

Instrumento teclado

Objetivo General.

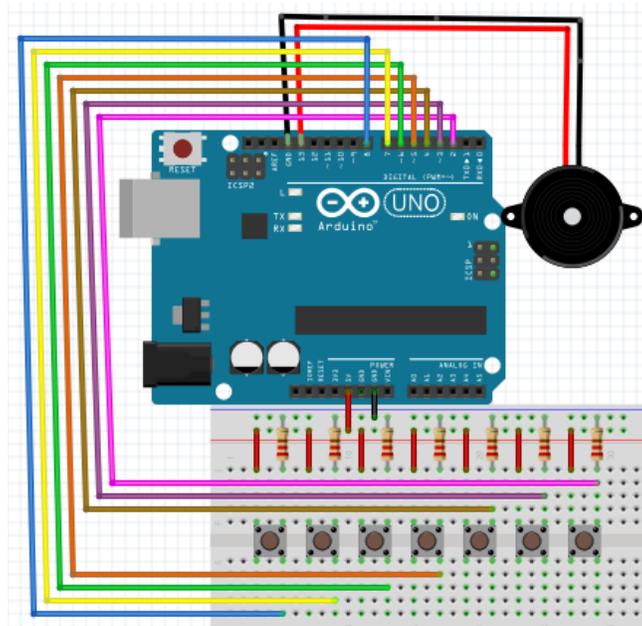
Aprender a utilizar la función: `tone()` de Arduino para crear diferentes frecuencias (Hz) y escucharlas mediante un parlante.

Desarrollo de la Actividad Práctica.

1. Realizar las conexiones.
2. Cargar código en la placa Arduino.
3. Comprobar el funcionamiento de la práctica.

Materiales	
Cantidad	Modelo
1	ARD-010
1	USB-476
1	ARD-310
1	BGD
7	AU-101
7	R220 1W
1	509-010

Diagrama de Conexión



Código

```
const int NOTA_DO = 262;      //Se declara la frecuencia DO.
const int NOTA_RE = 294;      //Se declara la frecuencia RE.
const int NOTA_MI = 330;      //Se declara la frecuencia MI.
const int NOTA_FA = 349;      //Se declara la frecuencia FA.
const int NOTA_SOL = 392;     //Se declara la frecuencia SOL.
const int NOTA_LA = 440;     //Se declara la frecuencia LA.
const int NOTA_SI = 494;     //Se declara la frecuencia SI.

// se declaran los pines asociados a las notas básicas
const int pulsador_DO = 8;    //Se declara en pin DO.
const int pulsador_RE = 7;    //Se declara en pin RE.
const int pulsador_MI = 6;    //Se declara en pin MI.
const int pulsador_FA = 5;    //Se declara en pin FA.
const int pulsador_SOL = 4;   //Se declara en pin SOL.
const int pulsador_LA = 3;    //Se declara en pin LA.
const int pulsador_SI = 2;    //Se declara en pin SI.

// se define las notas que se reproducirán
int notas_basicas[] = {NOTA_DO, NOTA_RE, NOTA_MI, NOTA_FA, NOTA_SOL, NOTA_LA,
NOTA_SI}; //Se crea un Array de las notas.
int pulsadores [ ] = {pulsador_DO, pulsador_RE, pulsador_MI, pulsador_FA,
pulsador_SOL, pulsador_LA, pulsador_SI}; //Se crea un Array de los pines.
int pulsador_activated = 0; //Variable que verifica el estado de los pines.

void setup() { //Inicia función setup.
// se define los pines de entrada.
for (int pin = 6; pin > 0; pin--) {
pinMode(pulsadores[pin] , INPUT);
}
} //Termina función setup.

void loop() { //Inicia función loop.
int pulsador_actual; // se declara como variable local.
// el bucle analiza el estado de los 7 pulsadores.
for (int pulsador = 0; pulsador < 7; pulsador++) {
pulsador_actual = digitalRead(pulsadores[pulsador]); // se lee el estado del
pulsador.
// si se presionó, se procede a reproducir la nota asociada.
if (pulsador_actual == HIGH) {
tone(13, notas_basicas[pulsador]); // reproduce la nota.
// Grabamos el pulsador que ha sido apretado.
pulsador_activated = pulsadores[pulsador];
break;
}
}
// su soltamos el pulsador activo apagamos cortamos el sonido.
if (digitalRead(pulsador_activated) == LOW) {
noTone(13); //detén el buzzer.
}
}
}
```

Conclusión

Se observa que al presionar los botones se producirá un tono en el buzzer, este cambiará la frecuencia con la que lo produce dependiendo del botón pulsado, esta frecuencia es posible modificarla al inicio del código, las notas que se reproducirán son: Do, Re, Mi, Fa, Sol, La, Si.