Embedded Systems lab 1

# Objective

The objective of this lab is to learn basic principles of input/output with the Raspberry Pi platform. These principles include:

* Create and clone a GitHub repository to store the project and allow it to be moved from the development machine to the Raspberry Pi
* Become familiar with available Python libraries for running web servers and threading
* \*\*\*\*\*\*\*

# Materials

Materials needed to complete this lab are:

* 1x Raspberry Pi (I used a Raspberry Pi 400) with Raspberry Pi OS installed
* 1x USB-C Power supply sufficient for powering the Raspberry Pi
* 1x Red LED
* 1x Yellow LED
* 1x Green LED
* 3x 220Ω Resistor
* 3x M-M jumper wires
* 1x Half size breadboard
* 1x Raspberry Pi GPIO ribbon cable
* 1x Raspberry Pi GPIO breakout

# Procedures

## Hardware

1. Connect the GPIO ribbon cable to the Raspberry Pi’s GPIO pins. On a Raspberry Pi 400, the connector is keyed, so it can only be inserted in the correct orientation. Connect the other end of the ribbon cable to the GPIO breakout.
2. Connect the GPIO breakout to the breadboard. Make sure the power pins on the side get connected to the power rails of the breadboard such that the +5v and +3.3v are on the rail marked as positive (a red stripe in my example).
3. Connect the LEDs, resistors, and jumper wires as pictured \*\*\*\*
4. Connect the power adapter to AC power, and plug in the Raspberry Pi.

## Software

1. Acquire the IP address of the Raspberry Pi by running the command **ifconfig** in a terminal.
2. Download the source code from GitHub or by copying the file structure and code onto your Raspberry Pi.
3. Start the Flask webserver by running the command **python3 app.py**. If there are any errors, make sure to resolve any dependencies by downloading them using PIP.

I am using Python3 because Python is a simple language with a robust support for most things including web servers and GPIO libraries for the Raspberry Pi.

Sources:

Flask routers tutorial: <https://flask.palletsprojects.com/en/2.0.x/quickstart/>

Serving static HTML in flask: <https://stackoverflow.com/questions/20646822/how-to-serve-static-files-in-flask>

Global mode to modify variables within flask routes: <https://stackoverflow.com/questions/19182963/global-variable-and-python-flask>

Threading in python: <https://www.geeksforgeeks.org/how-to-create-a-new-thread-in-python/>

Toggle switch code: <https://stackoverflow.com/questions/39846282/how-to-add-the-text-on-and-off-to-toggle-button>