

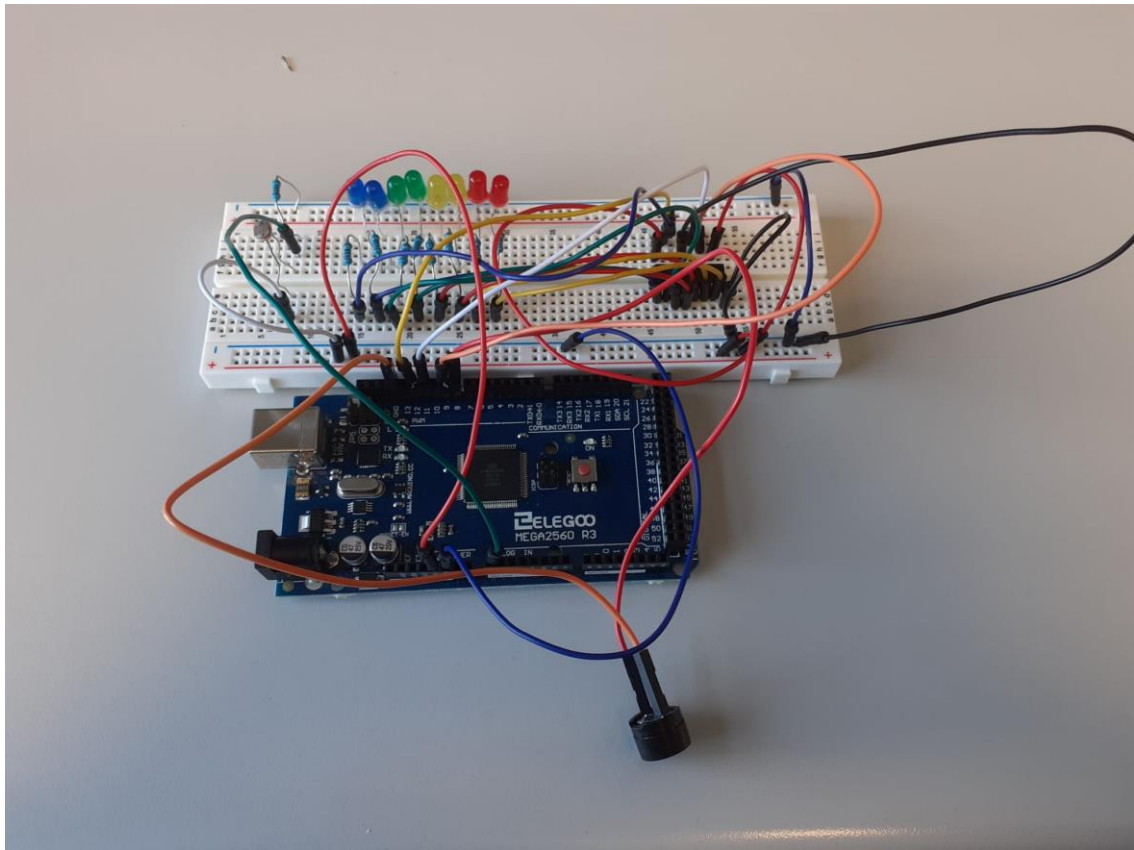
# Práctica 5 – Theremin con LEDs

## Descripción

Incorporamos al programa del Theremin un mapeo de valores adicional para que además de darnos un rango de sonido entre [50,4000] nos de un valor discreto entre [0,8], valor que utilizaremos para encender esa cantidad de LEDs conectados a la placa de Arduino.

Para incorporar la funcionalidad LED nos inspiramos en la lección 26 del StarterKit y utilizamos el chip 74hc595 IC que es el que nos permitirá distribuir la señal a los 8 leds a partir de la señal que produce nuestro programa.

## Montaje



## Código

```
int sensorValue;
int sensorLow = 1023;
int sensorHigh = 0;
const int ledPin = 13;
int lightPin = 0;
int latchPin = 11;
int clockPin = 9;
int dataPin = 12;

int leds = 0;

void setup() {
  // Make the LED pin an output and turn it on
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  // calibrate for the first five seconds after program runs
  while (millis() < 5000) {
    // record the maximum sensor value
    sensorValue = analogRead(A0);
    if (sensorValue > sensorHigh) {
      sensorHigh = sensorValue;
    }
    // record the minimum sensor value
    if (sensorValue < sensorLow) {
      sensorLow = sensorValue;
    }
  }
  // turn the LED off, signaling the end of the calibration period
  digitalWrite(ledPin, LOW);
}

void updateShiftRegister(){
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, LSBFIRST, leds);
  digitalWrite(latchPin, HIGH);
}

void loop() {
  //read the input from A0 and store it in a variable
  sensorValue = analogRead(A0);
  int numLEDSLit = map(sensorValue, sensorLow, sensorHigh, 0, 8);
  leds = (1 << numLEDSLit) - 1;
  // map the sensor values to a wide range of pitches
  int pitch = map(sensorValue, sensorLow, sensorHigh, 50, 4000);
  tone(8, pitch, 20);
  updateShiftRegister();
  delay(10);
}
```