

## ARDUINO : LECTOR

Este código se ejecuta en la placa del lector, su objetivo principal es el siguiente, se introduce por el lector un papel con un mensaje codificado en forma de marcas blancas y negras, se leen con unos sensores IR y se considera una nota asociada al valor, que se tocará

```
1  //declaramos pin sensor referencia
2  const int sensorJefe = 2;
3  // declaramos pines de las notas
4  const int sensorDo = 3;
5  const int sensorRe = 4;
6  const int sensorMi = 5;
7  const int sensorfa = 6;
8  const int sensorSol = 7;
9  const int sensorLa = 8;
10 const int sensorSi = 9;
11 // pin buzzer
12 const int buzzer = 10;
13 bool duplicate = true;
14 //frecuencias notas
15 #define DO 261.626
16 #define RE 293.665
17 #define MI 329.628
18 #define FA 349.228
19 #define SOL 391.995
20 #define LA 440
21 #define SI 493.883
22 //definimos duracion en ms
23 #define DURACION 300
24 ...
```

hoja marcas

Después de definir las variables y unas constantes para las frecuencias de las notas llegamos a las dos funciones principales.

En el *void setup* cabe destacar únicamente que se inicia la comunicación por puerto serial:

Y se ponen todos los pines de los sensores IR como input, utilizando un *for* para ahorrar trabajo y no escribir 8 líneas.

Por última se coloca el pin de buzzer como output

```
24 void setup()
25 {
26   Serial.begin(9600);
27   //ponemos los pines de los sensores en input
28   for (int i = 3; i < 10; i++)
29   {
30     pinMode(i, INPUT);
31   }
32   pinMode(buzzer, OUTPUT);
33
34 }
35
```

En el *void loop* podemos diferenciar 2 partes:

- Lectura de sensores, que se lleva a cabo en el *for*, cabe destacar la variable booleana *duplicate* que está para evitar que se tome una doble lectura, no permitiendo la lectura mientras no se deje de leer el sensor que ejerce de guía.

```
24 void setup()
25 {
26   Serial.begin(9600);
27   //ponemos los pines de los sensores en input
28   for (int i = 3; i < 10; i++)
29   {
30     pinMode(i, INPUT);
31   }
32   pinMode(buzzer, OUTPUT);
33
34 }
35
```

- Ejecución, una vez leídos los valores de los sensores podemos ver qué valor estaban representando y, consecuentemente, ejecutar una u otra nota musical en el *buzzer*.

```
52     switch (value)
53     {
54         case 3:
55             Serial.println("1000000");
56
57             tone(buzzer, DO, DURACION);
58
59             break;
60         case 4:
61             tone(buzzer, RE, DURACION);
62             Serial.println("0100000");
63
64             break;
65         case 5:
66             tone(buzzer, MI, DURACION);
67             Serial.println("0010000");
68
69             break;
70         case 6:
71             tone(buzzer, FA, DURACION);
72             Serial.println("0001000");
73
74             break;
75         case 7:
76             tone(buzzer, SOL, DURACION);
77             Serial.println("0000100");
78
79             break;
80         case 8:
81             tone(buzzer, LA, DURACION);
82             Serial.println("0000010");
83
84             break;
85         case 9:
86             tone(buzzer, SI, DURACION);
87             Serial.println("0000001");
88
89             break;
90     }
91     if (digitalRead(sensorJefe) != HIGH)
92     {
93         duplicate = true;
94     }
95 }
```