**Automated-Home-Heating-and-Automated-Plug-IOT-System**

**Overview**

This project is a home heating and automated smart plug IOT system. It was built using a Raspberry Pi, Sense Hat and a BG Smart Plug. Already existing devices were used such as a Hive hub for wireless home heating access and control.

For the project the Raspberry Pi collects temperature data from the Sense Hat every 10 seconds but in a real-world application could be set to every few hours. Once the temperature reached a set threshold the heating is activated, giving a boost for 30 minutes. The boost time can be modified in code.

The smart plug is set to come on at sunset in accordance with the user’s location. This will activate and switch on a light. This can be used as a security feature to imply someone is at home.

Applets were created for both the plug and the heating. This was done via IFTTT. Once triggered the heating is turned on or the plug turned on and an email is received to indicate the even has occurred.

The Data relating to the heating system is published to the ThingSpeak platform using the MQTT protocol.

Initially this project was to include energy consumption and bill estimation but due to a mix up with the plug received.

**Setup**

Connect the Sense Hat to the Raspberry Pi by placing it on the GPIO pins. Reboot the Raspberry Pi.

At the command line you will need to install the following

* sudo pip3 install paho-mqtt
* sudo pip3 install python-dotenv
* sudo apt-get install sense-hat
* pip install astral
* pip install pytz

A screenshot of a computer

Description automatically generatedAn IFTTT account will need to be created and Applets for the Plug and Hive hub created and using a WebHook to incorporate into the Python scripts. The Applets include an email which is sent once the event is triggered for both the Plug and Heating. Example of Applet settings below.

There are three scripts developed using Python for this project.

1. Temp.py – Run using python temp.py in the commend line.
2. Plug.py – Run using python plug.py in the commend line.
3. Thingspeak-pub.py.

To run the Thingspeak script use: “python thingspeak-pub.py mqtt://mqtt3.thingspeak.com:1883” in the command line. Opening your channel in the website you can have a display similar to the below.

A screenshot of a computer

Description automatically generated

Assuming your device is set up on ThingSpeak, you can proceed with the credentials for your device downloaded as plain text from ThingSpeak. A file is then created called .env where the credentials from ThingSpeak and your channelId (Again ThingSpeak) are inputted. The file will appear similar to the below.

username = ABC123

clientId = CBA321

password = xxxxxxxxxxxxxxx

channelId = xxxxxxxxxxxxxx

transmissionInterval = 15