#StepperMotor
import pigpio
from PigpioStepperMotor import StepperMotor, fullStepSequence

#servo
pulse = None
gpioServo = 4
servoPos = None

#Grove Sunlight Sensor
import sys
import os
pulse = None
gpioServo = 4
servoPos = None
highVisible = 0

sys.path.append('./SDL_Pi_SI1145');
import time
import RPi.GPIO as GPIO

#set up GPIO using BCM numbering
GPIO.setmode(GPIO.BCM)

LED = 4

GPIO.setup(LED, GPIO.OUT, initial=0)

from datetime import datetime

from apscheduler.schedulers.background import BackgroundScheduler

import SDL_Pi_SI1145

sensor = SDL_Pi_SI1145.SDL_Pi_SI1145()

setup apscheduler

def tick():
print('Tick! The time is: %s' % datetime.now())

def killLogger():
scheduler.shutdown()
print "Scheduler Shutdown…."
exit()

def blinkLED(times,length):

```
for i in range(0, times):
    GPIO.output(LED, 1)
    time.sleep(length)
    GPIO.output(LED, 0)
    time.sleep(length)
```

def readSunLight():

```
    vis = sensor.readVisible()
    IR = sensor.readIR()
    UV = sensor.readUV()
    uvIndex = UV / 100.0
```

```
    print('SunLight Sensor read at time: %s' % datetime.now())
    print '        Vis:             ' + str(vis)
    print '        IR:              ' + str(IR)
    print '        UV Index:        ' + str(uvIndex)

    if uvIndex <= 3 :
        print "Warning:" + "Wear Sun Glass; Low UV"
    elif uvIndex > 3 and uvIndex <= 6 :
        print "Warning:" + "Take cover when avalible; Moderate UV"
    elif uvIndex > 6 and uvIndex >= 8 :
        print "Warning:" + "Apply SPF 30+ sunscreen, don't stay out more than 3 hours; High UV"
    elif uvIndex > 8 and uvIndex >= 11 :
        print "Warning:" + "Do not stay in the sun for too long; Very High UV"
    else :
        print "Warning:" + "Take all Percautions; Extreme UV"

returnValue = []
returnValue.append(vis)
returnValue.append(IR)
returnValue.append(uvIndex)
return returnValue
```

def ScanStepMotor():
pi = pigpio.pi()
motor = StepperMotor(pi, 6, 13, 19, 26, sequence = fullStepSequence, delayAfterStep = 0)
for i in range(256):
motor.doCounterclockwiseStep()
motor.doCounterclockwiseStep()
motor.doCounterclockwiseStep()
motor.doCounterclockwiseStep()
vis = sensor.readVisible()
IR = sensor.readIR()
UV = sensor.readUV()
uvIndex = UV / 100.0
if highVisible < UV:
servoPos = i
highVisible = UV
pass
print('SunLight Sensor read at time: %s' % datetime.now())
print ' Vis: ' + str(vis)
print ' IR: ' + str(IR)
print ' UV Index: ' + str(uvIndex)

```
    if uvIndex <= 3 :
        print "Warning:" + "Wear Sun Glass; Low UV"
    elif uvIndex > 3 and uvIndex <= 6 :
        print "Warning:" + "Take cover when avalible; Moderate UV"
    elif uvIndex > 6 and uvIndex >= 8 :
        print "Warning:" + "Apply SPF 30+ sunscreen, don't stay out more than 3 hours; High UV"
    elif uvIndex > 8 and uvIndex >= 11 :
        print "Warning:" + "Do not stay in the sun for too long; Very High UV"
    else :
        print "Warning:" + "Take all Percautions; Extreme UV"

returnValue = []
returnValue.append(vis)
returnValue.append(IR)
returnValue.append(uvIndex)
return returnValue
```

def ScanServo():
for x in range(21):
if x == 21:
servoPos = servoPos * 9
pi.set_servo_pulsewidth(gpioServo, servoPos)
break
else :
pulse = (x * 100)+500 #turn servo 100 pulse from 500-2500
pi.set_servo_pulsewidth(gpioServo, pulse)

print(servoPos)
time.sleep(0.4)
vis = sensor.readVisible()
IR = sensor.readIR()
UV = sensor.readUV()
uvIndex = UV / 100.0
if highVisible < uvIndex:
servoPos = x
highVisible = uvIndex
pass
print('SunLight Sensor read at time: %s' % datetime.now())
print ' Vis: ' + str(vis)
print ' IR: ' + str(IR)
print ' UV Index: ' + str(uvIndex)

```
        if uvIndex <= 3 :
            print "Warning:" + "Wear Sun Glass; Low UV"
        elif uvIndex > 3 and uvIndex <= 6 :
            print "Warning:" + "Take cover when avalible; Moderate UV"
        elif uvIndex > 6 and uvIndex >= 8 :
            print "Warning:" + "Apply SPF 30+ sunscreen, don't stay out more than 3 hours; High UV"
        elif uvIndex > 8 and uvIndex >= 11 :
            print "Warning:" + "Do not stay in the sun for too long; Very High UV"
        else :
            print "Warning:" + "Take all Percautions; Extreme UV"
        pass
```

print "-----------------"
print "SunIOT"
print ""
print "SwitchDoc Labs"
print "-----------------"
print ""

if **name__ == '__main**':

```
    scheduler = BackgroundScheduler()


# DEBUG Mode - because the functions run in a separate thread, debugging can be difficult inside the functions.
# we run the functions here to test them.
#tick()
#print readSunLight()



# prints out the date and time to console
    scheduler.add_job(tick, 'interval', seconds=60)
    # blink life light
scheduler.add_job(blinkLED, 'interval', seconds=5, args=[1,0.250])

# IOT Jobs are scheduled here (more coming next issue)
scheduler.add_job(ScanServo, 'interval', seconds=1)

    # start scheduler
scheduler.start()
print "-----------------"
print "Scheduled Jobs"
print "-----------------"
    scheduler.print_jobs()
print "-----------------"

    print('Press Ctrl+{0} to exit'.format('Break' if os.name == 'nt' else 'C'))

    try:
        # This is here to simulate application activity (which keeps the main thread alive).
        while True:
                time.sleep(2)
    except (KeyboardInterrupt, SystemExit):
        # Not strictly necessary if daemonic mode is enabled but should be done if possible
        scheduler.shutdown
```