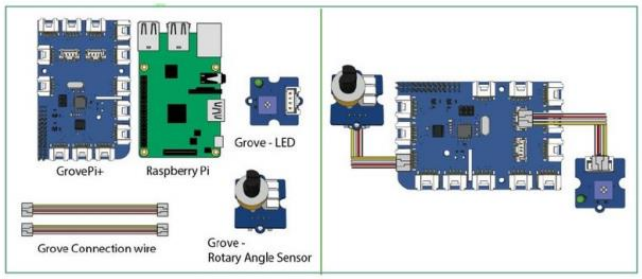
Objective: To learn how to control LED light intensity using an angle sensor.

Required Setup: Connect GrovePi+ board to RPi and have all GrovePi+ libraries installed.

Parts:

- RPi 3 B
- GrovePi+ board
- 3-4 Grove connection wires
- 3 x Grove LED modules (red, blue, green)
- 1 x Grove Angle Rotary Sensor (potentiometer)



Overview of the GrovePi and RPi Communication

The GrovePi is a shield that connects to the RaspberryPi (RPi) via the GPIO ports. Communication between the two boards take place via the I^2C interface (Inter-Integrated Circuit, pronounced as I-two-C or I-I-C) as introduced in Activity 1. The Grove modules (e.g. LEDs, sensors, etc.) connect to the GrovePi (blue board) using a universal 4-poin connector cable. The GrovePi board thus enables the RPi to access some of the Grove sensors directly (e.g. sending/receiving instructions).

There are signal modes in which these modules can operate in when connected to the ATMEGA328 microcontroller residing on the GrovePi: (a) analog and (b) digital. The microcontroller is used to filter and interpret data that moves between the GrovePi modules and the RPi (e.g. sending, receiving, executing commands received from the RPi). When you write a Python program to turn on a GrovePi LED module, for example, the Python script residing on the RPi sends commands via a GrovePi programming interface (e.g. API) and executes commands to the GrovePi via the ATMEAGA328 microcontroller which then execute these commands on the modules.

The RPi contains communication buses which enable the connection between the sensors and the GrovePi ports. There are two bus port types: (a) an I2C Bus and (2) a Serial Bus as shown in Figure 1.

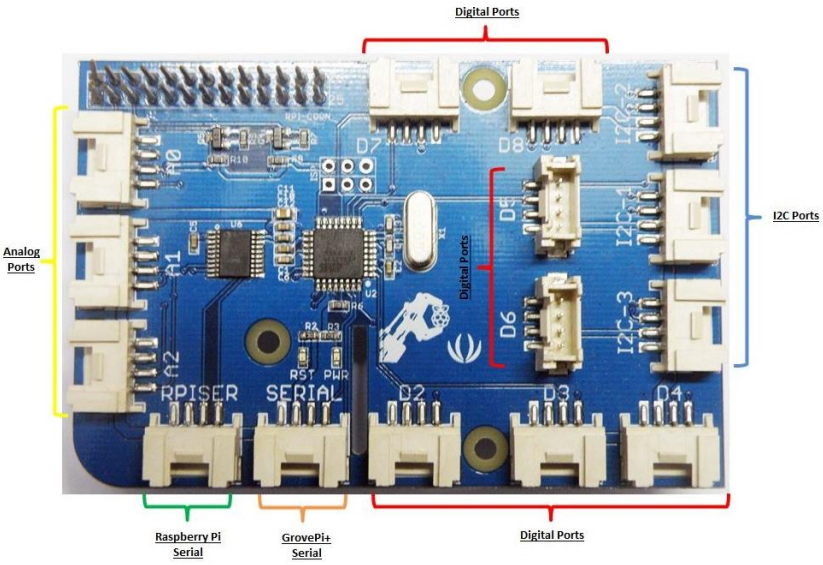


Figure 1: GrovePi Port Layout Diagram

Source: https://www.dexterindustries.com/GrovePi/engineering/port-description/

The GrovePi's ATMEGA328 microcontroller contains an on-board 6 channel Analog-to-Digital (AD) converter with each having 10-bit resolution returning values from 0 – 1023. Analog ports are usually used for reading analog modules or sensors but may be used also for general purpose I/O (same as digital pins 0-13) as shown in the diagram below:

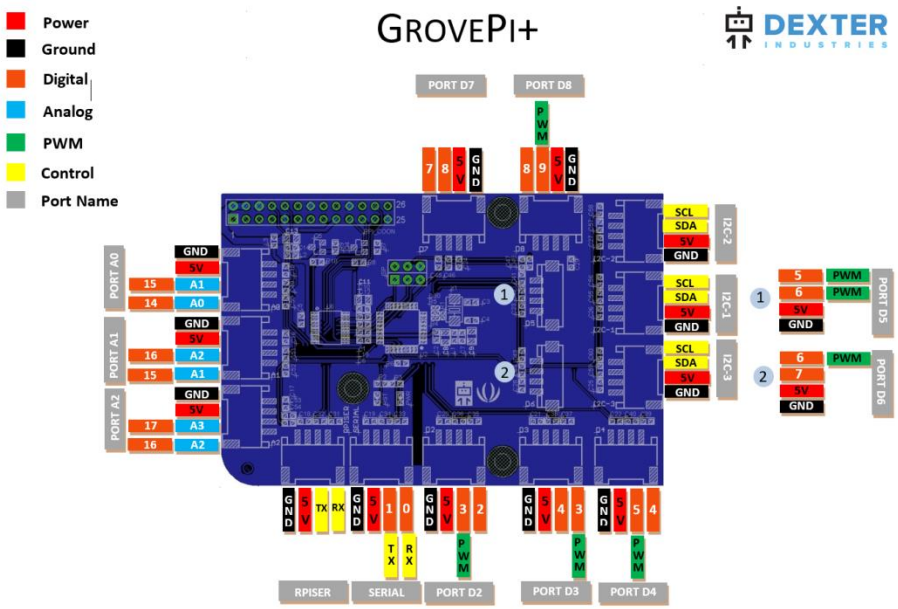


Figure 2: GrovePi Pinout Diagram

(Source: https://www.dexterindustries.com/GrovePi/engineering/port-description)

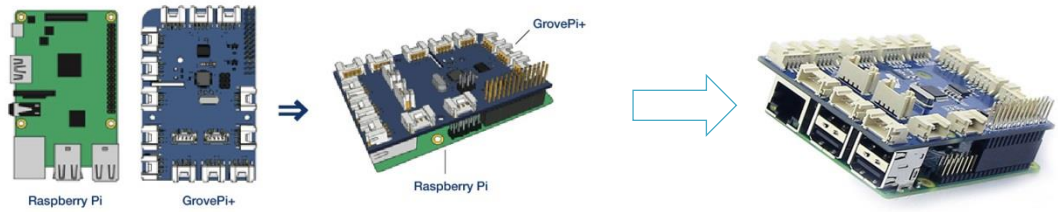
The **pinMode**() method is used to control the pin as an INPUT or OUTPUT mode. The methods **AnalogRead** and **DigitalRead** read data from various GrovePi sockets depending on the type. For example, AnalogRead(2) will cause the ATMEGA328 microcontroller to read from socket labeled A2 (Figure 2). The GrovePi board contains **seven** digital ports (D2-D8), **three** analog ports (A1-A3) and **three** I2C ports (I2C-1 – I2C-3) as shown in Figure 1. GrovePi sockets (or ports) labeled D2-D8 are all digital and support 1-bit input/output values (0-1) using digitalRead() and digitalWrite() methods. The GrovePi sockets (or ports) labeled **D3**, **D5** and **D6**) support **Pulse Width Modulation (PWM)** which enables writing 8-bit values (0-255) with **analogWrite**(). The **analogRead**() method is associated with analog sockets (A0, A1, and A2) and will not work with digital sockets (e.g. D2-D8). Similarly, digitalRead() method will only read from digital sockets (e.g. D2-D8). Not all GrovePi modules or sensors can be connected to analog, digital or I2C ports.

What is the Rotary Angle Sensor?

The rotary angle sensor produces analog output between 0 and Vcc (5V DC with Seeeduino) on its D1 connector. The D2 connector is not used. The angular range is 300 degrees with a linear change in value. The resistance value is 10k ohms, perfect for Arduino/Raspberry Pi use. This may also be known as a "potentiometer". A **potentiometer** has three terminal resistors which can control the flow of electric current. We will use the potentiometer to control the current that passes through to an LED (e.g. control LED brightness). The rotary angle sensor requires PWM for supporting 8-bit values and therefore will be associated with analog ports (i.e. A0, A1 or A2) (analogRead) or digital ports that support PWM (i.e. D3, D5 or D6) (analogWrite) on the GrovePi shield.

Preparation

1. Turn on the RPi with the GrovePi+ connected as shown below



2. **[Use GrovePi library from any directory]** Run the following command at the prompt to install or import the GrovePi Python library from any folder or directory on the RPi (e.g. you do not need to always run from the GrovePi folder that was created as part of the installation of the GrovePi library in Activity 1):

```
sudo pip install grovepi
```

This installation may take several minutes.

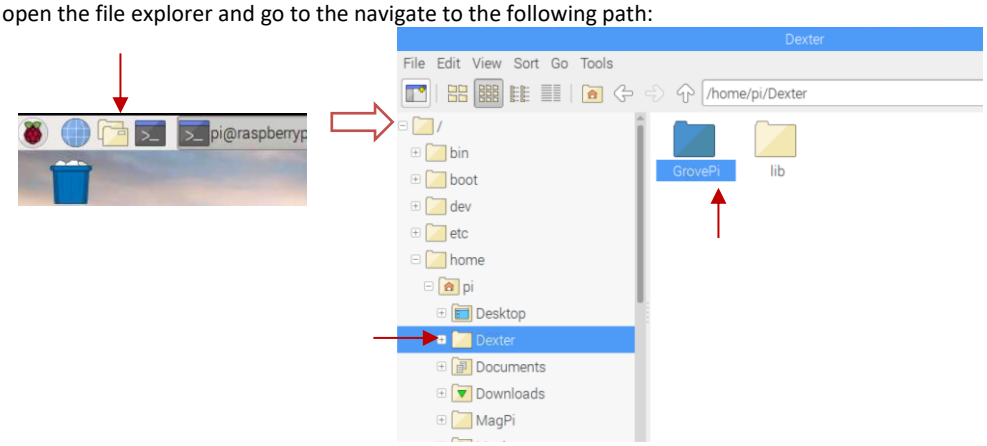
3. **[Update GrovePi Firmware]** Open a terminal window and then go to the following directory:

```
cd /home/pi/Desktop/GrovePi/Firmware
```

⚠ **If you receive the following message:**

```
pi@raspberrypi:~/Desktop/code/ac02 $ cd /home/pi/Desktop/GrovePi/Firmware
bash: cd: /home/pi/Desktop/GrovePi/Firmware: No such file or directory
```

Then, open the file explorer and go to the navigate to the following path:



Drag the GrovePi folder to the **Desktop** and re-run then navigate to the Firmware folder in the /home/pi/Desktop/GrovePi/Firmware path

```
cd /home/pi/Desktop/GrovePi/Firmware
```

Run the following command:

```
sudo bash firmware_update.sh
```

Then, install the GrovePi dependencies by entering the following commands:

```
cd Desktop/GrovePi/Script
sudo bash install.sh
```

```
sudo reboot
```

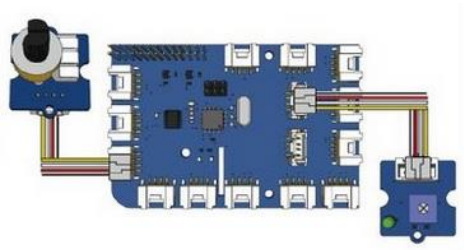
Once completed, reboot the RPi. In this case, you may run the following command. However, we need to turn off the power for the RPi in order to proceed to Part A to connect additional modules. Proceed to Part A.

> ⚠ **Caution**: Do not connect Grove modules while the RPi is **powered on** (this may damage the modules and the boards and may result in a hazard).

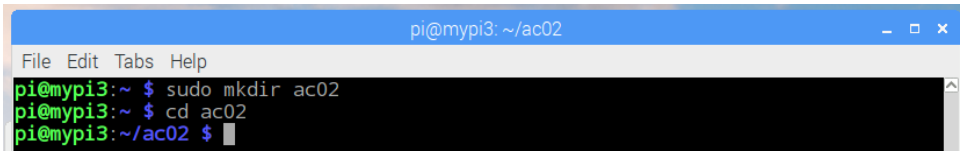
Part A: Using the Rotary Angle Sensor (or potentiometer)

(filename: **led_fade.py**)

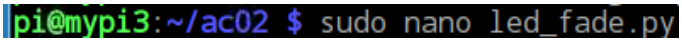
1. Connect your favorite LED **to Port D5** and connect the angle rotary sensor to **Port A2**. Then, turn on the RPi.



2. Turn on the RPi. Once the Raspbian OS loads launch the terminal window to start writing Python scripts. Let's now write some Python code to control the LED using the angle sensor. Open the Terminal program, create a folder called **ac02** and change directory (cd) to that folder

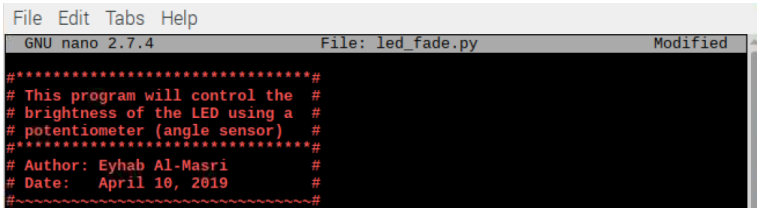


3. Open the terminal Create a new file called **led_fade.py**



At the top of the file, add the necessary documentation including:

a) brief description of the file
b) your name
c) date



Now, let's import the necessary libraries for (a) time and (b) GrovePi+ shield

```
import time
from grovepi import *
```

4. Let's create two global variables that will be used to identify the port numbers of the LED and angle sensor, respectively.

```
#*********************************#
# Global Variables                #
#*********************************#
myLED = 5               # connect LED to Digital Port 5 (D5)
angle_sensor = 2        # connec angle sensor to Analog Port 2 (A02)
```

Then, let's set the pinMode of the myLED to "OUTPUT" such that we control its output (e.g. behavior).

```
pinMode(myLED, "OUTPUT")        # Set the D5 pin mode to output
```

5. The **Pulse Width Modulation (PWM)** is a method for retrieving analog results with digital means. A digital control, for example, is used to create a square wave (signal switches ON/OFF). Such pattern simulates voltages between the ON (e.g. 5 Volts) and OFF (0 Volts). The ON duration is simply referred to as the pulse width. In order to vary the analog values, we simply change or modulate the pulse width.

Let's read the resistance value from the potentiometer and use it to control the LED fade:

```
while (1):
  try:
     # read resistance value from the angle sensor
     pot = analogRead(angle_sensor)
     print(pot)

     # Control the LED using PWM (pulse width)
     analogWrite(myLED, pot / 4)

  except KeyboardInterrupt:
    digitalWrite(myLED, 0)
    print("exiting ...")
    break
  except IOError:
     print("An error has occured.")
```

If you receive an error (other than syntax or logical errors) when executing the script, you may need to downgrade the kernel. Enter the following line (it is recommended that you copy/paste the following line to avoid any typos):

```
sudo rpi-update 52241088c1da59a359110d39c1875cda56496764
```

The complete code:

```
#*********************************#
# This program will control the   #
# brightness of the LED using a   #
# potentiometer (angle sensor)    #
#*********************************#
# Author: Eyhab Al-Masri          #
# Date:   April 10, 2018          #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
import time
from grovepi import *

#*********************************#
# Global Variables                #
#*********************************#
myLED = 5               # connect LED to Digital Port 5 (D5)
angle_sensor = 2        # connec angle sensor to Analog Port 2 (A02)

#*********************************#
# Initilization                   #
#*********************************#
pinMode(myLED, "OUTPUT")        # Set the D5 pin mode to output


while (1):
  try:
     # read resistance value from the angle sensor
     pot = analogRead(angle_sensor)
     print(pot)

     # Control the LED using PWM (pulse width)
     analogWrite(myLED, pot / 4)

  except KeyboardInterrupt:
    digitalWrite(myLED, 0)
    print("exiting ...")
    break
  except IOError:
     print("An error has occured.")
```

Part B: Using a Python-based GUI to Control LED

(filename: **led_gui.py**)

1. Let's install **tkinter** using your distribution's package manager. The package **tkinter** is Python's most widely used GUI (Graphical User Interface) package.

```
pi@raspberrypi:~/Desktop/code/ac02 $ sudo apt install python3-tk
```

Then run the following command to install a required library (**libjpeg-dev**):

```
pi@raspberrypi:~/Desktop/code/ac02 $ sudo apt-get install libjpeg-dev
```

Then, install the associated **guizero** package:

```
pi@raspberrypi:~/Desktop/code/ac02 $ sudo pip3 install guizero
```

> ⚠ It may take a few minutes to complete this installation.

2. Let's begin writing a program that displays a graphical user interface using Python. Call it **led_gui.py**

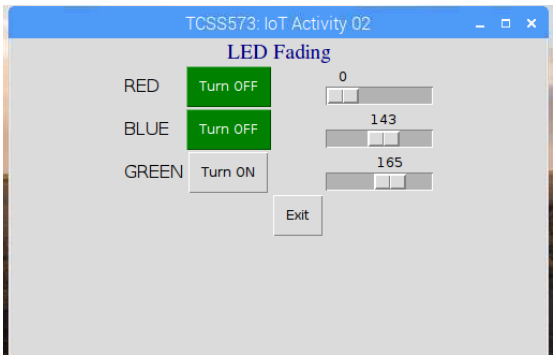
```
from guizero import App, Box, Text, TextBox, PushButton, Slider
from grovepi import *
import time

myLED1 = 5  # red
pinMode(myLED1, "OUTPUT")

def exitProgram():
    digitalWrite(myLED1, 0)
    app.destroy()

def sliderToggleLED1(slider_value):
    analogWrite(myLED1, int(slider_value))

def led1Toggle():
    button_status = digitalRead(myLED1)
    if (button_status == 0):
        digitalWrite(myLED1, 1)
        ledButton1.text="Turn OFF"
        ledButton1.bg = "green"
        ledButton1.text_color = "white"
        textLED1.color = "green"
    else:
        digitalWrite(myLED1, 0)
        ledButton1.text="Turn ON"
        ledButton1.bg = "red"
        textLED1.color = "black"

app = App(title="TCSS573: IoT Activity 02", height=300, width=500)
main = Text(app, text="LED Fading", size=14, font="Times New Roman", color="navy")
box = Box(app, layout="grid", grid=[1,0])

textLED1 = Text(box,text="RED", align="left", grid=[0,0])
ledButton1 = PushButton(box, command=led1Toggle, text="Turn ON", grid=[1,0])
sliderLED1 = Slider(box, start=0, end=255, command=sliderToggleLED1, grid=[4,0])

exitButton = PushButton(box, command=exitProgram, text="Exit", grid=[2,8])
app.display()
```

3. Run the **led_gui.py** Python script using the following command (note the use of **python3**):

```
pi@raspberrypi:~/Desktop/code/ac02 $ sudo python3 led_gui.py
```

Task: Modify the GUI such that it controls two more LEDs through this interface.

For example:



The **sliders** represent a software **emulation** of the rotary angle sensor (**potentiometer**). Therefore, the GUI sliders need to send commands to ports that support **Pulse Width Modulation (PWM)**. That is, there are only three ports on the GrovePi that support **PWM**. Hence, we need to use these ports for the potentiometer to work properly (or the emulation of that functionality via an **analogWrite** method). Which ports support PWM in the GrovePi)?

Hint: answer is in Part A!

Deliverable (upload to canvas → activities → activity 2) *(only one submission is required per group)*

☐	compressed ac02 folder containing files for Parts A and B
☐	short video exhibiting working ac02 **Parts A and B**

Grading

☐	correct output (3 marks for Part A, 4 marks for Part B)
☐	creative pattern (2 marks)
☐	documentation (2 marks) for 1 mark for Part A and 1 mark for Part B)

References (some content and images are adopted from the following reference):
- https://github.com/SeeedDocument/wiki_english/blob/master/docs/Grove-Rotary_Angle_Sensor.md
- https://www.dexterindustries.com/GrovePi/projects-for-the-raspberry-pi/led-fade/
- https://lawsie.github.io/guizero