Practical Internet of Things with Raspberry Pi
Project: Weather Station (WeatherDragon)

Marius-Catalin Ghiuri
Leonard Hörnlein

December 20, 2023

Contents

1 Introduction

The project entails the construction of a comprehensive weather station. This station is equipped with sensors to measure temperature, humidity, and air pressure.
The Sense HAT's LED screen displays one of the three data parameters, selectable via a joystick. Additionally, the text color on the LED screen dynamically changes based on predefined thresholds, with blue or red indicating values below or above acceptable limits, respectively, and green denoting normal conditions. In cases where at least one parameter exceeds the defined threshold for high values, a signal is visually conveyed through the blinking of an LED on the breadboard.
To monitor and analyze the data systematically, the project integrates with ThingSpeak, facilitating data uploads and providing a platform for visualization.
Furthermore, the system employs MQTT to enable remote monitoring via an app. In the event of sensed values surpassing predefined thresholds, an alarm is triggered, notifying the user through the app.

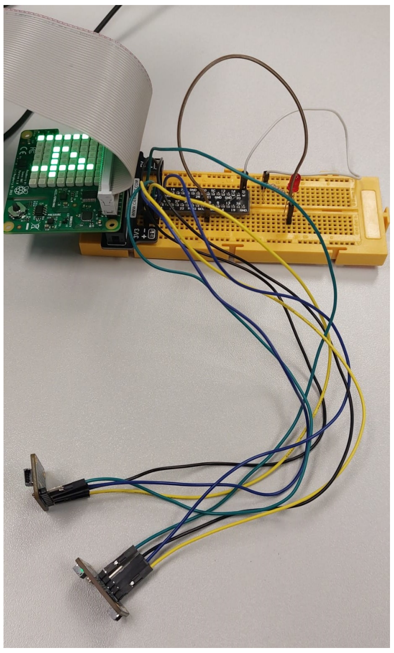


Figure 1: Hardware setup

2 Hardware

The project runs on a Raspberry Pi 400, utilizing the Sense HAT as the central hardware component. The Sense HAT serves as the primary interface for the weather station, providing essential functionalities for data display and interaction.
In terms of sensors, the project incorporates two external sensors for more accurate readings. The first is the Si7021 (Figure 2), dedicated to capturing temperature and relative humidity. The second is the BMP390 (Figure 3), designed to measure temperature and air pressure.
The decision to use external sensors, particularly for temperature, was motivated by the potential heat interference from the surrounding technology affecting the accuracy of the Sense HAT's onboard temperature sensor. Both external sensors are sourced from Adafruit, ensuring high-quality and reliable data acquisition.
Additionally, an LED with an accompanying resistor is connected to the setup on the breadboard, serving as a visual indicator for specific threshold conditions, which are explained further on.
The integration of these components on the breadboard forms the core infrastructure of the weather station system, providing a robust foundation for accurate and comprehensive environmental data monitoring.

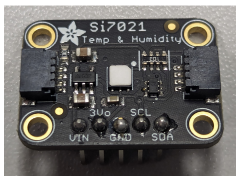


Figure 2: Si7021 sensor

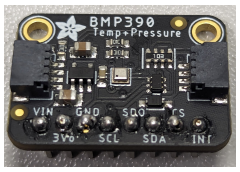


Figure 3: BMP390 sensor

2.1 Hardware Connection

In establishing the hardware connections, meticulous attention was given to ensure a robust and reliable configuration.
For both sensors, the grounding (`GND`) was seamlessly linked to a corresponding `GND` on the Raspberry Pi. The Serial Clock Line (`SCL`) and Serial Data Line (`SDA`) from each sensor found their designated counterparts on the Raspberry Pi, with both sensors sharing the same connection for streamlined communication. Furthermore, the `3Vo` (3.3V) outputs of the sensors were adeptly united with the `3Vo` on the Raspberry Pi, ensuring a consistent and stable power supply.
Concurrently, the LED was grounded to a `GND` pin on the Raspberry Pi, establishing a solid electrical reference. Its operational connection, intricately crafted, involved linking to `Pin 21`, ensuring precise control and integration with the broader system.
This meticulous arrangement, meticulously executed, forms the backbone of the hardware connections, guaranteeing optimal functionality and data integrity.

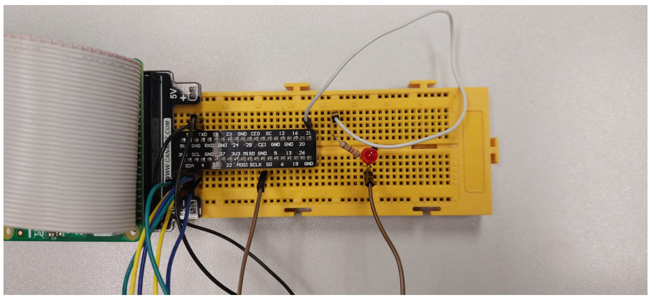


Figure 4: Hardware Connections

3 Software Implementation

The software implementation of the weather station project involves integrating various sensors, controlling the Sense HAT LED display, and communicating with external services for data storage and remote monitoring. The code is written in Python and takes advantage of libraries for sensor communication and data transmission.

3.1 General Configuration

The project uses several constants to define color thresholds and sensor limits. These include colors for the LED display (RED, GREEN, BLUE), minimum and maximum values for temperature (minTemp, maxTemp), humidity (minHum, maxHum), and pressure (minPrss, maxPrss).

```
# General Configuration
level = 0

RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)

minTemp = 15
maxTemp = 30
minHum = 30
maxHum = 60
minPrss = 900
maxPrss = 1000
```

3.2 LED Control and Functions

The code initializes the GPIO pin for the LED and defines functions for turning the LED on (`ledOn`), off (`ledOff`), and blinking (`ledBlink`). These functions are utilized later in the script to provide visual indications based on sensor readings.

```
# LED Control and Functions
LED_PIN = 21
GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)

def ledOn():
    GPIO.output(LED_PIN, GPIO.HIGH)

def ledOff():
    GPIO.output(LED_PIN, GPIO.LOW)

def ledBlink():
    ledOn()
    sleep(0.1)
    ledOff()
```

3.3 Sensor Configuration

The code initializes and configures two sensors: the Si7021 for temperature and humidity, and the BMP3XX for temperature and pressure. The Adafruit CircuitPython libraries for each sensor are utilized for communication.

```
# Sensor Configuration
si = adafruit_si7021.SI7021(board.I2C())
bmp = adafruit_bmp3xx.BMP3XX_I2C(board.I2C())
```

3.4 ThingSpeak Configuration

The script defines parameters for ThingSpeak integration, including the ThingSpeak API key (`key`) and headers for data transmission. It also creates an HTTP connection to the ThingSpeak server.

```
# ThingSpeak Configuration
key = "<API KEY>"
headers = {"Content-type": "application/x-www-form-urlencoded"}
conn = http.client.HTTPConnection("api.thingspeak.com:80")
```

3.5 MQTT Configuration

MQTT (Message Queuing Telemetry Transport) is configured with the broker address, and the script creates an MQTT client for publishing temperature, humidity, and pressure data.

```
# MQTT Configuration
broker_address = "broker.emqx.io"
client = mqtt.Client()
client.connect(broker_address, 1883, 60)
client.loop_start()
```

3.6 Main Loop

The main loop continuously reads sensor data, updates the LED display on the Sense HAT, uploads data to ThingSpeak, and publishes data to MQTT topics for remote monitoring.

```
# Main Loop
while True:
    # ... (Sensor readings, ThingSpeak upload, Sense HAT display, MQTT publishing)
```

3.6.1 Sensor Data Reading

Sensor data, including temperature (`temp`), humidity (`hum`), and pressure (`prss`), is continuously acquired from the Si7021 and BMP3XX sensors. For the temperature, we calculate the mean of both sensors to get an better result.

```
temp = round((round(bmp.temperature, 1) + round(si.temperature, 1)) / 2, 1)
hum = round(si.relative_humidity, 1)
prss = round(bmp.pressure, 1)
```

3.6.2 ThingSpeak Data Upload

The acquired sensor data is then uploaded to ThingSpeak using HTTP POST requests. The data, along with the ThingSpeak API key, is formatted and sent to the ThingSpeak server.

```
params = urllib.parse.urlencode({
    'field1': temp,
    'field2': hum,
    'field3': prss,
    'key': key
})

conn.request("POST", "/update", params, headers)
response = conn.getresponse()
```

3.6.3 Sense HAT Display

The Sense HAT LED display is updated based on the sensor readings. Color-coded messages are displayed to indicate the status of temperature, humidity, and pressure.

```
# Sense HAT Display
if hum > maxHum:
    hum_color = RED
    ledBlink()
elif hum > minHum:
    hum_color = GREEN
else:
    hum_color = BLUE

# ... (Similar logic for temperature and pressure)

# Sense HAT message display based on the selected data parameter
for event in sense.stick.get_events():
    if event.direction == "up" and event.action == "pressed":
        level = (level + 1) % 3
    # ... (Additional logic for joystick input and message display)
```

3.6.4 MQTT Publishing

The acquired sensor data is published to MQTT topics for remote monitoring. Topics include "/temperature," "/humidity," and "/pressure."

```
# MQTT Publishing
client.publish("/temperature", temp)
client.publish("/humidity", hum)
client.publish("/pressure", prss)
```

4 Results

As a user, data visualization and monitoring options are provided through multiple channels:

4.1 Sense HAT and Breadboard

The Sense HAT display presents real-time environmental data, focusing on one of the three parameters (temperature, humidity, or pressure) at a time. Users can navigate between these parameters using the joystick (up and down). The text color on the display dynamically changes based on predefined thresholds: green signifies values within the normal range, blue indicates values below, and red indicates values above acceptable limits. In case a parameter exceeds the defined upper threshold, the LED blinks as a warning.

4.2 IoT MQTT Panel App

For a convenient and real-time overview of the current weather conditions, users can utilize the IoT MQTT Panel app (see Figure 5). The app displays the latest temperature, humidity, and pressure values, providing a mobile interface for on-the-go monitoring.

4.3 MQTT Alert App

The MQTT Alert app (see Figure 6) offers customization for threshold-based alerts. Users can set alarms to trigger when sensor values surpass predefined limits. This feature enhances proactive monitoring, allowing users to receive immediate notifications on their mobile devices in case of unfavorable environmental conditions.

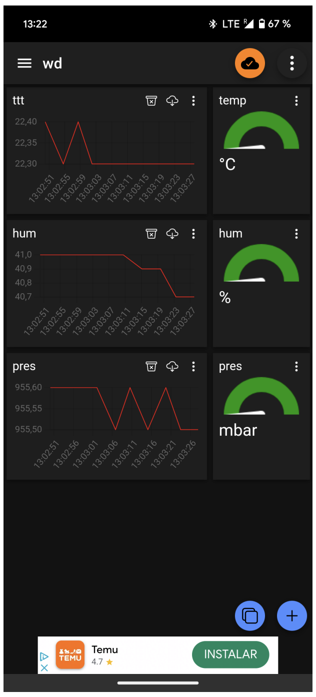
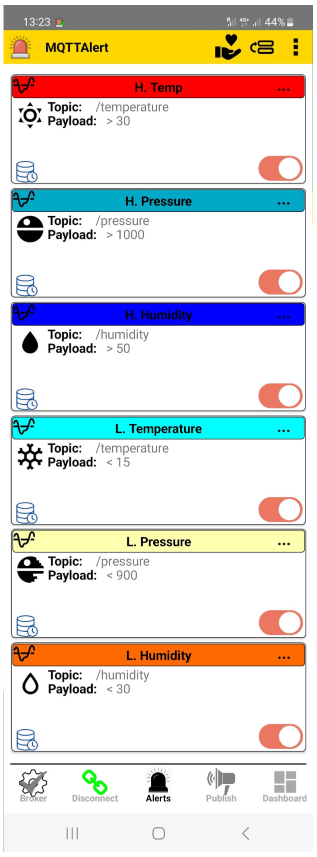


Figure 5: IoT MQTT Panel App



H. Temp
Topic: /temperature
Payload: > 30

H. Pressure
Topic: /pressure
Payload: > 1000

H. Humidity
Topic: /humidity
Payload: > 50

L. Temperature
Topic: /temperature
Payload: < 15

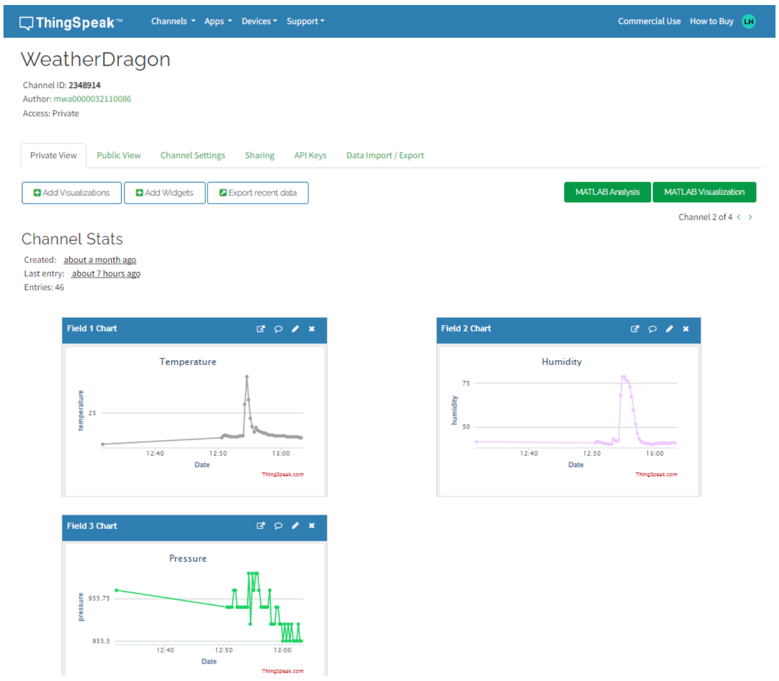
L. Pressure
Topic: /pressure
Payload: < 900

L. Humidity
Topic: /humidity
Payload: < 30

Figure 6: MQTT Alert App

4.4 ThingSpeak Online Platform

For long-term data tracking and analysis, users can access the ThingSpeak online platform (see Figure 7). ThingSpeak provides a comprehensive view of historical data, enabling users to observe trends and patterns over time.



WeatherDragon

Channel ID: 2348914
Author: mwa0000032110086
Access: Private

Channel Stats

Created: about a month ago
Last entry: about 7 hours ago
Entries: 46

Figure 7: ThingSpeak Online Platform