // Coloca el pin de pulso en estado bajo para que el dato se pueda almacenar
        output_bit(data, valor & bitMask[b]); // Salida bit a bit, donde "data" es el PIN A4 y
                                                // "valor" se compara bit por bit con el arreglo "bitMask"
        output_high(clock); // Se coloca en estado alto el pin del pulso para que el dato quede almacenado
        output_low(clock); // Se vuelve a colocar en estado bajo para que unicamente quede activo el bit
                            //seleccionado
    }
}

void tono(void)
{
    for(c=0; c<8; c++)
    {
        output_low(clockdos);
        output_bit(datos, valortono & bitMasktono[c]);
        output_high(clockdos);
        output_low(clockdos);
    }
}
```

Función de envío de tono

En esta función se configura la opción de enviar los tonos y selección de tiempo del envío.

Como primera parte, se guardan en la variable *r* los pines RB4 a RB7.

Al presionar las teclas * y # se puede disminuir y aumentar respectivamente el tiempo de envío de tono, cada pulso a la tecla correspondiente significa aumentar o disminuir el tiempo un segundo. El tiempo es guardado en la memoria eeprom del microcontrolador.

```
void tonos(void) {
    while(true){
        r=portb; // Las entradas de los pines RB4 a RB7 se guardan en la variable r
        r = r/16;

        if(r==12) // Si se presiona la tecla #, aumentan los segundos que durara el envio de tono
        {
            f=read_eeprom(1); // Al presionar la tecla, el conteo parte del tiempo guardado anteriormente

            f++; // Aumente de uno en uno los segundos
            if (f>59) { // Al llegar a sesenta segundos, el conteo vuelve a cero
                f=0;}
            delay_ms(300); //retardo de 300 milisegundos para evitar rebotes

            printf(lcd_putc,"\fEnviar Tono \npor %d Segundos",f); // En la pantalla LCD, se aprecia el mensaje
                                                                    // con los segundos seleccionados
            delay_ms(300); // retardo de 300 milisegundos para evitar parpadeos en la pantalla LCD
            write_eeprom(1,f); // La nueva cantidad de segundos se almacena en la memoria eeprom
        }
        if(r==11) // Mismo procedimiento, pero ahora de manera descendente al presionar la tecla *
        {
            f=read_eeprom(1);
            f--;
            if (f<0) {
                f=0;}
            delay_ms(300);

            printf(lcd_putc,"\fEnviar Tono \npor %d Segundos",f);
            delay_ms(300);
            write_eeprom(1,f);
        }
    }
}
```

Una vez seleccionado el tiempo, se puede enviar el tono por el par que se elija

```
if(r>0&&r<9) // Si se presiona una tecla del 1 al 8
{
    printf(lcd_putc, "\fPar %Lx", r); // En la pantalla LCD se muestra un mensaje diciendo el par elegido
    valortono = salidatono[r]; // Llama a la función donde se configuro al registro de desplazamiento
    // encargado enviar el tono

    tono();
    for (g=1; g<=read_eeprom(1); g++) { // Enviar el tono únicamente dura el tiempo guardado
// anteriormente

        lcd_gotoxy(1,2);
        printf(lcd_putc, "Tono por %2dS", g); // En la pantalla LCD se aprecie un mensaje mostrando el
//tiempo que el tono está siendo enviado
        delay_ms(1000);
    }

    printf(lcd_putc, "\fAbriendo..."); //Al terminar de enviar el tono, muestra un mensaje
    delay_ms(1000); // de un segundo
    valortono = salidatono[r=0];
    tono();

    printf(lcd_putc, "\fElija otro par \n# = +, * = - tiempo");

}
if(input(PIN_B4)==1&&input(PIN_B5)==0&&input(PIN_B6)==0&&input(PIN_B7)==1){
// Al presionar la tecla 9, se puede cambiar a la opcion de enviar cruce

    lcd_putc("\fELIJA PAR \nDE CRUCE");
    delay_ms(100);
    break;
    }
    }
    }
```

Con la ayuda de las interrupciones por cambio de estado de los pines RB4 a RB7, se pueden seleccionar los pares de cruce, al presionar la tecla nueve automáticamente las interrupciones se activan, por lo que se puede utilizar el mismo teclado para las dos opciones posibles.

El funcionamiento de la instrucción de envío de cruce, es el mismo que el envío de tono, la opción de agregar el tiempo funciona igual.

```

#int_RB
void RB_isr(void) {
    a=puerto_B;
    a = a/16;

    if (a==12) {
        t=read_eeprom(0);
        t++;
        if (t>59) {
            t=0;}
        delay_ms(300);
        printf(lcd_putc,"\fCruzar por \n %d Segundos",t);
        delay_ms(300);
        write_eeprom(0,t);

    } if (a==11) {
        t=read_eeprom(0);
        t--;
        if (t<0) {
            t=0;}
        delay_ms(300);

        printf(lcd_putc,"\fCruzar por \n %d Segundos",t);
        delay_ms(300);
        write_eeprom(0,t);
    }
    if (a>0&&a<9) {
        printf(lcd_putc,"\fPar %Lx",a);
        valor = salida[a];
        comparacion();

        for (ti=1;ti<=read_eeprom(0);ti++) {
            lcd_gotoxy(1,2);
            printf(lcd_putc,"Cruzando Por %2dS",ti);
            delay_ms(1000); }

            printf(lcd_putc,"\fAbriendo Par");
            delay_ms(1000);
            valor = salida[a=0];
            comparacion();
        printf(lcd_putc,"\fElija otro par \n# = +, *=- tiempo"); }
        if (a==10) {
            disable_interrupts(INT_RB);
            printf(lcd_putc,"\fENVIAR TONO \nELIJA PAR");
            delay_ms(100);
            tonos(); }
    }
}

```

Función principal

En la función principal se activan las interrupciones, el mensaje de bienvenida que es visualizado en la pantalla LCD 16x2

```
void main(){

    enable_interrupts(INT_RB);//habilita interrupción particular
    enable_interrupts(GLOBAL);//habilita interrupción global
    lcd_init();

    lcd_putc("\fCONMUTADOR \n  MANUAL V1.0");
    delay_ms(1000);
    lcd_putc("\fElija Accion");
    delay_ms(1000);
    lcd_putc("\fTecla 9 = Cruce\nTecla 0 = Tono");
    delay_ms(1000);
    lcd_putc("\fPres. # para +\nPres. * para -");
    delay_ms(2000);

    while(true)

    {

    if(input(PIN_B4)==1&&input(PIN_B5)==0&&input(PIN_B6)==0&&input(PIN_B7)==1)

    {
        enable_interrupts(INT_RB);
        lcd_putc("\fELIJA PAR \nDE CRUCE");
        delay_ms(100);
    }
    }
}
```

7.4 Diseño en software Proteus

Con la simulación del decodificador de tonos desarrollado al principio, se pudo simular el programa creado de manera más didáctica en Proteus.

Si bien, con *push buttons* o únicamente con el teclado que por defecto contiene el software simulador, se pueden representar las tareas del prototipo, considere necesario hacerlo, puesto que complementaria el proyecto al hacerlo más cercano a la realidad.

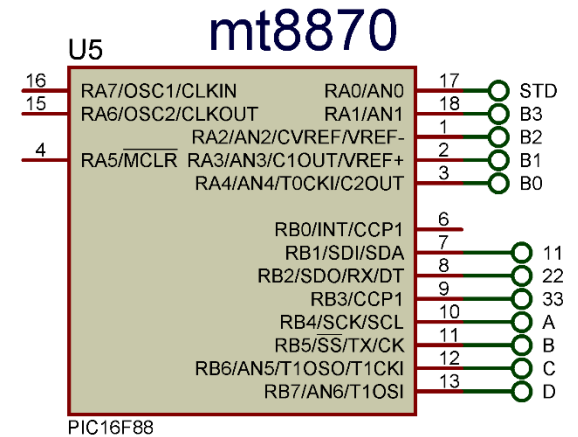


FIG. 10 PIC16F88 que simula el decodificador de tonos MT8870 en Proteus. Creación propia

En el prototipo real, utilice dos módulos de relevadores de 8 y 16 canales que se encargan de hacer el cruce y de los pares y enviar el tono, en Proteus, utilice relevadores animados para poder observar el momento en que conmutan.

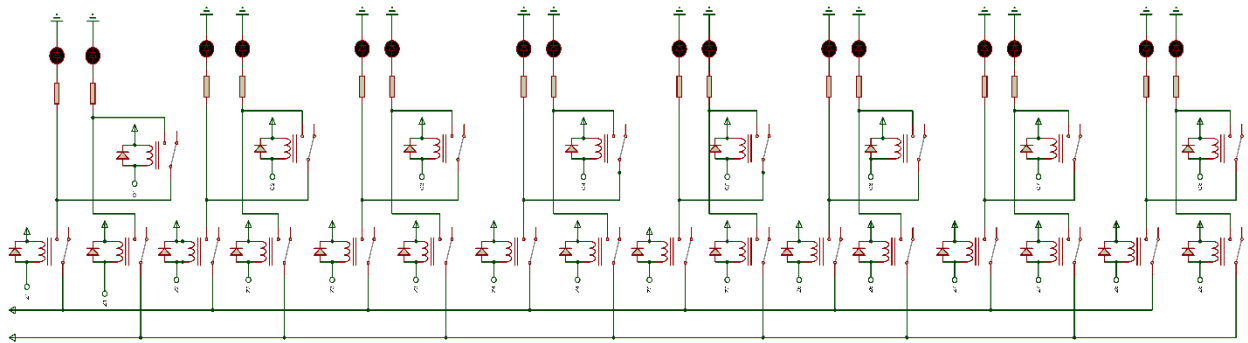


FIG.11 Relevadores de tono y cruce. Creación propia

En ocasiones, la corriente de salida de los microcontroladores no es suficiente para activar las cargas, utilice transistores 2N2222A por motivos de simulación, los módulos reales ya contienen los elementos necesarios para activar los relevadores sin ningún problema.

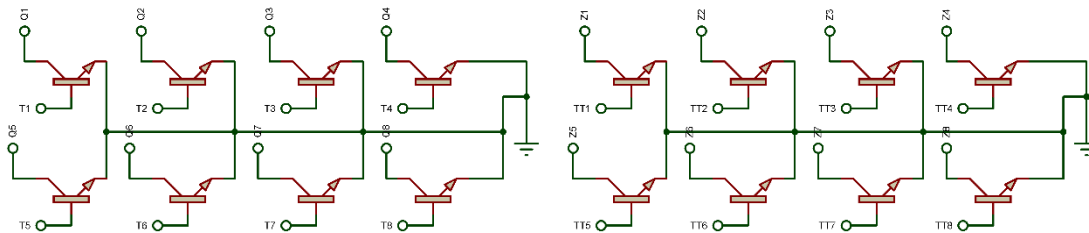


FIG.12 Transistores amplificadores. Creación propia

Pantalla LCD 16x2 con el registro de desplazamiento 74LS164 que permite ser usado únicamente con 3 pines.

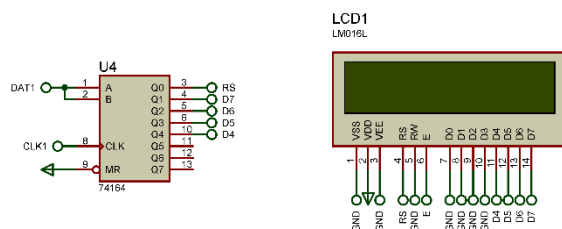


FIG. 13 LCD 16x2 con el registro de desplazamiento 74LS164 Creación propia

PIC principal, teclado, leds indicadores y registros de desplazamiento de los relevadores de conmutación

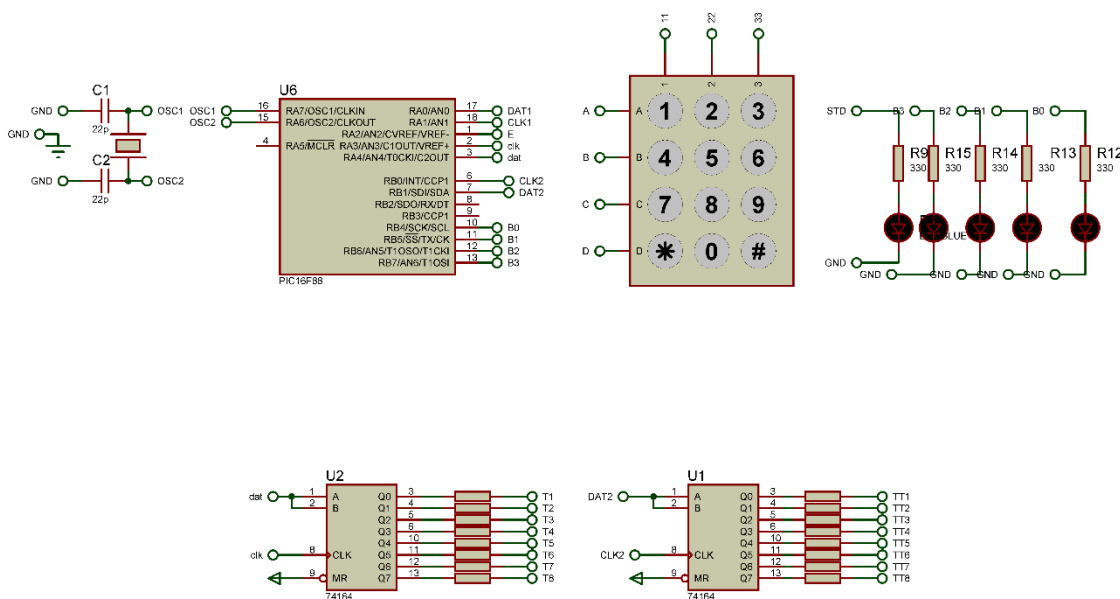


FIG.14 Elementos de control Elaboración propia

7.5 Pruebas en protoboard

Una vez que corroboré que en simulación todo funcionaba de manera correcta, procedí a armar el circuito en protoboard.

El archivo .hex generado al compilar el programa hecho en CCS, lo grabe en el pic con el quemador MicroProg y el software Pickit 2 de Microchip.

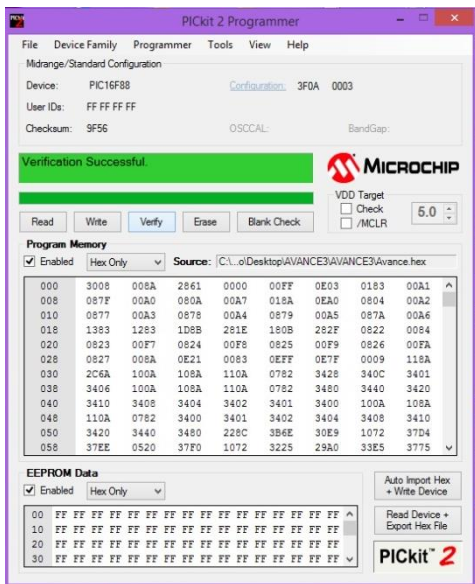


FIG.15 Interfaz del quemador Pickit 2
Captura propia

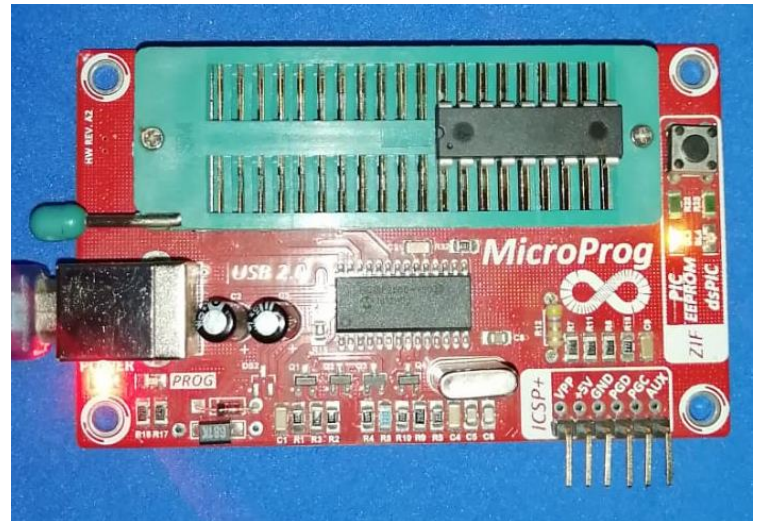


FIG. 16 Quemador MicroProg con el PIC16F88 Fotografía propia

Prueba en protoboard de la pantalla LCD 16x2 con el registro de Desplazamiento 74LS164

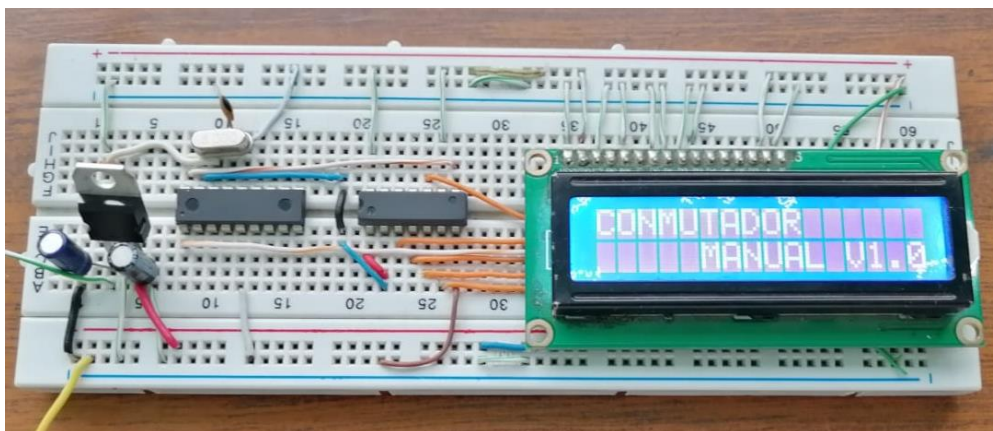


FIG. 17 LCD 16x2 con el registro de desplazamiento mostrando parte del mensaje de bienvenida
Fotografía propia

Montaje de completo de los elementos.

Conecté ambos módulos de relevadores e hice las pruebas de envío de tono y envío de cruce funcionando tal cual lo hacía en las simulaciones del software Proteus, por lo que procedí al siguiente paso que fue diseñar el PCB.

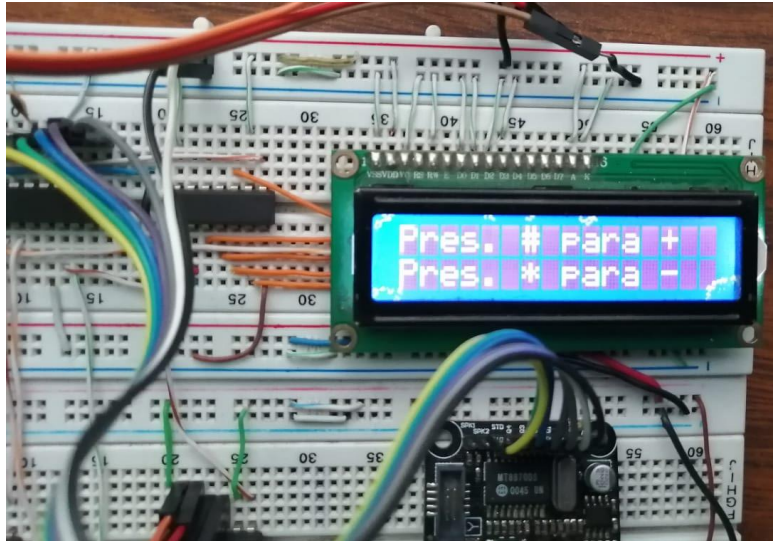


FIG. 18 Se aprecia la parte del menú que pide aumentar o disminuir el tiempo de cruce o envío de tono
Fotografía propia.

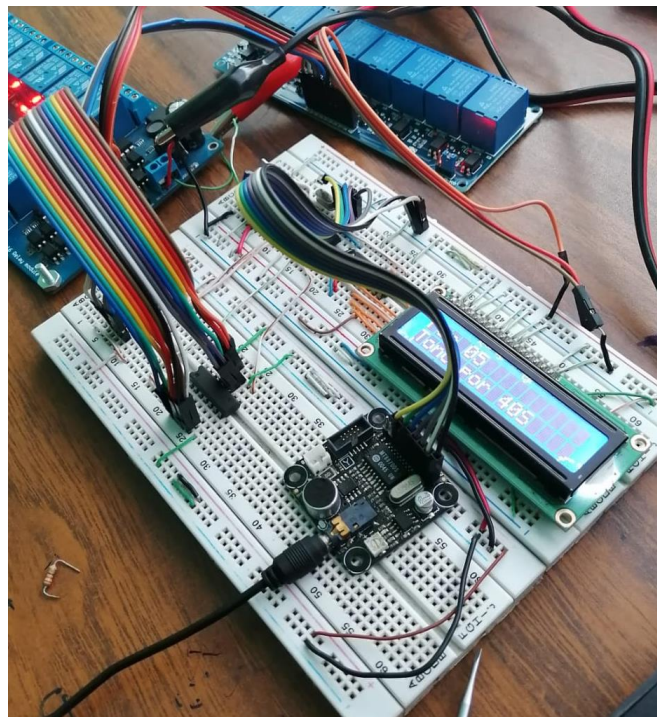


FIG. 19 Se observa en la pantalla que el tono está siendo enviado.
Fotografía propia

7.6 Diseño de PCB

Para comenzar a diseñar la PCB, en el software Proteus diseñe el circuito con los elementos que estarán soldados directamente a la placa. La pantalla LCD 16x2 y los módulos de relevadores serian conectados mediante cables a los conectores o *headers* de la placa. Además, agregue un regulador de voltaje para asegurar la tensión correcta. Ya que el espacio donde será montado el prototipo es pequeño, decidí realizar dos placas, una con los elementos de control y la otra con los registros y salidas hacia los relevadores

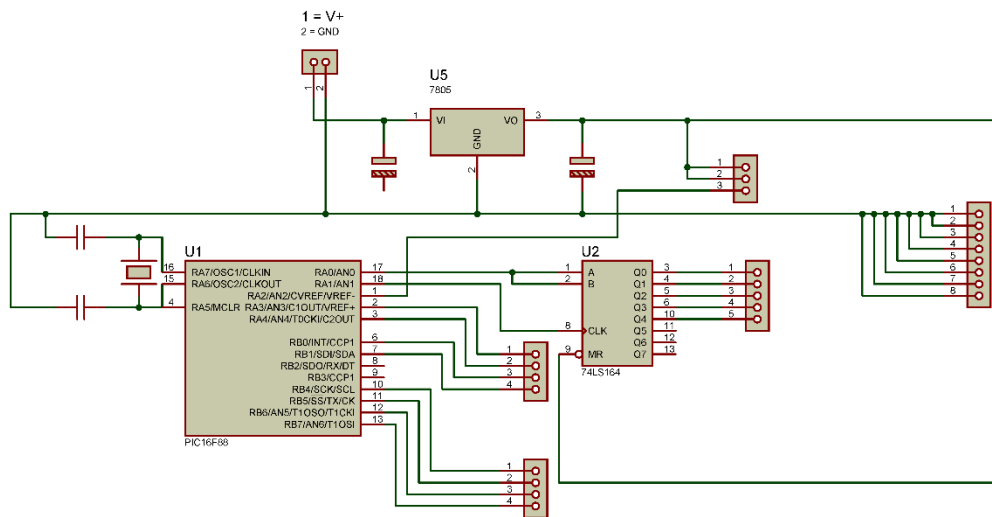


FIG. 20 Circuito con el regulador de voltaje, microcontrolador, 74LS164 y headers. Elaboración propia

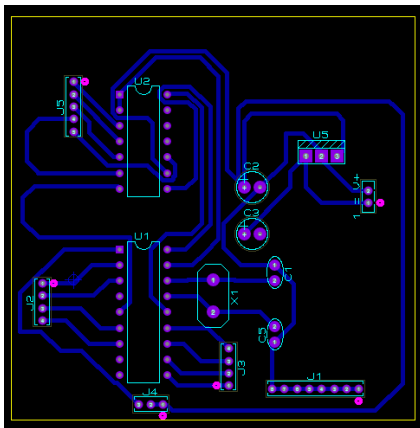


FIG. 21 Diseño de PCB Creación propia

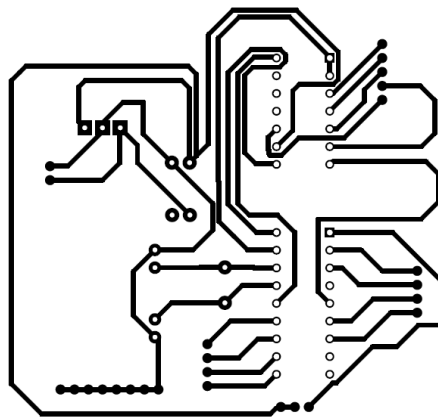


Fig. 22 Circuito para imprimir. Creación propia

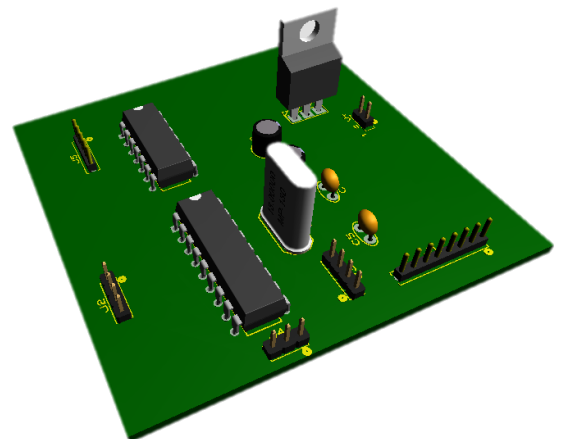


Fig. 23 Modelo en 3D. Creación propia

Diseño del PCB de los registros de desplazamiento y headers de las salidas

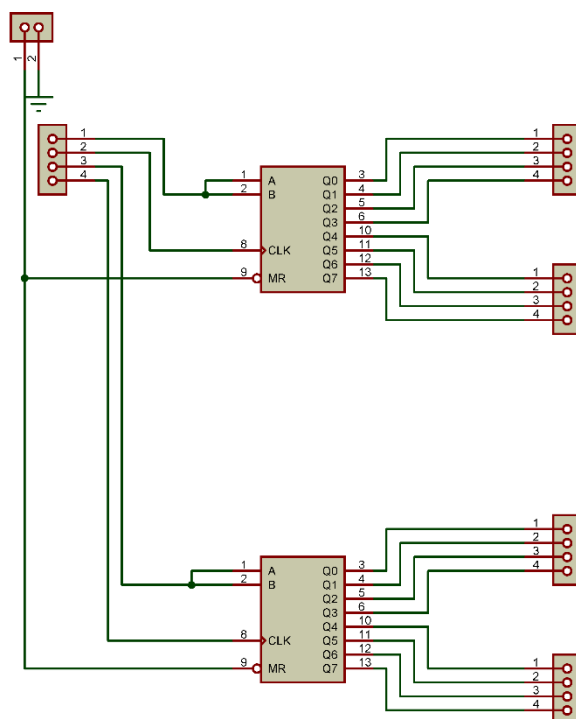


FIG. 24 Circuito con los registros de desplazamiento. Elaboración propia

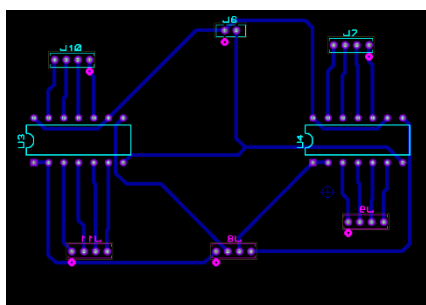


FIG. 25 Segundo diseño de PCB Creación propia.

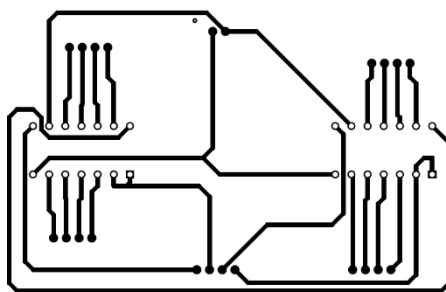


FIG. 26 Plot para Imprimir. Creación propia

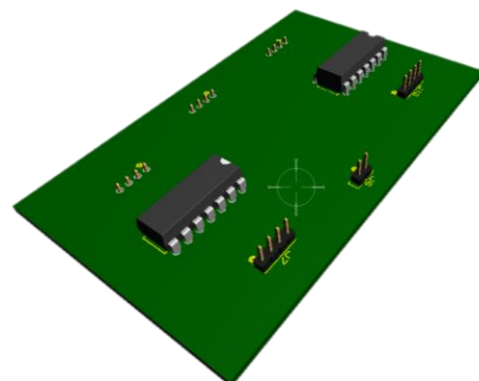


FIG. 27 Diseño en 3D. Creación propia

7.7 Ensamble Final

El prototipo está diseñado para ser ergonómico, pequeño y ligero, es por ello que tenía que ser montado en una carcasa que facilitara su traslado de un lado a otro. En el mercado existen diversas opciones, elegí esta porque cumple las características que buscaba.



FIG. 28 Vista frontal de la carcasa del prototipo
Fotografía propia



FIG. 29 Vista de la carcasa abierta.
Fotografía propia

Proceso de ensamble



FIG. 30 Ensamble de la pantalla LCD y los módulos de relevadores. *Fotografía propia*

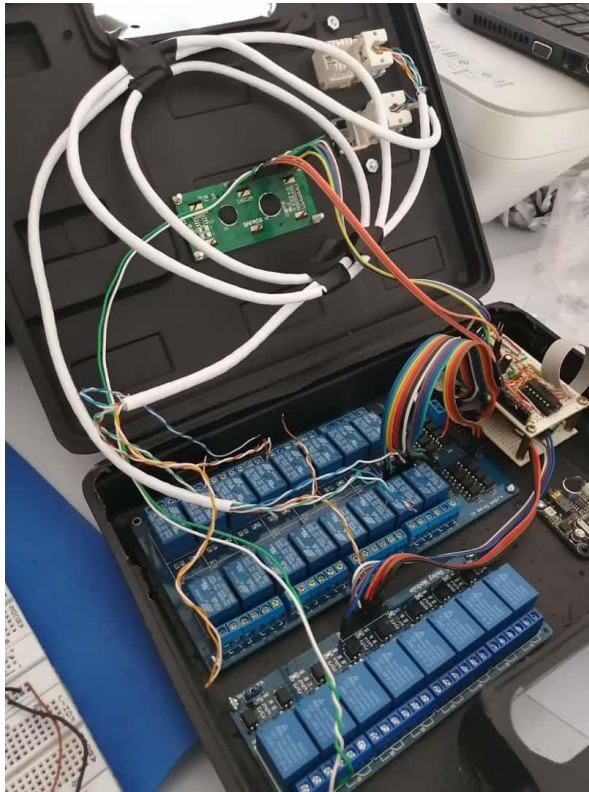


FIG. 31 Ensamble de todos los elementos Fotografía propia



FIG. 32 Vista frontal del prototipo. Fotografía propia

CAPITULO VIII

RESULTADOS

El resultado de estos meses trabajando en el proyecto que se me asignó, es un prototipo totalmente funcional, que cuenta con un programa eficiente, el hardware es suficiente para realizar las tareas para lo que fue diseñado y construido.

A continuación, se muestran las evidencias de cada una de las partes que conforman el Prototipo de Pruebas en Conmutadores:

8.1 Programa:

Como ya he mencionado en reiteradas ocasiones, el programa fue hecho en lenguaje C con el compilador CCS

```
#include <16F88.h>
#use delay (clock=8Mhz,crystal)
#fuses HS,NOWDT,NOPUT,NOMCLR,NOBROWNOUT,NOLVP,NOWRT
#use standard_io (a)
#use standard_io (b)
#define LCD_E    PIN_A2
#define LCD_CK    PIN_A1
#define LCD_DAT    PIN_A0
#define clock    PIN_A3
#define data    PIN_A4
#define clockdos    PIN_B0
#define datados    PIN_B1
#include <lcd3.c>
#byte puerto_B = 0x06
```

```
byte valor;
byte const bitMask[9] = {128,64,32,16,8,4,2,1,0};
byte const salida[9] = {0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
```

```
byte valortono;
byte const bitMasktono[9] = {128,64,32,16,8,4,2,1,0};
byte const salidatono[9] = {0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
```

```

void comparacion(void);
void tono(void);
void tonos (void);
int a=0,b=0, c=0, ti,t,r=0,f,g;

```

```

void comparacion(void)
{
    for(b=0; b<8; b++)
    {
        output_low(clock);
        output_bit(data, valor & bitMask[b]);
        output_high(clock);
        output_low(clock);
    }
}

```

```

void tono(void)
{
    for(c=0; c<8; c++)
    {
        output_low(clockdos);
        output_bit(datos, valortono & bitMasktono[c]);
        output_high(clockdos);
        output_low(clockdos);
    }
}

```

```

void tonos(void) {
while(true) {
    r=portb;
    r = r/16;

```

```

if(r==12)
{
    f=read_eeprom(1);
    f++;
    if (f>59) {
        f=0; }
    delay_ms(300);

    printf(lcd_putc,"\fEnviar Tono \npor %d Segundos",f);
    delay_ms(300);
    write_eeprom(1,f);
}
if (r==11)
{
    f=read_eeprom(1);
    f--;
    if (f<0) {
        f=0;}
    delay_ms(300);

    printf(lcd_putc,"\fEnviar Tono \npor %d Segundos",f);
    delay_ms(300);
    write_eeprom(1,f);
}

```

```

if (r>0&&r<9)
{

printf(lcd_putc,"\fPar %Lx",r);
valortono = salidatono[r];
tono();
for (g=1;g<=read_eeeprom(1);g++) { //ti=g
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Tono por %2dS",g);
    delay_ms(1000);
}

printf(lcd_putc,"\fAbriendo...");
delay_ms(1000);
valortono = salidatono[r=0];
tono();
printf(lcd_putc,"\fElija otro par \n#= +, *=- tiempo");

}

if(input(PIN_B4)==1&&input(PIN_B5)==0&&input(PIN_B6)==0&&input(PIN_B7)==1)
{
    lcd_putc("\fELIJA PAR \nDE CRUCE");
    delay_ms(100);
    break;
}
}
}
}

```

```

#int_RB

void RB_isr(void)
{
    a=puerto_B;
    a = a/16;
    if(a==12)
    {
        t=read_eeprom(0);
t++;
        if (t>59) {
            t=0;}
        delay_ms(300);
printf(lcd_putc,"\nCruzar por \n %d Segundos",t);
        delay_ms(300);
        write_eeprom(0,t);
    }
    if(a==11)
    {
        t=read_eeprom(0);
        t--;
        if (t<0) {
            t=0;}
        delay_ms(300);
printf(lcd_putc,"\nCruzar por \n %d Segundos",t);
        delay_ms(300);
        write_eeprom(0,t);
    }
}

```

```

if(a>0&&a<9)
    }

    printf(lcd_putc,"\fPar %Lx",a);
    valor = salida[a];
    comparacion();
for (ti=1;ti<=read_eeprom(0);ti++) {
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Cruzando Por %2dS",ti);
    delay_ms(1000);
}
printf(lcd_putc,"\fAbriendo Par");
delay_ms(1000);
valor = salida[a=0];
comparacion();
printf(lcd_putc,"\fElija otro par \n# = +, * = - tiempo");
}
if(a==10) {
disable_interrups(INT_RB);
printf(lcd_putc,"\fENVIAR TONO \nELIJA PAR");
delay_ms(100);
tonos();
}
}

```

```

void main(){

    enable_interrupts(INT_RB);//habilita interrupción particular
    enable_interrupts(GLOBAL);//habilita interrupción global
    lcd_init();
    lcd_putc("\fCONMUTADOR \n  MANUAL V1.0");
    delay_ms(1000);
    lcd_putc("\fElija Accion");
    delay_ms(1000);
    lcd_putc("\fTecla 9 = Cruce\nTecla 0 = Tono");
    delay_ms(1000);
    lcd_putc("\fPres. # para +\nPres. * para -");
    delay_ms(2000);
    while(true)
    {

        if(input(PIN_B4)==1&&input(PIN_B5)==0&&input(PIN_B6)==0&&input(PIN_B7)==1)
        {
            enable_interrupts(INT_RB);
            lcd_putc("\fELIJA PAR \nDE CRUCE");
            delay_ms(100);
        }
    }

}

```


8.3 Prototipo Concluido

Se ensambló en una carcasa que permite un fácil manejo, contiene una interfaz de usuario simple y clara visualizada en la pantalla LCD, como se pudo observar en el programa, se constituye de un mensaje de bienvenida y las opciones de Enviar cruce o Enviar par y seleccionar el tiempo respectivo, todo esto mediante el teclado del teléfono celular.

Especificaciones Técnicas

Tensión de alimentación	Min. 6v CD – Max. 12v CD
Dimensiones	30cm*30cm*9cm
Peso	1kg
Número de pares	8
Temperatura	Min 0° - Max 60°

TAB. 2 Ficha técnica. Creació



FIG. 34 Se observa que se ha seleccionado la opción de enviar tono. Fotografía propia.



FIG. 35 El tono está siendo enviado. Fotografía propia

CAPITULO IX

CONCLUSIONES

El prototipo en este momento se encuentra funcionando y cumple con las tareas para lo que fue diseñado.

Como recomendación, se podría diseñar en una PCB moderna, trasladarlo a un microprocesador SMD para miniaturizarlo lo más posible.

El desarrollo de este prototipo me deja demasiadas enseñanzas tanto personales como en el ámbito profesional.

Debido a la pandemia que está azotando al mundo, el prototipo lo realice en casa y si bien cuento con todas las herramientas necesarias para haberlo concluido satisfactoriamente, hubo momentos en los que se me complico hacerlo, sin embargo, siempre se debe sacar lo mejor de sí y con ayuda de las herramientas digitales, las fuentes de bibliográficas y sobre todo las asesorías del Ingeniero Emmanuel, pude resolver todos estos problemas y aprender de ellos.

Personalmente, este proyecto me hizo más responsable, el hecho que yo controlara mis horas de trabajo, lejos de tomarlo como una oportunidad de dejar todo al final, me hizo ver que si me dedico a realizar las cosas en tiempo y forma las recompensas serán satisfactorias, además de que no es necesario tener a alguien detrás ordenando lo que se debe hacer para cumplir.

Otra enseñanza personal que me dejo este proyecto es que haber estudiado Ingeniería Electrónica fue la mejor decisión que pude tomar, pues basta con imaginar para poder crear nuevas cosas y realizar innovaciones.

En el ámbito profesional, me quedo con la experiencia de trabajar en un prototipo que será de utilidad para alguien, ahora tengo más experiencia en programar microcontroladores y podre desarrollar proyectos mas adelante.

CAPITULO X

COMPETENCIAS DESARROLLADAS Y/O APLICADAS

Al término de la residencia profesional me siento capaz de diseñar y elaborar más y mejores proyectos, desarrolle las competencias de programación, análisis, diseño de circuitos, diseño de PCBs, trabajar bajo la modalidad de *home office* y cumplir con resultados positivos.

Además de aplicar las competencias que a lo largo de mi educación superior desarrolle como el uso eficiente de las tecnologías de información, la habilidad de resolver las dudas que surgieron en fuentes bibliográficas, trabajar bajo presión y análisis matemático.

Tener iniciativa y responsabilidad en un trabajo donde me dieron la confianza de trabajar en casa fue muy importante, así como también la comunicación efectiva.

La competencia que considero más importante y que me ayudo a cumplir con este proyecto fue la capacidad de aprendizaje, porque los conocimientos de programación que tenía antes de esto eran muy limitados, sin embargo, al principio del proyecto consulté diversos materiales tanto digitales como bibliográficos para recordar todos los conceptos que principalmente debí haber estudiado en materias como microcontroladores y electrónica digital. En el transcurso no solo aprendí lo suficiente para terminar este proyecto, también despertó en mi el interés por desarrollar toda clase de proyectos que tengan que ver con microcontroladores, tanto así que me gustaría emprender en el ámbito del Internet de las Cosas.

CAPITULO XI

FUENTES DE INFORMACIÓN

Fuentes bibliográficas

Axelson, J. (1999). USB Complete: Everything You Need to Develop USB Peripherals. Madison: Lakeview Research LLC.

Barnett, R. (2004). Embedded C Programming & Microchip PIC. Nueva York: Delmar Learning.

Benson, D. (2008). C What Happens. Hayden: Square Electronics.

Breijo, E. G. (2008). Compilador C CCS y Simulador PROTEUS para Microcontroladores PIC. Barcelona: Alfaomega.

Garder, N. (2002). PICmicro MCU C. Bluebird.

Ramiro, F., & López, L. J. (2004). Microcontrolador PIC16F84. Ciudad de Mexico: Alfaomega.

Saravia, A. R. B. (2008). Microcontroladores PIC en Lenguaje C con CCS. Ciudad de Mexico: Alfaomega.

Tokheim, R. L. (2000). Fundamentos de los Microprocesadores (Segunda Ed). Minnesota: McGraw Hill.

Unibertsitatea, N. (2005). Aprende Informatica. Barcelona.

William, B. (2001). Ingeniería de Control. Ciudad de Mexico: Alfaomega.

Fuentes digitales

Información general de los elementos utilizados. Consultados el 3 de octubre de 2020.

<https://www.microchip.com/wwwproducts/en/PIC16F88>

<https://www.ti.com/product/SN74LS164/>

<https://www.alldatasheet.es/datasheet-pdf/pdf/1132639/SONGLERELAY/SRD-05VDC-SL-C.html>