Abertay University

Twitter Bot

Kieran Cameron - 1900829

CMP408 – IOT and Cloud Secure Development

Introduction

Companies and Home users alike are constantly looking for ways to automatize technical tasks. This could be for a variety of reasons such as saving valuable time and resources which could otherwise be spent else where. In recent years with technology which was previously been kept by a select few individuals now being made available to the wider public, there is no better time than now to automate homes and business spaces. With this in mind there are many tools available on the open market to enable automation. These can vary from cloud services, IoT technologies and physical devices. The project in question uses a combination of all 3.

The Twitter-Bot project aims to utilize data gathered locally and demonstrate how it can work with cloud services and webhooks to be sent to the wider world whilst also being automized with minimal user interaction. This project aims to meet the following objectives:

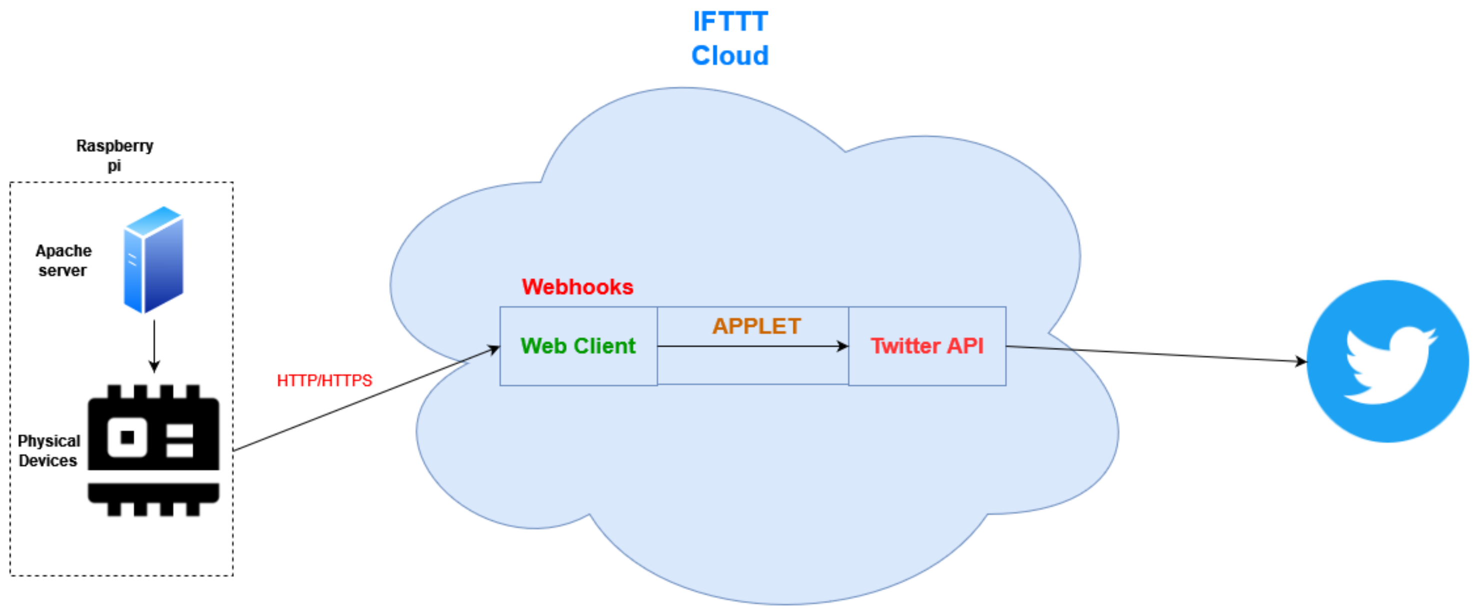
1. *Assemble and program a physical device to read in data and display locally.*
2. *Design and create a webapp to control the device from a user-friendly interface.*
3. *Using cloud services, API's and webhooks send the collected data through the cloud and to an application*
4. *Automate this system using python scripts*

Methodology

A raspberry Pi Zero w was used along with a DHT Temperature Sensor and LCD1602 Display module to accurately capture the temperature and humidity at the time and then display it on the LCD module. This was done by coding a python script on the pi. Once the physical components were confirmed to be working the next step was to connect the device to a cloud service to enable the data to be sent to a remote location. In the context of this project it was deemed that webhooks were the best solution as automation was a key component in the vision of this system. For this purpose IFTTT was used (if This Then That). The IFTTT cloud is used for connecting various services and devices together using its online applet. So in this context the two parts of the service would be as follows:

1) If This – Temperature sensor provides a value

2) Then That – send these values through the twitter API to make a post with the provided values

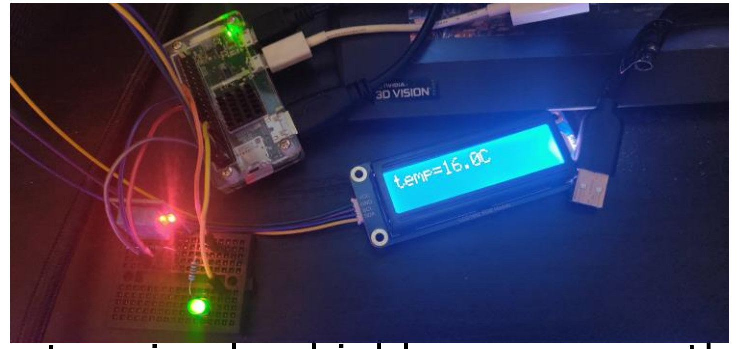


The original python script was altered so that a reading would be sent out every 60 minutes, although this can be changed at any time.

Once the core functionality was finished, a more user-friendly interface was required. This was done by creating a local Apache server on the raspberry pi. Once this server was setup a simple webpage was created using HTML and CSS. In order to control the GPIO pins on the device a python script would be made which would be connected to the webpage using a python micro-framework known as FLASK. An additional component LED was connected to the device to let the user know when the GPIO pins are active or not. When the activate button is triggered on the Web server the GPIO pin states for the Sensor and LED are set to High and when the disable button is triggered both pin states are set to Low. It is noted that an alternative solution to using Flask would be to manually use an modified LKM to control the GPIO pins but in the context of this project FLASK was the more efficient and secure option as altering a LKM could lead to problems down the line such as file corruption and random crashes.

Project Highlights

This project was successfully able to create a hardware based system able to read in data and then transfer this data automatically to the cloud using a html interface. In its current state the temperate sensor could be substituted with any physical component that gather data and with some minor adjustments to the python scripts it would work with anything



The system is also highly secure as the web server has CSRF protection. IFTTT also uses standard web-encryption when transferring data.

Future Work

Create the webserver on the cloud as opposed to locally so that the sensor can be triggered from anywhere and integrate HTTPS.

Create an app on android or IOS to control the sensor

References

push_resetFollowMore (n.d.). *Connecting Your Raspberry Pi to the Web*. [online] Instructables. Available at: https://www.instructables.com/Connecting-Your-Raspberry-Pi-to-the-Web/.