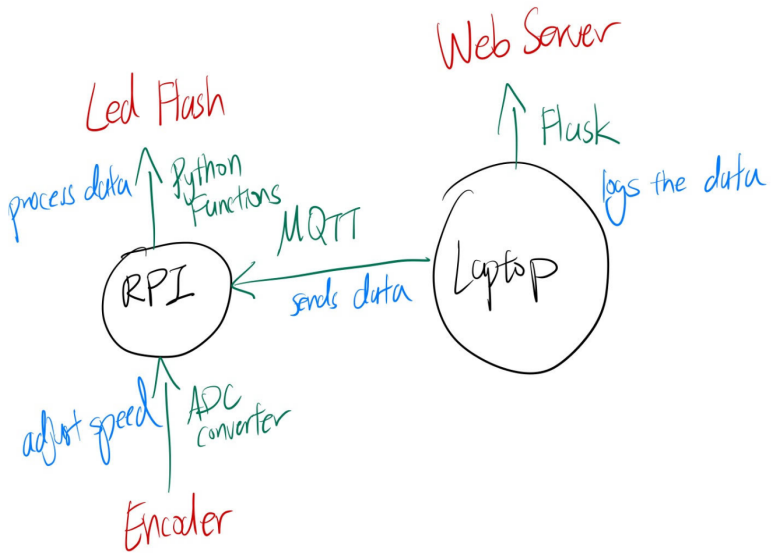
Team Members: Eric Chen

Clear description of what your IoT system is trying to achieve:

1. The server (laptop), should allow the user to send packets of text through mqtt to the client(RPI).
2. Whatever was sent through in the current session should be logged. The server uses Flask to save what was transmitted.
3. Upon receiving the packet, the client must use the data processing map defined in the sensor.py file to convert text into morse code.
4. The user may change the speed of how fast the morse code runs by using a rotary encoder attached to the client.

Block Diagram:




Description of components, platforms, protocols used, and processing/visualization techniques:

Components:

- RPI, used to process data and host sensors
- Virtual Machine, used to send user data to RPI
- Grove Pi Led, used to display the morse code
- Grove Pi rotary encoder, used to change the flashing speed

Platforms:

- Python (and it's included libraries), handles all the functionalities, arrows in the diagram
- Linux, is the OS for RPI and the laptop

Protocols used:

- Flask, used to create a dummy web server to log sent data
- MQTT, used to transmit information from laptop node to RPI node

Processing and Visualization:

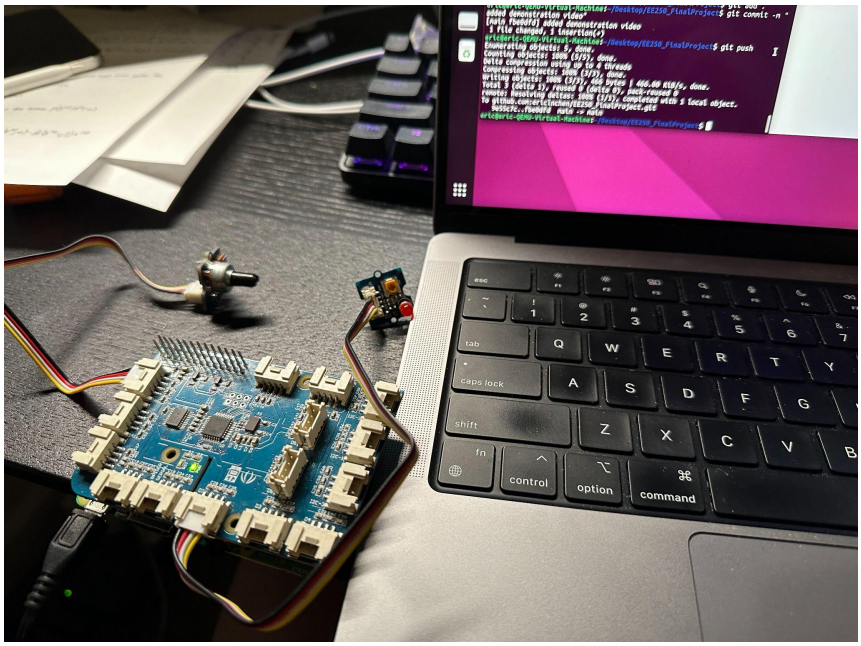
- Mapping words to morse code to flashing led
- Uses flask to store past transmissions

Reflection, discussion of limitations that demonstrates insights to their cause and possible remediation, lessons learned:

1. Was going to allow bi-directional communication, but I had a hard time setting up a server on the RPI side. The VM laptop with hotspot can't seem to connect to my RPI, even though the ip and hostnames were correct. Next time I would use a native device with linux and through wifi or ethernet, not a hotspot.
2. Would've been more challenging but better to use influxdb or any other data logger that visually makes the transmitted data much better.
3. Next time I will probably also implement a LCD and a buzzer on the RPI side. The LCD will act as a reminder of whether data was being sent or received. And the buzzer would be another way of recognizing morse code.

Presentation:

The Setup:




The demonstration videos are in the txt file on this project's github.
https://github.com/ericlnchen/EE250_FinalProject