IoT Weather Monitoring System

Hozefa Lakadawala
Electrical and Computer Engineering
hlakada1@binghamton.edu
State University of New York at Binghamton
Binghamton, New York, USA

ABSTRACT

Weather is a day-to-day state of atmosphere that is hard to predict which can affect even performing simple human activities. The aim of this project is to design a weather station with real time data monitoring using Internet of Things where we could view the data from any remote location. In this project, a weather station is assembled using SparkFun Photon Weather Shield Sensor and Raspberry Pi 4 Model B to collect weather parameters like temperature and pressure. Data collected from the sensors are then stored into a local SQLite database using Raspberry Pi. A webserver is developed using Flask framework to display the real-time data which we get from the SQLite database.

ACM Reference Format:
Hozefa Lakadawala. 2020. IoT Weather Monitoring System. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Weather Condition plays an important role in our daily life as it changes dynamically, which makes it important for us to get out of the house only after checking weather conditions through temperature, humidity, pressure. Also, integrating Internet of Things in the equation makes it easier to collect and watch the data in real-time from anywhere just by using internet.The data collected can also be used to predict future values.

The goal of this project is to create an online weather system through which a user can check parameters like temperature and pressure in real-time from anywhere only with the help of internet.

2 DESIGN

The design for the project consists of four parts:

- `Part A`: First, we connect the Photon Weather Shield Sensor to our Raspberry Pi 4 by using the GPIO pins and providing 3.3V of input to the sensor and taking the output through the SDA and SCL pins.
- `Part B`: Then, we program the Raspberry Pi to read the data from the MPL3115A2 sensor which can sense pressure, altitude and temperature.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

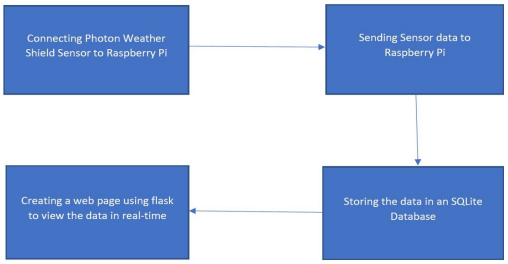

Figure 1: Final Project Design

Figure 2: Raspberry Pi 4 connected with Sparkfun Photon Weather Shield

- `Part C`: Once we get the data, we create a SQLite database to locally store the data inside a table.
- `Part D`: The last part consists of creating a basic flask webserver along with a basic html and css file to view the real-time data.

2.1 Hardware

The hardware consist of Rapsberry Pi 4, Sparkfun Photon Weather Shield, monitor, keyboard, mouse and 4 M-F cables to connect the sensor with raspberry pi. The final hardware of the project is shown above in figure 2:

2.2 Software

On the software side, we first install raspbian OS for our Raspbrerry Pi, then we install all the basic libraries needed to run the project which are i2ctools, smbus, sqlite3, python-flask.
Once the libraries are installed, we program the raspberry pi so that we can see read the sensor readings on the terminal.

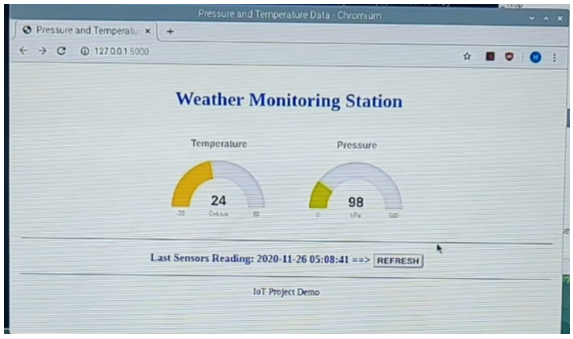

Figure 4: Final Result

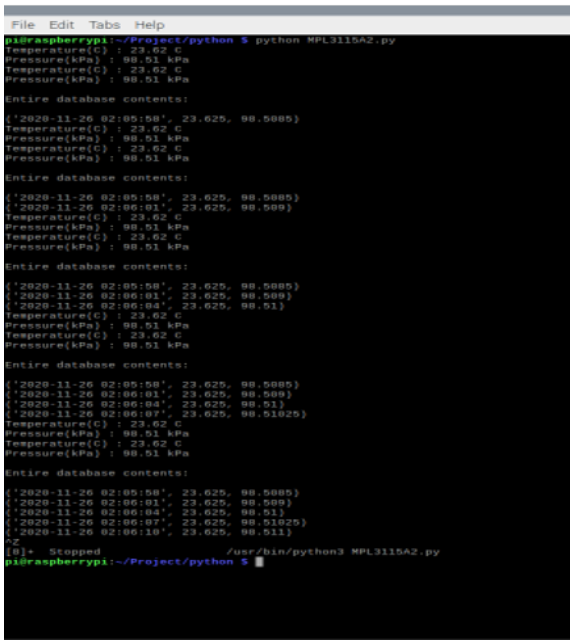

Figure 3: Filling Database with sensor data every 3 secoond

After getting values on the terminal, we create a sqlite3 database and a table inside it where we can store all the values from the sensor.
And, finally we created a flask web-server in a different script where we also created and added a html and css file to present the data in a clean manner.

3 IMPLEMENTATION

Once we wrote the script for reading, collecting, storing and viewing the data we run the code to check for results. When we run the code for the sensor file named MPL3115A2.py , it gives us sensor data as temperature and pressure and stores every new value into the table that we create inside the sqlite database.
Another file named appptWebServer.py is for the flask web-server which gets the value from the database and make is visible on the web-server.
As we can see from figure 3, the sqlite is storing new values periodically every 3 seconds.
And from figure 4 we can clearly see the values of temperature and pressure along with a refresh button which can udpdate the latest value recorded.

4 CONCLUSION

We can conclude that our goal of viewing the data on a web-server was achieved. For further improvements we can add more weather affecting values like humidity, wind speed, light intensity, etc. Also, we can include graph plots on the web-server to visualize the data more better.

5 ACKNOWLEDGMENTS

I would like to thank Prof. Mo Sha and TA Junyang Shi for helping us through the CS588 class. They helped me with all my doubts for the subject.