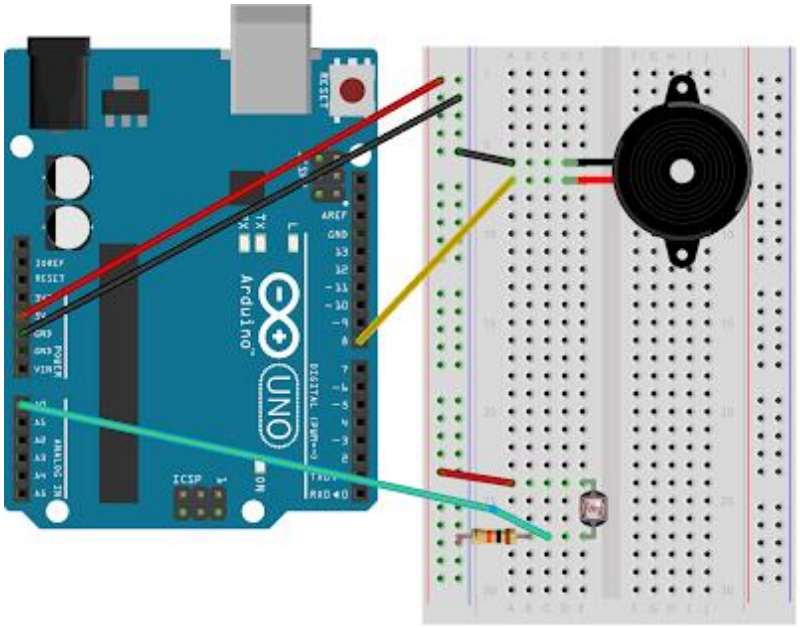
Pràctica-9: Brunzidor

UTILITZANT UNA FOTORESISTÈNCIA I UN BRUNZIDOR FARÀS UN THEREMIN BASAT EN LES VARIACIONS D'ONA DE LA LLUM



Programació:

```
int sensorValue;
int sensorLow = 1023;
int sensorHigh = 0;

const int ledPin = 13;

void setup(){
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);

  while (millis() < 5000) {
    sensorValue = analogRead(A0);
    if (sensorValue > sensorHigh){
      sensorHigh = sensorValue;
    }
    if (sensorValue < sensorLow) {
      sensorLow = sensorValue;
    }
  }
  digitalWrite(ledPin, LOW);
}

void loop() {
  sensorValue = analogRead(A0);

  int pitch = map(sensorValue, sensorLow,
sensorHigh, 50, 4000);

  tone (8, pitch, 20);

  delay(10);
}
```

Explicació:

— Es crea una variable per guardar el valor analogRead() de la fotoresistència. Després, s'estableixen les variables per els valors alts i els valors baixos (0-1023) per després compara aquests nombres amb els valors del sensor.

— Es crea la constant ledPin. El LED del pin 13 servirà d'indicador per el final del calibratge del sensor.

— En el setup() canviem el pinMode()del pin del LED com a sortida OUTPUT i encenem el LED.

— Per calibrar els valors màxims i mínims del sensor declarem un "mentre" while()per un bucle de 5 segons. Per la condició utilitzem la funció millis()que comprova l'hora actual i informa de quant temps l'Arduino ha estat funcionant des de que s'encén o es restableix.

— En el bucle llegirem el valor del sensor; si és menys que sensorLow (1023) o més que sensorHigh (0) s'actualitzarà la variable amb nous valors.

— Quan hagin passat 5 segons el bucle "mentre" acabarà, de manera que hem d'apagar el LED adjunt al pin 13.

— En el loop() llegirem el valor d'A0 i l'emmagatzemarem a sensorValue.

— Creem la variable pitch. El valor serà assignat des de sensorValue. Utilitzant sensorLow i sensorHigh com els límits dels valors entrants. Per els valors de partida a la sortida, prova 50 fins 4000 (defineixen el rang de freqüències que generarà Arduino).

— La funció tone()reprodueix un so. Té tres arguments: número de pin (8), freqüència a reproduir (pitch) i la duració del so (20 mili-segons).

— Finalment, posa un delay()de 10 mili-segons per donar més temps al so.