ระบบแจ้งเตือนการรดน้ำต้นไม้

PLANT WATERING NOTIFY SYSTEM

โดย

นายสหชัย แก้วเกียรติยศ

นายอริญชย์ จิตต์สวัสดิ์

นายอลงกต โสไธสง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

ระบบแจ้งเตือนการรดน้ำต้นไม้

PLANT WATERING NOTIFY SYSTEM

โดย

นายสหชัย	แก้วเกียรติยศ	58011284
นายอริญชย์	จิตต์สวัสดิ์	58011430
นายอลงกต	โสไธสง	58011434

อาจารย์ที่ปรึกษา

รศ.ดร.สุวิพล สิทธิชีวภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

21 / พ.ย. / 2562

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

21 / พ.ย. / 2562

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแจ้งเตือนการรดน้ำต้นไม้

PLANT WATERING NOTIFY SYSTEM

ผู้จัดทำ

1. นายสหชัย แก้วเกียรติยศ 58011284
2. นายอริญชย์ จิตต์สวัสดิ์ 58011430
3. นายอลงกต โสไธสง 58011434

..... อาจารย์ที่ปรึกษา

(รศ.ดร.สุวิพล สิทธิชีวภาค)

กิตติกรรมประกาศ

แม้ว่าในการดำเนินงานจัดทำปริญญานิพนธ์ฉบับนี้ได้มีอุปสรรคต่าง ๆ เกิดขึ้นตลอดในช่วงของการทำงาน แต่อุปสรรคต่าง ๆ เหล่านั้นได้ถูกแก้ไขให้สำเร็จลุล่วงไปได้ด้วยดีเพราะได้รับคำปรึกษาและคำแนะนำที่ดี

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.สุวิพล สิทธิชีวภาค ที่ให้คำปรึกษาคำแนะนำและข้อมูลตลอดทั้งกาลังใจในการทาปริญญานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ความรู้และอบรมแก่คณะผู้จัดทำตลอดจนคณาจารย์และรุ่นพี่ท่านอื่นที่ช่วยเหลือและให้คำแนะนำในการทำปริญญานิพนธ์ฉบับนี้ ปริญญานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงได้เลยถ้าไม่ได้ความช่วยเหลือจากบุคคลข้างต้น ทางคณะผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากทุกท่าน และขอกราบขอบพระคุณเป็นอย่างสูง

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุก ๆ เรื่อง จนทำให้ข้าพเจ้าสามารถทาปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

นายสหชัย แก้วเกียรติยศ
นายอริญชย์ จิตต์สวัสดิ์
นายอลงกต โสไธสง
ผู้จัดทำ

ระบบแจ้งเตือนการรดน้ำต้นไม้

PLANT WATERING NOTIFY SYSTEM

โดย นายสหชัย แก้วเกียรติยศ 58011284

นายอริญชย์ จิตต์สวัสดิ์ 58011430

นายอลงกต โสไธสง 58011434

อาจารย์ที่ปรึกษา รศ.ดร.สุวิพล สิทธิชีวภาค

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอระบบการแจ้งเตือนการรดน้ำต้นไม้ (Plant Watering Notify System) เพื่อประยุกต์ใช้งานเทคโนโลยีร่วมกับการทำเกษตร โดยเป็นการเขียนโค้ดโปรแกรม Python ให้สามารถแจ้งเตือนช่วงที่เหมาะสมแก่การรดน้ำต้นไม้ผ่าน LINE Application ระบบนี้จะทำงานด้วยไมโครคอนโทรลเลอร์ และวัดค่าอุณหภูมิและความชื้นผ่านทางเซ็นเซอร์ โดยได้มีการพัฒนาเซนเซอร์ให้มีการวัดแบบ Real-Time sensor เพื่อให้มีการตรวจวัดอุณหภูมิ ความชื้นสัมพัทธ์และความชื้นในดิน แล้วส่งข้อมูลเข้าสู่ Database และแจ้งเตือนมายัง LINE Application เพื่อทำการรดน้ำต้นไม้ต่อไป

ABSTRACT

This project is the presentation of the Plant Watering Notify System in order to apply technology gathering with agriculture. By writing the Python program code to alert the time range That is suitable for watering plants through the LINE application. This system works by measure the temperature and humidity through the sensor, after that, Sent the data to database and alerted to LINE Application to continue watering the plants.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		VI
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
	2.1 RASPBERRY PI	2
	2.1.1 คุณสมบัติที่สำคัญของบอร์ด RASPBERRY PI	3
	2.1.2 การควบคุมอุปกรณ์อิเล็กทรอนิกส์	4
	2.2 ภาษา PYTHON	4
	2.2.1 คุณสมบัติเด่นของภาษาไพธอน	5
	2.2.2 คุณสมบัติด้อยของภาษาไพธอน	6
	2.3 จาวาสคริปต์ (JAVASCRIPT)	7
	2.4 FIREBASE	7
	2.4.1 คุณสมบัติของ FIREBASE	8
	2.4.2 การจัดเก็บข้อมูลของ FIREBASE	8
	2.4.3 FIREBASE REALTIME DATABASE	9
	2.5 DIALOGFLOW	9
	2.6 DHT22 DIGITAL TEMPERATURE AND HUMIDITY SENSOR	10
	2.7 SOIL MOISTURE SENSOR WITH CABLE	10
	2.8 ADS1115	11

สารบัญ (ต่อ)

		หน้า
	2.9 MINI PUMP MOTOR 5V	12
	2.10 RELAY MODULE	13
	2.11 โนดด็อทเจเอส (NODE.JS)	14
	2.12 LINE APPLICATION	16
	2.12.1 จุดเด่นของ LINE	16
	2.12.2 ประโยชน์และการประยุกต์ใช้งาน LINE	16
	2.13 ปัจจัยที่ควบคุมการเจริญเติบโตของพืช (FACTORS AFFECTING PLANT GROWTH)	17
	2.13.1 แสง	17
	2.13.2 อุณหภูมิ	18
	2.13.3 ความชื้น	18
	2.13.4 แก๊สในดินและบรรยากาศ	18
	2.13.5 ปฏิกิริยาของดิน (SOIL REACTION)	18
	2.13.6 ศัตรูพืช	19
	2.13.7 สารพิษ	19
	2.13.8 ธาตุอาหารพืช	19
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**20**
	3.1 การออกแบบ	20
	3.1.1 การออกแบบการวัดค่าอุณหภูมิและค่าความชื้นในอากาศ	20
	3.1.2 การออกแบบการวัดค่าความชื้นในดิน	21
	3.1.3 การออกแบบการทำงานของวงจรปั๊มน้ำ	22
	3.1.4 การออกแบบการรับค่าข้อมูลจากเซนเซอร์และบันทึกลงในฐานข้อมูล	24
	3.1.5 การออกแบบซอฟต์แวร์ในส่วนของการแจ้งเตือนผ่าน LINE APPLICATION	28

สารบัญ (ต่อ)

		หน้า
	3.2 เครื่องมือที่ใช้ในการทดลอง	33
	3.2.1 SENSOR	33
	3.2.2 MICROCONTROLLER	33
	3.2.3 ANALOG TO DIGITAL CONVERTER	33
	3.2.4 HARDWARE	33
	3.3 การจัดเก็บผลการทดลอง	33
บทที่ 4	**ผลการทดลอง**	**35**
	4.1 ผลการทดสอบการวัดค่าอุณหภูมิและค่าความชื้นในอากาศ	35
	4.2 ผลการทดสอบการวัดค่าความชื้นในดิน	36
	4.3 ผลการทดสอบการทำงานบนฐานข้อมูล DATABASE	37
	4.4 ผลการทดสอบบนแอพพลิเคชั่น LINE	39
	4.5 ผลการทดสอบวัดอุณหภูมิ, ความชื้นสัมพัทธ์ และความชื้นในดิน	40
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**41**
	5.1 สรุปผล	41
	5.2 ข้อเสนอแนะ	41
บรรณานุกรม		**42**
ภาคผนวก	ชุดคำสั่งการทำงานของไมโครคอนโทรลเลอร์	43

สารบัญรูป

รูปที่		หน้า
2.1	บอร์ด RASPBERRY PI 3 MODEL B	2
2.2	ส่วนประกอบของ RASPBERRY PI 3 MODEL B	3
2.3	GPIO (GENERAL PURPOSE INPUT-OUTPUT)	4
2.4	PRODUCT ทั้งหมดของ FIREBASE	8
2.5	หลักการทำงานของเครื่องมือ DIALOGFLOW	9
2.6	DHT22 DIGITAL TEMPERATURE AND HUMIDITY SENSOR	10
2.7	SOIL MOISTURE SENSOR WITH CABLE	11
2.8	ADS1115	12
2.9	MINI PUMP MOTOR 5V	13
2.10	1 CHANNEL RELAY MODULE	14
2.11	โครงสร้างการทำงานของแม่ข่ายโนดด็อทเจเอส (NODE.JS SERVER)	15
2.12	โครงสร้างการทำงานแบบแม่ข่ายมัลติเทร็ด (MULTI-THREADED SERVER)	15
2.13	แพลตฟอร์มที่แอพพลิเคชั่นรองรับในปัจจุบัน	17
3.1	FLOW CHART การวัดค่าอุณหภูมิและความชื้นในอากาศ	20
3.2	FLOW CHART การวัดค่าความชื้นในดิน	21
3.3	FLOW CHART ของวงจรปั๊มน้ำ	23
3.4	FLOW CHART การรับค่าข้อมูลจากเซนเซอร์และบันทึกลงในฐานข้อมูล	24
3.5	การสร้าง FIREBASE PROJECT	25
3.6	REALTIME DATABASE	27
3.7	การสร้าง AGENT ของ DIALOGFLOW และเชื่อมต่อกับ FIREBASE PROJECT	27
3.8	การสร้าง INTENT ขึ้นมาใหม่ในการรับคำสั่งเพื่อไป QUERY ข้อมูลบน DATABASE	27
3.9	เปิดใช้งานคำสั่ง FUFILLMENT เพื่อเขียนโปรแกรม JAVASCRIPT	27

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.10	INTEGRATIONS เพื่อทำการเชื่อมต่อระหว่าง DIALOGFLOW และ LINE BOT	28
3.11	FLOW CHART การออกแบบซอฟต์แวร์ในส่วนของการแจ้งเตือนผ่าน LINE APPLICATION	29
3.12	BOT ที่สร้างขึ้นมาหลังจากที่สร้าง PROVIDER	30
3.13	ค่า CHANNEL ID, CHANNEL SECRET และ CHANNEL ACCESS TOKEN ที่จะต้องนำไปใช้เพื่อเชื่อมต่อกับ DIALOGFLOW	31
3.14	ค่า WEBHOOK URL เป็นค่าจาก DIALOGFLOW เพื่อนำมาใส่ใน LINE BOT	31
3.15	BOT กรณีตอบกลับด้วย WELCOME INTENT และ FALLBACK INTENT	32
3.16	ค่าที่วัดได้จากเซนเซอร์แล้วบันทึกลงในฐานข้อมูล FIREBASE	34
4.1	ค่าอุณหภูมิและค่าความชื้นในอากาศที่เซนเซอร์วัดได้	35
4.2	ค่าความชื้นในดินที่เซนเซอร์วัดได้	36
4.3	ค่าอุณหภูมิ, ความชื้นสัมพัทธ์และความชื้นในดิน	37
4.4	ค่าอุณหภูมิ, ความชื้นสัมพัทธ์, ความชื้นในดินและสถานะการรดน้ำที่ถูกบันทึกในฐานข้อมูล	38
4.5	ค่าอุณหภูมิ, ความชื้นสัมพัทธ์และความชื้นในดินบน LINE APPLICATION	39
4.6	ค่าอุณหภูมิ, ความชื้นสัมพัทธ์และความชื้นในดินที่เปลี่ยนแปลงทุก ๆ 1 ชั่วโมง	40

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันอุตสาหกรรมในประเทศไทยหลาย ๆ ส่วนได้มีการนำอุปกรณ์ไมโครคอลโทรลเลอร์มาช่วยให้เกิดประสิทธิภาพมากขึ้น ในแง่ของภาคครัวเรือนนั้นสามารถนำเทคโนโลยีมาประยุกต์ใช้กับสิ่งที่เป็นกิจวัตรประจำวันของเราได้ซึ่งก็คือ การรดน้ำต้นไม้ที่ปกติแล้วการรดน้ำต้นไม้นั้นต้องรดน้ำทุกวัน แต่บางทีนั้นการรดน้ำที่มากเกินไปนั้นอาจทำให้ต้นไม้เฉาได้เพราะว่ามันไม่สามารถแลกเปลี่ยนก๊าซ หรืออีกกรณีคือการที่เรารดน้ำน้อยเกินไปอาจทำให้ต้นไม้เหี่ยวลงได้ เนื่องจากน้ำในดินมีไม่พอให้รากดูดไปเลี้ยงลำต้น กลุ่มของข้าพเจ้าจึงเล็งเห็นว่าเทคโนโลยีเหล่านี้สามารถนำมาใช้เพื่อให้ชีวิตประจำวันของเรานั้นสะดวกสบายมากยิ่งขึ้น โดยได้มีการพัฒนาเซนเซอร์โดยมีการวัดแบบ Real-Time sensor ให้มีการตรวจวัดอุณหภูมิ ความชื้นสัมพัทธ์และความชื้นในดิน โดยกลุ่มของข้าพเจ้าได้มีการสร้างระบบการรดน้ำโดยใช้เซนเซอร์วัดความชื้นในดิน แล้วมีการส่งข้อมูลที่เซนเซอร์สามารถวัดได้เข้าสู่ระบบ Database เพื่อให้เราสามารถเข้าถึงข้อมูลได้ตลอดเวลาผ่านอินเทอร์เน็ต รวมถึงมีการพัฒนาระบบแจ้งเตือนผ่านทาง Line Application ทำให้ชีวิตเราสะดวกสบายมากยิ่งขึ้น

1.2 วัตถุประสงค์

1) เพื่อศึกษาและประยุกต์ใช้งานไมโครคอนโทรลเลอร์ในด้านการเกษตร
2) เพื่อพัฒนาเทคโนโลยีที่นำมาใช้ในระบบเกษตรกรรมให้มีความสะดวกมากยิ่งขึ้น
3) เพื่อการแสดงผลต่อผู้ใช้งานให้มีความสะดวกมากยิ่งขึ้น

1.3 ขอบเขตของปริญญานิพนธ์

1) ใช้อุปกรณ์ Raspberry PI ควบคุมการทำงานของ sensor
2) สามารถใช้ sensor วัดความชื้นสัมพัทธ์ ความชื้นในดิน และอุณหภูมิได้
3) ทำระบบแจ้งเตือน Line Notification

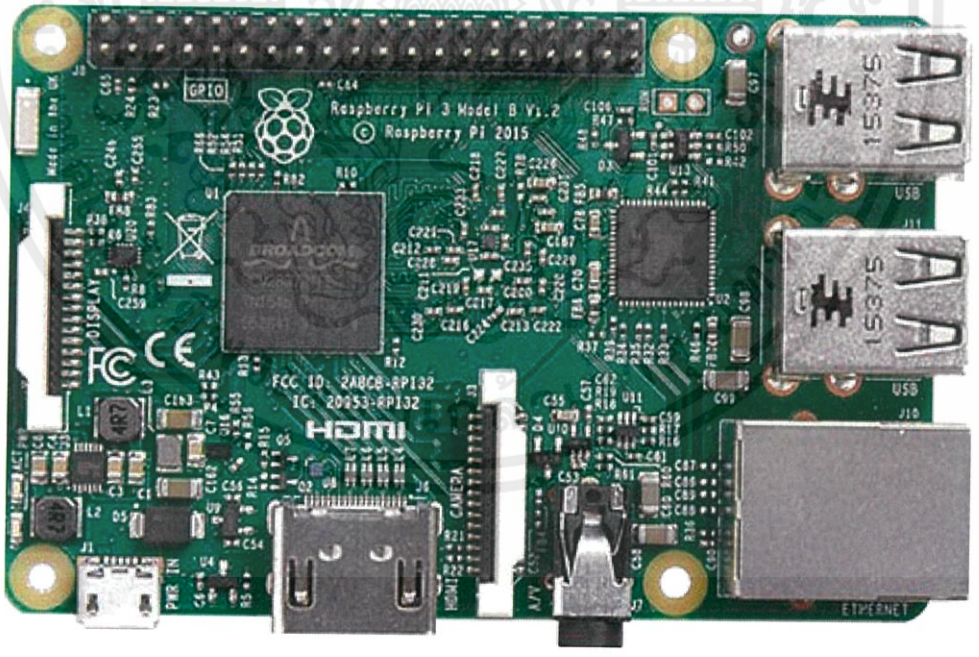
บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Raspberry Pi

Raspberry Pi เป็นเครื่องคอมพิวเตอร์ขนาดจิ๋ว ที่มีขนาดเพียงเท่ากับบัตรเครดิต ที่สำคัญคือมีราคาที่ถูกมาก เมื่อเทียบกับคอมพิวเตอร์ปกติ มีราคาเพียงแค่หนึ่งพันกว่าบาทเท่านั้น แต่เห็นราคาเท่านี้ ทำงานได้เหมือนเครื่องคอมพิวเตอร์ทุกอย่าง เราสามารถต่อราสเบอร์รี่พายนี้เข้ากับจอคอมพิวเตอร์หรือจอทีวีที่รองรับ HDMI หรือถ้าไม่มีพอร์ต HDMI ก็ไม่ต้องกังวล สามารถต่อผ่านสายสัญญาณวิดีโอปกติ (เส้นสีเหลือง) ได้เช่นกัน แต่ความละเอียดอาจจะต่ำกว่า

นอกจากต่อจอแสดงผลแล้ว ก็ต้องต่ออุปกรณ์รับข้อมูล ราสเบอร์รี่พายนี้รองรับเมาส์และคีย์บอร์ดผ่าน USB port ปกติ เพราะฉะนั้นสามารถนำเมาส์และคีย์บอร์ดที่มีอยู่แล้วมาต่อได้เลย ระบบจ่ายไฟของราสเบอร์รี่พายก็ง่ายมาก ๆ เพียงเสียบสาย Mini USB ที่เราใช้ชาร์จมือถือและอุปกรณ์อื่น ๆ เข้ากับคอมพิวเตอร์ หรือเข้ากับหัวชาร์จไฟมือถือก็ได้เช่นกัน

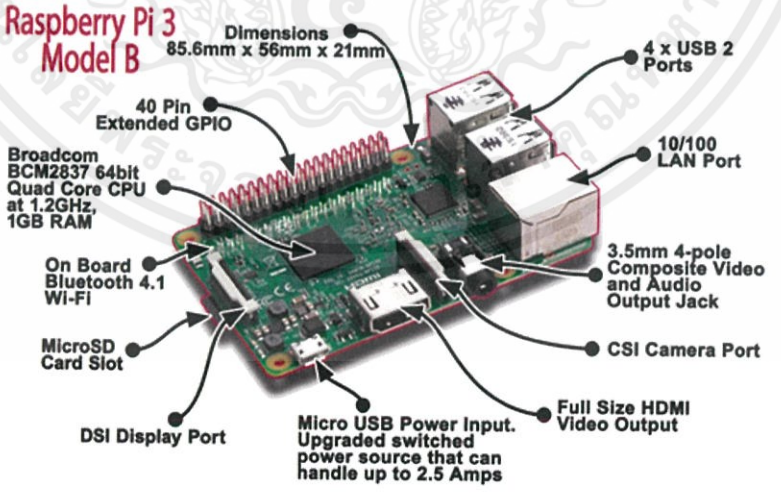


รูปที่ 2.1 บอร์ด Raspberry Pi 3 Model B

ราสเบอร์รี่พาย (Raspberry Pi) เกิดขึ้นในปี 2549 ที่มหาวิทยาลัยเคมบริดจ์ ประเทศอังกฤษ โดยผู้สร้างทั้งสี่คนคือ อีเบน อัพตั้น, ร๊อบ มูลลิ่นส์, แจ๊ค แลง และอลัน มายครอฟท์ มีจุดมุ่งหมายที่จะให้ราสเบอร์รี่พายเป็นคอมพิวเตอร์ราคาย่อมเยาที่ใคร ๆ ก็สามารถหามาครอบครองได้ และสามารถศึกษาการทำงานของคอมพิวเตอร์พร้อมทั้งเขียนโปรแกรมง่าย ๆ ได้ทันที การที่ราสเบอร์รี่พายเป็นบอร์ดวงจรรวมที่เปลือยเปล่า ทำให้เด็ก ๆ ได้เห็นชิ้นส่วนทั้งหมดที่เป็นส่วนประกอบของคอมพิวเตอร์ได้อย่างชัดเจน ซึ่งจะทำให้เข้าใจการทำงานของคอมพิวเตอร์ในปัจจุบันที่มาในกล่องสวยงามได้มากขึ้น

2.1.1 คุณสมบัติที่สำคัญของบอร์ด Raspberry Pi

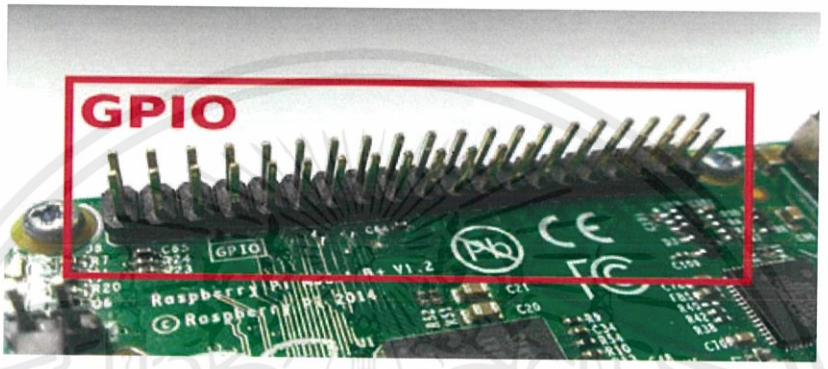
- CPU: Quad-core 1.2 GHz ARM Cortex-A53 แบบ 64 bits
- GPU: Broadcom VideoCore IV @ 400 MHz
- Memory ขนาด 1 GB (LPDDR2-900 SDRAM)
- หน่วยความจุแบบ MicroSD
- 4 USB ports
- 1 Ethernet port
- 802.11n Wireless LAN
- Bluetooth 4.0
- รองรับ HDMI/Composite ผ่านทาง RCA Jack
- GPIO 40 pins



รูปที่ 2.2 ส่วนประกอบของ raspberry pi 3 model B

2.1.2 การควบคุมอุปกรณ์อิเล็กทรอนิกส์

บนบอร์ด Raspberry Pi จะมีสิ่งที่เรียกว่า GPIO (General Purpose Input-Output) ซึ่งมันคือส่วนที่เอาไว้ต่อสัญญาณ Input/Output เข้ากับวงจรอิเล็กทรอนิกส์ภายนอกได้



รูปที่ 2.3 GPIO (General Purpose Input-Output)

การรับ/ส่งสัญญาณ Input/Output ผ่าน GPIO จำเป็นต้องเขียนโปรแกรมสั่งงาน โดยภาษาที่นิยมใช้กันทั่วไป ได้แก่ ภาษา Python แต่นอกจากภาษา Python แล้วก็ยังมีภาษาอื่น ๆ ให้เลือกใช้กันอีก เช่น C/C++, Shell Script และภาษาอื่น ๆ

2.2 ภาษา Python

ไพธอน (Python) คือ ภาษาระดับสูงที่ใช้ในการพัฒนาโปรแกรมอีกภาษาหนึ่งที่มีความสามารถสูงไม่แพ้ภาษาอื่น ๆ ที่มีอยู่ในปัจจุบัน ถูกสร้างขึ้นโดยนักพัฒนาโปรแกรมชื่อ Guido van Rossum เป็นชาวดัชท์ (Dutch) ประเทศเนเธอร์แลนด์ เกิดเมื่อวันที่ 31 มกราคม พ.ศ. 2499 ภาษาไพธอนได้รับอิทธิพลมาจากภาษา ABC ซึ่งมีความสามารถในการจัดการเกี่ยวกับข้อผิดพลาดของโปรแกรม (Exception handling) ได้ดี และดึงเอาความสามารถเด่น ๆ ของภาษาระดับสูงอื่น ๆ มาประยุกต์ดัดแปลงใช้กับไพธอนด้วย ส่งผลให้ภาษาไพธอนเป็นที่นิยม และใช้งานกันอย่างกว้างขวางในปัจจุบัน เนื่องจากเป็นภาษาที่สามารถเรียนรู้ได้ง่าย รวดเร็ว รูปแบบการเขียนโปรแกรมมีความกระทัดรัด และมีประสิทธิภาพสูง จากการนำเอาคุณลักษณะเด่น ๆ ของภาษาอื่น ๆ มาเป็นพื้นฐานในการพัฒนาต่อยอดนี้เอง ไพธอนจึงถูก เรียกว่าเป็นภาษาที่มีหลายกระบวนทัศน์หรือหลายมุมมอง (Multi-paradigm languages) ซึ่งเป็นการผสมผสานรวมเอา

แนวความคิดในการพัฒนาซอฟต์แวร์แบบต่าง ๆ เข้าไว้ด้วยกันให้อยู่ในตัวของไพธอน คือ Object-oriented programming, Structured programming, Functional programming and Aspect-oriented programming

ไพธอนถูกพัฒนาขึ้นมาโดยไม่ขึ้นกับแพลตฟอร์ม (Platform independent) กล่าวคือ สามารถทำงานได้ทั้งบนระบบปฏิบัติการตระกลูวินโดวส์ (Windows NT, 2000, 2008, XP, 7, 8, 8.1) ตระกูลยูนิกส์-ลินุกซ์ (Unix, Linux, xBSD) และตระกูลแมคด้วย (Macintosh) โดยระบบปฏิบัติการเหล่านี้ติดตั้งเพียงโปรแกรมแปลภาษาให้เป็นภาษาเครื่องของสถาปัตยกรรมนั้น ๆ เท่านั้น

ภาษาไพธอนเป็นซอฟต์แวร์เสรี (Open source software) เหมือนภาษาพีเอชพี (PHP) ทำให้ทุกคนสามารถนำไพธอนมาพัฒนาโปรแกรมได้ฟรีๆ โดยไม่ต้องเสียค่าใช้จ่าย และคุณสมบัติความเป็นซอฟต์แวร์เสรี ทำให้มีโปรแกรเมอร์ทั่วโลกเข้ามาช่วยกันพัฒนาให้ไพธอนมีความสามารถสูงขึ้นเรื่อย ๆ ส่งผลให้สามารถครอบคลุมงานในลักษณะต่าง ๆ อย่างกว้างขวาง สามารถสรุปคุณสมบัติและความสามารถของภาษาไพธอนได้ดังต่อไปนี้

2.2.1 คุณสมบัติเด่นของภาษาไพธอน

- โปรแกรมต้นฉบับที่ถูกเขียนขึ้นด้วยภาษาไพธอน สามารถนำไปประมวลผลได้กับหลายระบบปฏิบัติการ (Portable/Cross platform/Platform independent) เช่น Unix, Linux, Microsoft-windows เป็นต้น

- ตัวภาษาไพธอนถูกสร้างขึ้นมาจากภาษาซี ดังนั้นผู้ที่คุ้นเคยกับการเขียนโปรแกรมภาษาซีสามารถปรับตัวในเขียนภาษาไพธอนได้ไม่ยาก

- ภาษาไพธอนเป็นภาษาที่สวยงาม ง่ายต่อการเรียนรู้ (Readability) เขียนโปรแกรมได้กระชับ (Writability) เนื่องจากมีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย เป็นภาษาที่มีความยืดหยุ่นสูงมาก (Flexibility) และมีความเสถียรภาพ (Reliability)

- ไพธอนมีความสามารถในการจัดการหน่วยความจำอัตโนมัติ (Garbage collection) สามารถบริหารจัดการพื้นที่หน่วยความจำที่ใช้งานแบบไม่ต่อเนื่องให้สามารถทำงานได้อย่างมีประสิทธิภาพ ทำให้ผู้เขียนโปรแกรมไม่ต้องกังวลเกี่ยวกับการคืนหน่วยความจำคืนให้กับระบบเหมือนภาษาซี

- การแปลภาษาของภาษาไพธอนเป็นแบบอินเทอร์พรีเตอร์ เป็นภาษาสคริปต์คือจะประมวลผลไปทีละบรรทัด ทำให้ใช้เวลาในการเขียนโปรแกรม และการคอมไพล์ไม่มากเหมาะกับงานด้านการดูแลระบบ (System administration) เป็นอย่างยิ่ง

- ไวยากรณ์อ่านง่าย เนื่องจากภาษาไพธอนได้กำจัดการใช้สัญลักษณ์ที่ใช้ในการกำหนดขอบเขต {...} ของโปรแกรมออกไป

- ภาษาไพธอนถูกสร้างขึ้นโดยรวบรวมคุณสมบัติเด่น ๆ ของภาษาต่าง ๆ เข้ามาไว้ด้วยกันอาทิเช่น ภาษา C, C++, Java, Perl, ABC, Modula-3, Icon, Matlab, ANSI C, Lisp, Smalltalk และ Tcl เป็นต้น

- ไม่ต้องเสียค่าใช้จ่ายใด ๆ ทั้งสิ้น เพราะตัวแปรภาษาไพธอนอยู่ภายใต้ลิขสิทธิ์ GNU หรือซอฟต์แวร์เสรี

- ไพธอนมีฟังก์ชันที่สนับสนุนการเชื่อมต่อกับระบบฐานข้อมูลได้หลากหลายชนิด เช่น MySQL, Sybase, Oracle, Informix, ODBC และอื่น ๆ

- ภาษาไพธอนเป็นภาษาประเภท Server side script คือ การทำงานของภาษาไพธอนจะทำงานด้านฝั่งเซิร์ฟเวอร์ (Server) แล้วส่งผลลัพธ์กลับมายังไคล์เอนท์ (Client) ทำให้มีความปลอดภัยสูง

- ไพธอนเตรียมเครื่องมือสำหรับสร้าง Internet script หรือ CGI script สำหรับเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตผ่านซ็อกเก็ต (Sockets API) จึงทำให้สามารถเชื่อมต่อและใช้งานแอพพิเคชันต่าง ๆ แบบระยะไกลได้ เช่น FTP, Glopher, SSH เป็นต้น

2.2.2 คุณสมบัติด้อยของภาษาไพธอน

- ความเร็ว: ไพธอนเป็นภาษาสคริปต์(Scripting language) ซึ่งทำงานโดยมีตัวแปลภาษา (Interpreter) แปลงคำสั่งในแต่ละบรรทัดของโปรแกรมต้นฉบับ (Source code) ให้เป็นภาษาเครื่อง (Machine code) ในขณะที่โปรแกรมกำลังทำงาน ซึ่งแตกต่างจากภาษาซี C++ โคบอล หรือปาสคาล เพราะภาษาเหล่านี้จะทำการแปลรหัสต้นฉบับให้กลายเป็นภาษาเครื่องทั้งหมดก่อนเริ่มต้นทำงาน ส่งผลให้โปรแกรมขนาดใหญ่ที่เขียนขึ้นด้วยภาษาไพธอนจะทำงานได้ช้ากว่าโปรแกรมที่ใช้เทคนิคการคอมไพล์แฟ้มต้นฉบับทั้งหมดก่อน

- โอกาสเกิดข้อผิดพลาดชนิด Runtime Error สูงขึ้น: จุดด้อยในข้อนี้มีผลกระทบมาจากการแปลภาษาแบบ Interpreter และการไม่ได้แปลรหัสต้นฉบับทั้งหมดก่อนทำงานนั่นเองในการประกาศตัวแปรของภาษาสคริปต์ จะไม่มีการตรวจสอบความถูกต้องของ

การเรียกใช้ตัวแปร และชนิดของตัวแปรทั้งหมดก่อนเริ่มทำงาน ดังนั้นถ้าผู้พัฒนาโปรแกรมขาดความระมัดระวัง (Logic error) ในระหว่างพัฒนาโปรแกรม จะทำให้มีโอกาสเกิดความผิดพลาดจากการเรียกใช้ตัวแปรที่ไม่ได้ประกาศไว้หรือใช้งานตัวแปรผิดประเภทได้ง่าย

- การระบุขอบเขตของคำสั่ง และตัวแปร: ไพธอนไม่ใช้ {...} เป็นสัญลักษณ์สำหรับกำหนดขอบเขตของคำสั่ง หรือตัวแปรในการเขียนโปรแกรม เหมือนกับภาษาระดับสูง เช่น C/C++ และ Java แต่ใช้การย่อหน้าเพื่อบอกขอบเขตของคำสั่ง และตัวแปรแทน ส่งผลให้ยากต่อการพัฒนาโปรแกรมที่มีขนาดใหญ่ และโปรแกรมต้นฉบับมีความซับซ้อนมาก ๆ เช่น nested loop เพราะต้องสังเกตุการย่อหน้าให้ถูกต้อง

2.3 จาวาสคริปต์ (JavaScript)

ภาษาจาวาสคริปต์ (JavaScript) คือ ภาษาทางคอมพิวเตอร์ที่ใช้สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต จาวาสคริปต์เป็นภาษาสคริปต์เชิงวัตถุ (Object) ถูกใช้ในการสร้างและพัฒนาเว็บไซต์ร่วมกับภาษาเอชทีเอ็มแอล (HTML) เพื่อให้เว็บไซต์มีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น จาวาสคริปต์ (JavaScript) สามารถทำให้เว็บเพจมีลูกเล่นต่าง ๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม

เนื่องจากจาวาสคริปต์ช่วยให้ผู้พัฒนาสามารถสร้างเว็บเพจได้ตรงกับความต้องการจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง และได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของจาวาสคริปต์จะมีการแปลคำสั่งโดยเบราว์เซอร์ (Client-side Script) ดังนั้นจาวาสคริปต์จะสามารถทำงานได้เฉพาะบนเบราว์เซอร์ที่สนับสนุน ซึ่งปัจจุบันเบราว์เซอร์เกือบทั้งหมดก็สนับสนุนจาวาสคริปต์แล้ว

2.4 Firebase

Firebase คือ Platform ที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วนของ Backend หรือ Server Side ซึ่งทำให้สามารถ Build Mobile Application ได้อย่างมีประสิทธิภาพ และยังลดเวลาและค่าใช้จ่ายของการทำ Server Side หรือการวิเคราะห์ข้อมูลให้อีกด้วย โดยมีทั้งเครื่องมือที่ฟรีและเครื่องมีที่มีค่าใช้จ่าย (สำหรับการ Scale) โดยในส่วนของฟังก์ชันที่นำมาใช้คือ Realtime Database

รูปที่ 2.4 Product ทั้งหมดของ Firebase

2.4.1 คุณสมบัติของ Firebase

- API ในการเข้าถึงข้อมูลได้จาก Android, iOS, Java, JavaScript, Obj-C, Node.js
- REST API ที่มี libraries สำหรับคำสั่งทั่วไปของ JS frameworks
- ความสามารถในการ sync และ notifications ของข้อมูลที่มีการเปลี่ยนแปลง
- Cloud scaling สามารถรองรับการเข้าใช้งานได้พร้อม ๆ กันหลายคน
- สามารถเป็น web hosting, login/authentication

2.4.2 การจัดเก็บข้อมูลของ Firebase

มีการเก็บข้อมูลในรูปแบบของ map <key, value>

1). Value สามารถเป็นได้ทั้ง text, number, boolean, lists หรือ maps

2). Object สามารถจัดเก็บในรูปของ map from {field name => value}

3). List สามารถจัดเก็บในรูปของ map from {int index => value}

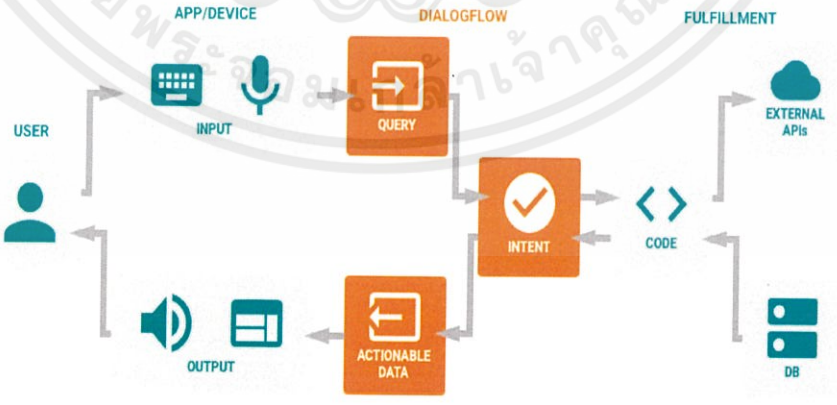
4). ฐานข้อมูลมีลักษณะคล้ายกับ tree-like map structure

2.4.3 Firebase Realtime Database

Firebase Realtime Database เป็น NoSQL Cloud Database ที่เก็บข้อมูลในรูปแบบของ JSON และมีการ Sync ข้อมูลแบบ Realtime กับทุก Devices ที่เชื่อมต่อแบบอัตโนมัติในเสี้ยววินาที รองรับ การทำงานเมื่อ Offline (ข้อมูลจะถูกเก็บไว้ใน Local จนกระทั่งกลับมา Online ก็จะทำการ Sync ข้อมูลให้อัตโนมัติ) รวมถึงมี Security Rules ให้เราสามารถออกแบบเงื่อนไขการเข้าถึงข้อมูลทั้งการ Read และ Write ได้ดังใจ ทั้ง Android, iOS และ Web

2.5 Dialogflow

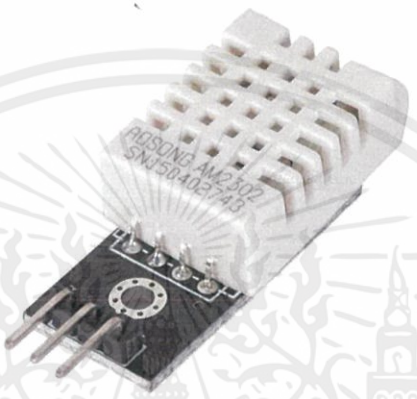
Dialogflow หรือ Api.ai เป็น Product ที่ถูกพัฒนาขึ้นโดย Speaktoit แต่ถูก Google ซื้อ และนำไปพัฒนาต่อยอดในปี 2016 และพึ่งเปลี่ยนชื่อมาเป็น Dialogflow จุดเด่นของ Dialogflow คือการรองรับการทำ Natural Language Understanding โดยที่แทบไม่ต้องเขียนโปรแกรมอะไรเพิ่มเติม สามารถแปลง Input หรือ Query ของผู้ใช้งานให้เป็น Intent โดยผ่านกระบวนการ Natural Language Processing ซึ่งจะช่วยให้แชทบอทสามารถหา Intent และทำ Entity Recognition ต่าง ๆ ได้โดยไม่ต้องเขียนโปรแกรมเพิ่ม โดย Dialogflow นั้นมีในส่วนของ Integrations มากมาย จึงได้ในมาใช้ประโยชน์ในการพัฒนาร่วมกับ Line Message API และ Fulfillment ร่วมกับ Realtime Firebase



รูปที่ 2.5 หลักการทำงานของเครื่องมือ Dialogflow

2.6 DHT22 Digital Temperature and Humidity Sensor

เป็นโมดูลวัดอุณหภูมิ (Temperature) และความชื้นสัมพัทธ์ (Humidity) โดยใช้ชิพ DHT22 ให้ Output ออกมาเป็นแบบ Digital Output ใช้ไฟ DC ขนาด 3 – 6 โวลต์ สามารถต่อกับบอร์ด Raspberry Pi ใช้งานได้ทันที แสดงดังรูปที่ 2.6



รูปที่ 2.6 Digital Temperature and Humidity Sensor (DHT22)

คุณสมบัติของ DHT22

- Size: 28mm X 12mm X 10mm
- Main chip: DHT22 temperature and humidity sensor
- Working voltage: DC 3.3-6 V
- Operating range humidity 0-100%RH; temperature -40~80Celsius
- Digital signal output
- Sensing period Average: 2s

2.7 Soil Moisture Sensor with Cable

Soil Moisture Sensor เป็น Sensor วัดความชื้นในดิน เพื่อใช้ในการตรวจวัดสำหรับการปลูกพืชหรือสวนดอกไม้ โดยเซนเซอร์ความชื้นในดินสามารถอ่านปริมาณความชื้นที่มีอยู่ในดินโดยรอบได้ เป็นเซนเซอร์เทคโนโลยีต่ำ แสดงดังรูปที่ 2.7

รูปที่ 2.7 Soil Moisture Sensor with Cable

คุณสมบัติของ Soil Moisture Sensor with Cable

- Size: 60x20x5mm
- Power supply: 3.3v or 5v
- Output voltage signal: 0~4.2v
- Current: 35mA
- Pin Definition:

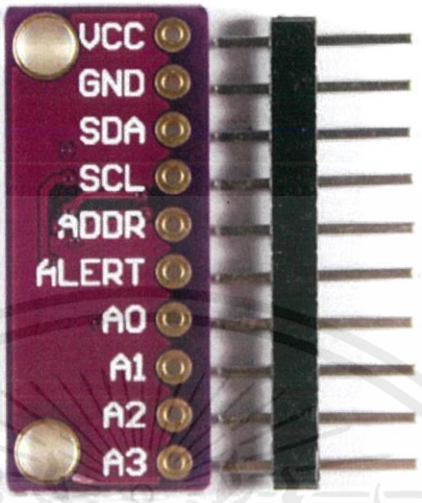
  Analog output (Blue wire)

  GND (Black wire)

  Power (Red wire)

2.8 ADS1115

โมดูล ADS1115 สำหรับรับและแปลงสัญญาณอนาล็อกเป็นดิจิตอล ความละเอียด 16-bit สั่งงานผ่านบัส I2C เปลี่ยนแอดเดรสเชื่อมต่อกันได้สูงสุด 4 โมดูลในบัสเดียว รองรับแรงดันตั้งแต่ 2 ถึง 5 โวลต์ เหมาะสำหรับไมโครคอนโทรลเลอร์ที่ไม่มี ADC (Raspberry Pi) หรือต้องการ ADC ความละเอียดสูงขึ้น (Arduino 10-bit ADC) แสดงดังรูปที่ 2.8



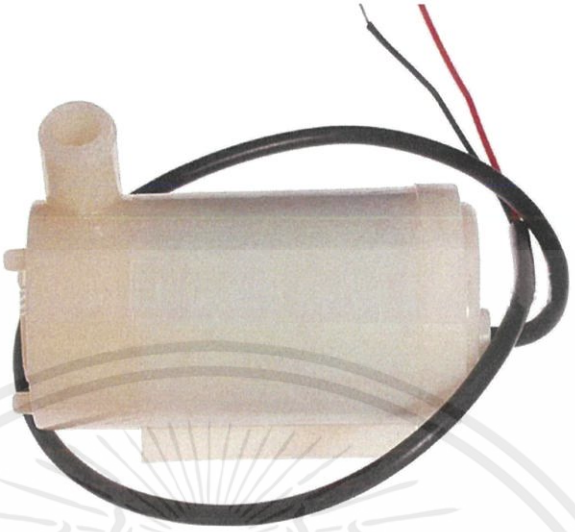
รูปที่ 2.8 ADS1115

คุณสมบัติของ ADS1115

- PCB size:(L*W)2.55*0.90cm/1.00"*0.35"
- Wide supply range: 2.0V to 5.5V
- Continuous Mode: Only 150μA
- Single-Shot Mode: Auto Shut-Down
- Internal low-drift voltage reference
- Internal oscillator
- I2C interface: Pin-Selectable Addresses
- Four single-ended or two differential inputs

2.9 Mini Pump Motor 5V

Mini Pump Motor คือปั๊มน้ำขนาดเล็กที่ใช้มอเตอร์ขนาด 5V ในการสูบน้ำขึ้นทางท่อยางไปยังภาชนะอื่น โดยการนำตัวปั๊มจุ่มลงไปในน้ำเพื่อสูบน้ำขึ้นมาใช้งาน แสดงดังรูปที่ 2.9



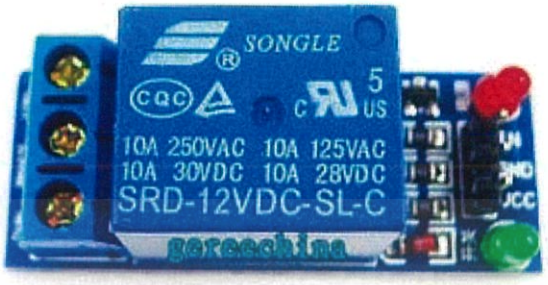
รูปที่ 2.9 Mini Pump Motor 5V

คุณสมบัติของปั๊มน้ำ

- Voltage: 2.5-6V
- Maximum lift: 40-110cm / 15.75"-43.4"
- Flow rate: 80-120L/H
- Driving mode: DC design, magnetic driving
- Continuous working life for 500 hours

2.10 Relay Module

โมดูลรีเลย์ 1ช่อง 5V (1 Channel Relay Module) เป็นโมดูลที่ใช้ควบคุมโหลดได้ทั้งแรงดันไฟฟ้า DC และ AC ซึ่งโหลดสูงสุด (Maximum Load) คือ AC 250V/10A, DC 30V/10A โดยใช้สัญญาณในการควบคุมการทำงานด้วยสัญญาณลอจิก TTL ทำงานด้วยสัญญาณแบบ Active Low, กระแสขับรีเลย์ (Drive Current) 15-20mA., มี LED แสดงสถานะ Power และ Relay สามารถนำไปประยุกต์ใช้งาน PLC Control, บ้านอัจฉริยะ, ใช้ในโรงงานอุตสาหกรรมหรืองานอื่น ๆ ขึ้นอยู่กับการเขียนโปรแกรมและการต่อใช้งานภายนอก สามารถเชื่อมต่อใช้งานกับบอร์ด Raspberry Pi, Arduino ได้



รูปที่ 2.10 1 Channel Relay Module

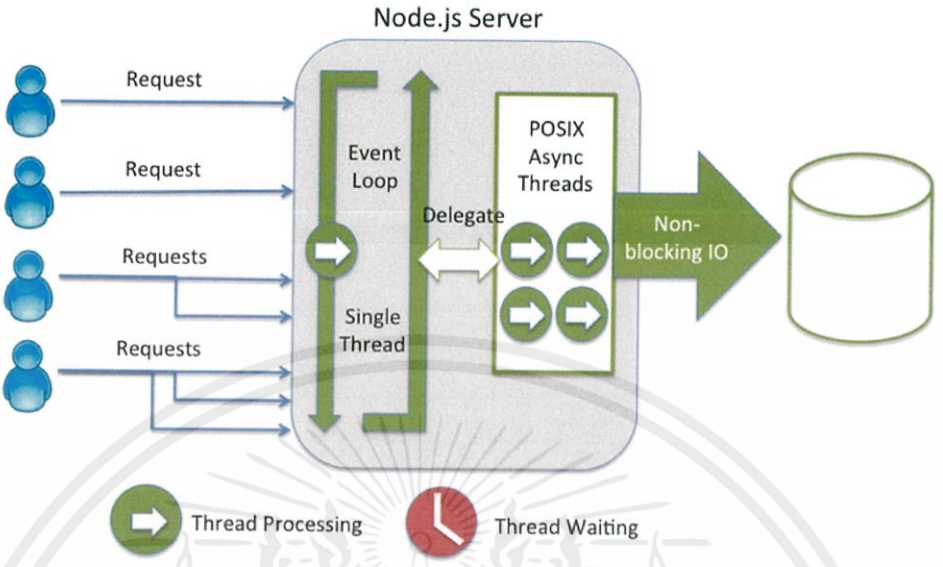
คุณสมบัติของรีเลย์

- ไฟเลี้ยงโมดูลรีเลย์ VCC = 5VDC
- ควบคุมโหลดได้ทั้งแรงดันไฟฟ้า AC ได้สูงสุด 250VAC 10A หรือ แรงดันไฟฟ้า DC ได้สูงสุด 30VDC 10A
- ระดับสัญญาณอินพุตควบคุมแบบ TTL ทำงานด้วยสัญญาณแบบ Active Low
- มี LED แสดงสถานะ Power และ Relay

2.11 โนดด็อทเจเอส (Node.js)

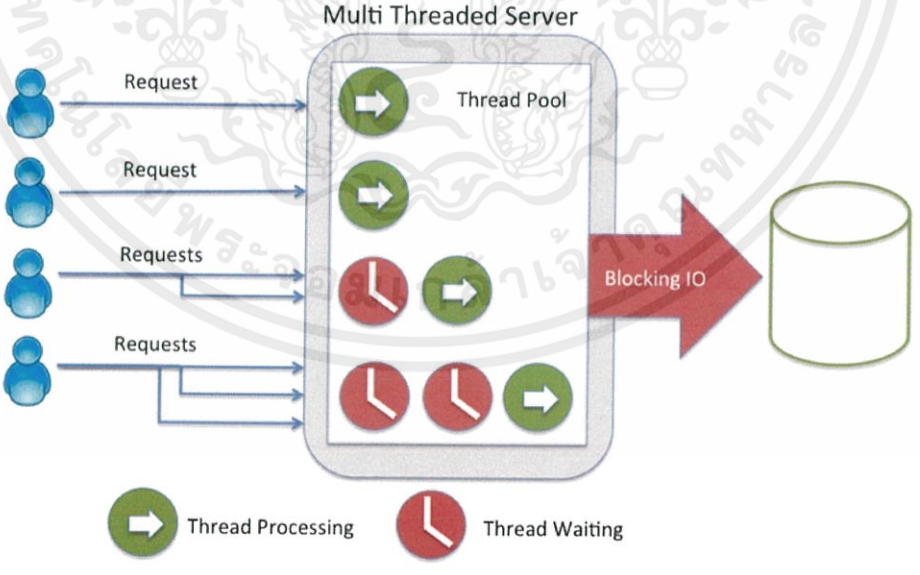
โนดด็อทเจเอส คือการเขียนโปรแกรมด้วยภาษาจาวาสคริปต์ (JavaScript) ที่ฝั่งเครื่องคอมพิวเตอร์แม่ข่าย (Server Side) แทนการเขียนที่ฝั่งผู้ใช้งาน (Client Side) โนดด็อทเจเอสมีความสามารถในการประมวลผลที่รวดเร็ว ใช้ได้กับระบบเครือข่ายขนาดใหญ่ รวมถึงรองรับการแสดงผลข้อมูลแบบเวลาจริงได้อย่างแท้จริง

โครงสร้างของการเขียนโปรแกรมภาษาโนดด็อทเจเอส มีความสามารถในการรองรับการเชื่อมต่อจากผู้ใช้งานจำนวนมาก (ระดับ 1,000 การเชื่อมต่อขึ้นไป) โปรแกรมจะทำการสร้างลูปของการเชื่อมต่อทั้งหมด แล้วแปลงให้เป็นการทำงานแบบไม่ต้องเรียงตามลำดับ (Asynchronous) และส่งการร้องขอในรูปของน็อนบล็อกกิ้งไอโอ (Non-blocking IO) ทำให้ไม่เกิดการรอคอยของภาระงาน (Thread Waiting) ซึ่งข้อได้เปรียบนี้ทำให้การเขียนโปรแกรมภาษาโนดด็อทเจเอสมีการทำงานที่เร็วกว่าแบบอื่นแสดงดังรูปที่ 2.11



รูปที่ 2.11 โครงสร้างการทำงานของแม่ข่ายโนดด็อทเจเอส (Node.js Server)

เมื่อทำการเปรียบเทียบกับการทำงานแบบปกติดังรูปที่ 2.12 แสดงให้เห็นว่าในการร้องขอจากผู้ใช้งานเข้าในปริมาณมาก (ปริมาณมากกว่า 1,000 การร้องขอขึ้นไป) เครื่องแม่ข่ายจะทำการจัดลำดับการทำงานขึ้นอยู่ในรูปของบล็อกกิ้งไอโอ (Blocking IO) ซึ่งจะทำให้เกิดการรอเวลาในการทำงานของแต่ละเทร็ด (Thread Waiting)



รูปที่ 2.12 โครงสร้างการทำงานแบบแม่ข่ายมัลติเทร็ด (Multi-Threaded Server)

2.12 Line Application

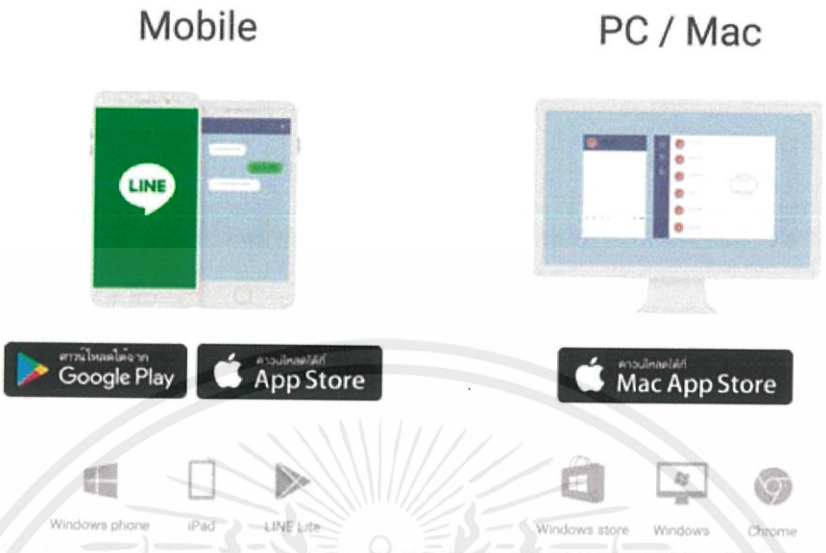
Line คือ แอพพลิเคชั่น ที่มีความสามารถในการสนทนา เช่น การแชท การส่งข้อความ การแชร์ไฟล์ การสร้างกลุ่มพูดคุย หรือการสนทนาผ่านเสียง ผ่านเครือข่ายอินเตอร์เน็ต บนอุปกรณ์ประเภทพกพา (Mobile Devices) เช่น สมาร์ทโฟน แท็บเล็ต เป็นต้น นอกจากนี้ Line ยังสามารถติดตั้งและใช้งานบนเครื่อง คอมพิวเตอร์ทั่วไปได้ด้วย

2.12.1 จุดเด่นของ Line

- ใช้งานง่าย
- เชื่อมต่อและใช้งานผ่านเครือข่ายอินเตอร์เน็ต
- สามารถโทรหาบุคคลที่อยู่ใน Friend List ในลักษณะเดียวกันกับการใช้งานโทรศัพท์ทั่วไป ผ่าน เครือข่ายอินเตอร์ โดยไม่ต้องเสียค่าใช้จ่ายเพิ่มในส่วนของการโทรออก
- สามารถสร้างกลุ่มสนทนาเฉพาะกลุ่มได้
- สามารถแชร์ไฟล์ภาพ เสียง วิดีโอ ตลอดจนข้อมูลสถานที่ได้
- สามารถถ่ายทอด อารมณ์ ความรู้สึก ในระหว่างการสนทนา ผ่าน Stickers แบบต่าง ๆ

2.12.2 ประโยชน์และการประยุกต์ใช้งาน Line

- การประยุกต์ใช้งานภายในองค์กรต่าง ๆ
- การประยุกต์ใช้งานในธุรกิจต่าง ๆ
- การประยุกต์ใช้งานในการติดต่อสื่อสารภายในครอบครัว หรือระหว่างกลุ่มเพื่อน
- สร้างโซเชียลเน็ตเวิร์คของคุณเอง
- รับข่าวสารล่าสุดกับบัญชีอย่างเป็นทางการของ LINE
- ใช้เป็นสื่อในการโฆษณา ประชาสัมพันธ์ หรือบริการลูกค้าสำหรับบริษัทและองค์กรต่าง ๆ



รูปที่ 2.13 แพลตฟอร์มที่แอพพลิเคชั่นรองรับในปัจจุบัน

2.13 ปัจจัยที่ควบคุมการเจริญเติบโตของพืช (Factors affecting plant growth)

ในการเจริญเติบโตของพืชนั้น จะถูกควบคุมโดยปัจจัยใหญ่ๆ 2 อย่าง คือพันธุกรรม (genetic factor) ปัจจัยทางด้านพันธุกรรมนี้ เป็นตัวควบคุมขนาด รูปร่าง สีสัน การให้ผลผลิต ความต้านทานต่อโรคและแมลง ฯลฯ ของพืช การที่พืชจะสามารถเจริญเติบโตได้สูงสุดแค่ไหนหรือให้ผลผลิตได้สูงสุดเพียงใดนั้นจะถูกควบคุมโดยยีน (gene) ซึ่งอยู่บนโครโมโซมภายในเซลล์

สภาพแวดล้อม (Environment factors) สภาพแวดล้อมเป็นอีกปัจจัยหนึ่งที่ควบคุมการเจริญเติบโตของพืชโดยทั่วไปแล้วสภาพแวดล้อมมีขอบเขตที่กว้างขวางมาก

2.13.1 แสง

แสงเป็นตัวที่ให้พลังงานแก่พืชเพื่อใช้ในขบวนการสังเคราะห์แสง ซึ่งเป็นขบวนการที่ก่อให้เกิดแป้งและน้ำตาล นอกจากนั้นแสงยังมีบทบาทสำคัญในขบวนการต่าง ๆ ในพืชอีกหลายอย่าง เช่น การสังเคราะห์โปรตีนการคายน้ำ การควบคุมทิศทางในการเจริญเติบโตของพืช ฯลฯ พืชแต่ละชนิดมีความต้องการแสงสำหรับการเจริญเติบโตในปริมาณมากน้อยแตกต่างกันแล้วแต่ชนิดของพืช

2.13.2 อุณหภูมิ

อุณหภูมิของดินและของบรรยากาศ มีส่วนเกี่ยวข้องกับการเจริญเติบโตของพืชคืออุณหภูมิเป็นตัวควบคุมขบวนการ metabolism ในพืช เช่น ขบวนการผลิตแป้ง ขบวนการให้พลังงานแก่พืช ขบวนการสร้างสารประกอบและขบวนการอื่น ๆ ในพืช กล่าวคือแต่ละขบวนการจะเกิดขึ้นได้ดีนั้น จะต้องมีอุณหภูมิที่พอเหมาะซึ่งโดยปกติจะอยู่ระหว่าง 15 – 40 องศาเซลเซียส ถ้าอุณหภูมิสูงหรือต่ำเกินไป ขบวนการ metabolismต่าง ๆ จะเกิดได้ช้า ซึ่งมีผลทำให้การเจริญเติบโตของพืชลดลงด้วย

2.13.3 ความชื้น (Moisture)

ความชื้นนับว่ามีความสำคัญต่อการเจริญเติบโตของพืชเป็นอย่างมาก เช่น ทำน้าที่ละลายธาตุอาหารพืช ลำเลียงธาตุอาหารพืช ควบคุมอุณหภูมิของต้นพืช เป็นต้น ดังนั้นจะเห็นได้ว่า พืชจะขาดน้ำสียไม่ได้ ถ้าพืชขาดน้าจะแสดงอาการเหี่ยวเฉา ขบวนการต่าง ๆ เกิดได้ช้า การเจริญเติบโตหยุดชะงัก และอาจตายในที่สุด

2.13.4 แก๊สในดินและในบรรยากาศ

แก๊สในดินและในบรรยากาศมีความสำคัญต่อการเจริญเติบโตของพืชหลายแง่ด้วย กันเช่น แก๊สออกซิเจนในดิน เป็นแก๊สที่จำเป็นต่อการหายใจของพืชเป็นอย่างยิ่ง เพราะการที่พืชจะเจริญเติบโตและดูดธาตุอาหารได้นั้น จะต้องได้รับพลังงานจากขบวนการหายใจ ดังนั้นในดินที่มีออกซิเจนไม่เพียงพอการเจริญเติบโตของรากพืชจะถูกจำกัด ซึ่งจะมีผลกระทบกระเทือนต่อการเจริญเติบโตของพืชทั้งต้น สำหรับในบรรยากาศ ออกซิเจน (O2) จะมีความจำเป็นต่อการหายใจของเซลล์พืชส่วนที่อยู่เหนือดินขึ้นมา และโดยปกติพืชจะไม่ค่อยขาด เนื่องจากในบรรยากาศมีแก๊สนี้อยู่ทั่วไป แก๊สในบรรยากาศที่นับว่ามีความสำคัญมากในขบวนการสังเคราะห์แสงของพืช

2.13.5 ปฏิกิริยาของดิน (Soil reaction)

ความเป็นกรดเป็นด่างของดินมีส่วนเกี่ยวข้องกับการเจริญเติบโตของพืช โดยที่ความเป็นกรดเป็นด่างของดินจะเป็นตัวควบคุมระดับความเป็นประโยชน์ของธาตุอาหารพืชในดิน ถ้าดินนั้นมีระดับความเป็นกรดเป็นด่างที่เหมาะสม จะมีธาตุอาหารละลายออกมาเป็น

ประโยชน์ต่อพืชได้มาก โดยทั่วไปธาตุอาหารพืชส่วนใหญ่จะละลายเป็นประโยชน์ต่อพืชได้มาก เมื่อดินมีค่า pH ระหว่าง 6.5 - 7.5

2.13.6 ศัตรูพืช

ศัตรูพืชมี 3 ชนิด คือ โรค แมลง และวัชพืช ซึ่งเป็นปัจจัยที่สำคัญอย่างหนึ่งที่มีผลต่อการเจริญเติบโตของพืชกล่าวคือ ถ้าพืชได้รับการรบกวนจากศัตรูพืชดังกล่าวอย่างใดอย่างหนึ่งหรือหลายๆอย่าง ก็จะทำให้การเจริญเติบโตและการให้ผลผลิตของพืชลดลง

2.13.7 สารพิษ

แก๊สบางชนิดอาจจะเป็นพิษต่อพืชได้ คืออาจทำให้พืชชะงักการเจริญเติบโตหรือตายได้ นอกจากนั้นยังมีสารประกอบบางชนิดซึ่งอาจเป็นพิษหรือทำให้พืชชะงักการเจริญเติบโตและอาจทำให้พืชถึงตายได้

2.13.8 ธาตุอาหารพืช

ธาตุอาหารพืชมีความสำคัญต่อพืชเช่นเดียวกับที่อาหารมีความสำคัญต่อสิ่งมีชีวิตทั่ว ๆ ไป กล่าวคือถ้าพืชได้รับธาตุอาหารที่จำเป็นครบทุกธาตุและในสัดส่วนที่เหมาะสมพืชก็จะมีการเจริญเติบโตได้ดีแต่ในทางตรงข้าม ถ้าพืชได้รับธาตุอาหารไม่เพียงพอหรืออยู่ในสัดส่วนที่ไม่เหมาะสมแล้ว ก็จะแสดงอาการผิดปกติหรือการเจริญเติบโตชะงักงันและอาจถึงตายในที่สุด

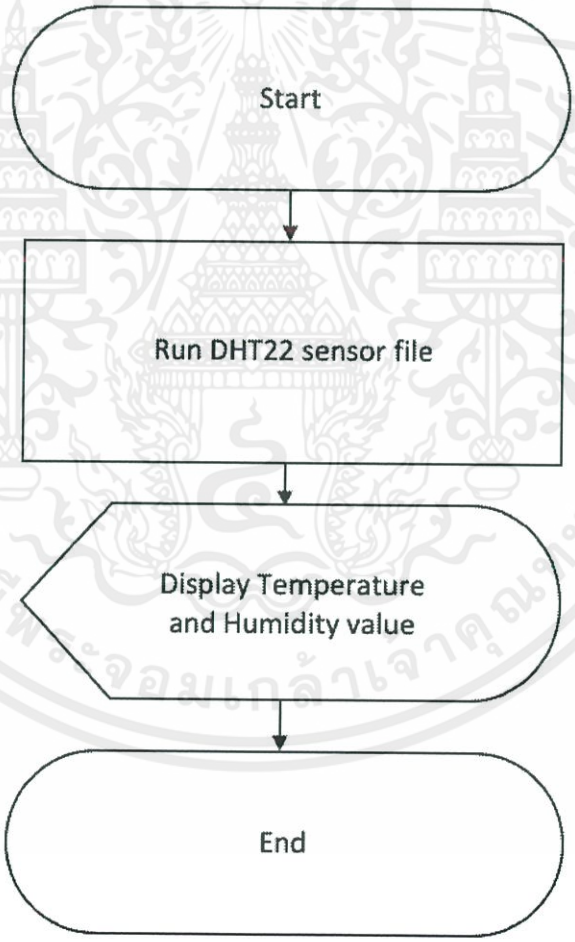
บทที่ 3

การออกแบบระบบแจ้งเตือนการรดน้ำต้นไม้

3.1 การออกแบบ

3.1.1 การออกแบบการวัดค่าอุณหภูมิและค่าความชื้นในอากาศ

ในส่วนของการวัดค่าอุณหภูมิและค่าความชื้นในอากาศ โดยจะใช้บอร์ด Raspberry pi เขียนคำสั่งให้เซนเซอร์ (DHT22) ทำงานวัดค่าอุณหภูมิและค่าความชื้นในอากาศ แล้วแสดงผลบนหน้าจอที่ต่อกับบอร์ด Raspberry pi แสดงดังรูปที่ 3.1



รูปที่ 3.1 Flow chart การวัดค่าอุณหภูมิและค่าความชื้นในอากาศ

ขั้นตอนของการต่อวงจร

1). DHT22 VCC ต่อกับ Raspberry pi 3.3v

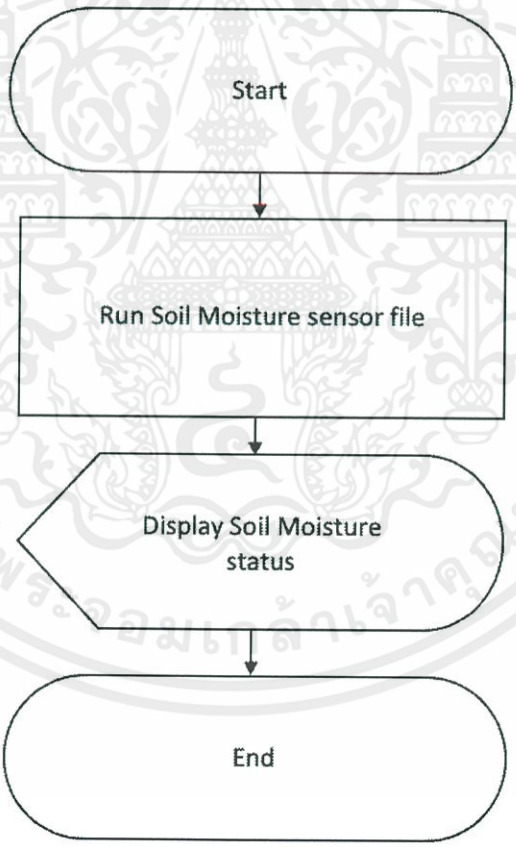
2). DHT22 GND ต่อกับ Raspberry pi GND

3). DHT22 Output ต่อกับ Raspberry pi pin24 (GPIO8)

4). ต่อตัวต้านทาน (10k ohm) เข้ากับ 3.3v และ pin24 (GPIO8)

3.1.2 การออกแบบการวัดค่าความชื้นในดิน

ในส่วนของการวัดค่าความชื้นในดิน โดยจะใช้บอร์ด Raspberry pi เขียนคำสั่งให้เซนเซอร์ (Soil Moisture Sensor) ทำงานวัดค่าความชื้นในดิน แล้วแสดงผลบนหน้าจอที่ต่อกับบอร์ด Raspberry pi แสดงดังรูปที่ 3.2



รูปที่ 3.2 Flow chart การวัดค่าความชื้นในดิน

ขั้นตอนของการต่อวงจร

1). ADS1115 VCC ต่อกับ Raspberry pi 3.3v

2). ADS1115 SDA ต่อกับ Raspberry pi pin3 (GPIO2)

3). ADS1115 SCL ต่อกับ Raspberry pi pin5 (GPIO3)

4). ADS1115 GND ต่อกับ Raspberry pi GND

5). Soil Moisture Output ต่อกับ ADS1115 pin A0

6). Soil Moisture VCC ต่อกับ Raspberry pi 3.3v

7). Soil Moisture GND ต่อกับ Raspberry pi GND

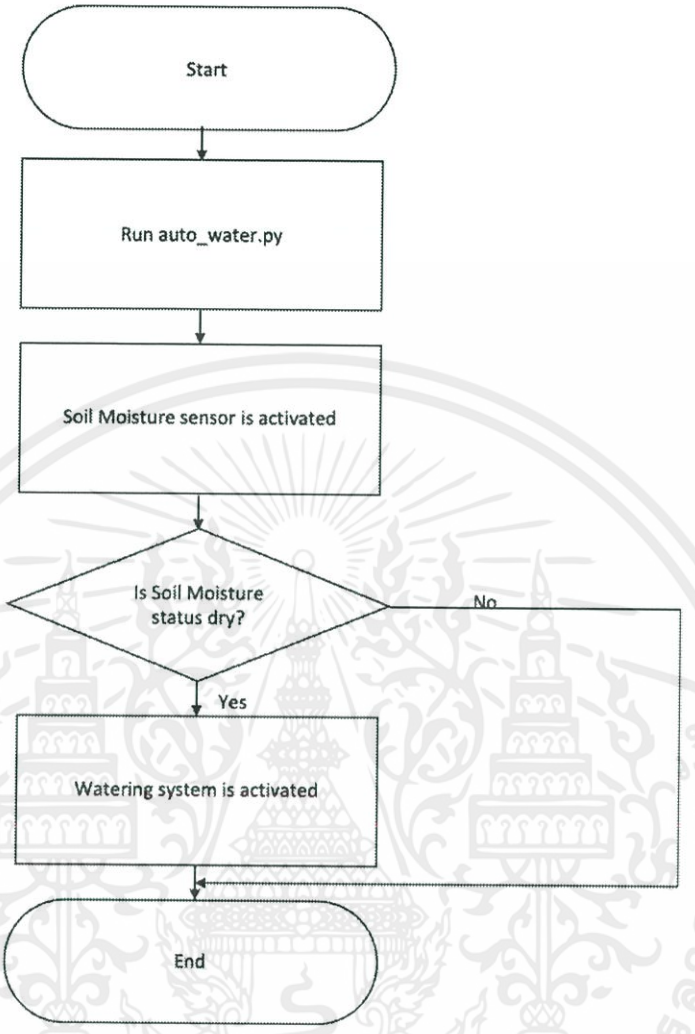
3.1.3 การออกแบบการทำงานของวงจรปั๊มน้ำ

ในส่วนของการทำงานของวงจรปั๊มน้ำ โดยจะใช้บอร์ด Raspberry pi เขียนคำสั่งให้เซนเซอร์ (Soil Moisture Sensor) ทำงานวัดค่าความชื้นในดิน แล้วแสดงผลบนหน้าจอที่ต่อกับบอร์ด Raspberry pi

โดยปั๊มน้ำจะทำงานหรือไม่ทำงานขึ้นอยู่กับสถานะความชื้นในดิน ที่เซนเซอร์ (Soil Moisture Sensor) ทำการวัดได้

- “Status = Dry” มอเตอร์ปั๊มน้ำจะทำงาน
- “Status = Humid” มอเตอร์ปั๊มน้ำจะไม่ทำงาน
- “Status = Wet” มอเตอร์ปั๊มน้ำจะไม่ทำงาน

มอเตอร์ปั๊มน้ำจะทำงาน โดยการสูบน้ำขึ้นทางท่อยางไปยังกระถางต้นไม้ เพื่อทำการรดน้ำ แสดงดังรูปที่ 3.3



รูปที่ 3.3 Flow chart การทำงานของวงจรปั๊มน้ำ

ขั้นตอนของการต่อวงจร

PUMP: 1). GND ของปั๊มน้ำกับ GND ของสาย USB ต่อเข้ากับ Relay (NO, COM ตามลำดับ)

2). สายขั้วบวกของปั๊มน้ำต่อเข้ากับขั้วบวกของสาย USB

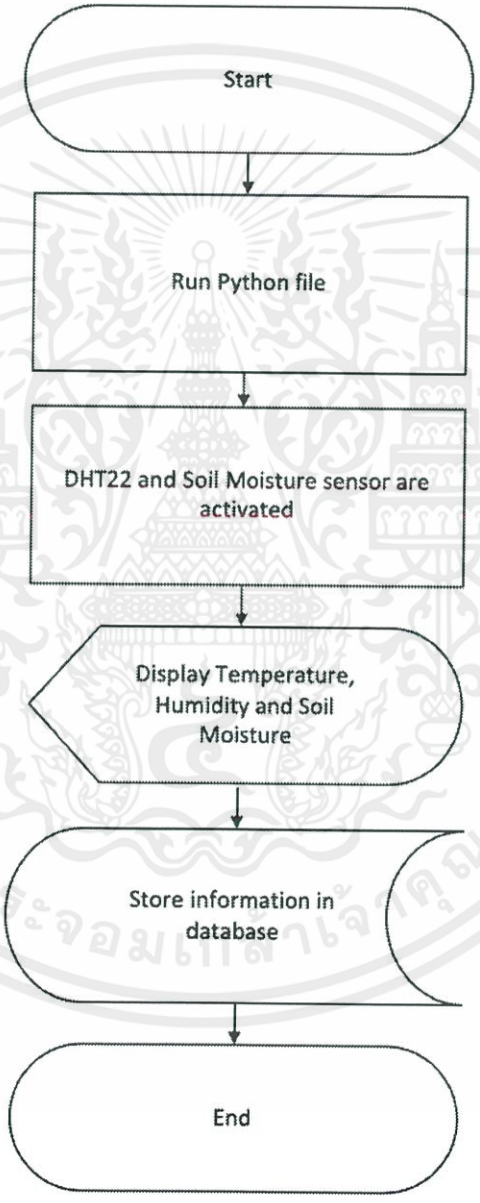
RELAY: 1). Input (IN) ต่อเข้า Raspberry pi pin7 (GPIO4)

2). VCC ต่อเข้า Raspberry pi 5V

3). GND ต่อเข้า Raspberry pi GND

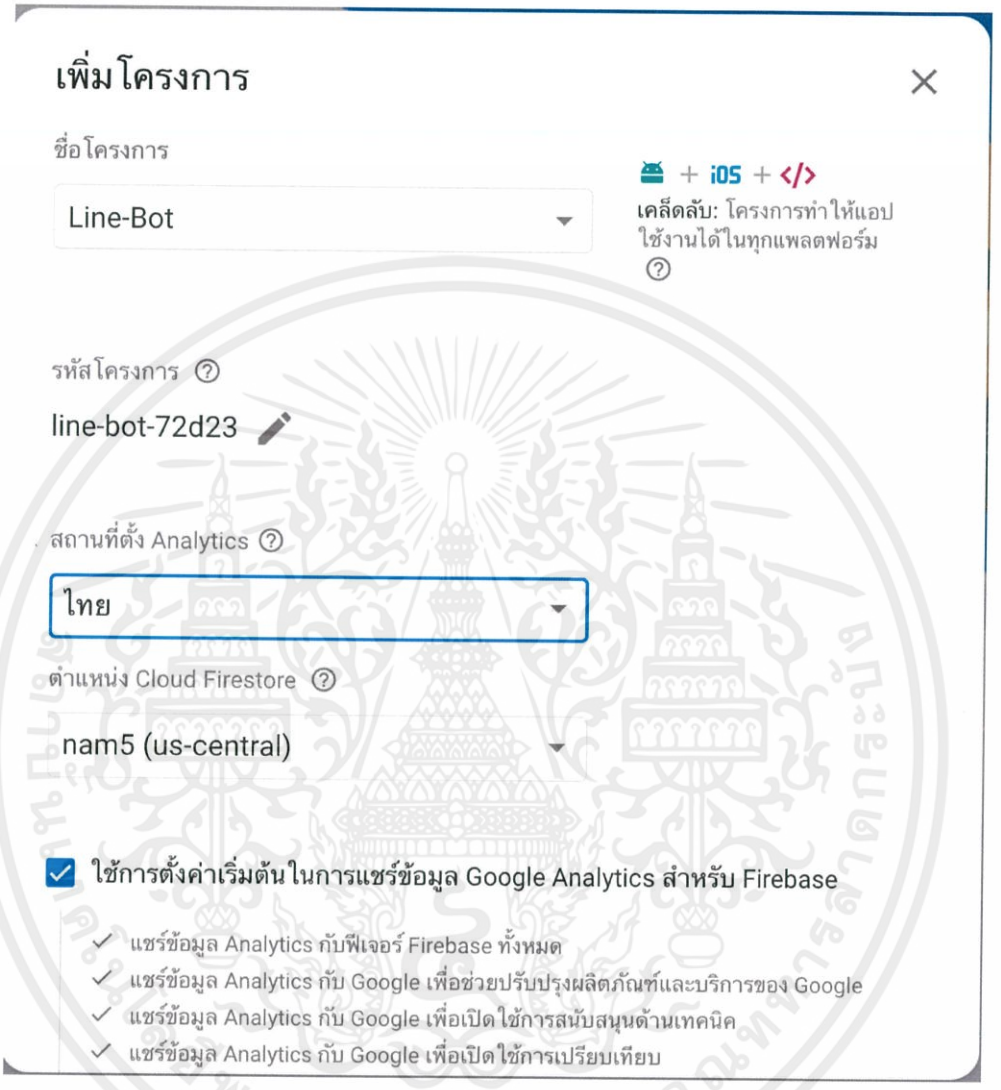
3.1.4 การออกแบบการรับค่าข้อมูลจากเซนเซอร์และบันทึกลงในฐานข้อมูล

ในส่วนของการรับค่าข้อมูลจากเซนเซอร์และบันทึกลงในฐานข้อมูล โดยจะใช้บอร์ด Raspberry pi เขียนคำสั่งให้เซนเซอร์ DHT22 และ Soil Moisture Sensor ทำงานวัดค่าอุณหภูมิ, ค่าความชื้นในอากาศ และค่าความชื้นในดิน แล้วแสดงผลบนหน้าจอที่ต่อกับบอร์ด Raspberry pi จากนั้นจะบันทึกค่าลงในฐานข้อมูลแสดงดังรูปที่ 3.4



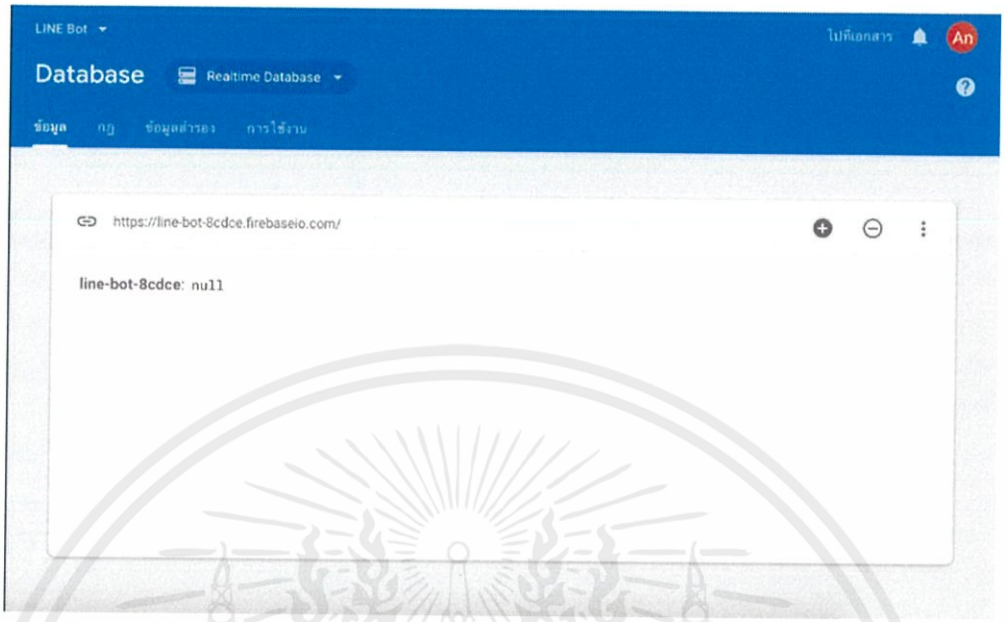
รูปที่ 3.4 Flow chart การรับค่าข้อมูลจากเซนเซอร์และบันทึกลงในฐานข้อมูล

3.1.4.1 ทำการสร้าง Firebase Project



รูปที่ 3.5 การสร้าง Firebase Project

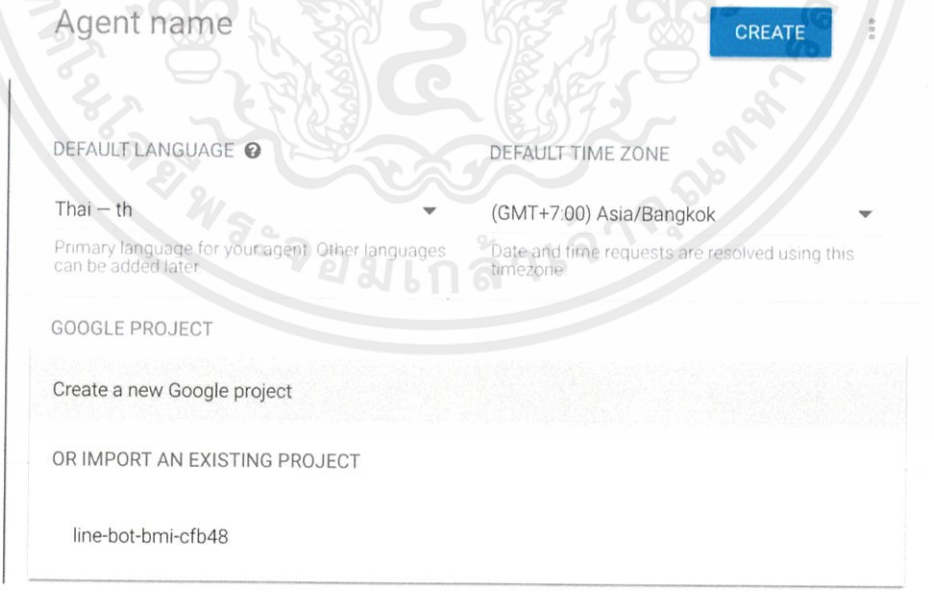
เมื่อสร้าง Firebase Project ดังรูปที่ 3.5 ขึ้นมาเรียบร้อยก็ทำการเลือกใช้บริการ Realtime Database แล้วรอรับค่าข้อมูลจากเซนเซอร์ DHT22 และ Soil Moisture Sensor ที่ส่งมาจากบอร์ด Raspberry Pi



รูปที่ 3.6 Realtime Database

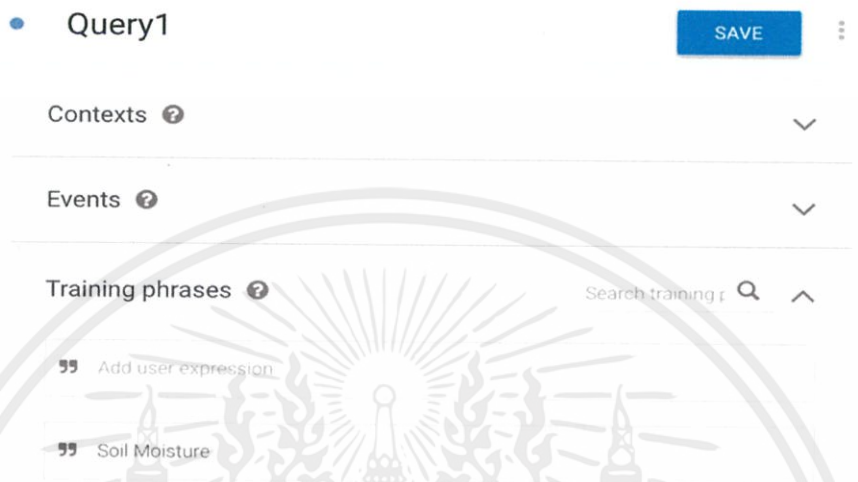
3.1.4.2 ทำการสร้าง Dialogflow Agent

สร้าง Dialogflow Agent แสดงดังรูปที่ 3.7 เมื่อเราสร้าง Agent ขึ้นมาจะมีส่วน Google Project ให้เราเลือก Firebase Project ที่ทำการสร้างในขั้นตอนที่แล้ว



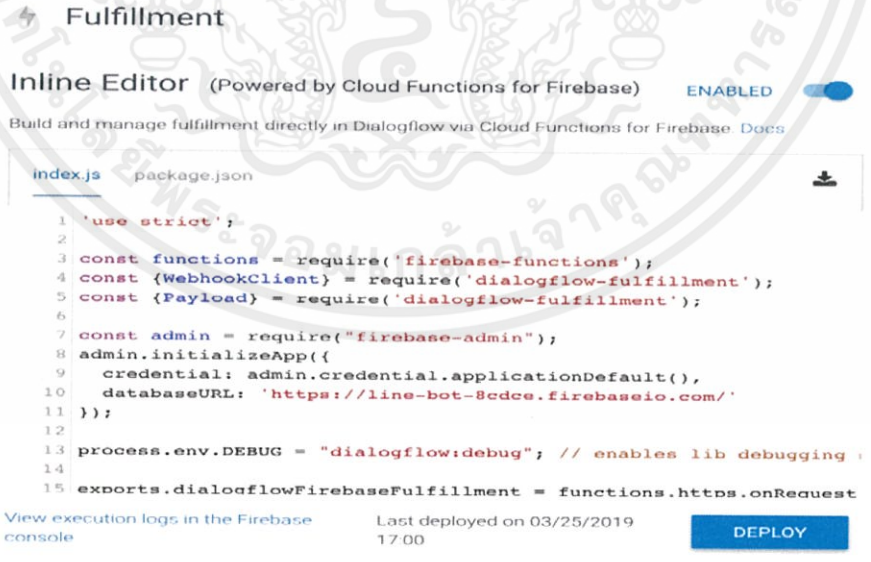
รูปที่ 3.7 การสร้าง Agent ของ Dialogflow และเชื่อมต่อกับ Firebase Project

จากนั้นทำการสร้าง Intent ขึ้นมาใหม่เพื่อให้เราสามารถ Query ข้อมูลจาก Firebase Realtime Database แสดงดังรูปที่ 3.8



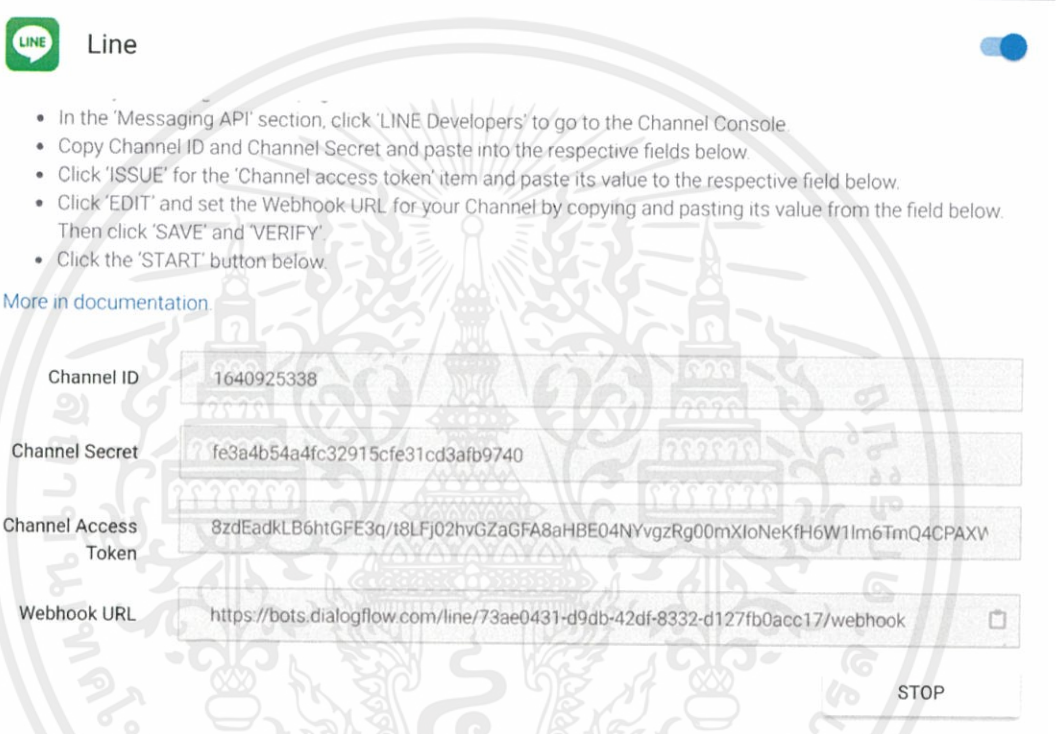
รูปที่ 3.8 การสร้าง Intent ขึ้นมาใหม่ในการรับคำสั่งเพื่อไป Query ข้อมูลบน Database

เมื่อสร้าง Intent เสร็จเรียบร้อยให้ไปทำการเขียนโปรแกรมเพิ่มในส่วนของ Fulfillment เลือกเปิดฟังก์ชัน Inline Editor แล้วเขียนโปรแกรมในช่อง index.js แสดงดังรูปที่ 3.9



รูปที่ 3.9 เปิดใช้งานคำสั่ง Fulfillment เพื่อเขียนโปรแกรม JavaScript

ในส่วนของการเชื่อมต่อกับ Line Bot ต้องไปทำที่ส่วนของ Integrations เลือกเปิดที่ Line Application และทำการใส่ข้อมูล Channel ID, Channel secret และ Channel access token ที่ได้จาก Line Bot ส่วน Webhook URL ทำการ Copy จาก Dialogflow ไปวางบน Line Bot เป็นการเชื่อมต่อกันระหว่าง Line Bot กับ Dialogflow แสดงดังรูปที่ 3.10

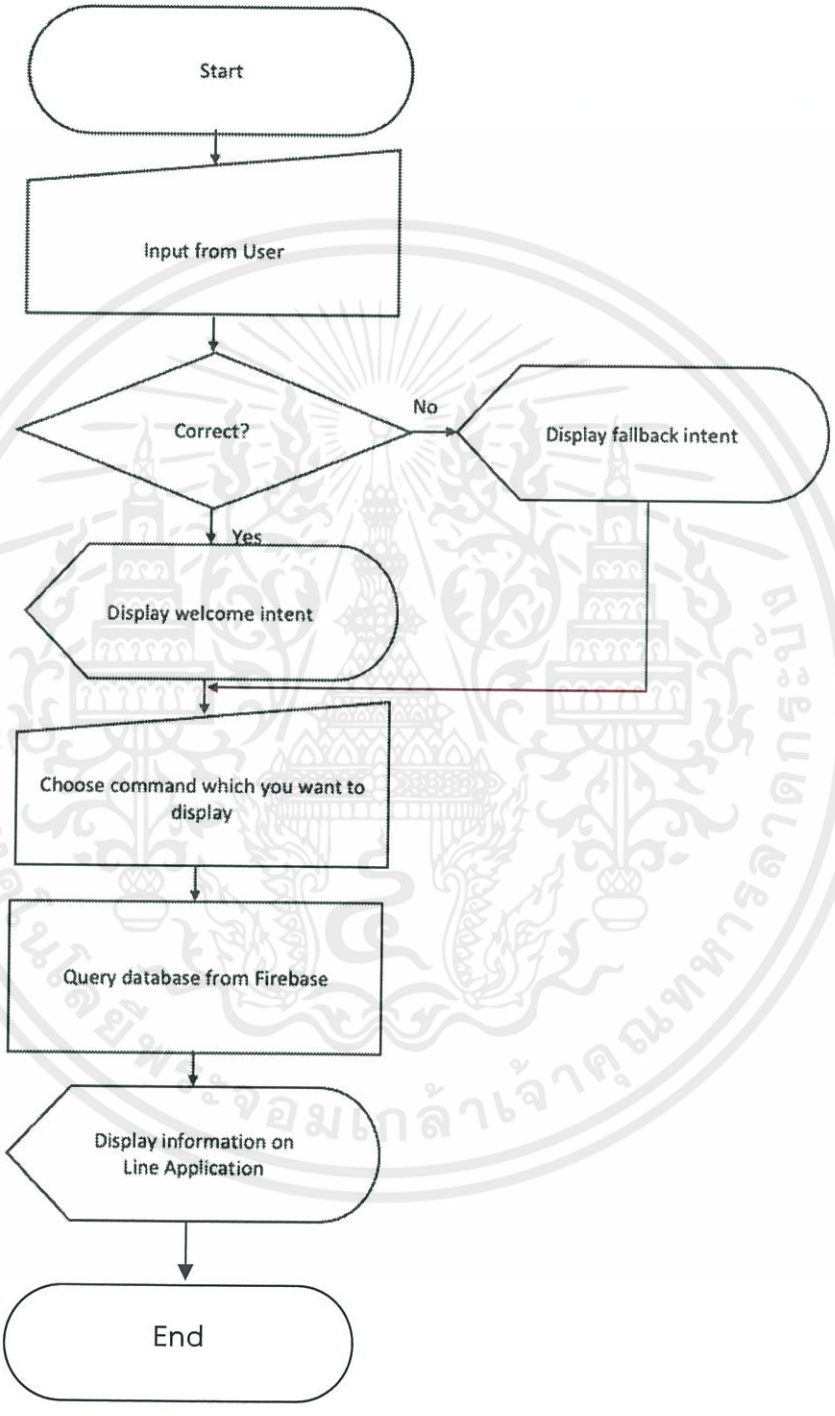


รูปที่ 3.10 Integrations เพื่อทำการเชื่อมต่อระหว่าง Dialogflow และ Line Bot

3.1.5 การออกแบบซอฟต์แวร์ในส่วนของการแจ้งเตือนผ่าน Line Application

ในส่วนของการออกแบบซอฟต์แวร์ในส่วนของการแจ้งเตือนผ่าน Line Application เมื่อเราเริ่มพิมพ์ข้อความลงไปในหน้าแชทของ Line Bot หากข้อความที่เราพิมพ์ลงไปเป็นข้อความที่ผ่านการ Training ใน Intent ที่สร้างขึ้น ทาง Dialogflow จะตอบกลับมาเป็นข้อความที่อยู่ในหมวด Default Welcome Intent ซึ่งจะเป็นข้อความทักทายพร้อมกับแสดง Commands เกี่ยวกับระบบรดน้ำต้นไม้ขึ้นมาแต่ถ้าข้อความที่พิมพ์ไปใน Line นั้นผิดพลาดหรือไม่ได้เป็นข้อความที่เรา Training ไว้ Dialogflow จะส่งข้อความตอบกลับในหมวด Default Fallback Intent กลับมาพร้อมกับแสดง Commands เกี่ยวกับระบบรดน้ำให้เราเลือก เมื่อเลือก

Command ที่ต้องการแล้วจะไปทำการเรียกขอข้อมูลจากฐานข้อมูลใน Firebase เพื่อส่งกลับมาที่หน้าแชท Line ของเรา



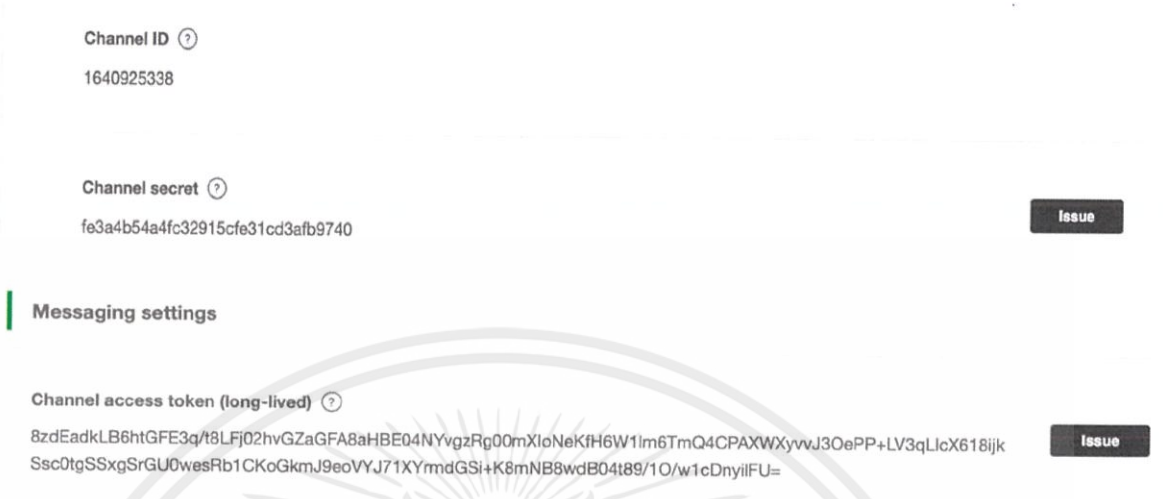
รูปที่ 3.11 Flow chart การออกแบบซอฟต์แวร์ในส่วนของการแจ้งเตือนผ่าน Line Application

3.1.5.1 ทำการสร้าง Line Bot

เข้าสู่เว็บไซต์ https://developers.line.me/en/ จากนั้นทำการเข้าสู่ระบบด้วย Line Account และทำการ Create Provider เมื่อเข้าสู่ระบบครั้งแรกสำเร็จต้องทำการตั้งชื่อ Provider Name และ Bot Name ให้เรียบร้อย แสดงดังรูปที่ 3.12

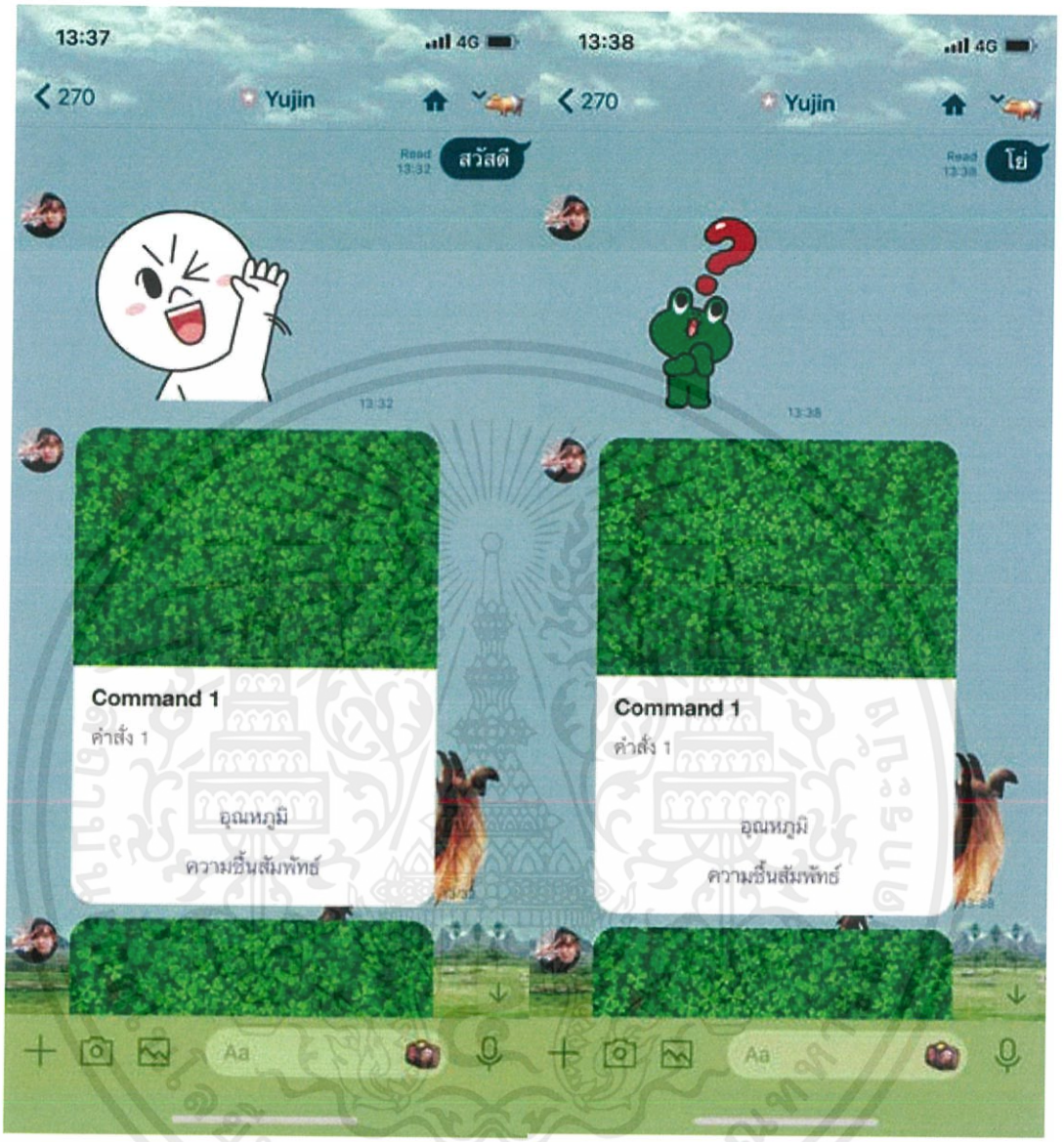
รูปที่ 3.12 Bot ที่สร้างขึ้นมาหลังจากที่สร้าง Provider

เมื่อเราสร้าง Bot สำเร็จเรียบร้อย จะได้ค่าที่นำไปใช้เชื่อมต่อกับ Dialogflow ในพาร์ทก่อนหน้านี้ แสดงค่าดังรูปที่ 3.13 และรูปที่ 3.14



รูปที่ 3.13 ค่า Channel ID, Channel secret และ Channel access token ที่จะต้องนำไปใช้เพื่อเชื่อมต่อกับ Dialogflow

รูปที่ 3.14 ค่า Webhook URL เป็นค่าจาก Dialogflow เพื่อนำมาใส่ใน Line Bot



รูปที่ 3.15 Bot กรณีตอบกลับด้วย Welcome Intent และ Fallback Intent

จากรูปที่ 3.15 เมื่อเราส่งข้อความ "สวัสดี" หรือข้อความทักทายอื่น ๆ ที่ผ่านการ Training ใน Intent Line Bot จะตอบกลับด้วย Welcome Intent เป็นสติ๊กเกอร์โบกมือทักทาย และแสดง Commands เกี่ยวกับระบบรดน้ำต้นไม้ขึ้นมา แต่ในกรณีที่เราส่งข้อความอื่น ๆ ที่ไม่ได้ผ่านการ Training Line Bot จะตอบกลับด้วย Fallback Intent เป็นสติ๊กเกอร์ทำท่าลักษณะงงและจะยังคงแสดง Commands ของระบบรดน้ำต้นไม้ขึ้นมาเพื่อความสะดวกและง่ายต่อผู้ใช้งาน

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Sensor

3.2.1.1 DHT22 Digital Temperature and Humidity Sensor

3.2.1.2 Soil Moisture Sensor with Cable

3.2.2 Microcontroller

3.2.2.1 Raspberry Pi 3

3.2.3 Analog to Digital converter

3.2.3.1 ADS1115

3.2.4 Hardware

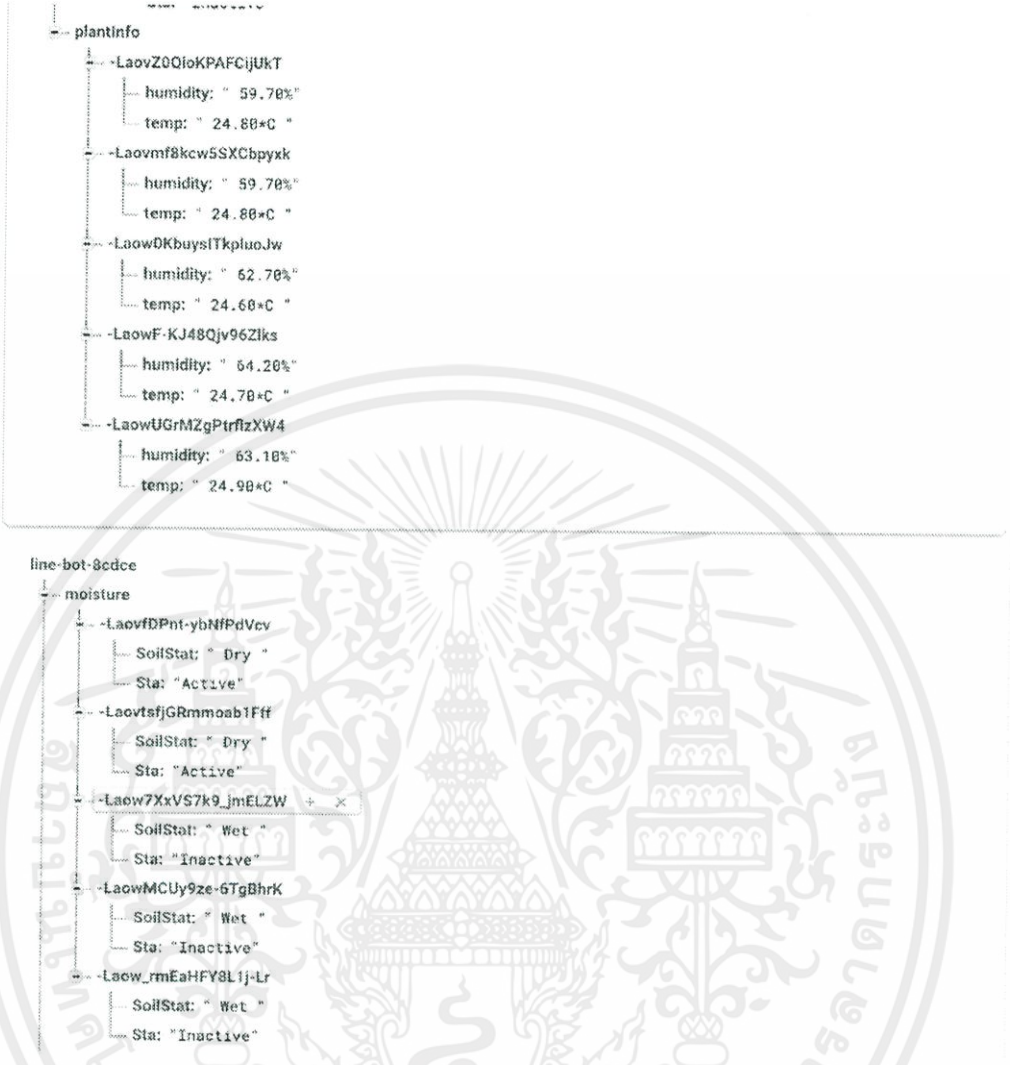
3.2.4.1 5V DC Pump Motor

3.2.4.2 5V Relay 1CH Module

3.2.4.3 Protoboard

3.3 การจัดเก็บผลการทดลอง

ในการเก็บผลการทดลองจะใช้เซนเซอร์จำนวนทั้งหมด 2 ตัวคือ DHT22 และ Soil Moisture Sensor ในการวัดค่าอุณหภูมิ, ความชื้นสัมพัทธ์ในอากาศและความชื้นในดินจากพืชที่ปลูก โดยได้ทำการทดสอบในสภาพแวดล้อมตามธรรมชาติและส่งค่าที่วัดได้จากเซนเซอร์เข้าไปเก็บในฐานข้อมูล Firebase แสดงดังรูปที่ 3.16 เพื่อรอให้ผู้ใช้งานเรียกใช้ข้อมูลผ่านแอพพลิเคชั่น Line ได้แบบเรียลไทม์



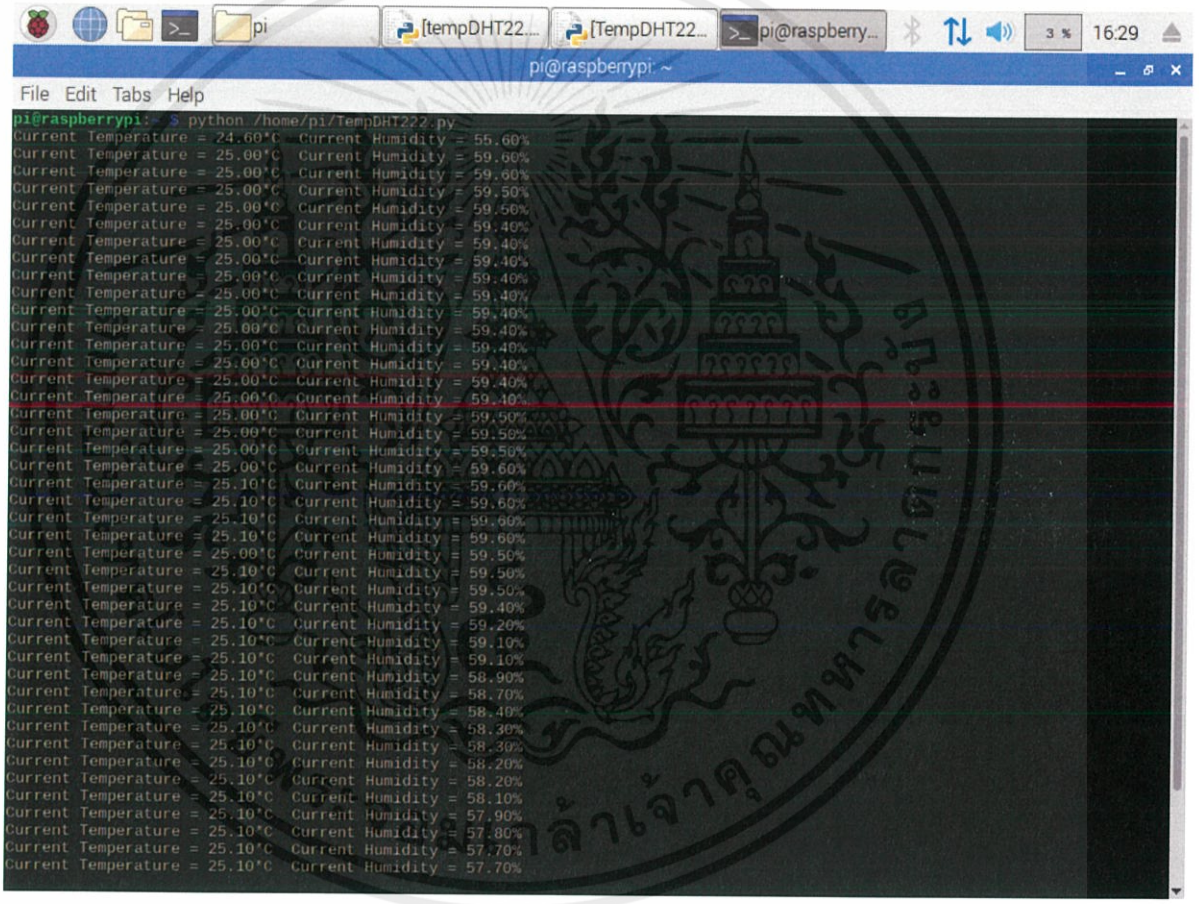
รูปที่ 3.16 ค่าที่วัดได้จากเซนเซอร์แล้วบันทึกลงในฐานข้อมูล Firebase

บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบการวัดค่าอุณหภูมิและค่าความชื้นในอากาศ

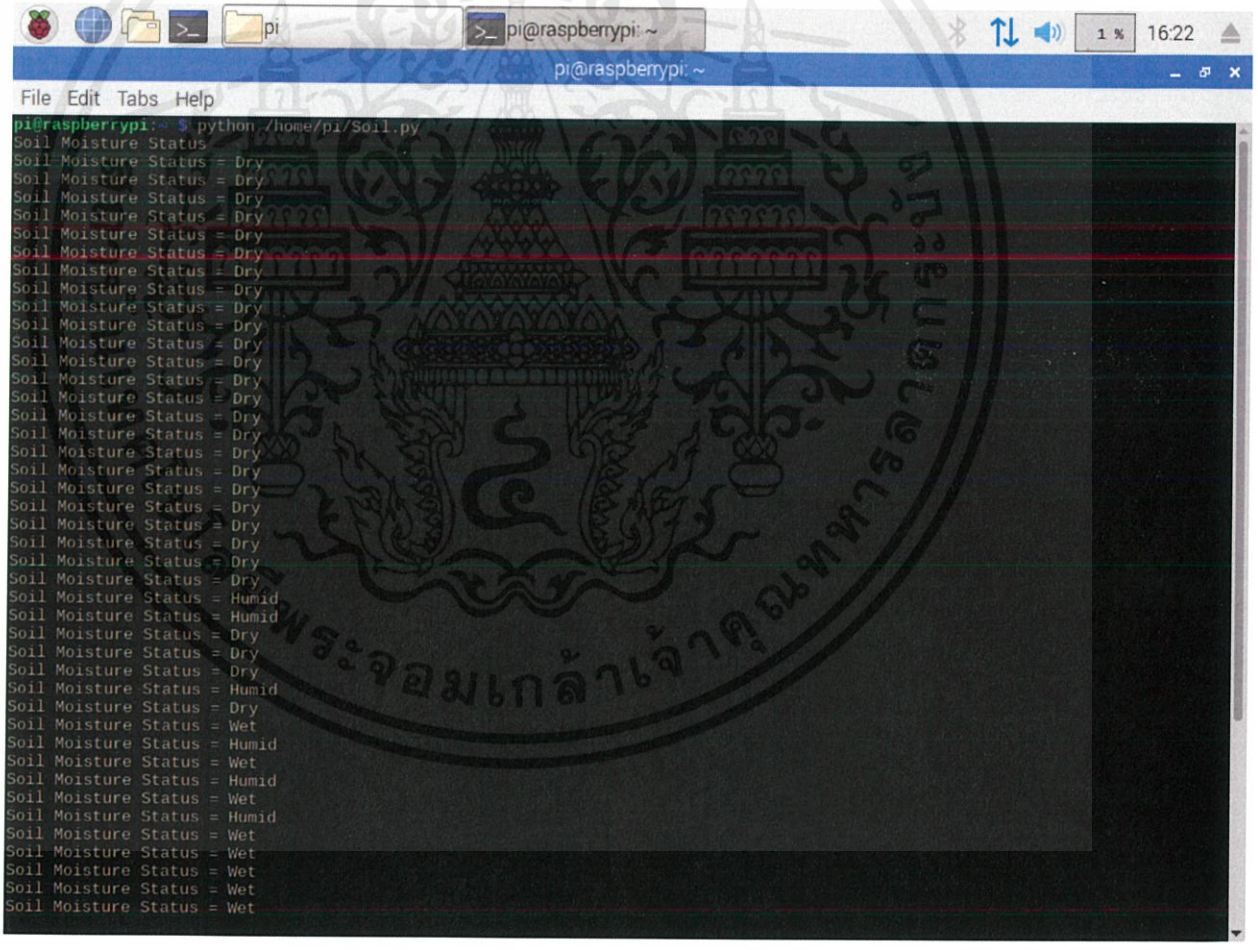
จากการทดสอบเซนเซอร์ โดยวัดอุณหภูมิและความชื้นสัมพัทธ์ในเวลาที่แตกต่างกัน ซึ่งผลที่ได้จากการทดสอบวัดในสภาพแวดล้อมที่ทำการปลูกพืชไว้ แสดงดังรูปที่ 4.1



รูปที่ 4.1 ค่าอุณหภูมิและค่าความชื้นในอากาศที่เซนเซอร์วัดได้

4.2 ผลการทดสอบการวัดค่าความชื้นในดิน

จากการทดสอบเซนเซอร์วัดความชื้นในดินนั้นได้ทำการแบ่งระดับความชุ่มชื้นในดินออกเป็น 3 ระดับ โดยได้ทดสอบกับพืชในสภาพแวดล้อมจริง ซึ่งได้แบ่งออกเป็นความชื้นระดับ “Dry” หรือดินมีความชุ่มชื้นน้อยมากหรือแห้งแล้ง โดยจะเป็นสภาพดินที่ไม่ได้ผ่านการรดน้ำเป็นเวลานานและถึงเวลาควรแก่การรดน้ำ สถานะนี้ระบบรดน้ำมีการทำงานโดยอัตโนมัติ, ความชื้นระดับ “Humid” หรือสภาพดินที่มีความชื้นอุดมสมบูรณ์ ผ่านการรดน้ำมาช่วงเวลาหนึ่ง ในสถานะนี้ระบบรดน้ำจะไม่มีการทำงานและสุดท้ายความชื้นระดับ “Wet” หรือดินที่มีสภาพเปียกหรือแฉะ สถานะนี้จะเกิดขึ้นหลังจากที่ดินได้ผ่านการรดน้ำเมื่อไม่นานมานี้และสถานะนี้ระบบรดน้ำจะไม่มีการทำงาน ซึ่งผลที่ได้จะแสดงเป็นแบบสถานะของความชื้น แสดงสถานะดังรูปที่ 4.2

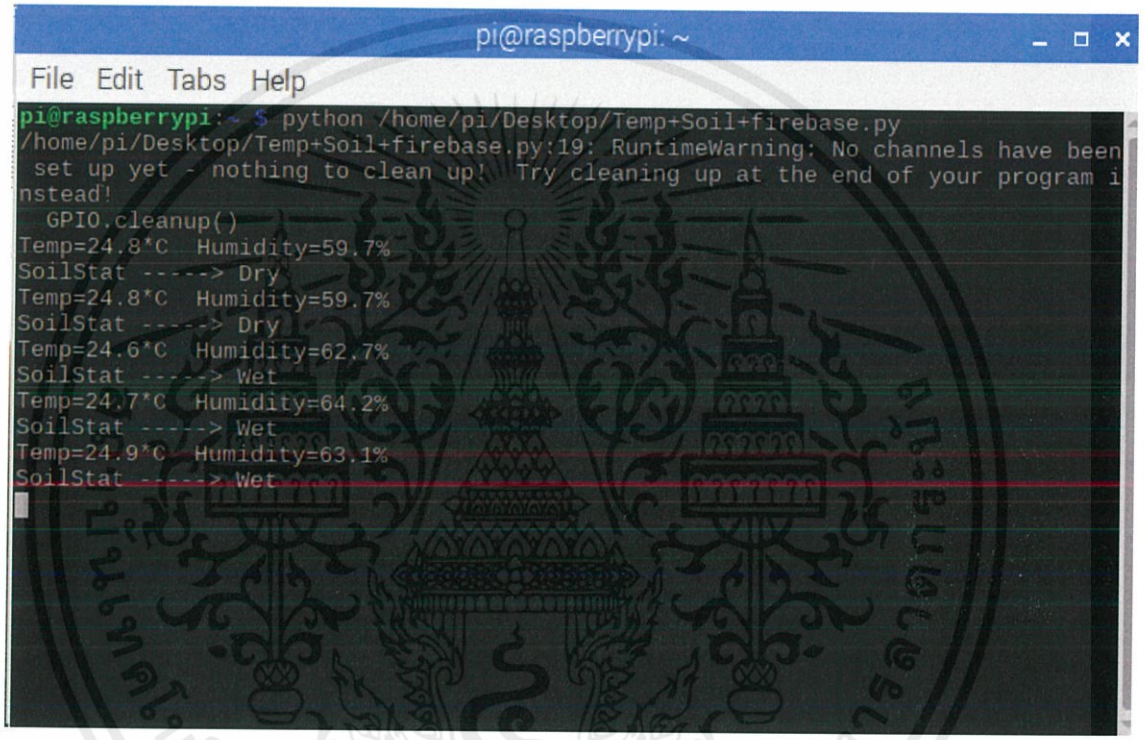


รูปที่ 4.2 ค่าความชื้นในดินที่เซนเซอร์วัดได้

4.3 ผลการทดสอบการทำงานบนฐานข้อมูล Firebase

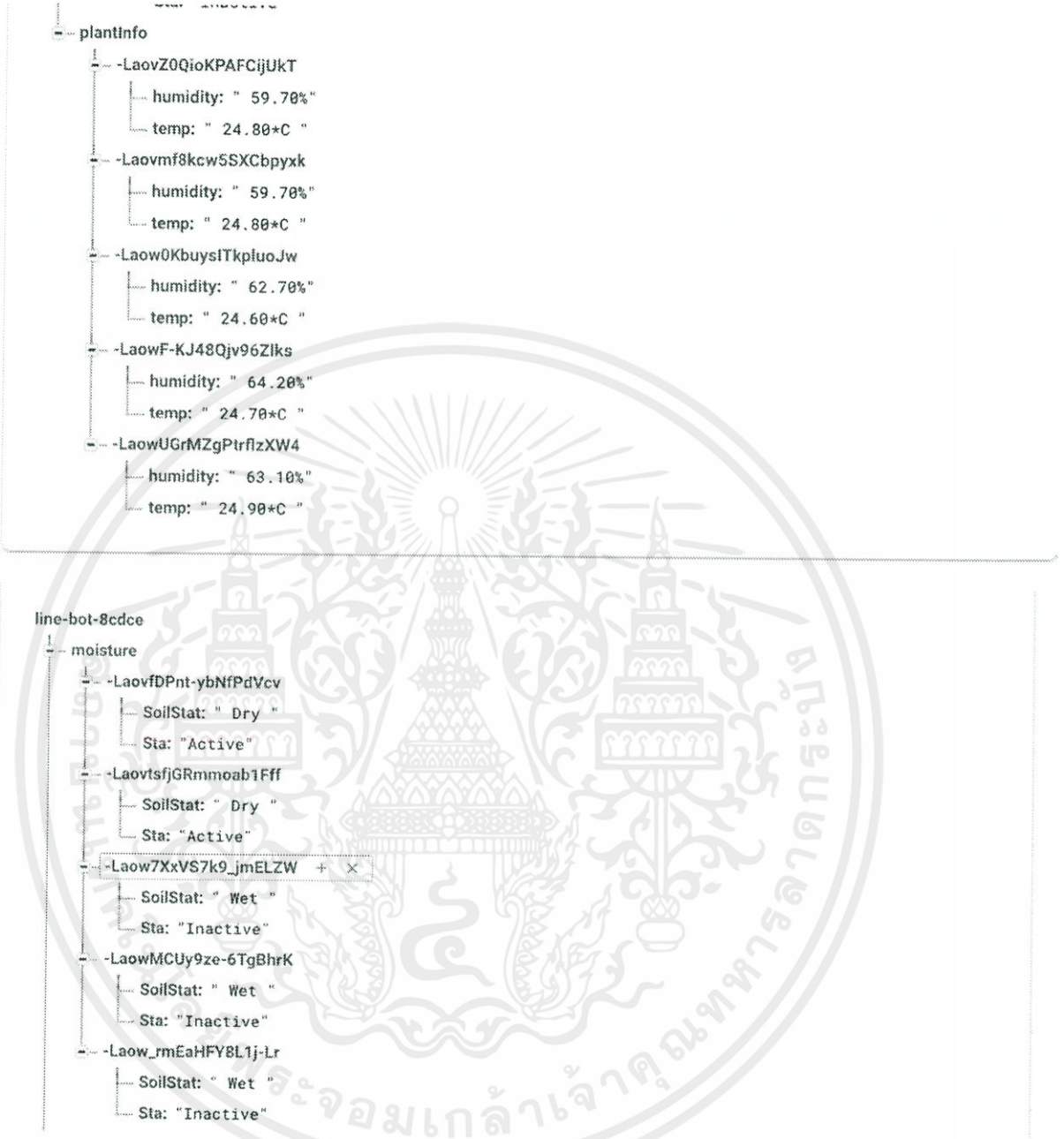
จากการทดสอบการทำงานบนฐานข้อมูล ซึ่งค่าที่ได้จากเซนเซอร์จะทำการส่งมาบันทึกลงบนฐานข้อมูล โดยจะมีทั้งหมด 3 ค่า ได้แก่ ความชื้นในดิน, อุณหภูมิของบริเวณระบบและความชื้นสัมพัทธ์บริเวณ

โดยจะแสดงค่าที่เซนเซอร์ทั้ง 2 สามารถวัดออกมาได้ดังรูปที่ 4.3



รูปที่ 4.3 ค่าอุณหภูมิ, ความชื้นสัมพัทธ์และความชื้นในดิน

จากนั้นจะบันทึกค่าอุณหภูมิ, ความชื้นสัมพัทธ์และความชื้นในดินลงในฐานข้อมูล (Firebase) แสดงดังรูปที่ 4.4



รูปที่ 4.4 ค่าอุณหภูมิ, ความชื้นสัมพัทธ์, ความชื้นในดินและสถานะการรดน้ำที่บันทึกในฐานข้อมูล

จากรูปที่ 4.4 บนฐานข้อมูลจะมีการเก็บข้อมูลแยกกันจากเซนเซอร์ 2 ตัวโดยใน /moisture จะทำการเก็บค่าความชื้นในดินจากเซนเซอร์ Soil Moisture Sensor และใน /plantInfo จะทำการเก็บค่าอุณหภูมิและความชื้นสัมพัทธ์ในอากาศจากเซนเซอร์ DHT22

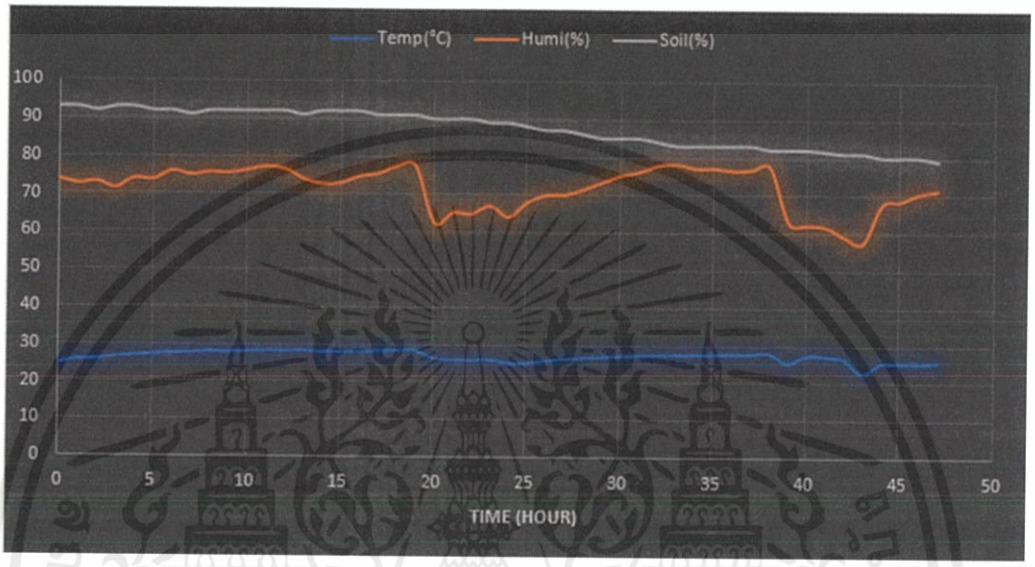
4.4 ผลการทดสอบบนแอพพลิเคชั่น Line

จากการทดสอบบนแอพพลิเคชั่น Line เป็นการเรียกดูข้อมูลจาก Firebase โดยได้ทำการสร้าง Line Bot และกำหนด Intent เพื่อใช้ในการขอเรียกดูข้อมูลผ่าน Dialogflow ที่เชื่อมต่อไว้กับ Line ซึ่งจะแสดงค่าที่ได้ทำการบันทึกลงในฐานข้อมูล ดังรูปที่ 4.5

รูปที่ 4.5 ค่าอุณหภูมิ, ความชื้นสัมพัทธ์และความชื้นในดินบน Line Application

4.5 ผลการทดสอบวัดอุณหภูมิ, ความชื้นสัมพัทธ์ และความชื้นในดิน

ทำการทดสอบวัดอุณหภูมิ, ความชื้นสัมพัทธ์ และความชื้นในดิน ณ ห้องปิด วัดทุก ๆ 1 ชั่วโมง เป็นเวลา 48 ชั่วโมง โดยไม่ได้ทำการรดน้ำเลย ผลการทดสอบที่ได้แสดงรูปที่ 4.6



รูปที่ 4.6 ค่าอุณหภูมิ, ความชื้นสัมพัทธ์และความชื้นในดินที่เปลี่ยนแปลงทุก ๆ 1 ชั่วโมง

จากกราฟเป็นการทดลองนำพืชมาปลูกในห้องปิดเป็นเวลา 48 ชั่วโมงโดยทำการวัดค่าทั้งหมด 3 ค่าได้แก่ค่าอุณหภูมิ, ค่าความชื้นสัมพัทธ์ และค่าความชื้นในดิน โดยจากค่าอุณหภูมิจะมีค่าค่อนข้างคงที่ ๆ บริเวณ 25 – 28 องศาเซลเซียสเนื่องจากทำการปลูกไว้ในห้องปิดซึ่งสามารถควบคุมอุณหภูมิได้ ค่าความชื้นสัมพัทธ์มีการเปลี่ยนแปลง โดยตามทฤษฎีค่าความชื้นสัมพัทธ์จะแปรผกผันกับค่าอุณหภูมิเมื่ออุณหภูมิสูงขึ้น ค่าความชื้นสัมพัทธ์จะต่ำลง ส่วนค่าความชื้นในดินมีค่าลดลงอย่างต่อเนื่อง เนื่องจากไม่ได้ทำการรดน้ำให้แก่พืช

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การทำปริญญานิพนธ์เรื่อง “ระบบแจ้งเตือนการรดน้ำต้นไม้” พบว่าสามารถทำงานได้จริง ถูกต้องและแม่นยำ โดยการทำงานของเซนเซอร์ทั้ง 2 ตัวไม่ว่าจะเป็นเซนเซอร์ที่ทำการวัดความชื้นในดินและเซนเซอร์ที่ใช้วัดอุณหภูมิและความชื้นสัมพัทธ์บริเวณสภาพแวดล้อมของระบบนั้น สามารถทำการส่งค่าต่าง ๆ ไปยังฐานข้อมูลเพื่อให้ผู้ใช้งานสามารถเรียกถามข้อมูลผ่านทางแอพพลิเคชั่น Line ได้ตลอดเวลาตามที่ผู้ใช้งานต้องการ สุดท้ายยังสามารถวัดค่าเพื่อสั่งให้ระบบรดน้ำต้นไม้รดน้ำ โดยอัตโนมัติเมื่อพืชมีความแห้งแล้งและถึงควรแก่เวลาในการรดน้ำ

จากการทดสอบวัดอุณหภูมิ, ความชื้นสัมพัทธ์ และความชื้นในดิน ณ ห้องปิด วัดทุก ๆ 1 ชั่วโมง เป็นเวลา 48 ชั่วโมง โดยไม่ได้ทำการรดน้ำเลย พบว่าผลที่ได้คือค่าอุณหภูมิจะค่อนข้างคงที่เนื่องจากทำการปลูกในห้องปิด ค่าความชื้นสัมพัทธ์มีการเปลี่ยนแปลง โดยตามทฤษฎีความชื้นสัมพัทธ์จะแปรผกผันกับอุณหภูมิ ส่วนค่าความชื้นในดินจะลดลงเมื่อไม่ได้ทำการรดน้ำให้แก่พืชเลย

5.2 ข้อเสนอแนะ

ในการทำปริญญานิพนธ์เรื่องนี้ได้มีการแบ่งส่วนงานใหญ่ๆเป็น 2 ส่วนคือส่วนจัดการเรื่องระบบรดน้ำต้นไม้และระบบการแจ้งเตือนมายังแอพพลิเคชั่น Line ซึ่งในส่วนนี้มีการเขียนโปรแกรมที่ใช้ภาษาแตกต่างกัน โดยระบบรดน้ำต้นไม้ที่ควบคุมโดยอุปกรณ์ Raspberry Pi ได้ใช้ภาษา Python เป็นภาษาหลักในการทำงาน ส่วนระบบที่ทำการแจ้งเตือนมายังแอพพลิเคชั่น Line ได้ใช้ Node.js และ JSON เป็นหลัก ซึ่งทั้งใน 2 ส่วนมีความซับซ้อนในการใช้งานจึงต้องทำการศึกษาเพิ่มเติม

บรรณานุกรม

[1] ผศ.ดร. สุชาติ คุ้มมะณี. “โปรแกรมภาษาไพธอน.” *ในเชี่ยวชาญการเขียนโปรแกรมด้วยภาษาไพธอน (Programming expert with Python)*. หน้า 18 - 22. 14 ตุลาคม 2558.

[2] นางสาววริศรา ปานพรม และ นางสาววิสาคร วงษ์พิมพา. *“ผลของระดับอุณหภูมิในบรรยากาศที่แตกต่างกันที่มีต่อปริมาณธาตุอาหารในดินและปริมาณผลผลิตของถั่วเหลือง (Glycine max (L.) Merrill).”* วท.บ. สาขาวิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม, มหาวิทยาลัยนเรศวร, 2556.

[3] firebase. database. “Reference.” https://firebase.google.com/docs/reference/js/firebase.database.Reference.

[4] firebase. database. “Query.” https://firebase.google.com/docs/reference/js/firebase.database.Query.

[5] firebase. database. “DataSnapshot.” https://firebase.google.com/docs/reference/js/firebase.database.DataSnapshot.

[6] LINE Developer. “Message API.” https://developers.line.biz/en/docs/messaging-api/overview/.

[7] LINE Developer. “Using quick replies.” https://developers.line.biz/en/docs/messaging-api/using-quick-reply/.

[8] Jirawatee. “เรียนรู้การ Integrate LINE Bot เข้ากับ Dialogflow และ Firebase ผ่าน BMI Bot.” https://medium.com/linedevth/เรียนรู้การ-integrate-line-bot-เข้ากับ-dialogflow-และ-firebase-ผ่าน-bmi-bot-5a30a672f6ae.

[9] Jirawatee. “เชื่อมต่อ Realtime DB และ Firestore จาก Dialogflow เพื่อให้ LINE Bot ของคุณเจ๋งกว่าเดิม.” https://medium.com/linedevth/เชื่อมต่อ-realtime-db-และ-firestore-จาก-dialogflow-เพื่อให้-line-bot-ของคุณเจ๋งกว่าเดิม-cd47b89eedcd.

ภาคผนวก

ชุดคำสั่งการทำงานของไมโครคอนโทรลเลอร์

File 1: DHT22.py

โค้ดคำสั่งการวัดอุณหภูมิและความชื้นสัมพัทธ์

```
import sys
import time
import Adafruit_DHT
sensor = Adafruit_DHT.DHT22
pin = 8
while True:
  humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
  if humidity is not None and temperature is not None:
    print 'Current Temperature = {0:0.2f}*C Current Humidity = {1:0.2f}%'
        .format(temperature, humidity)
    time.sleep(1)
  else:
    print 'Failed to get reading. Try again!'
```

File 2: Soil.py

โค้ดคำสั่งการวัดความชื้นในดิน

```
import time
import Adafruit_ADS1x15

adc = Adafruit_ADS1x15.ADS1115()

GAIN = 1

adc.start_adc(0,GAIN)
print('Soil Moisture Status')
start=time.time()
while True:
    value = int(round(adc.read_adc(0)/10.24))
    if value < 400:
        print('Soil Moisture Status = Dry'.format(value))
    elif value < 950:
        print('Soil Moisture Status = Humid'.format(value))
    else:
        print('Soil Moisture Status = Wet'.format(value))
    time.sleep(1)

adc.adc_stop()
```

File 3: water.py

โค้ดคำสั่งการควบคุมการทำงานของปั๊มน้ำ

```
import RPi.GPIO as GPIO
import datetime
import time
import Adafruit_ADS1x15
init = False
GPIO.setmode(GPIO.BOARD) # Broadcom pin-numbering scheme

def get_last_watered():
    try:
        f = open("last_watered.txt", "r")
        return f.readline()
    except:
        return "NEVER!"

def get_status(pin = 8):
    GPIO.setup(pin, GPIO.IN)
    return GPIO.input(pin)

def init_output(pump_pin = 7):
    GPIO.setup(pump_pin, GPIO.OUT)

def auto_water(delay = 5, pump_pin = 7, water_sensor_pin = 8):
    adc = Adafruit_ADS1x15.ADS1115()
    adc.start_adc(0,1)

    print("Here we go! Press CTRL+C to exit")
    try:
```

```
        while True:
            time.sleep(5)
            value = int(round(adc.read_adc(0)/20.48))
            print("Moisture Value : %s" %value)
            if value < 400:
                pump_on(pump_pin, 8)
            elif value < 950:
                print('SoilStat----->Humi')
            else:
                print('SoilStat----->Wet')

        adc.adc_stop()
    except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly:
        GPIO.cleanup() # cleanup all GPI

def pump_on(pump_pin = 7, delay = 1):
    f = open("last_watered.txt", "w")
    f.write("Last watered {}".format(datetime.datetime.now()))
    f.close()
    init_output(pump_pin)
    GPIO.output(pump_pin, GPIO.LOW)
    time.sleep(delay)
    GPIO.output(pump_pin, GPIO.HIGH)
    GPIO.cleanup()

def forever_water(pump_pin = 7, delay = 1):
    try:
        init_output(pump_pin)
        while True:
            pump_on(pump_pin)
```

```
except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly:
    GPIO.cleanup() # cleanup all GPI
```

File 4: auto_water.py

โค้ดคำสั่งให้รันโค้ด water.py

```
import water

if __name__ == "__main__":
    water.auto_water()
```

File 5: Temp+Soil+firebase.py

โค้ดคำสั่งการรับค่าข้อมูลจากเซนเซอร์และบันทึกลงในฐานข้อมูล

```
import RPi.GPIO as GPIO
import Adafruit_DHT
from time import sleep
import datetime
import time
import thread
import Adafruit_ADS1x15
from firebase import firebase

import urllib2, urllib, httplib
import json
import os
from functools import partial
```

```
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1

GPIO.setmode(GPIO.BCM)
GPIO.cleanup()
GPIO.setwarnings(False)

sensor = Adafruit_DHT.DHT22
pin = 8
humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

firebase = firebase.FirebaseApplication('https://line-bot-8cdce.firebaseio
.com/', None)
count = 0
type_ = "running.."

#tempory_Status = ""
#tempory_Motor  = ""

def combine(count,type_):
  if(count>0):
    adc.start_adc(0,GAIN)
    start=time.time()
    #while True:
    value = int(round(adc.read_adc(0)/10.24))
    if value < 400:
      str_val  = ' Dry '.format(value)
      str_P = 'Active'
      print('SoilStat -----> Dry'.format(value))
    elif value < 950:
      str_val  = ' Humid '.format(value)
```

```
                str_P = 'Active'
                print('SoilStat -----> Humid'.format(value))
            else:
                str_val  = ' Wet '.format(value)
                str_P = 'Inactive'
                print('SoilStat -----> Wet'.format(value))
            #tempory_Status = str_val
            #tempory_Motor  = str_P

            data = {"SoilStat": str_val, "Sta": str_P}
            firebase.post('/moisture', data)
            #adc.adc_stop()
            #time.sleep(3)

        else:
            humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
            if humidity is not None and temperature is not None:
                #sleep(3)
                str_temp = ' {0:0.2f}*C '.format(temperature)
                str_hum  = ' {0:0.2f}%'.format(humidity)
                print('Current Temperature = {0:0.1f}*C Current Humidity =
{1:0.1f}%'.format(temperature, humidity))
            else:
                print('Failed to get reading. Try again!')
                #sleep(3)
            data = {"temp": str_temp, "humidity": str_hum}
            firebase.post('/plantInfo', data)

    while True:
        try:
            count = count + 1
```

```
            triger= pow(-1,count)
            thread.start_new_thread(combine,(triger,type_))
            sleep(1)
        except:
            print "??"
```

File 6: index.js

โค้ดคำสั่งเพื่อเชื่อมต่อ Realtime database เข้ากับ dialogflow

```
'use strict';

const functions = require('firebase-functions');
const {WebhookClient} = require('dialogflow-fulfillment');
const {Payload} = require('dialogflow-fulfillment');

const admin = require("firebase-admin");
admin.initializeApp({
  credential: admin.credential.applicationDefault(),
  databaseURL: 'https://line-bot-8cdce.firebaseio.com/'
});

process.env.DEBUG = "dialogflow:debug"; // enables lib debugging statements

exports.dialogflowFirebaseFulfillment = functions.https.onRequest ((request, response) => {
    const agent = new WebhookClient({ request, response });

    function getSoilMoisture(agent) {
      return admin.database().ref("moisture")
```

```
    .orderByKey()
    .limitToLast(1)
    .on("child_added", function(snapshot) {
    var result = snapshot.val();
    agent.add("Soil Moisture Status : " + result.SoilStat);


  });
}
function getWateringSystem(agent) {
  return admin.database().ref("moisture")
    .orderByKey()
    .limitToLast(1)
    .on("child_added", function(snapshot) {
    var result = snapshot.val();
    agent.add("Watering System Status : " + result.Sta);


  });
}
```

```
function getTemperature(agent) {
 return admin.database().ref("plantInfo")
   .orderByKey()
   .limitToLast(1)
   .on("child_added", function(snapshot) {
   var result = snapshot.val();
   agent.add("Current Temperature : " + result.temp);




 });

}
 function getHumidity(agent) {
  return admin.database().ref("plantInfo")
   .orderByKey()
   .limitToLast(1)
   .on("child_added", function(snapshot) {
   var result = snapshot.val();
   agent.add("Current Humidity : " + result.humidity);




 });

}
```

```
let intentMap = new Map();
intentMap.set("Query1", getSoilMoisture);
intentMap.set("Query2", getWateringSystem);
intentMap.set("Query3", getTemperature);
intentMap.set("Query4", getHumidity);

agent.handleRequest(intentMap);

  }

);
```