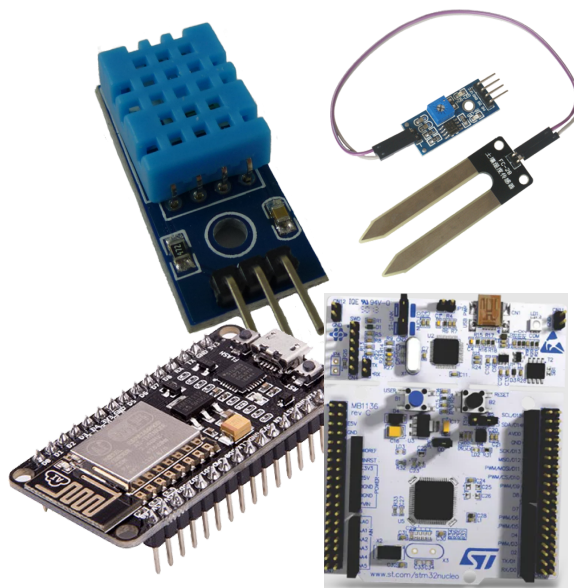


LockPaoMai



สมาชิก

1. 6431307421 Jirawat Lengnoi
2. 6431309721 Chanagun Viriyasathapornpong
3. 6431322821 Tharit Katichanang
4. 6431318321 Natthasith Wiriyayothin

เสนอ

อ.ดร.พิชญะ สิทธิอมร

รายงานนี้เป็นส่วนหนึ่งวิชา 2110366 Embedded System Laboratory

ภาคการศึกษาปลาย ปีการศึกษา 2565

สารบัญ

1 Description	3	
2 Device		3
3 Usage	3	
4 Website	4	
4.1 Website Usage.....	4	
5 Source Code	4	
6 Roles and Responsibility	4	
6.1 Team Management	4	
6.1.1 Responsibility.....	4	
6.1.2 Plan.....	5	
6.2 System Architecture	6	
6.2.1 Responsibility	6	
6.2.2 Hardware Design	6	
6.2.3 Database Design.....	6	
6.2.4 API Design.....	7	
6.3 Embedded System Development	7	
6.3.1 Responsibility.....	8	
6.3.2 Hardware Connectivity	8	
6.3.3 Function of Hardware	9	
6.3.4 ESP8266.....	9	
6.3.5 STM32.....	10	
6.3.6 Operation of the system.....	11	
6.4 UI/UX Designer and Development	11	
6.4.1 Responsibility	11	
6.4.2 UI Design	11	
6.4.3 Web Development	12	

1. Description

โปรเจกต์นี้เป็นโปรเจกต์ที่มีจุดประสงค์เพื่อตรวจสอบว่าต้นไม้ที่ต้องการให้น้ำหรือไม่โดยการวัดความชื้นที่อยู่ในดิน และมีฝนตกอยู่ในปัจจุบันหรือไม่ โดยคาดหวังว่าจะสามารถแก้ไขปัญหาการให้น้ำต้นไม้ไม่สม่ำเสมอจากนี้ยังสามารถเก็บค่าอุณหภูมิและความชื้นของอากาศเพื่อนำมาใช้เป็นเปอร์เซ็นต์ที่จะเกิดฝนตกขึ้นในอนาคตอีกด้วย

2. Device

อุปกรณ์ที่ใช้ประกอบด้วย

1. STM32 NUCLEO F411-RE
2. ESP8266
3. SOIL MOISTURE SENSOR
4. DHT-11
5. LIQUID RAIN WATER LEVEL SENSOR MODULE

3. Usage

1. Soil Moisture Sensor จะวัดความชื้นในดินเพื่อวัดว่าความชื้นดินต้องการน้ำหรือไม่
2. Liquid Rain Water Level Sensor Module วัดความสูงของน้ำเพื่อคำนวณว่าปัจจุบันมีฝนตกอยู่หรือไม่ (ถ้าฝนตกอยู่ก็ไม่จำเป็นต้องรดน้ำต้นไม้)
3. DHT-11 วัดความชื้นอากาศและอุณหภูมิอากาศเพื่อเก็บข้อมูลในเวลาที่ฝนตกเพื่อคำนวณเปอร์เซ็นต์ที่ฝนที่จะตกในอนาคต
4. STM32-NUCLEO F411RE คำนวณและนำค่าของ Sensor แต่ละอันมาประมวลผล
5. ESP8266 เชื่อมข้อมูลของ STM32-NUCLEO F411RE ไปยัง Firebase เพื่อแสดงค่าบน Website

4. Website

<https://embedded-watering-plant.vercel.app/>

4.1 Website Usage

1. บนหน้าเว็บไซต์จะแสดงข้อมูลของอุณหภูมิ ความชื้นอากาศ ความสูงของน้ำ และ ความชื้นในดิน
2. สามารถดูได้ว่าตอนนี้ดินต้องการการรดน้ำต้นไม้หรือไม่
3. สามารถกด How to tell status หรือ ? (ในกรณีจอขนาดเล็ก เช่น ไอแพด) ดูได้ว่าควรรดน้ำต้นไม้ที่แต่ละค่าเป็นเท่าไร

5. Source Code

Source Code และเอกสารเกี่ยวกับ Project สามารถเข้าถึงได้ที่

<https://github.com/nnatchy/2110366-project.git>

6. Roles and Responsibility

6.1 Team Management

By: Jirawat Lengnoi

6.1.1 Responsibility

- เสนอหัวข้อ Project ในการทำงาน
- ประสานงานกับทุกฝ่ายในกลุ่มในการกำหนดวันในการมาทำงานร่วมกัน การแบ่งงานในแต่ละส่วน และช่วยทำในส่วน
ของงานที่ต้องการความช่วยเหลือ
- จัดทำส่วนของการ Presentation
- ดูภาพรวมของ Project ให้เป็นไปตามเป้าหมาย

กำหนดช่องทางการสื่อสารหลักเป็น Discord และได้แบ่งงานให้แต่ละคนดังนี้

1. Tharit Katichanang : System Architecture

(จัดทำ sensor ต่าง ๆ ที่ต้องใช้ ออกแบบระบบว่าแต่ละอุปกรณ์เชื่อมต่อกันได้อย่างไร

2. Chanagun Viriyasathapornpong : Embedded System Development

(การโปรแกรมลง Board)

3. Natthasith Wiriayothin : รับผิดชอบในส่วนของ UI/UX Designer and Development

(ออกแบบพัฒนาเว็บไซต์)

6.1.2 Plan

วันที่ดำเนินการ	งาน
12/04/2023	กำหนดหัวข้อโปรเจกต์
14/04/2023	ออกแบบ UX/UI ใน figma
16/04/2023	เริ่มจัดทำ frontend
5/05/2023	นำเสนอหัวข้อโปรเจกต์ให้อาจารย์อนุมัติ
19/05/2023	นำ Sensor มาเชื่อมกับ STM32 ครั้งแรก
21/05/2023	สามารถเชื่อมต่อข้อมูลระหว่าง Soil moisture sensor, DHT-11 กับ STM32 ได้
23/05/2023	เชื่อมต่อข้อมูลระหว่าง ESP8266 กับ STM32 ได้
24/05/2023	นำค่าที่ได้จาก ESP8266 มาแสดงค่าบนเว็บไซต์
25/05/2023	Deploy เว็บไซต์ลงบน vercel

6.2 System Architecture

By: Tharit Katichanang

6.2.1 Responsibility

1. Hardware Design: กำหนด sensor ที่ต้องใช้ในระบบทั้งหมด

2. Database Design: ออกแบบโครงสร้าง Database

3. API Design: ออกแบบโครงสร้าง API เพื่อการติดต่อของ ESP8266

6.2.2 Hardware Design

ระบบจะต้องมีรายละเอียดและสามารถทำงานได้ดังนี้

- สามารถแสดงค่าความชื้นในดินได้ โดยใช้ Soil Moisture Sensor
- สามารถตรวจสอบดูได้ว่าฝนตกอยู่หรือไม่ โดยใช้ Liquid Rain Water Level Sensor Module
- สามารถเก็บค่าความชื้น และอุณหภูมิในอากาศ โดยใช้ DHT-11

6.2.3 Database Design

โครงสร้าง Database (Real Time Database) แบ่งออกเป็น 4 collections นั่นคือแต่ละ value ที่วัดได้จาก sensor ทั้ง 3 ตัว ได้แก่

- humidity
- soil-moisture
- temperature
- water-level

โดยในแต่ละ collection จะประกอบไปด้วย 4 fields คือ

1. status (string) มีค่าที่เป็นไปได้ค่า “Low”, “Normal”, “High” โดยแต่ละ status จะมีการวัดของแต่ละ collection ที่ต่างกัน
2. value (number) เก็บค่าที่วัดได้ของแต่ละ collection
3. max (number) เก็บค่าที่สูงสุดที่เคยวัดได้ของแต่ละ collection
4. min (number) เก็บค่าที่ต่ำสุดที่เคยวัดได้ของแต่ละ collection

6.2.4 API Design

การใช้งาน Embedded System กับ Firebase Firestore Database นั้นมีขั้นตอนการใช้งานที่ยุ่งยาก อีกทั้งยังต้องมี Boilerplate Code จำนวนมากในการใช้งาน และ Library ยังมี Feature น้อยกว่า Library ใน JS ecosystem เป็นอย่างมาก จึงมีการ Design API เพื่อให้การ Development ของ Embedded System สะดวกขึ้นอีกทั้งยังเป็นการลดภาระของ Embedded System ทำให้ระบบมีประสิทธิภาพและความยืดหยุ่นมากขึ้น

โดย Frontend มีการใช้งาน Next.js ซึ่งเป็น Frontend Web Development Framework และมีการใช้ TailwindCSS ที่เป็น framework ของตัว CSS มาช่วยในการตกแต่ง เพื่อให้มีการตกแต่งได้สะดวก ประหยัดเวลาและสวยงามมากขึ้น

API นั้นเขียนโดยใช้ภาษา Typescript โดยมี built-in function ของ firebase มาช่วยในการดึงข้อมูลจากตัว database สำหรับตัว config ที่ใช้สำหรับตัว firebase จะเก็บไว้ที่ ซึ่ง ณ ที่นี้จะใช้ real time database ที่ทางตัว firebase ได้จัดเตรียมมาให้

<https://github.com/nnatchy/2110366-project/blob/main/frontend/lib/firebase.ts>

Source Code ของ API อยู่ที่

<https://github.com/nnatchy/2110366-project/blob/main/frontend/components/sensor.tsx>

6.3 Embedded System Development

By: Chanagun Viriyasathapornpong

6.3.1 Responsibility

1. การเชื่อมต่อระหว่าง Sensor และ Hardware
2. การโปรแกรม STM32
3. การโปรแกรม ESP8266

6.3.2 Hardware Connectivity

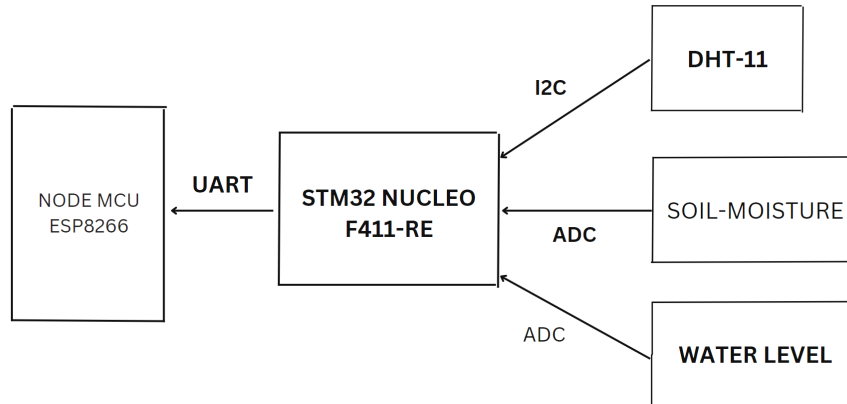
Hardware ทั้งหมดเชื่อมต่อกันโดยมี STM32 NUCLEO F411-RE เป็นศูนย์กลาง โดยมีการเชื่อมต่อกันดังนี้

1. NodeMCU ESP8266 เชื่อมต่อกับ STM32 โดย UART
2. Soil Moisture Sensor เชื่อมต่อกับ STM32 โดย ADC Interface

analog

3. DHT-11 เชื่อมต่อกับ STM32 โดยใช้ I2C

4. Liquid Rain Water Level Sensor Module เชื่อมต่อกับ STM32 โดย ADC Interface analog



6.3.3

Function of Hardware

- STM32 NUCLEO F411-RE มีหน้าที่เป็นศูนย์กลางการเชื่อมต่อระหว่าง Hardware ต่าง ๆ และประมวลผลที่ได้ส่งไปที่ ESP8266
- NodeMCU ESP8266 มีหน้าที่รับค่าของแต่ละ sensor ที่มาจาก STM32 NUCLEO F411-RE และนำค่าเหล่านี้ไป update บน real time database บน firebase
- Soil Moisture Sensor มีหน้าที่ตรวจสอบความชื้นของดิน
- DHT-11 มีหน้าที่อ่านค่าความชื้นและอุณหภูมิของอากาศ
- Liquid Rain Water Level Sensor Module มีหน้าที่ตรวจสอบค่าความสูงของน้ำ (เพื่อดูว่ามีฝนตกอยู่หรือไม่)

6.3.4 ESP8266

ในงานได้มีการใช้งาน ESP8266 ซึ่งได้ทำการโปรแกรมโดยใช้ Arduino IDE และมีการใช้งาน library **SoftwareSerial.h** ในการใช้งาน UART สำหรับการติดต่อกับ STM32 และ **ESP8266WiFi.h** ในช่วยเชื่อมต่อกับ wifi ของ

ESP8266 และ **Firestore** ในการเชื่อมต่อกับ firebase เพื่ออ่านและเขียนข้อมูลใน Realtime database ของ firebase

ESP8266 จะมีขั้นตอนการทำงานดังนี้

1. คอย monitor UART ว่ามีการส่งจาก STM32 หรือไม่โดยจะ monitor จนกว่าจะมีการส่งข้อมูล ข้อมูลที่ส่งจาก STM32 จะเป็น string โดยมี format คือ “T%dH%dS%dR%dE\n\r” โดย T, H, S, R, E หมายถึง temperature, humidity, soil-moisture, water-level และ End ตามลำดับ มี format แบบนี้เพื่อเก็บค่าได้ง่ายตอนส่งไปที่ ESP8266
2. เมื่อได้รับ string ครบถ้วนจะมีการหั่น string ตาม format ที่ได้ตั้งไว้ และ update ค่าต่างๆในตัวแปร temperature, humidity, soil-moisture และ water-level ถ้าตัวแปรไหนไม่ได้รับการวัด ก็จะไม่มีการ update ค่านั้นๆ
3. เมื่อ update ค่าเสร็จ ก็จะนำค่าที่ได้ส่งไปที่ real time database บน firebase โดยจะส่งแต่ละ collection แยกกัน นั่นคือ temperature, humidity, soil-moisture และ water-level และมี field ที่เหมือนกันคือ status, value, min และ max
4. เมื่อส่งข้อมูลเรียบร้อยแล้วจึงกลับไปทำงานขั้นตอนที่ 1 ใหม่

6.3.5 STM32

ในงานนี้ได้มีการใช้ STM32 NUCLEO F411-RE เป็นศูนย์กลางการเชื่อมต่อ hardware ทั้งหมด โดยมีการต่อกับ ESP8266 (โดย UART), Liquid Rain Water Level Sensor (โดย ADC interface analog), DHT-11(โดยใช้ I2C) และ Soil Moisture Sensor (โดย ADC interface analog) โดย STM32 จะคอยรับข้อมูลจาก Liquid Rain Water Level Sensor , DHT-11 , Soil Moisture Sensor และส่งข้อมูลไปยัง ESP8266 เพื่อติดต่อ API และแสดงผลการทำงานผ่าน Website

การเขียนโปรแกรมบน STM32 ใช้ STM32CubeIDE และใช้ภาษา C

1. เขียนโค้ดเพื่อรับค่าจาก DHT-11, Soil Moisture Sensor, Liquid Rain Water Level Sensor เพื่อวัดสภาพแวดล้อมที่อยู่รอบตัว
2. ปรับ ioc ให้เหมาะสมกับการติดต่ออุปกรณ์แต่ละตัว เช่น การใช้ DMA เข้ามาช่วยเพราะมี 2 sensor ที่ใช้ ADC เหมือนกัน

3. เมื่อได้รับค่าจาก sensor ทั้ง 3 ตัวแล้วจึงมาจัดเรียงเป็น String เพื่อส่งไปให้

ESP8266 ไปประมวลผลต่อ ผ่านทาง UART1

6.3.6 Operation of the system

1. STM32 คอยประมวลผลข้อมูลจาก Soil Moisture Sensor, DHT-11 และ Liquid Rain Water Level Sensor Module และส่งไปยัง ESP8266 ผ่าน UART
2. เมื่อ ESP8266 ได้รับข้อมูลผ่าน UART จึงส่งข้อมูลไปยัง API ที่ Firebase
3. หลังจากส่งข้อมูลไปยัง API แล้วจึงนำข้อมูลไปแสดงในหน้า website
4. เมื่อจบกระบวนการจึงกลับไปเริ่มในขั้นตอนแรก

6.4 UI/UX Designer and Development

By: Natthasith Wiriayothin

6.4.1 Responsibility

1. UI/UX Design: กำหนด Feature ของเว็บและการใช้งานของ user ให้มีประสิทธิภาพ และใช้งานง่าย อีกทั้งยังมีการ Design ความสวยงามของ UI
2. Web Development นำ Design ที่กำหนดไว้ในขั้นตอน UI/UX Design มาทำเป็นเว็บไซต์

6.4.2 UI Design

แบ่งส่วนแสดงผลออกเป็น 4 ส่วน ได้แก่

1. ส่วนแสดงข้อมูลอุณหภูมิอากาศ ความชื้นอากาศ ความชื้นในดิน และ เช็คว่าร่นตกอยู่หรือไม่ และแสดงค่า min กับ max ที่วัดได้ของแต่ละค่า
2. ส่วน how to tell status ใช้ในการเช็คว่าร่นน้ำต้นไม้ที่แต่ละคามีค่าเท่าไรบ้าง
3. ส่วนที่แสดงข้อมูลว่าค่าล่าสุดที่แสดงออกมานั้นออกมาตอนไหนเวลาใด
4. แสดงค่าว่าร่นน้ำต้นไม้หรือไม่

6.4.3 Web Development

Web Development นั้นได้ปฏิบัติงานโดยนำ Frontend Design ที่กำหนดไว้ในขั้นตอนการ Design UX/UI มาสร้างเป็น Single Page Application

โดย Frontend มีการใช้งาน Next.js ซึ่งเป็น Frontend Web Development Framework และมีการใช้ TailwindCSS ที่เป็น framework ของตัว CSS มาช่วยในการตกแต่ง เพื่อให้มีการตกแต่งได้สะดวก ประหยัดเวลาและสวยงามมากขึ้น

เว็บไซต์ใช้การดึงข้อมูลจาก Firebase มาแสดงผล การแก้ไขข้อมูลต่างๆ ใน Firebase จะมาจาก ESP8266 ที่ update ค่าต่างๆในตัว Firebase และทางตัวเว็บไซต์จะดึงค่าเหล่านั้นไปแสดงบนตัวเว็บไซต์