

Smart Trash

เสนอ

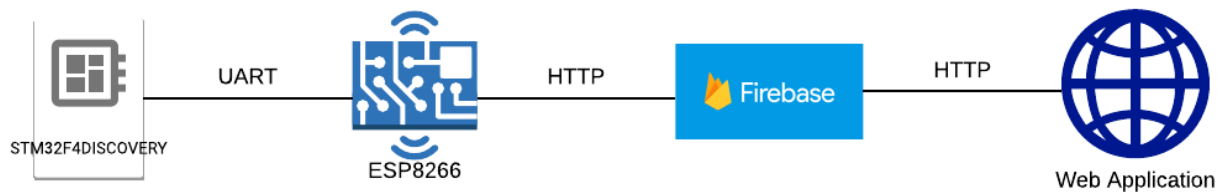
อ. ดร. พิชญะ สิทธิอมร

รายชื่อผู้จัดทำ

1. ธนวัฒน์	เจียรวัฒนกกน	6031020321
2. นิธิภัทร์	ตันติเจริญวิวัฒน์	6031032921
3. นนทนต์	ธีรธนาพัทธ์กุล	6031019821
4. คริษฐ์	ชลพันธุ์	6030070521

โครงการนี้เป็นส่วนหนึ่งของรายวิชา 2110366 การปฏิบัติการระบบฝังตัว
ภาคการศึกษาต้น ปีการศึกษา 2562
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

System Architecture



อุปกรณ์ที่เกี่ยวข้อง

1. Ultrasonic x2
2. Servo
3. STM32F4DISCOVERY
4. ESP8266

STM32F4DISCOVERY เป็นบอร์ดที่ควบคุมการทำงานต่างๆของ Ultrasonic, Servo และรับผิดชอบในการคำนวณ logic ทั้งหมดที่ใช้ในการทำงานของ Smart Trash ส่วนบอร์ด ESP8266 จะถูกใช้เพื่อเชื่อมต่อกับอินเทอร์เน็ต และติดต่อกับฐานข้อมูล

ทางทีมพัฒนาโครงการได้เชื่อมต่อ Ultrasonic และ Servo เข้ากับบอร์ด STM32F4DISCOVERY และส่งผ่านข้อมูลไปยัง ESP8266 ด้วย UART โดยได้มีการกำหนดรูปแบบข้อมูลที่ส่ง ดังตารางต่อไปนี้

รูปแบบข้อมูล	คำอธิบาย
<Q: Value>	ส่ง Value เป็นปริมาณขยะในถังเป็นค่าระหว่าง 0.00 - 1.00 (0.00 คือ ไม่มีขยะในถัง , 1.00 คือ ขยะเต็มถัง)
<O>	มีการเปิดฝาลัง

ซึ่ง ESP8266 จะทำการอ่านข้อมูลแต่ละ tag และจัดเก็บข้อมูลที่ได้รับใน Firebase จากนั้นข้อมูลจะถูกนำไปแสดงผลบน Web Application

Embedded System Development

Ultrasonic ตัวที่ 1 (Inner Ultrasonic) ใช้ในการตรวจวัดปริมาณขยะในถัง

Ultrasonic ตัวที่ 2 (Outer Ultrasonic) ใช้ในการวัดระยะเพื่อส่งสัญญาณ เปิด/ปิด ฝาถัง

Servo ใช้ในการ เปิด/ปิด ฝาถัง

โปรแกรมประกอบไปด้วย 5 states ดังตารางต่อไปนี้

State	คำอธิบาย
CLOSE	เป็น state เริ่มต้น ถ้าหาก Outer Ultrasonic สามารถตรวจจับสิ่งกีดขวางในระยะ 30 cm ได้ จะทำการ decrement ตัวแปร counter ถ้า counter น้อยกว่าหรือเท่ากับ 0 จะทำการส่งข้อมูล <O> ซึ่งบ่งบอกว่าการเปิดฝาลงไปยัง ESP8266 แล้วเปลี่ยนไปยัง state OPENING แต่หาก Outer Ultrasonic ตรวจจับสิ่งกีดขวางไม่ได้ไม่ครบตาม counter ที่กำหนด จะทำการ reset ค่า counter กลับเป็นค่าเริ่มต้น และอยู่ใน state เดิม
OPENING	จะทำการสั่งให้ Servo หมุนไปที่มุม 90 องศา เพื่อทำการเปิดฝาลง จากนั้นจะเปลี่ยนไปยัง state OPEN
OPEN	เป็น state ที่ฝาลงเปิดอยู่ ถ้าหากว่า Outer Ultrasonic ไม่สามารถตรวจจับสิ่งกีดขวางได้ในระยะ 40 cm โปรแกรมจะทำการ decrement ตัวแปร counter หาก counter น้อยกว่าหรือเท่ากับ 0 จะทำการเปลี่ยนไปยัง state CLOSING
CLOSING	จะทำการสั่งให้ Servo หมุนไปที่มุม 0 องศา เพื่อทำการปิดฝาลง ซึ่งถ้าเกิดว่าในระหว่างที่กำลังปิดฝาลงอยู่ Outer Ultrasonic ตรวจจับสิ่งกีดขวางในระยะ 30 cm ได้ จะทำการย้อนกลับไป state OPEN เหมือนเดิม แต่ถ้าไม่มีสิ่งกีดขวางจะทำการเรียกฟังก์ชัน trash_measure_process แล้วเปลี่ยนไปยัง state CLOSE ซึ่งฟังก์ชัน trash_measure_process จะทำการวัดปริมาณขยะในถังโดยใช้ Inner Ultrasonic ซึ่งจะทำการวัด 4 รอบแล้วนำมาหาค่าเฉลี่ยเพื่อให้ได้ผลลัพธ์ที่แม่นยำ แล้วส่งข้อมูลปริมาณขยะในรูปแบบ <Q: Value> ให้กับ ESP8266
FOREVER_OPEN	เมื่อ User ทำการกดปุ่ม Blue Button โปรแกรมจะอยู่ใน state FOREVER_OPEN ซึ่งจะทำการเปิดฝาลงขยะค้างไว้ ถ้าเกิดมีการกดปุ่ม Blue Button อีกครั้งหนึ่ง โปรแกรมจะทำการเปลี่ยนไปยัง state CLOSING เพื่อทำการปิดฝาลงขยะ

UI Design and Development

UI มีองค์ประกอบดังนี้

- Web application ใช้ Template ของ Creative Tim ซึ่ง license เป็นแบบ MIT
- RECENT GARBAGE : บอกปริมาณขยะล่าสุด
- STATUS : บอกสถานะของขยะ
 - Full เมื่อ RECENT GARBAGE ≥ 0.8
 - Not full เมื่อ RECENT GARBAGE < 0.8
- RECENT OPEN : บอกเวลาที่ถังขยะเปิดล่าสุด
- TODAY OPEN : บอกจำนวนครั้งที่ถังขยะเปิดในวันนี้
- Garbage 12 hours history : แสดงปริมาณขยะใน 12 ชั่วโมง ที่ผ่านมา
 - โดยหากภายในชั่วโมงเดียวกันเก็บปริมาณขยะหลายค่า ระบบจะนำค่าสุดท้ายของชั่วโมงนั้นมา plot graph
- Open time : แสดงเวลาที่ถังเปิด 7 อันล่าสุด
- Close time : แสดงเวลาที่ถังปิด 50 อันล่าสุด
- Garbage stats : แสดงข้อมูลปริมาณขยะ ณ เวลาต่างๆ 50 อันล่าสุด

Development

- ใช้ github page ในการ production
- ใช้ realtime database ของ firebase ในการเก็บข้อมูล ทำให้ web application ไม่จำเป็นต้องรีเฟรชเมื่อข้อมูลมีการอัปเดต
- ใช้ Chart.js ในการ plot graph
- ใช้ Moment.js ในการเปรียบเทียบเวลา และการแสดงผล
- ใช้ jQuery ในการอัปเดตส่วนต่างๆของ web application เมื่อ realtime database ถูก create / update / delete

ลิงค์ที่เกี่ยวข้อง

- <https://github.com/tirogen/SmartTrash>
- <https://tirogen.github.io/SmartTrash/>

Team Management

ทางทีมพัฒนาได้แบ่งงานออกเป็น 3 ส่วนหลักๆ คือ

1. พัฒนา Embedded System ให้สามารถควบคุมการทำงานต่างๆของ Smart Trash ได้
 2. พัฒนา ESP8266 ให้สามารถเชื่อมต่อกับฐานข้อมูลได้
 3. พัฒนา Web Application ให้สามารถนำข้อมูลที่อยู่ในฐานข้อมูลไปแสดงผลได้
- ซึ่งงานทั้ง 3 ส่วน สามารถเริ่มต้นพัฒนาได้พร้อมกัน แล้วนำมา integrate กันในภายหลัง

ในการพัฒนาได้มีการใช้ Git เป็น Version Control Tools โดยจะแบ่งเป็น 2 Directories คือ

1. ESP8266 เป็น Directory ที่เก็บไฟล์ที่พัฒนา ESP8266
 2. STM32_main_ioc เป็น Directory ที่เก็บไฟล์ที่พัฒนา Embedded System
- รวมทั้งมีไฟล์ README ระบุว่า อุปกรณ์ต่างๆ ต้องต่อเข้ากับ port ไດ

รายชื่อ	รับผิดชอบ
นิธิภัทร์ ตันติเจริญวิวัฒน์	Web Application Development
ศิษฐ์ ชลพันธุ์	Embedded System Development
นนท์ธนต์ อีรธนาพัทธ์กุล	ESP8266 Development
ธนวัฒน์ เจียรวัฒนกก	Team Management

ลิงค์ที่เกี่ยวข้อง

- <https://github.com/taan02991/SmartTrash>