Student IoT Project Guide

Sending Sensor Data to Your Live Dashboard

univtool

Contents

1 Introduction

This guide will help you set up your Raspberry Pi to read sensors and send the data to a live webpage that you can share with anyone. You will learn:

- How to obtain the necessary credentials from your admin.
- How to configure your Raspberry Pi.
- How to write a Python script that reads sensors and pushes data to the cloud.
- How to view and interact with your data on a custom dashboard.

2 What You Will Need

- A Raspberry Pi (any model with Wi-Fi) with Raspberry Pi OS installed.
- Sensors (e.g., DHT22 for temperature/humidity, soil moisture sensor, ultrasonic distance sensor).
- A USB power supply and microSD card.
- Internet connection.
- A Google account (for accessing the dashboard – provided by admin).

3 Obtaining Credentials from Your Admin

Your admin will give you two important pieces of information:

3.1 Firebase Web Configuration

This is a JavaScript object that looks like this:

```
const firebaseConfig = {
    apiKey: "AIzaSy...",
    authDomain: "your-project.firebaseapp.com",
    databaseURL: "https://your-project-default-rtdb.firebaseio.com",
    projectId: "your-project",
    storageBucket: "your-project.appspot.com",
    messagingSenderId: "...",
    appId: "..."
};
```

You will need these values later to build your web dashboard.

3.2 Service Account Key (JSON file)

This is a secret file (e.g., `project-xxxx-firebase-adminsdk.json`) that allows your Raspberry Pi to write data securely. Save this file – you will upload it to your Pi.

3.3 Your Project ID

You will also receive a unique project ID (e.g., `student123`). This identifies your project in the database.

4 Setting Up Your Raspberry Pi

4.1 Update and Install Python

Open a terminal on your Pi and run:

```
sudo apt update
sudo apt upgrade -y
sudo apt install python3 python3-pip -y
```

4.2 Install Required Python Libraries

```
pip3 install firebase-admin
pip3 install RPi.GPIO
pip3 install dht11          # if you use DHT11/22
```

(Install any other sensor libraries you need.)

4.3 Place the Service Account Key

Transfer the JSON key file you received from your admin to your Pi. For example, put it in your home folder:

```
/home/pi/firebase-key.json
```

Make sure the file path is correct – you will use it in your Python script.

5 Creating the Python Sensor Script

Create a new Python file, e.g., `sensor_push.py`, and copy the template below. Replace the placeholders with your actual project ID and the path to your JSON key.

```
import firebase_admin
from firebase_admin import credentials, db
import time
import random

# ---------- CONFIGURATION ----------
# Path to your service account key
cred = credentials.Certificate("/home/pi/firebase-key.json")

# Your Firebase database URL (ask admin)
database_url = "https://your-project-default-rtdb.firebaseio.com/"

# Your unique project ID (provided by admin)
PROJECT_ID = "student123"
# -----------------------------------

firebase_admin.initialize_app(cred, {
    'databaseURL': database_url
})

def read_sensors():
    """Replace this with actual sensor readings."""
    temp = round(random.uniform(20, 30), 1)        # simulated
        temperature
    humid = round(random.uniform(40, 60), 1)       # simulated humidity
```

```
    moisture = round(random.uniform(200, 800), 1)  # simulated soil
        moisture
    distance = round(random.uniform(10, 50), 1)    # simulated distance
        (cm)
    return {
        'temperature': temp,
        'humidity': humid,
        'soil_moisture': moisture,
        'distance': distance
    }

while True:
    data = read_sensors()
    data['timestamp'] = int(time.time() * 1000)   # milliseconds

    # Push data to Firebase under your project's "data" node
    ref = db.reference(f'projects/{PROJECT_ID}/data')
    ref.push(data)

    print("Sent:", data)
    time.sleep(5)   # send every 5 seconds
```

5.1 Running the Script

Run the script:

```
python3 sensor_push.py
```

You should see output showing that data is being sent.

6 Building Your Live Dashboard (HTML/JavaScript)

Your admin will provide a webpage where you can log in and see your data. Below is a sample HTML file that you can customise. It includes:

- Real-time updates from Firebase.
- A line chart for temperature and humidity.
- A simple grid map showing a moving point based on the distance sensor.
- A table of recent data with auto-append.
- Buttons to stop/start auto-refresh and to reload manually.
- A link to view historical data (more records).

6.1 How to Use the Sample Page

1. Save the HTML code (provided below) as `index.html` in a folder on your computer. 2. Replace the `firebaseConfig` object with the one you received from your admin. 3. Replace `PROJECT_ID` with your own project ID. 4. Upload the file to your web hosting (or open it locally – it will still work because Firebase is cloud-based).

The page will automatically detect whether you are logged in. If not, it shows a login form. After login, your data appears.

7 Understanding the Dashboard Features

- **Live Chart:** Plots the last 20 temperature and humidity readings.
- **Map:** A 400×400 canvas with 5 horizontal lines. A blue dot moves left/right based on the distance sensor value (mapped to 0–400 pixels). The lines help visualise movement.
- **Data Table:** Shows the most recent 10 readings. New entries are appended at the top.
- **Stop/Start Updates:** Click "Stop Updates" to freeze the display; "Start Updates" resumes listening.
- **Refresh Now:** Manually reloads the latest data.
- **Historical Data:** Click the link to view older records (you can extend this with a separate page or modal).
- **Logout:** Ends your session.

8 Troubleshooting

- **"Module not found" errors:** Make sure you installed all Python packages with `pip3`.
- **Firebase permission errors:** Check that your service account key is valid and that the database rules allow writes from your UID.
- **No data on the dashboard:** Open the browser's developer console (F12) to see if there are JavaScript errors. Verify that your `firebaseConfig` is correct.
- **Login fails:** Ensure you have created an account using the "Create Account" button on the login page. Use the same email/password every time.

9 Next Steps

Now you can expand your project by adding more sensors, improving the visualisation, or even controlling actuators from the webpage. Experiment and have fun!