



College of Engineering

# CS CAPSTONE DESIGN DOCUMENT

DECEMBER 1, 2017

## PHAZE — IoT WEATHER STATION

PREPARED FOR

### OPENLY PUBLISHED ENVIRONMENTAL SENSING LAB

CHET UDELL

\_\_\_\_\_  
*Signature*

\_\_\_\_\_  
*Date*

PREPARED BY

### GROUP 53 TEAM PHAZE

ALESSANDRO LIM

\_\_\_\_\_  
*Signature*

\_\_\_\_\_  
*Date*

TOUKU CHA

\_\_\_\_\_  
*Signature*

\_\_\_\_\_  
*Date*

JAYDEEP ROTITHOR

\_\_\_\_\_  
*Signature*

\_\_\_\_\_  
*Date*

#### Abstract

This document discusses the technologies used to complete the IoT weather station project. Detailed in this paper are technologies from multiple viewpoints: context, composition, dependency, information, interface, interaction, state dynamics, and resources. Each viewpoint will summarize the technologies used and how they impact our project. The technologies we discuss in this document are: microcontrollers, sensors, charging method, data storage, data acquisition, Geographic Information System (GIS), Internet of Things (IoT), and web server.

## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Scope . . . . .	2
1.3	Glossary . . . . .	2
<b>2</b>	<b>Context</b>	<b>2</b>
2.1	Internet of Things . . . . .	2
2.2	Web Development . . . . .	2
<b>3</b>	<b>Composition</b>	<b>3</b>
3.1	Microcontroller . . . . .	3
3.2	Sensor . . . . .	3
3.3	Charging Method . . . . .	3
<b>4</b>	<b>Dependency</b>	<b>3</b>
4.1	Internet of Things . . . . .	3
4.2	Web Server . . . . .	3
<b>5</b>	<b>Information</b>	<b>3</b>
5.1	Data Storage . . . . .	3
5.2	Geographic Information System . . . . .	4
<b>6</b>	<b>Interface</b>	<b>4</b>
6.1	Internet of Things . . . . .	4
6.2	Web Server . . . . .	4
<b>7</b>	<b>Interaction</b>	<b>4</b>
7.1	Microcontroller . . . . .	4
7.2	Sensor . . . . .	4
7.3	Data acquisition . . . . .	4
7.4	Internet of Things . . . . .	4
7.5	Web Server . . . . .	4
<b>8</b>	<b>State Dynamics</b>	<b>5</b>
8.1	Microcontroller . . . . .	5
8.2	Sensor . . . . .	5
8.3	Charging method . . . . .	5
8.4	Geographic Information System . . . . .	5
<b>9</b>	<b>Resource</b>	<b>5</b>
9.1	Data Storage . . . . .	5

	2
10 Conclusion	5
References	7

## 1 INTRODUCTION

### 1.1 Purpose

This document describes the system design and technologies used for the Evaporometer IoT weather station project (PHAZE). We discuss the system design and technologies through eight viewpoints: context, composition, dependency, information, interface, interaction, state dynamics, and resource. Each viewpoint will summarize why we choose the technology and how it supports the PHAZE devices.

### 1.2 Scope

This document covers the part of the Evaporometer project that involves the creation of 20 weather station devices, the methods of transmission of data from the devices to remote weather hubs, and the plotting of this data on web maps.

### 1.3 Glossary

- API - an application programming interface
- Address - a reference to a specific memory location
- Ethernet - networking technology used in local area networks
- GIS - Geographic Information System
- IoT - Internet of Things
- LoRa - Long range, low power wireless platform
- Microcontroller - A small computer
- Pushing Box - Software that manages notifications for Internet of Things devices
- Radio - a device that can transmit and receive data wirelessly
- Sensor - a piece of equipment used to measure data

## 2 CONTEXT

### 2.1 Internet of Things

The core concept of our project is the implementation of Internet of Things, all pieces of our project interact with each other over the Internet. This allows us to deploy devices that can manage themselves without human interaction. Our systems uses LoRa radios to transmit data wirelessly [1] along with an ethernet hub to upload our data efficiently.

### 2.2 Web Development

In order to create our web map that visualizes our collected data, we will use web development. This will allow us to create a web page in order to have a place to put our data map. We will also process the data coming in from the server and appropriately plot this data onto the map. To give the user more flexibility in how to view the map, we will need to add interactive features to the map, such as zooming and the ability to move the center of the map.

## 3 COMPOSITION

### 3.1 Microcontroller

The purpose of the Microcontroller is to act as the processing unit of the Evaporometer. We are able to connect sensors and charging devices to complement the microcontroller. The microcontroller we are using is the Feather M0. The specific model we are using has a built-in LoRa radio to transmit data wirelessly.

A separate microcontroller will be used as a hub to receive information from the Evaporometer. The hub ensures that data is transmitted correctly from the Evaporometers. The server microcontroller is connected to an ethernet shield via the FeatherWing Ethernet module which is used to upload data to the Internet.

### 3.2 Sensor

The Sensors are designed to collect weather information from the environment. Each sensor is attached to the microcontroller via wires and programmed to work together. The design of each sensor allows us to add/remove which ever ones we need.

Our current deployment involves: two TSL2561 Lux sensors for light, one HX711 load cell for rain, one DS3231 for time, and one SHT31-D for temperature and humidity. Each sensor can work together using the I2C address assigned to it from the Feather [2]. The built-in I2C controller of the Feather allow us to deploy up to 127 devices with no interference [2].

### 3.3 Charging Method

The modularity of the Feather M0 allows us to connect a charging board to the battery, charging the battery when it is in use or sleeping [3]. Each charging board is connected with a solar panel and battery that will be integrated with our microcontroller in an enclosure.

## 4 DEPENDENCY

### 4.1 Internet of Things

In order to share our collected data on the fly we are using PushingBox to pull data from the microcontrollers. The code we designed for the Feathers will make a call to the PushingBox function, calling our Google script to paste data onto our spreadsheet.

### 4.2 Web Server

The web server is used to host the website of the visualization. It requires data from the GIS to display the visualization properly. It also depends on the code used to set up the visualization.

## 5 INFORMATION

### 5.1 Data Storage

The data storage is used to hold data gathered by the Evaporometer and supply information to the GIS. The data storage used to receive data from the Evaporometer is Google Drive with 15GB of free capacity[4]. The data storage used to supply information to the GIS is MongoDB. The size of MongoDB depends on the size of the local server due it being a database.

## **5.2 Geographic Information System**

Since the GIS is used to visualize the data, it will be dealing with constant data input. There will be a persistent flow of information coming into the GIS, which will allow the system to visualize the information in various maps once the system processes the data[5]. The GIS will be set up to change in real-time, since the data is constantly coming in and changing.

## **6 INTERFACE**

### **6.1 Internet of Things**

The current plan provided by Pushing Box allows us to send 1000 notifications daily [6]. We are using one account to manage the services and scenarios. In our case we are using the services as a function call to our script and scenarios as our function return for the variables.

### **6.2 Web Server**

The web server is used to service any HTTP request to view the data visualization website.

## **7 INTERACTION**

### **7.1 Microcontroller**

The microcontroller in the evaporimeter is connected to the sensors. The microcontroller is also connected to RFM95 LoRa radio which is used to transmit information to the server. The server will then transmit the data to the data storage.

### **7.2 Sensor**

The sensors interact with the environment directly. Once enabled they will measure: humidity, temperature, rainfall, and albedo light. The data is measured directly within the code and stored in variables for the Feathers to manipulate.

### **7.3 Data acquisition**

For the weather hub to successfully acquire the data, it needs to interact with the weather device. The weather device will transmit its data to the weather hub. From there, the GIS will model the data on the maps that it creates. Both of these interactions are necessary for the data to be correctly modeled on the web maps.

### **7.4 Internet of Things**

In our current setup we have a set number of Feathers transmitting data wirelessly to a central Feather acting as a network hub. From there it will interact with our APIs, PushingBox, to push data out onto the cloud.

### **7.5 Web Server**

To prepare the data collected by the device for visualization, there needs to be an interaction between the device and the web server. This interaction is the transmission of the data from the device to the web server.

## 8 STATE DYNAMICS

### 8.1 Microcontroller

The microcontroller in the Evaporometer has two states: read state, and sleep state. The Evaporometer will read from sensors every 15 minutes. It will also transmit the data to the server during the read state. Once the read state is finished, it will go into the sleep stage. The sleep stage is to reduce the power used by the Evaporometer.

### 8.2 Sensor

Sharing the same power resources as the microcontroller, the sensors have two states: read and sleep. During read the sensors will take up to 15 samples, average the results and send it to the microcontroller for transmission. Once the samples have been collected the sensors will enter sleep mode. In sleep mode the sensors no longer collect data to preserve power.

### 8.3 Charging method

We plan to use solar power to charge the microcontroller. Due to variable levels of solar radiation reaching the device and the system constantly consuming battery power while it is on, the charge level of the device will always be changing its state. When the device is on, the battery power will constantly be reduced, but when solar radiation levels are higher, the battery power will increase. As a result, the state of the devices charge will be dynamic.

### 8.4 Geographic Information System

The GIS will be modeling live data coming from the sensors. Since the sensors provide data that is constantly changing, the map modeled by this data will also be changing in real time. The map that is modeled by the GIS will also be changing its state in real time to accurately model the data that comes from the forest. A GIS that is capable of changing state is necessary to accomplish these goals.

## 9 RESOURCE

### 9.1 Data Storage

The storage space on Google Drive has a capacity limit of 15GB for free plans. It can be expanded by paying monthly fees. However, 15GB will take a long time to fill since the data gathered by the evaporometer only takes a few KB per transmit. The capacity of the MongoDB depends on the size of the server. It has high availability and can be scaled horizontally [7].

## 10 CONCLUSION

In summary, the design of our project requires various technologies to work. We are using the availability of cheap microcontrollers such as the Adafruit Feather M0 to form the foundation of our weather station. As a cheap and efficient way of powering our devices we will implement solar charging to charge our devices. Along with that our microcontrollers can attach various sensors to collect data. However, the microcontrollers can only transmit data wirelessly which is not efficient for our stakeholders. For that reason, we are using an Ethernet hub to acquire all of the data in one spot. After our data has been collected we will be storing it on Google drive and MongoDB. This allows our

data to be accessible on our web server where our stakeholders can view and edit the data. To make the data pleasing to the eyes we are developing a website to visualize the data. All of this is only possible through the usage of Internet of Things. Having networked devices is only useful if it can operate without human intervention.



## REFERENCES

- [1] Semtech, "What is LoRa?".[Online]. Available at: <http://www.semtech.com/wireless-rf/internet-of-things/what-is-lora/>
- [2] Adafruit, "Overview—I2C addresses". [Online]. Available at:<https://learn.adafruit.com/i2c-addresses/overview>
- [3] Adafruit, "Design Notes". [Online]. Available at: <https://learn.adafruit.com/usb-dc-and-solar-lipoly-charger/design-notes>
- [4] Google Drive, "Drive". [Online]. Available at: <https://www.google.com/drive/pricing/>
- [5] B. Zhao, "Spatial data for Web mapping".[Online]. Available at: <https://github.com/jakobzhao/geog371/tree/master/lectures/lec06>
- [6] Pushing Box, "API". [Online]. Available at: <https://www.pushingbox.com/api.php>
- [7] MongoDB, "What is MongoDB?". [Online]. Available at: <https://www.mongodb.com/what-is-mongodb>