Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

Asignatura: Control Digital y Aplicaciones/

Control de Procesos por Computadora.

Práctica 1: Desarrollo de tonos por PWM con Arduino.

Por:

Cortes Vázquez Edmundo Miguel

Durán Santos Martín

López Marcos Fernando

Profesor: Dr. Jaime Julián Cid Monjaraz.

Periodo: Primavera 2013.

OBJETIVO

- Diseñar un sistema que genere melodías por medio de la modulación por ancho de pulso, utilizando la tarjeta de desarrollo Arduino.

MARCO TEÓRICO

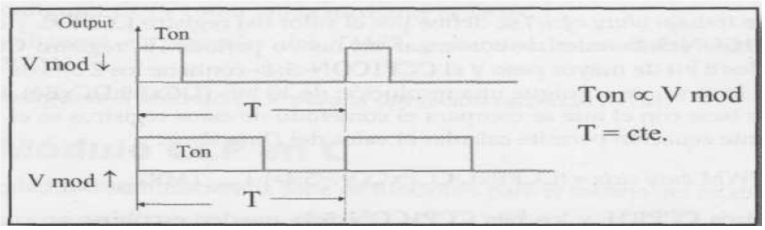
PROYECTO ARDUINO

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. La plataforma Arduino está basada en microcontroladores y las placas son programables mediante el lenguaje de programación Arduino (basasdo en el lenguaje Wiring, el cual tiene gran similitud con C) y el entorno de desarrollo Arduino (basado en Processing).

El desarrollo de proyectos con Arduino ofrece numerosas ventajas. La más importante de ellas es la flexibilidad, ya que Arduino opera en diferentes sistemas operativos (Windows, Mac y Linux), a diferencia de otros entornos de desarrollo de sistemas basados en microcontroladores que solamente trabajan con Windows, además de que tanto el hardware como el software son fácilmente personalizables por pertenecer a una plataforma de código abierto.

MODULACIÓN POR ANCHO DE PULSO (PWM)

El modo PWM (Pulse Width Modulation) o MODULACIÓN DE ANCHO DE PULSO, permite obtener de los pines una señal periódica en la que se puede modificar su ciclo de trabajo (Duty Cicle). Es decir, puede variarse el tiempo en el cual la señal está a nivel alto (TON) frente al tiempo que está a nivel bajo (TOFF), la cual se muestra en la figura. De esta forma, la tensión media aplicada a la carga es proporcional al tiempo de activación en alto TON, controlando, por ejemplo, la velocidad de motores, luminosidad de lámparas, etc.



Como observamos, T es el periodo, el cual es constante, TON es el tiempo de activación en alto, y observamos que existe una relación directa con el Voltaje promedio. Es decir, mientras el tiempo de activación en alto sea mayor, el voltaje promedio Vmod también lo será, por lo que, por ejemplo, la velocidad del motor o la intensidad luminosa serán mayores, según sea el caso.

DIVISIÓN DE LA ESCALA MUSICAL

En primera instancia definiremos el concepto de octava, que consiste en un conjunto de sonidos tal que la frecuencia de uno de los tonos que delimitan a éste sea el doble que la del otro. Para conocer el número de octavas que se encuentran en un intervalo determinado, basta con hallar el cociente de los logaritmos en base dos de las frecuencias de los sonidos que delimitan al intervalo. Por ejemplo, podemos ver que en la escala audible existen poco más de nueve octavas.

$$O = \log_2\left(\frac{f_2}{f_1}\right) = \log_2\left(\frac{20000\,Hz.}{20\,Hz.}\right) \cong 9.965$$

A partir de esto, convencionalmente se divide cada octava en doce semitonos. Si conocemos la frecuencia fundamental de uno de ellos, podemos obtener la del siguiente multiplicando por la raíz doceava de dos. Siguiendo ésta regla y basándonos en que el tono "La" de la tercera octava posee una frecuencia de 440Hz, podemos obtener lo siguiente:

Nota	Octava								
	0	1	2	3	4	5	6	7	8
Do	32.72	65.44	130.88	261.76	523.52	1047.04	2094.08	4188.16	8376.32
Do#	34.66	69.32	138.65	277.31	554.61	1109.23	2218.46	4436.93	8873.87
Re	36.72	73.44	146.89	293.78	578.56	1175.12	2350.24	4700.49	9400.98
Mib	38.9	77.8	155.61	311.23	622.46	1244.92	2489.84	4979.69	9959.39
Mi	41.21	82.42	164.85	329.72	659.43	1318.87	2637.64	5275.49	10550.98
Fa	43.60	87.31	174.64	349.31	698.6	1397.21	2794.42	5588.8	11177.71
Fa#	46.26	92.5	185.01	370.06	740.1	1480.2	2960.41	5920.43	11841.67
Sol	49	98	196	392.04	784.06	1568.13	3136.26	6272.53	12545.06
Sol#	51.91	103.82	207.64	415.32	830.64	1661.28	3322.56	6645.12	13290.24
La	54.99	109.98	219.98	440	879.98	1759.96	3519.92	7039.84	14079.68
Sib	58.26	116.52	233.04	466.13	932.25	1864.5	3729	7458	14916.02
Si	61.72	123.44	246.89	493.82	987.62	1975.25	3950.5	7901.01	15802.03

DESARROLLO

DESCRIPCIÓN

En el lenguaje Arduino, se utilizó la función “tone”, el cual, por medio de las salidas PWM genera tonos a cierta frecuencia y con cierta duración de tiempo.

La sintaxis de la función es así:

Tone(pin, frequency, duration)

Donde “pin” es la salida empleada (Ojo: solo pueden ocuparse las salidas PWM de la tarjeta), “frequency” es la frecuencia del tono que se va a generar y “duration” es la duración del tono.

Para poder generar melodías con ésta función, fue necesario conseguir las respectivas partituras y a partir de ahí generar el respectivo código de duración de tonos y silencios para que el sonido generado por la tarjeta fuera coherente al sonido generado con un instrumento.

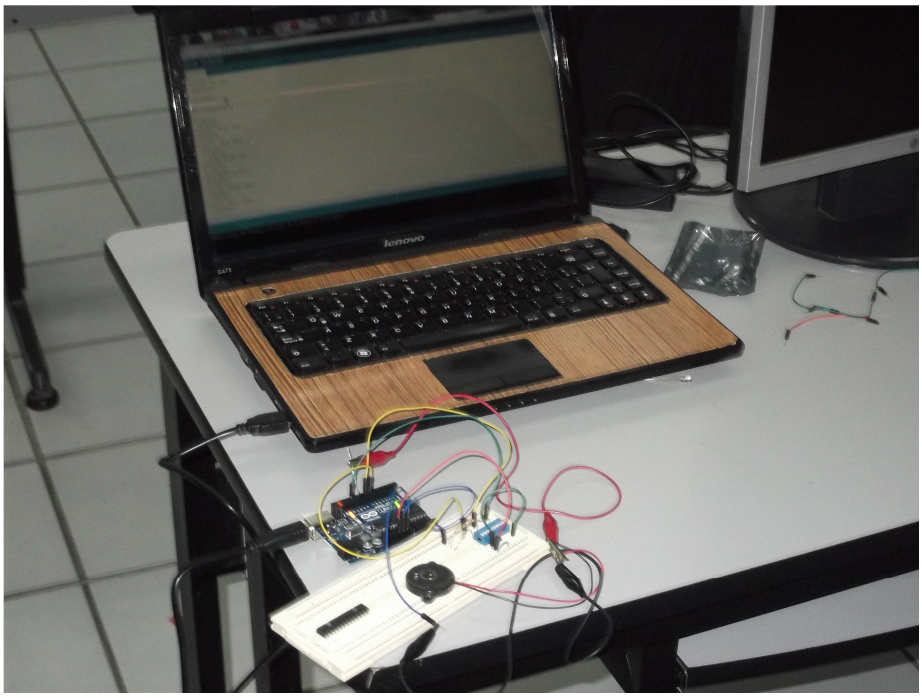
Las canciones interpretadas fueron las siguientes:

- Für Elise, de Ludwig van Beethoven, la cual venía prototipada en uno de los códigos abiertos del lenguaje Arduino.
- Chariots of Fire, de Vangelis.
- Clocks, de Coldplay.

Se implementó el sistema con una serie de “push button”, para que cada vez que se oprimiera uno de ellos, se generara una melodía diferente. En la fotografía de la implementación se puede observar un “dip switch”, pero las entradas fueron prototipadas para ser utilizadas con “push button”.

A la salida de la tarjeta, fue conectado un altavoz piezoeléctrico para poder escuchar las canciones.

IMPLEMENTACIÓN



CONCLUSIONES

- Es importante revisar los diferentes códigos desarrollados con Arduino, ya que al ser de "Open-Source", pueden ser utilizados sin pagar licencias por el usuario para fines académicos.

- La modulación por ancho de pulso resulta muy útil en una gran variedad de áreas de la electrónica, de ahí la importancia de comprender sus principios.

REFERENCIAS

- GARCÍA, Eduardo. *Compilador C CCS y simulador PROTEUS para Microcontroladores PIC*. Editorial Alfaomega, México 2008.
- Recuero López, Manuel. *Ingeniería Acústica*. Paraninfo, Madrid, 2000.
- Página del proyecto Arduino: www.arduino.cc
- Tutorial para interpretar "Chariots of Fire": https://www.youtube.com/watch?v=flVrzj6L6_s
- Tutorial para interpretar "Clocks": https://www.youtube.com/watch?v=UQ5pIbFxy18

ANEXOS

CÓDIGO REALIZADO

```
const int BOTON1=7;
const int BOTON2=8;
const int BOTON3=12;

int puch1=0;
int estado1=0;
int puch2=0;
int estado2=0;
int puch3=0;
int estado3=0;



void setup()
{
  pinMode(10, OUTPUT);
  pinMode(BOTON1,INPUT);
   pinMode(BOTON2,INPUT);
    pinMode(BOTON3,INPUT);
}


void loop() {
puch1=digitalRead(BOTON1);
puch2=digitalRead(BOTON2);
puch3=digitalRead(BOTON3);

//Secuencia Melodía 1
if (puch1==HIGH)
{
//Re#4
  delay(500);
  tone(10, 622.46, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
  delay(161);
//Sol3
  tone(10, 392.04, 300);
  delay(161);
 //Re#4
  tone(10, 622.46, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
  delay(161);
//Sol3
  tone(10, 392.04, 300);
  delay(161);
 //Re#4
  tone(10, 622.46, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
  delay(161);
//////////////////////////////
//////////////////////////
//Do#4
  delay(20);
  tone(10, 554.61, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
  delay(161);
//Fa3
  tone(10, 349.31, 300);
  delay(161);
//Do#4
  tone(10, 554.61, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
  delay(161);
//Fa3
  tone(10, 349.31, 300);
  delay(161);
 //Do#4
  tone(10, 554.61, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
  delay(161);
  delay(20);
//////////////////////////////
//////////////////////////
//Do#4
  tone(10, 554.61, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
```

```
  delay(161);
//Fa3
  tone(10, 349.31, 300);
  delay(161);
//Do#4
  tone(10, 554.61, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
  delay(161);
//Fa3
  tone(10, 349.31, 300);
  delay(161);
 //Do#4
  tone(10, 554.61, 300);
  delay(161);
//Sib3
  tone(10, 466.13, 300);
  delay(161);
////////////////////////////////
//////////////////////
 delay(20);
//Do#4
  tone(10, 523.52, 300);
  delay(161);
//Sib3
  tone(10, 415.32, 300);
  delay(161);
//Fa3
  tone(10, 349.31, 300);
  delay(161);
 //Do#4
  tone(10, 523.52, 300);
  delay(161);
//Sib3
  tone(10, 415.32, 300);
  delay(161);
//Fa3
  tone(10, 349.31, 300);
  delay(161);

  //Do#4
  tone(10, 523.52, 300);
  delay(161);
//Sib3
  tone(10, 415.32, 300);
  delay(161);
  delay(300);
}
```

```
//Secuencia melodía 2
else if (puch2==HIGH)
{

  // play e4
  delay(600);
  tone(10, 329.63, 300);
  delay(350);
  // play d4#
  tone(10, 311.13, 300);
  delay(350);
  // play e4
  tone( 10, 329.63, 300);
  delay(350);
  // play d4#
  tone( 10,311.13, 300);
  delay(350);
  // play e4
  tone(10, 329.63, 300);
  delay(350);
  // play b3
  tone( 10, 246.94, 300);
  delay(400);
  // play d4
  tone(10, 293.66,300);
  delay(400);
  // play c4
  tone(10, 261.63,300);
  delay(400);
  // play a3
  tone(10, 220, 900);
  delay(1000);
  // play d3
  tone(10,146.83, 300);
  delay(350);
  //play f3
  tone(10, 174.61, 300);
  delay(400);
  //play a3
  tone(10, 220, 300);
  delay(400);
  // play b3
  tone(10, 246.94, 900);
  delay(1000);
  // play f3
  tone(10, 174.61, 300);
  delay(400);
  // play a3#
  tone(10, 233.08, 300);
  delay(400);
```

```
// play b3
tone(10, 246.94, 300);
delay(400);
// play c4
tone(10, 261.63, 900);
delay(1000);
delay(300);
// play e4
tone(10, 329.63, 300);
delay(400);
// play d4#
tone(10, 311.13, 300);
delay(400);
// play e4
tone(10, 329.63, 300);
delay(400);
// play d4#
tone(10, 311.13, 300);
delay(400);
// play e4
tone(10, 329.63, 300);
delay(400);
// play b3
tone(10, 246.94, 300);
delay(400);
// play d4
tone(10, 293.66, 300);
delay(400);
// play c4
tone(10, 261.63, 300);
delay(400);
// play a3
tone(10, 220, 900);
delay(1000);
// play d3
tone(10, 146.83, 300);
delay(400);
// play f3
tone(10, 174.61, 300);
delay(400);
// play a3
tone(10, 220, 300);
delay(400);
// play b3
tone(10, 246.94, 900);
delay(1000);
// play f3
tone(10, 174.61, 300);
delay(400);
// play c4
tone(10, 261.63, 300);
delay(400);
// play b3
tone(10, 246.94, 300);
delay(400);
// play a3
tone(10, 220, 900);
delay(1000);
// play b3
tone(10,246.94, 300);
delay(400);
 // play c4
tone(10, 261.63, 300);
delay(400);
// play d4
tone(10, 293.66, 300);
delay(400);
// play e4
tone(10, 329.63, 900);
delay(1000);
// play g3
tone(10, 196, 300);
delay(400);
// play f4
tone(10, 349.23, 300);
delay(400);
//play e4
tone(10, 329.23, 300);
delay(400);
// play d4
tone(10, 293.63, 900);
delay(1000);
// play e3
tone(10,164.81, 300);
delay(400);
// play e4
tone(10, 329.63, 300);
delay(400);
// play d4
tone(10, 293.63, 300);
delay(400);
// play c4
tone(10, 261.63, 900);
delay(1000);
// play d3
tone(10, 146.83, 300);
delay(400);
  // play d4
tone(10, 293.63, 300);
delay(400);
```

```
// play c4
tone(10, 261.63, 300);
delay(400);
// play b3
tone(10, 246.94, 900);
delay(1000);
delay(400);
// play e4
tone(10, 329.63, 300);
delay(400);
// play d4#
tone(10, 311.13, 300);
delay(350);
// play e4
tone( 10, 329.63, 300);
delay(350);
// play d4#
tone( 10,311.13, 300);
delay(350);
// play e4
tone(10, 329.63, 300);
delay(350);
// play b3
tone( 10, 246.94, 300);
delay(400);
// play d4
tone(10, 293.66,300);
delay(400);
// play c4
tone(10, 261.63,300);
delay(400);
// play a3
tone(10, 220, 900);
delay(1000);
// play d3
tone(10,146.83, 300);
delay(350);
//play f3
tone(10, 174.61, 300);
delay(400);
//play a3
tone(10, 220, 300);
delay(400);
// play b3
tone(10, 246.94, 900);
delay(1000);
// play f3
tone(10, 174.61, 300);
delay(400);
// play a3
tone(10, 233.08, 300);
delay(400);
// play b3
tone(10, 246.94, 300);
delay(400);
// play c4
tone(10, 261.63, 900);
delay(1000);
delay(300);
// play e4
tone(10, 329.63, 300);
delay(400);
// play d4#
tone(10, 311.13, 300);
delay(400);
// play e4
tone(10, 329.63, 300);
delay(400);
// play d4#
tone(10, 311.13, 300);
delay(400);
// play e4
tone(10, 329.63, 300);
delay(400);
// play b3
tone(10, 246.94, 300);
delay(400);
// play d4
tone(10, 293.66, 300);
delay(400);
// play c4
tone(10, 261.63, 300);
delay(400);
// play a3
tone(10, 220, 900);
delay(1000);
// play d3
tone(10, 146.83, 300);
delay(400);
// play f3
tone(10, 174.61, 300);
delay(400);
// play a3
tone(10, 220, 300);
delay(400);
// play b3
tone(10, 246.94, 900);
delay(1000);
// play f3
tone(10, 174.61, 300);
```

```
  delay(400);
  // play c4
  tone(10, 261.63, 300);
  delay(400);
  // play b3
  tone(10, 246.94, 300);
  delay(400);
  // play a3
  tone(10, 220, 900);
  delay(1000);

  delay(5000);
}
else if (puch3==HIGH)
{
// play do#
  delay(250);
  tone(10, 554.61, 600);
  delay(250);
  // play fa#
  tone(10, 740.1, 600);
  delay(250);
  // play sol#
  tone( 10, 830.64, 600);
  delay(250);
  // play sib
  tone( 10,932.25, 600);
  delay(250);
  // play sol#
  tone( 10, 830.64, 1200);
  delay(500);
  // play fa
  tone(10, 698.6, 2400);
  delay(1000);
 ////////////////////////
// play do#
  delay(250);
  tone(10, 554.61, 600);
  delay(250);
  // play fa#
  tone(10, 740.1, 600);
  delay(250);
  // play sol#
  tone( 10, 830.64, 600);
  delay(250);
  // play sib
  tone( 10,932.25, 600);
  delay(250);
  // play sol#
  tone( 10, 830.64, 3600);
  delay(1500);
//////////////////////////////
///

  // play do#
  delay(250);
  tone(10, 554.61, 600);
  delay(250);
  // play fa#
  tone(10, 740.1, 600);
  delay(250);
  // play sol#
  tone( 10, 830.64, 600);
  delay(250);
  // play sib
  tone( 10,932.25, 600);
  delay(250);
  // play sol#
  tone( 10, 830.64, 1200);
  delay(250);
  // play fa
  tone(10, 698.6, 2400);
  delay(00);
//////////////////////////////
/////
// play fa
  tone(10, 698.6, 600);
  delay(250);
 // play fa#
  tone(10, 740.1, 600);
  delay(250);
// play fa
  tone(10, 698.6, 600);
  delay(250);
// play do#
  delay(250);
  tone(10, 554.61, 600);
  delay(250);
  delay(30);
// play do#
  delay(250);
  tone(10, 554.61, 3600);
  delay(1500);
  delay (1000);

}
```