

Rebecca's Code (Sensor- Light sensor)

Github code: <https://github.com/josecastroleon/max44009>

GNU nano 3.2	webtest2.py	Modified
--------------	-------------	----------

```
# light sensor code for ME30 Project 4
# foundation of code taken from a MAX44009 library
# https://github.com/ josecastroleon/max44009

import sys
import time
sys.path.append('./')
import max44009 # references max44009 script from library
status = ''
lumval = 0      # creating 'empty' variable that will be used to store lux #

from flask import Flask # importing flask
app = Flask(__name__)

@app.route('/')
def welcome():
    return 'Hello!' # prints Hello on the main page of URL
@app.route('/sensor/<string:status>') # adds /sensor/___ to URL

def sensorOn(status):
    if status == 'light': # /sensor/light to retrieve lux value
        max_sensor = max44009.MAX44009(1, 0x4a) # calls function
                                                # in max44009 code
                                                # i2c address is 4a
        max_sensor.configure(cont=0, manual=0, cdr=0, timer=0)
        time.sleep(2.0) # measures every 2 sec.
        lumval=max_sensor.luminosity() # stores lux in lumval
        lumvalround=round(lumval) # rounds lux to whole int
        return str(lumvalround) # returns rounded value
    else:
        return 'nothing interesting here' # if wrong URL won't call
```

Maia's Code (Actuator-Stepper Motor)

```
1  import RPi.GPIO as GPIO
2  import time
3  import sys
4
5  import requests
6
7  motorPin = (13,11,15,12)
8
9  i=0
10 positive=0
11 negative=0
12 y=0
13
14 sensorvalue = 0;
15 oldvalue = 0;      # starts at 0 and then at the end of
16 | | | | |         # the each loop sets current value to old value to compare next time
17
18 GPIO.setmode(GPIO.BOARD)
19 GPIO.setup(motorPin,GPIO.OUT)
20
21 print("The Iris is beginning to move....")
22
23 previousValue = None # setting first previous value as none so that it doesn't move the first time
24
25 try:
26     while(1):
27         GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.LOW,GPIO.LOW))
28         time.sleep(1)
29         value = requests.get('http://174.63.9.139:5000/sensor/light') # reads current sensor value
30         Value = value.text
31         #Value = int(value.text)
32         sensorValue = int(Value)
33         if previousValue is None:
34             previousValue = sensorValue #this is the initial condition
35             continue
36         difference = previousValue - sensorValue #this is the one from the get request
37         print(previousValue)
38         print(sensorValue)
39         print(difference)
40         previousValue = sensorValue
41         x = round(difference*0.4)
42         if x>0 and x<=400: #need range of values:
43             for y in range(x,0,-1): #function range(start,stop,step)
44                 if y in range(x,0,-1):
45                     if i==7:
46                         i=0
47                     else:
48                         i=i+1
49                     y=y+2
50                     negative=0
51                     positive=1
52
```

```

53         if i==0:
54             GPIO.output(motorPin, (GPIO.HIGH,GPIO.LOW,GPIO.LOW,GPIO.LOW))
55             time.sleep(0.2)
56         elif i==1:
57             GPIO.output(motorPin, (GPIO.HIGH,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
58             time.sleep(0.02)
59         elif i==2:
60             GPIO.output(motorPin, (GPIO.LOW,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
61             time.sleep(0.02)
62         elif i==3:
63             GPIO.output(motorPin, (GPIO.LOW,GPIO.HIGH,GPIO.HIGH,GPIO.LOW))
64             time.sleep(0.02)
65         elif i==4:
66             GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.HIGH,GPIO.LOW))
67             time.sleep(0.02)
68         elif i==5:
69             GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.HIGH,GPIO.HIGH))
70             time.sleep(0.02)
71         elif i==6:
72             GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.LOW,GPIO.HIGH))
73             time.sleep(0.02)
74         elif i==7:
75             GPIO.output(motorPin, (GPIO.HIGH,GPIO.LOW,GPIO.LOW,GPIO.HIGH))
76             time.sleep(0.02)
77         if i==7:
78             i=0
79             continue
80         i=i+1
81
82
83     elif x<0 and x>=-400:
84         x=x*-1
85         for y in range(x,0,-1):
86             if positive==1:
87                 if i==0:
88                     i=7
89                 else:
90                     i=i-1
91                 y=y+3
92                 positive=0
93             negative=1
94             if i==0:
95                 GPIO.output(motorPin, (GPIO.HIGH,GPIO.LOW,GPIO.LOW,GPIO.LOW))
96                 time.sleep(0.02)
97             elif i==1:
98                 GPIO.output(motorPin, (GPIO.HIGH,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
99                 time.sleep(0.02)
100            elif i==2:
101                GPIO.output(motorPin, (GPIO.LOW,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
102                time.sleep(0.02)
103            elif i==3:
104                GPIO.output(motorPin, (GPIO.LOW,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
105                time.sleep(0.02)
106            elif i==4:
107                GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.HIGH,GPIO.LOW))
108                time.sleep(0.02)
109            elif i==5:
110                GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.HIGH,GPIO.HIGH))
111                time.sleep(0.02)
112            elif i==6:
113                GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.LOW,GPIO.HIGH))
114                time.sleep(0.02)
115            elif i==7:
116                GPIO.output(motorPin, (GPIO.HIGH,GPIO.LOW,GPIO.LOW,GPIO.HIGH))
117                time.sleep(0.02)
118            if i==0:
119                i=7
120                continue
121            i=i-1
122

```

```
123 #clockwise closes the iris
124 #counterclockwise opens the iris
125
126
127
128
129 #press ctrl+c for keyboard interrupt
130 except KeyboardInterrupt:
131     print('The iris will shutdown')
132     GPIO.cleanup()
133     sys.exit(0)
```