Week 3: Laser Tripwire

CSCI 3907: IoT using Raspberry Pi

Components:

- Laser Emitter
- Photoresistor
- PCF8591 Analog-To-Digital Converter (ADC)
- Active Buzzer

- We'll use the laser emitter & photoresistor to make a tripwire
  - Photoresistor used to measure how much light being received
    - More light → Resistance decreases → output voltage changes
- We'll power the laser continuously & check how much light from the laser the photoresistor is receiving
- Photoresistor outputs analog signal
  - Need to convert to digital
    - Use PCF8591 Analog-To-Digital Converter

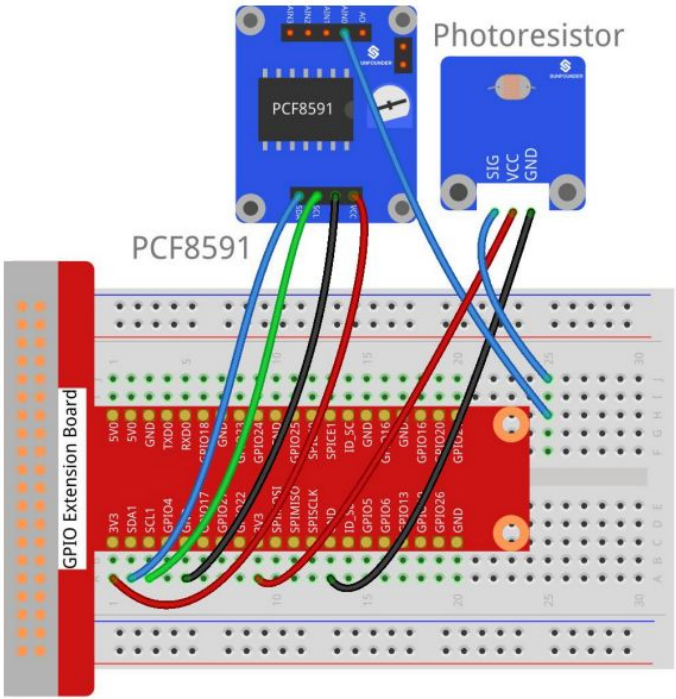
Laser

- Wire laser emitter & check you can turn it on before moving on to Photoresistor

- Laser emitter uses one GPIO pin
- Wire device to power (VCC to 3.3V) & one GPIO
- Write HIGH to GPIO to turn on laser
- Recall from week 1 slides:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(<Pin Number>, GPIO.OUT)
GPIO.output(<Pin Number>, GPIO.HIGH)
```

Photoresistor



- Wire photoresistor & PCF8591 ADC as shown in image
- We'll take readings from the ADC
  - PCF8591 uses i2c - fortunately Sunfounder provides a library for using the device

```
git clone https://github.com/sunfounder/SunFounder_SensorKit_for_RPi2.git
```

```
mv ./SunFounder_SensorKit_for_RPi2/Python/PCF8591.py ./
```

```
pip install smbus2 --user
```

Setup:

```
import PCF8591 as ADC

ADC.setup(0x48)
```

To read ADC:

```
input_val = ADC.read(0)
```

Setup:

- Orient laser so it points right at receiver on the Photoresistor
- Get measurements from photoresistor while laser on & off
  - This will give the measurement to check against to see if laser blocked

Assignment 2: Laser Tripwire

- Part A:
  - Continuously read input from Photoresistor
  - Check for drop below threshold indicating laser has been blocked
  - Print to terminal when laser is blocked
- Part B (optional):
  - Wire Active Buzzer
    - VCC to 3.3v, GND to GND, SIG to GPIO
  - Turn buzzer on when laser signal interrupted
    - Output HIGH to turn off
    - Output LOW to turn on

Note: buzzer is supposed to work with HIGH->off, LOW->on, but when we tried this assignment it stayed on either way; If this happens then use GPIO.setmode(OUT) for ‘on’ & GPIO.setmode(IN) for ‘off’ - this worked for us

Useful reference code:

Photoresistor:
https://github.com/sunfounder/SunFounder_SensorKit_for_RPi2/blob/master/Python/20_photoresistor.py

Laser:
https://github.com/sunfounder/SunFounder_SensorKit_for_RPi2/blob/master/Python/05_laser.py

Buzzer:

https://github.com/sunfounder/SunFounder_SensorKit_for_RPi2/blob/master/Python/10_active_buzzer.py