

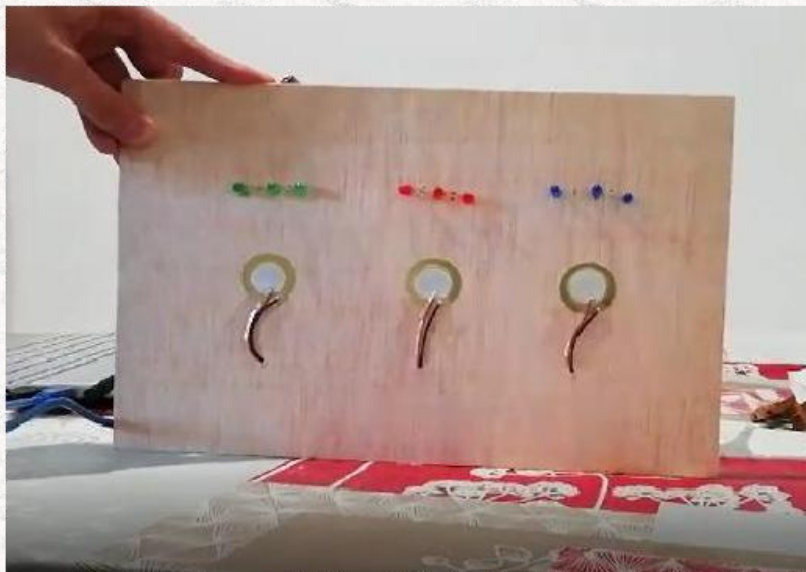
# TRABAJO INFORMÁTICA

## MIEMBROS DEL GRUPO

NOMBRE Y APELLIDOS	Nº DE MATRÍCULA	E-MAL
AIWEN ZHOU	54913	a.zhou@alumnos.upm.es
ZHANYAO LIN	54950	zhanyao.lin@alumnos.upm.es
WEI ZHENG	54911	wei.zheng13@alumnos.upm.es

## RESUMEN DEL PROYECTO: CAJA MUSICAL CON ARDUINO Y DEV C++

Nuestro proyecto consiste en una caja que contiene tres botones, cada uno representa un instrumento diferente: el primero coincide con el sonido de un arpa, el segundo con un gong chino, y el tercero con un platillo. Los botones van a ser unos sensores piezoeléctricos que miden la intensidad de fuerza aplicada, y la convierte en una diferencia potencial. Además hemos instalado tres leds encima de cada botón, que en función de la intensidad de fuerza, brillará una led (poca fuerza), dos leds (un poco más de fuerza) y las tres leds (bastante fuerza). Por otro lado, la diferencia potencial va a ser reproducida por pantalla en dev c++.



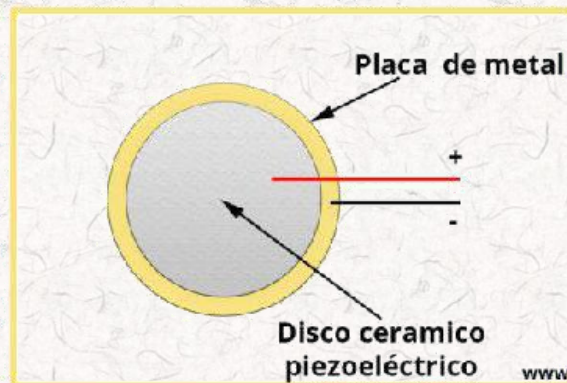


## SENSOR PIEZOELÉCTRICO

### Partes de un sensor piezoeléctrico

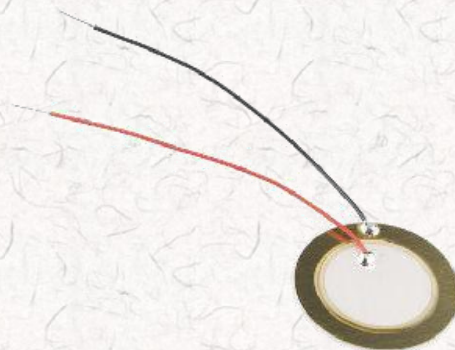
Al tener una construcción demasiado sencilla este sensor consta de 3 componentes principales:

- Material piezoeléctrico
- Placa de metal
- Cables ( Positivo y Negativo)



Para crear un sensor piezoeléctrico y poder generar energía eléctrica con la deformación, primero se debe de tratar el material para reordenar sus cargas, Ya que estas están inicialmente desordenadas y no es posible generar electricidad. Para obtener las propiedades de la piezoeléctricidad, el material se debe someter a un intenso campo eléctrico para ordenar las cargas eléctricas.

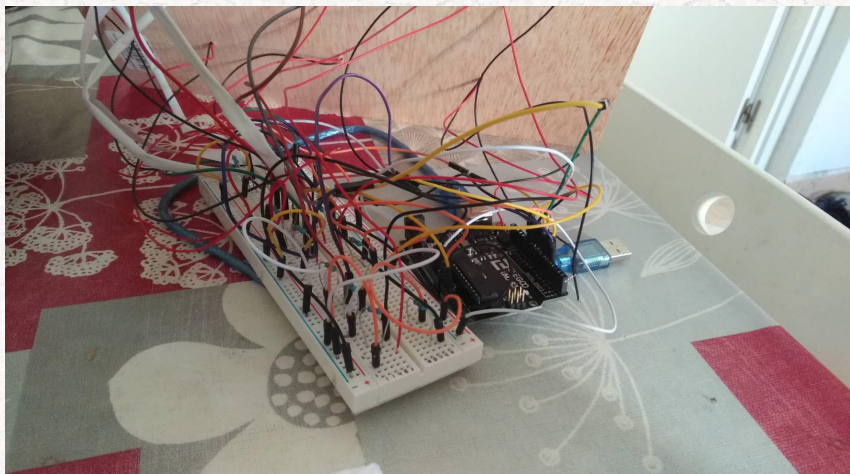
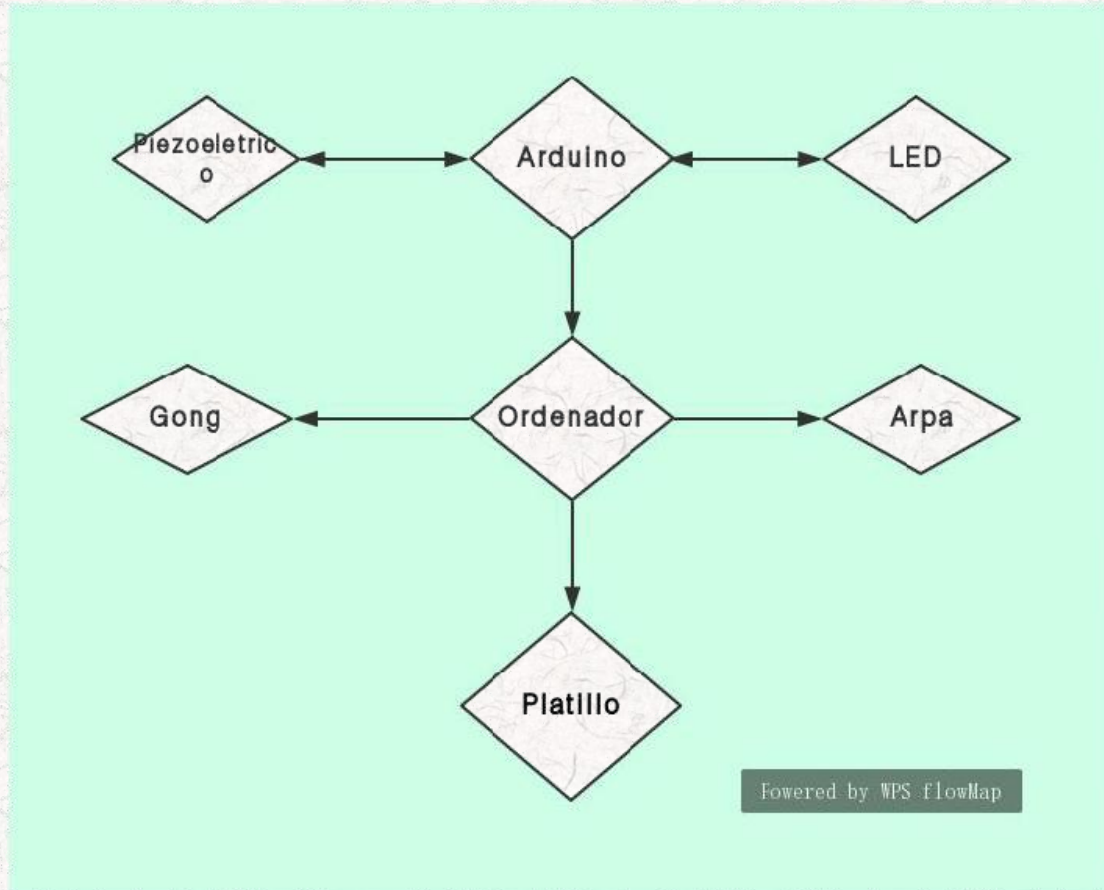
Una vez que se quita el campo eléctrico las cargas quedan libres y cuando se ejerce una presión o deformación se vuelven a ordenar generando la carga eléctrica deseada. Un dato importante sobre este sensor es que no debe sobrepasar una temperatura máxima, llamada "temperatura de curie" ya que al estar por encima de la temperatura máxima, las cargas vuelven a su estado inicial en donde están desordenadas y no serán capaces de generar estas cargas eléctricas.





(FUENTE: <https://www.ingmecafenix.com/automatizacion/sensores/sensor-piezoelectrico/>)

## Diagrama:





## Aplicación del piezoelectrico - Arduino

```
const int sensor1=A0; //se declara la variable para hacer referencia a que el sensor
está conectado en Pin 0
const int sensor2=A1;
const int sensor3=A2;
void setup()
{
  Serial.begin(9600); // Inicializa el serial
  for(int j=2;j<10;j++){
    pinMode(j,OUTPUT);
  }
}
void loop()
{
  int val1= analogRead(sensor1); // lee la señal de entrada del sensor y lo almacena
  en la variable val
  int val2=analogRead(sensor2);
  int val3=analogRead(sensor3);
  if(val1>=50){
    Serial.println("A");
    Serial.println(val1);// Imprime el valor de val en cada renglón
    Serial.println("@");
    if(val1>=50){
      digitalWrite(10,HIGH);
      delay(200);
      digitalWrite(10,LOW);
      //delay(200);
    }
  }
}
```



```
}  
  
if(val1>=200){  
  
    digitalWrite(11,HIGH);  
    delay(200);  
    digitalWrite(11,LOW);  
    //delay(200);  
    }  
if(val1>=600){  
    digitalWrite(12,HIGH);  
    delay(200);  
    digitalWrite(12,LOW);  
    //delay(200);  
    }  
}  
  
if(val2>=50){  
    Serial.println("B");  
    Serial.println(val2); // Imprime el valor de val en cada renglón  
    Serial.println("@");  
    if(val2>=50){  
        digitalWrite(9,HIGH);  
        delay(200);  
        digitalWrite(9,LOW);  
        //delay(200);  
    }  
    if(val2>=200){  
        digitalWrite(8,HIGH);  
        delay(200);  
        digitalWrite(8,LOW);  
        // delay(200);  
    }  
}
```



```
if(val2>=600){  
    digitalWrite(7,HIGH);  
    delay(200);  
    digitalWrite(7,LOW);  
    //delay(200);  
}  
}  
  
if(val3>=50){  
    Serial.println("C");  
    Serial.println(val3); // Imprime el valor de val en cada renglón  
    Serial.println("@");  
    if(val3>=50){  
        digitalWrite(6,HIGH);  
        delay(200);  
        digitalWrite(6,LOW);  
        //delay(200);  
    }  
    if(val3>=200){  
        digitalWrite(5,HIGH);  
        delay(200);  
        digitalWrite(5,LOW);  
        //delay(200);  
    }  
    if(val3>=600){  
        digitalWrite(4,HIGH);  
        delay(200);  
        digitalWrite(4,LOW);  
        //delay(200);  
    }  
}
```



```
delay(200);
```

```
}
```

## **Comunicación Serie – Arduino ↔ Ordenador**

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include "SerialPort.h"
```

```
#include <windows.h>
```

```
#include <mmsystem.h>
```

```
#define MAX_DATA_LENGTH 255
```

```
// Funciones prototipo
```

```
void autoConnect(SerialPort *arduino,char*);
```

```
int main(void)
```

```
{
```

```
    int num;
```

```
    //Arduino SerialPort object
```

```
    SerialPort *arduino;
```

```
    // Puerto serie en el que está Arduino
```

```
    char* portName = "\\\\.\\COM3";
```

```
    // Buffer para datos procedentes de Arduino
```

```
    char incomingData[MAX_DATA_LENGTH];
```

```
    // Crear estructura de datos del puerto serie
```

```
    arduino = (SerialPort *)malloc(sizeof(SerialPort));
```

```
    // Apertura del puerto serie
```

```
    crear_Conexion(arduino,portName);
```

```
    autoConnect(arduino,incomingData);
```

```
}
```

```
void autoConnect(SerialPort *arduino,char *incomingData)
```

```
{
```



```

int readResult;

int i,n,j;

char fichero1[]="arpa.wav";
char fichero2[]="gong.wav";
char fichero3[]="platillo.wav";

// Espera la conexión con Arduino
while (!isConnected(arduino))

{
    Sleep(100);
    Crear_Conexion(arduino,arduino->portName);
}

//Comprueba si arduino está conectado
if (isConnected(arduino))
{
    printf ("Conectado con Arduino en el puerto %s\n",arduino->portName);
}

// Bucle de la aplicación
while (isConnected(arduino))
{
    readResult = readSerialPort(arduino,incomingData,MAX_DATA_LENGTH); if
    (readResult!=0){
        printf("%s", incomingData);
        //printf("\n%d\n",readResult);
        for(i=0;i<MAX_DATA_LENGTH;i++){
            n=0;
            if(incomingData[i]=='A'){
                for(j=i+1;incomingData[j]!='@';j++){
                    {
                        if (incomingData[j]>='0' && incomingData[j]<='9')
                            n=n*10+incomingData[j]-'0';
                    }
                }
            }
        }
    }
}

```



```

        PlaySound(fichero1,NULL,SND_FILENAME | SND_ASYNC);

        if(incomingData[i]=='B'){
            for(j=i+1;incomingData[j]!='@';j++){
                if (incomingData[j]>='0' && incomingData[j]<='9')
                    n=n*10+incomingData[j]-'0';
            }

            PlaySound(fichero2,NULL,SND_FILENAME | SND_ASYNC);
        }

        if(incomingData[i]=='C'){
            for(j=i+1;incomingData[j]!='@';j++){
                if (incomingData[j]>='0' && incomingData[j]<='9')
                    n=n*10+incomingData[j]-'0';
            }

            PlaySound(fichero3,NULL,SND_FILENAME | SND_ASYNC);
        }
    }
}

Sleep(10);

If ( !isConnected(arduino))
    printf ("Se ha perdido la conexión con Arduino\n");
}

```