

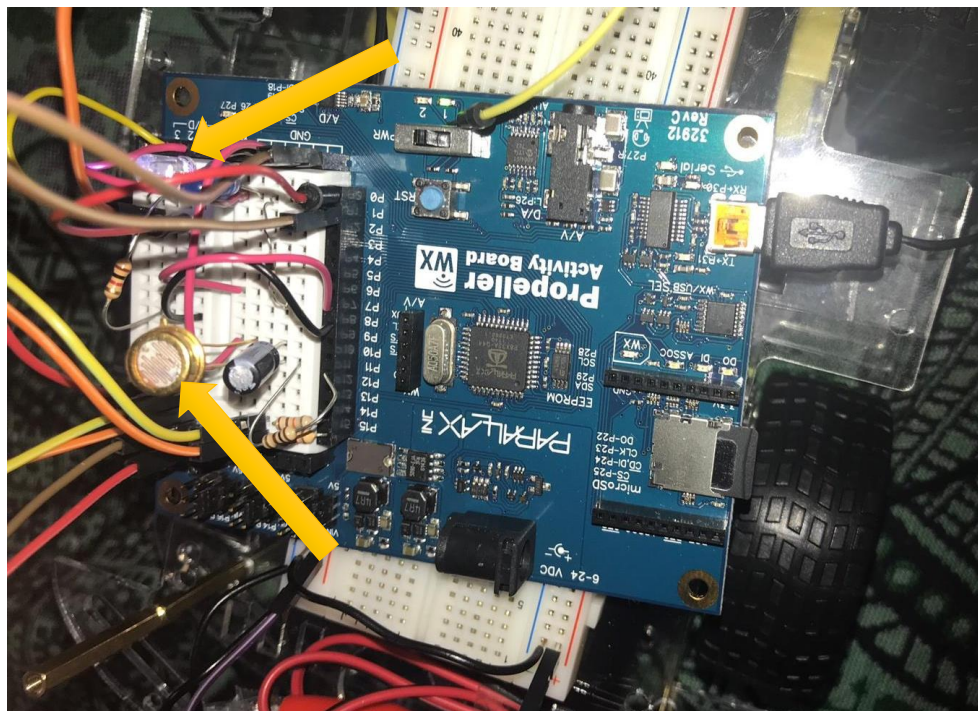
Head Lights of The Car Opens on Dark & build structure of Car (motors with RPI)

My project for this lab it to build a light sensor that turn the Head Light of the car when dark. The idea is to build the sensor using Propeller Activity Board that sense the environment that its driving in. When the Ligh-Sensor sense that the environment is dark, it will turn on a light on the board indicating that the environment is dark.

Section 1. lab 6 will be on the Propeller board only. I will build the sensor using

- Phototransistor
- capacitor
- resistor
- LED Light

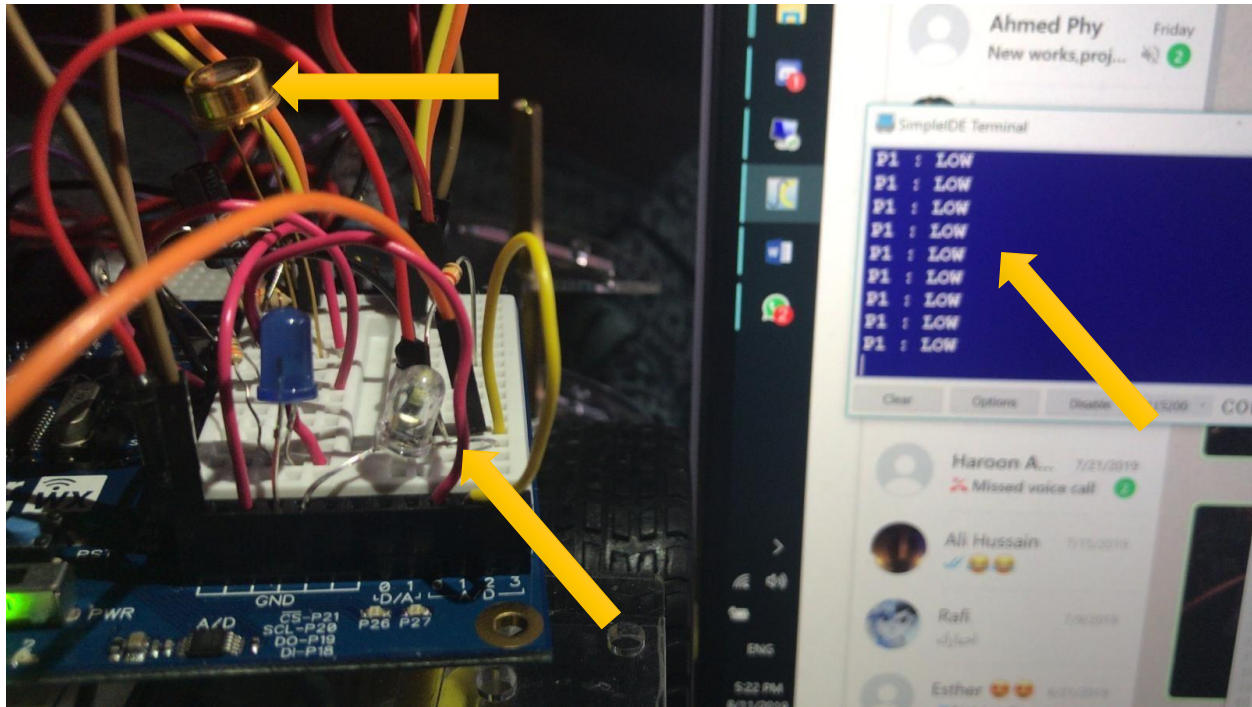
Logic Circuit : As you can see, when light sensor sense light is dark, it lights up LED.



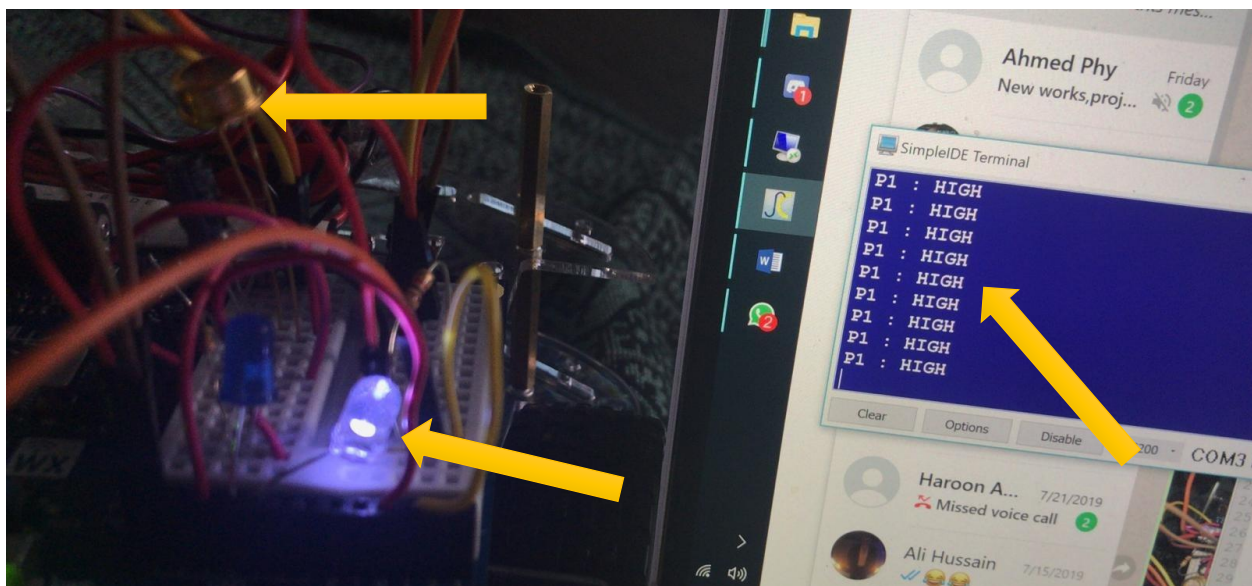
CODE:

```
1  /*
2   Sense Light.side
3
4   Display light sensor levels.
5
6   http://learn.parallax.com/propeller-c-si
7  */
8
9  #include "simpletools.h"
10
11 int main()
12 {
13     while(1)
14     {
15         high(11);
16         pause(1);
17         int t = rc_time(11, 1);
18         //print("t = %d\n", t);
19         //print("P1 : HIGH");
20
21         if (t < 100000)
22         {
23
24             dac_ctr(8, 0, 255);
25
26             dac_ctr(1, 1, 255);
27
28             print("P1 : HIGH\n");
29
30         }
31
32         if (t >= 100000)
33         {
34             dac_ctr(8, 0, 155);
35
36             dac_ctr(1, 1, 000);
37             print("P1 : LOW\n");
38
39         }
40         pause(200);
41     }
42 }|
43
44
```

Test – When No Dark – LED OFF : We can see that when the sensor sense that there is enough light, there is no reason to turn on the LED. Hence, headlights.



Test – When Dark – LED ON : We can see that when the sensor sense that there is NO enough light, the LED WILL TURN ON. Hence, headlights.



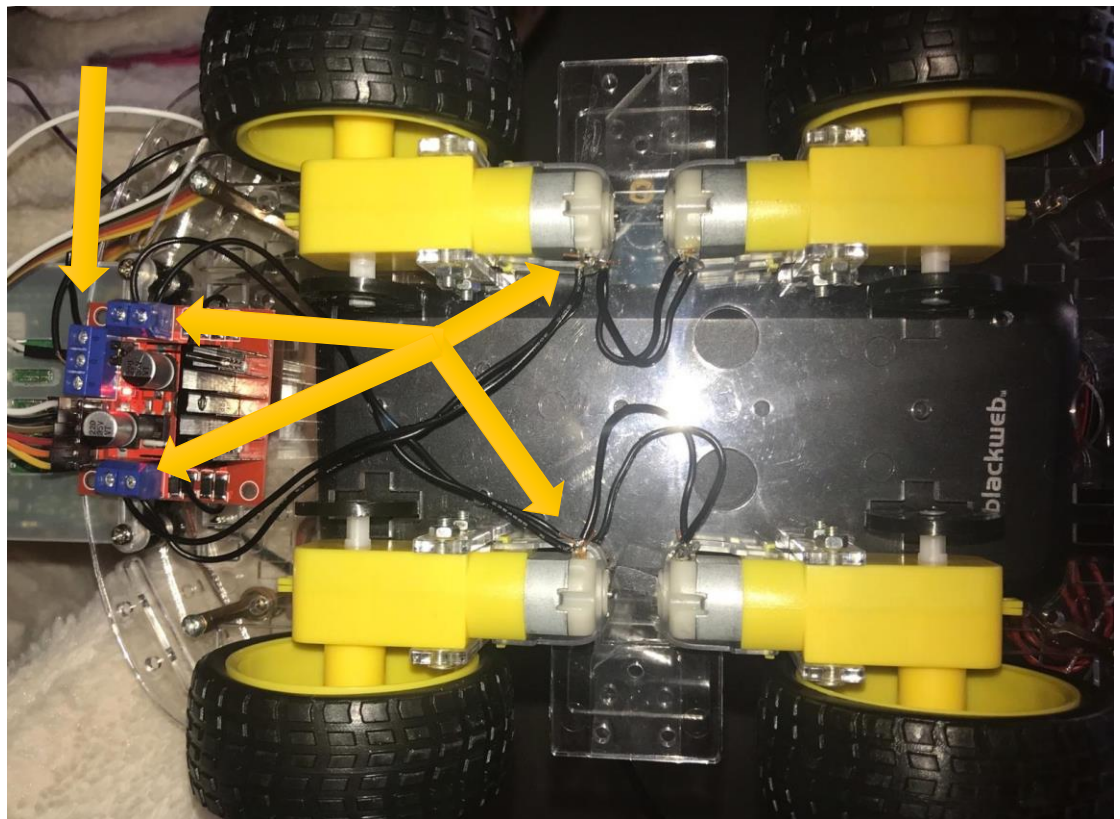
The purpose of this lab is to build a working automatic Light-Sensor system that will communicate with the RPI to let it know when to open lights and when to close it.

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 will be LAB 7.

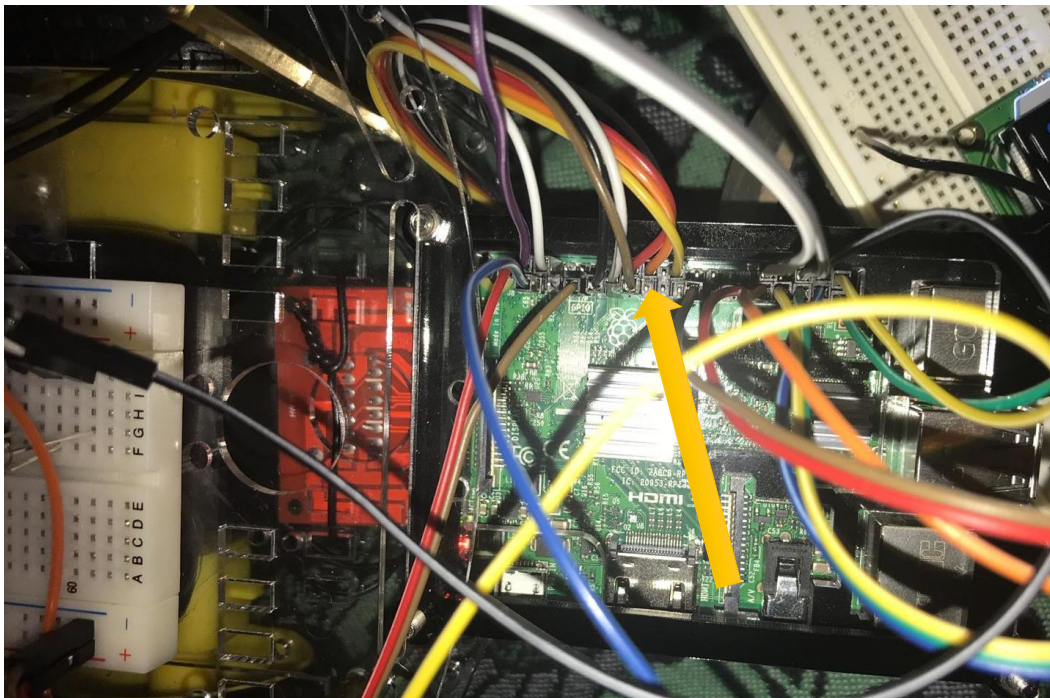
Section 2. Build Structure of the Car with RPI and L289 Bridge and 4 motors

In this section I will build a code using Python and RPI to control 4 motors. The idea is to design a system that runs the car foreword, backwards, turn left, turn right, speed 1, speed 2, speed 3.

- **Building the 4 motors and connecting it to the motor controller**



- As we can see in the above picture, each two motors are connected in reverse with each other and connected to one side of the Motor Controller (+, -). And same with the other side.
 - 6 pins coming out from it to the RPI GPIO pins as output to controller the motors.
 - 1 pin as 5v and 2 ground.
- **RPI Circuit :** The arrow is to show the pins coming from Motor controller to the RPI.



CODE to start and move the car

```

75 p2.start(27)
76
77 print("\n")
78
79 print("The default speed & direction of m
80
81 print("r-run s-stop f-forward b-backward
82
83 print("\n")
84
85
86
87 while(1):
88
89     x=raw_input()
90
91
92
93
94     if x=='r':
95
96         print("run")
97
98
99         if(temp1==1):
100
101             GPIO.output(in1,GPIO.HIGH)
102
103             GPIO.output(in2,GPIO.LOW)
104
105             GPIO.output(in3,GPIO.HIGH)
106
107             GPIO.output(in4,GPIO.LOW)
108
109             print("forward")
110
111             x='z'
112
113         else:
114
115             GPIO.output(in1,GPIO.LOW)
116
117             GPIO.output(in2,GPIO.HIGH)
118
119             GPIO.output(in3,GPIO.LOW)
120
121             GPIO.output(in4,GPIO.HIGH)
122
123             print("backward")
124
125             x='z'
126
127
128
129
130
131     elif x=='s':
132
133         print("stop")
134
135         GPIO.output(in1,GPIO.LOW)
136
137         GPIO.output(in2,GPIO.LOW)
138
139         GPIO.output(in3,GPIO.LOW)
140
141         GPIO.output(in4,GPIO.LOW)
142
143         x='z'
144

```

```

1
2 import RPi.GPIO as GPIO
3
4 from time import sleep
5
6
7 # first motor pins
8 in1 = 24
9
10 in2 = 23
11
12 en = 25
13
14 temp1=1
15
16 # second motor pins
17
18 in3 = 17
19
20 in4 = 22
21
22 en1 = 27
23
24 temp2=1
25
26 # pin from parellex to stop the car
27
28
29 in5 = 21
30
31
32
33 # first motor config
34 GPIO.setmode(GPIO.BCM)
35
36 GPIO.setup(in1,GPIO.OUT)
37
38 GPIO.setup(in2,GPIO.OUT)
39
40 GPIO.setup(en,GPIO.OUT)
41
42 GPIO.output(in1,GPIO.LOW)
43
44 GPIO.output(in2,GPIO.LOW)
45
46 p=GPIO.PWM(en,1000)
47
48
49 # second motor config
50
51 GPIO.setmode(GPIO.BCM)
52
53 GPIO.setup(in3,GPIO.OUT)
54
55 GPIO.setup(in4,GPIO.OUT)
56
57 GPIO.setup(en1,GPIO.OUT)
58
59 GPIO.output(in3,GPIO.LOW)
60
61 GPIO.output(in4,GPIO.LOW)
62
63 p2=GPIO.PWM(en1,1000)
64
65
66 # pin parellex to stop the car pin config
67
68 GPIO.setmode(GPIO.BCM)
69
70 GPIO.setup(in5,GPIO.IN,pull_up_down=GPIO.PUD_UP)

```

Test Code & Circuit.



VIDEO TEST.mp4

```
pi@raspberrypi: ~/Downloads
File Edit Tabs Help
head_lights_ON_ondark.py:13: RuntimeWarning: This channel is already in use, con
tinuing anyway. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(11,GPIO.OUT)
head_lights_ON_ondark.py:15: RuntimeWarning: This channel is already in use, con
tinuing anyway. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(12,GPIO.OUT)
Traceback (most recent call last):
  File "head_lights_ON_ondark.py", line 22, in <module>
    GPIO.output(11,GPIO.HIGH)
RuntimeError: Please set pin numbering mode using GPIO.setmode(GPIO.BOARD) or GP
IO.setmode(GPIO.BCM)
pi@raspberrypi:~/Downloads $ python main.py

The default speed & direction of motor is LOW & Forward.....
r-run s-stop f-forward b-backward l-low m-medium h-high e-exit

r
run
forward
h
high
```