

Technology review

Yizheng Wang

Group 35

Oregon State University

Computer Science 461 CS senior Capstone

Fall 2017

Abstract

Our project aims at creating an LOT moisture valve system to control the behavior of valve. This paper will give a general introduction about the project and the parts of the project Im working on. For each part it will discuss three different technologies that are possible to complete that part. It will compare those technologies and discuss which one will be use. The three parts I will take responsible are UI organization, generation and handling of the data, and user input and handling.

CONTENTS

I	Problem statement	2
II	Fundamental Part 1 - Machine to machine communication technologyn	2
II-A	Criteria	2
II-B	Technology 1 - Adafruit IO	2
II-C	Technology 2 - Plotly	2
II-D	Comparison	2
III	Part 1 - UI organization	2
III-A	Criteria	2
III-B	Technology 1 - Java Swing	3
III-C	Technology 2 - HTML	3
III-D	Technology 3 - C#	3
III-E	Comparison	3
IV	Part 2 - Generation and handling of the data	3
IV-A	Criteria	3
IV-B	Technology 1 - IFTTT	3
IV-C	Technology 2 Java Http request library	4
IV-D	Technology 3 - PHP	4
IV-E	Comparison	4
V	Part 3 User input and handling	4
V-A	Criteria	4
V-B	Technology 1 - IFTTT	4
V-C	Technology 2 - Java Http request library and swing library	5
V-D	Technology 3 - PHP	5
V-E	Comparison	5
VI	Appendix A - Bibliography	6

I. PROBLEM STATEMENT

The project will be divided in three parts. Soil moisture sensor/valve control, centralized data hub/command relay and web-based user interface/data tracker. I will take responsible for the third part: Web-based User Interface/Data Tracker Serving. Which can be treated as the face of the project, this will be the actual interface with which this systems users will interact.

The user will use this system to view recorded moisture data from various sensors, as well as use it to make decisions on when and where water is most needed in their eld. In particular, this software be used to track local soil moisture levels, and allow agriculturalists to choose the most optimal times to run the irrigation systems that water their crops.

This system will take user input and send it to desired watering schedules (via the relay hub) to the individual sensor/valve-control devices in the desired watering areas. This software should be deployed to both a web-based platform, as well as a desktop application platform. Optimally, the client would also like a mobile app developed for Android, iOS, or potentially both. The data that this software handles should also, preferentially, be available online for others to view.

II. FUNDAMENTAL PART 1 - MACHINE TO MACHINE COMMUNICATION TECHNOLOGYN

A. *Criteria*

The web client should be able to communicate with hub. Adafruit.io is asked by client to use for this aim. It can be treat as a server that can update data without any action from user.

B. *Technology 1 - Adafruit IO*

Adafruit IO is a system that makes data useful. It includes the libraries with MQTT APIs. We will use adafrui_MQTT broker for this project. Its simple and light weight. It only takes about 80 bytes to connect to a server and user can stay connected the entire time.

C. *Technology 2 - Plotly*

Plotly is a technical computing company that develops online data analytics and visualization tools. It provides online graphing, analytics, and statistics tools for individuals and collaboration, as well as scientific graphing libraries.

D. *Comparison*

I do not have much understanding about the machine to machine communication technology, so there isnt any comparison between these two technologies. We choose to use Adafruit IO system because its simple and it works very well.

III. PART 1 - UI ORGANIZATION

A. *Criteria*

The web client should display the data of moisture sensors appropriately. It should list the data in one page and be able to get the input from data with an expand input box or in a new page.

B. Technology 1 - Java Swing

Swing library is an API that supported by Java. Its a GUI subsystem. The components of Swing are lightweight and dont rely on peers. This library can meet most of the requirements of different applications. The purpose of Swing is to build a set of extensible GUI components to enable developers to more rapidly develop powerful Java front ends for commercial applications. By using Swing library, developers are able to implement entirely in Java to promote cross-platform consistency and easier maintenance. It enables the power of model-driven programming without requiring it in the highest API. The functionalities of the web client are displaying the data and handling the data that inputted by users. All of those functionalities can be completed by using built in library of Swing.

C. Technology 2 - HTML

The layout of a web-based platform is mainly based on html language. The Hypertext Markup Language (HTML) is a standard for describing the structure and presentation of information via the Internet. The purpose of HTML is to provide a set of general rules that suggest how content should look when rendered. A markup language doesn't dictate the methods used to display the content, nor does it have foreknowledge of the target context, so this control is imprecise. [6] HTML empowers document authors to apply typographical formatting, document structuring, and the inclusion of images without locking users into a closed, proprietary format and without being dependent upon document preparation specialists to provide the markup. Html can also provide for the platform-independent display of information.

D. Technology 3 - C#

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. [5] Its simple and easy to learn. We can easily organize the UI with application class in C# to adjust the position of buttons, list boxes and input boxes.

E. Comparison

All of technologies can work for this part. But we are not planning to use a database while we may need to store data. I prefer to use Java or C#. According to a research, Java can be compiled in most operating system while C# does not. [3] In this project, we need to create a mobile application as well, if we program with Java it will be easier to transplant the code of the application. Thus, I decide to use Java for UI organization.

IV. PART 2 - GENERATION AND HANDLING OF THE DATA

A. Criteria

The web client should be able to send request to Adafruit and receive data every 15 minutes. The data should be displayed appropriately.

B. Technology 1 - IFTTT

IFTTT is both a website and a mobile app. The idea of IFTTT is to automate everything from app or website to other app-enabled accessories and smart devices. It can be treat as a trigger, if something happens then related device will do another thing. For instance, user may receive an email from IFTTT every time when his door of bedroom is open. To receive data from Adafruit.io, we may set up a trigger for IFTTT. This trigger is when any new data is updated on the Adafruit.io. Then IFTTT can send the data to the web client.

C. Technology 2 Java Http request library

Adafruit.io supports a convenient API, which can simply receive the data from Adafruit.io with the AIO key and has been introduced in the API documentation of Adafruit.io. [2] Adafruit.io will send back the data that contained in the selected group. The type of data will be Json object, which can be resolved by importing Json.org library. Http request library is a library for using a `HttpURLConnection()` call to make requests and access the response.

D. Technology 3 - PHP

PHP is a widely-used open source general purpose scripting language that is especially suited for web development and can be embedded into HTML. We can also use the API supported by Adafruit.io to receive the data from it. To send the request to Adafruit.io, we can use `send()` function and then use `getResponse()` function to receive the JSON object returned by Adafruit.io.

E. Comparison

IFTTT is a convenient platform for user to solve simple LOT problems. However, according to the requirements of the web client, IFTTT may not be a good idea. There are mainly two reasons, first one is IFTTT cant check status. Which means if we want to receive the data from Adafruit.io broken every 15 minutes, IFTTT wont be able to check if the data is updated by the sensors on time. If the sensors or hub shut down, then web client wont be able to receive any data and user cant notice that until users check it. The second reason is IFTTT can only send very few data at once, according to the research of the Customer.io, IFTTT allows you to send up to 3 custom data values along with your request which you can customize with liquid variables. [1] Obviously, this is not enough for this project while we need to handle at least 6 different values at a time. PHP code and Java have similar complexity to achieve the goal of this part. Compared to Java, its more situational for PHP to use JSON object. [4] so I choose to use Java for part 2.

V. PART 3 USER INPUT AND HANDLING

A. Criteria

The web client should be able to generate data from user and send the input data to hub through Adafruit.io. Web client should also be able to change the mode of the valve and send the change to Adafruit.io as well.

B. Technology 1 - IFTTT

Like what has been mentioned in the previous section, IFTTT is the platform required by the client for control the behavior of client because its a convenient platform to control the behavior of board with Adafruit.io. If we want to send data to Adafruit.io through IFTTT. We may create an application that can link with IFTTT. Then set a trigger to wake up IFTTT when its necessary. However, there is no proper solution for IFTTT to get custom input from users to set or change the setting of the mode.

C. Technology 2 - Java Http request library and swing library

The is quite similar to the process to receive data from Adafruit.io. Adafruit.io also supports a web API to send data and change the value of feeds that stored in it. Its possible for Java to send data to Adafruit.io by pull http request using the Http request. The input from user may use an input box to receive. The input box can be created by using the built in library in Java Swing library. Then the input will be added to the end of the web API url, the basic structure of the url will be: API address + default group name + send + AIO key + input value If the data has been sent successfully, it will return 200 as the status code. We can check the status code to make sure the API works.

D. Technology 3 - PHP

PHP code can be embedded into HTML so it will really simple to create an input box and send those data with PHP code in the UI that it programmed by HTML. The web API supported by Adafruit.io can be used to transmit the input from users.

E. Comparison

IFTTT is required to use to control the behavior of the boards. But while there is no proper solution to connect the application with IFTTT or generate input from user by IFTTT, I will not choose this technology for this part. PHP code and Java have similar complexity to achieve the goal of this part. In the first part I choose Java for UI organization, while they are parts of the web client, I will still use Java for this part.

VI. APPENDIX A - BIBLIOGRAPHY

- [1] Send Data to IFTTT, CUSTOMER.IO
- [2] Adafruit IO REST API Documentation, Adafruit
- [3] Kirk, Radeck C# and Java: Comparing Programming Languages, Microsoft, October, 2003
- [4] Yoshitaka, Shiotsu PHP vs. JavaScript, Dec 22, 2016
- [5] Mairaw, Guardrex, Bill Wagner, Tompratt-AQ, Mike Blome Introduction to the C# Language and the .NET Framework, July 20, 2015
- [6] Scott, Loban HTML: Origins and Purpose, Peachpit, Nov 9, 2001