Workshop Day 1: Raspberry Pi with GrovePi+, Sensors and Camera

GrovePi+ with RaspberryPi

GrovePi+ is an add-on board that brings Grove Sensors to the Raspberry Pi. With the provided white connectors, it simplifies the effort to interface sensors to Raspberry Pi .

GrovePi+ consists of following features:

- 7 digital Ports
- 3 analogue Ports
- 3 I2C ports

- 1 Serial port connect to GrovePi
- 1 Serial port connect to Raspberry Pi
- Grove header Vcc output Voltage: 5Vdc

Digital ports refer to the port that can detect an input or output a signal with 0 and 1 logic. Sensors that operate in this way are called digital component. An LED is an exampleof a digital output component. It can either be on or off, and there is no value in-between. We can think of the on and off states as being either 1 or 0. You can send a 1 to the LED to illuminate it on and a 0 to the LED to turn it off again.

However, not all components are digital. Some are called analogue components. Analogue components can send and receive values in-between 1 and 0. A motor is an example of an analogue output component. You can send it values between 1 and 0, which will control the speed of the motor. If you send the motor a 1 it will drive at full speed. If you send it 0.5 it will drive at half speed. Sending a 0 will stop the motor.

I2C is a very commonly used standard designed to allow sensors or devices to talk to another. The I2C bus allows multiple devices to be connected to your Raspberry Pi, where each of the sensor has a unique address.

The serial ports consists of two signals (a 'transmit' signal, Tx and a 'receive' signal Rx) and they are often known as UART. Serial ports are used for transmitting data. To connect to another serial device, you connect the 'transmit' of one to the 'receive' of the other, and vice versa. You will also need to connect the Ground pins of the two devices together.

The guidelines for setting up GrovePi+ with Raspberry Pi can be found in the appendix.

Following sections help you to familiar yourself with GrovePi+, sensors and camera.

1. LED light indication (Output)
2. Buzzer (Output)
3. Button Sensor
4. DHT Temperature & Humidity Sensor
5. Light Sensor
6. Sound Sensor
7. Ultrasonic Sensor
8. Camera

You can find more information for other sensors from https://wiki.seeedstudio.com/Grove/

1. LED Light Indication

1. Connect Grove-Red Led to D3 port of GrovePi+.
2. Open the grove_led_blink.py python script located on /Dexter/GrovePi/Software/Python
3. Change LED = 4 to LED = 3 as the LED is connected to digital port 3. Try to understand the code and how it works.

```
import time
from grovepi import *

# Connect the Grove LED to digital port D3 instead of D4
led = 3

pinMode(led,"OUTPUT")
time.sleep(1)
```

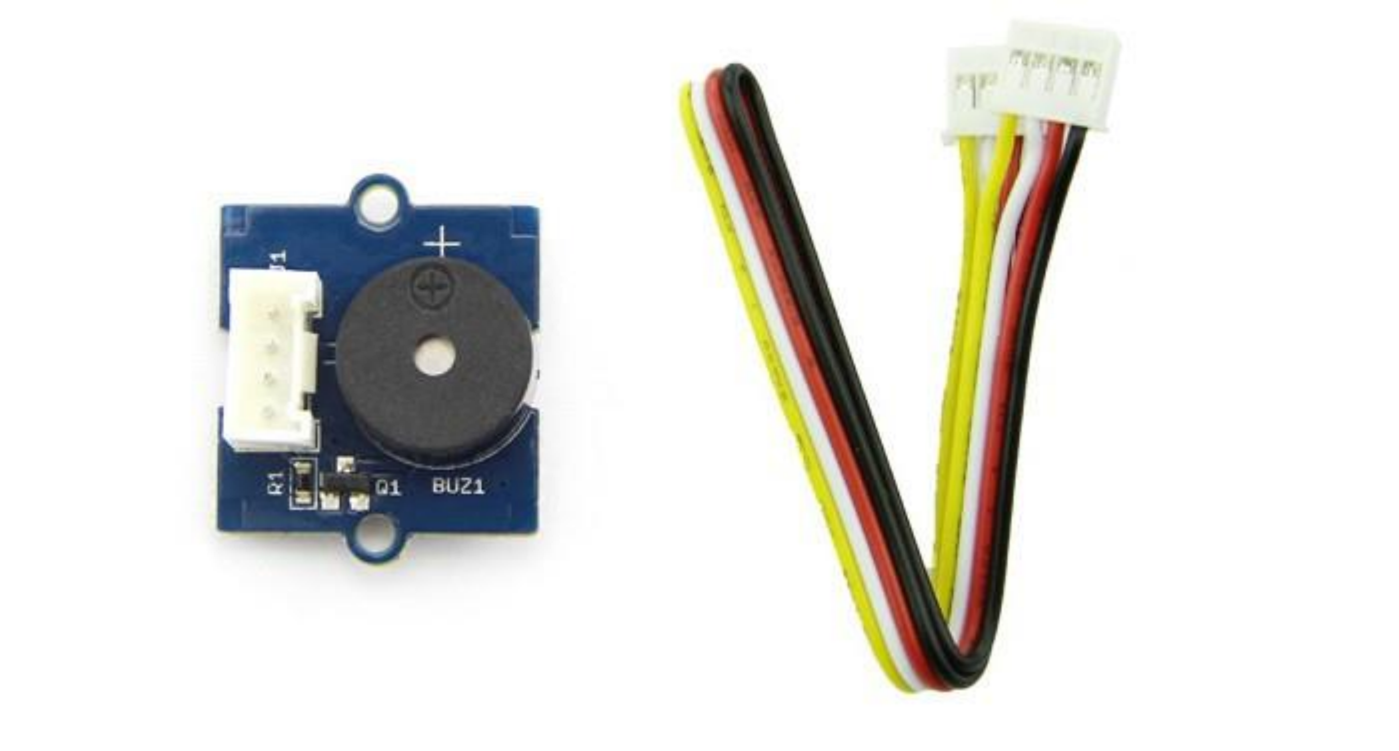
```
print ("This example will blink a Grove LED connected to the GrovePi+ on the port D{}.\nIf you're having trouble seeing the LED blink, be sure to check the LED connection and the port number.\nYou may also try reversing the direction of the LED on the sensor.".format(led))
print (" ")
print ("Connect the LED to the D{} port !".format(led))

while True:
    try:
        #Blink the LED
        digitalWrite(led,1)     # Send HIGH to switch on LED
        print ("LED ON!")
        time.sleep(1)

        digitalWrite(led,0)     # Send LOW to switch off LED
        print ("LED OFF!")
        time.sleep(1)

    except KeyboardInterrupt: # Turn LED off before stopping
        digitalWrite(led,0)
        break
    except IOError:         # Print "Error" if communication error encountered
        print ("Error")
```

4. Execute following commands and you will see the LED on and off.

```
cd ~/Dexter/GrovePi/Software/Python
```

```
python3 grove_led_blink.py
```

2. Buzzer



1. Connect Grove-Buzzer to D8 port of GrovePi+.
2. Try to understand the code and how it works.

```
import time
import grovepi

# Connect the Grove Buzzer to digital port D8
# SIG,NC,VCC,GND
buzzer = 8

grovepi.pinMode(buzzer,"OUTPUT")
```

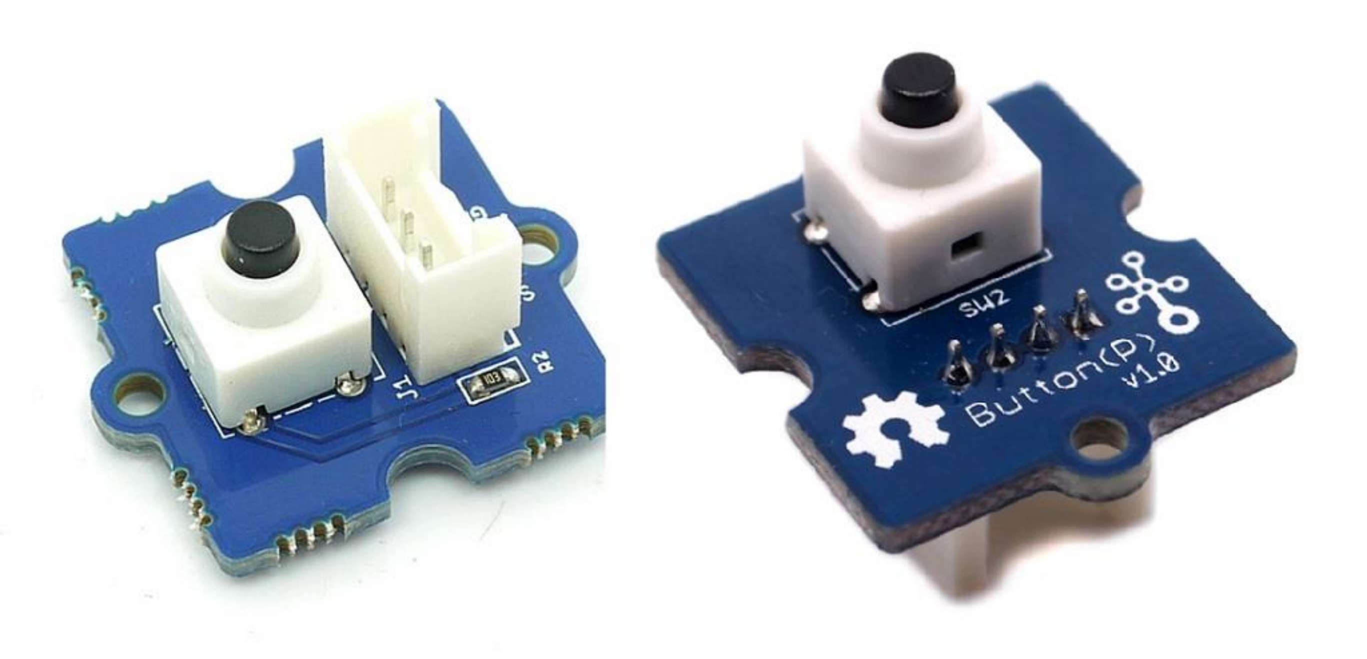
```
while True:
    try:
        # Buzz for 1 second
        grovepi.digitalWrite(buzzer,1)
        print ('start')
        time.sleep(1)

        # Stop buzzing for 1 second and repeat
        grovepi.digitalWrite(buzzer,0)
        print ('stop')
        time.sleep(1)

    except KeyboardInterrupt:
        grovepi.digitalWrite(buzzer,0)
        break
    except IOError:
        print ("Error")
```

3. Execute following commands and you will hear the buzzer on and off.

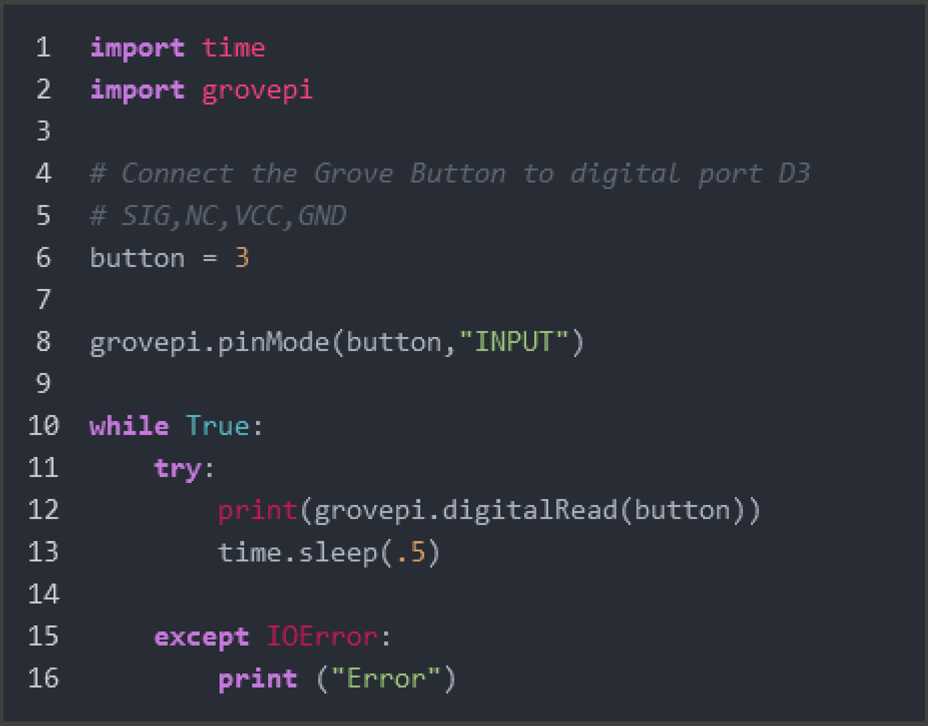
```
cd ~/Dexter/GrovePi/Software/Python
```

```
python3 grove_buzzer.py
```

3. Button Sensor



1. Connect Grove Button to D3 port of GrovePi+.
2. Open the grove_button.py python script located on /Dexter/GrovePi/Software/Python
3. Ensure button = 3 as the button is connected to digital port 3. Try to understand the code and how it works.



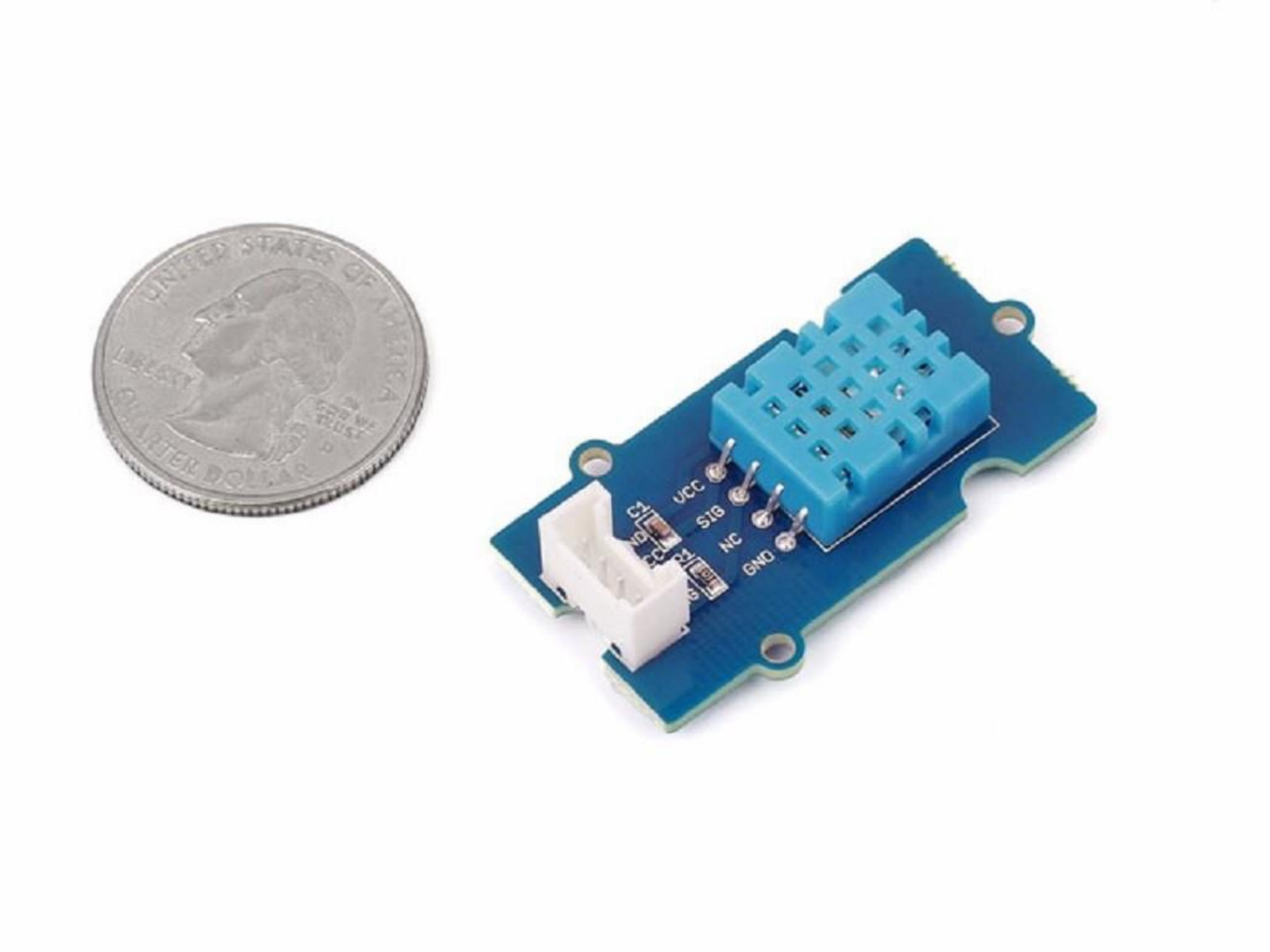
```python
import time
import grovepi

# Connect the Grove Button to digital port D3
# SIG,NC,VCC,GND
button = 3

grovepi.pinMode(button,"INPUT")

while True:
    try:
        print(grovepi.digitalRead(button))
        time.sleep(.5)

    except IOError:
        print ("Error")
```

4. Execute following command; press/release the button to see what happens.

   ```
   cd ~/Dexter/GrovePi/Software/Python
   ```

   ```
   python3 grove_button.py
   ```

4. Temperature & Humidity Sensor



1. Connect the Temperature & Humidity sensor to D4 port of GrovePi+.
2. Try to understand the code and how it works.

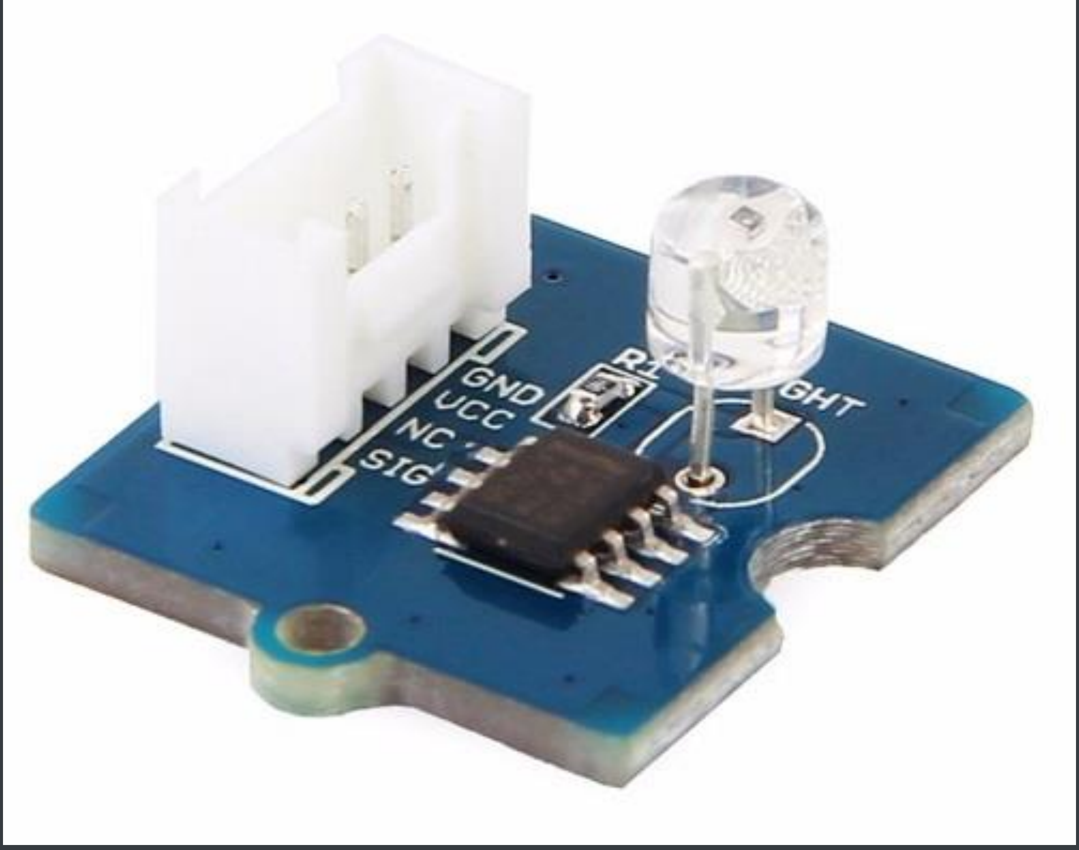
```
import grovepi
import math

# Connect the Grove Temperature & Humidity Sensor Pro to digital port D4
# This example uses the blue colored sensor.
# SIG,NC,VCC,GND
sensor = 4  # The Sensor goes on digital port 4.

# temp_humidity_sensor_type
# Grove Base Kit comes with the blue sensor.
blue = 0    # The Blue  colored  sensor.
white = 1   # The White colored sensor.

while True:
    try:
        # This example uses the blue colored sensor.
        # The first parameter is the port, the second parameter is the type of sensor.
        [temp,humidity] = grovepi.dht(sensor,blue)
        if math.isnan(temp) == False and math.isnan(humidity) == False:
            print("temp = %.02f C humidity =%.02f%%"%(temp, humidity))

    except IOError:
        print ("Error")
```

3. Execute following commands and you will see the temperature and humidity value printed on the terminal.

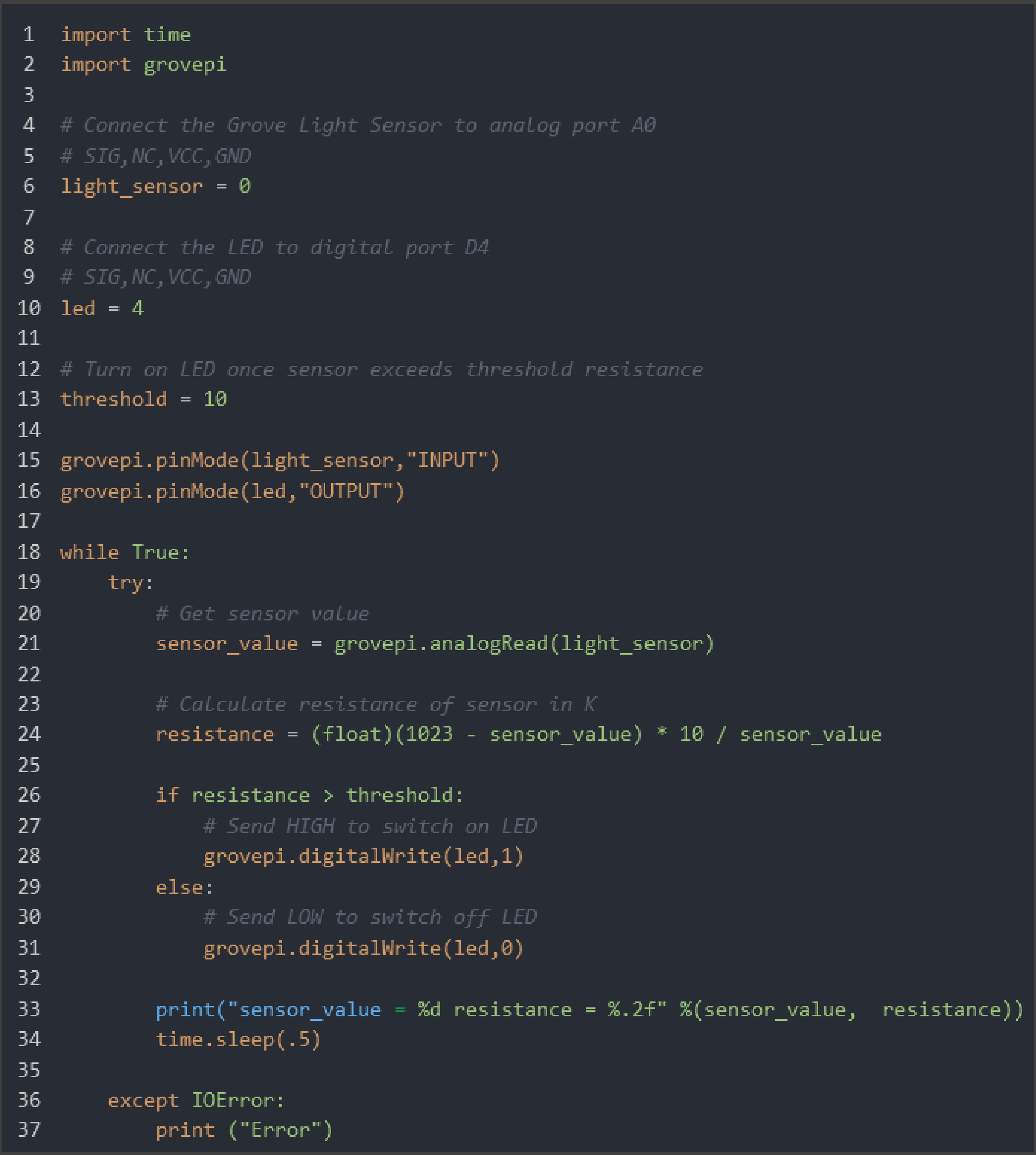
```
cd ~/Dexter/GrovePi/Software/Python
```

```
python3 grove_dht_pro.py
```

5. Light Sensor



1. Connect Grove Light sensor to A0 port of GrovePi+ and connect Grove LED to D4.
2. Open the grove_light_sensor.py python script located on /Dexter/GrovePi/Software/Python
3. Try to understand the code and how it works



```python
import time
import grovepi

# Connect the Grove Light Sensor to analog port A0
# SIG,NC,VCC,GND
light_sensor = 0

# Connect the LED to digital port D4
# SIG,NC,VCC,GND
led = 4

# Turn on LED once sensor exceeds threshold resistance
threshold = 10

grovepi.pinMode(light_sensor,"INPUT")
grovepi.pinMode(led,"OUTPUT")

while True:
    try:
        # Get sensor value
        sensor_value = grovepi.analogRead(light_sensor)

        # Calculate resistance of sensor in K
        resistance = (float)(1023 - sensor_value) * 10 / sensor_value

        if resistance > threshold:
            # Send HIGH to switch on LED
            grovepi.digitalWrite(led,1)
        else:
            # Send LOW to switch off LED
            grovepi.digitalWrite(led,0)

        print("sensor_value = %d resistance = %.2f" %(sensor_value,  resistance))
        time.sleep(.5)

    except IOError:
        print ("Error")
```

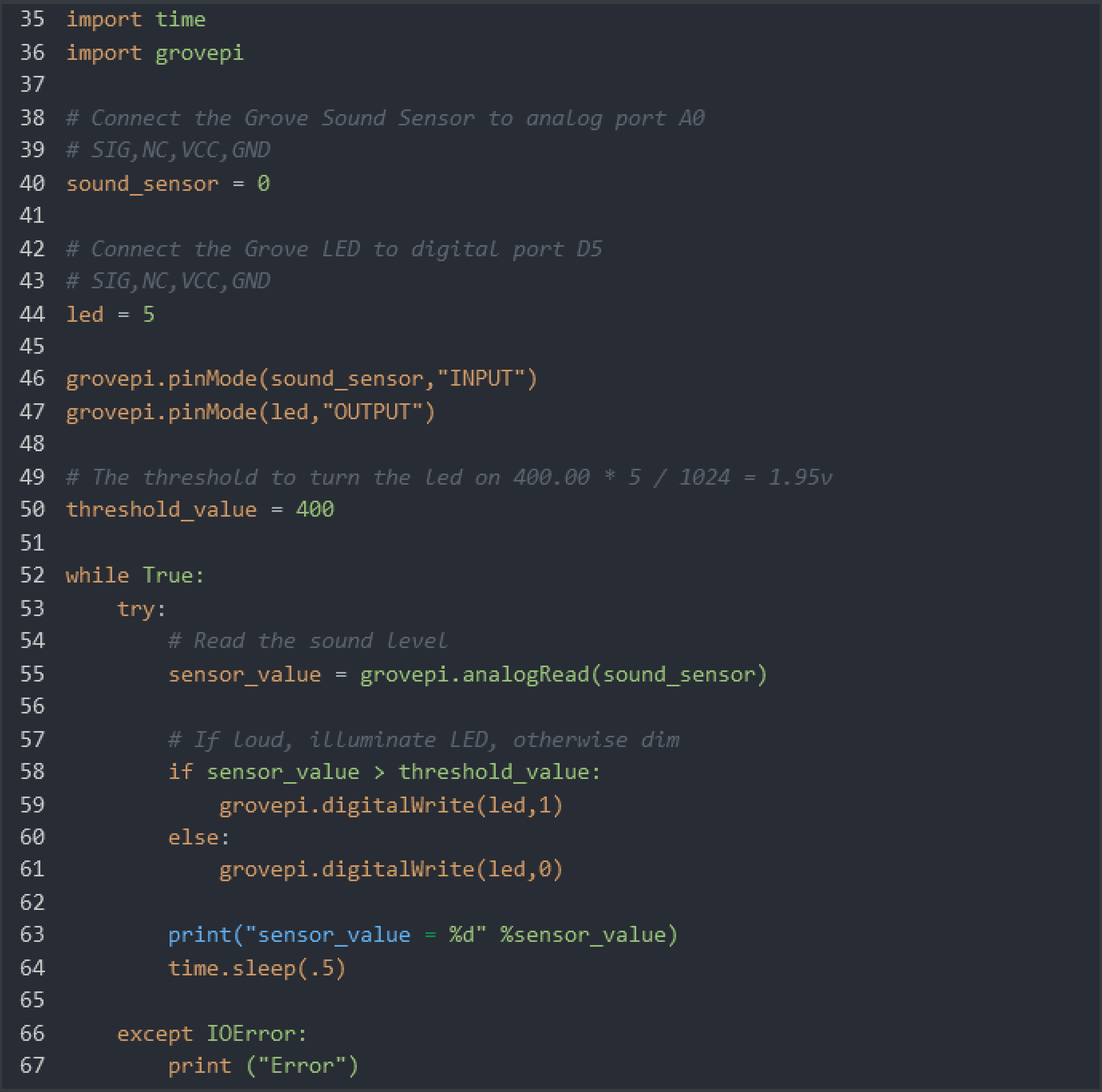
4. Execute following commands to obtain brightness values; cover (or shine bright light on) the light sensor and see what happens.

   ```
   cd ~/Dexter/GrovePi/Software/Python
   ```

   ```
   python3 grove_light_sensor.py
   ```

6. Sound Sensor

1. Connect Grove Sound to A0 port of GrovePi+ and connect Grove LED to D5.
2. Open the grove_sound_sensor.py python script located on /Dexter/GrovePi/Software/Python
3. Try to understand the code and how it works



```
import time
import grovepi

# Connect the Grove Sound Sensor to analog port A0
# SIG,NC,VCC,GND
sound_sensor = 0

# Connect the Grove LED to digital port D5
# SIG,NC,VCC,GND
led = 5

grovepi.pinMode(sound_sensor,"INPUT")
grovepi.pinMode(led,"OUTPUT")

# The threshold to turn the led on 400.00 * 5 / 1024 = 1.95v
threshold_value = 400

while True:
    try:
        # Read the sound level
        sensor_value = grovepi.analogRead(sound_sensor)

        # If loud, illuminate LED, otherwise dim
        if sensor_value > threshold_value:
            grovepi.digitalWrite(led,1)
        else:
            grovepi.digitalWrite(led,0)

        print("sensor_value = %d" %sensor_value)
        time.sleep(.5)

    except IOError:
        print ("Error")
```

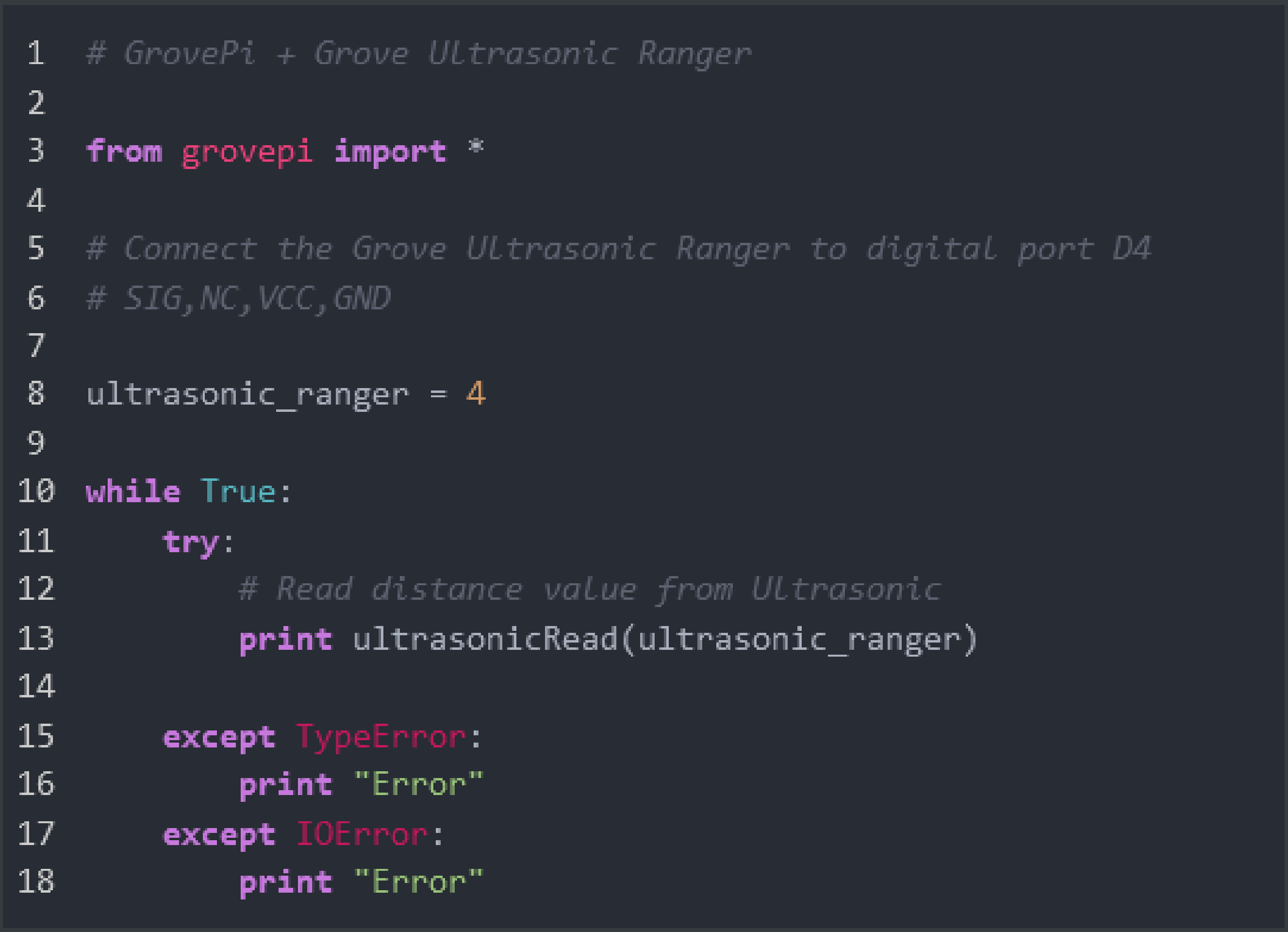
4. Execute following commands to obtain sound values; use your nail, and lightly and repeatedly knock it against the black area of the sound sensor; see what happens.

   ```
   cd ~/Dexter/GrovePi/Software/Python
   ```

   ```
   python3 grove_sound_sensor.py
   ```

7. Ultrasonic Sensor

1. Connect Grove Ultrasonic Sensor to D4 port of GrovePi+.
2. Open the grove_ultrasonic.py python script located on /Dexter/GrovePi/Software/Python
3. Try to understand the code and how it works



```python
# GrovePi + Grove Ultrasonic Ranger

from grovepi import *

# Connect the Grove Ultrasonic Ranger to digital port D4
# SIG,NC,VCC,GND

ultrasonic_ranger = 4

while True:
    try:
        # Read distance value from Ultrasonic
        print ultrasonicRead(ultrasonic_ranger)

    except TypeError:
        print "Error"
    except IOError:
        print "Error"
```

4. Execute following commands to obtain distance values; put a solid object (e.g. your hand) in front of the ultrasonic sensor and see what happens.

```
cd ~/Dexter/GrovePi/Software/Python
```

```
python3 grove_ultrasonic.py
```

8. Camera

1. Connect the camera to Raspberry Pi.
2. Try to understand the code and how it works.

```
from picamera import PiCamera
from time import sleep
```

```
camera = PiCamera()

# The camera preview only works when a monitor is connected to your
Raspberry Pi. If you are using remote access (such as SSH or VNC),
you won’t able to see the camera preview.
camera.start_preview()

# If your preview is upside-down, you can rotate it by 180 degrees
with the following code:
camera.rotation = 180

sleep(5)
camera.stop_preview()

# To save the captured image
camera.capture('/home/pi/Desktop/image.jpg')
```

3. Save above script to desktop as camera.py
4. Execute following commands and you will see the captured image on your desktop.

```
cd ~/Desktop
```

```
sudo python3 camera.py
```

Project 1: **Building a Smart Store System**

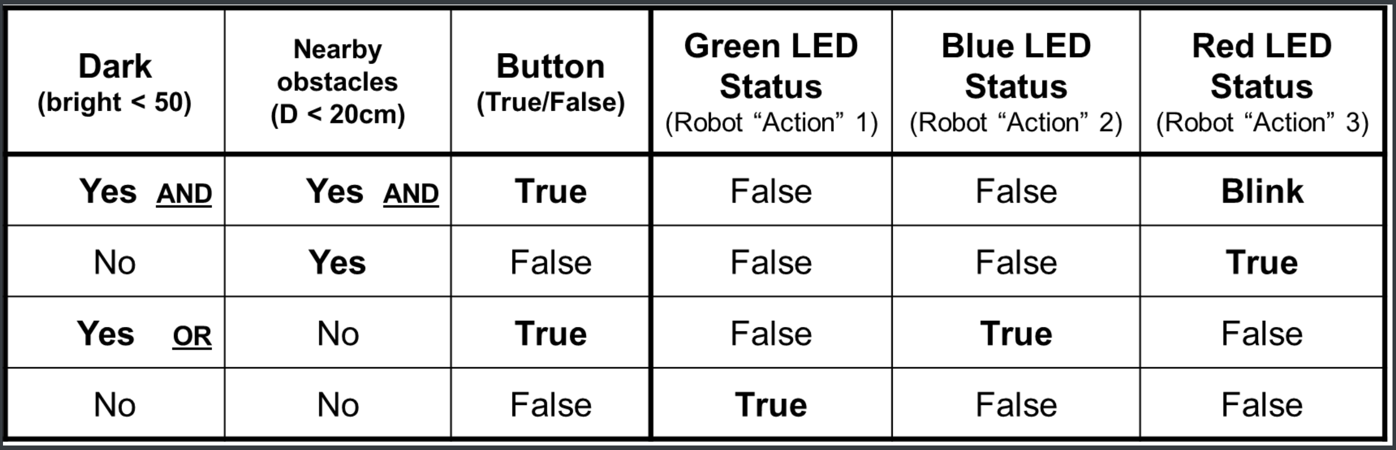
Musk is a store keeper and he was tasked to manually record down the store temperature and humidity every half an hour in order to preserve the quality of the goods storing inside. The temperature of the store must be kept below **30°C** and humidity below **70%**. There are currently no **indicators/alerts** in place to alert Musk. Hence, Musk has to manually and regularly monitor the temperature and humidity levels, and in the event if the **temperature or humidity** becomes too high (i.e. over the given threshold levels), Musk has to call the facilities department to activate/repair the air conditioner. Once the temperature and humidity are back to normal, Musk is required to take **a photo of the store** for record purposes.

The current process seems to be too tedious for Musk. You are to help him by **creating a system using raspberry pi, GrovePi+, any available sensors, actuators and camera to improve the efficiency.**

You are required to submit the python code(s) to Luminus.

Project 2: **Building a Smart "Robotic" System**

Sensors help robots to collect data from the environment, enabling them to react according to certain changes in the environment. You are to **create a robotic system (using raspberry pi, GrovePi+, selected sensors and actuators)** based on the condition table below:



Dark **(bright < 50)**	**Nearby obstacles** **(D < 20cm)**	**Button** **(True/False)**	**Green LED Status** (Robot "Action" 1)	**Blue LED Status** (Robot "Action" 2)	**Red LED Status** (Robot "Action" 3)
Yes **AND**	**Yes** **AND**	**True**	False	False	**Blink**
No	**Yes**	False	False	False	**True**
Yes **OR**	No	**True**	False	**True**	False
No	No	False	**True**	False	False

Note that we will not be using real robots; using LED to represent different robot actions

You are required to submit the python code(s) to Luminus.

Appendix: Guidelines for Setting up GrovePi+ with Raspberry Pi

1. Clone the Github repository at an appropriate location

```
git clone https://github.com/DexterInd/GrovePi.git
```

2. In the command line, type the following commands:

```
Curl -kL dexterindustries.com/update_grovepi | bash
```

3. Restart the Raspberry Pi, by typing in the following commands:

```
sudo reboot
```

4. After Raspberry Pi reboots, the GrovePi is ready to use.