



Universidad Politécnica de Madrid

# Grado en Ingeniería Electrónica y Automática

Asignatura  
**INFORMÁTICA**

Curso 2018-2019



Universidad Politécnica de Madrid

# Grado en Ingeniería Electrónica y Automática

## Datos del Grupo

*Natalia Borlaf Nieto, 54516*

*Daniel de la Cruz Redondo, 54938*

## Introducción

En el presente trabajo se ha desarrollado un código que, a partir de una melodía y dependiendo de la duración de las notas, hace que se encienda un color determinado de LED.

## Desarrollo

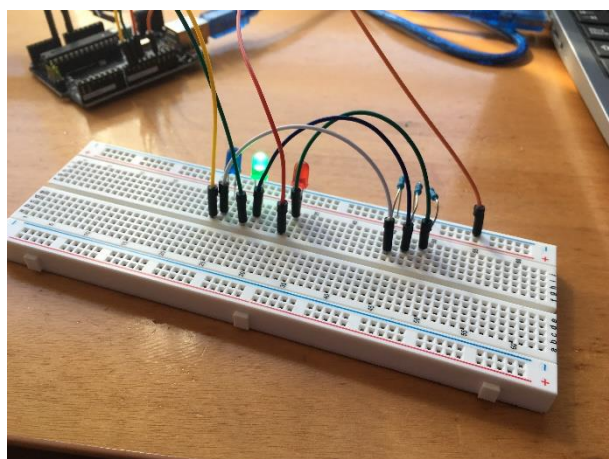
- **Material utilizado en el arduino:**
  - Tres luces LED (rojo, verde y azul)
  - Un zumbador
  - Cables macho-macho
  - Cables hembra-macho
  - Protoboard
  - UNOR3 controller board
  - Tres resistencias de 220 ohmios
  - Cable USB

- **Procedimiento:**

Para empezar el trabajo hicimos una búsqueda de información y la prueba del Arduino con programas más sencillos, como apagado y encendido de LEDs, o variar la frecuencia del zumbador.

Conectamos un cable macho-macho de la placa base a la protoboard al puerto GND, al igual que el zumbador al puerto A0, las resistencias y los LEDs a los puertos 9 (azul), 10 (verde) y 11 (rojo).

Al desarrollar el código probamos con vectores para la frecuencia, duración y color de los LED, hasta asociar un color a una duración de una nota. El uso de la asignación dinámica de memoria se hace asociado a las notas.



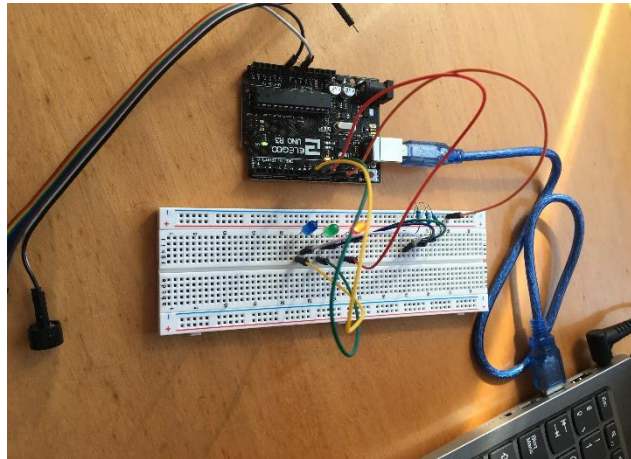
- **Explicación del código:**

Definimos los puertos en los que hemos conectado los LEDs y el zumbador. Damos un valor de frecuencia y de tiempo a cada nota. Creamos dos vectores: uno con las notas para la frecuencia y el otro con la duración de cada nota.

En este ejemplo hemos elegido un Waltz de Frederic Chopin.

Usamos la asignación dinámica de memoria para medir el tamaño del vector de las notas. Con un bucle “**for**” asociamos a cada duración de la nota un color de LED; en esta melodía se usa **verde** para *negra*, **roja** para *corchea* y **azul** para *blanca*, aunque también hemos asociado colores a semicorcheas y fusas.

Con otro bucle “**for**” desde 0 a la dimensión del vector lee el color que hemos asociado, la frecuencia y la duración.



- **Código:**

```
#include "pitches.h"
#define BLUE 9
#define GREEN 10
#define RED 11
#define ZUM A0
```

```
long DO=261.626,
    DoS=277.183,
    RE=293.665,
    RES=311.127,
    MI=329.628,
    FA=349.228,
    FAS=369.994,
    SOL=391.995,
    SOLS=415.305,
    LA=440,
    LAS=466.164,
    SI=493.883,
    DO2=523.251,
    DOS2=554.365,
```

```
RE2=587.330,  
RES2=622.254,  
MI2=659.255,  
FA2=698.456,  
SOL2=783.991,  
LA2=880,  
PAU=30000; //pausa
```

```
int dim;  
int i;
```

```
int melodia[]={MI, LA, SI, DO2, DO2, RE2, MI2, FA2, SI, DO2, RE2, LA2, SOL2, FA2, MI2,  
RES2, MI2, LA, SI, DO2, DO2, RE2, MI2, FA2, SI, DO2, RE2, LA2, SOL2, SI, DO2};  
int b=2000, n=1000, c=500, s=250, f=125;  
int duracionNota[] = {n, c, c, n, n, c, c, b, c, c, c, c, c, c, c, b, c, c, n, n, c, c, b, c, c, c, c,  
c, c, n};
```

```
void setup() {  
    dim=(sizeof(melodia) / sizeof(melodia[0]));
```

```
    pinMode(ZUM, OUTPUT);  
    pinMode(RED, OUTPUT);  
    pinMode(GREEN, OUTPUT);  
    pinMode(BLUE, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    int color[dim];  
    for(i=0; i<dim; i++){  
        if(duracionNota[i]==n){  
            color[i]= GREEN;  
        }  
        if(duracionNota[i]==c){  
            color[i]= RED;  
        }  
        if(duracionNota[i]==b){  
            color[i]= BLUE;  
        }  
        if(duracionNota[i]==s){  
            color[i]= BLUE;  
        }  
        if(duracionNota[i]==f){  
            color[i]= BLUE;  
        }  
    }
```

```
}
```

```
for(int notaActual=0; notaActual<dim; notaActual++){  
    digitalWrite(color[notaActual], HIGH);
```

```
tone(ZUM, melodia[notaActual]);  
delay(duracionNota[notaActual]);  
digitalWrite(color[notaActual], LOW);  
noTone(ZUM);  
}  
  
}
```

## Conclusiones

El código se puede rentabilizar mucho con la asignación dinámica de memoria y con los bucles asignando a cada valor otro elegido (color asignado a duración). La programación en Arduino es muy parecida a la programación en C.

## Bibliografía

- Duración de notas: <https://juegosrobotica.es/musica-con-arduino/>
- Conexión LEDs:  
[http://www.practicasconarduino.com/manualrapido/secuencia\\_de\\_led.html](http://www.practicasconarduino.com/manualrapido/secuencia_de_led.html)
- Tabla de frecuencias:  
[https://es.wikipedia.org/wiki/Frecuencias\\_de\\_afinaci%C3%B3n\\_del\\_piano](https://es.wikipedia.org/wiki/Frecuencias_de_afinaci%C3%B3n_del_piano)
- Combinación LEDs con música:  
<http://www.iescamp.es/miarduino/2016/02/15/parpadeo-de-leds-al-son-de-la-musica/>
- Asignación dinámica de memoria:  
<https://www.arduino.cc/reference/en/language/variables/utilities/sizeof/>
- <https://forum.arduino.cc/index.php?topic=265882.0>
- PDF tutorial Super Starter Kit de Elegoo