Temperature and Humidity Monitoring System

Joe Doe, John Doe, Politehnica University of Timisoara June, 2018

The purpose of this document is to provide a comprehensive documentation template. Some sections from the overall information might be dummy but it helps to emulate specific concerns. Paragraphs formatted using the italic style provide in-place advice. They should not be contained in your project documentation. Provided LATEX source file should be used as base format to write your documentation.

1 Repository

Related files to your work must be placed within a version control system. I strongly suggest to use Github but any git system should just be fine.

Schematics, diagrams and codebase are contained under the following git repository:
`https://github.com/claudiugroza/msa`

2 User requirements

Imagine talking to a person without any technical background. User requirements should express the intentions of a common user and not how you plan to engineer the system.

1. The system must provide information relevant to temperature and humidity in a certain area.
2. The system should be open for extension, eg. adding a barometer.
3. The system should run in an environment that provides a 24/24 access.
4. Temperature and humidity information should be accessed via a Web interface or a mobile application.
5. The system must provide the current temperature and humidity. The maximum latency should be 10 seconds.
6. The system might provide access to history data for the last 2 weeks.
7. The system may provide a module for data interpretation.

3 System overview

Illustrate how you engineered the system from a generic point of view. You should point out all the important subsystems, modules or entities.

The overview of the system is depicted in Figure 1.

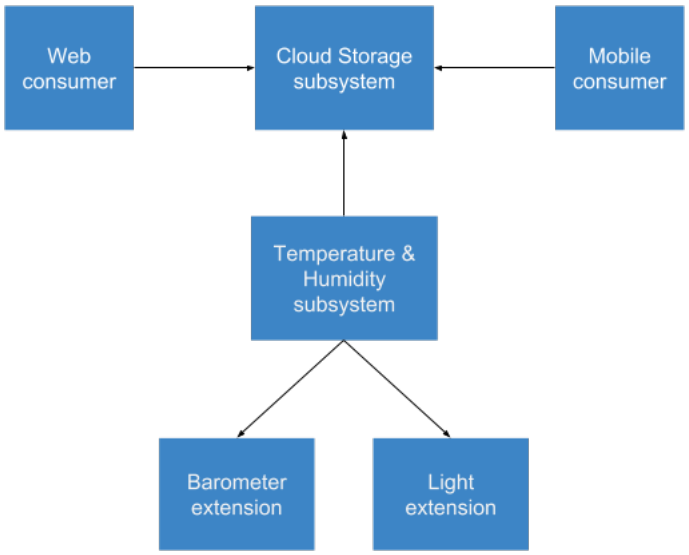

Figure 1: System overview diagram

Base (Temperature & Humidity) Subsystem encompasses the measurement functionality. It has the sole purpose to acquire information from it's sensors and extensions.

Barometer extension provides an interface for pressure querying. It must conform to an interface defined by the Base Subsystem.

Light extension provides an interface for light related data. It must implement the interface defined by the Base Subsystem.

Cloud Storage Subsystem stores the data pushed by the Base Subsystem. Additionally, it offers a possibility to interpret stored information.

Web consumer provides a UI interpretation for the data stored in the Cloud Storage Subsystem. This view is accessed within a Web browser.

Mobile consumer also provides a UI interpretation for the stored data. The view will be accessed via a specialised application which runs on a mobile device.

4 Circuit design

Provide a detailed perspective of the hardware components you've used in your project. You can use Fritzing tool to draw the schematics.

The hardware view of the system is depicted in Figure 2.

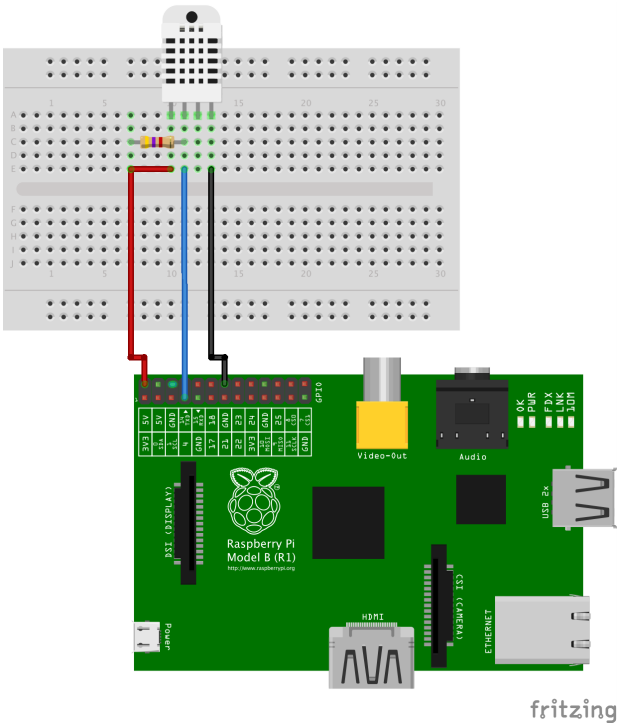

Figure 2: Circuit schematic

Raspberry Pi 2 provides support for quick prototyping. That makes it a perfect choice for quick prototyping but not adequate for real-time applications. We will use the one-wire interface it has, but also the implicit possibility of communicating with other devices over the Internet.

AM2302 (DHT-22) is an one-wire enabled sensor. It is a basic, digital-output relative temperature and humidity sensor.

The wiring of the components can be observed in Figure 2. An important aspect we need to mention is that the communication between the board and sensor is done via the one-wire interface.

5 Software design

Provide a walktrough of the most important components/modules/entities or concepts you've implemented in the software section.

The software components and data flow directions are depicted in Figure 3. Each of these will be presented in the following subsections.

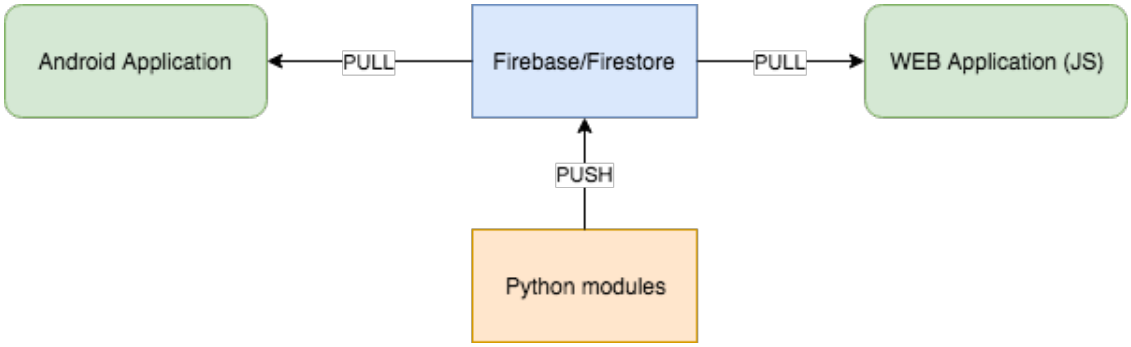

Figure 3: Software entities involved

5.1 Python modules

You should not concentrate on providing line level descriptions, but rather class/script/module level explanations. Short code comments are strongly advised. Third party libraries should also have a brief description and a reference link.

write_firebase_sensor.py: it retrieves the temperature and humidity tuple from DHT-22 sensor over the one-wire interface and then pushes it to Firebase every 1 second.

Adafruit_DHT22 library: it provides a quick implementation of one-wire interface communication with the sensor.

5.2 Firebase

Describe the purpose of the service and present the specifics you are using.

Firebase is a PaaS (Platform as a Service) which means it offers developers to a quick list of functionalities supported by a traditional backend. Realtime Database simplifies storing and synchronising data between different devices in realtime using a noSQL database.

The following code section shows the initialization of the Firebase project.

```
# the generated root for your project
FIREBASE_ROOT = 'https://ms-iot.firebaseio.com'
# init Firebase Database instance
firebase = firebase.FirebaseApplication(FIREBASE_ROOT, None)
```

5.3 Android application

Explain the responsibility of some key entities contained in your Android project.

5.4 Android application

Give a brief description of the main parts contained by the JavaScript module.

6 Results and further work

Describe what you accomplished so far, then what you plan to improve or extend.

The current version of the project supports the following functionalities:

- reliable reading of the temperature and humidity
- storing data to Firebase Database
- client implementations for retrieving data stored in Firebase Database (Android and WEB)

The following list of extensions and improvements was identified to be supported in the future:

- remote access for configuring temperature and humidity interval reading
- extend Android app functionality to display retrieved information in real-time diagrams
- extend JavaScript application to support .csv exports
- improve the hierarchy design of Firebase Database

7 References

1. Draw IO [last seen: May 2018], https://www.draw.io/
2. Fritzing [last seen: May 2018], http://fritzing.org/
3. Firebase Database [last seen: May 2018], https://firebase.google.com/docs/database/
4. DHT22Datasheet [last seen: May 2018], https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf
5. AdafruitDHT library [last seen: May 2018], https://github.com/adafruit/Adafruit_Python_DHT