



# Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

**Microcontroladores**

Materia

**Proyecto Final**

**Camarillo Bautista Alfredo, González Escalona Miguel Ángel, Lázaro Bonilla**

**Ramiro, López Arce Roberto**

Integrantes del equipo

**Ingeniería en mecatrónica**

Carrera

**Ricardo Álvarez González**

Docente

**18 de mayo de 2021, Primavera 2021**

Fecha de entrega

## Contenido

Objetivo.....	3
Marco teórico.....	3
Desarrollo práctico .....	6
Conclusiones .....	10
Bibliografía .....	11

## Objetivo

El objetivo de este proyecto tiene como prioridad el realizar la programación del microcontrolador 18f4550, mostrando los elementos electrónicos necesarios para construir un prototipo de simulación de un instrumento acústico de cuerdas que en este caso es un piano mediante la programación de este microcontrolador.

Dicho esto, se pretende diseñar este piano electrónico con fines de demostración didáctica como también académica sobre la importancia del desarrollo de un sistema programable enfocado en el mundo real y como es que podemos implementar lo aprendido en la materia y enfocarlo en algo real y que puede tener una gran utilización en el ámbito cotidiano.

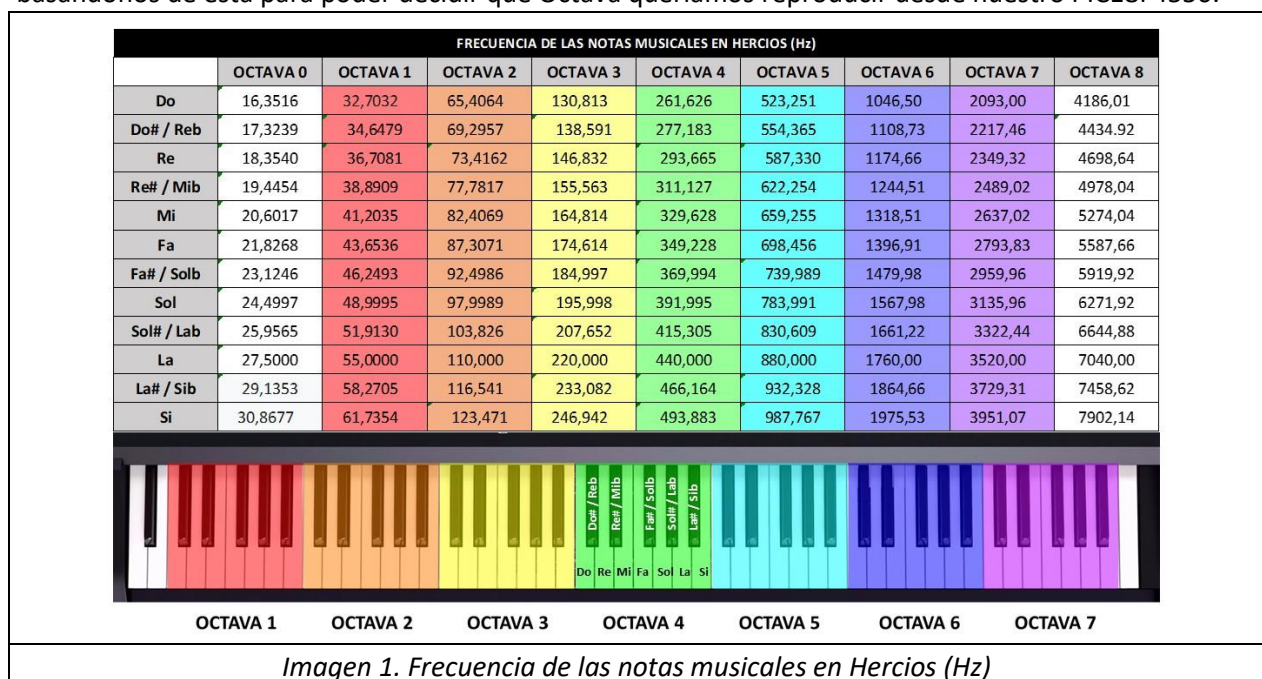
## Marco teórico

El pic 18f4550 es un microcontrolador de 8 bits de la empresa Microchip. Este microcontrolador cuenta con una gran cantidad de memoria RAM, diferentes módulos de comunicación, una gran cantidad de pines de entrada y salida y algunas otras grandes cualidades. Algunas de sus características son:

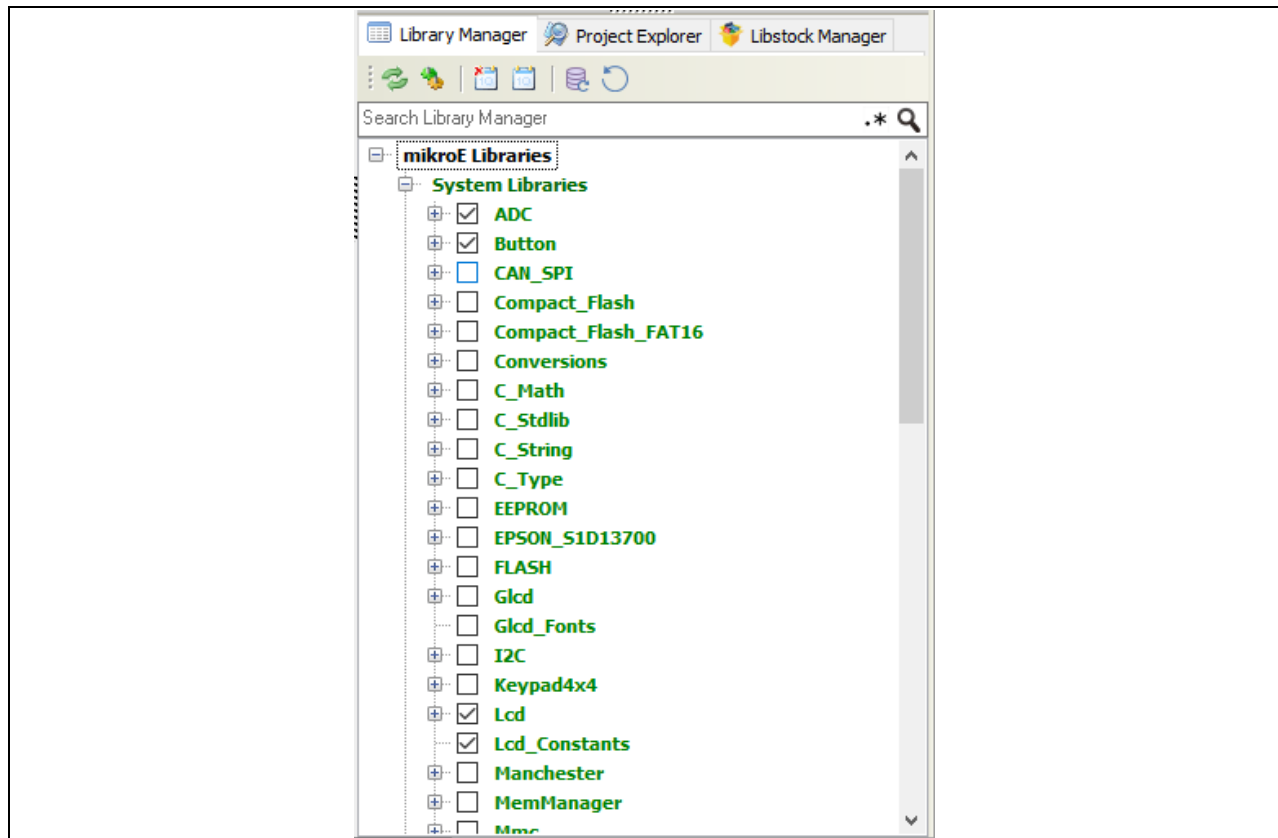
- 40 pines tipo DIP
- Interfaz USB 2.0 de alta velocidad, EEPROM 256 bytes
- Memoria RAM 2048 bytes, EEPROM 256 bytes
- Memoria de programa (memoria flash) 32 kb
- Voltaje de operación 2 a 5.5 V
- Frecuencia máxima 48 MHz
- 35 pines de entrada / salida

Las instrucciones utilizadas en la práctica se enlistarán y explicarán a continuación.

Para poder generar los sonidos, primero debíamos conocer la frecuencia necesaria para poder reproducir cada nota, por lo que tomamos como referencia la información de la página *Ciudad Pentagrama*, basándonos de esta para poder decidir que Octava queríamos reproducir desde nuestro PIC18F4550.



Posteriormente, para el desarrollo del código utilizamos el software “*mikroC PRO for PIC*”. Este software es de fácil uso puesto que su interfaz es amigable con el usuario además de que el uso de librerías especializadas también colabora en un desarrollo más productivo respecto al tiempo.



*Imagen 2. La interfaz del IDE nos permite simplemente seleccionar la librería con la que deseamos trabajar*

### Sound Library

Para poder reproducir sonidos, *mikro C* tiene una librería especial la cuál únicamente nos pide la frecuencia del sonido que queremos reproducir y el tiempo en milisegundos que deseamos que sea reproducido. La generación del sonido requiere hardware adicional, el cuál se trata de un piezo – speaker.

Rutinas de librería:

- *Sound\_init*
- *Sound\_play*

#### **Sound\_Init:**

##### *Sintaxis*

```
void Sound_Init(char *snd_port, char snd_pin);
```

##### *Return*

Nada

##### *Descripción*

Configura el pin apropiado para generar sonido

*Requerimientos*

Ninguno

*Ejemplo*

```
//Inicializa el pin RC3 para reproducir sonido  
Sound_Init(&PORTC, 3);
```

**Sound\_Play:**

*Sintaxis*

```
void Sound_Init(char *snd_port, char snd_pin);
```

*Return*

Nada

*Descripción*

Configura el pin apropiado para generar sonido

*Requerimientos*

Ninguno

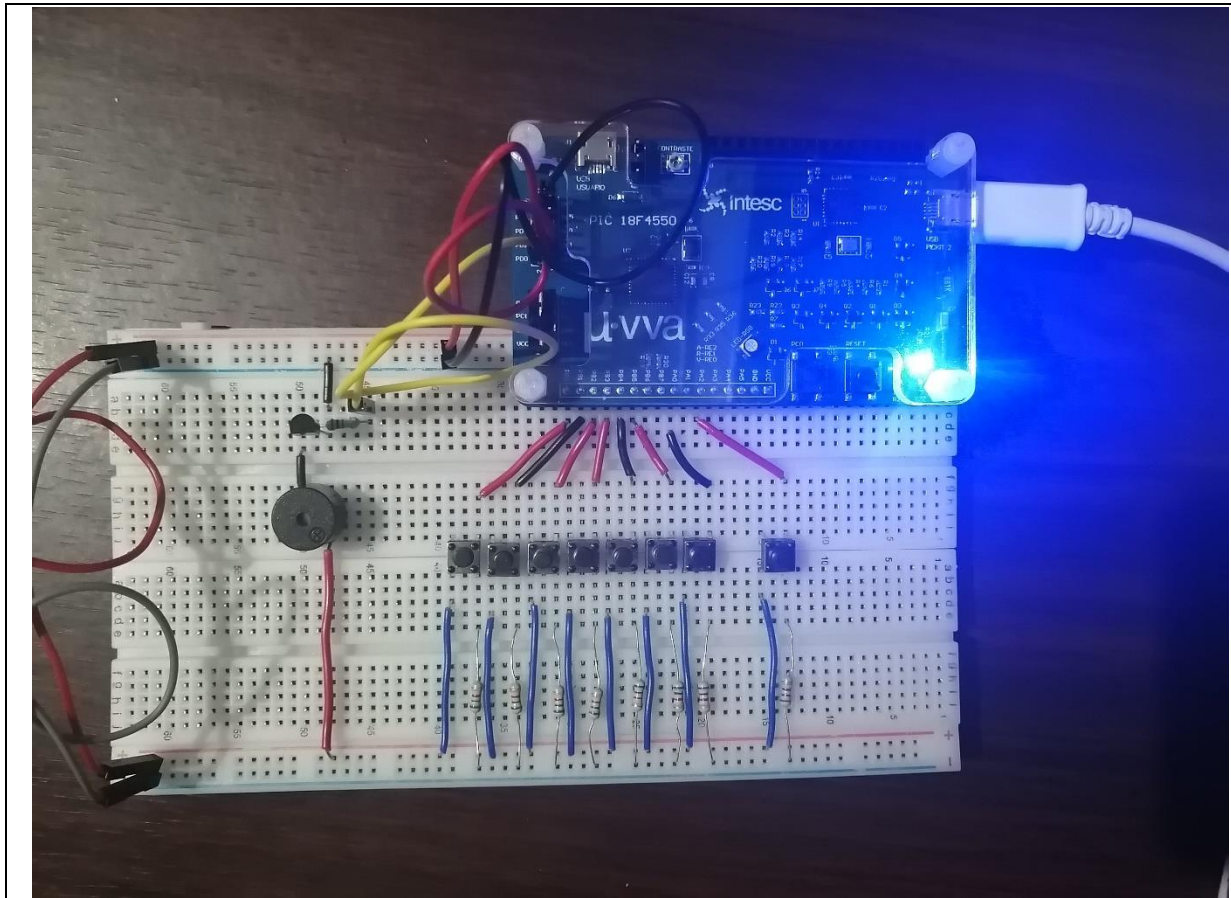
*Ejemplo*

```
//Inicializa el pin RC3 para reproducir sonido  
Sound_Init(&PORTC, 3);
```

## Desarrollo práctico

Para la práctica realizada se utilizaron los siguientes materiales a enlistar.

Cantidad	Concepto
1	Tarjeta de desarrollo Miuva
1	Laptop
1	Buzzer Piezoeléctrico
8	Push Button
8	Resistencia 4.7 KOhms
1	Transistor BC547
1	Metro de alambre
1	Resistencia 1 KOhms
1	Software mikroC PRO for Pic



*Imagen 3. Circuito implementado en la tarjeta de desarrollo miuva*

El código consiste en habilitar los puertos B y A como entradas y utilizar el puerto D como salida, además se debe inicializar los puertos en cero puesto que podrían presentarse problemas al dejar de alimentar al circuito y volver a encenderlo.

Se utiliza una función para cada nota (se decidió usar la frecuencia de las notas de la octava 6) para poder simplificar el llamado de cada una, así solamente se escribe en el programa principal el nombre de la nota y se reproduce al presionarse el push button indicado.

Los pines RB0 – RB5 y RA0 – RA1 se utilizan para reproducir las notas que van desde Do – Si y el pin RA1 se utiliza para reproducir las notas de la canción “My heart Will go on – Celine Dion”. Se escogió la famosa canción de la película “Titanic” debido a que no se contaba con los conocimientos amplios sobre música y es una de las canciones que se encuentra más simplificada (a nuestra manera de ver) y se puede reproducir fácilmente. A continuación se adjunta el código escrito para poder hacer funcionar el teclado musical:

```
void do0(){          //se le asigna el nombre de do0 a lo que este adentro
Sound_Play(1046, 1000); //se coloca una frecuencia de 1khz y 500ms para que este activo
}                    // y asi para los demas
void re(){
Sound_play(1174, 1000);
}
void mi(){
Sound_play(1318, 1000);
}
void fa(){
Sound_play(1397, 1000);
}
void sol(){
Sound_play(1568, 1000);
}
void la(){
Sound_play(1760, 1000);
}
void si(){
Sound_play(1975, 1000);
}
void do0_(){
Sound_play(1975, 1000);
}
void re_(){
Sound_play(1975, 1000);
}
void fa_(){
Sound_play(1975, 1000);
}
void sol_(){
Sound_play(1975, 1000);
}
void la_(){
Sound_play(1975, 1000);
}
void cancion(){
do0(); re(); mi(); mi(); mi(); mi(); delay_ms(500);
re(); do0(); re(); sol(); sol(); sol(); delay_ms(500);
fa(); mi(); do0(); do0(); do0(); do0(); do0(); delay_ms(4000);
do0(); re(); mi(); mi(); mi(); mi(); delay_ms(500);
re(); do0(); re(); sol(); sol(); sol(); delay_ms(500);
mi(); sol(); la(); la(); la(); sol(); sol(); delay_ms(100); re(); re(); re(); re(); delay_ms(100);
do0(); do0(); delay_ms(100); do0(); delay_ms(100); do0(); delay_ms(100); do0(); si(); do0(); do0();
```

```

delay_ms(100); do0(); si(); do0(); do0(); delay_ms(100); re(); mi(); mi(); mi(); re(); re(); re();
do0(); do0(); delay_ms(100); do0(); delay_ms(100); do0(); delay_ms(100); do0(); si(); do0(); do0();
delay_ms(100); do0(); sol(); sol(); sol(); sol(); delay_ms(1800);
do0(); do0(); delay_ms(100); do0(); delay_ms(100); do0(); delay_ms(100); do0(); delay_ms(100); si();
si(); si();
do0(); do0(); delay_ms(100); do0(); delay_ms(100); si(); si(); si(); do0(); do0(); do0();
re(); mi(); mi(); mi(); re(); re();
do0(); do0(); delay_ms(100); do0(); delay_ms(100); si(); si(); do0(); do0(); sol(); sol(); sol(); sol(); sol();
}

```

```

void main() {      //funcion principal
ADCON1 = 0x0F;
PORTA=0;          //el puerto A inicializa apagado
PORTB=0;
PORTD=0;
TRISA=1;          //todo el puerto A como entrada
TRISD=0;
TRISB=1;
Sound_Init(&PORTD, 0); //se inicializan los sonidos en el puerto D y el pin 1 es decir RD1

```

inicio: //se crea un bucle infinito con la funcion goto

```

if(RB0_Bit==1){    //si se preciona boton que esta en RA0 entonces
delay_ms(50);      //espera 50ms
do0();             //despues llama a do0 y todo lo que tenga dentro
}
if(RB1_Bit){        //si se preciona boton que esta en RA1 entonces
delay_ms(50);      //espera 50ms
re();              //despues llama a re y todo lo que tenga dentro
}
if(RB2_Bit){        //si se preciona boton que esta en RA2 entonces
delay_ms(50);      //espera 50ms
mi();              //despues llama a mi y todo lo que tenga dentro
}
if(RB3_Bit){        //si se preciona boton que esta en RA3 entonces
delay_ms(50);      //espera 50ms
fa();              //despues llama a fa y todo lo que tenga dentro
}
if(RB4_Bit){        //si se preciona boton que esta en RA4 entonces
delay_ms(50);      //espera 50ms
sol();             //despues llama a automatico y todo lo que tenga dentro
}
if(RB5_Bit){        //si se preciona boton que esta en RA4 entonces
delay_ms(50);      //espera 50ms
la();              //despues llama a automatico y todo lo que tenga dentro
}
if(RA0_Bit==1){
delay_ms(50);
si();
}
if(RA2_Bit==1){

```



```
cancion();  
}  
goto inicio;    //regresa a inicio para cumplir el ciclo infinito  
}
```

## Conclusiones

Con la realización de esta práctica podemos tener un panorama más amplio de las aplicaciones que se le puede dar a la utilización de todo lo aprendido durante las clases el como con ayuda de un microcontrolador podemos trabajar de manera didáctica como lo fue en este caso con un piano que nos ayudó a obtener distintas frecuencias que al poder apreciarlo en la vida real pudimos darnos cuenta que esto es una clara aplicación de la vida real y que podemos sacarle mas provecho a lo aprendido y poder tener como aprendizaje a que lo que vimos no esta nada alejado de la vida real y que podemos llegar a generar grandes proyectos con la base de todo el estudio que se le dio a este microcontrolador y que se puede dar una utilización en distintos campos, obteniendo resultados satisfactorios.

Enlace del video: <https://youtu.be/zyWbBQqjfZU>

## Bibliografía

Microchip PIC18F Instruction Set. (2021). Retrieved 9 February 2021, from [http://technology.niagarac.on.ca/staff/mboldin/18F\\_Instruction\\_Set/](http://technology.niagarac.on.ca/staff/mboldin/18F_Instruction_Set/)

(2021). Retrieved 23 February 2021, from <http://ww1.microchip.com/downloads/en/devicedoc/33014j.pdf>

Rafiquzzaman, M., 2007. *Microprocessors theory and applications*. 2nd ed. New Delhi, II: Prentice-Hall of India.

Medrano, G. D., Nandayapa, M., Ramos, L. R. R., Ponce, U. I., Jorge, R. R., & Portilla, L. R. V. (2016). Diseño de un PLC Basado en Microcontrolador para Secuencias Programables Bajo Lenguaje Basic. *Cultura Científica y Tecnológica*, (58).

Zapata, O. E. B. (2011). *Microcontroladores PIC con programación PBP*. Grupo Editorial RA-MA.

Download.mikroe.com. 2021. *Sound Library*. [online] Available at: <[https://download.mikroe.com/documents/compilers/mikroc/pic/help/sound\\_library.htm](https://download.mikroe.com/documents/compilers/mikroc/pic/help/sound_library.htm)> [Accessed 19 May 2021].

MikroElektronika. 2021. *programar-los-pic-utilizando-mikroc-pro-for-pic - MikroElektronika*. [online] Available at: <<https://www.mikroe.com/ebooks/microcontroladores-pic-programacion-en-c-con-ejemplos/programar-los-pic-utilizando-mikroc-pro-for-pic>> [Accessed 19 May 2021].