Raspberry Pi Automated Twitter Bot

Kieran Cameron - 1900829

CMP408: IoT and Secure Cloud Development

BSc Ethical Hacking Year 4

2022/23

Note that Information contained in this document is for educational purposes.

.

Contents

.

1 INTRODUCTION

1.1 PROJECT RELEVANCE

With new IoT and cloud technology becoming widely available to the public in recent years. Research has shown that open-source software now makes up around 70-90% of available resources (Jason Perlow, 2022) this has allowed for new possibilities in the realms of smart devices. One such benefit is automation, with these technologies becoming available it is growing ever easier to implement and design projects able to automatically carry out a specific purpose. Research shows that in the US 60% of existing jobs could be automated (Michael Chui, 2015). whereas before it would require human resources and time which could otherwise be spent elsewhere on more specialist tasks. This project allows for data to be gathered locally and then sent automatically to the cloud with minimal human interaction once it has been initially setup

1.2 SECURITY CONSIDERATIONS

Throughout the duration of the project security was taken into consideration and implemented where necessary some examples include:

- Utilizing HTTPS secure encryption
- CSRF protection
- Strong Passwords for user account
- Using security verified services.
- Requiring an authentication key to use the Twitter bot.

1.3 AIMS

The aim of this project is to create a automated Twitter bot which receives data from a local device and then uploads this data to the cloud where it is processed and then makes a specified tweet.

- Design and Assemble hardware to read in and display the current temperature.
- Design and create a local webapp to control the hardware from a user-friendly html interface.
- Use cloud services, Webhooks and APIs to send the collected data through the cloud and to a selected application.
- Automate the system.

1.4 REQUIRED HARDWARE

Component	Quantity
Raspberry Pi Zero W	X 1
DHT11 Temperature Sensor	X 1
330-ohm Resistors	X 2
Green LED	X 1
Breadboard	X 1
LCD1602 RGB Display	X 1
Micro SD card	X 1
Jumper Cables	X 5

2 PROCEDURE

2.1 RASPBERRY PI HARDWARE SETUP

The hardware should be set up as seen in figure 1.

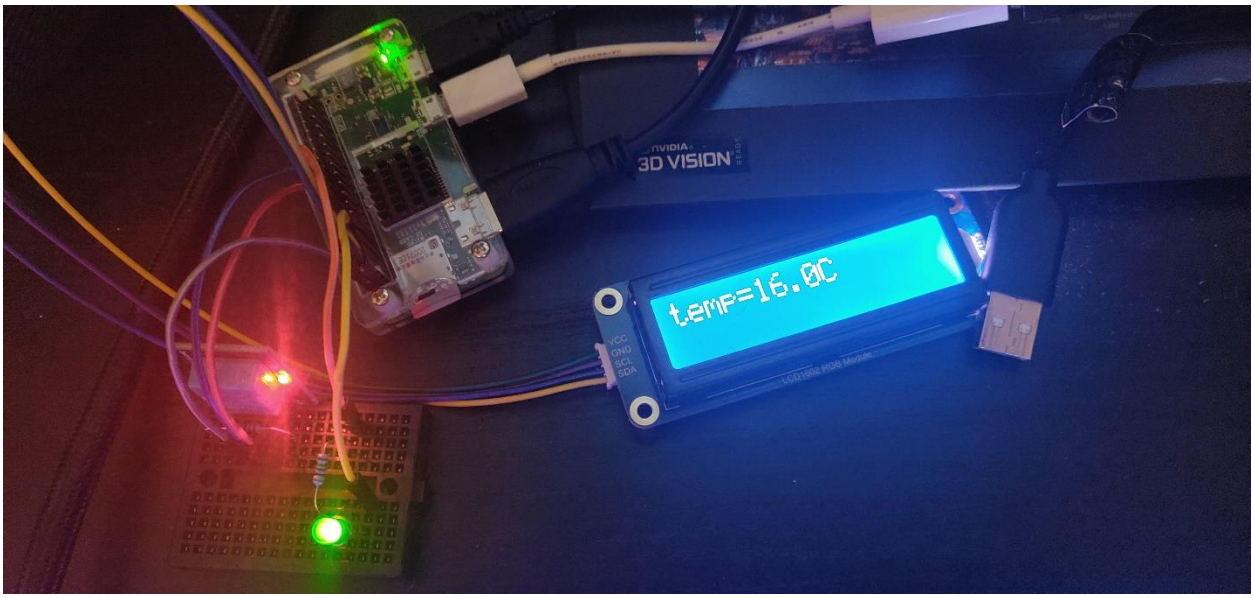

Figure 1 – Hardware assembly

- The temperature Sensor was connected to GPIO pin 23. Ensure there are resistors between the data and voltage pins on the Temperature sensor.
- The Display was connected to Pins 2 and 3.
- The LED is connected to pin 25. Ensure there is a resistor connected to the LED.
- Ensure other connections are configured correctly e.g., Ground and appropriate voltage pins.

2.2 SOFTWARE

2.2.1 Sensor and display script

With all the components set up correctly. Next software would need to be developed in order to program the Sensor to read the temperature and humidity and then display that information onto the LCD display. This was done using a simple python script. This script is called the adafruit library (library used for the sensor). An external python file was taken from the LCD1602 website which contains all the essential libraries for the display to work. This file was called within the Temp.py file called RGB1602. See appendix for full python script.

2.2.2 IFFFT

Once the initial software was working the and device was reading in the temperature the next step was to connect the device to the internet and send the data through the cloud to the twitter API. To do this the IFFFT cloud was used as this best suited the purpose of this project. An applet was configured to act as a webhook sending data from the sensor to the twitter API. IFFFT would firstly need to be configured. IFFFT stands for If This Then That, so the formula follows 2 parts:

1. **IF This:** Temperature reading is available.
2. **Then That:** Send these values to the twitter API to make a tweet with the provided values.

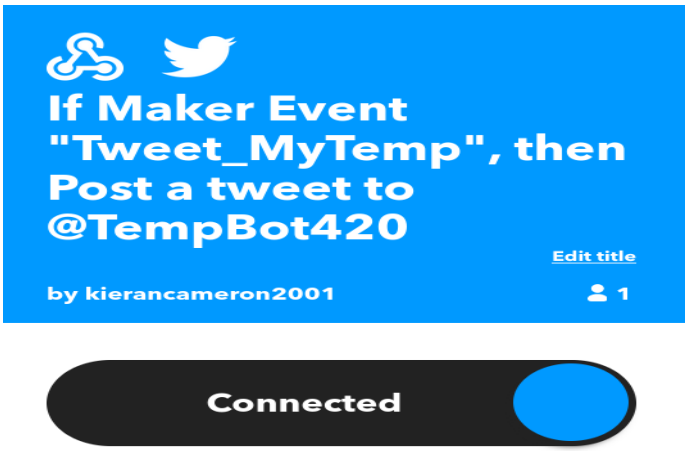

Figure 2 – IFFFT Formula

Once this was setup the webhook would also have to be configured with the message and twitter account that it should post to. A new account was created for the purpose of this project aptly name TempBot420. The message to display was plugged with the values read from the python script hosted on the Pi. The applet was able to read from the python file due to an Personal certification Key provided by the IFFFT website.

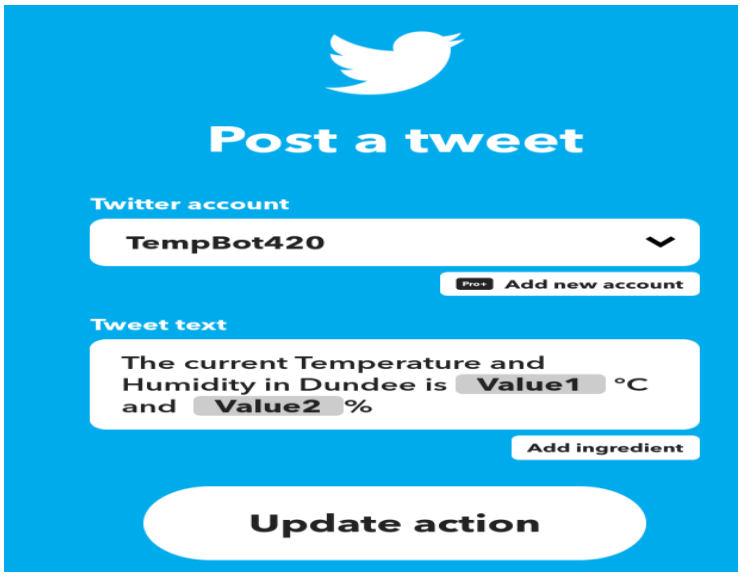

Figure 3 – Twitter account and text

Once this was set up it was time to test whether the webhook functioned properly. The script was run from the Pi and successfully made a tweet, see figures 4 and 5. This was able to be configured within the python file to regularly make tweets at specified intervals if the script was running automating the process.

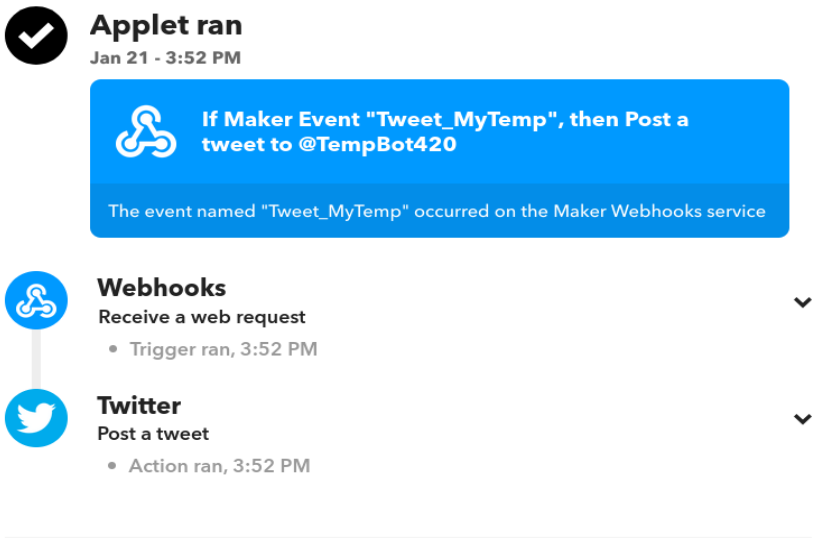

Figure 4 – Applet Activity

Figure 5 – TempBot Tweet

2.2.3 HTML Interface

With the essential functionality now working the last step was to make a user-friendly interface for controlling the Pi from an HTML Webpage. This was done by hosting an Apache web server from the Pi locally, See Figure 6.

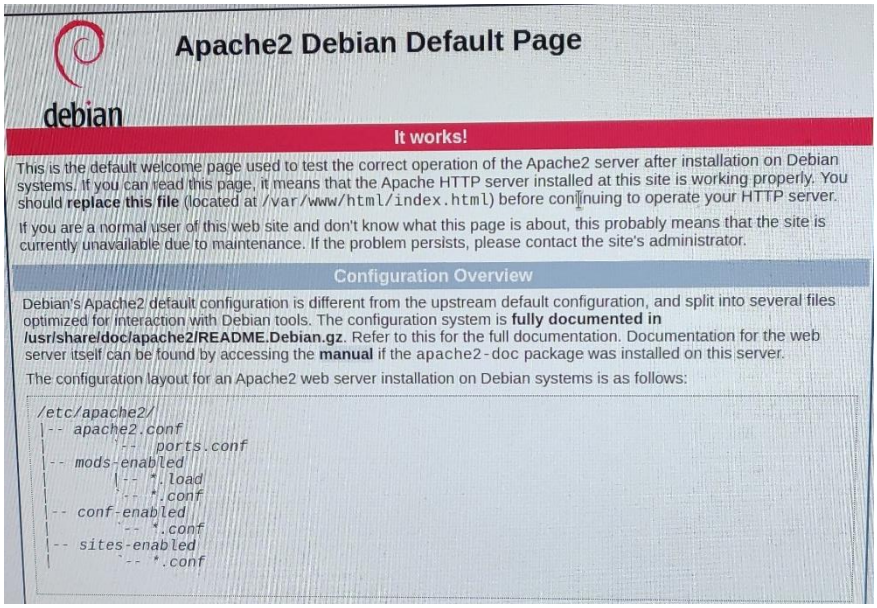

Figure 6 – Apache Webserver

With the Apache server now running, a basic interface was made using HTML and CSS. In order to control the Pi remotely from the webpage, a new python script was made to link the buttons on the webpage to the Pi's GPIO pins. This was done using a python micro-framework known as FLASK. An additional LED was connected to the Board to indicate when the GPIO pins are set to LOW or HIGH. See figure 7 for the interface.

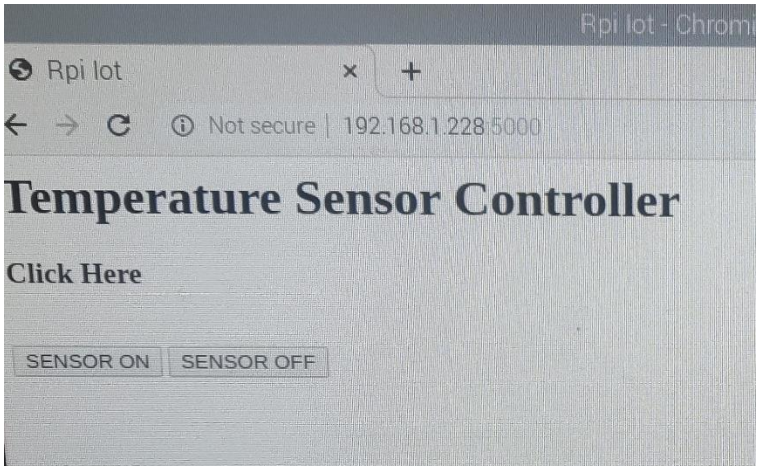

Figure 7 – HTML interface

Whenever Sensor On is clicked the webserver sends a signal to the GPIO pins for the LED and Sensor to be set to High and set to Low when Sensor OFF is clicked.

3 CONCLUSION

3.1 OVERALL CONCLUSION

In conclusion This project was able to deliver on its aims using minimal and affordable components. The Raspberry Pi Zero was able act as a Webserver and could communicate with cloud services in order to transfer data successfully. The system should work out of the box and would prove to be easily used with its simple HTML interface. Due to its autonomous nature, once, the system is set up it can be left to its own devices and will ensure the specified tweet is sent at the chosen intervals.

3.2 FUTURE WORK

This project could be worked on further by creating a Live Webapp in the cloud so that It could be accessed anywhere this webapp would have to use HTTPS and have certain security measures in place such as an user verification process or Login of some sort. The Web UI could also be expanded by adding more features such as choosing the intervals when the Tweet is sent instead of manually changing it in the python code. An app for android or IOS could potentially be developed giving the user even more options of control.

REFERENCES

VMware. (2022). *What is Cloud Automation? | VMware Glossary*. [online] Available at: https://www.vmware.com/topics/glossary/content/cloud-automation.html.

Chui, M., Manyika, J. and Miremadi, M. (2015). *How Many of Your Daily Tasks Could Be Automated?* [online] Harvard Business Review. Available at: https://hbr.org/2015/12/how-many-of-your-daily-tasks-could-be-automated.

www.linuxfoundation.org. (n.d.). *A Summary of Census II: Open Source Software Application Libraries the World Depends On - Linux Foundation*. [online] Available at: https://www.linuxfoundation.org/blog/blog/a-summary-of-census-ii-open-source-software-application-libraries-the-world-depends-on.

Instructables (2013). *Simple and Intuitive Web Interface for Your Raspberry Pi*. [online] Instructables. Available at: https://www.instructables.com/Simple-and-intuitive-web-interface-for-your-Raspbe/.

Programming, C.B. | and Pi | 34, R. (2015). *How to Write and Run a Python Program on the Raspberry Pi*. [online] Circuit Basics. Available at: https://www.circuitbasics.com/how-to-write-and-run-a-python-program-on-the-raspberry-pi/.