COMP6043 Physical Computing Lab 3

Note: if a task asks you to demonstrate your work to your lecturer, you must demonstrate and get your work signed off. Otherwise, no marks will be awarded for your work.

Once completed, upload your report to Canvas.

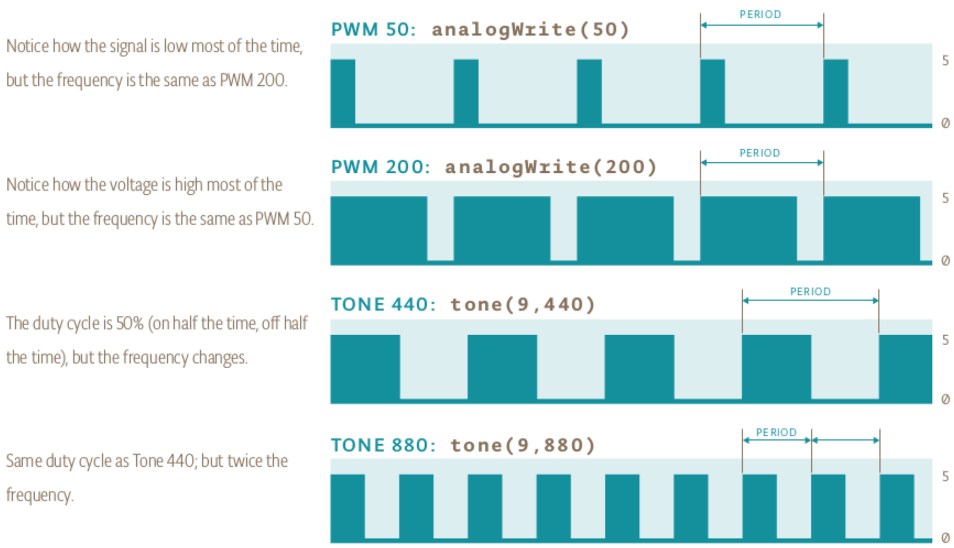
Pay particular attention when connecting the PCB to the Arduino power pins. It is very important that you connect with the correct polarity, i.e. 5 V on the Arduino to 5 V on the PCB, and GND on the Arduino to GND on the PCB. If you wire up with the incorrect polarity, you will burn out the on-board temperature sensor and damage the board.

Introduction

This lab is based on the "Light Theremin" project in the Arduino Projects Book (p.70). The basic idea is to vary the pitch of a piezo according to the amount of light incident on an LDR. By moving your hands over the LDR, you can vary its resistance and, hence, the voltage dropped across it (since it is connected in series with a fixed resistor to form a voltage divider circuit). Your code will read this voltage (with 10-bit resolution) and map it to a frequency to be played by the piezo. You will use the `tone()` function to sound the piezo. This function sends a pulse train with a specified frequency to a specified pin.

Note: `tone()` differs from `analogWrite()` (i.e. PWM) as follows:

- `tone()`: fixed duty cycle (50%), variable frequency.
- `analogWrite()`: variable duty cycle, fixed frequency.

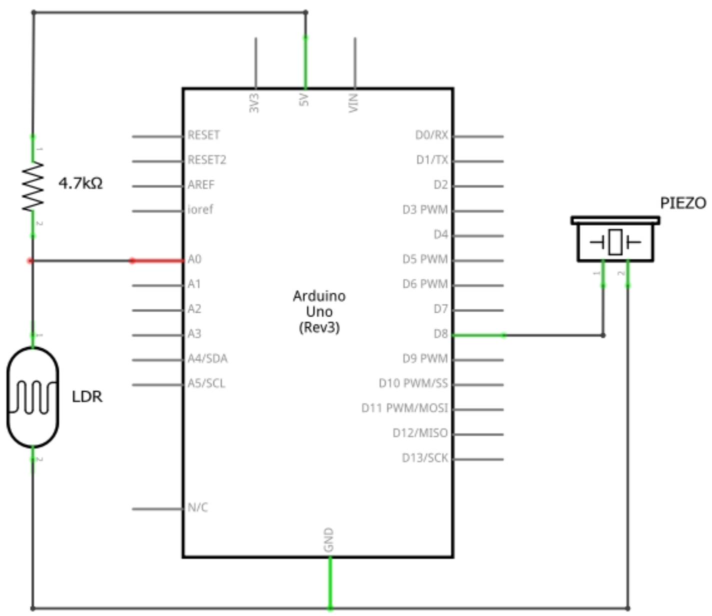


Also, unlike `analogWrite()`, it is possible to use the `tone()` function with any of the digital I/O pins.

In order to get a light-dependent sensor reading, the LDR is connected in series with a fixed resistor to form a voltage divider circuit. As the amount of incident light increases, the resistance of the LDR decreases and, hence also, the voltage dropped across it. This voltage will vary between 0 and 5 V. However, this voltage range corresponds to an LDR resistance swing from zero ohms (short circuit) to infinity (open circuit). In practice, the voltage range is limited by the brightness of the light and the value of the fixed resistor. In this lab, you will calibrate the sensor during the first five seconds of the program execution. There is an on-board LED connected to pin 13 that you will use to mark the beginning and end of the calibration period. When the program begins executing, the LED will be turned on. It will then turn off after five seconds, marking the end of the sensor calibration period. To calibrate the sensor, move your hands over the LDR, changing the amount of incident light. The closer you replicate the motions you expect to use while playing the instrument, the better the calibration will be.

Task 1 [60%]

Wire up your circuit as follows:



Notes:

- The 4.7 kΩ resistor is already connected in series with the LDR on the PCB.
- The PCB has a potentiometer connected to the piezo, which acts as a volume control. This is omitted from the above in order to simplify the schematic.
- To wire up your circuit, you simply need to connect the LDR interface on the PCB to pin A0 of the Arduino, and the piezo interface to digital I/O pin 8.

Open the Arduino IDE and enter the following code:

```
int sensorValue;
int sensorLow = 1023;
int sensorHigh = 0;
const int LED_PIN = 13;

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

  // calibrate for the first five seconds after program runs
  while (millis() < 5000) {
    sensorValue = analogRead(A0);
    if (sensorValue > sensorHigh) {
      sensorHigh = sensorValue;
    }
    if (sensorValue < sensorLow) {
      sensorLow = sensorValue;
    }
  }
  // turn the LED off, signaling the end of the calibration period
  digitalWrite(LED_PIN, LOW);
}

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

  // map the sensor values to a wide range of pitches
  int pitch = map(sensorValue, sensorLow, sensorHigh, 50, 4000);

  // play the tone for 20 ms on pin 8
  tone(8, pitch, 20);

  // wait for 10 ms to give sound time to play
  delay(10);
}
```

Upload your code to the Arduino and verify that your project works correctly.

Demonstrate your project to your lecturer once completed.

Task 2 [20%]

The voltage measured across the LDR is given by:

$$V = \frac{5R}{R + 4700},$$

where R is the resistance of the LDR. Rearranging for R gives:

$$R = \frac{4700V}{5 - V}.$$

Modify your code to display the LDR resistance on the serial monitor.

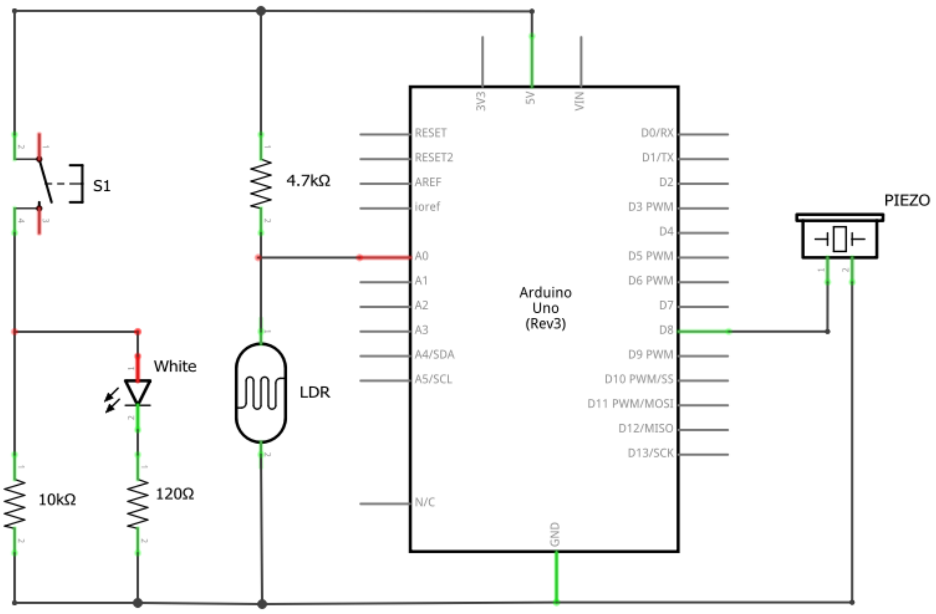
Note: you will need to convert from the sensor reading (0 – 1023) to voltage before evaluating the resistance.

Copy and paste the changes made to your code into your report.

Task 3 [10%]

Add one of the PCB push button switches in series with the white LED so that when the button is pressed, the LED illuminates the LDR.

In this case, your circuit will look as follows:



Note that the pull down resistors for both the switch and the LED are already in place on the PCB.

Demonstrate your project to your lecturer once completed.

Task 4 [10%]

Add the GRN2, AMB, and RED2 LEDs to your project such that:

- Pitch: 50 – 1000 → GRN2 off, AMB off, RED2 off.
- Pitch: 1001 – 2000 → GRN2 on, AMB off, RED2 off.
- Pitch: 2001 – 3000 → GRN2 on, AMB on, RED2 off.
- Pitch: 3001 – 4000 → GRN2 on, AMB on, RED2 on.

Copy and paste your code into your report.

Demonstrate your project to your lecturer once completed.