DORDT UNIVERSITY

Engineering Department

EGR 304 LAB #4 GETTING STARTED WITH RASPBERRY PI SPRING 2020

OBJECTIVES

- Get a Raspberry Pi up and running.
- Get a "railroad crossing" or "cylon eyes" style project running via the GPIO on the Raspberry Pi.

REFERENCES

- Getting started with Raspberry Pi: https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started
- Getting started with Mu https://projects.raspberrypi.org/en/projects/getting-started-with-mu
- Flashing an LED: https://projects.raspberrypi.org/en/projects/physical-computing/5
- GPIO Zero: A Friendly Python API (Library) for Physical Computing: https://gpiozero.readthedocs.io/en/stable/
- Python Documentation: https://docs.python.org/3/contents.html
- Python Time Library Documentation: https://docs.python.org/3/library/time.html
- Other Web resources as you see fit.

DELIVERABLES

After you get your LED program working please invite the instructor to your workbench and demonstrate your version. Use a web browser on the Raspberry Pi to log onto the EGR 304 Canvas page. Upload your source code file using the link provided in the "LABS" section of the EGR 304 Canvas page.

ACTIVITIES

Using the references above, set up a Raspberry Pi on your workbench. Install an IDE such as Mu or any other one of your choice if you have experience with something else. Use a "Pi Cobler" to connect the GPIO port to a breadboard. Use a buffer chip to drive some LEDs, at least two. That's all!

Along the way, feel free to update the Raspbian operating system (recommended).

Raspberry Pi programs always run under the supervision of the operating system (we use Raspbian.) The program does not have the "setup(), loop()" structure that the Arduino IDE does. You need to create that type of program flow.

A typical Python program has this general structure:

```
from gpiozero import LED   #imports the LED method from the gpiozero library
from time import sleep     #imports the sleep method from the time library

led = LED(17)            #led is a structured variable set up by method LED()

#Everything above corresponds to the setup() section for an Arduino program.
#The next "while" statement creates the same effect as loop() in Arduino.

while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```