



UPPSALA UNIVERSITET

WEATHER STATION

Project Report

Group 11

Anwar Chowdhury
Lingyan Duan
Nithesh Chandher Karthikeyan

November 1, 2019

Abstract

The goal of this project is to build a weather station with real time notifications for climatology monitoring, interactive it with a cloud platform and analyse weather parameters. It predicts and displays rainbows in the SenseHat LED Matrix. Also, it shows the values of pressure, temperature and humidity parameter in the SenseHat LED Matrix. The data collected from the sensors will then be sent to online cloud server called **Google Firebase** as it provides real time database, where the values of temperature, humidity, pressure and weather is stored along with timestamp in real time. Weather information is notified to the users using an app called **Pushover notification**.

1 Introduction

Weather is an important aspect of life as it can affect the day to day activities of the people. For example, it can be difficult for people to travel from one place to another while raining. The main objective of this project is to create an online weather prediction system which enables the user to check real time weather parameters at any place.

For the back-end database system, Google Firebase is chosen as it provides Real Time Database. It can be used to analyze data and to predict future weather. Also, user receives a push notification on their mobile devices from the app "Pushover notification" regarding on the weather condition.

2 Related work

The SenseHat sensors on our raspberry PI can give information about environmental conditions on a more local level and very easy to fetch these data. This prototype can be useful for local people as they can schedule their travel based on the climatic condition. Also, Airlines companies need to know about the local weather conditions to plan a schedule for the flights. Weather prediction project can be more helpful for farmers so that they can harvest the crops before severe weather conditions.

3 Research problem

If the raspberry pi hardware gets heated, it will give incorrect temperature reading which will affect the weather prediction. Also, it is difficult to predict a rainbow in real life. The rainbow gets displayed in the LED Matrix only when humidity value is greater than 80 RH and temperature value is greater than 20 °C. Even if the humidity and temperature values match the theoretical

condition, rainbow can't be guaranteed to appear. Also, updating on firebase is impossible when raspberry pi didn't connect to the Internet.

4 Theory and Method

The project predicts weather and displays the temperature, humidity and pressure readings on the LED matrix. This is done by Pressure sensor, temperature sensor and humidity sensor inside Sense HAT. The sensors gather the data from the environment and the Raspberry Pi transmits the data to Firebase through Internet where the data is stored and can be used for the future. Also, we log the timestamp for collected data.

Besides, users can get weather information by installing an app called "pushover notification" and register their device. This app is not free but has 7 days free trail which is enough to test our project. You can download this app from App Store and Google play store.

4.1 Project Structure

Figure 1 shows the procedure of this project. Pressure, temperature and humidity data are collected and sent to Google Firebase. On Firebase, data can be easily send and retrieve in Real Time. The outcome of weather information can be sent as a notification message to mobile device and it is also displayed on SenseHat LED Matrix.

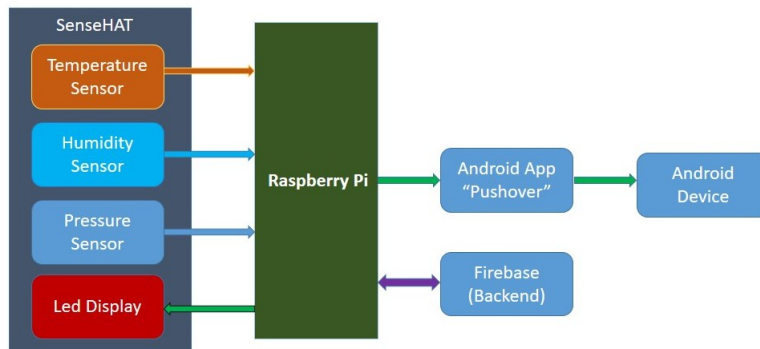


Figure 1: Procedure of Weather Station

4.2 Environment

- Use python to implement our project, there are many helpful libraries to use.

- Configure SSH connection to reach the server.
- Set up a Google Firebase project to store our collected data.
- Download app "Pushover notification" and register the device to receive weather information.

5 Implementation

5.1 Setting Up Firebase

Firebase is an online cloud platform developed by Google. In order to create a project on Firebase, go to Firebase Console and select to add project. Give a fancy title to the project (in this case we have given "RainbowSweden") and set analytics location as the host country (Sweden) and continue by adding the project to Firebase. Now the project is created in the Firebase. Add a real time database (Go to Develop - Database - Real Time Database). The figure below shows the Real Time Database that has been set up for the project "RainbowSweden".

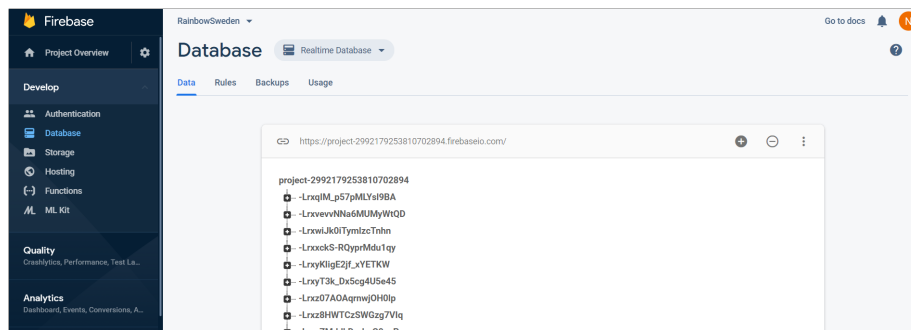


Figure 2: Real Time Database for RainbowSweden

The database can be updated with the http link given above the database which is shown in Figure 2. Besides, the rules of the database has to be changed so that everyone can access the database. Set the rules as Listing 1.

Listing 1: Rules to access the database

```

1 {
2   "rules": {
3     ".read": true,
4     ".write": true
5   }
6 }
```

5.2 Store and Retrieve Data in Firebase

Since the language used in the project is python, a package called "Firebase" needs to be installed in order to store and retrieve data. Use the following command as Listing 2 to install Firebase package.

Listing 2: Command to Install Firebase Package

```
1 $ sudo pip install requests==1.1.0
2 $ sudo pip install python-firebase
```

After installation of Firebase, the data can be stored and accessed using the library Firebase in Python.

Listing 3: data connection

```
1 from firebase import firebase
2 import time
3 data= {
4     'Temp' : temp,
5     'Humidity' : humidity,
6     'Pressure' : pressure,
7     'Weather' : weather,
8     'TimeStamp' : now
9 }
10 firebase = firebase.FirebaseApplication('https://project-2992179253810702894.
11     firebase.com/')
12 firebase.post('https://project-2992179253810702894.firebaseio.com/',data)
```

The data structure contains the values of temperature, humidity, pressure, weather and timestamp. An object called "firebase" is created and used to establish connection with Real Time Database using the function `FirebaseApplication()`. The function accepts the http link that is shown in Figure 2 to establish the connection with the Firebase. Values in data is sent to the database using the function `post()`. The way how data is stored in Firebase is shown as Figure 3.

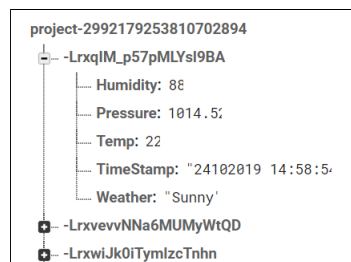


Figure 3: Data structure on Firebase

Similarly, the data retrieval is done using the function `get()`. Listing 4 shows the code snippet that weather data are retrieved from the database whenever there is a possibility of Rainbow.

Listing 4: Retrieving data when rainbow appears

```

1 i=0
2 climate = 'Rainbow'
3 print('Weather Report for Rainbow')
4 getweather = firebase.get('https://project-2992179253810702894.firebaseio.com
  /', None)
5
6 try:
7     for i in getweather:
8         if(climate in getweather[i]['Weather']):
9             result1 = getweather[i]['Temp']
10            result2 = getweather[i]['Humidity']
11            result3 = getweather[i]['TimeStamp']
12            print('Timestamp:%s' %result3)
13            print('Temperature:%s C' %result1)
14            print('Humidity:%s RH' %result2)
15 except KeyError:
16     print(' ')

```

5.3 Getting Data from SenseHAT

Python uses a library for SenseHAT to communicate with it. An object ap is assigned to a function called SenseHAT() and the data are collected as Listing 5.

Listing 5: Data collection

```

1 from sense_hat import SenseHat
2 import time
3 ap = SenseHat()
4 temp = ap.get_temperature()
5 humidity = ap.get_humidity()
6 pressure = ap.get_pressure()

```

5.4 Displaying Weather On LED matrix

There are few conditions that the program checks to predict the weather. These conditions are showing in Figure 4.

| Condition | Humidity | Temperature | Weather |
|-----------|----------|-------------|---------|
| 1 | > 80 | > 20 °C | Rainbow |
| 2 | > 80 | < 0 °C | Snow |
| 3 | <= 80 | > 20 °C | Sunny |

Figure 4: Weather classification

Whenever a condition is satisfied, a rainbow or sunshine or snowing gets displayed on the LED Matrix using the function set_pixels().

5.5 Push notification to mobile devices

Firstly download the app "pushover notification" to your mobile phone and register an account. After registration, an user key is generated as Figure 5. This key can identify your device and use this key to push message to certain mobile phone.

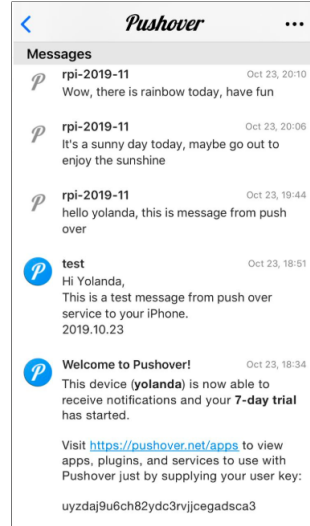


Figure 5: Weather classification

Figure 5 shows that weather information can be pushed successfully to a mobile device. Project name for our weather station is rpi-2019-11 and there are 3 messages appeared. A test message and two weather information messages regarding on sunny and rainbow.

6 Result

The following results were achieved in this project.

- A. The weather sensor collects all the weather data (ie., temperature, humidity and atmospheric pressure). The data can be collected using senseHAT functions.
- B. The weather data is sent to the Firebase, which is a NoSQL database service and data stored as JSON objects. The data is retrieved whenever there is a possibility of Rainbow.
- C. If the weather satisfies the conditions for a rainbow, it gets displayed in the LED Matrix. Also it displays sun whenever the respective condition is satisfied. The conditions are listed in Figure 4.

D. Real-time weather condition will be sent to the user via the app called pushover notification on mobile device. This allows the user to prepare themselves for their future plans.

7 Conclusion

An adaptive weather station and rainbow predictor is implemented using Raspberry Pi 3 and SenseHAT. The project also alerts the users regarding the weather outside using an app called "Pushover". The weather data is updated in an online cloud server, where the data can be accessed in Real-time. Also, timestamp and other values are retrieved whenever weather satisfies the condition for Rainbow. The colors of rainbow is displayed in the LED Matrix whenever there is a possibility of rainbow. The data can't be updated in the server, if the Raspberry Pi 3 is not connected to the Internet

8 Future Work

Due to limitation of time, the following features can be developed in the future.

- Graphical view of weather history and show it on website page or an native app.
- Adding Music using OMX player based on the weather.
- Collect more climate parameters and classify more weather condition like rainy, windy, cloudy etc.

References

- [1] Rainbow Predictor, <https://projects.raspberrypi.org/en/projects/rainbow-predictor>
- [2] The Budding Inventor Within You With These Top 50 Raspberry Pi Project Ideas, <https://www.technorms.com/68238/best-raspberry-pi-project-ideas>
- [3] Installation of Firebase in Python, <https://ozgur.github.io/python-firebase/>
- [4] Sensor HAT API, <https://pythonhosted.org/sense-hat/api/>