# Smart Home Monitoring System – Operation Manual

## 1. Hardware Setup

Connect the following components to Raspberry Pi Pico using a breadboard and jumper wires:

|  |  |  |
| --- | --- | --- |
| Component | Connection Pin | Function |
| Button | GPIO18 (Pin 18) | Toggle the buzzer ON/OFF manually |
| Buzzer | GPIO16 (Pin 16) | 1 beep for low light, 3 beeps if distance is less than 10cm for more than 5s |
| Ultrasonic | GPIO14 (Pin 14) | Detect object distance continuously |
| DHT11 Sensor | GPIO15 (Pin 15) | Measure temperature & humidity every 20 sec |
| Light Sensor | ADC0 (Pin 26) | Monitor ambient light every 10 sec |

## 2. System Functionality Overview

On Raspberry Pi Pico:

* Reads sensor values and applies alert logic
* Sends sensor readings and alert\_type via UART to Raspberry Pi

- Example UART output:

Distance is : 6.67135 cm  
Too close! Alert Triggered!  
Temperature is: 23 Humidity is: 25  
DHT11 data sent!  
Light Level: 16291

## 3. Gateway: Raspberry Pi 4

Runs sqlite.py to receive UART data, parse it and store into SQLite database.

Database schema:

timestamp | temperature | humidity | light | distance | alert\_type

Use the following to check data in SQLite CLI:

sqlite3 sensor\_data.db

SELECT \* FROM sensor\_data ORDER BY timestamp DESC LIMIT 10;

Sample Output:

4164|2025-05-26T15:56:00.819422|23.0|29.0|40521|6.67135|too\_close

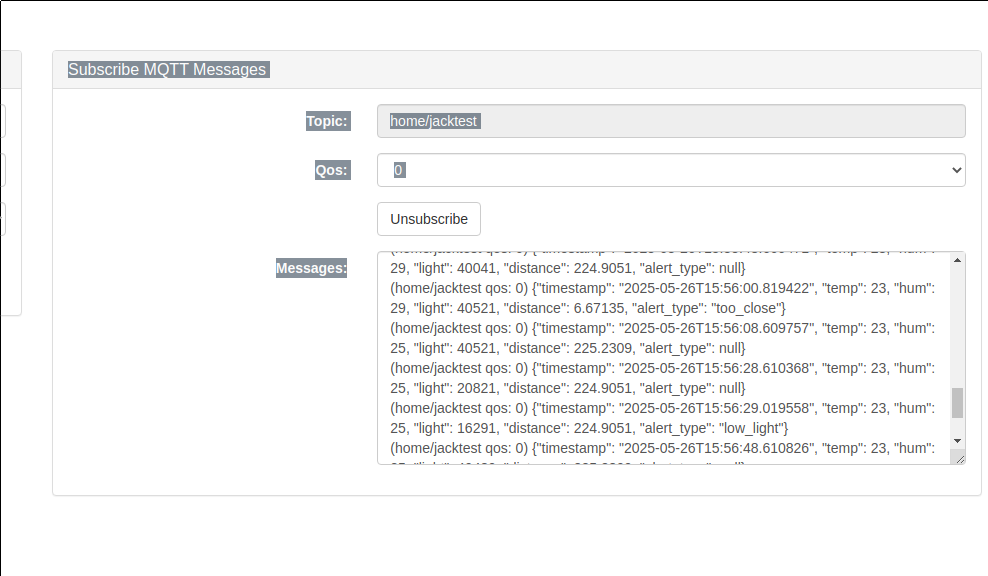
## 

## 4. Web Dashboard (Flask-MQTT)

Steps to start the dashboard:

1. Activate virtual environment： source venv/bin/activate
2. Install packages: pip install -r requirements.txt
3. Launch Flask: python app.py
4. Open browser at http://0.0.0.0:5000

Subscribe to topic: **home/jacktest** and receive messages with alert types in real-time.



## 5. Publishing MQTT Messages (Using Web Dashboard)

You can use the 'Publish MQTT Message' section on the dashboard to manually publish a message to any MQTT topic.

Steps:

1. In the 'Topic' field, enter a valid MQTT topic (e.g., home/jacktest or test/topic).

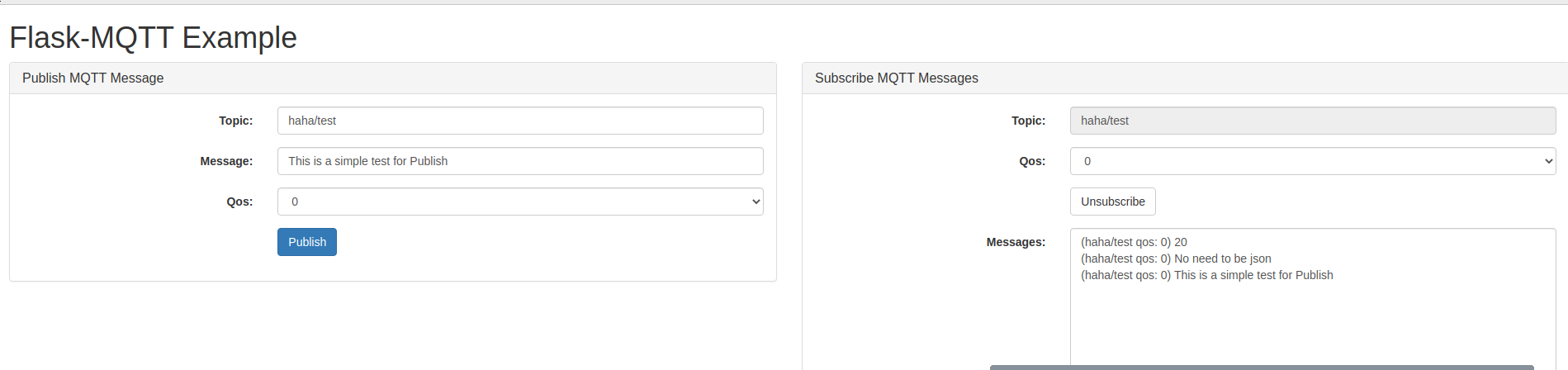
2. In the 'Message' field, enter any text or command. It does not have to be JSON unless required by your backend.( Can’t be json, just simple text will work)

3. Choose QoS level (usually 0 is fine).

4. Click 'Publish' to send the message.

If subscribed to the same topic in the right-hand panel, you will immediately see the message appear.

Example:



## 6. Email Alert Feature (Automatic Notification)

The system can automatically send email alerts when an urgent situation is detected based on sensor readings.

Trigger Condition:

If the field 'alert\_type' is NOT null (e.g., 'too\_close' or 'low\_light'), an email will be triggered.

This is implemented in the backend script (email\_alert.py) using logic like:

if alert\_type:  
 send\_alert\_email(alert\_type, distance, light, timestamp)

