# **IoT Room heating and temperature control system - Project Instructions**

# Introduction:

The system is designed to monitor and control room temperature, automatically managing a portable fan heater based on predefined temperature thresholds.

# Equipment:

The following equipment has been used:

* Arduino MKR Wifi 1010
* Arduino MKR IoT Carrier
* Smart plug (TP-Link Tapo P105)
* Arduino IDE
* IFTTT
* ThingSpeak
* Firebase
* Glitch

# References:

This is a list of setup instructions for each component:

1. **Smart Plug Setup:** [Tapo P105](https://www.tp-link.com/en/home-networking/smart-plug/tapo-p105/)
2. **Arduino MKR Setup:** [Cheat Sheet](https://docs.arduino.cc/tutorials/mkr-iot-carrier-rev2/cheat-sheet)
3. **Arduino IDE Installation:** [Download and Install Arduino IDE](https://support.arduino.cc/hc/en-us/articles/360019833020-Download-and-install-Arduino-IDE)
4. **IFTTT Setup:** [IFTTT Registration](https://ifttt.com/join)
5. **ThingSpeak Setup:** [ThingSpeak Documentation](https://uk.mathworks.com/help/thingspeak/)
6. **Firebase Setup:** [Firebase Configuration](https://www.hackster.io/OscarF10/mkr-wifi-1010-firebase-9a7399)
7. **Glitch:** [Glitch Tutorials](https://support.glitch.com/c/tutorials/23)

# Step by Step:

## Step 1: Set Up References

Make sure to follow the provided references for each component to set up accounts, install software, and configure devices.

## Step 2: Arduino MKR1010 Wifi and MKR IoT Carrier

#### Connect Arduino MKR1010 Wifi and MKR IoT Carrier to Your Computer and Open the Arduino IDE.

## Step 3: Run Code from GitHub Repository

Download and run the 'IoT-RoomTempConSys.ino' file from the [GitHub repository](https://github.com/mmazzet/IoT-project). This code allows real-time temperature monitoring and communication with connected services. Remember to add a secret.h file and include all the log in details.

## Step 4: ThingSpeak and ThingHTTP requests

#### Setting Up a Channel:

1. Click on "New Channel" and fill in the required information:
   * **Name:** Use a descriptive name
   * **Field 1:** This corresponds to the temperature data to monitor.
2. **Once the channel is created, click on “Add Widgets”.**
   * **Add one or more widgets**
   * Once the widget appears in the dashboard, ensure you edit it and select the Field 1 for the temperature data.

#### Setting Up ThingHTTP:

1. **Navigate to ThingHTTP:**
   * In the ThingSpeak menu, select the "Apps" and then "ThingHTTP".
2. **Create a New ThingHTTP:**
   * Click on "New ThingHTTP" to create a new ThingHTTP app.
3. **Configure ThingHTTP for Low Temperature:**
   * For Temperature LOW, set the following:
     + **Name:** Temperature LOW
     + **URL:** https://maker.ifttt.com/trigger/low\_temp/json/with/key/YOUR\_IFTTT\_KEY
   * Replace YOUR\_IFTTT\_KEY with your actual IFTTT key. This can be found in your IFTTT profile > My services > Webhooks > Documentation.
4. **Configure ThingHTTP for High Temperature:**
   * Create another new ThingHTTP for Temperature HIGH with the URL: https://maker.ifttt.com/trigger/high\_temp/json/with/key/YOUR\_IFTTT\_KEY
5. **Save ThingHTTP Settings:**
   * Save the ThingHTTP settings.

## Step 5: IFTTT Setup

1. **Create an Applet:**
   * Log into your account and click on “Create”.
2. **Set Trigger ("If This"):**
   * Search for "Webhooks" and select the Webhooks service.
   * Choose "Receive a web request" as the trigger.
3. **Define Event Name for Low Temperature:**
   * Enter the event name as "low\_temp" and click on "Create Trigger."
4. **Set Action ("Then That"):**
   * Search for "TP-Link Tapo" and select the "Turn On" action.
   * Connect your TP-Link Tapo account, choose your device model, and create the action.
   * Click "Continue" to review the applet actions and then click "Finish."
5. **Create Applet for High Temperature:**
   * Repeat the process to create a second applet with the event name "high\_temp."
   * Select the "Turn Off" action for this applet and complete the setup.
6. **Review and Activate Applets:**
   * Review the created applets to ensure they match the project requirements.

The use of Webhooks as triggers requires a PRO plan.

## Step 6: Firebase Setup

Create a Project in Firebase following the instructions in the link provided in Reference section.

The configuration settings needed for the next step can be found by clicking on “Project Overview” > (cog Icon)1 > Project settings > General, in the section Your apps of the Project you have created.

## Step 7: Glitch Setup

In Glitch, create a new project with two files using the provided 'index.html' and 'script.js' code. Replace the 'firebaseConfig' data with the information from Firebase Project settings.