Project Report

An embedded-water control system – Smart Bathroom

Presented to

Dr. Pitchaya Sitthi-amorn

Developed by

5731083221	Peerawut	Luengruengroj
5730604021	Suppasek	Leesomprasong
5831048421	Penpicha	Sinrattanavong
5831054121	Phootip	Tulayavorases
5831072421	Supavich	Asavakanoksilp

Delivered

21 December 2017

This project is a part of 2110363 Hardware Synthesis Laboratory I

Semester 2 of 2017 Class, Department of Computer Engineering,

Faculty of Engineering, Chulalongkorn University

Table of Contents

Chapter 1 Introduction 3

1.1 Introduction 3

1.2 Objective 3

1.3 Functional Requirement 3

1.4 System overview 4

Chapter 2 Developing Tool and Related Technologies 5

2.1 STM32 Board 5

2.2 NodeMCU 5

2.3 Liquid Flow Sensor 6

2.4 Logic Level Convertor Bi-Directional 7

2.5 NetPie and Freeboard 7

2.6 STM32CubeMX and Ozone Debugger 7

Chapter 3 Development Report 7

3.1 Develop Team 7

3.2 System Architecture Report 9

3.3 UI Designer and Development 11

3.4 Embedded System Development 14

3.5 Front-End Developer 15

Chapter 4 User Interface 16

4.1 System Overview 16

4.2 Report section 16

4.3 Valve Status 17

4.4 Water flow section 17

Chapter 5 Conclusion 19

5.1 Limitation 19

5.2 Benefit 19

Chapter 1 Introduction

1.1 Introduction

Forgot to turn off the water after use in bathroom is seem to be most common mistake in dairy life. It can cause over paying bill and wasted the water unnecessary. The classic solution is to warn yourself not to forget to turn off the water after use. However, it’s not the best solution.

Nowadays, an internet of thing (IOT) is one of the parts of our dairy life. They can be seen every day and everywhere. Because they are simple and small that it can apply to almost everything in the world. So, we have an idea about using IOT to solve the common problem such as forgot to turn off the water

As the result, an embedded water control system named “Smart Bathroom” is developed to solve this common mistake. We use liquid flow sensor to measure the water flow and send the data to process if water is on or off and then report to the user by webpage. In addition, user can also remote the valve to turn off the water from faraway. This embedded system will help people solve the common mistake and can show how much the water we used. Moreover, we can save the wasted water and reduce the bill.

1.2 Objective

This project we aim to develop an embedded water control system to check if the water is used and report the usage to user for determining to close it remotely or not.

1.3 Functional Requirement

1.3.1. The system can read the water usage from liquid flow sensor.

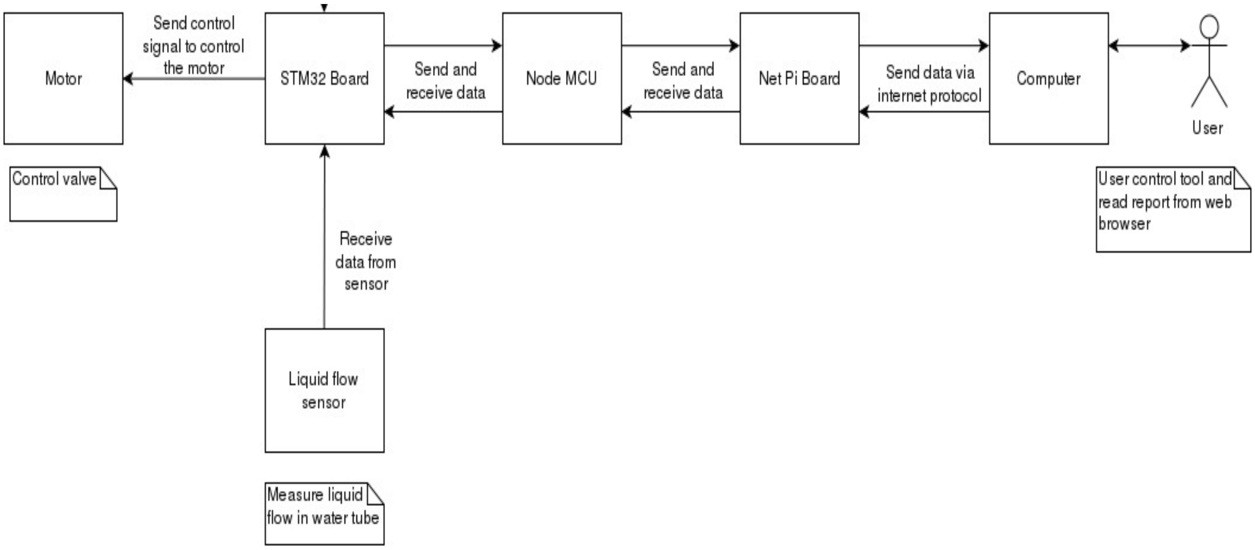
1.3.2. The system can send the data collected from liquid flow sensor and send to user by NodeMCU via internet connection.

1.3.3. The user can read the report about the water usage from the webpage.

1.3.4. The user can stop the valve from the webpage.

1.4 System overview

This system can describe as following block diagram.

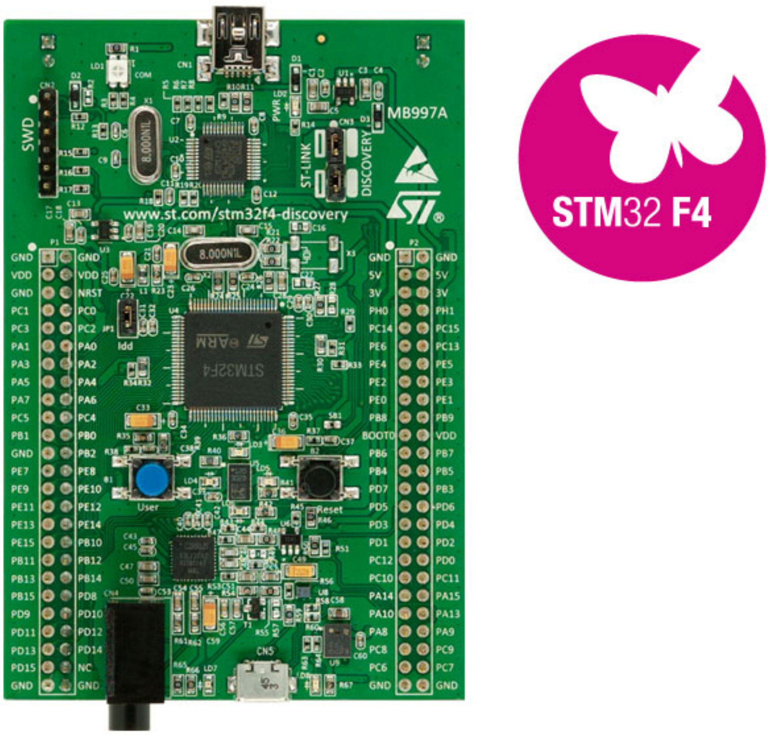

Motor
Send control signal to control the motor
STM32 Board
Send and receive data
Node MCU
Send and receive data
Net Pi Board
Send data via internet protocol
Computer
User
Control valve
User control tool and read report from web browser
Receive data from sensor
Liquid flow sensor
Measure liquid flow in water tube

Picture 1.4-1 Block diagram of Smart Bathroom

Chapter 2 Developing Tool and Related Technologies

2.1 STM32 Board

We use STM32F4DISCOVERY developing kit board to process the data from sensor and send them to NodeMCU by USART (Universal Synchronous/Asynchronous Receiver-Transmitter) which is built-in module in board.

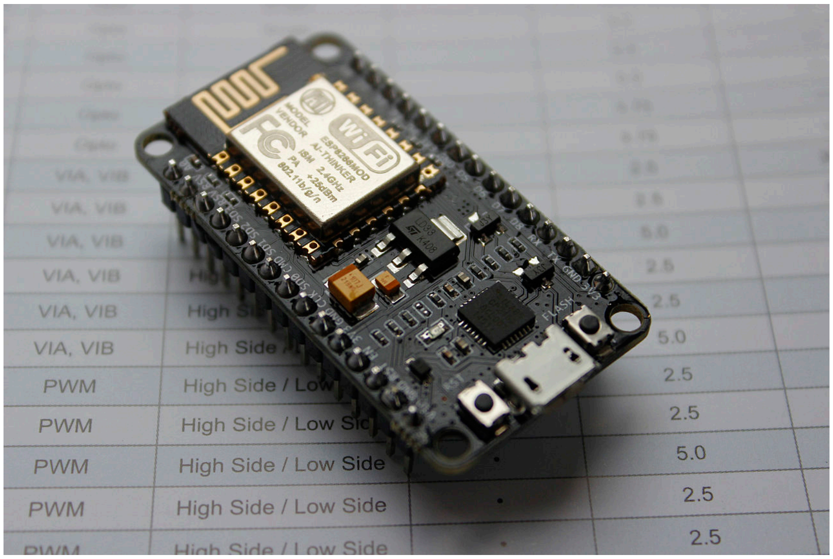


Picture 2.1-1 STM32F4DISCOVERY model STM32F407VG

(Picture from http://www.st.com/en/evaluation-tools/stm32f4discovery.html)

2.2 NodeMCU

NodeMCU is microcontroller which have built-in WiFi chipset. We use it for send the data from STM32 board to internet.



Picture 2.2-1 NodeMCU model ESPN8266

(Picture from https://en.wikipedia.org/wiki/NodeMCU)

2.3 Liquid Flow Sensor

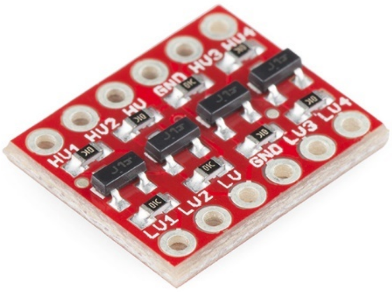
We use liquid flow sensor to collect the data from water tube and send to STM32 for processing.

Picture 2.3-1 Liquid Flow Sensor

(Picture from https://www.seeedstudio.com/G1%26amp%3B2%26quot%3B-Water-Flow-Sensor-p-635.html)

2.4 Logic Level Convertor Bi-Directional

Because STM32 cannot provide 5V source and it cannot stand 5V input signal. It is necessary that we have to converse the voltage between liquid flow sensor which send signal at 5V and use 5V power source.



Picture 2.4-1 Logic Level Convertor Bi-Directional

(Picture from http://www.thaieasyelec.com/products/interface-modules/logic-level/logic-level-converter-bi-directional-detail.html)

2.5 NetPie and Freeboard

We use packaged software name Freeboard which can received the data from NodeMCU to show to webpage. NodeMCU also have a library called NetPie which can send the data to internet.

2.6 STM32CubeMX and Ozone Debugger

We use STM32CubeMX to create the project and developed on Eclipse IDE (STM32CubeMX is also Eclipse Plugin). We also use Ozone Debugger as deploy and debugging tool to STM32 board.

Chapter 3 Development Report

3.1 Develop Team

The team members and their roles are described in following table.

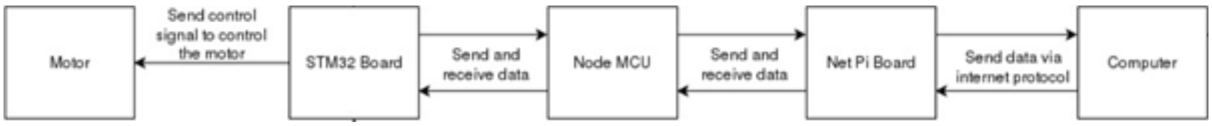
Name	Role
Peerawut Luengruengroj	Embedded system development
Penpicha Sinrattanavong	System architecture
Phootip Tulayavorases	Front-end development
Supavich Asavakanoksilp	UI designer and development
Suppasek Leesomprasong	UI designer and development

Table 3.1-1 Member and their roles.

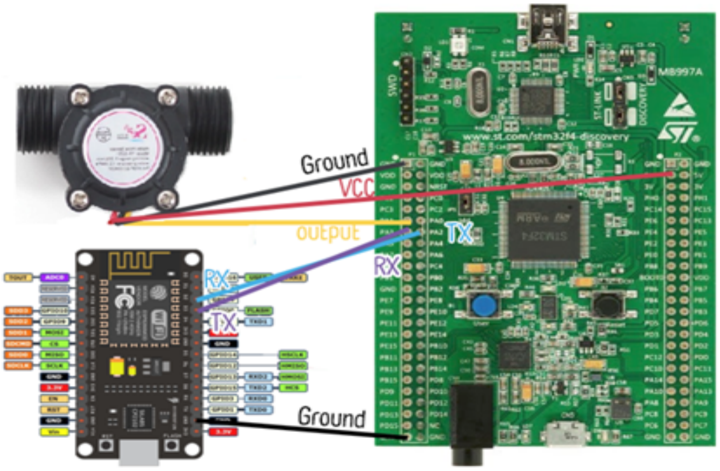
3.2 System Architecture Report

Name: Penpicha Sinrattanavong

- We connect Board STM32F4 and Web Server via NodeMCU and NETPIE. The below chart shows how we connect them.

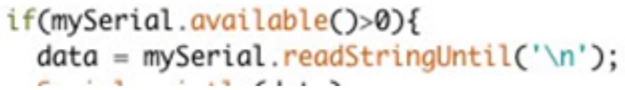


- Communication between Board STM32F4 and NodeMCU can be done by using UART which transmit data from board to nodeMCU with a help from SoftwareSerial. We wire pin D2 to RX which will recieve data from board and pin D3 to TX which will transmit data to board. Below picture is the example.



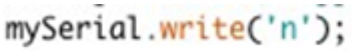
*nb : in reality, you can’t wire 5V directly so additional wiring may require.

- Data from board is in String format which tell us the flow rate from the sensor. After receiving it, We then detect data which ended with ‘/n’ then decode it.



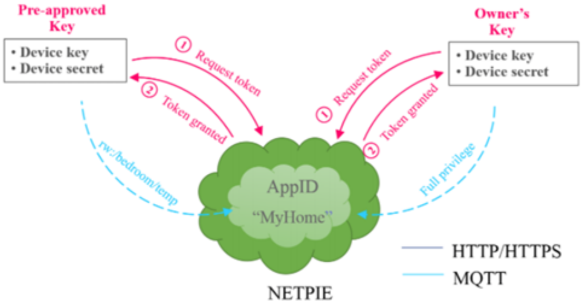
```
if(mySerial.available()>0){
  data = mySerial.readStringUntil('\n');
```

- To send a signal to turn on/off water valve to board, we send a signal in char format. ‘n’ is a signal to turn on the valve while ‘f’ is a signal to turn it off.

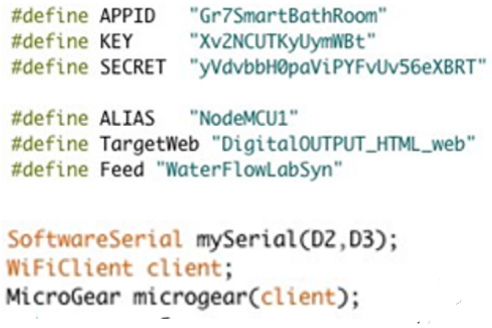


```
mySerial.write('n');
```

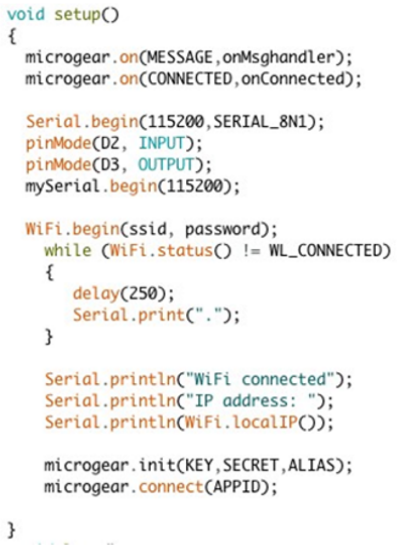
- Communication between NodeMCU and NETPIE can be done by using NETPIE’s library named Microgear. We first have to register our devices to NETPIE’s cloud server by giving it three parameters: AppID, App Key, and Token. Devices with the same AppID can communicate with each other's.



We then input parameters and setup as the following.

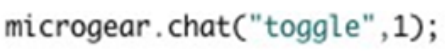


```
#define APPID   "Gr7SmartBathRoom"
#define KEY     "Xv2NCUTKyUymWBt"
#define SECRET  "yVdvbbH0paViPYFvUv56eXBRT"

#define ALIAS   "NodeMCU1"
#define TargetWeb "DigitalOUTPUT_HTML_web"
#define Feed "WaterFlowLabSyn"

SoftwareSerial mySerial(D2,D3);
WiFiClient client;
MicroGear microgear(client);
```



```
void setup()
{
  microgear.on(MESSAGE,onMsghandler);
  microgear.on(CONNECTED,onConnected);

  Serial.begin(115200,SERIAL_8N1);
  pinMode(D2, INPUT);
  pinMode(D3, OUTPUT);
  mySerial.begin(115200);

  WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
       delay(250);
       Serial.print(".");
    }

    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    microgear.init(KEY,SECRET,ALIAS);
    microgear.connect(APPID);

}
```

- Data from nodeMCU can be sent to the server by using this command:
  microgear.chat (alias , message)



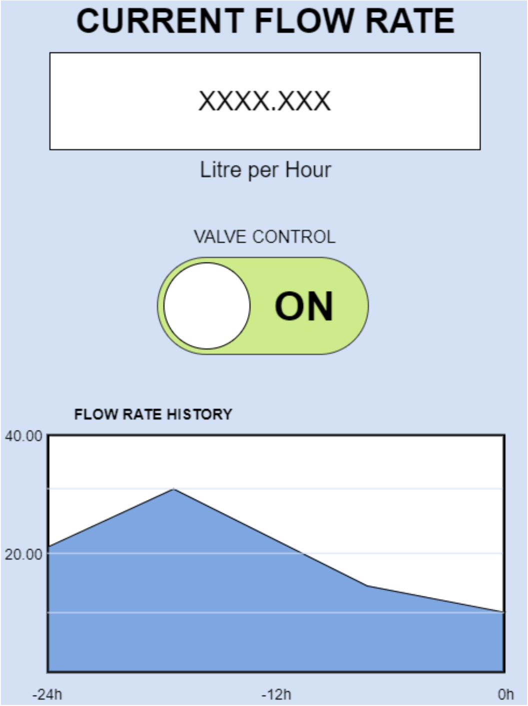
```
microgear.chat("toggle",1);
```

3.3 UI Designer and Development

Name: Supavich Asavakanoksilp

- To meet our objective, UI must include these parts.
  - A number shown current flow rate.
  - A graph which shown us flow rate.
  - A number unit of water that is used.
  - On/off button to turn on/off valve.
  - An input to limit amount of water allowed to flow.

- This is the first draft.

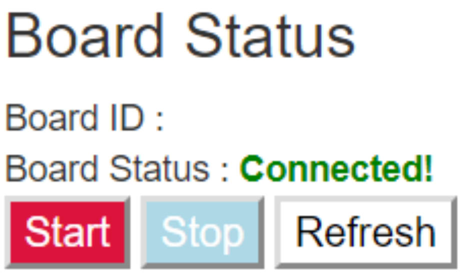


- When using Freeboard, customization is not freely so it is hard to match the draft as the most of UIs have their own template, but this is just a minor problem because we still can create an informative UI.

Name: Suppasek Leesomprasong

ในส่วนของ Front-End UI เพื่อให้ User ได้ทราบถึงข้อมูลต่าง ๆ ของบอร์ด จะแสดงผลดังนี้

1. Board Status


Board Status

Board ID :

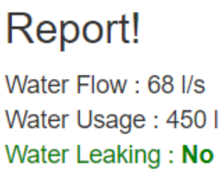
Board Status : Connected!

แสดง Board ID และ Board Status โดย ถ้าเชื่อมต่อกับบอร์ดอยู่จะแสดงข้อความ Connected! แต่ถ้าบอร์ดไม่ได้เชื่อมต่อจะแสดงข้อความ Disconnected!

2. Graph

จะแสดงข้อมูลการใช้งานน้ำของแต่ละวันในรูปแบบของแผนภูมิ

3. Report


Report!

Water Flow : 68 l/s

Water Usage : 450 l

Water Leaking : No

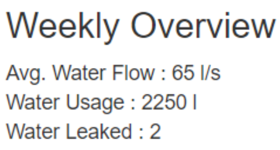
แสดงข้อมูลการใช้งานน้ำปัจจุบัน ดังนี้

Water Flow แสดงอัตราการไหลของน้ำปัจจุบัน (ถ้าน้ำปิดอยู่จะเป็น 0)

Water Usage แสดงปริมาณน้ำที่ใช้

Water Leaking จะแสดง No เมื่อไม่มีการรั่วไหลของน้ำ และจะแสดง Yes เมื่อบอร์ดสามารถตรวจจับการรั่วไหลของน้ำได้

4. Weekly Overview


Weekly Overview

Avg. Water Flow : 65 l/s

Water Usage : 2250 l

Water Leaked : 2

แสดงการใช้งานน้ำของสัปดาห์นั้น ดังนี้

Avg. Water Flow แสดงอัตราการไหลเฉลี่ยของน้ำ

Water Usage แสดงปริมาณน้ำที่ใช้ทั้งหมด

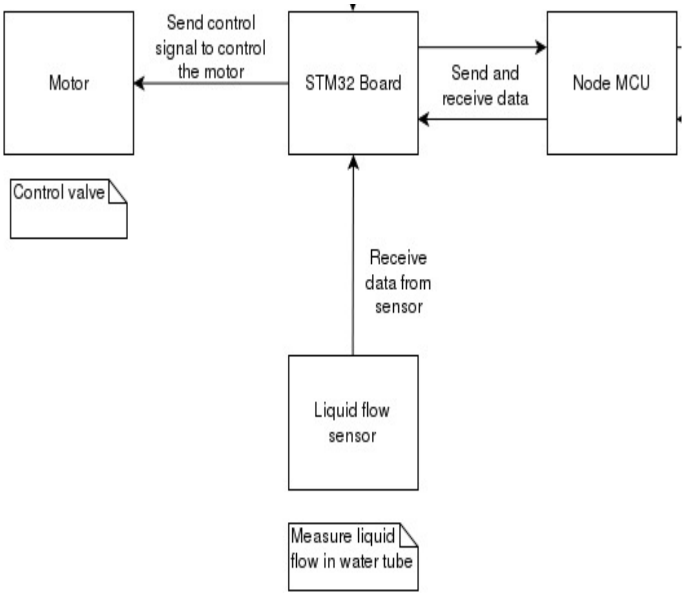
Water Leaked แสดงจำนวนครั้งที่น้ำรั่วไหล

3.4 **Embedded System Development**

Name: Peerawut Luengruengroj

I have an embedded system development role and I also work as project manager too. My responsivities are

- Design the system overview, gathering the system requirements.
- Program the STM32 board to process the data collected from sensor and send the data between STM32 board and NodeMCU.
- Research the tool and support the team about tools and development technologies.
- Initial project and control the source code version in GitHub.
- Schedule the tasks in project and manage the team to finished the work on time.
- Control the document quality and provide document contents.
- Control project’s changed requirements.



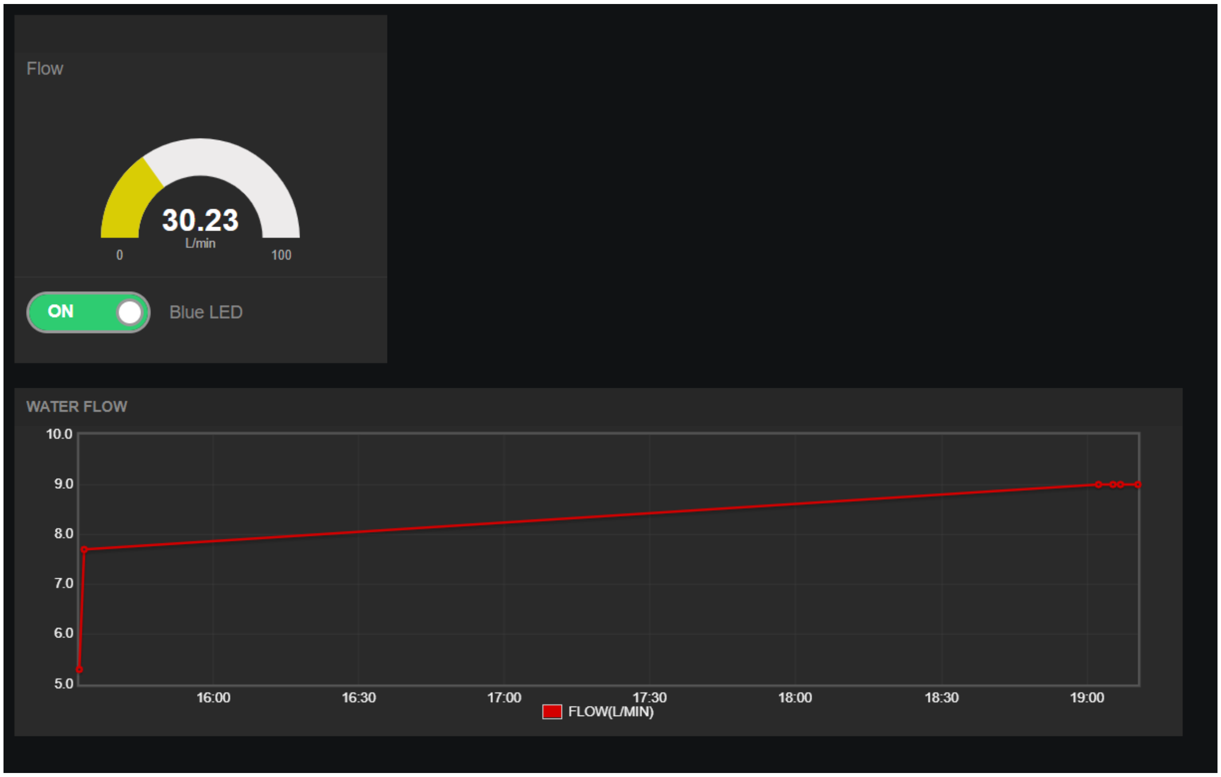
Picture 3.4-1 Role’s scope

3.5 Front-End Developer

Name: Phootip Tulayavorases

I am a Front-End Developer who work on whole webpage since front-end (Implement the design) to back-end (Process the raw data). I also work as collaborator to embedded system developer and system architecture.

- Research and implement the webpage's back-end using NetPie and Freeboard.
- Co-op with system architecture to work on NodeMCU.
- Implement the UI as designed by UI designer.
- Manage the Freeboard webpage and program the water flow calculation included water usage from raw data collected by sensor and STM32 board and sent by NodeMCU.
- Design the report and graph to webpage using JavaScript.

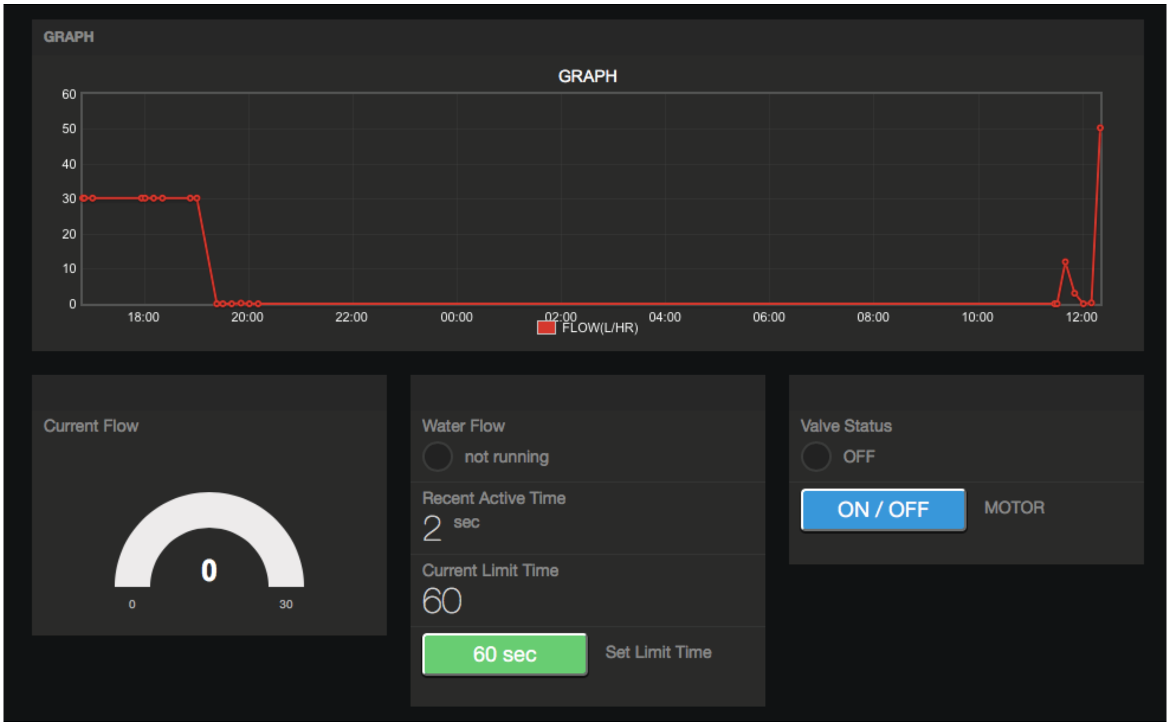


Picture 3.5-1 Webpage Mockup

Chapter 4 User Interface

4.1 System Overview

Shown in following picture



Water Flow
○ not running

Recent Active Time
2 sec

Current Limit Time
60

Valve Status
○ OFF

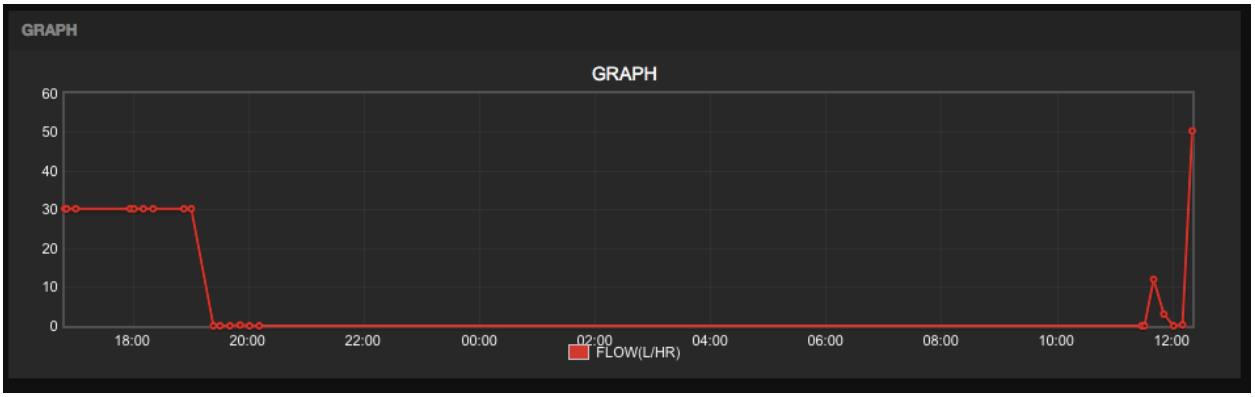
Picture 4.1-1 Overview UI

4.2 Report section

There are two parts of reports

4.2.1 Graph

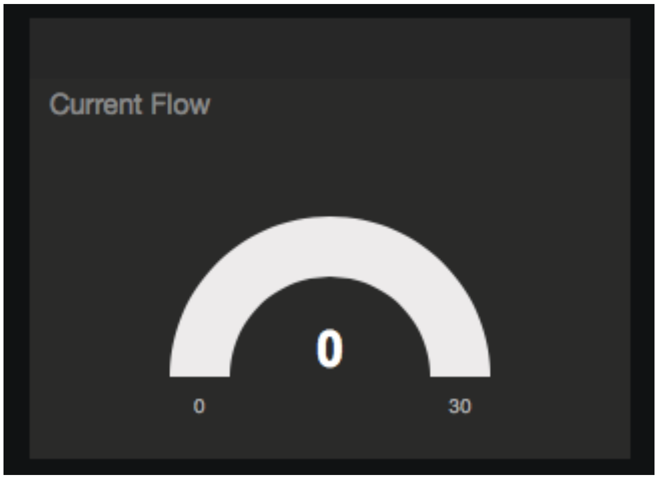
Graph will show the water flow as show in picture



Picture 4.2-1 Graph section

4.2.2 Current Flow

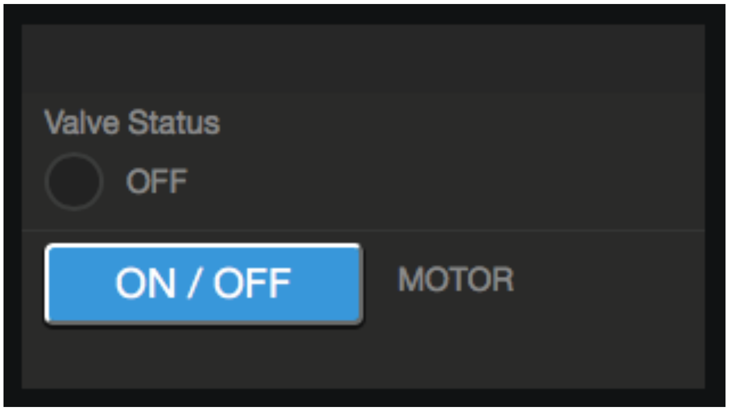
Will show current water flow collected from sensor



Picture 4.2-2 Current Flow Section

4.3 Valve Status

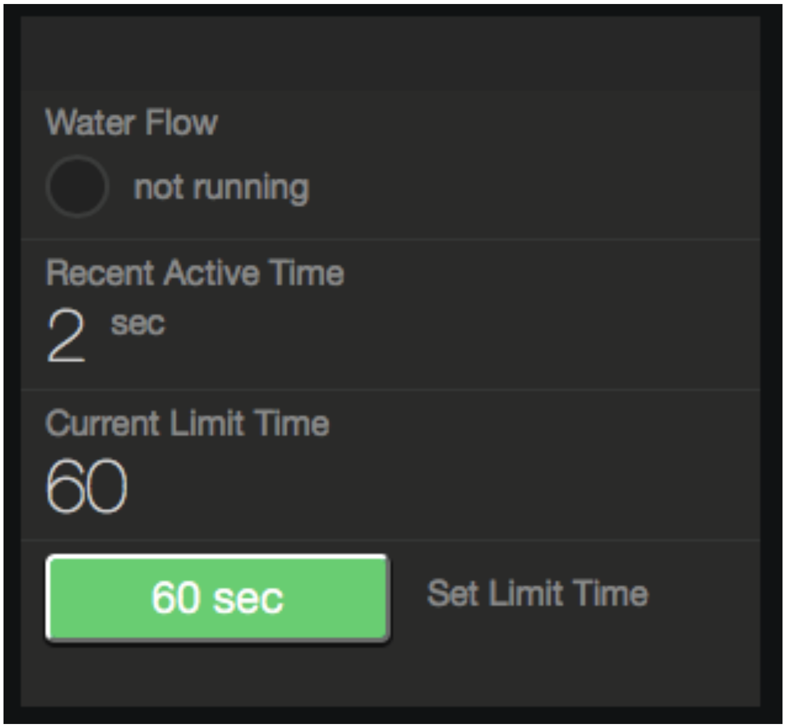
Will show the status of valve if it on or off. The user can control the valve from here.



Picture 4.3-1 Valve Status section

4.4 Water flow section

This section will report the active time of water flow. The user can set limit time of active time in this section for automatic turn off.



Water Flow

○ not running

Recent Active Time

2 sec

Current Limit Time

60

Picture 4.4-1 Water flow section

Chapter 5 Conclusion

5.1 Limitation

4.1.1. Liquid sensor signal is not stable.

4.1.2 Because we don't have real motor, we use the LED to mock the motor working.

4.1.3 Because we don't have motion sensor so we change a requirement from automatic turn off to report to user to remote turn off.

5.2 Benefit

4.2.1 User can turn off the water from anywhere.

4.2.2 Reduced wasted water occurred by mistake.

4.2.3 User can track the water usage.