

Melodía Star Wars

Arduino



Miguel Torres Medina y Jorge León Sánchez

Índice

1-Introducción.

2-Código con la explicación.

3-Foto de la placa.

4-Componentes.

Introducción

En este trabajo hemos usado la placa de arduino, junto con un led y un speaker, para reproducir la melodía clásica de la película de Star Wars.

El led se enciende y se apaga al ritmo de la melodía, mientras el speaker reproduce las canciones.

Código con la explicación

```
#define c3 7634  
  
#define d3 6803  
  
#define e3 6061  
  
#define f3 5714  
  
#define g3 5102  
  
#define a3 4545  
  
#define b3 4049  
  
#define c4 3816 // 261 Hz  
  
#define d4 3401 // 294 Hz  
  
#define e4 3030 // 329 Hz  
  
#define f4 2865 // 349 Hz  
  
#define g4 2551 // 392 Hz  
  
#define a4 2272 // 440 Hz  
  
#define a4s 2146  
  
#define b4 2028 // 493 Hz
```

En esta primera parte del código, definimos las notas y su frecuencia, además también se define las pausas entre notas, como una nota especial.

```

#define c5  1912  // 523 Hz

#define d5  1706

#define d5s 1608

#define e5  1517  // 659 Hz

#define f5  1433  // 698 Hz

#define g5  1276

#define a5  1136

#define a5s 1073

#define b5  1012

#define c6  955

        #define R   0   // Define a special note, 'R', to represent a rest


// Melody 1: Star Wars Imperial March

int melody1[] = { a4, R, a4, R, a4, R, f4, R, c5, R, a4, R, f4, R, c5, R, a4, R, e5, R, e5, R, e5, R,
f5, R, c5, R, g5, R, f5, R, c5, R, a4, R};

int beats1[] = { 50, 20, 50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 80, 50, 20, 50, 20,
50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 40};


// Melody 2: Star Wars Theme

int melody2[] = { f4, f4, f4, a4s, f5, d5s, d5, c5, a5s, f5, d5s, d5, c5, a5s, f5, d5s, d5, d5s,
c5};

int beats2[] = { 21, 21, 21, 128, 128, 21, 21, 21, 128, 64, 21, 21, 21, 128, 64, 21, 21, 21,
128 };

int MAX_COUNT = sizeof(melody1) / 2;

long tempo = 10000;

int pause = 1000;

int rest_count = 50;

```

En esta parte se definen dos melodías, con las notas utilizadas, además se define la duración de la melodía, la pausa entre melodías y las conexiones de salida y entrada.

```
int toneM = 0;

int beat = 0;

long duration = 0;

int potVal = 0;

int speakerOut = 9;

int led = 6;


void setup() {

    pinMode(speakerOut, OUTPUT);

    pinMode(led, OUTPUT);
    Serial.begin(9600);

}


void loop() {

    // Melody1
    for (int i=0; i<MAX_COUNT; i++) {
        toneM = melody1[i];
        beat = beats1[i];
        duration = beat * tempo;
        playTone();
        delayMicroseconds(pause);
    }

    // Melody2
```

```
MAX_COUNT = sizeof(melody2) / 2;

for (int i = 0; i < MAX_COUNT; i++) {

    toneM = melody2[i];

    beat = beats2[i];

    duration = beat * tempo;

    playTone();

    delayMicroseconds(pause);

}

}
```

```
void playTone() {

    long elapsed_time = 0;

    if (toneM > 0) {

        digitalWrite(led,HIGH);

        while (elapsed_time < duration) {

            digitalWrite(speakerOut,HIGH);

            delayMicroseconds(toneM / 2);

            digitalWrite(speakerOut, LOW);

            delayMicroseconds(toneM / 2);

            elapsed_time += (toneM);

        }

    }

}
```

```
digitalWrite(led,LOW);
```

```
}
```

```
else {
```

```
for (int j = 0; j < rest_count; j++) {
```

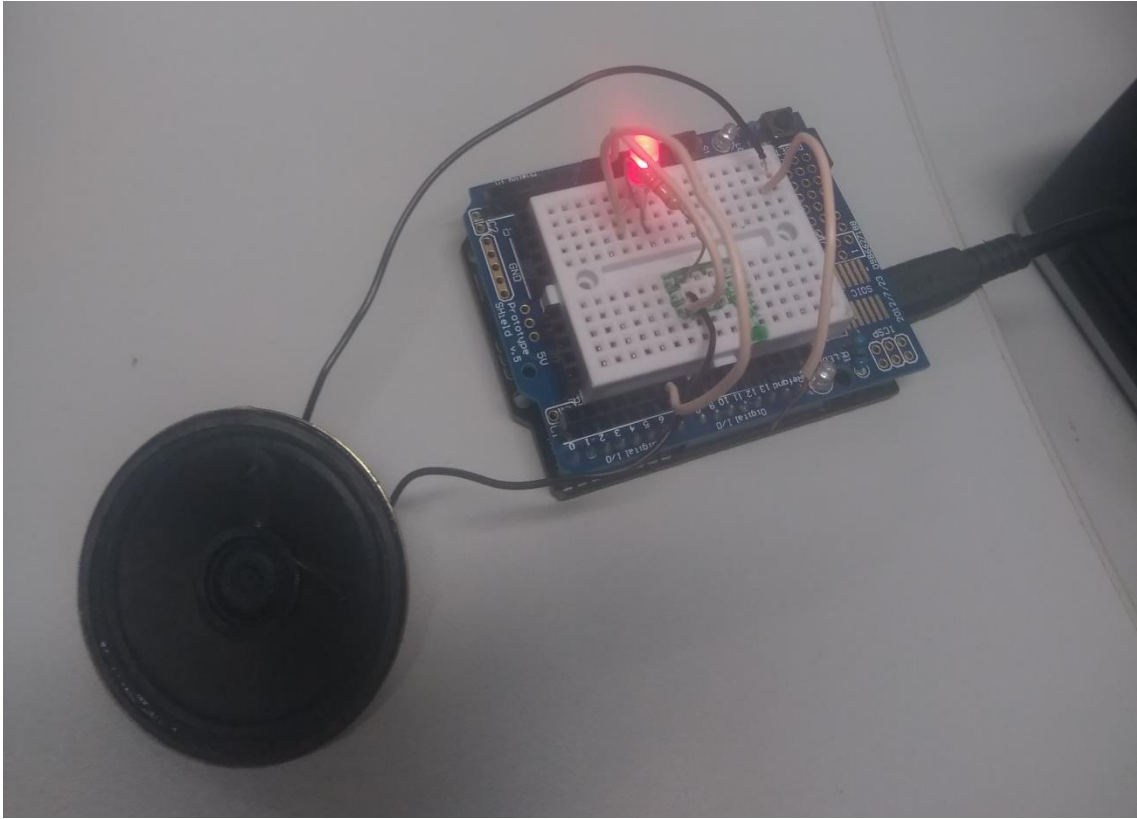
```
delayMicroseconds(duration);
```

```
}
```

```
}
```

```
}
```

Foto de la placa



Componentes

- 1-Placa Arduino.
- 2-Speaker.
- 3-Led.
- 4-Resistencia.
- 5-Cables.