

# Pi Temperature Bot

Jack Bowker

CMP408

## Introduction

This project will involve the development of a temperature monitoring bot. The project will make use of a Raspberry Pi Zero, a digital thermometer, and a button to trigger the bot.

To power the bot, Amazon Web Services will be used to provide an API using API Gateway, as well as hosting the bot in a Lambda function in the cloud.

This will be connected to a Twitter account which is where the temperature will be posted.

## Objectives

- Create a Loadable Kernel Module that will accept input from a button.
- Create a Python userspace program that will communicate with the driver using signals, receive the output from the thermometer, and transmit the temperature.
- Create an AWS API using API Gateway to make an endpoint that the userspace application can communicate with.
- Create an AWS Lambda function to communicate with the Twitter API and post the tweet.
- Ensure only the userspace application can post tweets using the API endpoint.
- Give user feedback when the button is pressed.
- Ensure the temperature is output in a readable format.

## Methodology

### Hardware & Kernel

- The Kernel layer handles communication between the button and the userspace application, accounting for debouncing and sending UNIX signals to the userspace app.
- Also controls the LED that is lit when the button is pressed.
- When userspace application is launched, the PID is sent to this module so it knows where to send button press signals.



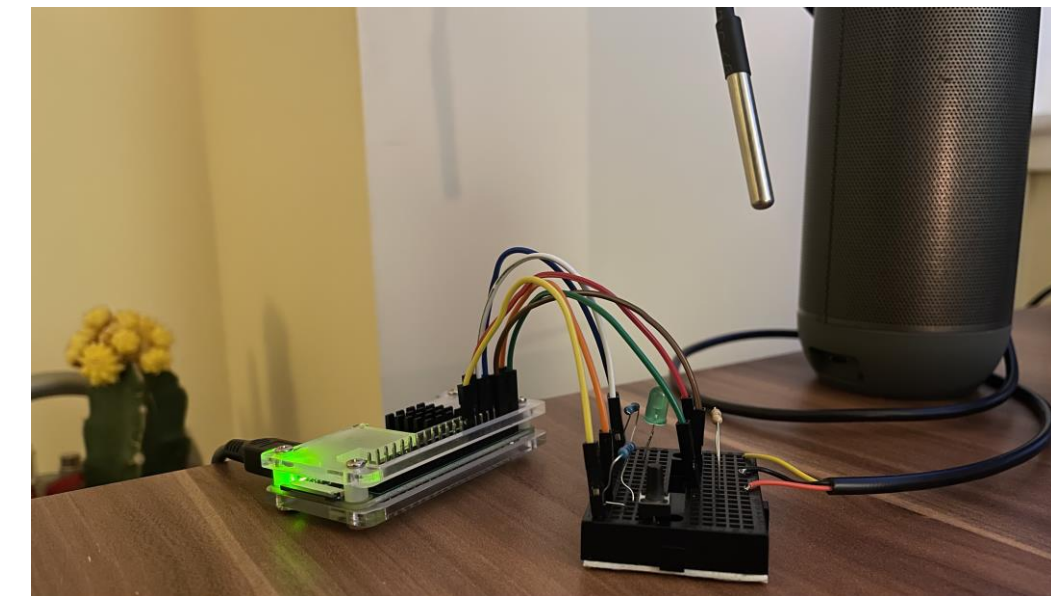
Pi Temperature Bot  
@TemperaturePi

Temperature is currently 19.187°C  
[2021-01-11 17:04:13]

17:04 · 11/01/2021

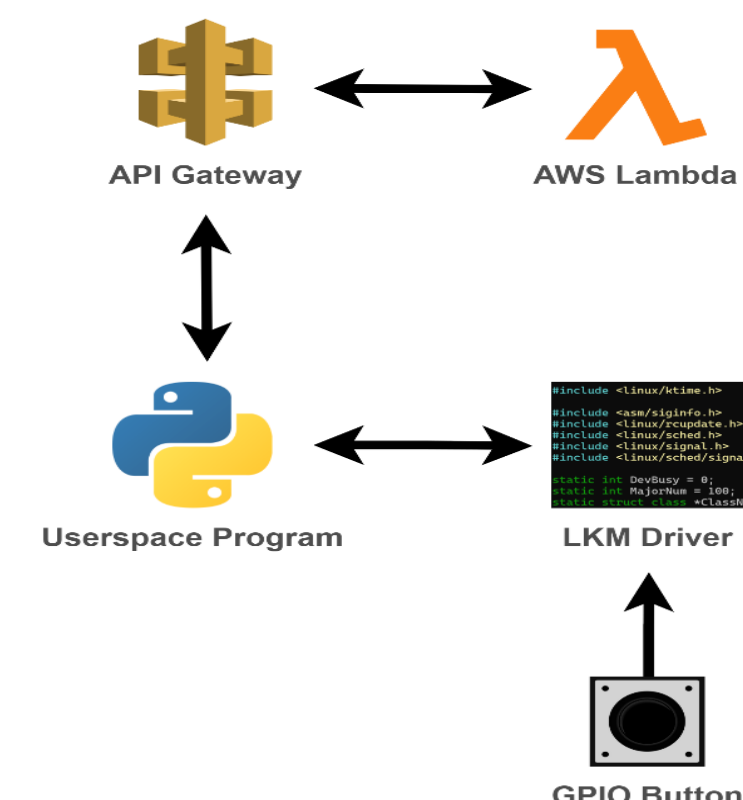
### Cloud

- AWS was used to host an API using API Gateway, where the userspace application can post the temperature along with an authorisation key.
- AWS Lambda was used to host a Python function, which parses data from the API along with checking the authorisation key.
- Lambda function communicates with Twitter API using Tweepy, after parsing data sends tweet containing the temperature along with the current time.



### Userspace

- Python was used to develop the userspace application.
- When launched, the PID of the program is fetched and sent to the kernel module to let it know where to send button signals.
- Application communicates with the thermometer to receive the temperature.
- When a signal is received, the data is posted to the cloud API.



## Project Highlights

### Features

- Button that when pushed gives feedback to the user via LED.
- Bot sends a tweet with current room temperature and time.

### Security Features

- API endpoint uses HTTPS to securely transfer data.
- Authorisation key is used to verify POST requests.

## Future Work

- Implement a better form of authentication, such as JSON Web Tokens in AWS.
- Run the temperature bot on a timer to get a consistent output without user intervention.
- Implement a graphing feature and a way to find the average temperature.

## References

- Pinout, 2020. W1-GPIO - One-Wire Interface. [Online] Available at: [https://pinout.xyz/pinout/1\\_wire](https://pinout.xyz/pinout/1_wire)
- Python, 2020.fcntl — The fcntl and ioctl system calls. [Online] Available at: <https://docs.python.org/3/library/fcntl.html>
- Tweepy, 2020. Tweepy Documentation. [Online] Available at: <http://docs.tweepy.org/en/latest/>