Chulalongkorn University

2110366 Embedded System Laboratory

Semester 2 Year 2021

Final Project Documentation

“Smart Watering”

Created by

Napat Srithavalya	6330143521
Narongdaet Data	6330146421
Natthaphon Suphansri	6330151521
Nuttapol Onton	6330166021

Introduction

Nowadays, we're living in a touchless society. We're too busy and lazy to water the plants. Our project will solve this problem. We're representing the brand new web-controlled watering with light, soil humidity values.

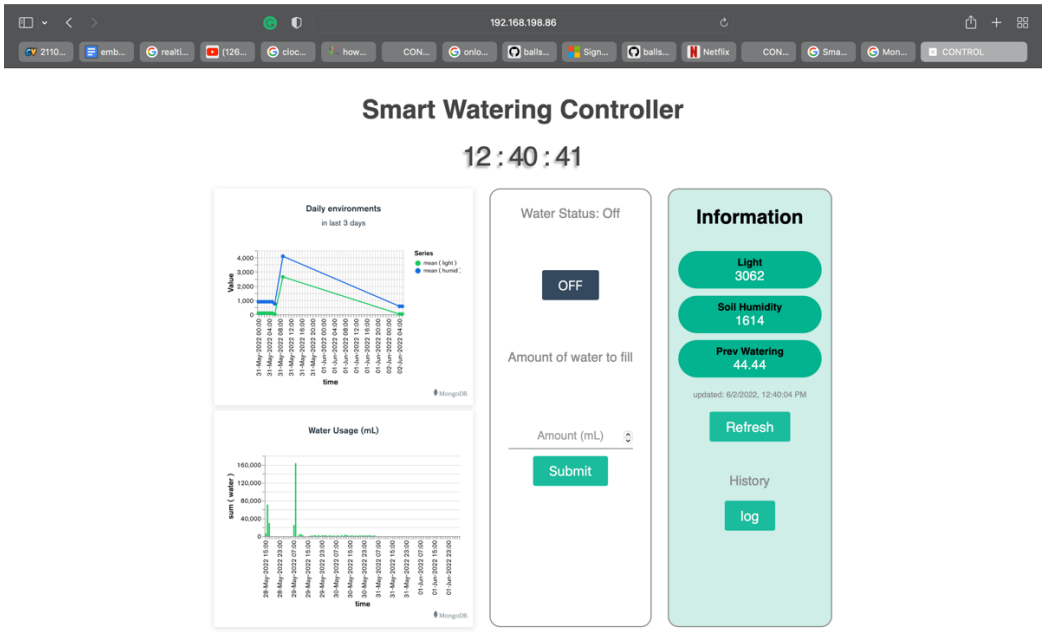
Description

the project that can water the plant by controlling from the website and monitoring the value of all sensors such as light index , soil humidity index , and last watering values by appearing on the website. Moreover, you can see the log or the history of these values on the website.

Code from github

our code : https://github.com/ballsri/IOT-Embbeded-Final

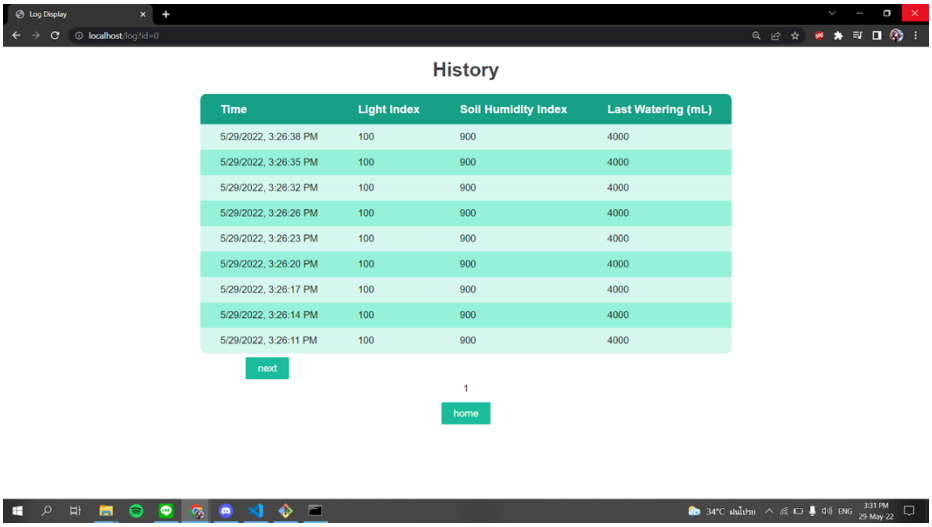
How to use Application



- Left Part is analysis part
  - have Daily environments graph for user to analysis soil humidity light index of your plant
    - green line is about soil humidity index and date
    - blue line is about light index and date
  - have Water usage graph(ml.) for user to analysis amount of water that user watering
    - bar graph is about amount of water and date
- Middle Part is a control part
  - upper part is use to control by switching ON and OFF button
    - use ON/OFF button to control system by manual has the water status label above the button.
  - The lower part is used to control the input amount of water that you need to water.
    - type the amount of water in the box and hit the submit button to water your plant by the amount of water that you type.
    - When the board is watering you need to wait a minute and the soil humidity sensor will be detecting the new values.

- Right Part is monitoring part
  - have a real time info of soil humidity , light , pre-Watering that updates every minute.

- hit the log button to see the history of values that are stored in the database every minute.



- Light index, soil humidity , light , last-Watering(mL) to see all value at that date and time.
- you can go to the next or previous page by hitting the next and prev button to see older logs.
- You can see the page at the bottom.
- when you need to go to the homepage hitting the home button.

Basic Requirements

- using up to 3 different sensors to implement the project.
  - Our group use Light sensor, Water flow sensor and soil humidity sensor to detecting value for watering the plant
  - And use water flow sensor to monitoring the amount of water dispensed
- Be able to control the water dispensing remotely
  - Water pump is controlled by a relay attached to stm32
- IoT project by using node MCU
  - Our group use node MCU to communicate between board STM and server

Project planning

We planned and worked from April 19th to June 2th

	Topics
April 19th - 22th	First meeting - Brainstorm inspiration and topic of project - Choose the best topic of the project and list the function of the project
April 23th- 26th	list the all type the sensor and device that you need to use in this project - find the sensor and choose the one that suitable to our project
April 27th - May 8th	Start the project separated into 4 roles. - plan to set up the board, node MCU and server. - plan to use a database and framework. **role 1 : board STM - hardware implement** - get the value from sensor - implement all function of the project **role 2 : Node MCU and communication between board STM and Server** - implement the communication(UART) of project by using arduino IDE - starting node mcu to connect the server **role 3 : Front End Developer** - Let's start html and css to implement ux/ui interface - Let's start javascript to implement each function of the button. **role 4 : Back End Developer** - Open mongodb and link to the frontend - starting server to connect node mcu
May 9th - 18th	-//---------- Examination Week----------//

May 20th - 31th	- Let's start the project by implementing the code of your role.(next page)
June 1th - 2th	- last check for everything and Present the project

Progression and ROLE

Napat Srithavalya 6330143521

(role 4 : Back End Developer)

- May 20th to May 21th.
    - Decided to use **Mongodb and Nodejs** to host the server and save logs in database, also install and use plugin called **"node-cron"** to be the job's scheduler, **"ejs"** to send data from server to clients and **"request"** to send GET or POST request from server to NodeMCU.
    - Used Express in Nodejs to start the server.
    - Started a free database from MongoDB.
    - Design network for STM32, NodeMCU, Server and Clients
- May 22th to May 23th.

    - Implemented server and **routing** for **POST** and **GET** requests.
    - Decided to transfer data from NodeMCU as **JSON**.
    - Consulted with the team to make the decision about **protocol** to communicate between **NodeMCU**, **STM32** and **Server**.
    - **Implemented communication** between **NodeMCU and Server** with Narongdaet Data.
    - **Connect** the server to the **database** and **create** a **query function** to display **logs** and display **latest data** from the **database**.

- May 24th to May 25th.
    - Implement **Job's scheduler** to receive data from NodeMCU every **10 minutes**
    - Decide to use **Jquery** to retrieve some element (easier than normal Js)
    - Implement timer and function to change button state in JavaScript at **frontend** to deal with "**watering**" time and "**cooldown**" time that **users need to wait for some time** after

sending the amount of water to the STM32 (**while STM32 is watering the plants**) and after watered the plants (**Plants shouldn't been watered frequently**) with Nuttapol Onton.

- Implement backend communicator with JavaScript at frontend with Nuttapol Onton. The communication between backend and frontend should use JSON as the main response data.
- Implement **EJS** logic with data from backend to display **the right state of button** with Nuttapol Onton
- Implement **routing method** to display **logs as page**
- Design logic to make every client be able to connect to the webcontroller at the same time (but not necessary).

- May 26th to May 27th
  - Participate in Frontend part that use to **switch button and change class** in the webpage
  - Implementing **Automated logic** on server to control stm32 from sensor's value.

- May 28th to May 29th
  - Started a **Git repository** and debugging code

- May 30th to May 31th
  - Debug some minor bugs
  - Team members meet at the Faculty of Engineering. and combine each part to test the functionality and fix minor errors.

Narongdaet Data 6330146421

(role 2 : Node MCU and communication between board STM and Server)

- May 20th to May 21th
    - research and learn how to use Node MCU, how to connect node MCU to STM board by port and pin.
    - research how node MCU communicate by using UART, SPI, I2C.
    - how node MCU connects and sends the value to the server.

- May 22th to May 23th
    - set up the node and implement the code of the node MCU by using arduino IDE.
        - we use UART in node MCU to communicate between node MCU and STM board
        - we start the server at NodeMCU and receive request from Nodejs server and this part, I implemented with Napat Srithavalya
        - we connect NodeMCU to the internet

- May 24th to May 25th
    - debugging and continue implement the code in node MCU
        - implement part of communication between node MCU and server

- May 26th to May 27th
    - debugging and continue implement the code in node MCU
        - design encoding method to transfer data. The method is to transfer a character that has bytecode between 0-255 by uart from NodeMCU to STM32 and from STM32 to NodeMCU and implement it.

- May 28th to May 29th
    - Started a **Git repository** and debugging code

- May 30th to May 31th
    - every member in team meeting at university and integrate all part of the program in your role to makes it use

Natthaphon Suphansri 6330151521

(role 1 : board STM - hardware implement)

- May 20th to May 21th
    - Gather **technical information** of all sensors, controls and electronic parts
    - Design the **system wiring** and pin configuration
    - Design the **prototype structure, and water piping**
    - Design STM32 programs (eg. Interrupt callbacks, functions, UART interface cmd)

- May 22th to May 23th
    - **Assembling** all electronic parts that require soldering
    - **Check** if all modules could operate normally
    - Starts the STM32 **program developments:**
        - creating EXTI callback for water flow sensor.
        - create UART interrupt as a preparation for UART interface

- May 24th to May 25th
    - Continue on STM32 program:
        - Develop UART interface for NodeMCU to control STM32 by invoking operating functions.
        - design and implement operating functions, such as, valve opening/closing, watering by volume, parameter getter.
        - implement valve controller circuit and valve power supply
        - implement “watering cooldown” process, which prevent the device from watering the plant too frequent
    - Discuss and design the project’s web server and UX/UI design with other team members

- May 26th to May 27th
    - Implement water piping and protype body
    - solving water pressure problem by removing solenoid valve and use water pump (controlled by relay) instead
    - attach every part onto the prototype body
    - last functionality check

- May 28th to May 29th

- Started a **Git repository** and debugging code

- May 30th to May 31th
  - Team members meet at Fac of Eng. and combine each part to test the functionality and fixing minor errors.

Nuttapol Onton 6330166021
(role 3 : Front End Developer)

- May 20th to May 21th
  - Talking with the team to design User interface by drawing on goodnote for first draft.
  - Brainstorm with teammate, and plan for all component in this project.
  - Decide about function that can do on website and data that be shown on website.

- May 22th to May 23th
  - Try to put some basic website to run with nodeMCU.
  - Try to connect to another device with same wi-fi.
  - Implement the code of the front end such as html and css.

- May 24th to May 25th
  - debugging and continue implement the code of javascript
    - communicate with backend team to make it work
  - implement javascript to make every component can work with back-end.
  - Pull data from back-end to show on website.

- May 26th to May 27th
  - Improve UX/UI to make website more user friendly.
  - Add realtime clock and cooldown time.

- May 28th to May 29th
  - design and implement table to show all of sensor data.
  - Started a Git repository and debugging code to make it work with every component.

- May 30th to May 31th
  - every member in team meeting at university and integrate all part of the program in your role to makes it use.