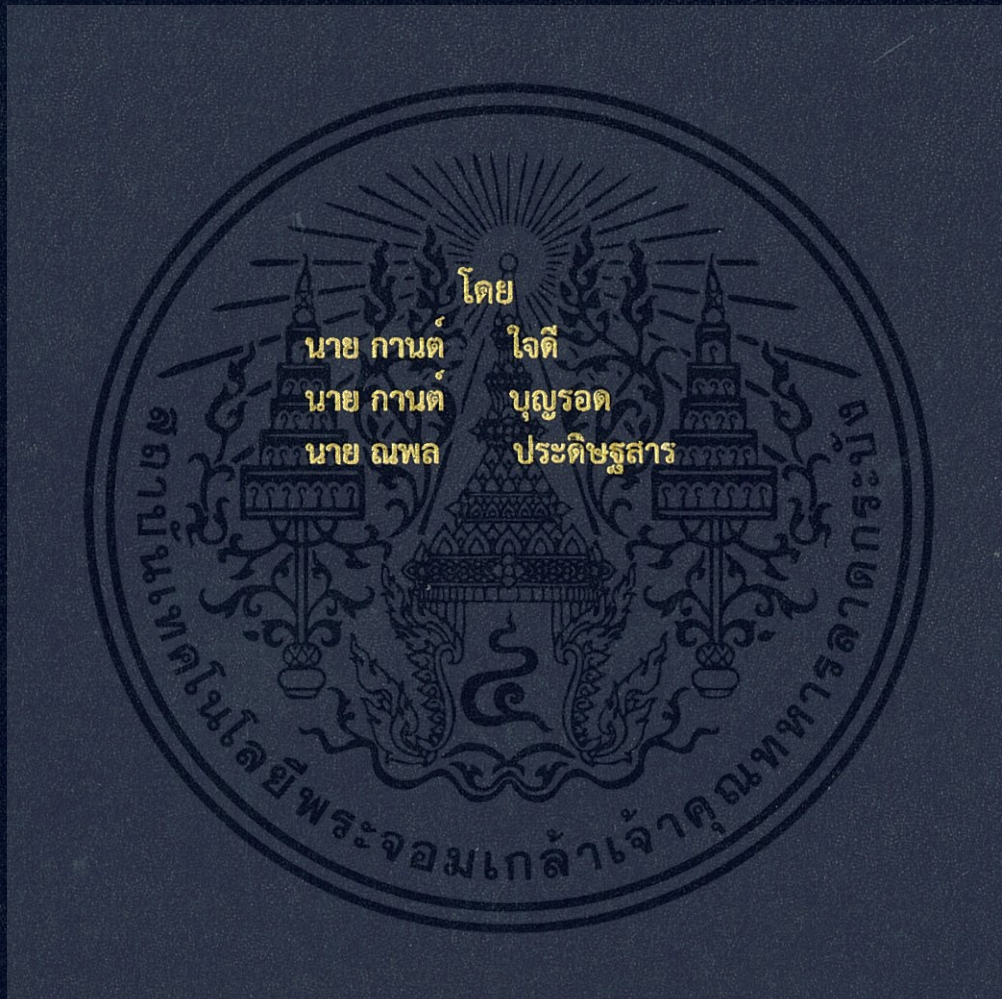


ระบบควบคุมอัตโนมัติเพื่อการเกษตร
AUTOMATIC CONTROL SYSTEM FOR AGRICULTURE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบควบคุมอัตโนมัติเพื่อการเกษตร
AUTOMATIC CONTROL SYSTEM FOR AGRICULTURE

โดย

นาย กานต์	ใจดี	57010076
นาย กานต์	บุญรอด	57010077
นาย ณพล	ประดิษฐ์สาร	57010366

อาจารย์ที่ปรึกษา
รศ.ดร.สุวิมล สิริชีวะภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์


สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

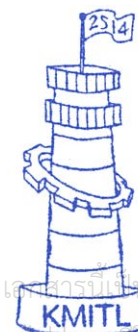
ปีการศึกษา 2560

ผ่านการตรวจรูปเล่มแล้ว

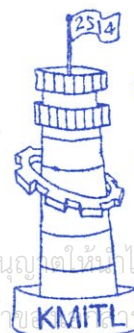

.....
อาจารย์ที่ปรึกษา
18/11/2561

ผ่านการตรวจชิ้นงานแล้ว


.....
กรรมการผู้ตรวจชิ้นงาน
17/11/2561



วิศวกรรมโทรคมนาคม
Telecommunications Engineering



วิศวกรรมโทรคมนาคม
Telecommunications Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่น และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์

ปริญญาโทปีการศึกษา 2560

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมอัตโนมัติเพื่อการเกษตร

AUTOMATIC CONTROL SYSTEM FOR AGRICULTURE

ผู้จัดทำ

- | | |
|------------------------|----------|
| 1. นาย กานต์ ใจดี | 57010076 |
| 2. นาย กานต์ บุญรอด | 57010077 |
| 3. นาย ณพล ประดิษฐ์สาร | 57010366 |

..... อาจารย์ที่ปรึกษา
(รศ.ดร.สุวิมล สิริชีวะภาค)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

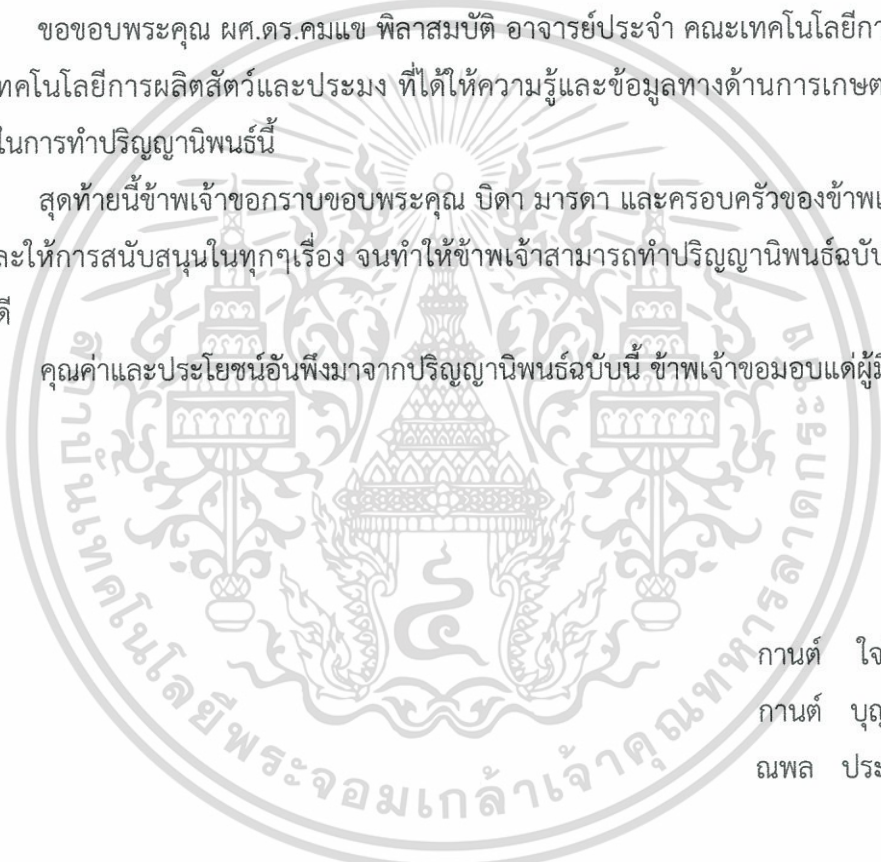
ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยคำแนะนำ และคำปรึกษาจาก อาจารย์ที่ปรึกษา รศ.ดร.สุวิพล สิริชีวะภาค ซึ่งเป็นอาจารย์ควบคุมการทำปริญญาานิพนธ์ ผู้จัดทำ รู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆท่าน ที่ได้ประสิทธิ์ประสาทวิชา

ขอขอบพระคุณ ผศ.ดร.คมแห พิลาสสมบัติ อาจารย์ประจำ คณะเทคโนโลยีการเกษตร ภาควิชาเทคโนโลยีการผลิตสัตว์และประมง ที่ได้ให้ความรู้และข้อมูลทางด้านการเกษตร ที่เป็นประโยชน์ในการทำปริญญาานิพนธ์นี้

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็น กำลังใจและให้การสนับสนุนในทุกๆเรื่อง จนทำให้ข้าพเจ้าสามารถทำปริญญาานิพนธ์ฉบับนี้สำเร็จ ลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญาานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณ ทุกท่าน



กานต์ ใจดี
กานต์ บุญรอด
ณพล ประดิษฐ์สาร
ผู้จัดทำ

ระบบควบคุมอัตโนมัติเพื่อการเกษตร

AUTOMATIC CONTROL SYSTEM FOR AGRICULTURE

โดย	นายกานต์ ใจดี	57010076
	นายกานต์ บุญรอด	57010077
	นายณพล ประดิษฐ์สาร	57010366

อาจารย์ที่ปรึกษา รศ.ดร. สุวิพล ลิทธิชีวะภาค

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการสร้างระบบควบคุมอัตโนมัติเพื่อการเกษตร โดยออกแบบการทำงานของเซนเซอร์วัดความชื้นในดิน และเซนเซอร์วัดอุณหภูมิกับความชื้นสัมพัทธ์ในอากาศ ซึ่งเซนเซอร์ทั้งหมดถูกควบคุมการทำงานผ่านไมโครคอนโทรลเลอร์ จากนั้นจะทำการรับส่งข้อมูล และจัดเก็บข้อมูลที่ได้จากการวัดไปยัง ฐานข้อมูล อีกทั้ง ยังสามารถควบคุมระบบผ่านแอปพลิเคชันบนมือถือสามารถโพระบบปฏิบัติการแอนดรอยด์ โดยจะมีการแจ้งเตือนให้กับผู้ใช้งาน ทำให้ระบบนี้คอยช่วยอำนวยความสะดวกในการทำเกษตรกรรมและประหยัดเวลา

ABSTRACT

This thesis is an Automatic control system for agriculture by designing the soil moisture sensor and temperature sensor with relative humidity. All sensors are controlled via Raspberry Pi3 and then transmit and store the measured data to the database. Moreover, we can control the system through the android operating system application on smartphone. It will alert the user. So, this system facilitates farming and saves time.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III-V
สารบัญรูป	VI-VII
สารบัญตาราง	VIII
บทที่ 1	บทนำ
	1
	1.1 ความเป็นมาและความสำคัญของปัญหา
	1
	1.2 วัตถุประสงค์
	1
	1.3 ขอบเขตของโครงการ
	2
บทที่ 2	ทฤษฎีและหลักการที่เกี่ยวข้อง
	3
	2.1 ภาษา PYTHON
	3
	2.1.1 ภาษา PYTHON คืออะไร
	3
	2.1.2 ประวัติของภาษา PYTHON
	3
	2.1.3 PYTHON INTERPRETER
	4
	2.2 JAVA SCRIPT
	4
	2.2.1 JAVA SCRIPT คืออะไร
	4
	2.2.2 JAVA SCRIPT ทำอะไรได้บ้าง
	4-5
	2.2.3 ข้อดีและข้อเสียของ JAVA SCRIPT
	5
	2.3 RASPBERRY PI 3 MODEL B
	5
	2.3.1 RASPBERRY PI คืออะไร
	5
	2.3.2 คุณสมบัติทางเทคนิคของบอร์ด
	6-7
	2.3.3 ส่วนประกอบของบอร์ด RASPBERRY PI (MODEL B)
	7-10

สารบัญ (ต่อ)

	2.4 ADS1115	10-11
	2.5 ESEN229 – SOIL MOISTURE SENSOR WITH CABLE	11-12
	2.6 DHT 22 DIGITAL TEMPERATURE AND HUMIDITY SENSOR	12-13
	2.7 MINI PUMP MOTOR 5V	13
	2.8 NODE JS DATABASE	14
	2.9 BLYNK APPLICATION	14-15
	2.10 ผักบั้งจีน	15-16
บทที่ 3	การออกแบบและการจัดทำปริญญานิพนธ์	17
	3.1 การออกแบบ	17
	3.1.1 การออกแบบการทำงานรวมของระบบ	17
	3.1.2 แผนการปฏิบัติงาน	18
	3.1.3 การออกแบบวงจร SOIL MOISTURE SENSOR	19
	3.1.4 การออกแบบวงจร TEMPERATURE & HUMIDITY SENSOR	20
	3.1.5 การออกแบบวงจรปั้มน้ำและรีเลย์	21
	3.1.6 การออกแบบ PLATFORM บน BLYNK APPLICATION	22-23
	3.2 เครื่องมือที่ใช้ในการทดลอง	24
	3.3 การจัดเก็บผลการทดลอง	24-25
บทที่ 4	ผลการทดลอง	26
	4.1 ผลการทดสอบ SOIL MOISTURE SENSOR	26-28
	4.2 ผลการทดสอบ TEMPERATURE & HUMIDITY SENSOR	28-29
	4.3 ผลการทดสอบ การทำงานของระบบ	30-31
	4.4 ผลการทดสอบ การทำงานบนแอปพลิเคชัน	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

บทที่ 5	สรุปผลและข้อเสนอแนะ	33
	5.1 สรุปผล	33
	5.2 ข้อเสนอแนะ	33
บรรณานุกรม		34
ภาคผนวก		35-43



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า	
2.1	บอร์ด RASPBERRY PI (MODEL B)	7
2.2	พอร์ต GPIO ระหว่าง MODEL A และ B	8
2.3	พอร์ตเชื่อมต่อสัญญาณภาพออกแบบ RCA	8
2.4	สถานะของบอร์ด	9
2.5	RASPBERRY PI CAMERA MODULE	9
2.6	สาย HDMI	10
2.7	HDMI TO VGA	10
2.8	ADS 1115	10
2.9	เซนเซอร์วัดความชื้นในดิน	11
2.10	เซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์	12
2.11	ปั้มน้ำ 5V	13
2.12	ตัวอย่าง APP BLYNK	14
2.13	ข้อมูลหน้าจอต่างๆที่มีให้ใน APP	15
2.14	ผักบุงจิ้น	16
3.1	บล็อกไดอะแกรมของระบบการทำงาน	17
3.2	วงจร SOIL MOISTURE SENSOR	19
3.3	วงจร TEMPERATURE & HUMIDITY SENSOR	20
3.4	วงจรปั้มน้ำและรีเลย์ 4 โมดูล	21
3.5	หน้าต่างแอปพลิเคชันเมื่อเริ่มต้น	22
3.6	หน้าต่างการกำหนด DETAIL ที่จะใช้งาน	22
3.7	หน้าต่าง PLATFORM ที่จะออกแบบ WIDGETS	23
3.8	หน้าต่างแสดงผลเมื่อใช้แอปพลิเคชัน	23
3.9	แปลงปลุกผักบุงจิ้น	25
3.10	หน้าต่างแสดงผลพร้อมมี CLOUD DATABASE	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.1	CODE ที่ใช้ใน SOIL MOISTURE SENSOR	26
4.2	ผลการทดลองในดินความชื้นน้อย	27
4.3	ผลการทดลองในดินความชื้นปกติ	27
4.4	ผลการทดลองในดินความชื้นสูง	28
4.5	CODE ที่ใช้ใน TEMPERATURE & HUMIDITY SENSOR	28
4.6	ผลการทดลองวัดอุณหภูมิของอากาศและความชื้นสัมพัทธ์	29
4.7	เมื่อนำเซนเซอร์ทั้ง 2 ตัว ไปวัดลงบนแปลงปลูกผักบุ้งจีน	30
4.8	ค่าความชื้นในดินที่เก็บบน NODE JS	31
4.9	ค่าอุณหภูมิกับความชื้นสัมพัทธ์ของอากาศที่เก็บบน NODE JS	31
4.10	การแสดงผลข้อมูลบนแอปพลิเคชัน	32

สารบัญตาราง

ตารางที่		หน้า
2.1	คุณสมบัติของบอร์ด RASPBERRY PI	6-7
3.1	แผนการปฏิบัติงาน	18



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันอุตสาหกรรมในหลายๆ ด้าน ได้มีการนำอุปกรณ์และระบบอิเล็กทรอนิกส์ มาช่วยทำให้การผลิตเกิดประสิทธิภาพและประสิทธิผลมากขึ้น รวมทั้งได้ใช้เซนเซอร์มาช่วยทำให้เกิดระบบอัจฉริยะที่สามารถประเมินผลและควบคุมตนเองได้ ด้วยการส่งและรับข้อมูลต่างๆ จากทุกสิ่งๆ ที่เชื่อมต่อเข้าหากัน ซึ่งในปัจจุบันที่ได้ยินกันบ่อยๆ ก็คือ IOT (Internet Of Things)

ซึ่งในแง่ของการเกษตรแล้ว ปัจจุบันเทคโนโลยีเซนเซอร์เริ่มถูกลงมาก ทำให้มีการปรับใช้กับการเกษตรมากยิ่งขึ้น เพื่อเพิ่มผลผลิต หรือช่วยในการบริหารจัดการ ทั้งนี้กลุ่มของข้าพเจ้า จึงเล็งเห็นว่าเทคโนโลยีเหล่านี้ที่นำมาใช้ในทางเกษตรยังเกิดขึ้นค่อนข้างน้อย ไม่ว่าจะเป็นเรื่องของการตรวจวัดภูมิอากาศ เช่น ปริมาณแสงแดด อุณหภูมิ ความชื้น ซึ่งจำเป็นระดับหนึ่งในการเกษตร แต่ก็ยังสามารถนำมาจัดการเพื่อการเพิ่มผลผลิตการเกษตรได้ไม่มากนัก อีกทั้งเทคโนโลยีเซนเซอร์ที่เป็น

แนวทางในการพัฒนาด้านการเกษตรโดยเฉพาะการผลิตพืชคงหนีไม่พ้นเรื่องของการควบคุมการรดน้ำ ดังนั้นกลุ่มของข้าพเจ้าจึงออกแบบ ระบบควบคุมอัตโนมัติเพื่อการเกษตร ขึ้นมา โดยมี การวัดแบบ real-time sensor ซึ่งสามารถวัดข้อมูลและส่งเข้าระบบ database เพื่อให้เข้าถึงข้อมูลได้ตลอดเวลา และยังมีการทำระบบแจ้งเตือนผ่านทางแอปพลิเคชันอีกด้วย อันจะเป็นประโยชน์ให้แก่ชาวเกษตรกรและผู้ที่เกี่ยวข้องต่อไป

1.2 วัตถุประสงค์

- 1) เพื่อศึกษาพฤติกรรมของเซนเซอร์และหลักการทำงาน
- 2) เพื่ออำนวยความสะดวกในการทำเกษตรกรรม

1.3 ขอบเขตของปริญญาานิพนธ์

- 1) ออกแบบการทำงานของ Sensor ให้ใช้ได้กับโครงงาน
- 2) ใช้อุปกรณ์ Raspberry Pi ควบคุมการทำงานของ Sensor
- 3) รับส่งข้อมูล และจัดเก็บข้อมูลที่ได้จากการวัดค่า เข้าในส่วนของ Database
- 4) ทำระบบแจ้งเตือนและจัดการผ่าน Andriod OS Application



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ภาษา Python

2.1.1 ภาษา Python คืออะไร

Python เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นสร้างโดย Guido van Rossum Python นั้นเป็นภาษาแบบ interprete ที่ถูกออกแบบโดยมีปรัชญาที่จะทำให้โค้ดอ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ตั้งใจให้การเขียนโค้ดเข้าใจง่าย

ตัวแปรในภาษา Python นั้นมีให้ใช้ในหลายระบบปฏิบัติการ ทำให้โค้ดของภาษา Python สามารถรันในระบบต่างๆ ได้อย่างกว้างขวาง

2.1.2 ประวัติของภาษา Python

ภาษา Python นั้นกำเนิดขึ้นในปลายปี 1980 และการพัฒนาของมันนั้นเริ่มต้นใน December 1989 โดย Guido van Rossum ที่ Centrum Wiskunde & Informatica (CWI) ในประเทศเนเธอร์แลนด์ เนื่องในผู้ประสบความสำเร็จในการสร้างภาษา ABC ที่มีความสามารถสำหรับการ exception handling และการติดต่อประสานกับระบบปฏิบัติการ Amoeba ซึ่ง Van Rossum นั้นเป็นผู้เขียนหลักการของภาษา Python และเขาทำหน้าที่เป็นกลางในการตัดสินใจสำหรับทิศทางการพัฒนาของภาษา Python

2.1.3 Python Interpreter

Python interpreter นั้นเป็นตัวแปรภาษาของภาษา Python เพื่อให้สามารถรันโค้ด Python ได้ ซึ่งได้มากับไลบรารีมาตรฐานที่สามารถใช้งานได้ฟรี นอกจากนี้ interpreter ยังสนับสนุนการเขียนโปรแกรมกับ Interactive shell ซึ่งเป็นการเขียนโค้ดของภาษา Python ลงไป และเห็นผลลัพธ์การทำงานของคำสั่งได้ในทันที

2.2 Java Script

2.2.1 Java Script คืออะไร

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปที่ละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียนเตด (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต

2.2.2 Java Script ทำอะไรได้บ้าง

- 1) JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น
- 2) JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งาน
- 3) JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้

2.2.3 ข้อดีและข้อเสียของ Java Script

การทำงานของ JavaScript เกิดขึ้นบนบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เซิร์ฟเวอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า server-side script) อย่างไรก็ตาม จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น

2.3 Raspberry Pi 3 Model B

2.3.1 Raspberry Pi คืออะไร

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงการทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่อง

อินเทอร์เน็ต ส่งอีเมลล์ หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย

2.3.2 คุณสมบัติทางเทคนิคของบอร์ด

บอร์ด Raspberry Pi ปัจจุบันมีด้วยกัน 2 โมเดล คือ โมเดล A และ โมเดล B ซึ่งทั้ง 2 โมเดลมีคุณสมบัติทางเทคนิคที่ใกล้เคียงกัน แตกต่างกันเพียงบางส่วน รายละเอียดดังตารางที่ 2.1

ตารางที่ 2.1 คุณสมบัติของบอร์ด Raspberry Pi

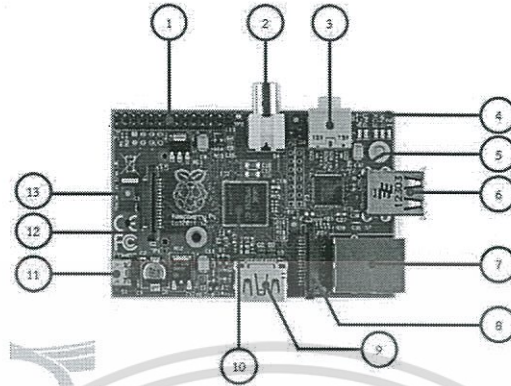
	โมเดล A	โมเดล B (Revision 2)
System on a chip (SoC)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM and Single USB Port)	
CPU	700MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)	
GPU	Broadcom VideoCore IV @ 250 MHz OpenGL ES 2.0 (24 GFLOPS) MPEG-2 and VC-1, 1080p 30 h.264/MPEG-4 AVC high-profile decoder and encoder	
Memory (SDRAM)	256 MB (Shared with GPU)	512 MB (Shared with GPU)
USB 2.0 Ports	1 (direct form BCM2835)	2 (via the build in integrated 3-port USB hub)
Video Input	A CSI input connector allows for the connection of RPF designed camera module (ออกแบบมาให้เชื่อมต่อกับ Raspberry Pi Camera Module โดยเฉพาะ)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Video Outputs	Composite RCA (PAL and NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards. (มีทั้งสองแบบ คือ แบบ RCA และแบบ HDMI)	
Audio Outputs	3.5 mm jack, HDMI, and as of revision 2 boards, I ² S audio (also potentially for audio input)	
Onboard storage	SD/ MMC/ SDIO card slot (3.3V card power support only)	
Onboard network	None	10/100 Ethernet (8P8C) USB adapter on the third port of the USB hub
Low-level peripherals Low-level peripherals	8 x GPIO, UART, I ² C Bus, SPI Bus with two chip selects, I ² S audio +3.3V, +5V, Ground	
Power ratings	300 mA (1.5 W)	700 mA (3.5 W)
Power source	5 Volt via Micro USB or GPIO header	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 ส่วนประกอบของบอร์ด Raspberry Pi (Model B) ดังรูปที่ 2.1



รูปที่ 2.1 บอร์ด Raspberry Pi (Model B)

[<http://www.hackthings.com/raspberry-pi-model-a-and-b/>]

1) พอร์ต GPIO ซึ่งในโมเดล A และ B (Revision 1) ทุก Pin จะเหมือนกัน แต่โมเดล B (Revision 2) จะแตกต่างกัน ดังรูปที่ 2.2

Raspberry Pi Model A & B (Revision 1)

Raspberry Pi Model B (Revision 2)

3.3V	1		
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

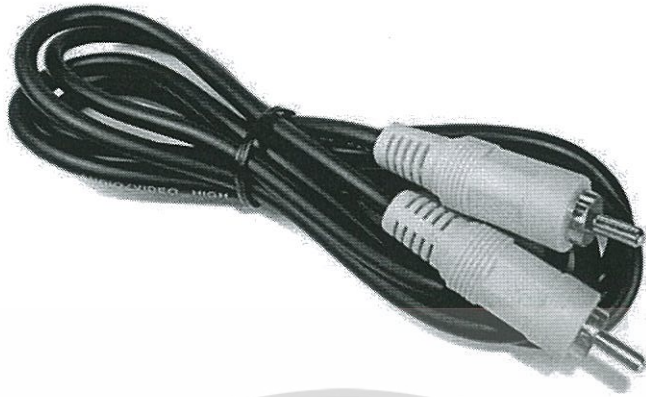
3.3V	1		
I2C1 SDA	3	4	DNC
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
GROUND	25	26	SP10 CE1 N

รูปที่ 2.2 พอร์ต GPIO ระหว่าง model A และ B

[<http://www.hobbytronics.co.uk/raspberry-pi-gpio-pinout>]

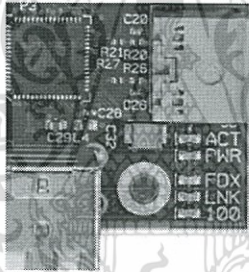
2) พอร์ตเชื่อมต่อสัญญาณภาพออกแบบ RCA ตัวอย่างของสายที่เชื่อมต่อแสดงดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 พอร์ตเชื่อมต่อสัญญาณภาพออกแบบ RCA
[<https://www.crazypi.com/raspberry-pi-starter-kit>]

- 3) จุดเชื่อมต่อสัญญาณเสียงขนาด 3.5 มิลลิเมตร
- 4) LED แสดงสถานะของบอร์ด อยู่ในบริเวณกรอบสีแดง ดังรูปที่ 2.4

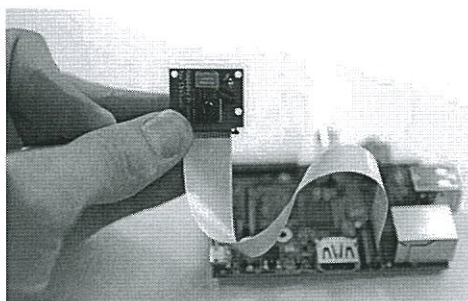


รูปที่ 2.4 สถานะของบอร์ด

[<http://www.rpiblog.com/2012/12/raspberry-pi-status-indicator-led-info.html>]

- 5) ชิพควบคุม LAN (LAN Controller)
- 6) พอร์ต USB 2.0 จำนวน 2 พอร์ต
- 7) พอร์ต RJ-45 Ethernet LAN 10/100Mbps
- 8) พอร์ต CSI (Camera Serial Interface) สำหรับเชื่อมต่อโมดูลกล้องดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 Raspberry PI Camera Module

[<http://thaieasyelec.com/article-wiki/embedded-electronics-application/บทความการพัฒนาโปรแกรม-raspberry-pi-ด้วย-qt.html>]

9) พอร์ต HDMI สำหรับเชื่อมต่อสัญญาณภาพและเสียง ตัวอย่างสาย HDMI และตัวแปลง HDMI to VGA แสดงดังรูปที่ 2.6 และ 2.7



รูปที่ 2.6 สาย HDMI

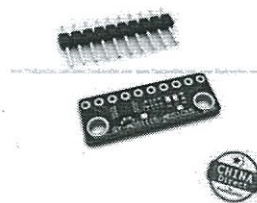
รูปที่ 2.7 HDMI to VGA

[<http://thaieasyelec.com/article-wiki/embedded-electronics-application/บทความการพัฒนาโปรแกรม-raspberry-pi-ด้วย-qt.html>]

2.4 ADS 1115

โมดูล ADS1115 สำหรับรับ และแปลงสัญญาณอนาล็อกเป็นดิจิทัล ความละเอียด 16-bit ส่งงานผ่านบัส I2C เปลี่ยนแอดเดรสเชื่อมต่อกันได้สูงสุด 4 โมดูลในบัสเดียว เหมาะสำหรับไมโครคอนโทรลเลอร์ที่ไม่มี ADC (Raspberry Pi) หรือต้องการ ADC ความละเอียดสูงขึ้น ดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 ตัว ADS 1115

[<http://www.thaieasyelec.com/en/interface-modules/adc/ads1115-16-bit-adc-module-with-programmable-gain-amplifier-detail.html>]

Specifications:

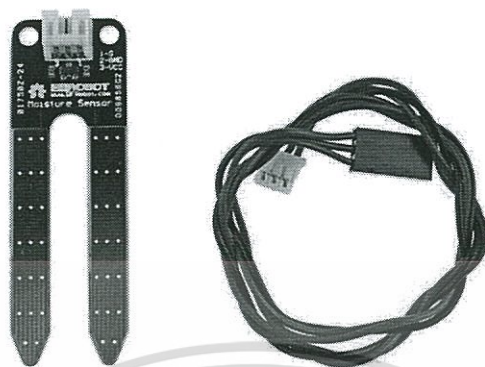
- Wide supply range: 2.0V to 5.5V
- Continuous Mode: Only 150 μ A
- Single-Shot Mode: Auto Shut-Down
- Single-Shot Mode: Auto Shut-Down
- Internal low-drift voltage reference

Product included:

- EFDV535 ADS1115 16-Bit ADC Module with Programmable Gain Amplifier x 1pcs

2.5 ESEN229 - Soil Moisture Sensor with Cable

เซนเซอร์ความชื้นดินนี้ใช้ Immersion Gold ซึ่งช่วยป้องกันนิกเกิลจากการเกิดออกซิเดชัน (และมีราคาถูกลง) เช่น HASL (บัดกรี) ทนต่อการเกิดปฏิกิริยาออกซิเดชันที่ดีและการทำงานสำหรับการสัมผัสที่ไม่ได้รับการบำบัด พื้นผิวเช่นสวิตช์เมมเบรน และจุดติดต่อ ดังรูปที่ 2.10



รูปที่ 2.9 เซนเซอร์วัดความชื้นในดิน

[<http://www.thaieasyelec.com/en/sensors/humidity-temperature/soil-moisture-sensor-with-cable-detail.html>]

Specification

- 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)

2.6 DHT22 Digital Temperature and Humidity Sensor

เป็นโมดูลวัดอุณหภูมิ (Temperature) และความชื้นสัมพัทธ์ (Humidity) โดยใช้ชิพ DHT22 ให้ Output ออกมาเป็นแบบ Digital Output ใช้ไฟ DC ขนาด 3.3 – 6 โวลต์ สามารถต่อกับบอร์ด Arduino ใช้งานได้ทันที ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 เซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์

[<http://www.thaieasyelec.com/en/products-en/china-direct-en/dht22-digital-temperature-and-humidity-sensor-detail.html>]

Description:

- Size: 28mm X 12mm X 10mm
- Main chip: DHT22 temperature and humidity sensor
- Working voltage: DC 3.3-6 V
- Sensing element Polymer capacitor
- Operating range humidity 0-100%RH; temperature -40~80Celsius
- Accuracy humidity +2%RH (Max +-5%RH); temperature <+-0.5Celsius

2.7 Mini Pump Motor 5V

Mini Pump Motor คือปั๊มน้ำขนาดเล็กที่ใช้มอเตอร์ขนาด 5V ในการสูบน้ำขึ้นทางท่อออกไปยังภาชนะอื่น โดยการนำตัวปั๊มจุ่มลงไปใต้น้ำเพื่อสูบน้ำขึ้นมาใช้งานต่อไป ดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ปั๊มน้ำ Motor 5V

[<http://www.thaieasyelec.com/en/home/mini-pump-motor-5v-detail.html?tmpl=component&flexiblelayout=print>]

Specification :

- Voltage : 2.5-6V
- Maximum lift : 40-110cm / 15.75"-43.4"
- Flow rate : 80-120L/H
- Outside diameter : 7.5mm / 0.3"

2.8 Node Js Database

- NodeJS คือ JavaScript มีการ Compiled เป็น Byte Code ด้วย V8 Engine ของ Google
- Debug ได้ ต่างจาก JavaScript ในยุคแรกๆ ทำให้แก้ปัญหาได้ง่ายขึ้น
- รันได้บนทุกระบบปฏิบัติการยอดนิยม
- มาพร้อมกับเทคโนโลยีที่เรียกว่า Non – Blocking I/O

เหตุผลที่ได้รับความนิยมหลักๆคือ

- เขียนโค้ดเข้าใจได้ง่าย
- มี Library ฟรีมากมาย
- ใช้ทรัพยากรน้อย
- เรียนรู้ได้เร็วสำหรับโปรแกรมเมอร์ทุกระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 Blynk Application

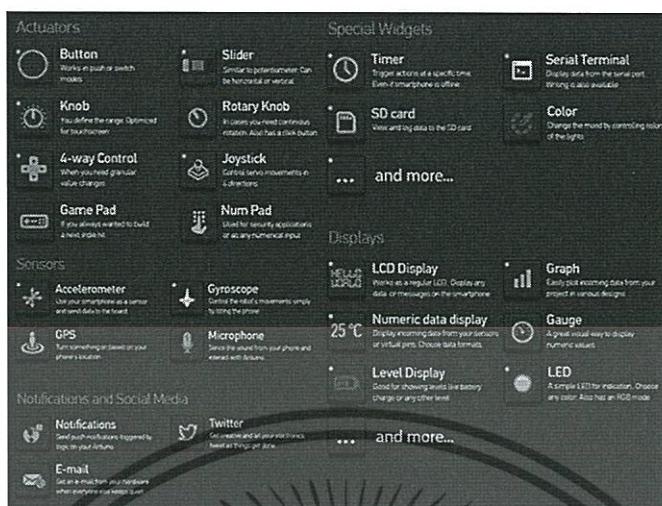
Blynk คือ Application สำเร็จรูปสำหรับงาน IOT มีความน่าสนใจคือการเขียนโปรแกรมที่ง่าย สามารถใช้งานได้จริงแบบ Real time สามารถเชื่อมต่อ Device ต่างๆเข้ากับ Internet ได้ ไม่ว่าจะเป็น Arduino, Esp8266, Esp32, Nodemcu, Raspberry pi นำมาแสดงบน Application ดังรูปที่ 2.13



รูปที่ 2.12 ตัวอย่าง App Blynk [<https://www.9arduino.com/article/59/app-1-blynk>]

คอมพิวเตอร์ Smartphone ก็จะเชื่อมต่อกับ Server ของ Blynk โดยตรง กลายเป็นว่า มี Server เป็นสะพานให้เชื่อมต่อหากันจึงหมดปัญหาและข้อจำกัดทุกอย่างทำให้อุปกรณ์ของเรามีความฉลาดมากขึ้น ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ข้อมูลหน้าจอต่างๆที่มีให้ใน App [<https://www.9arduino.com/article/59/app-1-blynk>]

2.10 ผักบุงจีน

ผักบุงจีน มีรากแขนงแตกออกทางด้านข้างของรากแก้วและยังสามารถแตกรากฝอยออกมาจากข้อของลำต้นได้ด้วย โดยมักจะเกิดตามข้อที่อยู่บริเวณโคนเถา เป็นไม้ล้มลุก ในระยะแรกของการเจริญเติบโตจะมีลำต้นตั้งตรง ในระยะต่อไปลำต้นจะเลื้อยทอดยอดไปตามพื้นดินหรือน้ำ ลำต้นมีสีเขียวและปล้องข้างในกลวง รากจะเกิดที่ข้อทุกข้อที่สัมผัสกับพื้นดินหรือน้ำ ที่ข้อมักมีตาแตกออกมา ทั้งตาใบและตาดอก

ใบผักบุงจีน เป็นใบเดี่ยว รูปร่างคล้ายหอก โคนใบกว้างแล้วค่อยๆ เรียวเล็กไปตอนปลาย ปลายใบแหลมที่โคนใบเป็นรูปหัวใจ ขอบใบเรียบหรือเป็นคลื่น สำหรับการจัดเรียงของใบเป็นแบบเรียงสลับ ข้อหนึ่งจะมีใบเพียงใบเดียว

ผลผักบุงจีน เป็นผลเดี่ยว รูปร่างค่อนข้างกลม มีขนาดใหญ่ที่สุดเมื่ออายุ ประมาณ 30 วันหลังดอกบาน ลักษณะผิวภายนอกจะเหี่ยวย่น ขรุขระ ไม่แตก เมื่อแห้งสีของผลเมื่อแก่จะมี สีน้ำตาลหรือน้ำตาลเข้ม ใน 1 ผลมีเมล็ด 4-5 เมล็ด ดังรูปที่ 2.15



รูปที่ 2.14 ผักบุงจีน [<https://www.samunpri.com/ผักบุงจีน>]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานรวมของระบบดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของระบบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

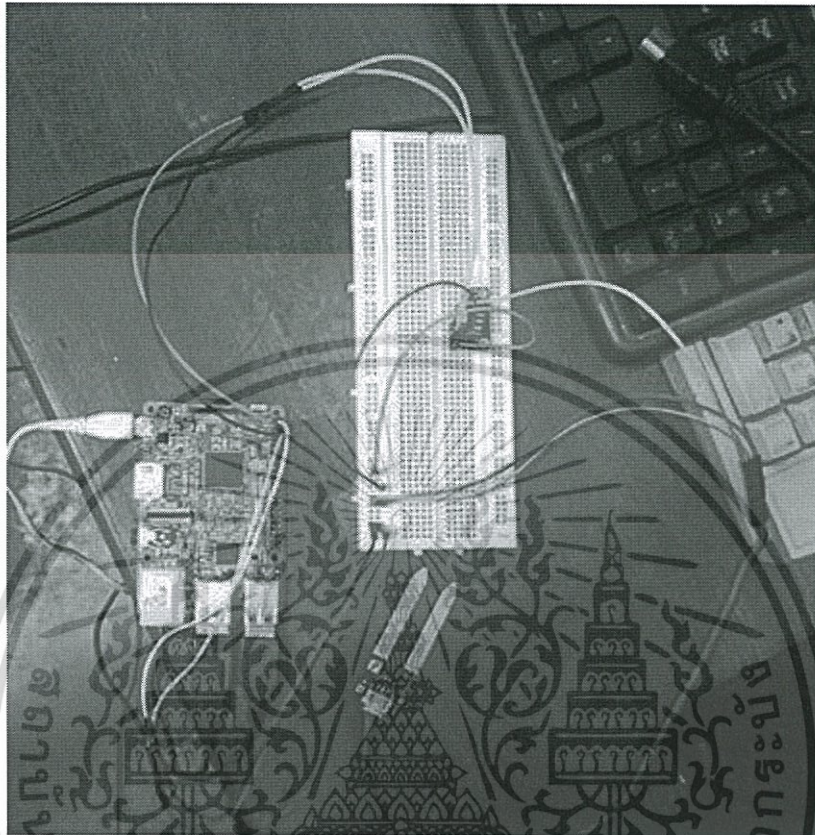
3.1.2 แผนการปฏิบัติงานดังตารางที่ 3.1

ตารางที่ 3.1 แผนการปฏิบัติงาน

ช่วงการดำเนินงาน	แผนงานที่จะดำเนินการ
สิงหาคม 60 – กันยายน 60	-ศึกษาหลักการดำเนินงานของไมโครคอนโทรลเลอร์ -ศึกษาการทำงานของเซนเซอร์
ตุลาคม 60 – พฤศจิกายน 60	-จัดซื้ออุปกรณ์ที่จำเป็นต้องใช้ -ทดลองการทำงานของเซนเซอร์
พฤศจิกายน 60 – ธันวาคม 60	-สร้างชิ้นงานและทดสอบการทำงาน -ปรับปรุงแก้ไขการทำงานของโครงงาน
มกราคม 61- กุมภาพันธ์ 61	- ทดสอบระบบรวมทั้งหมดของโครงงาน - สังเกตการทำงาน
กุมภาพันธ์ 61 - มีนาคม 61	-ทดลองโครงงานกับพื้นที่จริง
มีนาคม 61 – เมษายน 61	- สังเกตผลการเจริญเติบโตของพืช - ทำการจัดเก็บข้อมูลโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การออกแบบวงจร soil moisture sensor ดังรูปที่ 3.2



รูปที่ 3.2 วงจร Soil Moisture sensor

ขั้นตอนการต่อวงจร

ADS1115 VCC ต่อกับ Raspberry pi 3.3v

ADS1115 SDA ต่อกับ Raspberry pi 3

ADS1115 SDL ต่อกับ Raspberry pi 5

ADS1115 GND ต่อกับ Raspberry pi GND

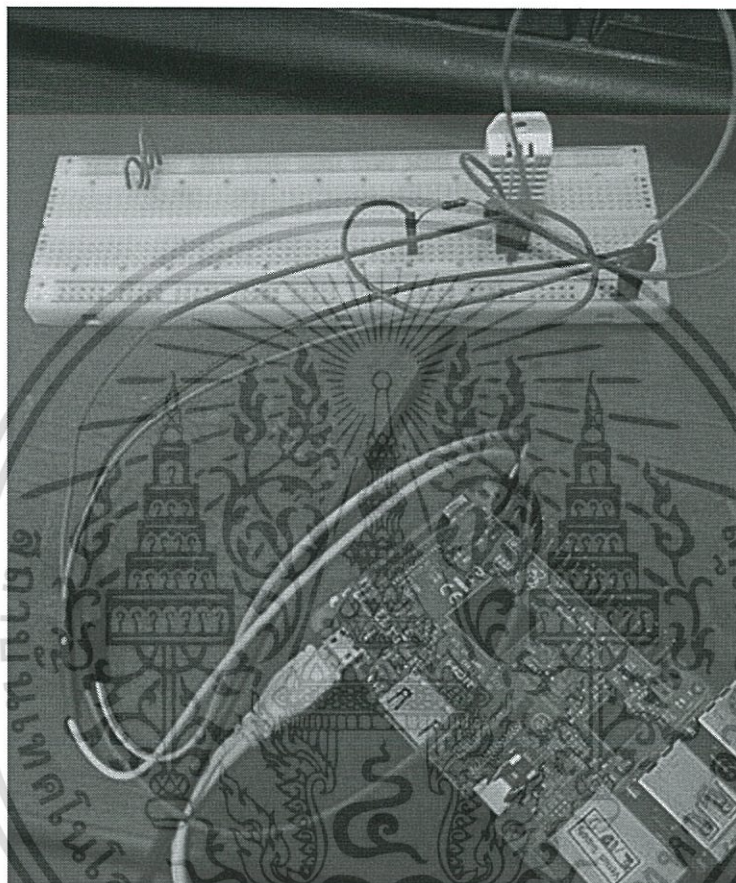
Soil Moisture Output ต่อกับ ADS1115 pin 0

Soil Moisture VCC ต่อกับ Raspberry pi 3.3v

Soil Moisture GND ต่อกับ Raspberry pi GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การออกแบบวงจร Temperature & Humidity sensor ดังรูปที่ 3.3



รูปที่ 3.3 วงจร Temperature & Humidity sensor

ขั้นตอนการต่อวงจร

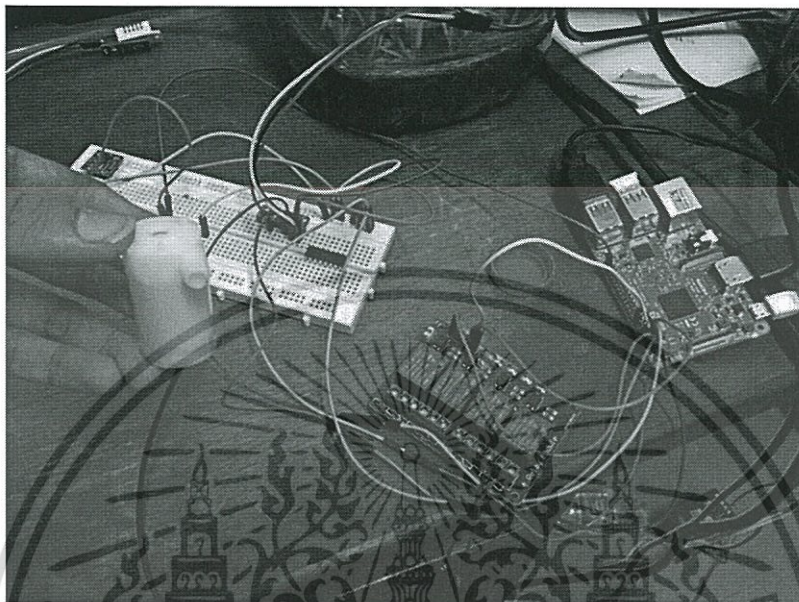
DHT22 VCC ต่อกับ Raspberry pi 3.3v

DHT22 GND ต่อกับ Raspberry pi GND

DHT22 Output ต่อกับ Raspberry pi pin4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 การออกแบบวงจร ป้อนน้ำและรีเลย์ ดังรูปที่ 3.4



รูปที่ 3.4 วงจรป้อนน้ำและรีเลย์ 4 โมดูล

ขั้นตอนการต่อวงจร

PUMP : GND ของปั๊มกับ GND ของสาย USB ต่อเข้ากับ Relay

สายขั้วบวก ต่อเข้ากับ ขั้วบวกของสาย USB

RELAY : Input (IN1) ต่อเข้า PIN7 (GPIO4)

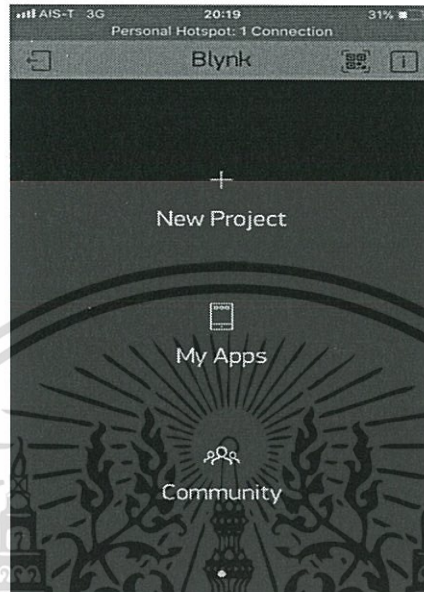
VCC ต่อเข้า PIN1 (DC 5V)

GND ต่อเข้า PIN9 (GND)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

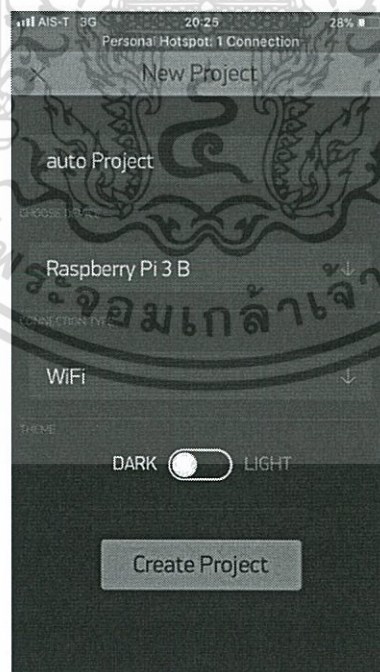
3.1.6 การออกแบบ Platform บน Blynk Application

1) เริ่มเข้าสู่การออกแบบ โดยเลือกที่ New Project ดังรูปที่ 3.5



รูปที่ 3.5 หน้าต่างแอปพลิเคชันเมื่อเริ่มต้น

2) กำหนด Detail ที่จะใช้งานในแอปพลิเคชัน ดังรูปที่ 3.6



รูปที่ 3.6 หน้าต่างการกำหนด Detail ที่จะใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) จะได้ Platform ที่จะเริ่มใช้ออกแบบ widgets ดังรูปที่ 3.7



รูปที่ 3.7 หน้าต่าง Platform ที่จะออกแบบ widgets

4) ทำการออกแบบ widgets ต่างๆ ให้เสร็จสมบูรณ์ดังรูปที่ 3.8



รูปที่ 3.8 หน้าต่างแสดงผลเมื่อใช้แอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Sensor

3.2.1.1 Soil moisture sensor

3.2.1.2 Temperature & Humidity sensor

3.2.2 Microcontroller

3.2.2.1 Raspberry Pi3

3.2.3 Hardware

3.2.3.1 Monitor

3.2.3.2 Mouse

3.2.3.3 Keyboard

3.2.3.4 5V DC Pump Motor

3.2.3.5 5V Relay 4CH Module

3.2.4 Analog to Digital converter

3.2.4.1 ADS1115

3.3 การจัดเก็บผลการทดลอง

ในการเก็บผลการทดลองนั้น จะทำการนำแปลงสำหรับปลุกผักบุงจีน แล้วทดลองปลุกผักบุงจีน โดยการนำชิ้นงานเข้ามาทดสอบการทำงานในสิ่งแวดล้อมตามธรรมชาติ ดังรูปที่ 3.9 แล้วทำการเก็บผลการทดลองแบบ real-time บน Cloud Database ภายในแอปพลิเคชัน ดังรูปที่ 3.10



รูปที่ 3.9 แปลงปลูกผักนึ่งจีน



รูปที่ 3.10 หน้าต่างแสดงผลพร้อมมี Cloud Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบ Soil moisture sensor

ทดสอบการทำงานของเซนเซอร์โดยการจำลองสถานะต่างๆตามธรรมชาติ เช่น สถานะดินที่มีความชื้นสูง ดินที่มีความชื้นปกติ และดินที่มีความชื้นต่ำหรือแห้ง แล้วนำค่าที่ได้มาเปรียบเทียบเพื่อประเมินความแม่นยำของเซนเซอร์ ดังรูปที่ 4.1 และมีผลการทดลองแต่ละแบบออกมา ดังรูปที่ 4.2 , 4.3 และ 4.4



```
File Edit Tabs Help
GNU nano 2.7.4 File: project moist.py

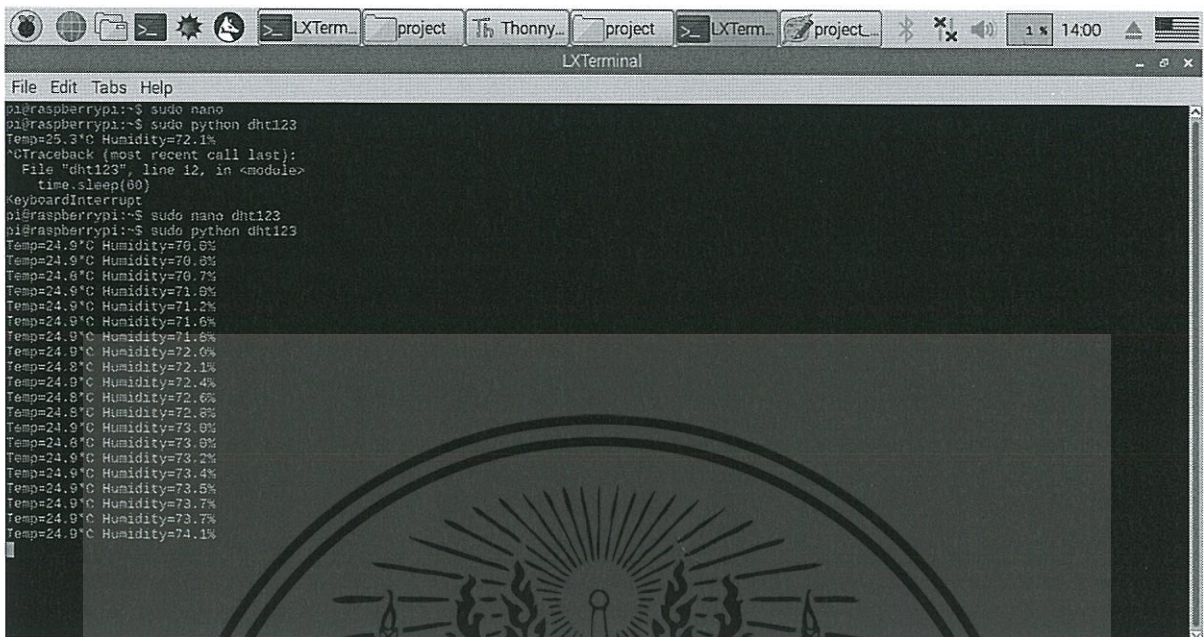
import time
import Adafruit_ADS1x15

adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1

adc.start_adc(0,GAIN)
print('=====')
print('Soil Moisture Status')
print('=====')
start = time.time()
while True:
    value = int(round(adc.read_adc(0)/10.24))
    if value < 200:
        print('Soil Stat -----> dry '.format(value))
    elif value < 950:
        print('Soil Stat -----> Humid soil '.format(value))
    else:
        print('Soil Stat -----> In water '.format(value))
    time.sleep(5)
adc.adc_stop()
```

รูปที่ 4.1 code ที่ใช้ใน soil moisture sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

pi@raspberrypi:~$ sudo nano
pi@raspberrypi:~$ sudo python dht123
Temp=25.3°C Humidity=72.1%
^CTraceback (most recent call last):
  File "dht123", line 12, in <module>
    time.sleep(60)
KeyboardInterrupt
pi@raspberrypi:~$ sudo nano dht123
pi@raspberrypi:~$ sudo python dht123
Temp=24.9°C Humidity=70.0%
Temp=24.9°C Humidity=70.0%
Temp=24.6°C Humidity=70.7%
Temp=24.9°C Humidity=71.0%
Temp=24.9°C Humidity=71.2%
Temp=24.9°C Humidity=71.6%
Temp=24.9°C Humidity=71.8%
Temp=24.9°C Humidity=72.0%
Temp=24.8°C Humidity=72.1%
Temp=24.9°C Humidity=72.4%
Temp=24.8°C Humidity=72.6%
Temp=24.8°C Humidity=72.8%
Temp=24.9°C Humidity=73.0%
Temp=24.8°C Humidity=73.0%
Temp=24.9°C Humidity=73.2%
Temp=24.9°C Humidity=73.4%
Temp=24.9°C Humidity=73.5%
Temp=24.9°C Humidity=73.7%
Temp=24.9°C Humidity=73.7%
Temp=24.9°C Humidity=74.1%

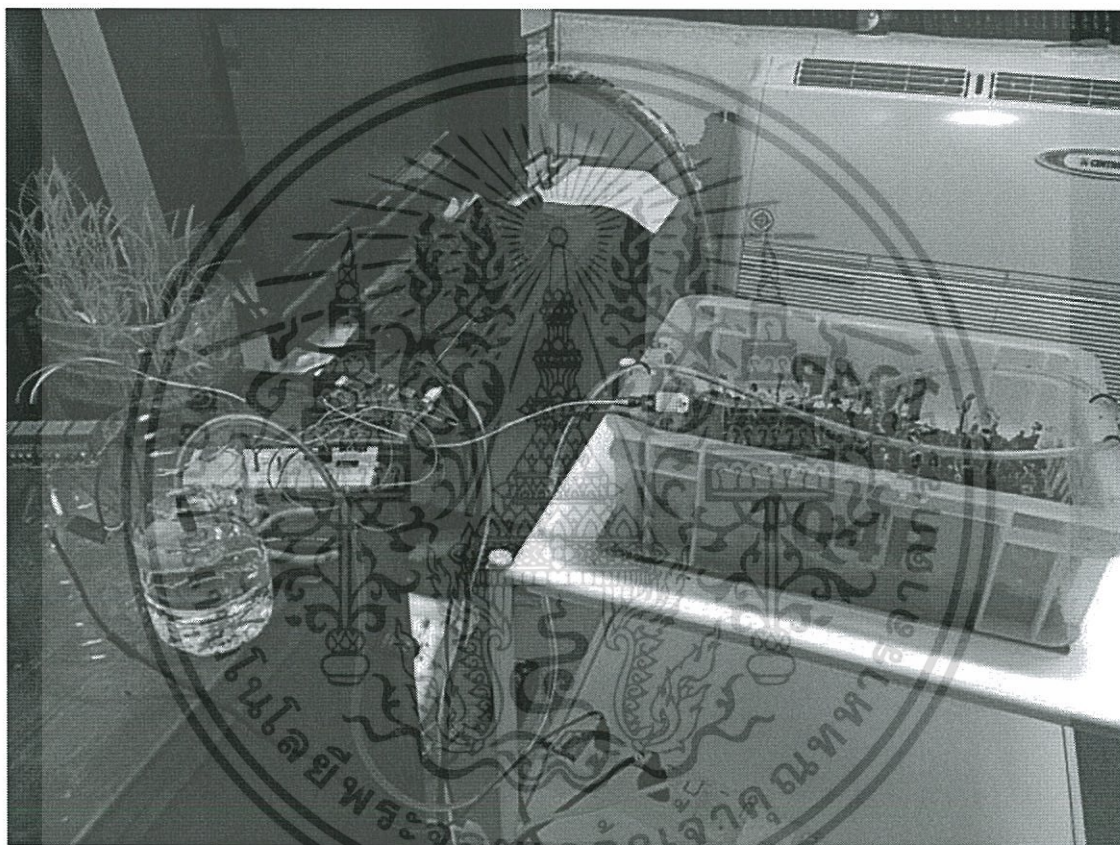
```

รูปที่ 4.6 ผลการทดลองวัดอุณหภูมิของอากาศและความชื้นสัมพัทธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบ การทำงานของระบบ

ทดสอบการทำงานของระบบ โดยการนำเซนเซอร์วัดความชื้นในดินและเซนเซอร์วัดอุณหภูมิกับความชื้นสัมพัทธ์ของอากาศ ไปวัดลงบนแปลงปลูกผักบุงจิ้นดังรูปที่ 4.7 ซึ่งเมื่อสั่งการโปรแกรมที่เขียนไว้ จะมีผลการทดลองดังรูปที่ 4.8 และ 4.9 ถ้าเกิดตรวจพบว่ามีค่าใด น้อยกว่าค่าที่กำหนดไว้ ก็จะสั่งให้รดน้ำโดยอัตโนมัติ

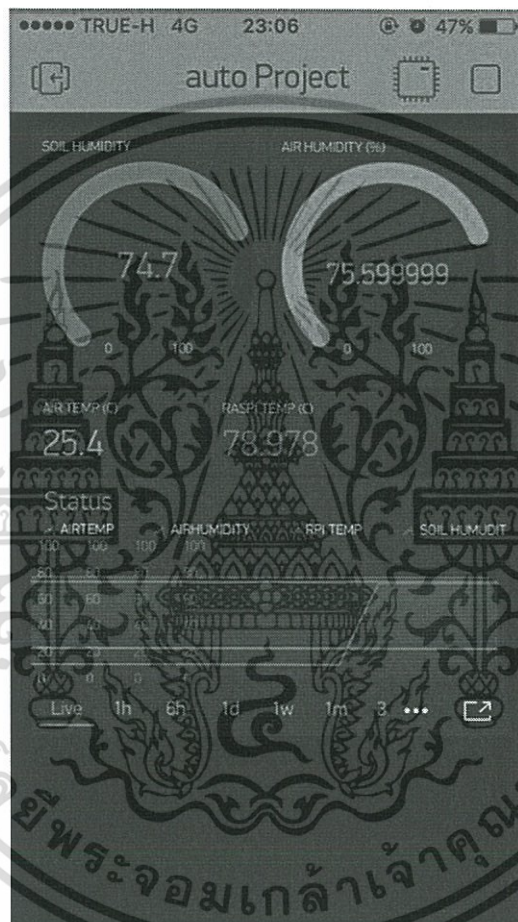


รูปที่ 4.7 เมื่อนำเซนเซอร์ทั้ง 2 ตัว ไปวัดลงบนแปลงปลูกผักบุงจิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดสอบ การทำงานบนแอปพลิเคชัน

ทดสอบการทำงานของระบบ โดยการนำเซนเซอร์วัดความชื้นในดินและเซนเซอร์วัดอุณหภูมิกับความชื้นสัมพัทธ์ของอากาศ ไปวัดลงบนแปลงปลูกผักบุงจิ้น จากนั้นจะได้ค่าออกมาแล้วส่งไปแสดงบนแอปพลิเคชัน ดังรูปที่ 4.10



รูปที่ 4.10 การแสดงผลข้อมูลบนแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

การทดสอบครั้งนี้เป็นการทดสอบเพื่อหาความแม่นยำในการทำงานและความถูกต้องของเซนเซอร์ก่อนที่จะนำไปใช้ในชิ้นงานต่อไป

5.1 สรุปผล

จากการทำปริญญานิพนธ์เรื่อง “ระบบควบคุมอัตโนมัติเพื่อการเกษตร” พบว่า การทำงานของ Soil moisture sensor นั้นมีความแม่นยำ โดยจำลองสถานะตามธรรมชาติ อีกทั้งการทำงานของ Temperature & Humidity sensor เมื่อทดสอบการทำงานของเซนเซอร์แล้วนั้นสามารถวัดอุณหภูมิและความชื้นสัมพัทธ์ของอากาศได้เช่นเดียวกัน ซึ่งส่งข้อมูลแบบ real time ตามที่ต้องการ จากนั้นข้อมูลจะถูกส่งไปยัง Node JS database และ แอปพลิเคชันที่ได้ออกแบบไว้ต่อไป อีกทั้งถ้าตรวจพบว่า มีค่าความชื้นใดๆต่ำกว่าค่าที่ตั้งไว้ ระบบก็จะสั่งให้รดน้ำให้ผักบุงจิ้นโดยอัตโนมัติ

5.2 ข้อเสนอแนะ

จากการทดลองที่ผ่านมาทำให้กลุ่มของข้าพเจ้าทราบว่า การใช้งาน Raspberry Pi นั้นมีความซับซ้อนมากกว่าที่คาดไว้ ตัวอย่างเช่น การใช้ภาษา Python ซึ่งแตกต่างจากภาษา C ทำให้ต้องศึกษาเพิ่มเติมอย่างมาก และการที่ Raspberry Pi สามารถรับได้แค่สัญญาณดิจิตอลเท่านั้น ทำให้ต้องมี ADC ที่แปลงสัญญาณแอนาล็อกเป็นดิจิตอลก่อนจึงจะสามารถนำไปใช้ได้

บรรณานุกรม

- [1] Marcacode. “แนะนำ ภาษา Python”
<http://marcuscode.com/lang/python/introduction>
- [2] Thaieasyelec. “บทความการพัฒนาโปรแกรมบน Raspberry Pi ด้วย Qt”
<http://thaieasyelec.com/article-wiki/embedded-electronics-application-บทความการพัฒนาโปรแกรมบน-raspberry-pi-ด้วย-qt.html>
- [3] Thaieasyelec. “ ADS1115 16-Bit ADC Module with Programmable Gain Amplifier”
<http://www.thaieasyelec.com/en/interface-modules/adc/ads1115-16-bit-adc-module-with-programmable-gain-amplifier-detail.html>
- [4] Thaieasyelec. “Mini Pump Motor 5V”
<http://www.thaieasyelec.com/en/home/mini-pump-motor-5v-detail.html?tmpl=component&flexiblelayout=print>
- [5] Mindphp. “Java Script”
<http://mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html>
- [6] DFRobot Electronic Product. “ Moisture Sensor (SKU:SEN0114)”
[https://www.dfrobot.com/wiki/index.php/Moisture_Sensor_\(SKU:SEN0114\)](https://www.dfrobot.com/wiki/index.php/Moisture_Sensor_(SKU:SEN0114))
- [7] Thaieasyelec. “DHT22 Digital Temperature and Humidity Sensor”
<http://www.thaieasyelec.com/en/products-en/china-direct-en/dht22-digital-temperature-and-humidity-sensor-detail.html>
- [8] APAICHON PUNOPAS. “Node JS”
<https://thaiprogrammer.org/nodejs-ตอนที่-1-nodejs-คืออะไร>
- [9] 9Arduino Research & Development. “Blynk Application”
<https://www.9arduino.com/article/59/app-สำเร็จรูป-88-1-blynk-ตอนที่-1-blynk-คืออะไร>
- [10] สมุนไพรดอทคอม. “ผักบั้งจีน”
<https://www.samunpri.com/ผักบั้งจีน>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File 1: Dht22.py

```
import os
import sys
import time
import Adafruit_DHT as dht

def callback(message):
    print(message)

while True:
    h, t = dht.read_retry(dht.DHT22,4)
    print 'Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(t,h)

time.sleep(10)
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File 2 : moist code.py

```
import time
```

```
import Adafruit_ADS1x15
```

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

```
GAIN = 1
```

```
adc.start_adc(0,GAIN)
```

```
print('=====')
```

```
print('Soil Moisture Status')
```

```
print('=====\\n')
```

```
start = time.time()
```

```
while True:
```

```
    value = int(round(adc.read_adc(0)/20.48))
```

```
    if value < 300:
```

```
        print ('Soil Stat -----> dry'.format(value))
```

```
    elif value < 700:
```

```
        print ('Soil Stat -----> Humid soil'.format(value))
```

```
    else:
```

```
        print ('Soil Stat -----> In Water'.format(value))
```

```
    f = open("soilhumid.txt", "w")
```

```
    f.write("{}".format(value))
```

```
    f.close( )
```

```
    time.sleep(5)
```

```
adc.adc_stop()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File 3: soilblynk.js

```

var blynkLib = require('blynk-library'),
sys = require('util'),
exec = require('child_process').exec,
child;
var AUTH = 'd0a69cefe7fc431c8ae6a4cf5489b4f5';
// Setup Blynk
var blynk = new blynkLib.Blynk(AUTH);
// Automatically update sensor value every 5 seconds
setInterval(function() {
  child = exec("cat /home/pi/soilhumid.txt", function (error, stdout, stderr)
  {
    if (error !== null) {
      console.log('exec error: ' + error);
    }
  }
else {
    var humudity = parseFloat(stdout)/10;
    blynk.virtualWrite(2, humudity);
    console.log('Humidity: ', humudity + '%');
  });}, 5000);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File 4: RPItemperature.js

```

var blynkLib = require('blynk-library'),
sys = require('util'),
exec = require('child_process').exec,
child;

var AUTH = 'd0a69cefe7fc431c8ae6a4cf5489b4f5';

// Setup Blynk
var blynk = new blynkLib.Blynk(AUTH);
// Automatically update sensor value every 5 seconds

setInterval(function()
{
  child = exec("cat /sys/class/thermal/thermal_zone0/temp", function (error,
stdout, stderr) {

if (error !== null) {

console.log('exec error: ' + error);

} else
{

var temp = parseFloat(stdout)/1000;

blynk.virtualWrite(5, temp);

}

});, 5000);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File 5: dht22blynk.js

```

var blynkLib = require('blynk-library');
var sensorLib = require('node-dht-sensor');
var AUTH = 'd0a69cefe7fc431c8ae6a4cf5489b4f5';

// Setup Blynk
var blynk = new blynkLib.Blynk(AUTH);
// Setup sensor, exit if failed

var sensorType = 22; // 11 for DHT11, 22 for DHT22 and AM2302
var sensorPin = 21; // The GPIO pin number for sensor signal

if (!sensorLib.initialize(sensorType, sensorPin)) {
  console.warn('Failed to initialize sensor');
  process.exit(1);
}
// Automatically update sensor value every 2 seconds
setInterval(function() {
  var readout = sensorLib.read();

  blynk.virtualWrite(3, readout.temperature.toFixed(1));

  blynk.virtualWrite(4, readout.humidity.toFixed(1));

  console.log('Temperature:', readout.temperature.toFixed(1) + 'C');

  console.log('Humidity: ', readout.humidity.toFixed(1) + '%');

}, 2000);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File 6: water.py

```

import RPi.GPIO as GPIO
import datetime
import time
import Adafruit_ADS1x15

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(pin):
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, GPIO.LOW)
    GPIO.output(pin, GPIO.HIGH)

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

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

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

time.sleep(5)
value = int(round(adc.read_adc(0)/20.48))
print("Moisture Value : %s" %value)
if value < 300:
    pump_on(pump_pin, 8)
elif value < 700:
    pump_on(pump_pin, 4)
else:
    print('Soil Stat -----> In Water')

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

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

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File 7: auto_water.py

```
import water
```

```
if __name__ == "__main__":
```

```
    water.auto_water()
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้