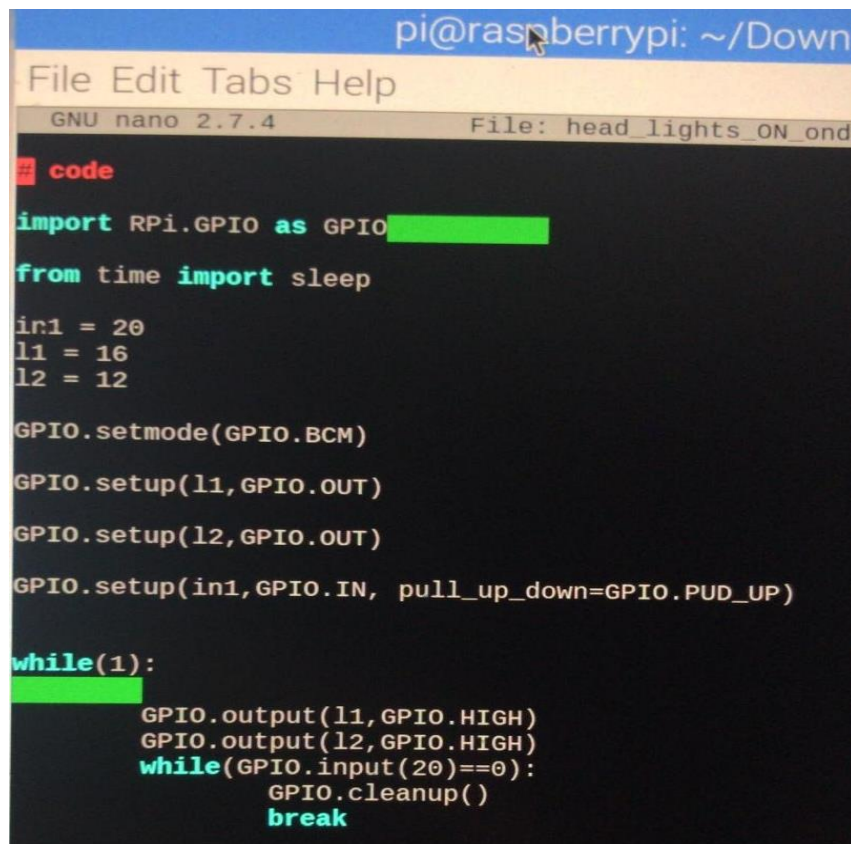


Connect RPI to Propeller

The purpose of this lab is to build a working automatic Light-Sensor system that will communicate with the RPI to let it know when to open lights and when to close it. So, the secondary microcontroller, in this case is Propeller, will determine when to send signal as HIGH or LOW to RPI, and RPI will handle the lights OFF or ON. This what I am going to work on this for this lab.

To make the connection, the Propeller will send a 3.3 voltage or 0 voltage to pin 1 on the Propeller, depending on the light sensor. There is a wire coming out from pin 1 to a GPIO pin on the RPI. This signal will be GPIO.IN, as input. Either High or Low, meaning 0 or 1, true or false. Using this logic, we are able to determine when to open and close the headlight.

- CODE

A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~/Downl'. The terminal shows the GNU nano 2.7.4 editor with a file named 'head_lights_ON_ond'. The code is a Python script that imports RPi.GPIO as GPIO and time module. It sets up three pins: in1 (20) as an input with pull-up, and l1 (16) and l2 (12) as outputs. The script enters a while(1) loop where it sets l1 and l2 to HIGH. It then enters a nested while loop that checks if GPIO.input(20) is 0. If it is, it calls GPIO.cleanup() and breaks the loop. The code is as follows:

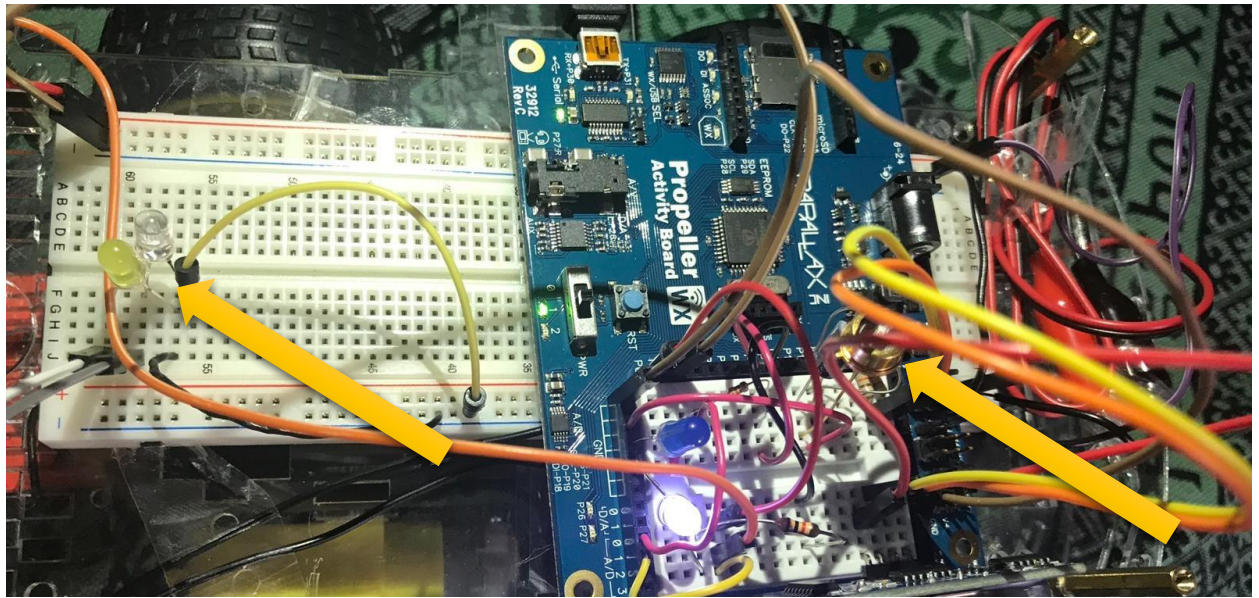
```
import RPi.GPIO as GPIO
from time import sleep

in1 = 20
l1 = 16
l2 = 12

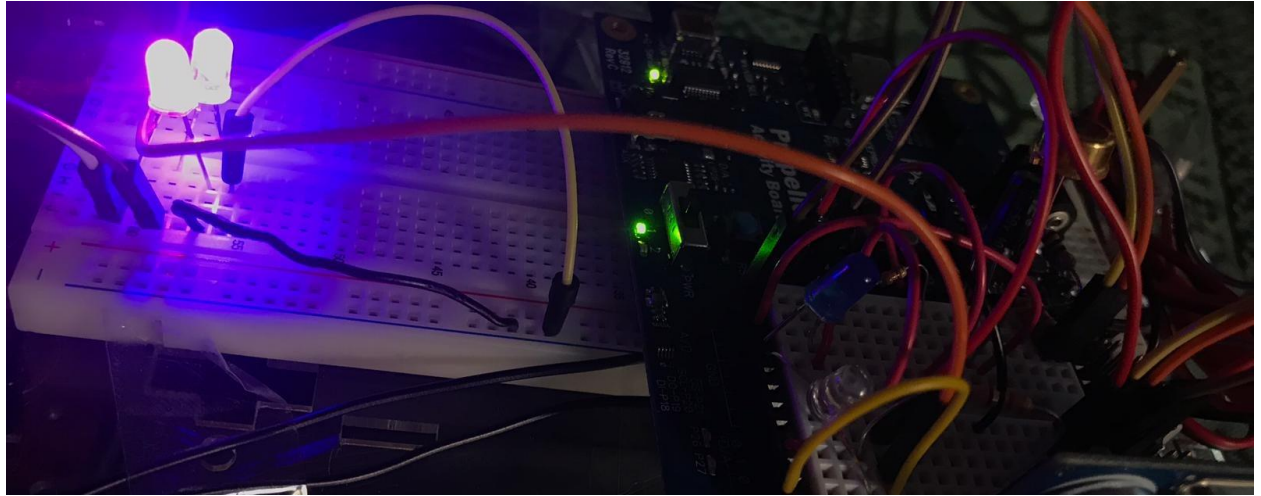
GPIO.setmode(GPIO.BCM)
GPIO.setup(l1, GPIO.OUT)
GPIO.setup(l2, GPIO.OUT)
GPIO.setup(in1, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while(1):
    GPIO.output(l1, GPIO.HIGH)
    GPIO.output(l2, GPIO.HIGH)
    while(GPIO.input(20)==0):
        GPIO.cleanup()
        break
```

- **Test – When No Dark – LED OFF :** We can see that when the sensor sense that there is enough light, the propeller will send a 0v through PIN1 to the RPI. RPI will not open the LEDs. Hence, headlights.



- **Test – When Dark – LED ON :** We can see that when the sensor sense that there is no enough light, the propeller will send a 3.3v through PIN1 to the RPI. RPI will open the LEDs. Hence, headlights.



As you saw in the pictures above. The RPI was able to receive the data coming from the second microcontroller and based on that data will decide whether to turn lights on or off. For lab 8, I am going to assemble the different parts together and add more LEDs to it increase the visibility . Also, I am going to assemble the rest of the car.