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 librarv
# 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)

```
import RPi.GPIO as GPIO
import time
import sys

import requests

motorPin = (13,11,15,12)

i=0
positive=0
negative=0
y=0

sensorvalue = 0;
oldvalue = 0;       # starts at 0 and then at the end of
                    # the each loop sets current value to old value to compare next time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(motorPin,GPIO.OUT)

print("The Iris is beginning to move....")

previousValue = None   # setting first previous value as none so that it doesn't move the first time

try:
    while(1):
        GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.LOW,GPIO.LOW))
        time.sleep(1)
        value = requests.get('http://174.63.9.139:5000/sensor/light')     # reads current sensor value
        Value = value.text
        #Value = int(value.text)
        sensorValue = int(Value)
        if previousValue is None:
            previousValue = sensorValue     #this is the initial condition
            continue
        difference = previousValue - sensorValue    #this is the one from the get request
        print(previousValue)
        print(sensorValue)
        print(difference)
        previousValue = sensorValue
        x = round(difference*0.4)
        if x>0 and x<=400:    #need range of values:
            for y in range(x,0,-1): #function range(start,stop,step)
                if y in range(x,0,-1):
                    if i==7:
                        i=0
                    else:
                        i=i+1
                    y=y+2
                    negative=0
                positive=1

```

```
                if i==0:
                    GPIO.output(motorPin, (GPIO.HIGH,GPIO.LOW,GPIO.LOW,GPIO.LOW))
                    time.sleep(0.2)
                elif i==1:
                    GPIO.output(motorPin, (GPIO.HIGH,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
                    time.sleep(0.02)
                elif i==2:
                  GPIO.output(motorPin, (GPIO.LOW,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
                  time.sleep(0.02)
                elif i==3:
                  GPIO.output(motorPin, (GPIO.LOW,GPIO.HIGH,GPIO.HIGH,GPIO.LOW))
                  time.sleep(0.02)
                elif i==4:
                  GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.HIGH,GPIO.LOW))
                  time.sleep(0.02)
                elif i==5:
                  GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.HIGH,GPIO.HIGH))
                  time.sleep(0.02)
                elif i==6:
                  GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.LOW,GPIO.HIGH))
                  time.sleep(0.02)
                elif i==7:
                  GPIO.output(motorPin, (GPIO.HIGH,GPIO.LOW,GPIO.LOW,GPIO.HIGH))
                  time.sleep(0.02)
                if i==7:
                  i=0
                  continue
                i=i+1


        elif x<0 and x>=-400:
            x=x*-1
            for y in range(x,0,-1):
                if positive==1:
                    if i==0:
                      i=7
                    else:
                      i=i-1
                    y=y+3
                    positive=0
                negative=1
                if i==0:
                    GPIO.output(motorPin, (GPIO.HIGH,GPIO.LOW,GPIO.LOW,GPIO.LOW))
                    time.sleep(0.02)
                elif i==1:
                    GPIO.output(motorPin, (GPIO.HIGH,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
                    time.sleep(0.02)
                elif i==2:
                    GPIO.output(motorPin, (GPIO.LOW,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
                    time.sleep(0.02)
                elif i==3:
                    GPIO.output(motorPin, (GPIO.LOW,GPIO.HIGH,GPIO.LOW,GPIO.LOW))
                    time.sleep(0.02)
                elif i==4:
                    GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.HIGH.GPIO.LOW))
                    time.sleep(0.02)
                elif i==5:
                    GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.HIGH,GPIO.HIGH))
                    time.sleep(0.02)
                elif i==6:
                    GPIO.output(motorPin, (GPIO.LOW,GPIO.LOW,GPIO.LOW,GPIO.HIGH))
                    time.sleep(0.02)
                elif i==7:
                    GPIO.output(motorPin, (GPIO.HIGH,GPIO.LOW,GPIO.LOW,GPIO.HIGH))
                    time.sleep(0.02)
                if i==0:
                    i=7
                    continue
                i=i-1

```

```
#clockwise closes the iris
#counterclockwise opens the iris




#press ctrl+c for keyboard interrupt
except KeyboardInterrupt:
    print('The iris will shutdown')
    GPIO.cleanup
    sys.exit(0)
```