



เครื่องให้อาหารแมวจากระยะไกล

นายชนภัทร ดวงจรัส

นายผดุงเดช มิ่งขวัญ

นางสาววิยะดา นุ่นกระจาย

นายชัยวัฒน์ ก่อนเก่า

โครงการนี้เป็นส่วนหนึ่งของรายวิชาอินเทอร์เน็ตทุกสรรพสิ่ง (IoT)

สาขาวิเคราะห์และจัดการข้อมูลขนาดใหญ่

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

คำนำ

รายงานเล่มนี้เรื่อง เครื่องให้อาหารแมวระยะไกล เป็นส่วนหนึ่งของรายวิชาอินเทอร์เน็ตทุกสรรพสิ่ง (IOT) โดยมีจุดประสงค์เพื่อศึกษาความรู้เกี่ยวกับการทำงานของ code แพลตฟอร์มและการประยุกต์ใช้ ซึ่งรายงานเล่มนี้มีเนื้อหาเกี่ยวกับการทำงานของ code การใช้แพลตฟอร์มและการประยุกต์ใช้ในงานต่างๆได้

ซึ่งรายงานเรื่อง เครื่องให้อาหารแมวระยะไกล เล่มนี้พวกเราได้วางแผนการดำเนินงานการศึกษาค้นคว้าเป็นเนื้อหาเพื่อให้ช่วยต่อการศึกษาต่อหรือนำไปประยุกต์ใช้ในงานต่างๆต่อไปได้ ทางกลุ่มของเราต้องขอขอบคุณ อาจารย์ผู้ให้ความรู้และแนวทางในการศึกษา วิจัยที่เกี่ยวข้องกับชิ้นงาน และสมาชิกในกลุ่มที่ช่วยในการปฏิบัติงานตลอดมา

จึงหวังว่า รายงานเล่มนี้จะเป็นประโยชน์แก่ผู้ที่ได้อ่านหรือศึกษาต่อหากผิดพลาดประการใดขออภัยมา ณ ที่นี้ด้วย

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	5-7
1.1 ที่มาและความสำคัญของปัญหา	5
1.2 ขอบเขตของการดำเนินงาน	5
1.3 ผลที่คาดว่าจะได้รับ	5
1.4 รายการวัสดุอุปกรณ์ที่ต้องใช้	6
1.5 ตารางการดำเนินงาน	7
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	8-13
2.1. TCP/IP	8
2.1.1 การทำงานของ TCP/IP	9
2.2 ESP 8266	9
2.2.1 การเขียนโปรแกรมควบคุมการทำงาน ESP 8266	9
2.2.2 GPIO ของ ESP 8266	10
2.2.3 ขา GPIO ที่ใช้ในการสร้างชิ้นงานชิ้นนี้ประกอบด้วย	10
2.3 Blynk	11
2.4 Arduino IDE	11
2.4.1 Libraries คืออะไร	11
2.4.1.1 Libraries ที่ใช้ในการเขียน โค้ดลง ESP 8266	12
งานวิจัยที่เกี่ยวข้อง	13
บทที่ 3 วิธีการดำเนินงาน	14-15
3.1 ภาพวงจรวิธีการดำเนินงาน	14
3.1.1 Push Switch module	15
3.2.1 LED module	15
3.3.1 Servo Motor	15
3.4.1 Ultrasonic sensor	15
บทที่ 4 ผลการดำเนินงาน	16-23
4.1 การเรียกใช้ Libraries	16
4.2 การ Define	17

4.3 การกำหนดค่าใน function setup	18
4.4 กำหนดการทำงานใน Function loop	18
4.5 กำหนดการทำงานของ Servo ผ่าน ขา V1 ของ Blynk	19
4.6 Function reset_wifi() สำหรับ reset wifi	20
4.7 Function read_ultraSonic() อ่านค่าจาก ultrasonic()	20
4.8 Function ultrasonic() ทำการรับค่าจากอุปกรณ์ ultrasonic	21
4.9 Set-Up ของแอปพลิเคชัน Blynk	22
4.10 การ Config WIFI	23
บทที่ 5 สรุปผลและข้อเสนอแนะ	24
5.1 สรุปผลการดำเนินการ	24
5.2 ข้อเสนอแนะ	24
บรรณานุกรม	25

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีมีความสำคัญอย่างมากในการใช้ชีวิต รวมไปถึงการควบคุมอุปกรณ์จากระยะไกล ซึ่งมีความรวดเร็วอย่างมาก โดยพวกเรานั้นจึงได้ศึกษาข้อมูลเกี่ยวกับอุปกรณ์ที่เกี่ยวข้องกับระบบ IOT หรือเรียกอีกอย่างหนึ่งว่า “ อินเทอร์เน็ตทุกสรรพสิ่ง ” ซึ่งเราได้เล็งเห็นความสำคัญตรงนี้ เราจึงได้สร้างอุปกรณ์ที่จะเชื่อมต่อกับอินเทอร์เน็ตขึ้นมา โดยอุปกรณ์นั้นมีตัวกลางการเชื่อมต่อเป็น แอปพลิเคชัน Blynk และ โปรแกรมในการใช้เขียนนั้นเป็น Arduino

ซึ่งที่กล่าวมานั้นทำให้เราได้สร้างอุปกรณ์เกี่ยวกับระบบ IOT ขึ้นมาโดยมีการทำงานเกี่ยวกับการให้อาหารแมวจากระยะไกล โดยพวกเราพบปัญหาจากตัวพวกเราเองโดยที่พวกเรานั้นลืมให้อาหารสัตว์เลี้ยง เหตุผลเหล่านี้ เราจึงนำปัญหาและไปสร้างเครื่องมือที่เกี่ยวกับการให้อาหารแมวจากระยะไกล และต่อยอดให้ดีขึ้นในภายภาคหน้า

1.2 ขอบเขตของการดำเนินงาน

- 1.2.1 ระบบตัวเครื่องนั้นทำงานผ่านการใช้งานโดย แอปพลิเคชัน Blynk
- 1.2.2 ตัวเครื่องนั้นสามารถควบคุมการให้อาหารจากระยะไกลได้
- 1.2.3 ตัวเครื่องมีการแจ้งเตือนผ่านแอปพลิเคชัน LINE ว่ามีการเคลื่อนไหวของสัตว์เลี้ยง

1.3 ผลที่คาดว่าจะได้รับ

- 1.3.1 เครื่องให้อาหารสัตว์จะสามารถใช้งานได้จริง
- 1.3.2 เครื่องให้อาหารสัตว์จะสามารถใช้งานระยะไกลผ่านอินเทอร์เน็ตได้

1.4 รายการวัสดุอุปกรณ์ที่ต้องใช้

- 1.4.1 บอร์ด ESP8266
- 1.4.2 เซนเซอร์ตรวจจับการเคลื่อนไหว
- 1.4.3 มอเตอร์ Servo
- 1.4.4 สวิตช์
- 1.4.5 ไฟ LED
- 1.4.6 สาย Jumper
- 1.4.7 สาย Micro USB
- 1.4.8 ปืนกาว
- 1.4.9 กาวร้อน
- 1.4.10 กาวแท่ง
- 1.4.11 กาวร้อน
- 1.4.12 ตะกั่ว
- 1.4.13 บัดกรีแผงวงจร
- 1.4.14 ไม้กระดานหนา 3 มม. ขนาด A4
- 1.4.15 ไม้กระดานหนา 3 มม. ขนาด A3
- 1.4.16 เลื่อยไม้
- 1.4.17 เกลบิลไทล์
- 1.4.18 แผงไขปลา

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทความนี้จะกล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องกับอุปกรณ์และการสร้างเครื่องให้อาหารแมวจากระยะไกลซึ่งประกอบด้วย TCP/IP ,แผงวงจร ESP 8266 ,Blynk จะทำงานโดยมี Blynk เป็นตัวควบคุมจากระยะไกล ผ่าน Http โดยเป็นตัวนำส่งข้อมูลผ่านเครือข่ายไร้สาย

2.1.TCP/IP

TCP/IP หรือ Transmission Control Protocol / Internet Protocol คือชุดของโปรโตคอลการสื่อสารที่ใช้ในการเชื่อมต่ออุปกรณ์เครือข่ายบนอินเทอร์เน็ต TCP/IP ยังสามารถใช้เป็นโปรโตคอลการสื่อสารในเครือข่ายคอมพิวเตอร์ส่วนตัว ชุด IP Address ทั้งหมด – ชุดของกฎและขั้นตอน – มักเรียกกันว่า TCP / IP TCP และ IP เป็นโปรโตคอลหลักสองโปรโตคอลแม้ว่าจะมีโปรโตคอล อื่น ๆ รวมอยู่ในชุดก็ตาม ชุดโปรโตคอล TCP / IP ทำหน้าที่เป็นขนานนามธรรมระหว่างแอปพลิเคชันอินเทอร์เน็ตและฝ่ายการกำหนดเส้นทาง / การสลับ TCP / IP ระบุวิธีการแลกเปลี่ยนข้อมูลทางอินเทอร์เน็ตโดยการให้การสื่อสารแบบ end-to-end ซึ่งระบุว่าควรแบ่งออกเป็นแพ็กเก็ตกำหนดแอดเดรสส่งกำหนดเส้นทางและรับที่ปลายทางอย่างไร TCP / IP ต้องการการจัดการจากส่วนกลางเพียงเล็กน้อยและได้รับการออกแบบมาเพื่อให้เครือข่ายมีความน่าเชื่อถือด้วยความสามารถในการกู้คืนโดยอัตโนมัติจากความล้มเหลวของอุปกรณ์ใด ๆ บนเครือข่าย

โปรโตคอลหลักสองโปรโตคอลในชุด IP Address ทำหน้าที่เฉพาะ TCP กำหนดวิธีที่แอปพลิเคชันสามารถสร้างช่องทางการสื่อสารผ่านเครือข่าย นอกจากนี้ยังจัดการวิธีการรวมข้อความเป็นแพ็กเก็ตขนาดเล็กก่อนที่จะส่งผ่านอินเทอร์เน็ตและประกอบใหม่ตามลำดับที่ถูกต้องตามที่อยู่ปลายทาง

IP กำหนดวิธีกำหนดแอดเดรสและกำหนดเส้นทางแต่ละแพ็กเก็ตเพื่อให้แน่ใจว่าไปถึงปลายทางที่ต้องการ คอมพิวเตอร์เกตเวย์แต่ละเครื่องบนเครือข่ายจะตรวจสอบที่อยู่ IP นี้เพื่อกำหนดตำแหน่งที่จะส่งต่อข้อความ

ชั้นเน็ตมาส์คือสิ่งที่บอกคอมพิวเตอร์หรืออุปกรณ์เครือข่ายอื่น ๆ ส่วนใดของที่อยู่ IP ที่ใช้เพื่อแสดงเครือข่ายและส่วนใดที่ใช้แทนโฮสต์หรือคอมพิวเตอร์เครื่องอื่นบนเครือข่าย

การแปลที่อยู่เครือข่าย (NAT) คือการจำลองเสมือนของที่อยู่ IP NAT ช่วยปรับปรุงความปลอดภัยและลดจำนวนที่อยู่ IP ที่องค์กรต้องการ

2.1.1 การทำงานของ TCP/IP

TCP / IP ใช้ รูปแบบการสื่อสารไคลเอนต์เซิร์ฟเวอร์ที่ผู้ใช้หรือเครื่อง (ไคลเอนต์) ให้บริการเช่นการส่งเว็บเพจโดยคอมพิวเตอร์เครื่องอื่น (เซิร์ฟเวอร์) ในเครือข่าย

โดยรวมแล้วชุดโปรโตคอล TCP / IP ถูกจัดประเภทเป็นแบบไม่ระบุสถานะซึ่งหมายความว่าคำขอของไคลเอนต์แต่ละรายการถือว่าใหม่เนื่องจากไม่เกี่ยวข้องกับคำขอก่อนหน้า การไร้สถานะทำให้เส้นทางเครือข่ายเป็นอิสระเพื่อให้สามารถใช้งานได้อย่างต่อเนื่อง

2.2 ESP 8266

ESP8266 เป็นชื่อเรียกของชิพของโมดูล ESP8266 สำหรับติดต่อสื่อสารบนมาตรฐาน WiFi ทำงานที่แรงดันไฟฟ้า 3.0-3.6V ทำงานใช้กระแสโดยเฉลี่ย 80mA รองรับคำสั่ง deep sleep ในการประหยัดพลังงาน ใช้กระแสต่ำกว่า 10 ไมโครแอมป์ สามารถ wake up กลับมาส่งข้อมูลใช้เวลาต่ำกว่า 2 มิลลิวินาที ภายในมี Low power MCU 32bit ทำให้เราเขียนโปรแกรมสั่งงานได้ มีวงจร analog digital converter ทำให้สามารถอ่านค่าจาก analog ได้ความละเอียด 10bit ทำงานได้ที่อุณหภูมิ -40 ถึง 125 องศาเซลเซียส รายละเอียดเพิ่มเติมจากผู้ผลิตอ้างอิงตามลิงค์นี้ ESP8266 Datasheet

เมื่อนำชิพ ESP8266 มาผลิตเป็นโมดูลหลายรุ่น ก็จะขึ้นต้นด้วย ESP866 แล้วตามด้วยรุ่น เช่น ESP-01 , ESP-03 , ESP-07 , ESP-12E

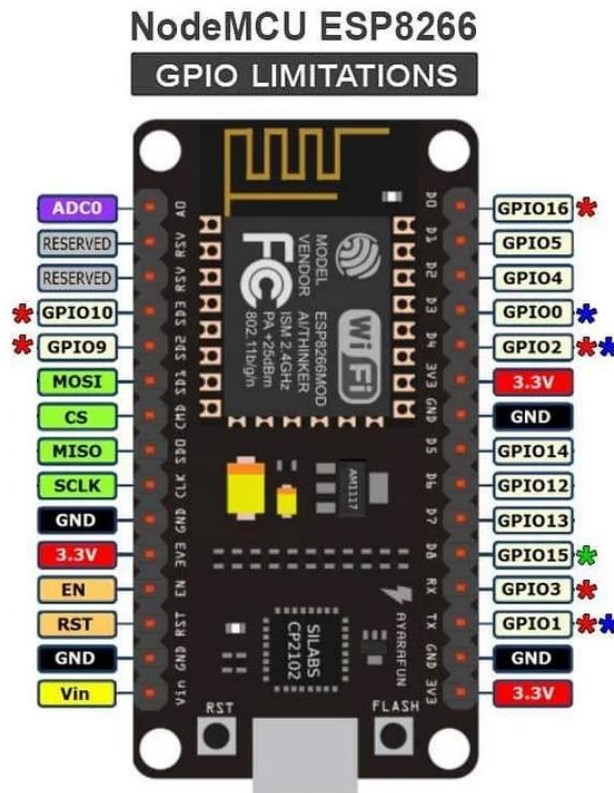
ESP8266 ติดต่อกับ WI-FI แบบ Serial สามารถเขียนโปรแกรมลงไปในชิพ โดยใช้ Arduino IDE ได้ ทำให้การเขียนโปรแกรมและใช้งานเป็นเรื่องง่าย คล้ายกับการใช้ Arduino แน่นอนว่าสามารถติดต่ออุปกรณ์อื่น ๆ เช่นเซอร์ ต่าง ๆ แบบสไต์ล Arduino ถ้ามีพื้นฐาน Arduino อยู่แล้ว ก็เข้าใจและใช้งานได้รวดเร็ว

2.2.1 การเขียนโปรแกรมควบคุมการทำงาน ESP 8266

การเขียนโปรแกรมควบคุมจะต้องใช้การเขียนชื่อที่ขาของอุปกรณ์ ที่เป็น GPIO – General Purpose Input/Output เป็นการเขียนควบคุมการทำงานได้หลากหลาย ไม่ว่าจะเป็น Input รับค่าจาก Sensor มาแปลงเป็นค่า Digital หรือการทำให้เป็น Output ส่งสัญญาณไปยังอุปกรณ์ หรือทริกเกอร์สัญญาณ ให้กับอุปกรณ์ที่รองรับการทำงานนี้ และทำได้หลากหลายมากกว่า 1 คำสั่งใน Pin เดียว

2.2.2 GPIO ของ ESP 8266

มีทั้งหมด 9 ขา D0 – D8 จะรองรับการทำงานที่เป็น Digital Input/Output และ 1 ขา A0 สำหรับ Analog ตามตารางด้านล่าง (ฟังก์ที่เป็น SD2, SD3 เป็นการรับค่าจาก SD Card จะไม่แนะนำให้ไปใช้งานในการควบคุมต่างๆ)



***Pin is high on boot**
***Boot failure if pulled low**
***Boot failure if pulled high**

Best Pins for Input (best to worst)	
Board Label	Raw Pin Number
D1	5
D2	4
D5	14
D6	12
D7	13
D0	16
SD2	9
SD3	10
RX	3

Best Pins for Output (best to worst)	
Board Label	Raw Pin Number
D1	5
D2	4
D5	14
D6	12
D7	13
D8	15

2.2.3 ขา GPIO ที่ใช้ในการสร้างชิ้นงานชิ้นนี้ประกอบด้วย

D1 ต่อกับขา Echo ของ Ultrasonic

D2 ต่อกับขา Trig ของ Ultrasonic ทำหน้าที่ตรวจจับวัตถุ

D3 ต่อกับขา Data ของ Servo ทำหน้าที่หมุน

D7 ต่อกับ Switch ทำหน้าที่ Reset WIFI

D8 ต่อกับ LED ทำหน้าที่เป็นไฟสัญญาณแจ้งสถานะของ WIFI

2.3 Blynk

Blynk App คือ แอปพลิเคชันสำเร็จรูปที่ใช้สำหรับงานที่เกี่ยวกับอินเทอร์เน็ตของสรรพสิ่ง (Internet of Things, IoT) ที่ทำให้เราสามารถเชื่อมต่ออุปกรณ์ต่าง ๆ เข้ากับอินเทอร์เน็ตในลักษณะการเชื่อมต่อเครื่องแม่ข่าย (Server) ไปยังอุปกรณ์ลูกข่าย (Client) เช่น Arduino, ESP-8266, ESP-32, NodeMCU และ Raspberry Pi ซึ่งแอปพลิเคชัน Blynk สามารถใช้งานได้ฟรีและใช้งานได้ทั้งบนระบบปฏิบัติการ IOS และ Android รูปที่ 12.2 แสดงภาพรายการอุปกรณ์ต่าง ๆ ที่สามารถเชื่อมต่อ แสดงผล และ/หรือ ควบคุมด้วย Blynk App ได้ โดยเริ่มต้นหลังจากสมัครเข้าใช้งาน

2.4 Arduino IDE

บอร์ด Arduino เป็นไมโครคอนโทรลเลอร์ ที่สามารถอ่านอินพุตจากตัวตรวจจับแสง, ใช้นิ้วกดบนปุ่ม หรือส่งข้อความไปยัง Twitter และเปลี่ยนเป็นแอตต์พุดเปิดใช้งานมอเตอร์, เปิดไฟ LED หรือเผยแพร่ข้อมูลไปยังระบบอินเทอร์เน็ตได้อีกด้วย ซึ่งผู้ใช้งานสามารถควบคุมบอร์ดว่าต้องทำอะไร โดยส่งชุดคำสั่งไปยังไมโครคอนโทรลเลอร์บนบอร์ด ในการทำเช่นนั้นคุณต้องใช้ภาษา Arduino ซึ่งมีคำสั่งเพิ่มขึ้นมาเพื่อเขียนในรูปแบบภาษา C++ และใช้ซอฟต์แวร์ Arduino IDE เป็นหลักในการประมวลผล

ในช่วงหลายปีที่ผ่านมา Arduino เป็นส่วนหลักของโครงการมากมาย ตั้งแต่วัตถุประสงค์ประจำวันไปจนถึงเครื่องมือวิทยาศาสตร์ที่ซับซ้อน ชุมชนออนไลน์ ของ Maker ทั่วโลก ซึ่งมี นักเรียน/นักศึกษา, ผู้ชอบทำงานอดิเรก, ศิลปิน, นักเขียนโปรแกรมและผู้เชี่ยวชาญ ได้รวมตัวกันใช้งานสำหรับ แพลตฟอร์มแบบเปิดนี้ การมีส่วนร่วมของพวกเขาได้เพิ่มความรู้ที่เข้าถึงได้อย่างเหลือเชื่อซึ่งสามารถเป็นประโยชน์อย่างมากสำหรับมือใหม่และผู้เชี่ยวชาญ

Arduino มีจุดกำเนิดเริ่มต้นขึ้นที่สถาบันการออกแบบปฏิสัมพันธ์ Ivrea ประเทศอิตาลี ซึ่งเป็นเครื่องมือที่ง่ายสำหรับการสร้างต้นแบบที่รวดเร็วมุ่งเป้าไปที่นักเรียนที่ไม่มีพื้นฐานด้านอิเล็กทรอนิกส์และการเขียนโปรแกรม แต่ก็มีผู้ใช้หลายคนพยายามนำ Arduino ไปใช้ในระบบงานจริง ๆ ทั้งนี้ขึ้นอยู่กับปัญหาและความยากง่ายของงานนั้น ๆ Arduino เริ่มมีการเปลี่ยนแปลงเพื่อปรับให้เข้ากับความต้องการและความท้าทายใหม่ ๆ จากบอร์ด 8 บิตแบบง่าย ๆ กับผลิตภัณฑ์สำหรับแอปพลิเคชันสำหรับ IoT, อุปกรณ์สวมใส่, เครื่องพิมพ์ 3 มิติ และสภาพแวดล้อมแบบฝังตัว

2.4.1 Libraries คืออะไร

Libraries คือ โค้ดที่ทำให้ sensor และ โมดูลต่างๆ ใช้งานได้ง่ายขึ้นสะดวกต่อการใช้งานและไม่มี ความซับซ้อนมากนัก ซึ่งการใช้งาน Libraries บางตัวผู้ใช้งานจะต้องเพิ่มเข้าไปด้วยตนเอง

2.4.1.1 Libraries ที่ใช้ในการเขียนโค้ดลง ESP 8266

TridentTD_LineNotify ไลบรารีนี้ทำหน้าที่แจ้งเตือนผ่านแอปพลิเคชันLINE

WiFiManager ไลบรารีนี้ทำหน้าที่เชื่อมต่อไวไฟตัวใหม่ถ้าหากย้ายไปที่อื่น และสามารถเชื่อมไวไฟตัวเก่า

SimpleTimer ไลบรารีนี้ทำหน้าที่หน่วงเวลาในการให้อาหารสัตว์ให้ลงมาตามระยะเวลาที่กำหนด

Servo ไลบรารีนี้ทำหน้าที่หมุนเพื่อเปิดอาหารสัตว์

Blynk ไลบรารีนี้ทำหน้าที่สั่งการเปิด-ปิดอาหารจากทางไกลโดยผ่านแอปพลิเคชัน Blynk

งานวิจัยที่เกี่ยวข้อง

ชื่อโครงการ เครื่องให้อาหารสัตว์จากระยะไกล

ผู้จัดทำ นักศึกษามหาวิทยาลัยราชภัฏวชิรฯ คณะวิทยาศาสตร์และเทคโนโลยี สาขาวิเคราะห์และการจัดการข้อมูลขนาดใหญ่ กลุ่มที่ 5

ศึกษาจาก เนื่องจากกลุ่มบุคคลที่เราได้ทำการสำรวจส่วนใหญ่มักจะเลี้ยงแมวและได้พบประสบปัญหาเวลาออกไปข้างนอกหรือไม่ว่าง อาจจะไม่มีการให้อาหารสัตว์เลี้ยงของเรา

เครื่องให้อาหารสัตว์นี้ จึงทำให้หมดปัญหาเกี่ยวกับการให้อาหารเมื่อเวลาเราไม่อยู่บ้านหรือไม่ว่างที่จะให้อาหารแก่สัตว์ โดยเครื่องให้อาหารสัตว์จากระยะไกลเครื่องนี้จะคอยให้อาหารสัตว์เลี้ยงโดยจะแจ้งเตือนเมื่อสัตว์เลี้ยงของเราเข้ามาใกล้เครื่องให้อาหาร ซึ่งตัวเครื่องจะตรวจจับและส่งข้อความแจ้งเตือนผ่านแอปพลิเคชันไลน์และเราคดสั่งอาหารผ่านแอปพลิเคชัน Blynk เพื่อปล่อยอาหารลงมา อีกทั้งยังช่วยในเรื่องของระเบียบในการกินของสัตว์เลี้ยง และสามารถควบคุมปริมาณอาหารของสัตว์ได้ เพื่อที่จะไม่ให้สัตว์เลี้ยงของเราได้กินเยอะไปจนเหลือทิ้งหรือไม่พอต่อการกินของสัตว์

เนื่องจากโครงการที่เราทำมาอ้างอิงยังไม่มีมีการแจ้งเตือนและยังไม่มีเซนเซอร์ตรวจจับว่าสัตว์เลี้ยงเดินมากินอาหาร กลุ่มของเราจึงได้ใส่เซนเซอร์เข้าไปให้รู้ว่าสัตว์เลี้ยงมากินอาหารตอนไหนและต้องให้อาหารเมื่อใด แต่จากอ้างอิงที่นำมาใช้นั้น จากวิจัยดังกล่าวได้มีการบันทึกจำนวนปริมาณในการกิน และโหมดคำสั่งที่สามารถกดปุ่มคำสั่งได้จากตัวแป้นที่ติดกับตัวเครื่อง ซึ่งในส่วนนี้นั้นเราเป็นจุดอ่อนของชิ้นงานเรา เพราะเราไม่มีการบันทึกอาหารว่าสัตว์เลี้ยงบริโภคไปเท่าใด เราจึงนำการควบคุมจากระยะไกลผ่านแอปพลิเคชัน Blynk เข้ามาแทนในส่วนนี้

เอกสารอ้างอิง

ชื่อเรื่อง : เครื่องให้อาหารสัตว์อัตโนมัติ

นักวิจัย : กลุ่มนักศึกษาคณะวิศวกรรม ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยนเรศวร

คำค้น : เครื่องให้อาหารสัตว์อัตโนมัติ

หน่วยงาน :-

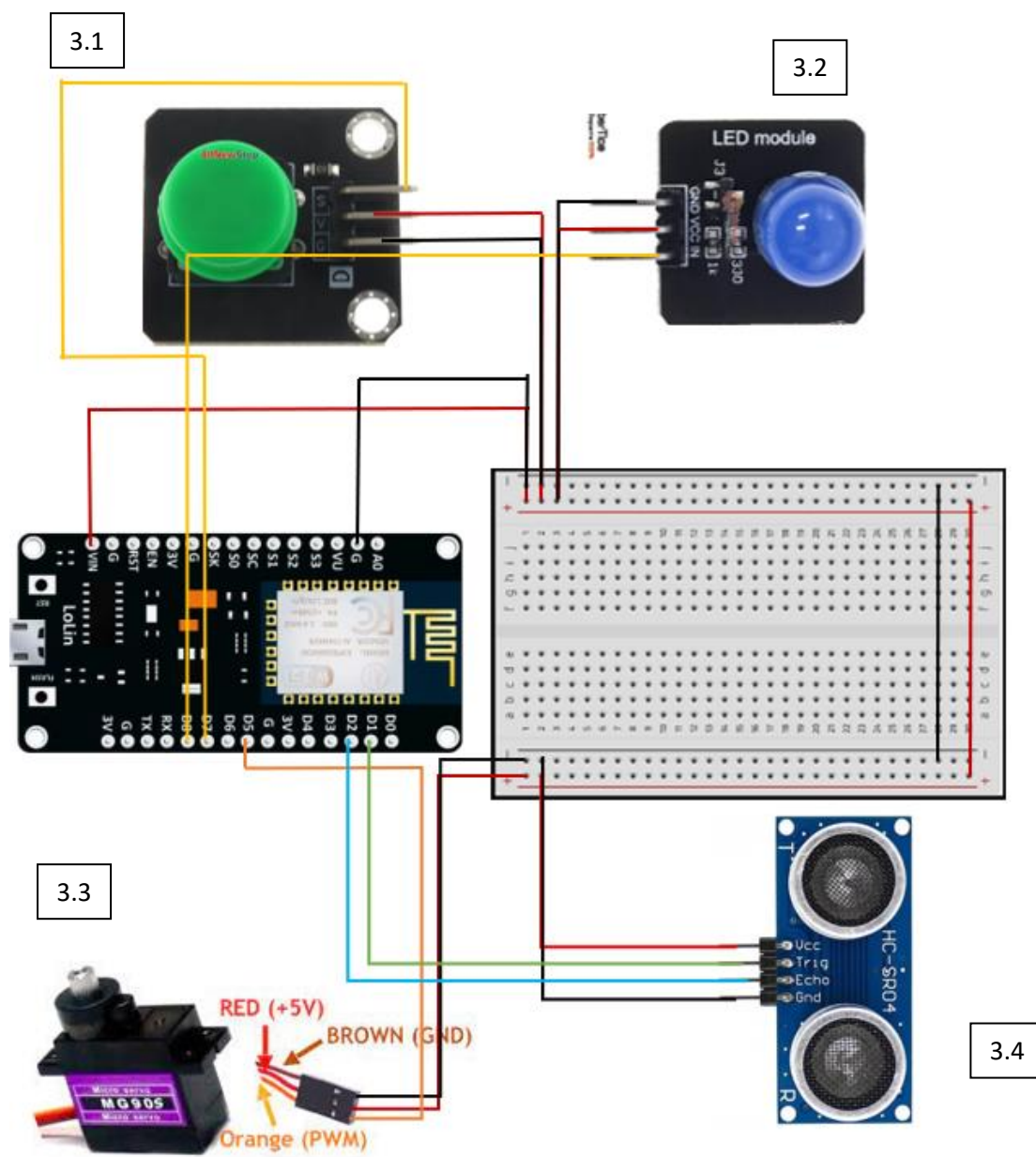
ผู้ร่วมงาน :-

ปีที่พิมพ์ :2557

อ้างอิง : <http://nuir.lib.nu.ac.th/dspace/bitstream/123456789/3134/1/PongpetInvakul.pdf>

บทที่ 3

วิธีการดำเนินการ



3.1 Push Switch module

ขา V ต่อไฟเลี้ยง 5V(+) ที่ Breadboard (สีแดง)

ขา G ต่อลง Ground(-) ที่ Breadboard(สีดำ)

ขา S ต่อเข้า D7 ที่ ESP8266(สีเหลือง)

3.2 LED module

ขา VCC ต่อไฟเลี้ยง 5V(+) ที่ Breadboard (สีแดง)

ขา GND ต่อลง Ground(-) ที่ Breadboard(สีดำ)

ขา IN ต่อเข้า D8 ที่ ESP8266(สีเหลือง)

3.3 Servo Motor

ขา VCC ต่อไฟเลี้ยง 5V(+) ที่ Breadboard (สีแดง)

ขา GND ต่อลง Ground(-) ที่ Breadboard(สีดำ)

ขา PWM ต่อเข้า D5 ที่ ESP8266(สีส้ม)

3.4 Ultrasonic sensor

ขา VCC ต่อไฟเลี้ยง 5V(+) ที่ Breadboard (สีแดง)

ขา GND ต่อลง Ground(-) ที่ Breadboard(สีดำ)

ขา Trig ต่อเข้า D1 ที่ ESP8266(สีเขียว)

ขา Echo ต่อเข้า D1 ที่ ESP8266(สีฟ้า)

บทที่ 4

ผลการดำเนินงาน

ในบทความนี้จะแสดงรายละเอียดของการทำงานภายในโปรแกรมเครื่องให้อาหารแมวจากระยะไกล โดยจะอธิบายการทำงานในแต่ละส่วนที่ได้ใช้อย่างละเอียดภายในรูปและข้อความด้านล่าง

4.1 การเรียกใช้ Libraries

```
1 #include <TridentTD_LineNotify.h> //เรียกใช้ Library แจ้งเตือนโดย Line
2 #include <BlynkSimpleEsp8266.h> //เรียกใช้ Library Blynk แบบง่ายสำหรับ ESP8266 https://github.com/blynkkk/blynk-library/releases/download/v1.1.0/Blynk\_Release\_v1.1.0.zip
3 #include <WiFiManager.h> //เรียกใช้ Library WifiManager โดยเป็นการจัดการเชื่อมต่อ Wifi ผ่าน WebManager
4 #include <ESP8266WiFi.h> //เรียกใช้ Wifi ของ ESP8266
5 #include <SimpleTimer.h> //กำหนดเวลาการทำงานของแต่ละ Function แบบแยกกันโดยไม่ใช้ Delay
6 #include <Servo.h> //เรียกใช้ Library Servo เพื่อให้ใช้งาน คำสั่งควบคุม Servo ได้
7
```

#include <TridentTD_LineNotify.h> //เรียกใช้ Library แจ้งเตือน โดย Line

#include <BlynkSimpleEsp8266.h> //เรียกใช้ Library Blynk แบบง่ายสำหรับ ESP8266

https://github.com/blynkkk/blynk-library/releases/download/v1.1.0/Blynk_Release_v1.1.0.zip

#include <WiFiManager.h> //เรียกใช้ Library WifiManager โดยเป็นการจัดการเชื่อมต่อ Wifi ผ่าน WebManager

#include <ESP8266WiFi.h> //เรียกใช้ Wifi ของ ESP8266

#include <SimpleTimer.h> //กำหนดเวลาการทำงานของแต่ละ Function แบบแยกกันโดยไม่ใช้ Delay

#include <Servo.h> //เรียกใช้ Library Servo เพื่อให้ใช้งาน คำสั่งควบคุม Servo ได้

4.2 การ Define

```
#define BLYNK_PRINT Serial
#define sw D7 //ประกาศตัวแปร sw รับค่าจากขา D7
#define led D8 //ประกาศตัวแปร led รับค่าจากขา D8
#define LINE_TOKEN "yBK8Z5TBpvbRFqJU8IsZmNqUX1ctHiQKVDAmWPMepKU" //รับ Token จาก line notify
#define BLYNK_TOKEN "H14XPmbsr26dYvZDm66vIRpiH4ICrp5J" //รับ Blynk_Token(auth)
```

#define sw D7 //ประกาศตัวแปร sw รับค่าจากขา D7

#define led D8 //ประกาศตัวแปร led รับค่าจากขา D8

#define LINE_TOKEN "yBK8Z5TBpvbRFqJU8IsZmNqUX1ctHiQKVDAmWPMepKU" //รับ Token จาก line notify

#define BLYNK_TOKEN "H14XPmbsr26dYvZDm66vIRpiH4ICrp5J" //รับ Blynk_Token(auth)

```
WiFiManager wm; //ประกาศตัวแปร wm รับค่าจาก Function WiFiManager
SimpleTimer timer; //ประกาศตัวแปร timer รับค่าจาก Function SimpleTimer
Servo servo; //ประกาศตัวแปร servo รับค่าจาก Function Servo
WidgetLED detected(V4); ///ประกาศตัวแปร detected รับค่าจากขา V4 จาก application Blynk
```

WiFiManager wm; //ประกาศตัวแปร wm รับค่าจาก Function WiFiManager

SimpleTimer timer; //ประกาศตัวแปร timer รับค่าจาก Function SimpleTimer

Servo servo; //ประกาศตัวแปร servo รับค่าจาก Function Servo

WidgetLED detected(V4); ///ประกาศตัวแปร detected รับค่าจากขา V4 จาก application Blynk

```
const int pingPin = D1; //ประกาศตัวแปร PingPin รับค่าจากขา D1 ที่ต่อกับตัว Ultrasonic
const int inPin = D2; //ประกาศตัวแปร PingPin รับค่าจากขา D2 ที่ต่อกับตัว Ultrasonic
long distance; //ประกาศตัวแปร distance รับค่าระยะทาง
int timeout = 180; //ตัวแปร Timeout กำหนดเวลาสำหรับ Wifimanager
```

const int pingPin = D1; //ประกาศตัวแปร PingPin รับค่าจากขา D1 ที่ต่อกับตัว Ultrasonic

const int inPin = D2; //ประกาศตัวแปร PingPin รับค่าจากขา D2 ที่ต่อกับตัว Ultrasonic

long distance; //ประกาศตัวแปร distance รับค่าระยะทาง

int timeout = 180; //ตัวแปร Timeout กำหนดเวลาสำหรับ Wifimanager

4.3 การกำหนดค่าใน function setup

```
void setup() {
  WiFi.mode(WIFI_STA); //กำหนด Wifi เป็นโหมด Station
  Serial.begin(9600); //กำหนดความเร็วในการสื่อสาร 9600
  servo.attach(D5); //กำหนดให้ Servo รับค่าจากขาสัญญาณ D5
  pinMode(sw, INPUT_PULLUP); //ทำการกำหนด PinMode INPUT_PULLUP เป็นสัญญาณเข้ามา
  pinMode(led, OUTPUT); //ทำการกำหนด PinMode OUTPUT เป็นสัญญาณออก
  Blynk.config(BLYNK_TOKEN,"blynk.iot-cm.com", 8080); //กำหนด Token,Domain , port ของ Blynk
  LINE.setToken(LINE_TOKEN); // กำหนด LineToken รับค่ามาจาก LINE_TOKEN
  servo.write(80); //กำหนดให้ Servo เริ่มที่ 80 องศา
  timer.setInterval(100L,reset_wifi); //กำหนดเวลาในการทำงานของ Function โดยใช้ SimpleTimer
  timer.setInterval(5000L,read_ultraSonic); //กำหนดเวลาในการทำงานของ Function โดยใช้ SimpleTimer
}
```

WiFi.mode(WIFI_STA); //กำหนด Wifi เป็นโหมด Station

Serial.begin(9600); //กำหนดความเร็วในการสื่อสาร 9600

servo.attach(D5); //กำหนดให้ Servo รับค่าจากขาสัญญาณ D5

pinMode(sw, INPUT_PULLUP); //ทำการกำหนด PinMode INPUT_PULLUP เป็นสัญญาณเข้ามา

pinMode(led, OUTPUT); //ทำการกำหนด PinMode OUTPUT เป็นสัญญาณออก

Blynk.config(BLYNK_TOKEN,"blynk.iot-cm.com", 8080); //กำหนด Token,Domain , port ของ Blynk

LINE.setToken(LINE_TOKEN); // กำหนด LineToken รับค่ามาจาก LINE_TOKEN

timer.setInterval(100L,reset_wifi); //กำหนดเวลาในการทำงานของ Function โดยใช้ SimpleTimer

timer.setInterval(5000L,read_ultraSonic); //กำหนดเวลาในการทำงานของ Function โดยใช้ SimpleTimer

4.4 กำหนดการทำงานใน Function loop

```
36 void loop() {
37   Blynk.run(); //สั่งให้ Blynk ทำงาน
38   timer.run(); //สั่งให้ Timer ทำงาน
39 }
```

Blynk.run(); //สั่งให้ Blynk ทำงาน

timer.run(); //สั่งให้ Timer ทำงาน

4.5 กำหนดการทำงานของ Servo ผ่าน ขา V1 ของ Blynk

```
BLYNK_WRITE(V1){
  int pinValue = param.asInt(); // รับค่า pinvalue 0 1 จาก V1
  if (pinValue == 1){
    servo.write(20); //ทำการสั่งให้ Servo หมุน 20 องศาเพื่อเปิดช่องให้อาหารแมว
    LINE.notify("ให้อาหารแมว"); //ทำการส่งการแจ้งเตือนว่าได้ให้อาหารแมวแล้วไปยัง Line Notify
  }
  else if(pinValue == 0){
    servo.write(80); //ทำการสั่งให้ Servo หมุน 80 องศาเพื่อปิดช่องให้อาหารแมว
  }
}
```

สั่งให้ทำงานถ้า pinValue == 1 และ pinValue == 0

pinValue == 1

servo.write(20); //ทำการสั่งให้ Servo หมุน 20 องศาเพื่อเปิดช่องให้อาหารแมว

LINE.notify("ให้อาหารแมว"); //ทำการส่งการแจ้งเตือนว่าได้ให้อาหารแมวแล้วไปยัง Line Notify

pinValue == 0

servo.write(80); //ทำการสั่งให้ Servo หมุน 80 องศาเพื่อปิดช่องให้อาหารแมว

```
void led_blink() {
  //ทำการสั่งให้ LED กระพริบติด/ดับ 7 ครั้ง
  for (int i = 0; i <= 6; i++){
    delay(500);
    digitalWrite(led,!digitalRead(led));
  }
}
```

ทำการสั่งให้ LED กระพริบติด/ดับ 7 ครั้ง

4.6 Function reset_wifi() สำหรับ reset wifi

```
void reset_wifi(){
  //Function reset_Wifi ทำงานเมื่อ Sw มีค่าเป็น LOW โดยการกดปุ่มที่ Switch
  if (digitalRead(sw) == LOW) {
    led_blink();//เรียกใช้ Function กระพริบเพื่อขอการ Reset Wifi
    wm.setConfigPortalTimeout(timeout);//ทำการเรียกใช้ WebPortal ในการกำหนด Config(SSID>Password) WifiManager
    //เปลี่ยน Wifi ESP8266 ให้เป็น AP Mode และ กำหนดชื่อให้ Wifi ของ ESP8266
    if (!wm.startConfigPortal("Automatic_Cat_Feeder")) {
      Serial.print("Fail to connect and timeout");
      delay(3000);
      ESP.restart(); //ทำการ Reset ESP8266
      delay(5000);
    }
    digitalWrite(led, LOW); //สั่งให้ไฟดับเมื่อทำการต่อ Wifi สำเร็จ
    LINE.notify("WIFI CONNECTED"); //ส่งข้อความว่า "WIFI CONNECTED" ไปยัง Line
    Serial.print("Wifi Connected");
    Serial.print("IP = ");
    Serial.println(WiFi.localIP());
  }
}
```

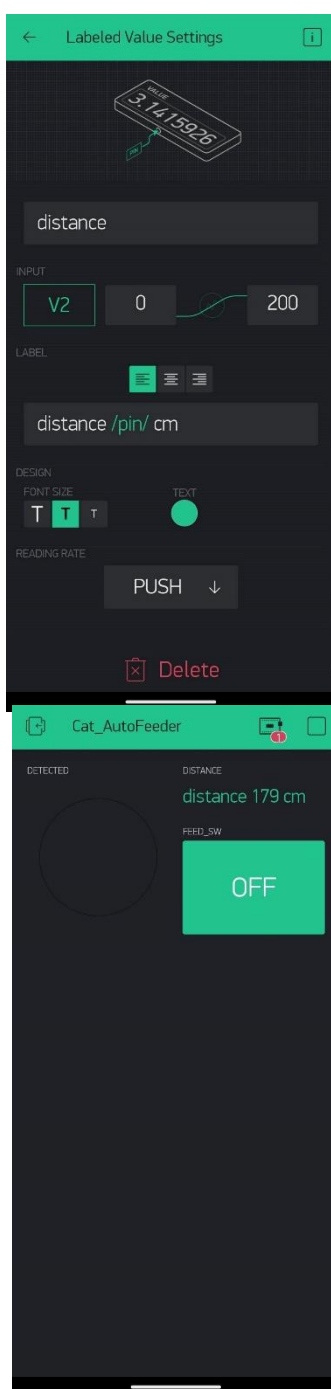
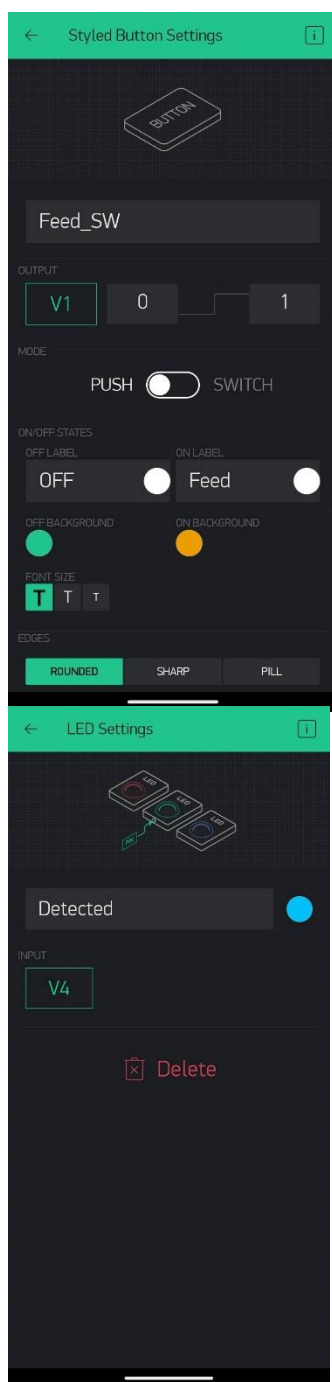
4.7 Function read_ultraSonic() อ่านค่าจาก ultrasonic()

```
void read_ultraSonic(){
  //ทำการอ่านค่าจาก function ultrasonic
  ultraSonic();
  if (distance <= 35){
    digitalWrite(led,HIGH);
    detected.on();
    LINE.notify("ตรวจพบแมวระยะ " +String(distance)+ " CM"); // เมื่อเจอวัตถุในระยะ 35 CM จะส่งแจ้งเตือนทางเป็น LED Blynk และข้อความทาง LINE
  }
  else if (distance >= 36){
    digitalWrite(led,LOW);
    detected.off();
  }
  Blynk.virtualWrite(V2,distance);//ส่งค่าระยะทางไปยัง Blynk
}
```

4.8 Function ultrasonic() ทำการรับค่าจากอุปกรณ์ ultrasonic

```
void ultraSonic() {  
  //Function กำหนดค่าต่างๆของตัว ultrasonic โดยแปลงระยะเวลาเป็นระยะทางหน่วย CM  
  long duration;  
  pinMode(pingPin, OUTPUT);  
  digitalWrite(pingPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(pingPin, HIGH);  
  delayMicroseconds(5);  
  digitalWrite(pingPin, LOW);  
  pinMode(inPin, INPUT);  
  duration = pulseIn(inPin, HIGH);  
  distance = (duration/29)/2;  
  Serial.print(distance);  
  Serial.print("cm");  
  Serial.println();  
}
```

4.9 Set-Up ของแอปพลิเคชัน Blynk



รูปที่ 1

V1 เช็ตค่าสำหรับให้อาหารจากระยะไกล โดยมีชื่อว่า Feed_SW

รูปที่ 2

V2 เช็ตค่าเพื่อแสดงระยะห่างเครื่องกับวัตถุ ตรงหน้าเครื่องให้อาหารสัตว์ โดยตั้งชื่อว่า distance

รูปที่ 3

V4 เช็ตค่าสำหรับแจ้งเตือนการกระพริบไฟในแอปพลิเคชันเมื่อมีวัตถุใกล้ 35 cm โดยตั้งชื่อว่า Detected

รูปที่ 4

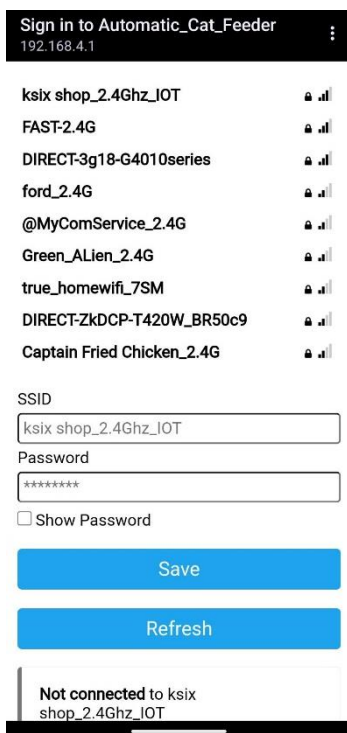
แสดงหน้าหลักของ แอปพลิเคชัน Blynk โดยมีปุ่ม ควบคุมการให้อาหาร ระยะห่างจากวัตถุ และการแจ้งเตือนเมื่อมีวัตถุเข้าใกล้

4.10 การ Config WIFI



หากต้องการเปลี่ยนการเชื่อมต่อจาก WIFI ใหม่ สามารถเข้าไปตั้งค่าเปลี่ยนได้โดยการกด ปุ่ม รีเซ็ตค้างเครื่องแล้วเชื่อม WIFI ที่มีชื่อว่า

Automatic_Cat_Feeder แล้วหลังจากเชื่อมแล้ว เว็บเบราว์เซอร์จะนำไปสู่การ Config WIFI ซึ่งจะขึ้นรูปภาพดังรูป ให้กดปุ่ม Configure WiFi



หลังจากที่เราได้กดเข้ามาแล้ว ให้เลือก WIFI ที่ต้องการเชื่อมต่อ และใส่รหัสผ่านที่ถูกต้อง

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินการ

จากผลการดำเนินการ ศึกษาค้นคว้าข้อมูลในการทำงาน และระบบของเจ้าของการดูแลและการบริการทำให้เกิดความเข้าใจระบบการทำงานของตนเองมากขึ้นซึ่งได้มีการวิเคราะห์ความต้องการในระบบการดูแลและการบริการ ระบบที่ถูกพัฒนาขึ้นมาสามารถใช้งานได้ ดังนี้

- 5.1.1 สามารถสมัครสมาชิกได้
- 5.1.2 สามารถเข้าสู่ระบบได้
- 5.1.3 สามารถสั่งให้เครื่องทำงานได้

5.2 ข้อเสนอแนะ

- 5.2.1 เริ่มจากการรวบรวมข้อมูลทั้งหมดก่อนที่จะออกแบบ
- 5.2.2 ศึกษาข้อมูลและการวางแผนพร้อมการออกแบบ
- 5.2.3 เรียนรู้เกี่ยวกับการเขียนโปรแกรม
- 5.2.4 ควรมีคอมพิวเตอร์ในการคิดแบบและออกแบบในการทดลองใช้

บรรณานุกรม

__(มปป.) //2566.//TCP/IP คืออะไร.// สืบค้นเมื่อ 5 พฤษภาคม 2566 ,/ <https://thaiconfig.com/network/tcp-คืออะไร/>

__(มปป.) 2566. ESP 8266. สืบค้นเมื่อ 5 พฤษภาคม 2566,
<https://www.artronshop.co.th/article/11/esp8266-ตอนที่1-รู้จักกับ-esp-และรุ่นที่นิยมใช้งาน>

__(มปป.) 2566. ขา GPIO ESP 8266. สืบค้นเมื่อ 5 พฤษภาคม 2566,
<https://medium.com/educate/esp8266-nodemcu-gpio-การควบคุม-input-output-แบบ-digital-5772faa584cd>

__(มปป.) 2566. Blynk สืบค้นเมื่อ 5 พฤษภาคม 2566,
<http://suwitkiravittaya.eng.chula.ac.th/B2i2019BookWeb/blynkapp1.html>

__(มปป.) 2566.//Arduino IDE สืบค้นเมื่อ 5 พฤษภาคม 2566,/ <https://www.scimath.org/article-technology/item/9815-arduino>

__(มปป.) 2566.Library คืออะไร สืบค้นเมื่อ 5 พฤษภาคม 2566,
<https://www.analogread.com/article/16/arduino-เอาไปทำอะไรได้บ้าง-ตอนที่11-การเพิ่ม-library-ให้กับโปรแกรม-arduino-ide-ให้ใช้งานได้ใน-4-step>

__(มปป.) 2566. ศึกษาโค้ดเพื่อใช้งาน. สืบค้นเมื่อ 5 พฤษภาคม 2566, <https://www.cybertice.com/article>

__สยามทูเดฟ สยามทูเดฟ. IoT Project การพัฒนาเครื่องให้อาหารแมว Cat Feeder [Video]. สืบค้นเมื่อ 5 พฤษภาคม 2566, <https://www.youtube.com/watch?v=1k9ctS1Krt8&t=326s>

__(มปป.) 2566. เครื่องให้อาหารสัตว์อัตโนมัติ. สืบค้นเมื่อ 5 พฤษภาคม 2566,
<http://nuir.lib.nu.ac.th/dspace/bitstream/123456789/3134/1/PongpetInvakul.pdf>