Charlotte Kroll & Charlie Welland

EE 250

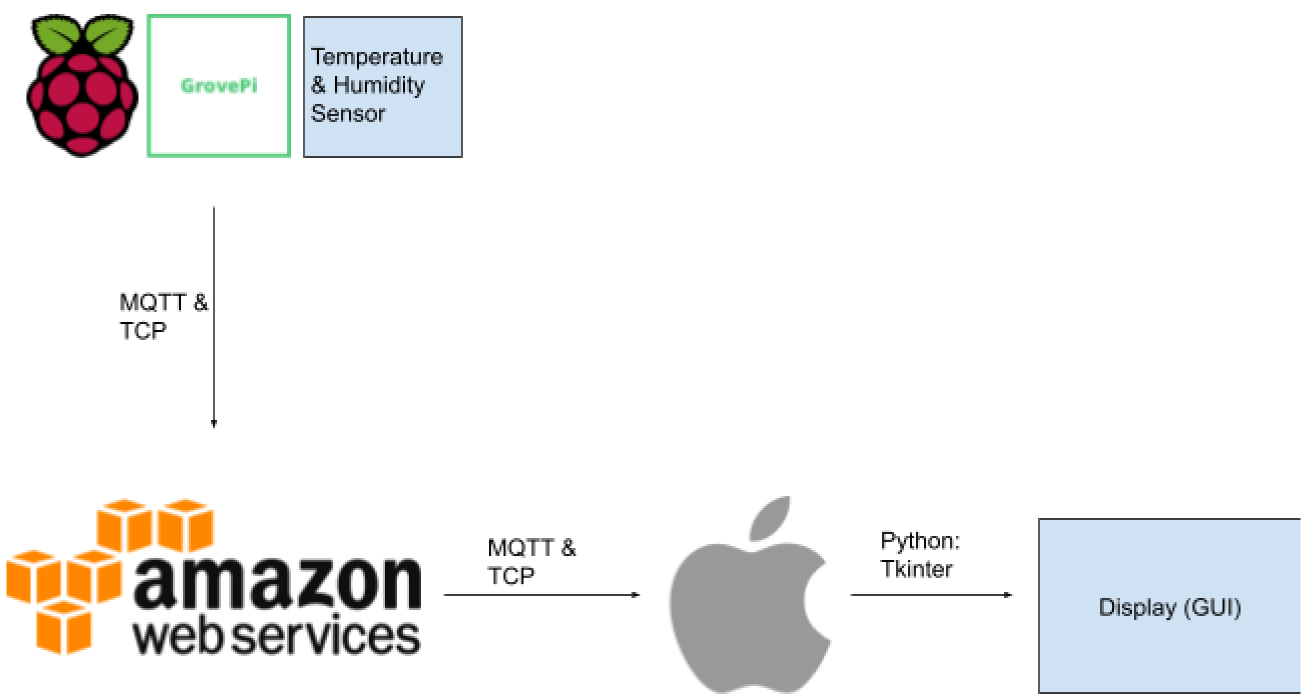
5 May 2023

IoT Weather Display

Project Goal

In this project, we decided to create a weather display interface that employs the use of a RaspberryPi and a GrovePi temperature and humidity sensor. The goal in this IoT system is to collect data from the GrovePi temperature and humidity sensor and display it on a computer interface while simultaneously running each of the collected values through an equation (sourced from the National Oceanic and Atmospheric Administration, or NOAA) to calculate and display the Heat Index. Each of these values are displayed in real time, with a minimal delay, to show clients accurate weather information.

Block Diagram of our IoT System



Components/Platforms/Protocols

The basis of our project revolves around the AWS (Amazon Web Services) cloud and two main protocols, MQTT and TCP. Starting with the temperature and humidity sensor, which has a built-in ADC, the data is collected and connects to an iPhone via Personal Hotspot which allows the data to be sent via MQTT and TCP to AWS. The only direct connection between the RPi/GrovePi to the computer itself is through an SSH. AWS then sends the data to the computer using TCP and MQTT, then a Python library called Tkinter is used to create a GUI, or Graphical User Interface, to display the data. This connection also required certificates from AWS to ensure that this information is authorized to access and display.

Reflection

Throughout this project, we ran into a few roadblocks and checkpoints that required some extra attention and troubleshooting. For starters, instead of using AWS, we originally were using Microsoft Azure, which created a myriad of issues. It was slow-acting and glitchy with MQTT and TCP connections, so after identifying that the cloud computing platform was the issue, we went ahead and tried AWS, which is far less buggy. One limitation of this IoT system is the speed at which it updates. When the temperature and humidity are changed, the sensor takes about a full second before it updates. Since the MQTT communications used are fairly lightweight, it is likely part of the processing speed of the temperature sensor itself. Upon further research, we found that the sensor is intended for data collection and transmission over time, especially since weather rarely changes drastically within the matter of seconds. This IoT system is much better suited for a weather situation rather than another temperature-sensing need, such as the measurement of temperature in a rapidly heating/cooling environment. A possible remediation of this limitation would be to replace the sensor altogether with one that is more suited for rapid updates.