ระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง
CLOSED LOOP CONTROL ENVIRONMENT SYSTEM FOR CORDYCEPS MILITARIS CULTIVATION

โดย

นายธฤต วัฒนากุลกิจไพศาล

นายนภัสกร มาตังคพงศ์

นายปิยพัทธ์ ลักษณะไทย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง

CLOSED LOOP CONTROL ENVIRONMENT SYSTEM FOR CORDYCEPS MILITARIS CULTIVATION

โดย

นายธฤต	วัฒนากุลกิจไพศาล	57010602
นายนภัสกร	มาตังคพงศ์	57010657
นายปิยพัทธ์	ลักษณะไทย	57010776

อาจารย์ที่ปรึกษา

ผศ.ดร.สิรภพ ตู้ประกาย

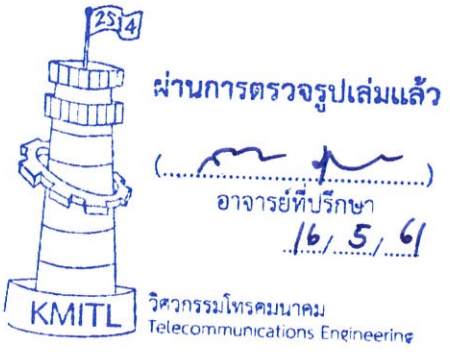
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560



ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

16/5/61

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

16/05/61

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง
CLOSED LOOP CONTROL ENVIRONMENT SYSTEM FOR CORDYCEPS MILITARIS CULTIVATION

ผู้จัดทำ

1. นายธฤต วัฒนากุลกิจไพศาล 57010602
2. นายนภัสกร มาตังคพงศ์ 57010657
3. นายปิยพัทธ์ ลักษณะไทย 57010776

..... อาจารย์ที่ปรึกษา

(ผศ.ดร.สิรภพ ตู้ประกาย)

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับการสนับสนุนและความร่วมมือจากหลายฝ่าย บุคคลแรกที่ควรถูกกล่าวถึงเป็นผู้ที่มีส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้คือ ผศ.ดร.สิรภพ ตู้ประกาย อาจารย์ที่ปรึกษาโครงงาน ผู้ให้คำปรึกษา และคำแนะนำในด้านต่างๆ และแนะแนวทางการแก้ไขปัญหาข้อบกพร่องของโครงงานนี้มาโดยตลอด รวมถึงสนับสนุนเกี่ยวกับสถานที่ทดลองสำหรับการสร้างชิ้นงาน เครื่องมือ และอุปกรณ์ต่าง ๆ ทำให้การสร้าง และการทดลองตลอดงานวิจัยผ่านไปได้ด้วยดี

ขอขอบพระคุณ บิดา มารดา และครอบครัวที่คอยให้คำแนะนำ คอยสนับสนุน ให้กำลังใจในยามที่เกิดปัญหา และรับฟังปัญหาต่างๆพร้อมให้คำปรึกษาที่ดีเสมอมา

สุดท้ายนี้ ข้าพเจ้าขอขอบพระคุณทุกท่านที่ช่วยทำให้โครงงานนี้เกิดขึ้นมา และช่วยชี้ให้เห็นถึงแนวทางการแก้ปัญหาต่างๆ ที่ทำให้โครงงานฉบับนี้สำเร็จลุล่วงไปด้วยดี ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้

นายธฤต วัฒนากุลกิจไพศาล
นายนภัสกร มาตังคพงศ์
นายปิยพัทธ์ ลักษณะไทย
ผู้จัดทำ

ระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง
CLOSED LOOP CONTROL ENVIRONMENT SYSTEM FOR CORDYCEPS MILITARIS CULTIVATION

โดย นายธฤต วัฒนากุลกิจไพศาล 57010602
นายนภัสกร มาตังคพงศ์ 57010657
นายปิยพัทธ์ ลักษณะไทย 57010776

อาจารย์ที่ปรึกษา ผศ.ดร.สิรภพ ตู้ประกาย

บทคัดย่อ

ปริญญานิพนธ์นี้ มีวัตถุประสงค์เพื่อออกแบบ และสร้างระบบควบคุมสภาพแวดล้อม รวมถึงการแสดงค่าสถานะของระบบควบคุมอุณหภูมิ ความชื้น และแสงสว่าง ภายในสภาพแวดล้อมแบบปิด ที่สามารถติดตามการเจริญเติมโตได้จากฐานข้อมูลที่บ่งบอกถึงค่าสถานะภายในระบบสำหรับการเพาะเลี้ยงเห็ดถั่งเช่าสีทอง โดยอุณหภูมิที่เห็ดถั่งเช่าสีทองจะเจริญเติมโตได้ดีในช่วงระยะการกินสารอาหารของเชื้อเห็ดอยู่ที่อุณหภูมิ 23 องศาเซลเซียส ในระยะการเกิดดอกสีเหลืองอยู่ที่อุณหภูมิ 18 องศาเซลเซียส ความชื้นสัมพัทธ์ที่ 75-95 เปอร์เซ็นต์ และแสงสว่างจากหลอดไฟที่มีค่าประมาณ 1000 LUX ซึ่งการควบคุมระบบจะใช้ไมโครคอนโทรลเลอร์ในการควบคุมอุปกรณ์ ผลการทดลองระบบสามารถควมคุมอุณหภูมิ ความชื้น และแสงสว่างให้เปลี่ยนแปลงตามสภาพแวดล้อมที่เหมาะสมกับการเจริญเติบโตของเห็ดถั่งเช่าสีทอง และทำการเก็บข้อมูลการทำงานของระบบบนฐานข้อมูล และเมื่อครบตามจำนวนวันที่กำหนด ภายในโหลเห็ดจะปรากฏดอกเห็ดถั่งเช่าสีทอง ที่งอกขึ้นมามีสีเหลืองทองสวยงาม โดยที่สภาพของเห็ดในแต่ละช่วงอายุ และค่าสถานะต่าง ๆ ภายในระบบสามารถติดตามได้จากเว็บบราวเซอร์

ABSTRACT

The objective of this thesis is to design and built up closed environment control system together with displaying the value of temperature, humidity and illumination in closed environment that can following up the growth of Cordyceps Militaris from the database that indicates the state of the environment for

mushroom cultivation. The temperature that Cordyceps Militaris grows well is 23 degree Celsius in incubation period. for blooming period it require 18 degree Celsius of temperature 75-95 percent of humidity and 1000 LUX of illumination. The system controlled by microcontroller and the experiment result show that the system can control temperature, humidity and illumination for the requirement of Cordyceps Militaris growth and store data of system operation in database. Finally Cordyceps Militaris will appear in the jars with beautiful yellow-gold color and all of the status of Cordyceps Militaris growing can following up from web browser.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		IV
สารบัญรูป		VIII
สารบัญตาราง		XI
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 เห็ดถั่งเช่าสีทอง	3
	2.1.1 สารสำคัญในเห็ดถั่งเช่าสีทอง	3
	2.1.2 สรรพคุณของเห็ดถั่งเช่าสีทอง	4
	2.1.3 วิธีการเพาะเห็ดถั่งเช่าสีทอง	5
	2.2 RASPBERRY PI 3 MODEL B	7
	2.2.1 สเปค RASPBERRY PI 3 MODEL B	7
	2.2.2 การเชื่อมต่ออุปกรณ์	8
	2.2.3 ขั้นตอนการทำ RASPBERRY PI ให้เป็น WEB SERVER	9
	2.3 รีเลย์โมดูล	14
	2.4 TEMPERATURE AND HUMIDITY MODULE AM2302	16
	2.5 BH1750FVI (GY-302) LIGHT SENSOR MODULE	17
	2.6 THERMOELECTRIC COOLER PELTIER	18
	2.6.1 หลักการทำงานของ PELTIER	19
	2.6.2 การนำ PELTIER มาใช้งาน	20
	2.7 ULTRASONIC MIST MAKER	20

สารบัญ (ต่อ)

		หน้า
	2.8 ภาษา PYTHON	21
	2.9 ภาษา PHP	22
	2.9.1 ความสามารถของภาษา PHP	22
	2.10 ภาษา SQL	23
	2.10.1 SQL SYNTAX	23
	2.11 ภาษา HTML	24
	2.12 NETPIE	24
	2.12.1 MICROGEAR	26
	2.12.2 ฟังก์ชั่นหลักของ MICROGEAR	27
	2.12.3 FREEBOARD	28
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**30**
	3.1 การออกแบบ	30
	3.1.1 ออกแบบลำดับการทำงานของโปรแกรมในส่วนของการควบคุมความชื้น	30
	3.1.2 ออกแบบลำดับการทำงานของโปรแกรมในส่วนของการควบคุมแสงสว่าง	31
	3.1.3 การออกแบบตู้ที่ติดตั้งระบบควบคุมอุณหภูมิ ความชื้น และแสง	33
	3.1.4 การออกแบบ และสร้างฐานข้อมูล	35
	3.1.5 ออกแบบการแสดงบนหน้าเว็บไซต์ NETPIE.IO	35
	3.2 เครื่องมือที่ใช้ในการทดลอง	37
	3.3 การจัดเก็บผลการทดลอง	38
	3.3.1 การทดลองวัดสัญญาณเอาต์พุตจาก เซนเซอร์วัดอุณหภูมิ และความชื้นด้วยเครื่องออสซิลโลสโคป	41
	3.3.2 การทดลองการทำงานของเซนเซอร์	41
	3.3.3 การทดลองการแสดงผลอุณหภูมิ ความชื้น และความเข้มแสงเว็บบราวเซอร์	41
	3.3.4 ผลการทดสอบการแสดงค่าอุณหภูมิ ความชื้น และความเข้มแสงจากฐานข้อมูล	42

สารบัญ (ต่อ)

หน้า

3.3.5 การทดสอบหน้าเว็บแสดงภาพที่ถูกถ่ายด้วย PICAMERA MODULE 42

3.3.6 การทดสอบปลูกเห็ดถั่งเช่าสีทองในระบบควบคุมสภาพแวดล้อมแบบปิด 42

บทที่ 4 ผลการทดลอง 43

4.1 ผลทดลองวัดสัญญาณเอาต์พุตจาก เซนเซอร์วัดอุณหภูมิ และความชื้นด้วยเครื่องออสซิลโลสโคป 43

4.2ผลทดลองการทำงานของเซนเซอร์ 45

4.2.1 การทดลองวัดอุณหภูมิของเซ็นเซอร์อุณหภูมิ DHT22 เทียบกับเทอร์โมมิเตอร์ 45

4.2.2 ผลการทดลองวัดความเข้มแสงของเซนเซอร์ BH1750FVI เทียบกับ LUX METER YOKOGAWA 51002 46

4.3 ผลทดลองการแสดงผลอุณหภูมิ ความชื้น และความเข้มแสงเว็บบราวเซอร์ 49

4.4 ผลการทดสอบการแสดงค่าอุณหภูมิ ความชื้น และความเข้มแสงจากฐานข้อมูล 50

4.5 ผลการทดสอบหน้าเว็บแสดงภาพที่ถูกถ่ายด้วย PICAMERA MODULE 51

4.6 ผลการทดสอบปลูกเห็ดถั่งเช่าสีทองในระบบควบคุมสภาพแวดล้อมแบบปิด 53

บทที่ 5 สรุปผลและข้อเสนอแนะ 56

5.1 สรุปผล 56

5.2 ข้อเสนอแนะ 56

บรรณานุกรม 57

ภาคผนวก ก DATASHEET HUMIDITY & TEMPERATURE SENSORAM2302/DHT22 59

ภาคผนวก ข DATASHEET LIGHT SENSOR IC BH1750FVI 65

สารบัญ (ต่อ)

หน้า

ภาคผนวก ค โค้ดการทำงานของระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง 74

สารบัญรูป

รูปที่		หน้า
1.1	บล็อกไดอะแกรมโดยรวมรวมของปริญญานิพนธ์	2
2.1	แผนภาพการเพาะเลี้ยงเห็ดถั่งเช่าสีทอง	6
2.2.1	ตำแหน่งของอุปกรณ์บน RASPBERRY PI 3 MODEL B	7
2.2.2	คำสั่งแสดงการติดตั้ง APACHE บน RASPBERRY PI	9
2.2.3	แสดงการทำงานของ APACHE ว่าสามารถทำงานได้ปกติ	9
2.2.4	คำสั่งแสดงการติดตั้ง PHP บน RASPBERRY PI	10
2.2.5	การเปลี่ยน นามสกุลไฟล์ให้เป็น PHP โดยใช้คำสั่ง MV INDEX.HTML INDEX.PHP	10
2.2.6	การเปิดไฟล์ INDEX.PHP โดยใช้คำสั่ง SUDO NANO INDEX.PHP	11
2.2.7	การเปิด WEB BROWSER โดยกำหนดเป็น IP ADDRESS ของ RASPBERRY PI	11
2.2.8	การติดตั้ง MYSQL โดยใช้คำสั่ง SUDO APT-GET INSTALL MYSQL-SERVER PHP5-MYSQL -Y	12
2.2.9	การติดตั้ง PHPMYADMIN โดยใช้คำสั่ง SUDO APT-GET INSTALL PHPMYADMIN –Y	13
2.2.10	หน้า WEB BROWSER บน RASPBERRY PI เมื่อพิมพ์ HTTP://LOCALHOST/PHPMYADMIN	13
2.2.11	หน้า WEB BROWSER เมื่อเข้าสู่ PHPMYADMIN	14
2.3.1	รีเลย์โมดูล	15
2.3.2	ขาอินพุตของรีเลย์โมดูล	15
2.3.3	ขาเอาต์พุตของรีเลย์โมดูล	16
2.4	DHT22 TEMPERATURE AND HUMIDITY ON MODULE AM2302	16
2.5.1	BH1750FVI (GY-302) LIGHT SENSOR MODULE	17
2.5.2	การใช้งานของ BH1750FVI	18
2.6.1	COOLER PELTIER	19
2.6.2	การทำงานของPELTIER	20
2.7	ULTRASONIC MIST MAKER	21
2.12.1	วิธีการสื่อสารของสิ่งต่างๆ ผ่าน NETPIE	25
2.12.2	ลำดับการเรียกฟังก์ชั่นพื้นฐานของ MICROGEAR	27

สารบัญรูป(ต่อ)

		หน้า
2.12.3	DASHBOARD ของ NETPIE FREEBOARD	29
3.1	แผนภาพการทำงานของระบบควบคุมความชื้น	30
3.2	แผนการทำงานของระบบควบคุมแสง (ระยะที่เห็ดต้องได้รับแสงเพื่อให้ใยเหลืองปกคลุมอาหาร)	31
3.3	ออกแบบลำดับการทำงานของระบบควบคุมแสง (ระยะที่เห็ดต้องได้รับแสงเพื่อออกดอก)	32
3.4	ตู้ควบคุมสภาพแวดล้อมแบบปิดเพาะเลี้ยงเห็ดถั่งเช่าสีทอง	33
3.5	ฮีทซิงค์ และพัดลม	34
3.6	ตำแหน่งที่ทำการติดตั้งหลอดไฟ LED	34
3.7	หน้าเว็บ LOCALHOST/PHPMYADMIN ที่รับค่าจากเซนเซอร์วัดอุณหภูมิ ความชื้น และเซนเซอร์วัดความเข้มแสง	35
3.8	หน้าเว็บแสดงผลผ่านหน้าเว็บไซต์ NETPIE.IO บน NETPIE FREEBOARD	36
3.9	ลำดับการเข้าถึงในส่วนของการเข้าสู่ NETPIE.IO และแสดงบนหน้า NETPIE FREEBOARD	37
3.10	RASPBERRY PI 3 MODEL B	38
3.11	รีเลย์โมดูล	38
3.12	RASPBERRY PI CAMERA BOARD	39
3.13	DHT22 TEMPERATURE AND HUMIDITY SENSOR MODULE	39
3.14	BH1750FVI (GY-302) LIGHT SENSOR MODULE	40
3.15	ULTRASONIC MIST MAKER	40
3.16	COOLER PELTIER	41
4.1	สัญญาณเอาต์พุตจากเซนเซอร์อุณหภูมิ และความชื้น	43
4.2	การอ่านบล็อกข้อมูลค่าความชื้น	44
4.3	การอ่านบล็อกข้อมูลค่าอุณหภูมิ	45
4.4	ตำแหน่งในการวัดแสงของเซนเซอร์กับ LUX METER	48
4.5	หน้าบราวเซอร์ NETPIE FREEBOARD แสดงสภานะค่าอุณหภูมิ ความชื้น และแสง	49
4.6	หน้าเว็บบราวเซอร์แสดงหน้าการค้นหาข้อมูลตัวแปรต่าง ๆ จากฐานข้อมูล	50

สารบัญรูป(ต่อ)

		หน้า
4.7	หน้าเว็บแสดงหน้าการค้นหาข้อมูลตัวแปรต่าง ๆ จากฐานข้อมูลเมื่อทำการกดค้นหา	51
4.8	หน้าเว็บแสดงหน้าการค้นหา	52
4.9	หน้าเว็บบราวเซอร์แสดงหน้าเมื่อทำการกดค้นหารูปถ่าย และแสดงภาพที่ถูกถ่ายบนหน้าเว็บบราวเซอร์	52
4.10	ทดลองเพาะเห็ดถั่งเช่าสีทองเดือนที่ 1	53
4.11	ทดลองเพาะเห็ดถั่งเช่าสีทองเดือนที่ 2	53
4.12	กราฟค่าอุณหภูมิจากเซนเซอร์ DHT22 ภายในระบบในช่วงเวลา 1 ชั่วโมง	54
4.13	กราฟค่าความชื้นจากเซนเซอร์ DHT22 ภายในระบบในช่วงเวลา 1 ชั่วโมง	55
4.14	กราฟค่าความความเข้มแสงจากเซนเซอร์ BH1750FVI ภายในระบบในช่วงเวลา 1 ชั่วโมง	55

สารบัญตาราง

ตารางที่		หน้า
4.1	ผลการทดลองของเซนเซอร์AM2302เทียบกับเทอร์โมมิเตอร์	45
4.2	ผลการทดสอบเซนเซอร์อุณหภูมิเทียบกับ LUX METER YOKOGAWA 51002	46
4.3	ผลการทดสอบเซ็นเซอร์โดยการวัดตามตำแหน่ง	48

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันประเทศไทยกำลังอยู่ในช่วงการเปลี่ยนผ่านเข้าสู่สังคมผู้สูงอายุ การที่จะต้องดูแลร่างกายให้แข็งแรงจึงเป็นสิ่งสำคัญ ซึ่งเห็ดถั่งเช่าสีทองที่เป็นที่รู้จักแพร่หลาย และเป็นที่นิยมมาตั้งแต่อดีตจนถึงปัจจุบัน จนถึงทุกวันนี้เห็ดถั่งเช่าสีทองมีการเพาะเลี้ยงมากขึ้น จึงเกิดความนิยมมากขึ้นจากค่านิยมการดูแลสุขภาพร่างกายของคนในยุคปัจจุบัน เพราะมีความเชื่อว่าช่วยบำรุงกำลัง เสริมภูมิคุ้มกัน เสริมสมรรถภาพทางเพศ และป้องกันโรคมะเร็ง ธรรมชาติของเห็ดถั่งเช่าสีทองเกิด และออกดอกในเฉพาะช่วงฤดูหนาวเท่านั้น ช่วงอุณหภูมิที่เหมาะสมที่สุดของเห็ดถั่งเช่าสีทองคือ 18-25 องศาเซลเซียส ในระดับความชื้นสัมพัทธ์ 75-95 เปอร์เซ็นต์ ปัจจุบันมีการสร้างสถานที่จำลองสภาพแวดล้อมสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทองขึ้นมาเพื่อให้สามารถเพาะเลี้ยงได้ตามบ้านเรือน สำหรับความต้องการของผู้บริโภคในปัจจุบันที่มากขึ้น รวมทั้งยังเพิ่มรายได้ให้กับผู้สนใจเพาะเลี้ยง ซึ่งในระบบควบคุมอุณหภูมิ ความชื้น แสงสว่าง นั้นมีระบบ และอุปกรณ์ที่ใช้หลายชนิด ไมโครคอนโทรลเลอร์ที่เป็นอุปกรณ์หนึ่งที่สะดวกต่อการใช้งาน หาซื้อได้ง่าย และใช้กันแพร่หลายจึงได้นำไมโครคอนโทรลเลอร์มาใช้ ซึ่งสามารถควบคุมการทำงานระบบควบคุมอุณหภูมิ ความชื้น และแสงสว่างได้อย่างเหมาะสมต่อการเพาะเลี้ยงเห็ดถั่งเช่าสีทองได้ทุกฤดูกาล ตอบสนองความต้องการสำหรับค่านิยมการดูแลสุขภาพมากขึ้น ระบบสามารถที่จะติดตามการเจริญเติมโต และควบคุมระบบได้จากระยะไกล ซึ่งมีประโยชน์สำหรับผู้ที่ไม่มีเวลาที่จะดูแล สังเกต หรือเฝ้าดูได้ตลอดเวลา ดังนั้นระบบติดตามการเจริญโตจึงมีความสำคัญ ทำให้ผู้เพาะเลี้ยงทราบถึงการเจริญเติมโตของเห็ดถั่งเช่าสีทอง

1.2 วัตถุประสงค์

1) ออกแบบ และสร้างระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง โดยควบคุมอุณหภูมิ ความชื้น แสง และปัจจัยต่างๆที่ส่งผลต่อการเจริญเติบโตของเห็ดถั่งเช่าสีทอง

2) ออกแบบ และติดตามการเจริญเติบโตของเห็ดถั่งเช่าสีทองจากระยะไกล

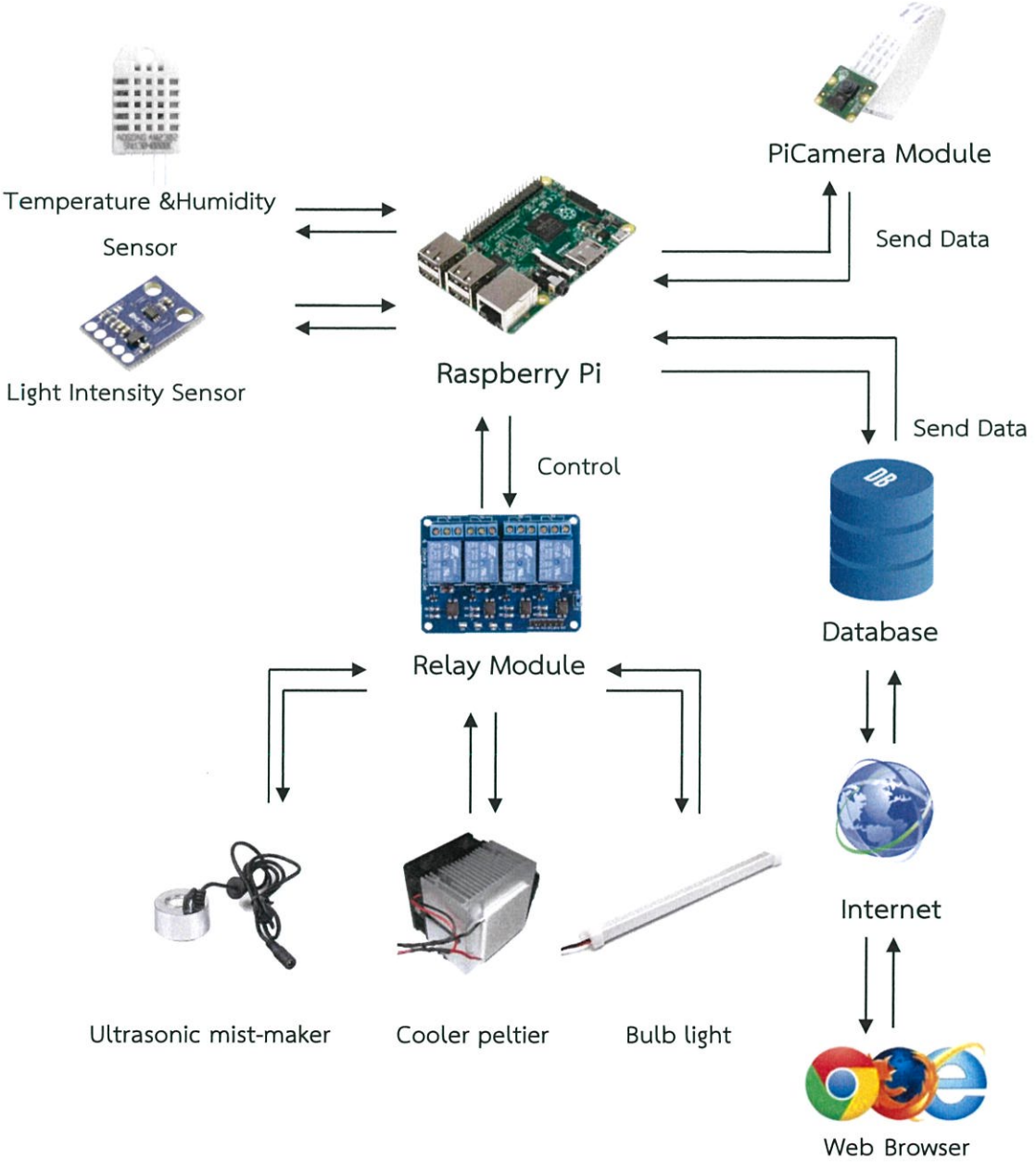
3) เพื่อศึกษา และสร้างการจัดเก็บฐานข้อมูลสำหรับการเพาะเลี้ยงเห็ดถั่งเช่าสีทอง

1.3 ขอบเขตของปริญญานิพนธ์

1) ใช้เซนเซอร์สำหรับวัดอุณหภูมิ วัดความชื้น และวัดความสว่างของแสง เพื่อวัดปัจจัยสภาพแวดล้อมภายในระบบปิดที่ถูกประมวลผลด้วยไมโครคอนโทรลเลอร์ และส่งไปยังฐานข้อมูลแสดงผ่านเว็บบราวเซอร์

2) ระบบปรับสภาพภายในแบบปิดใช้อุปกรณ์ปรับอุณหภูมิ Cooler Peltier อุปกรณ์สร้างความชื้น Ultrasonic mist-maker และหลอดไฟให้แสงว่าง โดยทั้งหมดถูกควบคุมด้วยไมโครคอนโทรลเลอร์

3) สามารถแจ้งเตือนสถานะได้เมื่อระบบที่ถูกควบคุมมีความผิดปกติผ่านหน้าเว็บบราวเซอร์



รูปที่ 1.1 บล็อกไดอะแกรมโดยรวมรวมของปริญญานิพนธ์

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 เห็ดถั่งเช่าสีทอง

เห็ดถั่งเช่าสีทอง (Cordyceps Militaris) เป็นปรสิตที่ขึ้นได้ในแมลงหลากหลายชนิด ส่วนใหญ่เกิดในตัวหนอนและดักแด้ผีเสื้อโดยเห็ดถั่งเช่าสีทอง สายพันธุ์ Militaris ซึ่งเป็นเห็ดตระกูลเดียวกับเห็ดถั่งเช่าทิเบต (Cordyceps Sinensis) แต่ต่างสายพันธุ์กัน เห็ดถั่งเช่าสีทองได้มีการเพาะเลี้ยงเป็นการค้ามานานหลายสิบปีแล้วที่ประเทศจีน เกาหลี และญี่ปุ่น ซึ่งทำการเพาะเลี้ยงได้ง่ายโดยสามารถเพาะได้ด้วยการใช้แมลงหลายชนิด เช่น ตัวหนอนหรือดักแด้ไหม หรือใช้อาหารสังเคราะห์ ชักนำให้ออกเป็นดอกเห็ดในที่อากาศเย็นที่มีอุณหภูมิระหว่าง 10-28 องศาเซลเซียส ชาวจีนเชื่อว่าเห็ดถั่งเช่าสีทอง เป็นยาอายุวัฒนะ มีสรรพคุณช่วยปรับสมดุลของร่างกาย เป็นสมุนไพรธาตุร้อน ในสมัยโบราณเห็ดถั่งเช่าถูกจำกัดการใช้เฉพาะจักรพรรดิ และเชื้อพระวงศ์ชั้นสูงของจีนเท่านั้น คนธรรมดาสามัญไม่มีสิทธิ์บริโภค เป็นของที่หายาก และมีค่าดั่งทองคำ ตำราการแพทย์ทิเบตมีการบันทึกไว้ว่า เห็ดถั่งเช่าถูกใช้เป็นยาชูกำลัง ใช้รักษาสารพัดโรค และเป็นยาบำรุงร่างกายชั้นยอด ที่ได้จากการผสมผสานกันระหว่างตัวหนอนและเห็ดตังถั่งเช่า หรือ ตังถั่งแห่เช่า แปลว่า "ฤดูหนาวเป็นหนอน ฤดูร้อนเป็นหญ้า" มีรสหวาน ฤทธิ์ไม่ร้อน เข้าเส้นลมปราณไต บำรุงไต เสริมภูมิคุ้มกัน และพลังชีวิต แก้อาการอ่อนเพลีย ภูมิแพ้ แก้ไอ ละลายเสมหะ หอบหืด ไอเรื้อรัง อาการหย่อนสมรรถภาพทางเพศ เข่าอ่อน เอวอ่อน ทำให้แก่ช้า และเป็นยาบำรุงสำหรับผู้ป่วยฟื้นไข้

2.1.1 สารสำคัญในเห็ดถั่งเช่าสีทอง

เหตุที่ตัวของเห็ดถั่งเช่าสีทองนั้น มีสรรพคุณต่าง ๆ มากมายนั้น เป็นเพราะว่าในตัวของเห็ดถั่งเช่าสีทอง มีสารออกฤทธิ์ที่สำคัญอยู่ภายใน โดยสารออกฤทธิ์ที่สำคัญของเห็ดถั่งเช่าสีทองที่ค่อนข้างเป็นประโยชน์และได้รับการพิสูจน์ทางวิทยาศาสตร์แล้ว มีดังนี้

1) Cordycepin สารตัวนี้ออกฤทธิ์ ได้เทียบเท่ากับฮอร์โมนเพศชาย โดยจะช่วยเสริมสร้างการทำงานของสมรรถภาพทางเพศ ของผู้ชายให้ทำงานได้อย่างเต็มที่ และมีประสิทธิภาพมากขึ้น

2) Nitric Oxides เป็นสารที่ช่วยในกระบวนการแข็งตัวของอวัยวะเพศชายให้ทำงานได้ดียิ่งขึ้น แข็งเร็ว และนานขึ้น โดยสารตัวนี้จะออกฤทธิ์ในการขยายหลอดเลือดให้เข้าสู่องคชาติมากขึ้น ทำให้การแข็งตัวของอวัยวะเพศนานขึ้นอย่างสมบูรณ์

3) Adrenaline สารตัวนี้จะช่วยเพิ่มพลังงานให้กับร่างกาย จะทำให้ร่างกายแข็งแรงขึ้น ไม่มีอาการอ่อนเพลีย และสามารถพักผ่อนได้อย่างเต็มที่ หลับเต็มอิ่มมากขึ้น อีกทั้งยังช่วยชะลอความแก่ได้อีกด้วย

4) Polysaccharide เป็นสารที่ช่วยเสริมสร้างภูมิคุ้มกันให้กับร่างกาย โดยจะช่วยสร้างกลไกลการป้องกันโรค และป้องกันการเกิดมะเร็งได้เป็นอย่างดี

5) Adenosine สารตัวนี้ถูกใช้ในการรักษาอาการหัวใจเต้นผิดจังหวะ โดยจะชะลอคลื่นไฟฟ้าที่ถูกส่งไปยังหัวใจ ส่งผลให้หัวใจเต้นช้าลง และกลับมาสู่ภาวะการเต้นที่เป็นปกติ

6) Mannitol เป็นสารให้ความหวานที่มีแคลรอรี่ต่ำโดยจะให้พลังงานเพียงครึ่งหนึ่งของน้ำตาลทราย ถูกใช้ในอาหารสำหรับผู้ที่ควบคุมน้ำหนักและ ผู้ป่วยโรคเบาหวาน

7) Ergosterol ทำหน้าที่ช่วยเสริมสร้างความแข็งแรงของกระดูกและฟัน โดยจะสามารถสังเคราะห์วิตามินดี2 เพื่อช่วยในการดูดซึมแคลเซียมและฟอสฟอรัสในลำไส้

2.1.2 สรรพคุณของเห็ดถั่งเช่าสีทอง

เห็ดถั่งเช่าสีทองถือเป็นสมุนไพรที่อุดมไปด้วยสารที่มีประโยชน์ต่อร่างกายไม่ว่าจะเป็น นิวคลีโอไทด์ (adenosine) กรดคอร์ไดเซฟิน (Cordycepin Acid) ซึ่งเป็นสารที่มีเฉพาะในเห็ดถั่งเช่า อีกทั้งยังมีกรดอะมิโน และเออร์โกสเตอรอล (Ergosterol) ที่มีฤทธิ์ในการต้านเชื้อรา ไม่เพียงเท่านั้น ยังมีสารอาหารที่สำคัญ อาทิ โปรตีน วิตามิน E วิตามิน K วิตามิน B1 วิตามิน B2 วิตามิน B12 โพแทสเซียม โซเดียม แคลเซียม แมกนีเซียม เหล็ก สังกะสี และซิลิเนียม จึงทำให้ถั่งเช่ากลายเป็นสมุนไพรที่ถูกนำมาใช้ในการบำรุงสุขภาพและรักษาอาการบางชนิด ซึ่งสรรพคุณต่าง ๆ ที่กล่าวมาของเห็ดถั่งเช่าสีทองมีดังต่อไปนี้

1) ช่วยปรับการทำงานของหัวใจ ในเรื่องของหัวใจถั่งเช่าถือเป็นสมุนไพรที่มีสรรพคุณช่วยปรับอัตราการเต้นของหัวใจให้เป็นปกติได้ อีกทั้งยังช่วยบรรเทาอาการหัวใจขาดออกซิเจน และเพิ่มออกซิเจนให้หัวใจได้ เหมาะสำหรับบำรุงผู้ที่ป่วยเป็นโรคหัวใจเต้นผิดจังหวะ

2) เสริมสร้างการทำงานของระบบภูมิคุ้มกัน ถั่งเช่ามีสรรพคุณช่วยปรับปรุงการทำงานของระบบภูมิคุ้มกันให้เป็นปกติ ช่วยให้ร่างกายสร้างเซลล์ภูมิคุ้มกันมากขึ้น กระตุ้นการสร้างแอนติบอดีในร่างกาย เพื่อเพิ่มจำนวนของเซลล์ที่ใช้เพื่อกำจัดเซลล์แปลกปลอมหรือเซลล์ที่ตายแล้ว แต่ก็ต้องใช้ในปริมาณที่พอเหมาะ เพราะการใช้มากเกินไปสารในถั่งเช่าอาจไปกดการทำงานบางอย่างของระบบภูมิคุ้มกันได้

3) ต้านมะเร็ง นอกจากเสริมสร้างระบบภูมิคุ้มกันแล้ว ถั่งเช่าก็ยังมีฤทธิ์ในการต้านมะเร็ง โดยสารในถั่งเช่าถือเป็นสารที่มีความสำคัญในการต่อต้านการเกิดมะเร็ง ป้องกันการเกิดและการแพร่กระจายของเนื้อร้าย รวมทั้งยังป้องกันไม่ให้ผู้ป่วยโรคมะเร็งที่รักษาหายแล้วกลับมาเป็นซ้ำอีกด้วย

4) ลดไขมันในเลือด อีกสรรพคุณหนึ่งที่น่าสนใจจะเป็นการควบคุมระดับไขมันในเลือด ลดคอเลสเตอรอล และไตรกลีเซอร์ไรด์ ซึ่งเป็นสาเหตุของโรคภัยอื่น ๆ อย่างเช่น โรคหัวใจและหลอดเลือด นอกจากนี้ก็ยังเพิ่มปริมาณคอเลสเตอรอลชนิดที่ดี ซึ่งจะส่งผลโดยตรงต่อการลดความเสี่ยงโรคหลอดเลือดแดงแข็งได้

5) ฟื้นฟูการทำงานของไต สำหรับผู้ป่วยโรคไตเรื้อรัง การรับประทานถั่งเช่าจะช่วยบรรเทาอาการลง และทำให้สุขภาพไตดีขึ้น อีกทั้งยังลดความเสียหายของไตที่เกิดจากสารพิษตกค้างได้

6) เสริมสร้างการทำงานของตับ สารพิษเป็นสาเหตุหนึ่งที่ทำให้ตับถูกทำลาย การกินถั่งเช่าเป็นอาหารเสริมจะช่วยลดผลกระทบจากสารพิษ และป้องกันการเกิดพังพืดในตับ ขณะที่สารต้านอนุมูลอิสระก็ยังเข้าไปเพิ่มประสิทธิภาพการทำงานของระบบภูมิคุ้มกัน ลดความเสี่ยงในการเกิดโรคไวรัสตับอักเสบได้อีกด้วย

7) บำรุงโลหิต นอกจากจะบำรุงตับ ไต รวมทั้งหัวใจแล้ว สารที่อยู่ในถั่งเช่าก็ยังช่วยเสริมสร้างการทำงานของระบบโลหิต ทำให้ร่างกายสร้างไขกระดูกมากขึ้นซึ่งทำให้เซลล์เม็ดเลือดแดงและเซลล์เม็ดเลือดขาวถูกสร้างในปริมาณที่เพียงพอต่อร่างกาย

8) ลดระดับน้ำตาลในเลือด สำหรับผู้ป่วยเบาหวาน ถั่งเช่าถือเป็นสมุนไพรอีกชนิดที่ช่วยลดน้ำตาลได้ โดยมีการศึกษาพบว่า การรับประทานถั่งเช่าวันละ 3 กรัม จะช่วยควบคุมระดับน้ำตาลในเลือดได้ถึง 95% ซึ่งมากกว่าการใช้ยาแผนปัจจุบันที่ควบคุมได้เพียงแค่ 54%

อย่างไรก็ตาม ยังมีกลุ่มบุคคลที่ไม่เหมาะกับการรับประทานเห็ดถั่งเช่าสีทองอยู่บ้าง เช่นผู้ป่วยที่ต้องรับยากลุ่มป้องกันการเกาะกลุ่มกันของเกล็ดเลือด และผู้ป่วยที่ต้องรับยากดภูมิคุ้มกัน เพราะถั่งเช่ามีฤทธิ์ช่วยลดระดับน้ำตาลในเลือด ต้านการเกาะกลุ่มของเกล็ดเลือด และกระตุ้นระบบภูมิคุ้มกัน ซึ่งอาจเสริมกับฤทธิ์ของยาปฏิชีวนะที่กำลังทานอยู่ จนให้ผลมากเกินไปกับร่างกาย จนเกิดเป็นอันตรายได้

2.1.3 วิธีการเพาะเห็ดถั่งเช่าสีทอง

ในปัจจุบันการเพาะเลี้ยงเห็ดสามารถเพาะเลี้ยงบนอาหารแข็ง โดยมีวิธีการเพาะ ข้อดีและข้อเสีย ดังนี้

2.1.3.1 การเพาะเลี้ยงเห็ดถั่งเช่าสีทองบนอาหารแข็ง

การเพาะเลี้ยงบนอาหารแข็ง คือ การเพาะเลี้ยงบนอาหารที่เป็นของแข็ง หรือกึ่งแข็ง เช่นการเพาะเลี้ยงบนเมล็ดพืช ซากแมลง หรือบนวุ้นเลี้ยงเชื้อ วิธีการเพาะเลี้ยงแบบนี้มีข้อดี คือ สามารถสังเกตการปนเปื้อน ได้ง่าย เชื้อที่ได้จะมีความแข็งแรง แต่ข้อเสียคือ เห็ดจะมีอัตราการเติบโตที่ต่ำกว่าเพาะบนอาหารเหลว ซึ่งการเพาะนั้นจะมีวิธีการดังนี้

1) ในส่วนแรกนำมันฝรั่ง 20 กรัม มาต้มในน้ำ 1 ลิตร ให้พอเดือดสังเกตว่าน้ำเปลี่ยนเป็นสีเหลืองใส

2) ยกลงแล้วกรองเอาเฉพาะน้ำมันฝรั่ง

3) เติมกลูโคส 20 กรัมลงในน้ำมันฝรั่ง คนจนเข้ากัน แล้วเติมผงวุ้น 15 กรัม ลงในส่วนผสม และคนจนละลายเป็นเนื้อเดียวกัน

4) แบ่งใส่ขวดหรือโหลที่นำไปลวกน้ำร้อนแล้ว โดยบรรจุ ¼ หรือ 2/5 ของปริมาตรขวด ปิดด้วยจุกสำลี แล้วหุ้มปากขวดและจุกสำลีด้วยกระดาษ รัดยาง และนำไปนึ่งฆ่าเชื้อที่ 121 องศาเซลเซียส ความดัน 15 ปอนด์ต่อตารางนิ้ว เป็นเวลา 15 นาที

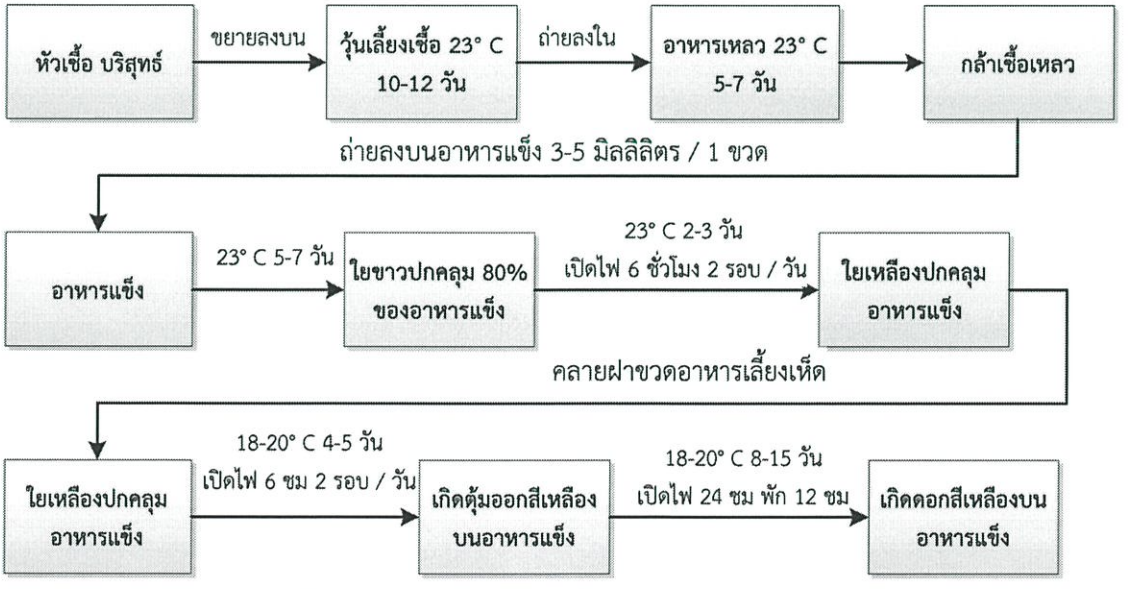
5) ทิ้งพอให้อุ่น และวุ้นยังไม่แข็งเป็นก้อน แล้ววางขวดนอนทำมุม 30 องศาเซลเซียส ตั้งทิ้งไว้ 1 ชั่วโมง จะได้ขวดที่บรรจุวุ้นเลี้ยงเชื้อพร้อมใช้งาน

6) เตรียมข้าวเสาไห้ ผสมกับน้ำในอัตราส่วน 1:1 จากนั้นปิดฝาขวดไม่ต้องแน่นมาก แล้วนำไปนึ่งเพื่อฆ่าเชื้อที่ 121 องศาเซลเซียส ความดัน 15 ปอนด์ต่อตารางนิ้ว เป็นเวลา 15 นาที

7) เมื่อนึ่งเสร็จแล้ว รอให้เย็น แล้วนำเชื้อที่เตรียมไว้ ฉีดลงไปในขวดที่ใส่ข้าว แล้วนำไปวางไว้ในที่ควบคุมสภาพแวดล้อม

2.1.3.2 สภาพแวดล้อมของการเพาะเลี้ยงเห็ดถั่งเช่าสีทอง

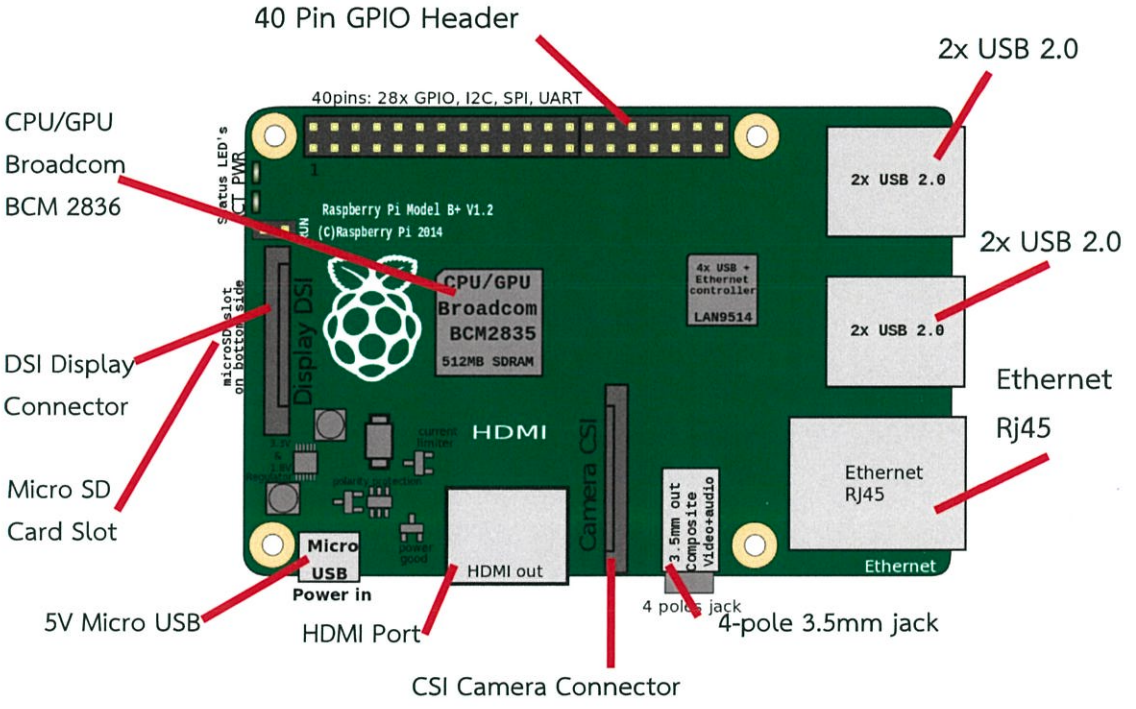
การควบคุมสภาพแวดล้อมในสภาวะต่าง ๆ ให้มีความคงที่นั้น มีความสำคัญสำหรับการเพาะเลี้ยงเห็ดถั่งเช่าสีทอง เป็นอย่างมาก ซึ่งความชื้นในสภาพแวดล้อม ควรมีความชื้นไม่น้อยกว่า 75%-95% และมีสภาวะแวดล้อมในระยะต่าง ๆ ดังรูปที่ 2.1



รูปที่ 2.1 แผนภาพการเพาะเลี้ยงเห็ดถั่งเช่าสีทอง [3]

2.2 Raspberry Pi 3 Model B

Raspberry Pi (ราสเบอรี่พาย) หรือขอเรียกย่อๆว่า RPi เป็นเครื่องคอมพิวเตอร์ขนาดเล็ก ออกแบบโดย Raspberry Pi Foundation (www.raspberrypi.org) สำหรับใช้ในการทดลองและการศึกษา มีพอร์ตหรือช่องต่ออุปกรณ์ เช่น HDMI, USB, Ethernet



รูปที่ 2.2.1 ตำแหน่งของอุปกรณ์บน Raspberry Pi 3 Model B

2.2.1 สเปค Raspberry Pi 3 Model B

1) ขนาด 85 x 56 x 17 มม.
2) ซีพียู Broadcom BCM2837 64-bit ARM Quad-core 900MHz
3) แรม 1 GB แชร์กับ GPU
4) Wireless LAN 802.11n
5) Lan Speed 10/100Mbps
6) USB 2.0 4 ช่อง
7) ช่องใส่ Micro SD card 1 ช่อง
8) Pin GPIO 40 ขา
9) ช่องต่อจอแสดงผล HDMI (Full size)

10) Combined 3.5mm audio jack and composite video

11) Camera interface (CSI), Display interface (DSI), , VideoCore IV 3D graphics core

2.2.2 การเชื่อมต่ออุปกรณ์

การเชื่อมต่ออุปกรณ์ต่างๆ ให้กับ Raspberry Pi ไม่ว่าจะเป็นจอ เม้าส์ คีย์บอร์ด เหมือนการเชื่อมต่อกับ PC หรือ NB ทุกประการ พอร์ตต่างๆก็เหมือนกัน

อุปกรณ์ต่อพ่วง ที่ต้องติดตั้งเข้ากับ Raspberry Pi ในการใช้งานครั้งแรก มีดังต่อไปนี้

1. Adapter 5V 2.5A ต่อทางพอร์ต mini USB เพื่อป้อนไฟฟ้าให้กับ Raspberry Pi
2. จอใช้จอทีวี โดยต่อทางพอร์ต HDMI ในภายหลังอาจไม่จำเป็นต้องใช้จอ แล้วแต่ว่าเราจะใช้ Raspberry Pi ทำอะไร
3. เมาส์และคีย์บอร์ด ต่อทางพอร์ต USB ใช้ USB Hub ช่วยในการต่อ เพื่อประหยัดพอร์ต USB
4. Micro SD Card เสียบเข้ากับบอร์ด Raspberry Pi Micro SD Card จะเป็นพื้นที่หลักที่ใช้เก็บระบบปฏิบัติการของ Raspberry Pi คล้ายกับฮาร์ดดิสก์ที่ใช้บูตเครื่อง PC หรือ NB
5. ถ้าต้องการใช้งาน USB Drive ก็ให้ต่อทางพอร์ต USB

2.2.3 ขั้นตอนการทำ Raspberry Pi ให้เป็น Web Server

การทำ Web Server บน Raspberry Pi นั้นเริ่มเป็นที่นิยมมากขึ้นเรื่อยๆ เพราะมีต้นทุนถูก และเหมาะกับงานที่ต้องการใช้งานส่วนตัว อีกทั้งยังมีตัวระบบปฏิบัติการที่มีพื้นฐานมาจาก Linux จึงทำให้การใช้งานไม่ต่างจาก Web Server ในปัจจุบันที่เริ่มหันมาใช้ Linux กันแล้ว

ดังนั้นขั้นตอนการติดตั้งจึงคล้ายคลึงกับ Linux Web Server โดยมีขั้นตอนดังนี้

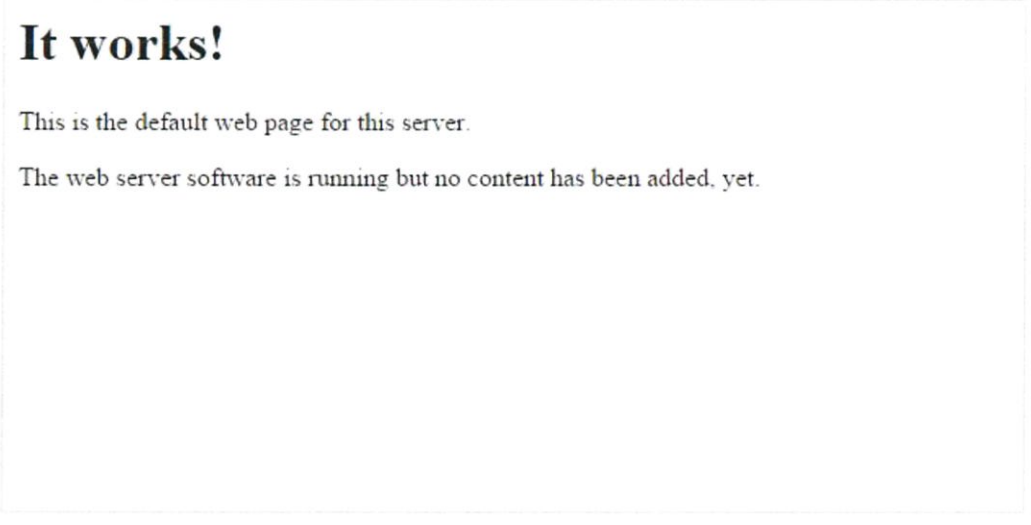
2.2.3.1 ติดตั้ง Apache

ทำการติดตั้ง Apache ด้วยคำสั่ง sudo apt-get install apache2 -y (-y คือติดตั้งแพคเกจทันทีโดยไม่ต้องถาม)

```
pi@raspberrypi ~ $ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
Suggested packages:
  apache2-doc apache2-suexec apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 113 not upgraded.
Need to get 1,355 kB of archives.
After this operation, 4,929 kB of additional disk space will be used.
```

รูปที่ 2.2.2 คำสั่งแสดงการติดตั้ง Apache บน Raspberry Pi [5]

เมื่อติดตั้งเสร็จแล้วให้ลองเปิด Web Browser แล้วเข้าไปที่ IP Address ของ Raspberry Pi เช่น 192.168.1.37 หรือถ้าเปิด Web Browser บน Raspberry Pi ก็ให้เข้าไปที่ http://localhost ก็ได้เช่นกัน จะเห็นว่ามีข้อความแสดงขึ้นต้นว่า It works! หมายความว่า Apache สามารถทำงานได้ปกติ



รูปที่ 2.2.3 แสดงการทำงานของ Apache ว่าสามารถทำงานได้ปกติ [5]

2.2.3.2 ติดตั้ง PHP

ต่อไปให้ทำการติดตั้ง PHP ด้วยคำสั่ง sudo apt-get install php5 libapache2-mod-php5 -y

```
pi@raspberrypi ~ $ sudo apt-get install php5 libapache2-mod-php5 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-mpm-prefork libonig2 libqdbm14 lsof php5-cli php5-common
Suggested packages:
  php-pear
The following packages will be REMOVED:
  apache2-mpm-worker
The following NEW packages will be installed:
  apache2-mpm-prefork libapache2-mod-php5 libonig2 libqdbm14 lsof php5
  php5-cli php5-common
0 upgraded, 8 newly installed, 1 to remove and 113 not upgraded.
```

รูปที่ 2.2.4 คำสั่งแสดงการติดตั้ง PHP บน Raspberry Pi [5]

เมื่อติดตั้งเสร็จแล้วก็ลองเข้าไปที่ Directory ที่ใช้เป็น Web Server โดยจะอยู่ที่ /var/www ดังนั้นให้พิมพ์คำสั่ง cd /var/www เพื่อไปยัง Directory ดังกล่าว

เช็คดูว่ามีไฟล์อะไรอยู่ในนี้บ้าง ด้วยคำสั่ง ls ก็จะเห็นว่ามีไฟล์ index.html อยู่ โดยไฟล์นี้นี่แหละคือข้อความ It work ที่แสดงบน Web Browser

ต่อไปจะลองเรียกคำสั่ง PHP อย่างง่ายๆ แต่ทว่าไฟล์นั้นเป็น HTML ให้ทำการเปลี่ยนนามสกุลไฟล์ให้เป็น PHP โดยใช้คำสั่ง mv index.html index.php

```
pi@raspberrypi /var/www $ sudo mv index.html index.php
```

รูปที่ 2.2.5 การเปลี่ยน นามสกุลไฟล์ให้เป็น PHP โดยใช้คำสั่ง mv index.html index.php [5]

ลองใช้คำสั่ง ls อีกครั้งเพื่อดูไฟล์ที่อยู่ใน Directory นี้ จะเห็นว่าชื่อไฟล์เปลี่ยนจาก index.html เป็น index.php แล้ว

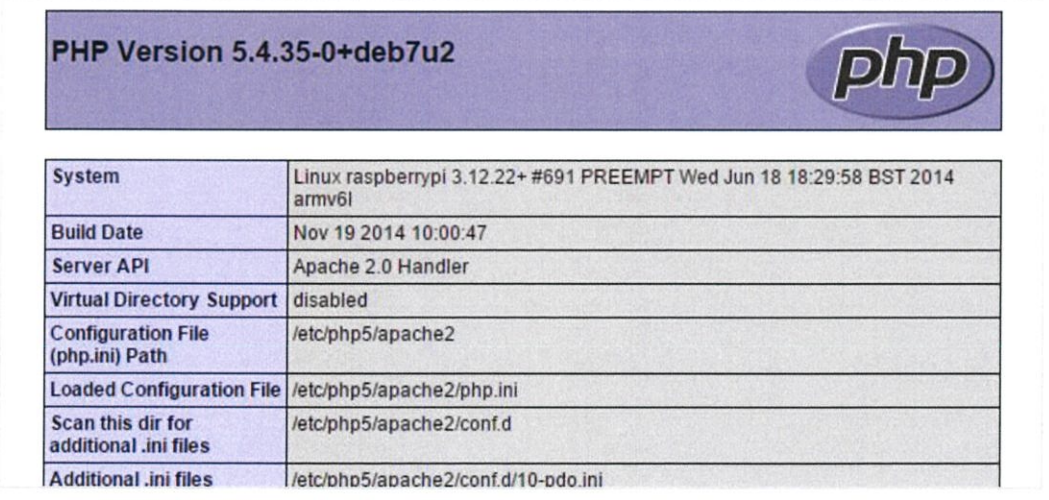
ใช้ nano เพื่อเปิดไฟล์ index.php โดยใช้คำสั่ง sudo nano index.php

```
pi@raspberrypi /var/www $ sudo nano index.php
```

รูปที่ 2.2.6 การเปิดไฟล์ index.php โดยใช้คำสั่ง sudo nano index.php [5]

ลบ Script ที่อยู่ในไฟล์นี้ให้หมด แล้วเพิ่มเข้าไปแค่ <?php phpinfo(); ?> แล้วทำการบันทึกไฟล์ (Ctrl + X เพื่อปิดไฟล์ แล้วกด Y เพื่อทำการบันทึก และ Enter เพื่อกำหนดให้บันทึกชื่อไฟล์เดิม)

เปิด Web Browser แล้วกำหนดเป็น IP Address ของ Raspberry Pi อีกครั้ง (ถ้าเปิดบน Raspberry Pi ใช้ http://localhost ได้) ก็จะเห็นว่าหน้าเว็บมีการเปลี่ยนแปลงไป มีการแสดงข้อมูลของ PHP ที่ติดตั้งอยู่ใน Raspberry Pi แทน ซึ่งมาจาก phpinfo();



PHP Version 5.4.35-0+deb7u2

System	Linux raspberrypi 3.12.22+ #691 PREEMPT Wed Jun 18 18:29:58 BST 2014 armv6l
Build Date	Nov 19 2014 10:00:47
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files	/etc/php5/apache2/conf.d/10-pdo.ini

รูปที่ 2.2.7 การเปิด Web Browser โดยกำหนดเป็น IP Address ของ Raspberry Pi [5]

2.2.3.3 ติดตั้ง MySQL

ถ้าต้องการใช้งานฐานข้อมูลด้วยก็ติดตั้ง MySQL ได้ด้วยคำสั่ง sudo apt-get install mysql-server php5-mysql -y โดยจะใช้เวลาในการติดตั้งที่ค่อนข้างนานพอสมควร

```
pi@raspberrypi /var/www $ sudo apt-get install mysql-server php5-mysql -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  heirloom-mailx libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl
  libmysqlclient16 libmysqlclient18 mysql-client-5.5 mysql-common
  mysql-server-5.5 mysql-server-core-5.5
Suggested packages:
  exim4 mail-transport-agent libipc-sharedcache-perl libterm-readkey-perl
  tinyca
Recommended packages:
  mailx
The following NEW packages will be installed:
  heirloom-mailx libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl
  libmysqlclient16 libmysqlclient18 mysql-client-5.5 mysql-common mysql-server
  mysql-server-5.5 mysql-server-core-5.5 php5-mysql
0 upgraded, 13 newly installed, 0 to remove and 113 not upgraded.
```

รูปที่ 2.2.8 การติดตั้ง MySQL โดยใช้คำสั่ง sudo apt-get install mysql-server php5-mysql -y [5]

ระหว่างการติดตั้งจะมีการกำหนดรหัสผ่านด้วย โดยให้กำหนดรหัสผ่านสำหรับ MySQL

2.2.3.4 ติดตั้ง PHPMyAdmin

ถ้าต้องการติดตั้ง PHPMyAdmin เพื่อเข้าไปจัดการฐานข้อมูลก็ให้ใช้คำสั่ง sudo apt-get install phpmyadmin –y

```
pi@raspberrypi /var/www $ sudo apt-get install phpmyadmin -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  dbconfig-common libmcrypt4 php5-gd php5-mcrypt
Suggested packages:
  libmcrypt-dev mcrypt
The following NEW packages will be installed:
  dbconfig-common libmcrypt4 php5-gd php5-mcrypt phpmyadmin
0 upgraded, 5 newly installed, 0 to remove and 113 not upgraded.
Need to get 6,112 kB of archives.
After this operation, 17.2 MB of additional disk space will be used.
```

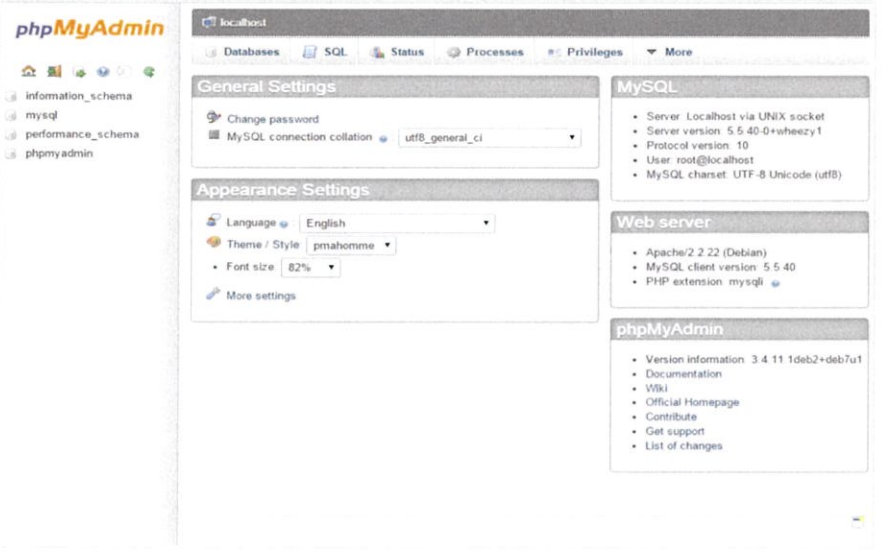
รูปที่ 2.2.9 การติดตั้ง PHPMyAdmin โดยใช้คำสั่ง sudo apt-get install phpmyadmin –y [5]

ระหว่างการติดตั้งจะมีให้กำหนดว่าใช้ Web Server แบบใดอยู่ ให้เลือก เป็น Apache2 แล้วเลือกที่ OK (ใช้ Space Bar เพื่อเลือก และใช้ Tab เพื่อสลับ Cursor ไปมาระหว่างปุ่ม OK) จะมีการตั้งค่าฐานข้อมูล ให้เลือก Yes กำหนดรหัสผ่านสำหรับ Administrator ของฐานข้อมูล กำหนดรหัสผ่านสำหรับฐานข้อมูล

เมื่อติดตั้งเสร็จแล้วให้ลองเปิด Web Browser ไปที่ PHPMyAdmin เช่น 192.168.1.37/phpmyadminหรือ http://localhost/phpmyadmin ถ้าเปิด Web Browser บน Raspberry Pi ก็จะพบกับหน้า Login ของ PHPMyAdmin

รูปที่ 2.2.10 หน้า Web Browser บน Raspberry Pi เมื่อพิมพ์ http://localhost/phpmyadmin [5]

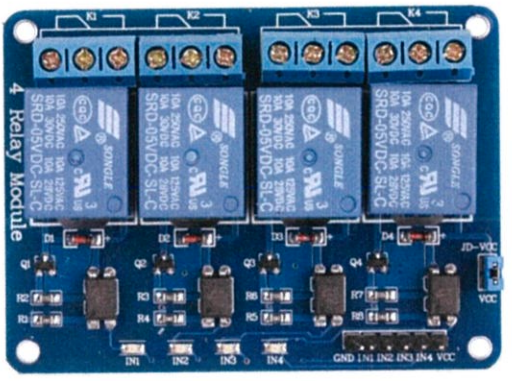
สำหรับ Username ให้ใช้ว่า root และ Password ให้ใช้ตามที่กำหนดไว้ตอนติดตั้ง PHPMyAdmin จะสามารถเข้ามาใช้งาน PHPMyAdmin ได้แล้ว



รูปที่ 2.2.11 หน้า Web Browser เมื่อเข้าสู่ PHPMyAdmin [5]

2.3 รีเลย์โมดูล

รีเลย์โมดูลเป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย



รูปที่ 2.3.1 รีเลย์โมดูล [7]

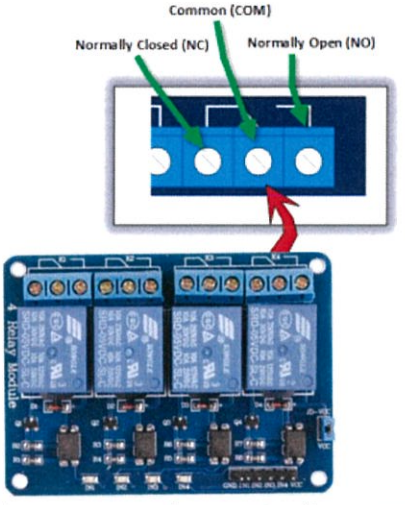
คุณสมบัติ

1. แรงดันไฟฟ้าของคอยล์ 5-48 V
2. สามารถทนอุณหภูมิได้ -40°C ถึง 85°C
3. สามารถทนการสั่นสะเทือนได้ 10 ถึง 55 Hz
4. มีอินพุต 6 ขา
    - -PIN 1 = VCC
    - -PIN 2 = IN1
    - -PIN 3 = IN2
    - -PIN 4 = IN3
    - -PIN 5 = IN4
    - -PIN 6 = GND

รูปที่ 2.3.2 ขาอินพุตของรีเลย์โมดูล [7]

5. ขาเอาต์พุตมี 3 ขา ต่อ Relay 1ตัว

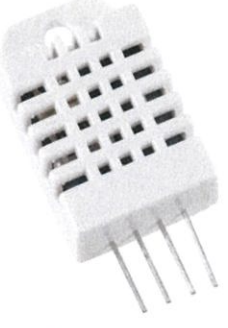
    PIN 1 = COM (Common)
    PIN 2 = NO (Normally open)
    PIN 3 = NC (Normally close)



รูปที่ 2.3.3 ขาเอาต์พุตของรีเลย์โมดูล [7]

2.4 Temperature and humidity module AM2302

โมดูลวัดอุณหภูมิ และ ความชื้น (DHT22 Temperature and humidity on module AM2302) เป็นโมดูลที่สามารถวัดอุณหภูมิ และ ความชื้นบริเวณรอบๆ หรือ สามารถนำไปประยุกต์ใช้กับงานอื่นๆได้เช่น Testing, Inspection equipment, Automatic control, Data logger, Weather station, Humidity Regulator โดยขึ้นอยู่กับการเขียนโปรแกรม และรูปแบบการต่อใช้งาน สามารถเชื่อมต่อกับบอร์ด Rasberry Pi, Arduino, ARM, MCD-51, AVR, PIC โดยมีความถูกต้องแม่นยำสูง ให้สัญญาณเอาต์พุทแบบ Digital Output โดยหากจะเชื่อมต่อเข้ากับบอร์ด Arduino จะต้องต่อตัวต้านทานขนาด 470k โอห์ม คร่อมระหว่าง Pin1 และ Pin2



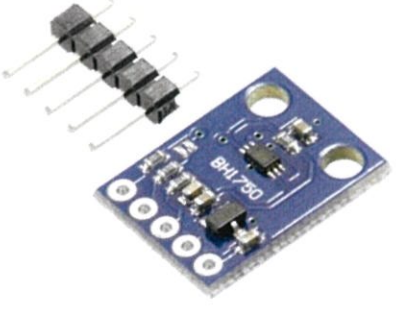
รูปที่ 2.4 DHT22 Temperature and humidity sensor on module AM2302 [8]

คุณสมบัติ

1. รองรับแรงดันไฟฟ้าตั้งแต่ 3.3V – 5.5V DC
2. ใช้กระแสสูงสุด 1.5mA
3. วัดค่าความชื้นได้ตั้งแต่ 0–100 %
4. ย่านวัดอุณหภูมิ -40 ถึง 80 องศาเซลเซียส ค่าความผิดพลาดในการวัดอุณหภูมิ 0.5%
5. ย่านวัดความชื้น 0-100% RH ค่าความผิดพลาดในการวัดความชื้น 2%
6. Connector แบบ 4 ขา โดยแต่ละขามีหน้าที่ดังนี้
    - Pin1 : VCC
    - Pin2 : SDA (Serial data, bidirectional)
    - Pin3 : N.C. (Not Connect)
    - Pin4 : GND

2.5 BH1750FVI (GY-302) Light sensor module

เซนเซอร์วัดความเข้มแสง BH1750FVI เป็นเซ็นเซอร์วัดแสง ที่ให้เอาต์พุตออกมาเป็นสัญญาณดิจิตอล โดยเอาต์พุตที่ได้มีหน่วยเป็นลักซ์ (LUX) จึงไม่จำเป็นต้องใช้การคำนวณเอาต์พุตเพิ่มเติมใดๆ ทำให้เซ็นเซอร์ BH1750FVI มีข้อดีกว่าเซ็นเซอร์วัดแสงจำพวก Photo Transistor หรือ Photodiode เนื่องจากเซนเซอร์ดังกล่าวให้เอาต์พุตเป็นการเปลี่ยนแปลงแรงดัน หรือเซ็นเซอร์ LDR ที่ให้ให้เอาต์พุตเป็นการเปลี่ยนแปลงค่าความต้านทาน



รูปที่ 2.5.1 BH1750FVI (GY-302) Light sensor module [10]

ส่วนประกอบภายในของ BH1750FVI

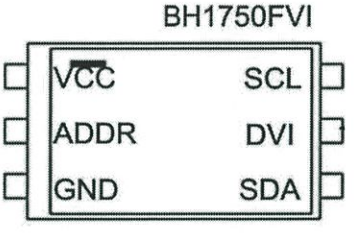
1. PD: เป็นโฟโต้ไดโอดทำหน้าที่รับแสงซึ่งถือว่าเป็นตัวแปรในรูปแบบไดนามิค

2. AMP: เป็นส่วนวงจรแปลงกระแสที่ไหลผ่านโฟโต้ไดโอดให้เป็นแรงดัน

3. ADC: เป็นส่วนวงจรแปลงแรงดันซึ่งถือว่าเป็นสัญญาณแบบอะนาล็อกให้เป็นสัญญาณทางดิจิตอลขนาด 16 บิต

4. Logic+I²C Interface: เป็นส่วนวงจรคำนวณค่าเอาต์พุตให้อยู่ในรูปแบบ ลักซ์ และเป็นส่วนที่ทำหน้าที่สื่อสารแบบ I²Cด้วย

5. OSC: เป็นส่วนวงจรกำเนิดความถี่ 320kHz เพื่อใช้ภายใน



รูปที่ 2.5.2 การใช้งานของ BH1750FVI [11]

คุณสมบัติ

1. รองรับแรงดันไฟฟ้าตั้งแต่ 2.4V – 3.6V

2. เชื่อมต่อแบบ I^2C (มีขา SCL และ SDA)

3. ใช้งานและเชื่อมต่อแบบ I^2C slave device

4. ความเร็วสำหรับบัส I^2C ได้ถึง 400kHz

5. มีขา ADDR สำหรับต่อกับ LOW หรือ HIGH เพื่อใช้กำหนดเลขที่อยู่ของ I^2C Slave (สามารถกำหนดเลขที่อยู่ได้สองค่า)

6. เลือกโหมดการวัดได้ (Measurement Mode) เช่น H-resolution Mode, H-resolution Mode2, L-resolution Mode ซึ่งเป็นตัวกำหนดความละเอียดในการวัด รวมถึงระยะเวลาในการวัดและอ่านค่า (measurement time)

7. ความละเอียด: 16 บิต ได้ค่า 1-65536 หน่วยเป็น Lux (step: 0.5 Lux, 1 Lux, หรือ 4 Lux ขึ้นอยู่กับโหมดการวัดที่เลือก)

8. ระยะเวลาในการวัดแต่ละครั้ง: ประมาณ 120 msec (สำหรับ 0.5 Lux หรือ 1 Lux), 16 msec (สำหรับ 4 Lux) ขึ้นอยู่กับโหมดการวัดที่เลือก

2.6 Thermoelectric Cooler Peltier

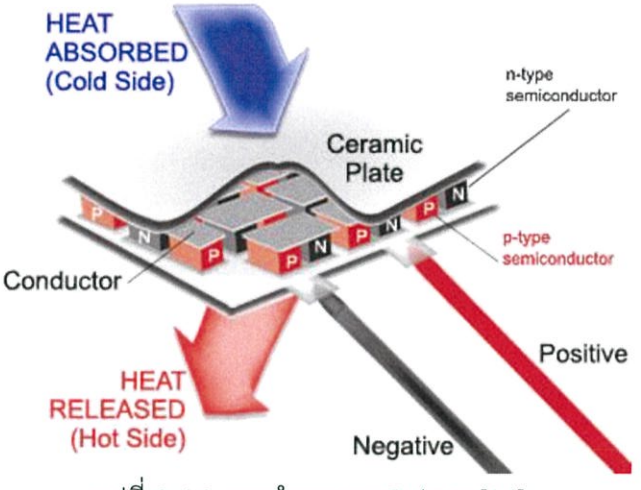
เทอร์โมอิเล็กทริค คูลเลอร์ เพลเทียร์ TEC (Thermoelectric Cooler Peltier) เป็นอุปกรณ์ที่สามารถสร้างความเย็นได้ที่ด้านหนึ่ง และปล่อยความร้อนออกมาที่อีกด้านหนึ่ง เพียงแค่จ่ายไฟฟ้ากระแสตรงเข้าไปในตัวอุปกรณ์เท่านั้น ซึ่งการทำความเย็นในรูปแบบนี้ไม่ต้องใช้คอมเพรสเซอร์ หรือส่วนที่เป็นกลไกขับเคลื่อนใด ๆ และไม่ต้องใช้สารทำความเย็นเป็นตัวกลางในระบบเหมือนคอมเพรสเซอร์

รูปที่ 2.6.1 Cooler Peltier [14]

2.6.1 หลักการทำงานของ Peltier

หลักการทำงานของแผ่นทำความเย็นเพลเทียร์นั้น เป็นหลักการที่มีชื่อว่า เทอร์โมอิเล็กทริก (Thermoelectric) หลักการทำความเย็นแบบนี้เกิดขึ้นได้ โดยการใช้สารกึ่งตัวนำแบบ พี-เอ็น (P-N Type) ซึ่งสารกึ่งตัวนำแบบพี-เอ็น คือส่วนประกอบหลักของแผ่นทำความเย็นเพลเทียร์ โดยการทำความเย็นจะเกิดขึ้นได้ก็ต่อเมื่อ มีการจ่ายไฟฟ้ากระแสตรง (Direct Current : DC) หรือไฟดีซี ให้กับแผ่นทำความเย็นเพลเทียร์ เพราะเมื่อกระแสไฟฟ้าเดินทางผ่านวัสดุที่มีคุณสมบัติเป็นสารกึ่งตัวนำแล้วก็จะเกิดการทำปฏิกิริยาขึ้น สารกึ่งตัวนำ แบบพี-เอ็น ซึ่งต่างชนิดกัน เมื่อมีกระแสไฟฟ้าไหลผ่าน ก็จะมีการดูดกลืนกันของอิเล็กตรอนที่เคลื่อนจากระดับพลังงานต่ำทางด้าน

สารกึ่งตัวนำแบบพี ไปสู่ระดับพลังงานที่สูงกว่าทางด้านสารกึ่งตัวนำแบบเอ็น กระบวนการดังกล่าวส่งผลให้ที่ผิวด้านหนึ่งของแผ่นเพลเทียร์มีการดูดพลังงานความร้อน ซึ่งก็ได้จากความร้อนที่อยู่โดยรอบนั่นเอง เมื่อความร้อนในบริเวณรอบ ๆ ถูกดูดเข้ามา ก็จะทำให้ในบริเวณนั้นมีอุณหภูมิต่ำลง ซึ่งด้านนี้ก็คือด้านทำความเย็นนั่นเอง และในขณะเดียวกัน ก็จะเกิดการดูดกลืนของอิเล็กตรอนจากระดับพลังงานที่สูง ในสารกึ่งตัวนำแบบเอ็น สู่ระดับพลังงานที่ต่ำกว่า ในสารกึ่งตัวนำแบบพี ส่งผลให้เกิดการคายความร้อนออกมาที่บริเวณผิวหน้าของอีกด้านหนึ่ง จากหลักการทำงานที่ได้อธิบายไปข้างต้นนั้น ทำให้สามารถนำแผ่นเพลเทียร์มาประยุกต์ใช้งานได้กับหลายๆสิ่ง ที่โดดเด่นสุดคงหนีไม่พ้น การนำคุณสมบัติในด้านการทำความเย็นมาประยุกต์ใช้งาน เพื่อเป็นอีกหนึ่งทางเลือกสำหรับการทำความเย็นหรือลดอุณหภูมิ แบบไม่ต้องพึ่งพาระบบทำความเย็นที่ใช้คอมเพรสเซอร์



รูปที่ 2.6.2 การทำงานของPeltier [14]

จากรูปที่ 2.6.2 จะเห็นได้ว่าเมื่อเพลเทียร์ได้รับกระแสไฟจากแหล่งจ่าย ผ่านสารกึ่งตัวนำแบบพี-เอ็น อิเล็กตรอนจะวิ่งจากสารกึ่งตัวนำด้านพี ไปสู่สารกึ่งตัวนำด้านเอ็น จึงส่งผลให้ เกิดความเย็นบนแผ่นเซลามิคด้านหนึ่ง และเกิดความร้อนบนแผ่นเซลามิคอีกด้านหนึ่ง

2.6.2 การนำ Peltier มาใช้งาน

สำหรับแผ่นทำความเย็นเพลเทียร์ การใช้งานโดยมีวัตถุประสงค์เพื่อการทำความเย็นในบ้านเราก็มีการนำไปใช้งานในหลายรูปแบบ ยกตัวอย่างเช่น การใช้เพื่อทำความเย็นให้กับอุปกรณ์อิเล็กทรอนิกส์แบบเฉพาะจุด รวมไปถึงการให้ความเย็นในลักษณะของตู้เย็นขนาดเล็ก ซึ่งในบ้านเราตอนนี้ก็ได้มีผู้ผลิตบางราย นำแผ่นทำความเย็นเพลเทียร์ มาใช้งานกับตู้แช่เย็นขนาดเล็ก ๆ สำหรับแช่เครื่องดื่ม ตลอดจนตู้เย็นขนาดเล็ก(มินิบาร์) ความจุประมาณ 1.6 - 1.8 ลิตร ซึ่งเป็นการนำมาใช้งานทดแทนคอมเพรสเซอร์ แต่ในตอนนี้ ประสิทธิภาพการทำความเย็นของแผ่นทำความเย็นเพลเทียร์ ก็ยังเป็นรองระบบทำความเย็นที่ใช้คอมเพรสเซอร์อยู่อีกหลายเท่า ตู้เย็นที่ใช้แผ่นทำความเย็น

เพลเทียร์จึงเหมาะกับการใช้งานบางประเภทเท่านั้น ซึ่งอาจจะไม่เหมาะกับการแช่อาหารเพื่อคงความสด แต่ก็เหมาะสำหรับการใช้เก็บรักษายาหรือเวชภัณฑ์ทางการแพทย์

2.7 Ultrasonic Mist Maker

เครื่องทำหมอกอัลตราโซนิคใช้น้ำเพื่อสร้างหมอก เครื่องผลิตหมอกทำงานบนหลักการของการสั่นที่ทำให้น้ำกลายเป็นหยดเล็ก ๆ ที่ระเหยกลายเป็นเมฆที่มีหมอก ผลที่ตามมาเมื่อการสั่นสะเทือนในเชิงบวกคลื่นความดันสูงจะบังคับให้น้ำที่มีโพรงอากาศผ่านพื้นผิวของน้ำเป็นหมอกที่ละเอียดมากซึ่งจะถูกดูดซึมเข้าสู่อากาศได้ง่าย

รูปที่ 2.7 Ultrasonic Mist Maker [15]

ด้วยความช่วยเหลือของแผ่นดิสก์เซรามิคภายในเครื่องผลิตหมอกอัลตราโซนิคแช่อยู่ในน้ำแปลงสัญญาณอิเล็กทรอนิกส์ความถี่สูงซึ่งโดยทั่วไปมีความถี่สะท้อนประมาณ 1.6MHz เป็นความถี่สูงสั่นสะเทือนเชิงกล คลื่นอัลตราโซนิคเน้นไปที่น้ำน้ำไม่สามารถทำตามการเคลื่อนไหวของตัวแปลงสัญญาณความถี่สูง เพื่อให้เกิดสุญญากาศชั่วขณะ ถูกสร้างขึ้นบนการสั่นเชิงลบของตัวแปลงซึ่งน้ำ cavitates เข้าไปในหมอกละอองนี้ประกอบด้วยอนุภาคน้ำขนาด มีขนาดไม่ถึง 5 ไมครอน สามารถเจาะทะลุช่องว่างเล็กๆ

2.8 ภาษา Python

ภาษาไพทอนเป็นภาษาคอมพิวเตอร์ระดับสูง และเป็นซอฟต์แวร์รหัสเปิด (open source software) ซึ่งเริ่มมีการนำมาใช้งานในปี พ.ศ. 2532 และพัฒนาโดย นายกุยโด แวน โรสซัม (Guido Van Rossum) ซึ่งเป็น นักเขียนโปรแกรมชาวเนเธอร์แลนด์ ปัจจุบันนายโรสซัม ทำงานที่

บริษัทดรอปบล๊อก (Dropbox) และเป็นผู้พัฒนา ที่เก็บข้อมูลบนคลาว์ด ภาษาไพทอนได้รับการพัฒนา มาจากภาษามอดูลา-3 (Modula-3) และภาษาเอบีซี (ABC) โดยใช้ชื่อตามรายการโทรทัศน์ชื่อ Monty Python ปัจจุบันภาษาไพทอนได้พัฒนาถึงรุ่น 3.5

ภาษาไพทอนสามารถใช้งานได้บนระบบ ปฏิบัติการต่างๆ เช่น วินโดวส์ ลินุกซ์ และแมค เพราะรูปแบบ คำสั่งในการเขียนโปรแกรมด้วยภาษาไพทอนไม่ซับซ้อน ทำให้เขียนง่าย และส่งผลให้ผู้เขียนเข้าใจโปรแกรมได้ง่ายด้วย จึงเหมาะสำหรับผู้เริ่มต้นเขียนโปรแกรม และผู้มีความเชี่ยวชาญที่ต้องการพัฒนาโปรแกรมขนาดใหญ่ ที่สามารถนำไปประยุกต์ใช้งานในชีวิตประจำวัน เนื่องจากมีไลบรารีหรือส่วนของโปรแกรมที่มีผู้พัฒนา ไว้จำนวนมาก ทำให้ไม่ต้องเขียนโปรแกรมตั้งแต่เริ่มต้น อีกทั้งสามารถนำมาต่อยอดสร้างงานอื่นได้และใช้งาน ร่วมกับโปรแกรมที่เขียนขึ้นใหม่ได้อย่างสะดวกและรวดเร็ว ภาษาไพทอนสามารถรองรับงานวิทยาศาสตร์และ คณิตศาสตร์ได้เป็นอย่างดี

2.9 ภาษา PHP

พีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และภาษาเพิร์ล ซึ่งภาษาพีเอชพีนั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีความตอบโต้ได้อย่างรวดเร็ว

2.9.1 ความสามารถของภาษา PHP

1) เป็นภาษาที่มีลักษณะเป็นแบบ Open source ผู้ใช้สามารถ Download และนำ Source code ของ PHP ไปใช้ได้โดยไม่เสียค่าใช้จ่าย

2) เป็นสคริปต์แบบ Server Side Script ดังนั้นจึงทำงานบนเว็บเซิร์ฟเวอร์ ไม่ส่งผลกับการทำงานของเครื่อง Client โดย PHP จะอ่านโค้ด และทำงานที่เซิร์ฟเวอร์ จากนั้นจึงส่งผลลัพธ์ที่ได้จากการประมวลผลมาที่เครื่องของผู้ใช้ในรูปแบบของ HTML ซึ่งโค้ดของ PHP นี้ผู้ใช้จะไม่สามารถมองเห็นได้

3) PHP สามารถทำงานได้ในระบบปฏิบัติการที่ต่างชนิดกัน เช่น Unix, Windows, Mac OS หรือ Risc OS อย่างมีประสิทธิภาพ เนื่องจาก PHP เป็นสคริปต์ที่ต้องทำงานบนเซิร์ฟเวอร์ ดังนั้นคอมพิวเตอร์สำหรับเรียกใช้คำสั่ง PHP จึงจำเป็นต้องติดตั้งโปรแกรมเว็บเซิร์ฟเวอร์ไว้ด้วย เพื่อให้สามารถประมวลผล PHP ได้

4) PHP สามารถทำงานได้ในเว็บเซิร์ฟเวอร์หลายชนิด เช่น Personal Web Server(PWS), Apache, OmniHttpd และ Internet Information Service(IIS) เป็นต้น

5) ภาษา PHP สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)

6) PHP มีความสามารถในการทำงานร่วมกับระบบจัดการฐานข้อมูลที่หลากหลาย ซึ่งระบบจัดการฐานข้อมูลที่สนับสนุนการทำงานของ PHP เช่น Oracle, MySQL, FilePro, Solid, FrontBase, mSQL และ MS SQL เป็นต้น

7) PHP อนุญาตให้ผู้ใช้สร้างเว็บไซต์ซึ่งทำงานผ่านโปรโตคอลชนิดต่างๆ ได้ เช่น LDAP, IMAP, SNMP, POP3 และ HTTP เป็นต้น

8) โค้ด PHP สามารถเขียน และอ่านในรูปแบบของ XML ได้

2.10 ภาษา SQL

SQL คือ Structured Query Language ซึ่งมันคือ ภาษา programming ที่ออกแบบมาเพื่อทำการจัดการข้อมูลที่อยู่ใน relational database management system (RDBMS) หรือก็คือไว้สำหรับค้นหาข้อมูล เปลี่ยนแปลง เพิ่ม และ ลด ข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลในรูปแบบตารางที่มีลักษณะเป็น column และ row เราเรียกข้อมูลเหล่านี้ว่าถูกเก็บอยู่ใน table ด้วยความสามารถของ SQL เรายังสามารถสร้างตารางขึ้นมาใหม่ (create) รวมถึง ลบ (drop) และเปลี่ยนแปลงค่า (alter) ของ table ได้ ซึ่งในความเป็นจริงแล้วคำสั่ง SQL ประกอบไปด้วย

1. Data definition language – DDL ใช้ในการสร้างฐานข้อมูล
2. Data manipulation language – DML ใช้ เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลใน table
3. Data control language – DCL ใช้ในการกำหนดสิทธิการอนุญาต

ทั้งหมดนี้อยู่ในขอบเขตของการทำ insert, query, update, delete, schema creation and modification และ data access control

2.10.1 SQL Syntax

ภาษา SQL ถูกแบ่งออกมาเป็นส่วนประกอบต่างๆ เราเรียกส่วนเหล่านี้ตามรูปแบบ เช่น

Clauses คือ องค์ประกอบหนึ่งของ statement และ query (ส่วนนี้เป็น Optional)

Expressions คือ การสร้างผลลัพธ์ออกมาในรูป table ที่ประกอบด้วย column และ row จากข้อมูล

Predicates คือ รูปแบบเงื่อนไขที่มีผลลัพธ์เป็น true/false/unknown หรือก็คือ Boolean

Queries คือการดึงข้อมูลตามเงื่อนไข (clause) เป็นส่วนสำคัญใน SQL

Statements คือ สามารถมีผลต่อโครงสร้างข้อมูล, จัดการข้อมูล trasactions, program flow, session หรือแม้กระทั้ง วิเคราะห์ปัญหา โดยจำเป็นต้องจบด้วย semicolon (;) ซึ่งเป็นสิ่งจำเป็นที่ต้องมีทุกครั้งสำหรับ SQL

Insignificant whitespace หรือช่องว่าง สำหรับใน SQL statement และ query จะไม่สนใจ ทำให้ SQL สามารถเขียนในรูปแบบที่หลากหลาย

ประเภทของ SQL query

4. Select query ใช้ดึงข้อมูลในรูป table คือเป็น row และ column

```
SELECT column1, column2, ...
FROM table_name;
```

2. Update query ใช้แก้ไขข้อมูลที่มีอยู่แล้วใน table

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

5. Insert query ใช้เพิ่มข้อมูลใน table

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

6. Delete query ใช้ลบข้อมูลใน table

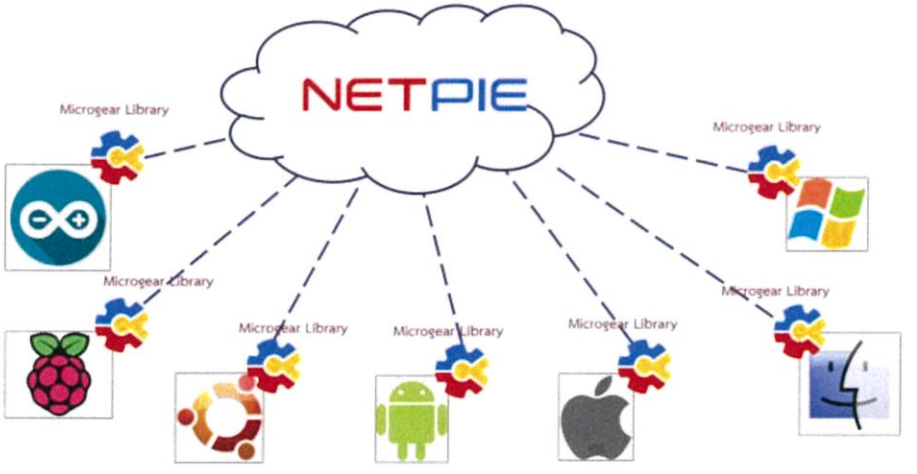
```
DELETE FROM table_name
WHERE condition;
```

2.11 ภาษา HTML

HTML ย่อมาจาก Hypertext Markup Language เป็นภาษาคอมพิวเตอร์รูปแบบหนึ่ง ที่มีโครงสร้างการเขียนโดยอาศัยตัวกำกับ (Tag) ควบคุมการแสดงผลข้อความ รูปภาพ หรือวัตถุ อื่นๆ ผ่านโปรแกรมเบราเซอร์ แต่ละ Tag อาจจะมีส่วนขยาย เรียกว่า Attribute สำหรับระบุหรือควบคุมการแสดงผล ของเว็บได้ด้วย HTML เป็นภาษาที่ถูกพัฒนาโดย World Wide Web Consortium (W3C) จากแม่แบบของภาษา SGML (Standard Generalized Markup Language)โดยตัดความสามารถบางส่วนออกไป เพื่อให้สามารถทำความเข้าใจและเรียนรู้ได้ง่าย และด้วยประเด็นดังกล่าว ทำให้บริการ www เติบโตขยายตัวอย่างกว้างขวาง

2.12 NETPIE

NETPIE เป็น IoT (Internet of Things) Cloud Platform ที่พัฒนาขึ้นโดยทีมงานวิจัยและเปิดให้ บุคคลทั่วไปใช้งานโดยมี Web Portal ที่ให้สามารถลงทะเบียนและจัดการตัวตนและสิทธิ์ของแอปพลิเคชั่น และอุปกรณ์ได้ที่เว็บไซต์https://netpie.io ตั้งแต่เดือนกันยายน 2558 เป็นต้นมา NETPIE เป็น Middleware ที่มีหัวใจหลัก(นอกเหนือจากส่วนอื่นๆ) เป็น Distributed MQTT Brokers ซึ่งเป็นเสมือนจุด นัดพบให้สิ่งต่างๆ(Things) มาติดต่อสื่อสารและทำงานร่วมกันผ่านวิธีการส่งข้อความแบบ Publish/Subscribe NETPIE มีโครงสร้างสถาปัตยกรรมเป็นคลาวด์อย่างแท้จริงในทุกองค์ประกอบ ทำให้ สามารถขยายตัวได้อย่างอัตโนมัติ (Auto-scale) สามารถดูแลและซ่อมแซมตัวเองได้อัตโนมัติเมื่อส่วนหนึ่ง ส่วนใดในระบบมีปัญหา (Self-healing, Self-recovery) โดยไม่ต้องพึ่งผู้ดูแลระบบ การบริหารจัดการระบบ เป็นแบบ Plug-and-Play ไม่ต้อง Configure หรือปรับแต่ง ในฝั่งอุปกรณ์ NETPIE มี Client Library หรือที่ เรียกว่า Microgear ซึ่งทำหน้าที่สร้างและดูแลช่องทางสื่อสารระหว่างอุปกรณ์กับ NETPIE รวมไปถึงรักษา ความปลอดภัยในการส่งข้อมูล Microgear เป็น Open Source และสามารถดาวน์โหลดได้จาก https://github.com/netpieio โดย ณ ปัจจุบันมี Microgear สำหรับ OS และ Embedded Board หลักๆ ที่ เป็นที่นิยมในหมู่นักพัฒนาเกือบทุกชนิด โมเดลการสื่อสารของ NETPIE



รูปที่ 2.12.1 วิธีการสื่อสารของสิ่งต่างๆ ผ่าน NETPIE [19]

ประโยชน์ของ NETPIE

1. ช่วยลดการใช้ทรัพยากรของการเชื่อมต่อ NETPIE ช่วยให้อุปกรณ์สามารถสื่อสารกันได้โดยผู้ใช้ไม่ต้องกังวลว่า อุปกรณ์นั้นจะอยู่ที่ใด เพียงแค่นำ Microgear Library ไปติดตั้งในอุปกรณ์ NETPIE จะรับหน้าที่ดูแลเชื่อมต่อให้ทั้งหมด ไม่ว่าอุปกรณ์นั้น จะอยู่ในเครือข่ายชนิดใด

ลักษณะใด หรือแม้กระทั่งเคลื่อนย้ายไปอยู่ที่ใด ผู้ใช้สามารถตัดปัญหาในการ เข้าถึงอุปกรณ์จากระยะไกล (Remote Access) ด้วยวิธีการแบบเดิมๆ เช่น การใช้ Fixed Public IP Address หรือการตั้ง Port Forwarding ในเราท์เตอร์และการต้องไปลงทะเบียนกับผู้ให้บริการ Dynamic DNS ซึ่งทั้งหมดล้วนมีความยุ่งยาก ลดความยืดหยุ่นของระบบ ไม่เพียงเท่านั้น NETPIE ยังช่วยให้การเริ่มต้นใช้งานเป็นไปโดยง่าย โดยออกแบบให้อุปกรณ์ถูกค้นพบและเข้าสู่บริการโดยอัตโนมัติ (Automatic Discovery, Plug-and-Play)

2. ช่วยลดภาระด้านความปลอดภัยของข้อมูล NETPIE ถูกออกแบบให้มีระดับและสิทธิ์ในการเข้าถึงในระดับ Fine Grain กล่าวคือผู้ใช้สามารถ ออกแบบได้เองทั้งหมดว่า สิ่งใดมีสิทธิ์คุยกับสิ่งใด สิ่งใดมีสิทธิ์หรือไม่เพียงใดในการอ่านหรือเขียนข้อมูล และสิทธิ์เหล่านี้จะมีอายุการใช้งานนานเท่าใด หรือจะถูกเพิกถอนภายใต้เงื่อนไขใด เป็นต้น

3. ยืดหยุ่นต่อการขยายระบบ NETPIE มีสถาปัตยกรรมเป็นคลาวด์เซิร์ฟเวอร์อย่างแท้จริงในทุกองค์ประกอบของระบบ ทำให้เกิดความยืดหยุ่นและคล่องตัวสูงในการขยายตัว นอกจากนี้โมดูลต่าง ๆ ยังถูกออกแบบให้ทำงานแยกจากกัน เพื่อให้เกิดสภาวะ Loose Coupling และสื่อสารกันด้วยวิธี Asynchronous Messaging ช่วยให้แพลตฟอร์มมีความน่าเชื่อถือได้สูง นำไปใช้ซ้ำและพัฒนาต่อได้ง่าย ดังนั้นผู้พัฒนาไม่จำเป็นต้องกังวลกับการขยายตัวเพื่อรับโหลดที่เพิ่มขึ้นในระบบอีกต่อไป

2.12.1 Microgear

Microgear คือซอฟต์แวร์ไลบรารี่ของ NETPIE ที่ติดตั้งอยู่บนอุปกรณ์ที่ต้องการเชื่อมต่อสื่อสารผ่านคลาวด์ของ NETPIE Microgear เปรียบเสมือนตัวกลางและผู้ช่วยในการสร้างและดูแลการเชื่อมต่อให้มี ความเสถียร ปลอดภัย ให้การสื่อสารแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์เป็นไปอย่างราบรื่น บทบาทหน้าที่ของ Microgear สามารถแบ่งออกเป็น 4 ด้านคือ

1. ด้านการสื่อสาร (Communication) Microgear จะเป็นผู้ช่วยในการสร้างการเชื่อมต่อ (Connection) ไปยังคลาวด์ของ NETPIE และคอยตรวจสอบสถานะของการเชื่อมต่อ หากการเชื่อมต่อมีปัญหา Microgear สามารถช่วยเชื่อมต่อให้ใหม่เพื่อให้การสื่อสารเป็นไปได้อย่างราบรื่น นอกจากนี้ Microgear ยังช่วยอำนวยความสะดวก ในการสร้างช่องทางการสื่อสารแบบเข้ารหัสในกรณีที่ผู้ใช้ต้องการส่วนการแลกเปลี่ยนข้อมูลระหว่าง Microgear และคลาวด์ของ NETPIE จะใช้โปรโตคอล MQTT ในการสื่อสาร

2. ด้านการยืนยันตัวตน (Authentication) ในขั้นตอนการสร้างการเชื่อมต่อ Microgear จะช่วยยืนยันตัวตนของอุปกรณ์กับคลาวด์ของ NETPIE โดยการพิสูจน์ตัวตน (Identity) ของอุปกรณ์จะ ใช้ข้อมูลประกอบกันสามส่วนคือ App ID, App Key และ Token

3. ด้านการขออนุญาตสิทธิ์ (Authorization) การขออนุญาตสิทธิ์ในการสื่อสารจะเกิดขึ้นใน ขั้นตอนการสร้างการเชื่อมต่อควบคู่กับการยืนยันตัวตน คลาวด์ของ NETPIE จะเป็นผู้ออกใบอนุญาต (Token) ที่ระบุว่าอุปกรณ์ตัวนี้ สามารถสื่อสารได้กับอุปกรณ์ตัวใดบ้าง ในกรณีปกติ

อุปกรณ์ที่อยู่ภายใต้กลุ่ม App ID เดียวกันเท่านั้น จึงจะมีสิทธิ์สื่อสารกันได้ (ยกเว้นในกรณีการใช้ Freeboard Microgear ที่อนุญาตให้สื่อสารข้าม App ID

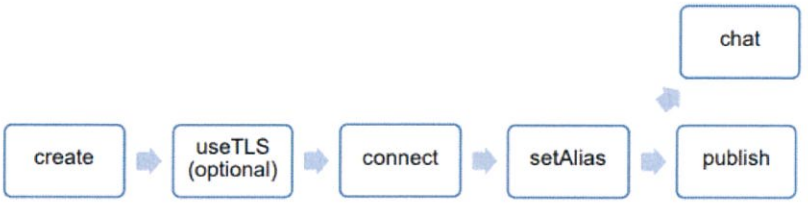
4. ด้านการประสานงาน (Coordination) Microgear มีฟังก์ชั่นที่ช่วยให้อุปกรณ์ต่างๆ ภายในกลุ่ม App ID เดียวกันทราบสถานะของกันและกันเช่น ทราบว่ามีอุปกรณ์ใดออนไลน์เข้ามาใหม่ในกลุ่ม หรือมีอุปกรณ์ใดออกไปจากกลุ่ม รวมถึงทราบการเปลี่ยนแปลงสถานะของอุปกรณ์ที่สนใจติดตาม จากข้อมูลดังกล่าวผู้ใช้สามารถกำหนดบทบาทหน้าที่ให้อุปกรณ์ในกลุ่มตามสถานะของอุปกรณ์อื่น ๆ ในกลุ่ม เช่น หากเป็นอุปกรณ์ตัวแรกในกลุ่มให้ทำหน้าที่เป็นหัวหน้ากลุ่ม เป็นต้น Microgear ถูกพัฒนาขึ้นเพื่อให้ทำงานได้กับอุปกรณ์ที่หลากหลาย ในส่วนของซอฟต์แวร์มี Microgear ให้เลือกใช้กับ Programming Language ได้แก่ Node.js, Python, HTML5, Java Android และ C# สำหรับอุปกรณ์ฮาร์ดแวร์ประเภทไมโครคอนโทรลเลอร์ Microgear เปรียบเสมือน Firmware ซึ่งมี Microgear ที่รองรับ Arduino with Ethernet Shield (ใช้ได้กับ Arduino Mega) Raspberry Pi และ Microgear สำหรับ WiFi ไมโครคอนโทรลเลอร์ ESP8266 Microgear ที่พัฒนาขึ้นทั้งหมดถูกรวบรวมไว้ที่ https://github.com/netpieio โดยมีสัญญาอนุญาตให้ใช้สิทธิ์แบบเปิดประเภท ISC License ซึ่งอนุญาตให้ทำซ้ำ ดัดแปลง หรือส่งต่อไลบรารี่นี้ได้ ทั้งในการใช้งานเชิงสาธารณประโยชน์ และเชิงพาณิชย์ รายละเอียดของสัญญาอนุญาตให้ใช้สิทธิ์แบบ ISC License มีดังนี้

2.12.2 ฟังก์ชั่นหลักของ Microgear

Microgear แต่ละชนิดอาจมีชื่อและชนิดของฟังก์ชั่นแตกต่างกันตามลักษณะของการเขียนโปรแกรมในภาษานั้นๆ ในที่นี้ขอยกตัวอย่างฟังก์ชั่นที่มีเหมือนกันอยู่ในหลาย Microgear โดยขออ้างอิงชื่อฟังก์ชั่นจาก HTML5 Microgear

- create สร้าง Microgear เพื่อเริ่มต้นใช้งาน
- connect เชื่อมต่อ Microgear เข้ากับคลาวด์ของ NETPIE
- setAlias กำหนดชื่อเล่นของอุปกรณ์เพื่อใช้ระบุตัวตนของอุปกรณ์ภายใน NETPIE
- chat ส่งข้อความแบบเจาะจงผู้รับ
- publish ส่งข้อความแบบไม่เจาะจงผู้รับไปยังหัวข้อสนทนาที่กำหนด
- subscribe ระบุความสนใจในหัวข้อสนทนา บอกรับข้อความที่เกิดขึ้นบนหัวข้อนั้นๆ
- unsubscribe ยกเลิกการบอกรับข้อความในหัวข้อสนทนาที่เคยsubscribe ไว้
- resetToken ยกเลิกใบอนุญาต (Token) และลบใบอนุญาตออกจากcache บนอุปกรณ์
- useTLS ระบุว่าต้องการสร้างการเชื่อมต่อแบบเข้ารหัสระหว่าง Microgear กับคลาวด์ของ NETPIE

- on ตอบสนองต่อเหตุการณ์ที่สนใจผ่านการเรียก Callback Function ลำดับในการเรียกฟังก์ชั่นพื้นฐานเพื่อเริ่มส่งข้อมูลเป็นไปตามแผนภาพในรูปที่ 1.2



รูปที่ 2.12.2 ลำดับการเรียกฟังก์ชั่นพื้นฐานของ Microgear [19]

Events ของ Microgear การทำงานของ Microgear เป็นแบบ Event-driven จึงต้องตอบสนองต่อเหตุการณ์ต่างๆ ด้วยการเขียน Callback Function ซึ่งชนิดของเหตุการณ์ที่สามารถเกิดขึ้น มีดังนี้

- connected เกิดขึ้นเมื่อ Microgear เชื่อมต่อกับ NETPIE สำเร็จ
- closed เกิดขึ้นเมื่อ Microgear ปิดการเชื่อมต่อกับ NETPIE
- error เกิดขึ้นเมื่อมีความผิดพลาดเกิดขึ้นกับ Microgear
- message เกิดขึ้นเมื่อมีข้อความเข้ามาที่อุปกรณ์
- present เกิดขึ้นเมื่อมีอุปกรณ์ใน App ID เดียวกันเชื่อมต่อเข้ามาบน NETPIE
- absent เกิดขึ้นเมื่อมีอุปกรณ์ใน App ID เดียวกันหายไปจากการเชื่อมต่อกับ NETPIE

2.12.3 FREEBOARD

Freeboard เป็น Web Application ที่สามารถสร้าง Dashboard เพื่อแสดงผลสำหรับ IoT แอปพลิเคชั่นโดยสามารถใช้เป็นกระดานส่วนตัว สามารถวางปุ่มกดสวิตช์ไว้ใช้สำหรับควบคุมอุปกรณ์ หรือวางหน้าปัดเพื่อแสดงผลข้อมูลต่าง ๆ ที่ได้จากอุปกรณ์ เช่น เซนเซอร์ในระบบ IoT นอกจากนี้ยังสามารถ แสดงผลเป็นกราฟได้ ส่วนหน้ากระดานหรือ Dashboard นั้นสามารถปรับแต่งได้โดยง่ายเพียงแค่ป้อนข้อมูลเข้า หรือกำหนดคำสั่งก็สามารถทำงานได้แล้ว โดยที่ผู้ใช้ไม่จำเป็นต้องเขียน HTML Web Page และที่สำคัญคือข้อมูลนั้นมีการอัพเดทแบบ Real-time มีความเสถียรและเชื่อถือได้ และเป็น Open-Source

NETPIE Freeboard คือ Freeboard สำหรับการควบคุมและการแสดงผล (Visualization) ข้อมูลที่ดึงมาจากอุปกรณ์ที่ต่อกับ NETPIE ทีมงานได้พัฒนา Widget Plugins

ขึ้นมาให้ผู้ใช้งานสามารถทำตามความต้องการได้หลากหลาย เช่น สามารถสร้างปุ่มควบคุมและใส่คำสั่ง Javascript สำหรับ Action ต่าง ๆ ได้การใช้งาน NETPIE Freeboard นั้นสามารถใช้ Browser เปิดไฟล์ index.html ที่ได้จากการติดตั้ง NETPIE Freeboard หรือสามารถใช้ Freeboard ผ่านทางหน้าเว็บของ NETPIE

2.12.3 Dashboard ของ NETPIE Freeboard [19]

บทที่ 3

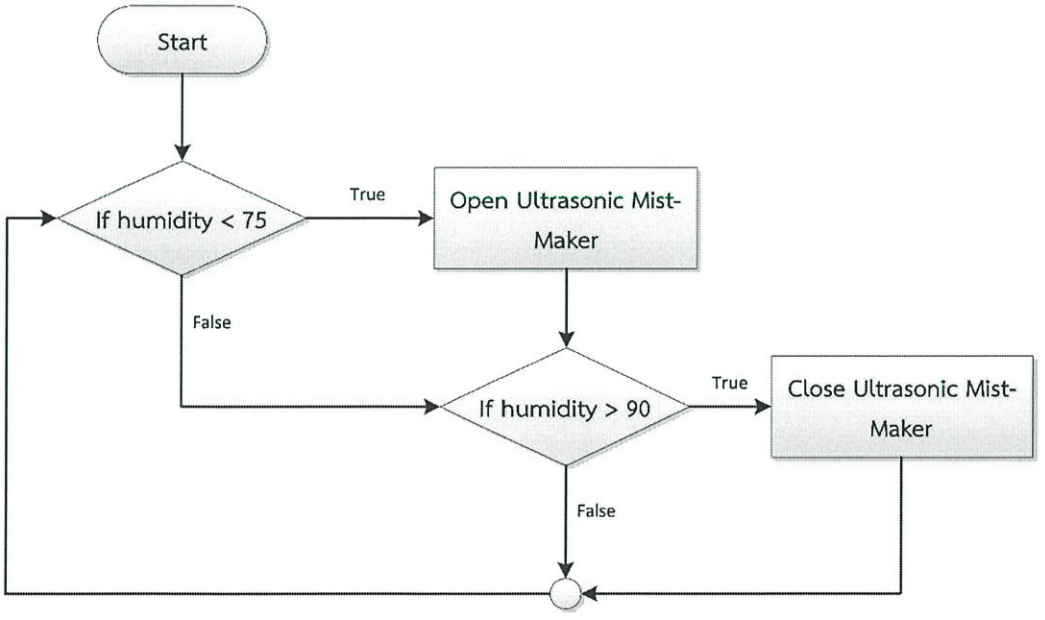
การออกแบบและการจัดทำปริญญานิพนธ์

ในส่วนของการออกแบบและจัดทำปริญญานิพนธ์นั้น จะนำอุปกรณ์และทฤษฎีจากบทที่ 2 มาใช้เป็นความรู้พื้นฐาน โดยเนื้อหาของบทนี้จะกล่าวถึง ภาพรวมของการออกแบบ ซึ่งมีทั้งในส่วนของฮาร์ดแวร์ และซอฟท์แวร์ ของระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง รวมทั้งการทดสอบ ประสิทธิภาพของระบบ เช่นความสามารถในการทำความเย็น และความสามารถในการอ่านค่าของเซนเซอร์ โดยจะอธิบายรายละเอียดวิธีการต่าง ๆ ที่ใช้ในการสร้างในแต่ละส่วน

3.1 การออกแบบ

3.1.1 ออกแบบลำดับการทำงานของโปรแกรมในส่วนของการควบคุมความชื้น

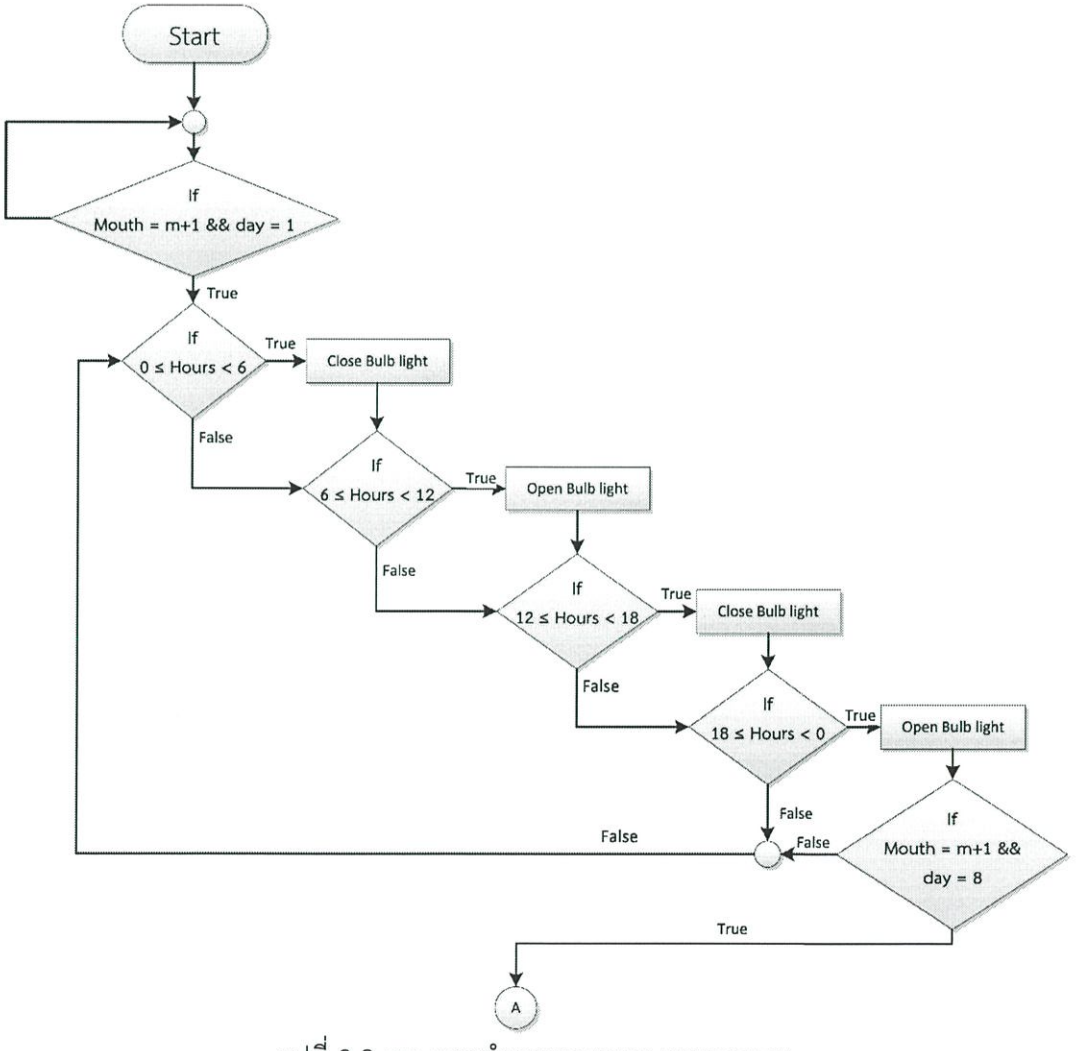
การออกแบบระบบการทำงานของ Raspberry Pi ที่เชื่อมต่อกับรีเลย์โมดูล ในการควบคุมความชื้น กำหนดให้การทำงานเริ่มจากการตรวจสอบเงื่อนไข ถ้าความชื้นภายในระบบควบคุมความชื้นน้อยกว่า 75 เปอร์เซ็นต์ ระบบจะเปิดการทำงาน Ultrasonic Mist-Maker หากความชื้นมากกว่า 90 เปอร์เซ็นต์ ระบบจะปิดการทำงาน Ultrasonic Mist-Maker แต่ถ้าหากยังไม่มากกว่า 90 เปอร์เซ็นต์ระบบจะเปิดการทำงาน Ultrasonic Mist-Maker ต่อไป แผนภาพของระบบแสดงดังรูปที่ 3.1



รูปที่ 3.1 แผนภาพการทำงานของระบบควบคุมความชื้น

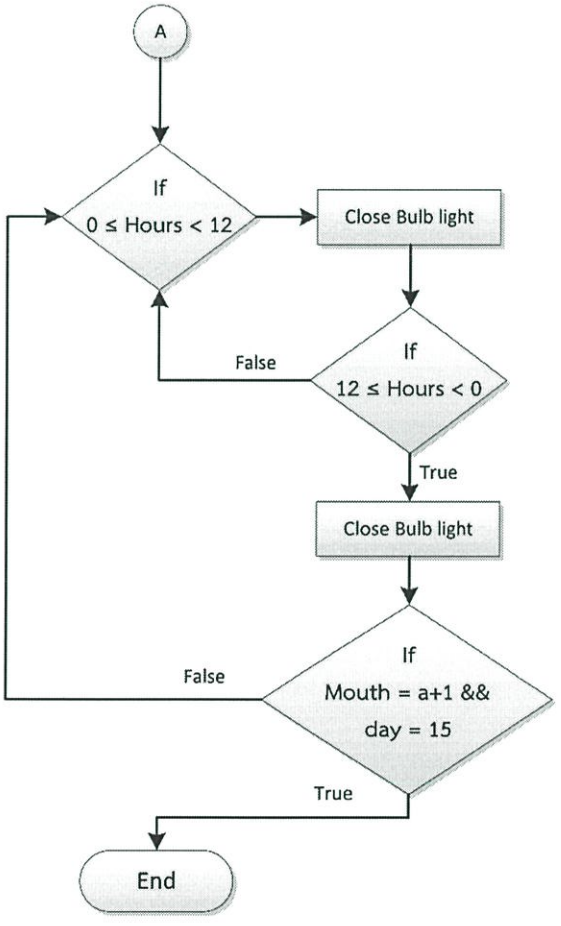
3.1.2 ออกแบบลำดับการทำงานของโปรแกรมในส่วนของการควบคุมแสงสว่าง

การออกแบบระบบการทำงานของ Raspberry Pi ที่เชื่อมต่อกับรีเลย์โมดูล ในการควบคุมแสง ซึ่งเป็นระยะที่สองที่เห็ดต้องได้รับแสงเพื่อให้ใยเหลืองปกคลุมอาหาร หลังจากระยะแรกที่เป็นระยะที่ใยขาวปกคลุมอาหาร 80 เปอร์เซ็นต์ ต้องทำการเปิด และปิดไฟทุก ๆ 6 ชั่วโมง กำหนดให้การทำงานระบบเริ่มจากการตรวจสอบเงื่อนไข ถ้าขึ้นเดือนที่ 2 และวันที่ 1 ตั้งแต่เวลาการเริ่มเพาะเลี้ยงเป็นจริง และต่อด้วยเงื่อนไขดังนี้ ถ้าอยู่ในช่วงเวลา 0 นาฬิกา ถึง 5.59 นาฬิการะบบจะเปิดไฟ ถ้าอยู่ในช่วงเวลา 6 นาฬิกา ถึง 11.59 นาฬิการะบบจะปิดไฟ ถ้าอยู่ในช่วงเวลา 12 นาฬิกา ถึง 17.59 นาฬิการะบบจะเปิดไฟ และถ้าอยู่ในช่วงเวลา 18 นาฬิกา ถึง 23.59 นาฬิการะบบจะปิดไฟ ตลอดระยะเวลา 7 วัน แผนภาพของระบบแสดงดังรูปที่ 3.2



รูปที่ 3.2 แผนการทำงานของระบบควบคุมแสง
(ระยะที่เห็ดต้องได้รับแสงเพื่อให้ใยเหลืองปกคลุมอาหาร)

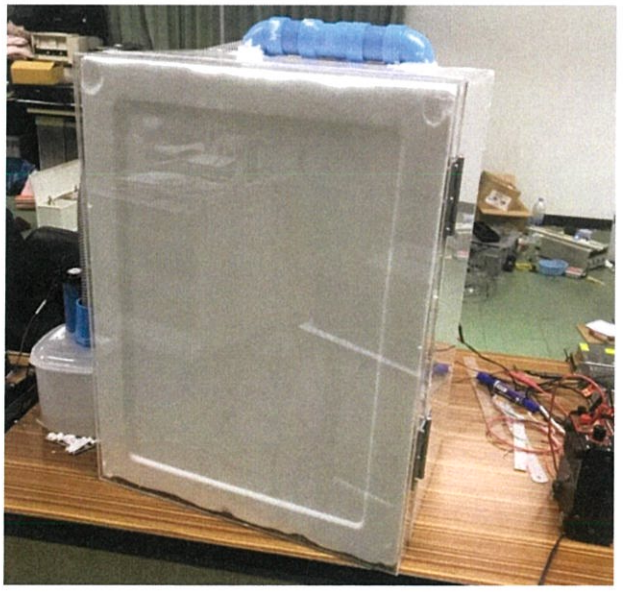
ระยะที่สามเป็นระยะที่เห็ดต้องได้รับแสงเพื่อออกดอก ต้องทำการเปิด และปิดไฟทุก ๆ 12 ชั่วโมง กำหนดให้การทำงานระบบเริ่มจากการตรวจสอบเงื่อนไข ถ้าขึ้นเดือนที่ 2 และวันที่ 8 ต่อด้วยเงื่อนไขดังนี้ ถ้าอยู่ในช่วงเวลา 0 นาฬิกา ถึง 11.59 นาฬิการะบบจะเปิดไฟ และถ้าอยู่ในช่วงเวลา 12 นาฬิกา ถึง 23.59 นาฬิกา ระบบจะปิดไฟ ตลอดระยะเวลา 15 วัน แผนภาพของระบบแสดงดังรูปที่ 3.3



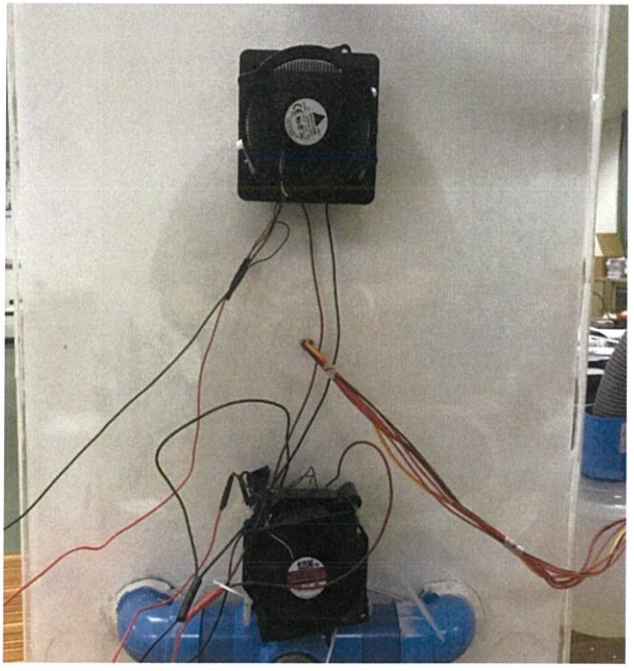
รูปที่ 3.3 ออกแบบลำดับการทำงานของระบบควบคุมแสง
(ระยะที่เห็ดต้องได้รับแสงเพื่อออกดอก)

3.1.3 การออกแบบตู้ที่ติดตั้งระบบควบคุมอุณหภูมิ ความชื้น และแสง

ในการออกแบบและการจัดทำตู้ควบคุมสภาพแวดล้อมแบบปิดเพาะเลี้ยงเห็ดถั่งเช่าสีทอง แสดงดังรูปที่ 3.4 สิ่งที่เป็นปัจจัยหลักของระบบจะมีทั้งหมด 3 อย่างโดยอย่างแรกคือ เรื่องอุณหภูมิ ซึ่งในที่นี้จะใช้ Peltier Cooler ในการทำความเย็น โดยจะใช้ทำงานร่วมกับ ฮีทซิงค์ และพัดลม แสดงดังรูปที่ 3.5 เพื่อทำให้ภายในตู้ควบคุมสภาพแวดล้อม มีอุณหภูมิที่ใกล้เคียงกัน ในแต่ละจุด ซึ่งการเก็บรักษาความเย็นจะใช้กล่องโฟมที่หุ้มด้วยโครงที่เป็นแผ่นอะคริลิก เพื่อเพิ่มความแข็งแรงให้แก่ตัวตู้ควบคุมสภาพแวดล้อม ปัจจัยอย่างต่อไป คือความชื้น ซึ่งเห็ดจะสามารถเจริญเติบโตได้ในอากาศที่มีความชื้นสัมพัทธ์ ระหว่าง 75-95 เปอร์เซ็นต์ ดังนั้นจึงต้องสร้างเครื่องทำหมอกเพื่อเพิ่มความชื้นสัมพัทธ์ในอากาศ โดยจะใช้ Ultrasonic Mist Maker เพื่อสร้างหมอก จากนั้นจะใช้พัดลมขนาดเล็ก พัดหมอกเข้าสู่ตู้ควบคุมสภาพแวดล้อม ซึ่งจะมีด้วยกันสองชุด โดยตำแหน่งที่จะปล่อยให้หมอกเข้าสู่ระบบนั้น จะให้หมอกเข้าจากด้านบน ทางซ้าย และขวาจะเป็นชุดที่หนึ่ง และเข้าจากด้านล่างทั้งซ้ายและขวาเป็นชุดที่สอง เพื่อจะให้ความชื้นกระจายให้ทั่วทั้งตู้ และควบคุมปริมาณของหมอก ในแต่ละตำแหน่งได้ง่ายทั้งนี้ อุณหภูมิและความชื้นภายในตู้ จะถูกวัดค่าจากเซนเซอร์วัดอุณหภูมิและความชื้น AM2302 และปัจจัยสุดท้าย เป็นเรื่องของแสงสว่าง โดยเห็ดถั่งเช่าสีทอง จะมีช่วงระยะเวลาหนึ่งในชีวิต ที่ต้องการแสงสว่าง ดังนั้นตู้ควบคุมสภาพแวดล้อมจำเป็นที่จะต้องติดตั้งหลอดไฟ LED ที่เป็นแสงชนิด Daylight แสดงดังรูปที่ 3.6 เพื่อจำลองแสงของพระอาทิตย์ ซึ่งจะถูกควบคุมให้ เปิด-ปิด โดยจะถูกควบคุมด้วยรีเลย์โมดูล และจะใช้เซนเซอร์ BH1750FVI ในการวัดค่าความสว่างภายในตู้ ซึ่งค่าจากเซนเซอร์แต่ละตัวจะถูกเก็บไว้ในฐานข้อมูลเพื่อที่จะนำค่าสถานะต่าง ๆที่อยู่ภายในตู้ มาแสดงบนเว็บบราวเซอร์



รูปที่ 3.4 ตู้ควบคุมสภาพแวดล้อมแบบปิดเพาะเลี้ยงเห็ดถั่งเช่าสีทอง



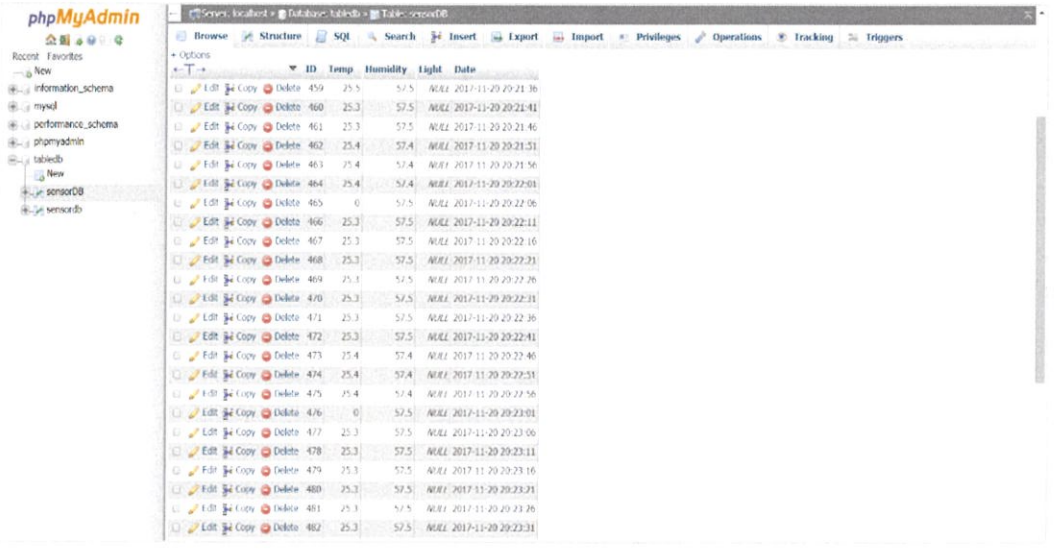
รูปที่ 3.5 ฮีทซิงค์ และพัดลม

รูปที่ 3.6 ตำแหน่งที่ทำการติดตั้งหลอดไฟ LED

3.1.4 การออกแบบ และสร้างฐานข้อมูล

Raspberry Pi รับค่าจากเซนเซอร์อุณหภูมิ ความชื้น และความเข้มแสง นำมาทำการประมวลผลโดยการเขียนโปรแกรมภาษา Python ดึงค่าอุณหภูมิ ค่าความชื้น และค่าความเข้มแสง แสงมาแสดงบนบราวเซอร์

ในการสร้างฐานข้อมูล และการสร้างหน้าเว็บแสดงค่าจากฐานข้อมูล ขั้นแรกนำข้อมูลเข้าสู่ฐานข้อมูลโดยการเชื่อมต่อเซนเซอร์กับ Raspberry Pi ให้ Raspberry Pi นำค่าที่ได้รับนั้นเข้าสู่ฐานข้อมูลที่ได้สร้างขึ้นให้ แสดงค่าอุณหภูมิ ค่าความชื้น ค่าความเข้มแสง และแสดงเวลาที่เซนเซอร์ได้ทำการตรวจวัด แสดงดังรูปที่ 3.7



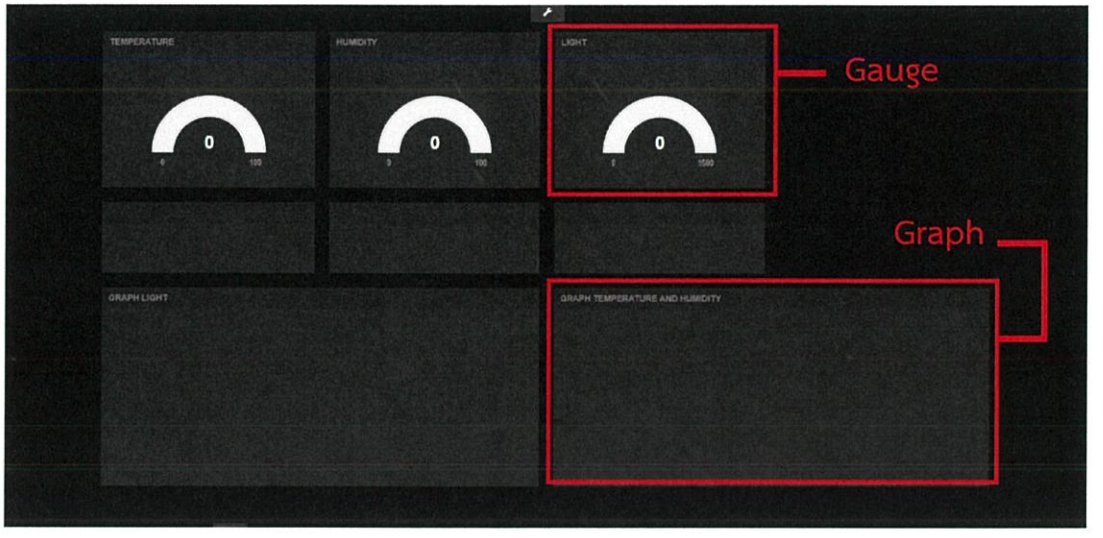
รูปที่ 3.7 หน้าเว็บ localhost/phpmyadmin ที่รับค่าจากเซนเซอร์วัดอุณหภูมิ ความชื้น และเซนเซอร์วัดความเข้มแสง

เมื่อรับค่าตัวแปรต่าง ๆ และเวลาที่ทำการวัดค่าตัวแปรต่อมาทำการเขียนโปรแกรมโดยใช้ภาษา html และ php ดึงค่าตัวแปรจากฐานข้อมูล phpmyadmin ให้แสดงค่าตัวแปรผ่านหน้าจอหน้าเว็บบราวเซอร์ที่สร้างขึ้นโดยค่าที่ทำการแสดงได้แก่ ค่าอุณหภูมิ ค่าความชื้น ค่าความเข้มแสง และแสดงเวลาที่เซนเซอร์ทำการตรวจวัด

3.1.5 ออกแบบการแสดงบนหน้าเว็บไซต์ NETPIE.io

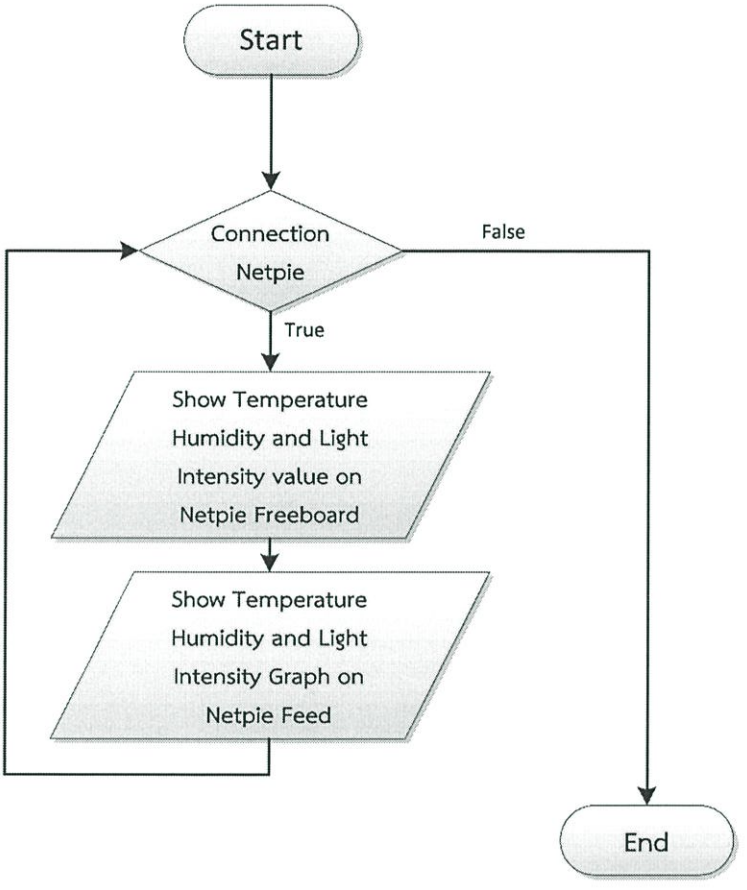
การออกแบบหน้าเว็บไซต์ NETPIE.io เพื่อแสดงผลค่าสถานะต่าง ๆ ซึ่งประกอบไปด้วยอุณหภูมิ ความชื้น และความเข้มแสง ถูกแสดงในรูปแบบมาตรวัด Gauge โดยกำหนดขอบเขตของอุณหภูมิ 0 ถึง 100 องศาเซลเซียส ความชื้น 0 ถึง 100 เปอร์เซ็นต์ และความเข้มแสง 0 ถึง 1500

ลักซ์ และการแสดงค่าสถานะในรูปแบบกราฟเส้น กำหนดให้สร้างกราฟเส้นทุก ๆ 1 นาที และกราฟออกแบบให้แสดงเป็นช่วงเวลา 1 ชั่วโมง การออกแบบหน้าเว็บแสดงดังรูปที่ 3.8



รูปที่ 3.8 หน้าเว็บแสดงผลผ่านหน้าเว็บไซต์ NETPIE.io บน Netpie Freeboard

ในการเข้าสู่หน้าเว็บ NETPIE Freeboard การนำข้อมูลที่ถูกส่งเข้า Netpie.io ต้องทำการระบุตัวตน ซึ่งประกอบด้วย Application ID, key และ Secret ในการเชื่อมต่อเพื่อสามารถจะทำให้ค่าที่ถูกส่งมานั้นแสดงบน NETPIE Freeboard แผนภาพการเข้าถึงแสดงดังรูปที่ 3.9



รูปที่ 3.9 ลำดับการเข้าถึงในส่วนของการเข้าสู่ NETPIE.io และแสดงบนหน้า NETPIE Freeboard

3.2 เครื่องมือที่ใช้ในการทดลอง

ในการออกแบบระบบโดยรวมจะใช้เครื่องมือที่มีทั้งฮาร์ดแวร์ และซอร์ฟแวร์ โดยจะแบ่งโปรแกรมออกเป็นส่วนของฮาร์ดแวร์ และซอร์ฟแวร์ที่ใช้ในการออกแบบได้ดังนี้

ฮาร์ดแวร์ที่ใช้ได้แก่

1) Microcontroller

สำหรับ Microcontroller ที่เลือกใช้ในปริญญานิพนธ์ คือ Raspberry Pi 3 model B แสดงดังรูปที่ 3.10

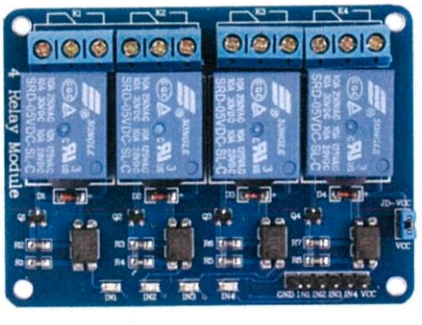
รูปที่ 3.10 Raspberry Pi 3 model B

Raspberry Pi 3 model B แสดงดังรูปที่ 3.10 เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถรันด้วยระบบปฏิบัติการลินุกซ์ (Linux) เช่น Raspbian (Debian), Pidora (Fedora) โดย Broadcom BCM2836 900 MHz รองรับ 802.1n ทั้งคลื่น 2.4 Ghz และ 5 GHz มีพอร์ตหน้าจอ HDMI, USB 2.0 4 พอร์ต, Ethernet และขา GPIO 40 ขา

ปริญญานิพนธ์นี้เลือกใช้ระบบปฏิบัติการ Raspbian ในการติดตั้งให้ Micro SD Card และ Raspberry Pi 3 ทำหน้าหน้าที่เป็นตัวประมวลผลของระบบ และทำตัวเป็นเซิร์ฟเวอร์เพื่อส่งค่าสถานะเข้าสู่ระบบฐานข้อมูล

2) รีเลย์โมดูล

สำหรับปริญญานิพนธ์นี้ได้ใช้รีเลย์โมดูล 4 ช่อง แสดงดังรูปที่ 3.11 แรงดันไฟฟ้าของคอยล์ 5-48 V ทนกระแสสูงสุด 10 A ทนอุณหภูมิได้ -40°C ถึง 85°C และทนการสั่นสะเทือนได้ 10 ถึง 55 Hz มีช่องสั่งสัญญาณ INT1, INT2, INT3, INT4, VCC และ GND เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ เป็นรีเลย์โมดูล 4 ช่อง ใช้สำหรับสวิตช์เพื่อเปิด และปิดการทำงาน Ultrasonic Mist Maker และหลอดไฟ LED



รูปที่ 3.11 รีเลย์โมดูล

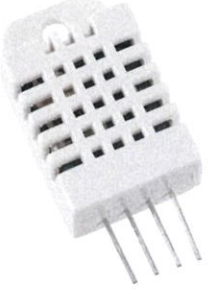
3) Raspberry Pi Camera Board

สำหรับปริญญานิพนธ์นี้ได้ใช้ Raspberry Pi Camera Board DHT22 แสดงดังรูปที่ 3.12 เป็นโมดูลกล้องที่ออกแบบมาใช้งานร่วมกับบอร์ด Raspberry Pi โดย สามารถเชื่อมต่อกับซ๊อกเก็ต CAMERA บนบอร์ด ที่มีการเชื่อมต่อแบบ CSI bus ความละเอียดของกล้อง 5MP (2592 x 1944 pixels) สามารถถ่ายวิดีโอความละเอียด 1080p30 , 720p60 และ 640x480p60/9 เพื่อใช้ในการติดตามการเจริญเติบโตของเห็ดถั่งเช่าสีทอง

รูปที่ 3.12 Raspberry Pi Camera Board

4) เซนเซอร์วัดความชื้น และอุณหภูมิ (Temperature and Humidity sensor)

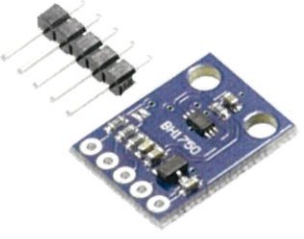
สำหรับปริญญานิพนธ์นี้ได้ใช้เซนเซอร์วัดความชื้น และอุณหภูมิ รุ่น DHT22 แสดงดังรูปที่ 3.13 ตรวจวัดความชื้น และอุณหภูมิ ภายในกล่องควบคุมอุณหภูมิ ที่สามารถวัดค่าความชื้นได้ตั้งแต่ 0 - 100% ค่าความผิดพลาดในการวัดความชื้น 2% และสามารถวัดอุณหภูมิได้ตั้งแต่ -40 ถึง 80 องศาเซลเซียส ค่าความผิดพลาดในการวัดอุณหภูมิ 0.5%



รูปที่ 3.13 DHT22 Temperature and Humidity Sensor Module

5) เซนเซอร์วัดความเข้มแสง (Light Intensity Sensor Module)

สำหรับปริญญานิพนธ์นี้ได้ใช้เซนเซอร์วัดความเข้มแสง รุ่น BH1750FVI (GY-302) แสดงดังรูปที่ 3.14 ที่ให้เอาต์พุตออกมาเป็นสัญญาณดิจิตอล โดยเอาต์พุตที่ได้มีหน่วยเป็นลักซ์ (LUX) โดยใช้ขาแค่ 4 ขา (VCC, SCL, SDA, GND) ความเร็วสำหรับบัส I2C ได้ถึง 400kHz ความละเอียด: 16 บิต ได้ค่า 1-65536 หน่วยเป็น Lux (step: 0.5 Lux, 1 Lux, หรือ 4 Lux ขึ้นอยู่กับโหมดการวัดที่เลือก)



รูปที่ 3.14 BH1750FVI (GY-302) Light Sensor Module

6) Ultrasonic Mist Maker

สำหรับปริญญานิพนธ์นี้ได้ใช้ Ultrasonic Mist Maker แสดงดังรูปที่ 3.15 เป็นเครื่องทำหมอกอัลตราโซนิคใช้น้ำเพื่อสร้างหมอก ทำงานบนหลักการของการสั่นที่ทำให้น้ำกลายเป็นหยดเล็ก ๆ ที่ระเหยกลายเป็นเมฆที่มีหมอก

รูปที่ 3.15 Ultrasonic Mist Maker

7) Cooler Peltier

สำหรับปริญญานิพนธ์นี้ได้ใช้ Cooler Peltier หรือแผ่นร้อนเย็น ขนาด 40mm x 40mm แสดงดังรูปที่ 3.16 เป็นอุปกรณ์ที่สามารถสร้างความเย็นได้ที่ด้านหนึ่ง และปล่อยความร้อนออกมาที่อีกด้านหนึ่ง เพียงแค่จ่ายไฟฟ้ากระแสตรงเข้าไปในตัวอุปกรณ์เท่านั้น

รูปที่ 3.16 Cooler Peltier

ซอร์ฟแวร์ที่ใช้ได้แก่

1) Python
2) SQL
3) PHP
4) HTML

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดลองวัดสัญญาณเอาต์พุตจาก เซนเซอร์วัดอุณหภูมิ และความชื้นด้วยเครื่องออสซิลโลสโคป

การทดลองวัดสัญญาณเอาต์พุตจาก DHT22 เซนเซอร์วัดอุณหภูมิ และความชื้น อ่านค่าสัญญาณเอาต์พุตด้วยเครื่องออสซิลโลสโคป เพื่อทดสองความถูกต้องในการประมวลผลสัญญาณที่ถูกแสดงเป็นตัวเลข

3.3.2 การทดลองการทำงานของเซนเซอร์

3.3.2.1 การทดลองวัดอุณหภูมิของเซ็นเซอร์อุณหภูมิ DHT22 เทียบกับเทอร์โมมิเตอร์

3.3.2.2 ผลการทดลองวัดความเข้มแสงของเซนเซอร์ BH1750fvi เทียบกับ Lux meter Yokogawa 51002

3.3.3 การทดลองการแสดงผลอุณหภูมิ ความชื้น และความเข้มแสงเว็บบราวเซอร์

การทดลองการแสดงผลการทำงานระบบบน Web browser เป็นการแสดงค่าสถานะของระบบที่ประกอบไปด้วย อุณหภูมิ ความชื้น และความเข้มแสง ทำการส่งข้อมูลเข้าฐานข้อมูล

ระบบควบคุม และแสดงข้อมูลผ่านหน้า NETPIE Freeboard แสดงผลในรูปแบบ Gauge แสดงผลในรูปแบบ Graph

3.3.4 ผลการทดสอบการแสดงค่าอุณหภูมิ ความชื้น และความเข้มแสงจากฐานข้อมูล

การทดลองการแสดงผลข้อมูลที่ถูกส่งจากเซนเซอร์เข้าสู่ Server ฐานข้อมูลของ Raspberry Pi ค่าสถานะของระบบที่ประกอบไปด้วย อุณหภูมิ ความชื้น ความเข้มแสง วัน และเวลาที่ถูกบันทึกเข้าสู่ฐานข้อมูล หน้าเว็บจะแสดงค่าสถานะต่าง ๆ บน Web browser จากปุ่มค้นหาจากการใส่ ปี เดือน วัน ชั่วโมง และนาที

3.3.5 การทดสอบหน้าเว็บแสดงภาพที่ถูกถ่ายด้วย PiCamera module

การทดลองการแสดงผลภาพที่ถูกถ่ายด้วย PiCamera module เพื่อติดตามการเจริญเติบโตเห็ดถั่งเช่าสีทองด้วยการสังเกตการณ์ ระบบมีการถ่ายภาพภายในกล่องควบคุมสภาพแวดล้อมสำหรับการสังเกตการเจริญเติบโต โดยแสดงผลผ่านหน้าเว็บเป็นภาพถ่ายที่ถูกถ่ายทุก ๆ ชั่วโมงตลอดระยะเวลาการเจริญเติบโต ด้วยการค้นหาจากไฟล์ภาพที่ชื่อภาพถ่ายประกอบไปด้วย ถ่าย ปี เดือน วัน ชั่วโมง และนาที ที่ทำการบันทึกภาพ

3.3.6 การทดสอบปลูกเห็ดถั่งเช่าสีทองในระบบควบคุมสภาพแวดล้อมแบบปิด

การทดลองการเพาะเลี้ยงในระบบควบคุมสภาพแวดล้อมแบบปิด ที่ติดตั้งระบบควบคุมอุณหภูมิ ความชื้น และแสง โดยทำการทดลองเพาะเลี้ยงเห็ดถั่งเช่าสีทอง

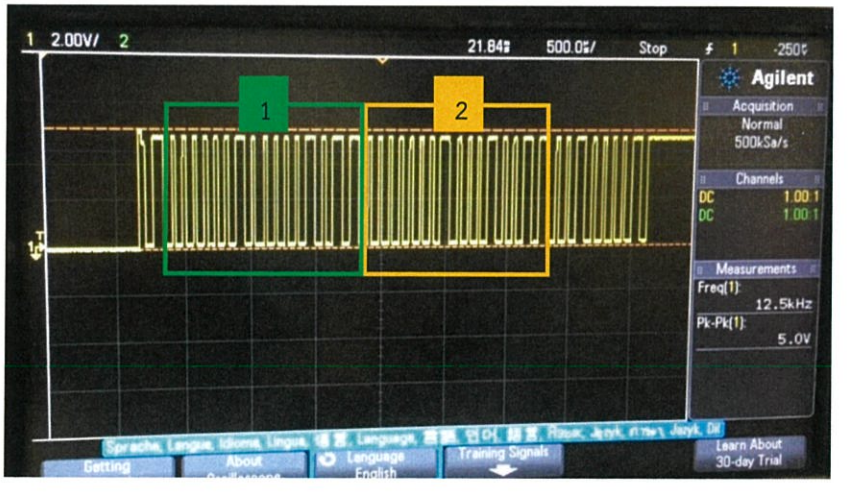
บทที่ 4

ผลการทดลอง

ในบทนี้จะแสดงผลการทดสอบของระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทองจากบทที่ 3 ซึ่งได้ทำการเก็บผล โดยแบ่งออกเป็นส่วนต่าง ๆ ดังต่อไปนี้

4.1 ผลการทดสอบวัดสัญญาณเอาต์พุตจาก เซนเซอร์วัดอุณหภูมิ และความชื้นด้วยเครื่องออสซิลโลสโคป

การทดสอบเซนเซอร์อุณหภูมิ และความชื้นสามารถอ่านค่าสัญญาณเอาต์พุตที่เครื่องออสซิลโลสโคป สัญญาณมีลักษณะเป็นสัญญาณพัลส์ที่สามารถแปลงเป็นบิตข้อมูลที่มีขนาด 40 บิต ซึ่งประกอบด้วย สัญญาณพัลส์ค่าความชื้น 16 บิต สัญญาณพัลส์ค่าอุณหภูมิ 16 บิต และสัญญาณพัลส์ Check-sum 8 บิต โดยการอ่านค่าบิตสัญญาณข้อมูลด้วยความกว้างพัลส์ที่ต่างกันสัญญาณบิต 0 มีความกว้างพัลส์ 26 µs และสัญญาณบิต 1 มีความกว้างพัลส์ 70 µs แสดงดังรูปที่ 4.1



รูปที่ 4.1 สัญญาณเอาต์พุตจากเซนเซอร์อุณหภูมิ และความชื้น

สำหรับสัญญาณเอาต์พุตในช่วงสัญญาณที่บอกค่าความชื้น และการอ่านบล็อกข้อมูลแสดงดังรูปที่ 4.2 การหาค่าความชื้นสามารถหาจากการแปลงอ่านสัญญาณเอาต์พุตออกเป็นเลขฐานสอง และออกมาเป็นตัวเลขฐานสิบได้ด้วยการแปลงค่าเลขฐานสองเป็นฐานสิบดังนี้

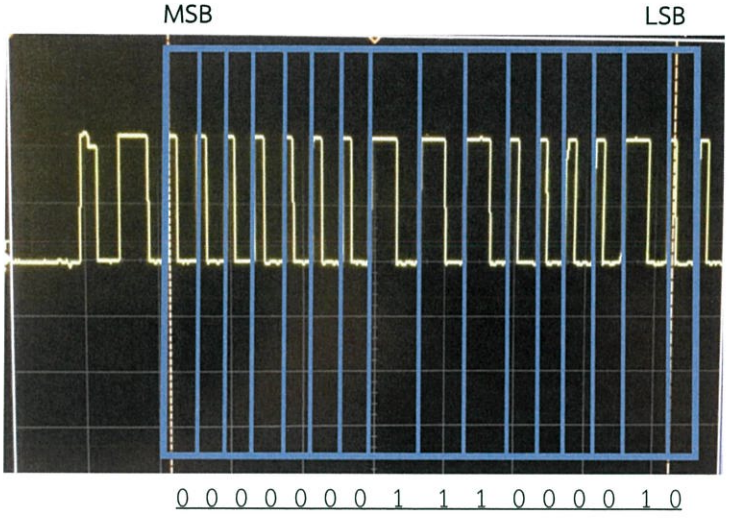
Binary system: 0000 0001 1100 0010

Decimal system: $2^8 + 2^7 + 2^6 + 2^2 + 2^1 = 256 + 128 + 64 + 2$

$= 450$

หารด้วย 10 = 450/10

= 45.0 %



รูปที่ 4.2 การอ่านบล็อกข้อมูลค่าความชื้น

สำหรับสัญญาณเอาต์พุตในช่วงสัญญาณที่บอกค่าอุณหภูมิ และการอ่านบล็อกข้อมูล แสดงดังรูปที่ 4.3 การหาค่าความอุณหภูมิสามารถหาจากการแปลงสัญญาณเอาต์พุตเป็นเลขฐานสอง และออกมาเป็นตัวเลขฐานสิบได้ด้วยการแปลงค่าเลขฐานสองเป็นฐานสิบดังนี้

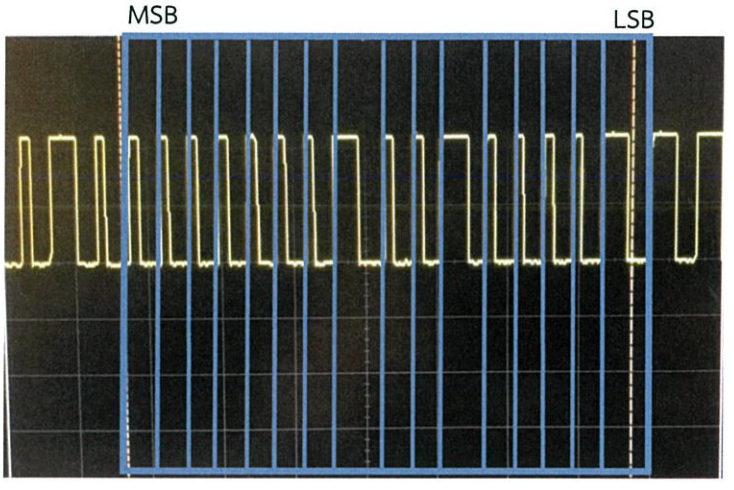
Binary system: 0000 0001 0010 0001

Decimal system: $2^8 + 2^5 + 2^0 = 256+32+1$

$= 289$

หารด้วย 10 = 289/10

= 28.9 Celsius



รูปที่ 4.3 การอ่านบล็อกข้อมูลค่าอุณหภูมิ

4.2 ผลทดลองการทำงานของเซนเซอร์

4.2.1 ผลทดลองวัดอุณหภูมิของเซนเซอร์อุณหภูมิ DHT22 เทียบกับเทอร์โมมิเตอร์

การทดลองนี้มีจุดประสงค์เพื่อทดสอบความสามารถในการอ่านค่าอุณหภูมิ ของเซนเซอร์ AM2302 ว่ามีความแม่นยำหรือไม่ จะทดสอบโดยการนำเซนเซอร์และเทอร์โมมิเตอร์ ลงไปแช่ในน้ำแข็ง แล้วให้เซนเซอร์และเทอร์โมมิเตอร์ อ่านค่าอุณหภูมิที่ได้ ซึ่งจะทำการทดลองทั้งหมดสิบครั้ง โดยผลการทดลองทั้งหมดจะถูกนำมาแสดงตามตารางที่ 4.1 ดังนี้

ตารางที่ 4.1 ผลการทดลองของเซนเซอร์AM2302เทียบกับเทอร์โมมิเตอร์

ครั้งที่	ค่าที่วัดได้จากเซ็นเซอร์ AM2302 (°C)	ค่าที่วัดได้จากเทอร์โมมิเตอร์ (°C)	เปอร์เซ็นต์ความผิดพลาด
1	0.10	-0.50	0.22
2	0.30	-0.20	0.18
3	0.10	0.00	0.03
4	0.10	0.00	0.03
5	0.00	-0.10	0.03
6	0.00	-0.20	0.07
7	0.00	-0.20	0.07
8	0.00	-0.10	0.03

9	0.00	0.00	0
10	0.10	0.00	0.03

4.2.2 ผลการทดลองวัดความเข้มแสงของเซนเซอร์ BH1750fvi เทียบกับ Lux meter Yokogawa 51002

4.2.2.1 ทดสอบเซนเซอร์ตามมุมของแหล่งกำเนิดแสง

การทดลองเพื่อทดสอบการตอบสนองของเซนเซอร์แสง รุ่น BH1750FVI บน Module GY302 โดยการทดสอบทำการเปลี่ยนมุมของเซนเซอร์กับแหล่งกำเนิดแสง เทียบกับ Lux meter โดยที่ระยะห่างของจุดที่วัดแสงกับแหล่งกำเนิดแสง ห่างกัน 3 เมตร โดยหลังจากทำการทดลอง จะได้ผลการทดลองตามตารางที่ 4.2.1 ดังต่อไปนี้

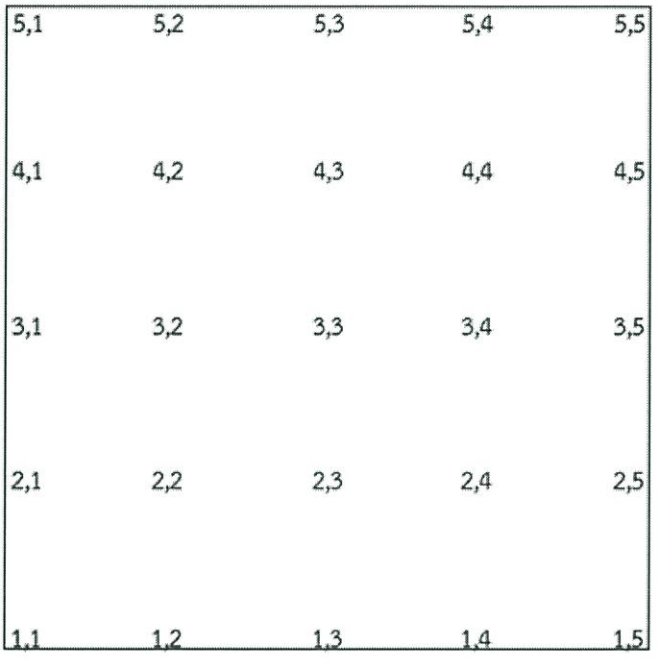
ตารางที่ 4.2 ผลการทดสอบเซนเซอร์อุณหภูมิเทียบกับ Lux meter Yokogawa 51002

มุมที่ติดตั้งหลอดไฟ (องศา)	มุมที่เซนเซอร์ทำกับหลอดไฟ (องศา)	ค่าที่วัดได้จากเครื่องวัด (Lux)	ค่าที่วัดได้จากเซ็นเซอร์ (Lux)
0	0	40	41
	10	41	41.6
	20	40	40
	30	39	39
	40	36	36
	50	29	29
	60	24	23
	70	20	21
	80	13	14
	90	0	1.2
45	0	40	41
	10	41	41
	20	40	40
	30	38	39
	40	36	36
	50	33	33
	60	25	25

	70	18	19
	80	12	12
	90	1	2
90	0	41	42
	10	41	42
	20	41	42
	30	39	40
	40	35	36
	50	29	31
	60	24	24
	70	17	17
	80	8	8
	90	1	2

4.2.2.2 ผลการทดสอบเซนเซอร์โดยการวัดตามตำแหน่ง

จากการทดลองเพื่อทดสอบความสามารถในการวัดค่าความสว่างของเซนเซอร์วัดความสว่าง BH1750FVI เปรียบเทียบกับ Lux meter Yokogawa 51002 โดยวัดตามตำแหน่งบนพื้นผิวทั้งหมด 25 จุด โดยที่พื้นผิวมีระยะห่างจากแหล่งกำเนิดแสง 2.2 เมตร ซึ่งตำแหน่งที่ทำการวัดจะเป็นไปตามรูปที่ 4.4 และเมื่อได้ทำการทดสอบแล้ว จะได้ผลการทดสอบตามตารางที่ 4.2.2.2 ดังต่อไปนี้



รูปที่ 4.4 ตำแหน่งในการวัดแสงของเซนเซอร์กับ Lux meter

ตารางที่ 4.3 ผลการทดสอบเซ็นเซอร์โดยการวัดตามตำแหน่ง

ตำแหน่ง	ค่าที่วัดได้จากเครื่องวัด (Lux)	ค่าที่วัดได้จากเซ็นเซอร์ (Lux)
1,1	281	227
1,2	351	315
1,3	475	408
1,4	549	503
1,5	589	560
2,1	296	234
2,2	372	310
2,3	458	407
2,4	528	485
2,5	579	540
3,1	278	201
3,2	345	276
3,3	424	366

3,4	496	434
3,5	538	490
4,1	256	194
4,2	315	250
4,3	386	325
4,4	440	380
4,5	469	420
5,1	237	211
5,2	282	230
5,3	341	305
5,4	400	345
5,5	418	366

4.3 ผลการทดลองการแสดงผลอุณหภูมิ ความชื้น และความเข้มแสงผ่านบราวเซอร์

การแสดงผลค่าสถานะประกอบด้วย อุณหภูมิ ความชื้น และความเข้มแสง บนหน้าเว็บ NETPIE.io ผ่าน NETPIE Freeboard ด้วยการใช้ Microgear ซึ่งเป็น Client Library ทำหน้าที่นำค่าที่ถูกส่งจาก Raspberry Pi มาแสดงผลในรูปแบบ Gauge และในรูปแบบ Graph ประกอบไปด้วย 2 Graph ได้แก่ Graph เปรียบเทียบอุณหภูมิ และความชื้น และ Graph ความเข้มแสง แสดงดังรูปที่ 4.5

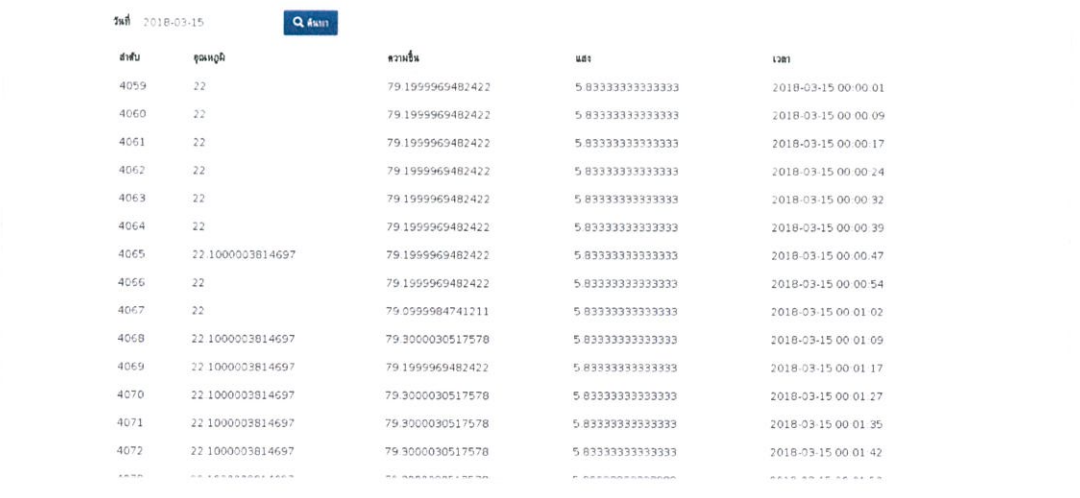
รูปที่ 4.5 หน้าบราวเซอร์ NETPIE Freeboard แสดงสภานะค่าอุณหภูมิ ความชื้น และแสง

4.4 ผลการทดสอบการแสดงค่าอุณหภูมิ ความชื้น และความเข้มแสงจากฐานข้อมูล

การติดตามการเจริญเติบโตของเห็ดถั่งเช่าสีทองจากการส่งค่าตัวแปรต่าง ๆ ที่ตรวจวัดจากเซ็นเซอร์อุณหภูมิ ความชื้น และแสง เข้าสู่ฐานข้อมูล การนำค่าต่าง ๆ มาแสดงโดยทำการเขียนหน้าเว็บค้นหาข้อมูลจากวันเวลาที่ได้ทำการบันทึกข้อมูลลงฐานข้อมูล การค้นหาทำได้โดยการพิมพ์เวลาซึ่งประกอบด้วย ปี เดือน วัน ชั่วโมง และนาที แสดงดังรูปที่ 4.6

รูปที่ 4.6 หน้าเว็บบราวเซอร์แสดงหน้าการค้นหาข้อมูลตัวแปรต่าง ๆ จากฐานข้อมูล

ค่าตัวแปรต่าง ๆ ที่ถูกแสดงเมื่อทำการกดค้นหาจะแสดงค่าที่ประกอบด้วย ลำดับ อุณหภูมิ ความชื้น แสง และเวลาที่ถูกบันทึกลงฐานข้อมูล ดังรูปที่ 4.7 เพื่อติดตามการเจริญเติบโตและสามารถสังเกตความผิดปกติได้จากอุณหภูมิที่เปลี่ยนไป ความชื้นที่เปลี่ยนไป หรือการไม่ทำงานของระบบที่ผิดปกติ


วันที่ 2018-03-15 ค้นหา

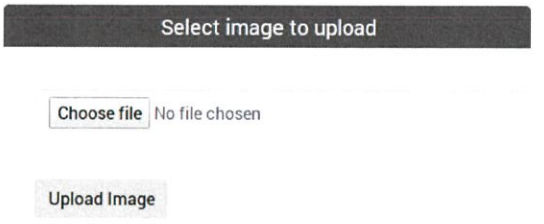
ลำดับ	อุณหภูมิ	ความชื้น	แสง	เวลา
4059	22	79.1999969482422	5.83333333333333	2018-03-15 00:00:01
4060	22	79.1999969482422	5.83333333333333	2018-03-15 00:00:09
4061	22	79.1999969482422	5.83333333333333	2018-03-15 00:00:17
4062	22	79.1999969482422	5.83333333333333	2018-03-15 00:00:24
4063	22	79.1999969482422	5.83333333333333	2018-03-15 00:00:32
4064	22	79.1999969482422	5.83333333333333	2018-03-15 00:00:39
4065	22.1000003814697	79.1999969482422	5.83333333333333	2018-03-15 00:00:47
4066	22	79.1999969482422	5.83333333333333	2018-03-15 00:00:54
4067	22	79.0999984741211	5.83333333333333	2018-03-15 00:01:02
4068	22.1000003814697	79.3000030517578	5.83333333333333	2018-03-15 00:01:09
4069	22.1000003814697	79.1999969482422	5.83333333333333	2018-03-15 00:01:17
4070	22.1000003814697	79.3000030517578	5.83333333333333	2018-03-15 00:01:27
4071	22.1000003814697	79.3000030517578	5.83333333333333	2018-03-15 00:01:35
4072	22.1000003814697	79.3000030517578	5.83333333333333	2018-03-15 00:01:42

รูปที่ 4.7 หน้าเว็บแสดงหน้าการค้นหาข้อมูลตัวแปรต่าง ๆ จากฐานข้อมูลเมื่อทำการกดค้นหา

4.5 ผลทดสอบหน้าเว็บแสดงภาพที่ถูกถ่ายด้วย PiCamera module

การติดตามการเจริญเติบโตเห็ดถั่งเช่าสีทองด้วยการสังเกตการณ์ ระบบมีการถ่ายภาพภายในกล่องควบคุมสภาพแวดล้อมสำหรับการสังเกตการเจริญเติบโตว่ามีการเจริญเติบโตมากน้อยเพียงใด โดยทำการสร้างหน้าเว็บบราวเซอร์แสดงภาพถูกถ่ายจาก PiCamera module โดยการเขียนโปรแกรมให้ PiCamera module ถ่ายภาพทุก ๆ ชั่วโมง ตลอดช่วงระยะเวลาการเติบโตของเห็ดถั่งเช่าสีทองในช่วงระยะใยขาวปกคลุม 80 เปอร์เซ็นต์ จึงถึงเกิดตุ้มดอกสีเหลือง

ภายในหน้าเว็บบราวเซอร์แสดงหน้าการค้นหาประกอบด้วย ปุ่มค้นหา ที่ใช้ค้นหารูปถ่ายจากโฟลเดอร์ file_image ภายใน Raspberry Pi แสดงดังรูปที่ 4.8



รูปที่ 4.8 หน้าเว็บแสดงหน้าการค้นหา

ในการค้นหารูปที่ถูกถ่าย ชื่อของรูปจะประกอบไปด้วย ลำดับภาพที่ถูกถ่าย ปี เดือน วัน ชั่วโมง และนาที ที่ทำการบันทึกภาพ แสดงเมื่อทำการกดค้นหารูปถ่าย และแสดงภาพถ่ายที่ถูกถ่ายบนหน้าเว็บ แสดงดังรูปที่ 4.9

รูปที่ 4.9 หน้าเว็บบราวเซอร์แสดงหน้าเมื่อทำการกดค้นหารูปถ่าย และแสดงภาพที่ถูกถ่ายบนหน้าเว็บบราวเซอร์

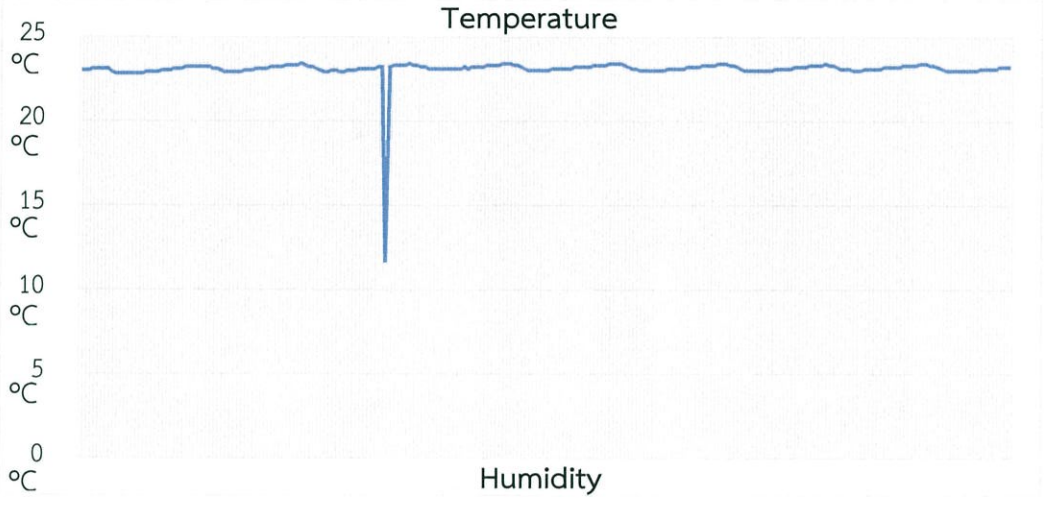
4.6 ผลการทดสอบปลูกเห็ดถั่งเช่าสีทองในระบบควบคุมสภาพแวดล้อมแบบปิด

ในการเพาะเลี้ยงเห็ดถั่งเช่าสีทองค่าตัวแปรต่าง ๆ ที่จำเป็นต่อการเจริญเติบโตประกอบไปด้วย อุณหภูมิ ความชื้น และแสงสว่าง ในระยะแรกของการเจริญเติบโตสภาพแวดล้อมที่เหมาะสมคือ อุณหภูมิ 22-23 องศาเซลเซียส และเป็นสภาพแวดล้อมที่มืดสนิทแสงจากภายนอกห้ามถูกเชื้อเห็ดเป็นระยะเวลา 1 เดือน ผลการเพาะเลี้ยงแสดงดังรูปที่ 4.10 และระยะการเกิดดอกสีเหลืองเป็นระยะเวลา 2 เดือน ผลการเพาะเลี้ยงแสดงดังรูปที่ 4.11

รูปที่ 4.10 ทดลองเพาะเห็ดถั่งเช่าสีทองเดือนที่ 1

รูปที่ 4.11 ทดลองเพาะเห็ดถั่งเช่าสีทองเดือนที่ 2

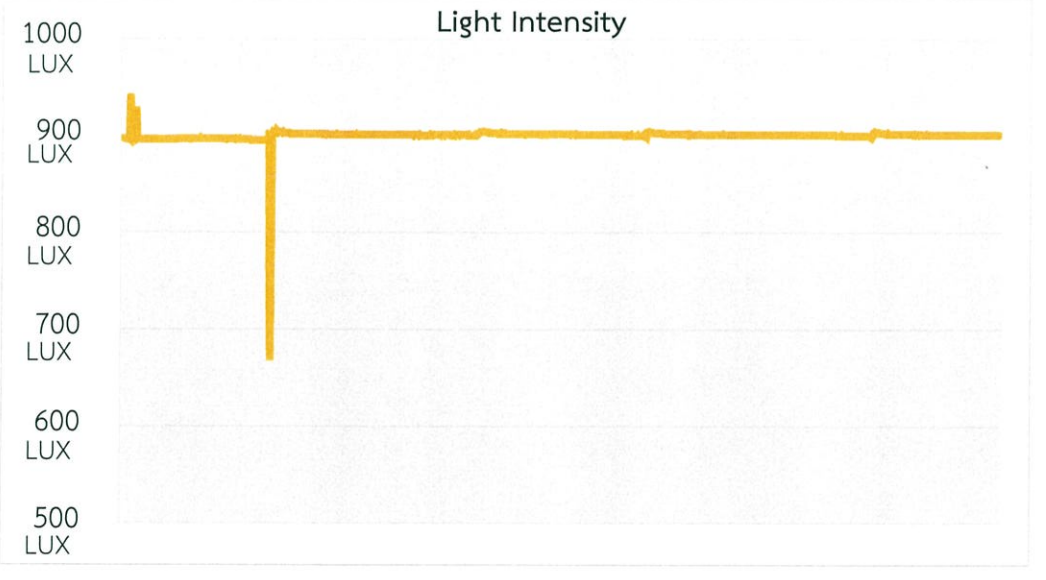
ผลทดลองของระบบควบคุมสภาพแวดล้อมค่าอุณหภูมิที่ได้เมื่อเปิดการทำงาน Cooler Peltier 2 ตัว โดยจ่ายแรงดัน 12 Volt และกระแส 5 Amp โดยการนำค่าจากฐานข้อมูลมาพล็อตกราฟ ซึ่งค่าที่ทำการตรวจวัดนั้นอยู่ระหว่าง 22.8–23.4 องศาเซลเซียส แสดงเป็นกราฟความสัมพันธ์ระหว่างอุณหภูมิ (องศาเซลเซียส) และช่วงเวลาภายใน 1 ชั่วโมง กราฟแสดงดังรูปที่ 4.11



รูปที่ 4.12 กราฟค่าอุณหภูมิจากเซนเซอร์ DHT22 ภายในระบบในช่วงเวลา 1 ชั่วโมง

ระยะที่สองของการเจริญเติบโตสภาพแวดล้อมที่เหมาะสมคือ อุณหภูมิ 18-19 องศาเซลเซียส ความชื้น 75-95 เปอร์เซ็นต์ และแสงสว่างที่มีความเข้มแสง 500-1000 LUX ซึ่งเป็นระยะที่ใยขาวปกคลุม 80 เปอร์เซ็นต์ของอาหาร และเป็นระยะที่เห็ดต้องได้รับแสงเพื่อออกดอก โดยทำการเปิดไฟให้แสง 6 ชั่วโมง และปิดไฟ 6 ชั่วโมง 2 รอบต่อวัน ซึ่งผลการทดลองระบบควบคุมสภาพแวดล้อม โดยการนำค่าจากฐานข้อมูลมาพล็อตกราฟแสดงเป็นกราฟความสัมพันธ์ระหว่างความชื้น (เปอร์เซ็นต์) ช่วงเวลาภายใน 1 ชั่วโมง กราฟแสดงดังรูปที่ 4.13 เป็นการสร้างเงื่อนไขให้เครื่องสร้างหมอกทำงานเมื่อความชื้นน้อยกว่า 75 เปอร์เซ็นต์ และปิดการทำงานเครื่องสร้างหมอกเมื่อความชื้นมากกว่า 95 เปอร์เซ็นต์ และในส่วนของการให้แสงสว่างผลการทดลอง แสดงเป็นแสดงเป็นกราฟความสัมพันธ์ระหว่างความเข้มแสง (LUX) และช่วงเวลาภายใน 1 ชั่วโมง กราฟแสดงดังรูที่ 4.14

รูปที่ 4.13 กราฟค่าความชื้นจากเซนเซอร์ DHT22 ภายในระบบในช่วงเวลา 1 ชั่วโมง



รูปที่ 4.14 กราฟค่าความความเข็มแสงจากเซนเซอร์ BH1750FVI ภายในระบบในช่วงเวลา 1 ชั่วโมง

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การทำงานของระบบควบคุมสภาพแวดล้อมแบบปิดสำหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง เมื่อนำขวดโหลเห็ดที่มีเชื้อ และอาหารแข็งสำหรับเชื้อเห็ดถั่งเช่าสีทอง เมื่อเริ่มใส่ขวดโหลเห็ดเข้าไปในตู้ควบคุมสภาพแวดล้อม ในระยะแรกที่ไม่ให้เชื้อเห็ดโดนแสง และปรับอุณหภูมิในตู้ประมาณ 22-23 องศาเซลเซียส เป็นระยะเวลาประมาณ 1 เดือน เมื่อถึงระยะนี้จะสังเกตได้ว่าใยสีขาวจะปกคลุมอาหาร ประมาณ 80 เปอร์เซ็นต์ หลังจากนั้น จะเริ่มให้แสงแก่เห็ด โดยจะให้แสงทุก ๆ 6 ชั่วโมง ซึ่งจะตั้งเวลา ให้เปิดไฟ 6:00 น. ถึง 12:00 น. และ 18:00 น. ถึง 24:00 น. เป็นระยะเวลา 1 สัปดาห์ เมื่อถึงระยะเวลาภายในขวดโหลเห็ด จะมีใยสีเหลืองปรากฏขึ้น ขั้นตอนต่อไปจะเป็นการลดอุณหภูมิภายในตู้ให้ลดลงถึง 18 องศาเซลเซียส และเปลี่ยนเวลาการเปิด-ปิดไฟเป็นทุก ๆ 12 ชั่วโมง เป็นเวลา 15 วัน เมื่อครบตามจำนวนวันที่กำหนด ภายในโหลเห็ดจะปรากฏดอกเห็ดถั่งเช่าสีทอง ที่งอกขึ้นมามีสีเหลืองทองสวยงาม โดยที่สภาพของเห็ดในแต่ละช่วงอายุ และค่าสถานะต่าง ๆ ภายในตู้ควบคุมสภาพแวดล้อม สามารถติดตามได้จากเว็บบราวเซอร์

5.2 ข้อเสนอแนะ

จากการออกแบบของระบบพบว่าอุปกรณ์ที่ใช้งานในระบบบางอย่าง ยังมีความสามารถในการใช้งานที่ยังไม่ดีพอ เช่น การทำความเย็นโดยการใช้ Peltier Cooler ที่ไม่สามารถทำความเย็นได้ดีตามความต้องการของการเจริญเติบโตของเห็ดถั่งเช่าสีทอง ในระยะที่เห็ดต้องสร้างตุ่มดอกเห็ด จึงทำให้เห็ดเจริญเติบโตได้ช้า อีกทั้งการทำความเย็นจากการใช้ Peltier Cooler จำเป็นที่จะต้องใช้กระแสไฟที่สูง จึงทำให้ค่าใช้จ่ายในเรื่องของค่าไฟสูงขึ้น หากต้องการนำไปใช้ตามครัวเรือน นอกจากนั้นการทำงานของเซนเซอร์วัดอุณหภูมิ และความชื้น เกิดการอ่านค่าที่ผิดปกติในบางเวลา ทางผู้คณะจัดทำจึงแนะนำให้ใช้อุปกรณ์ที่มีประสิทธิภาพดีกว่านี้ หากต้องการระบบที่สมบูรณ์แบบ

บรรณานุกรม

[1] นพ.สมยศ กิตติมั่นคง. "ถั่งเช่า สรรพคุณและประโยชน์ของถั่งเช่า 26 ข้อ! (เห็ดถั่งเช่า)." https://medthai.com/ถั่งเช่า/.

[2] สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ, คณะเภสัชศาสตร์ มหาวิทยาลัยมหิดล. "ถั่งเช่า สรรพคุณสุดอัศจรรย์ แก้นกเขาไม่ขัน ขุมพลังแห่งการต้านมะเร็ง." https://health.kapook.com/view11023.html/.

[3] สัมพันธ์ พิพัฒน์วรการ. "เห็ดถั่งเช่าสีทอง." กรุงเทพ: เฟรชวิลล์ฟาร์ม, 2560.

[4] อภัย ราษฎรวิจิตร. "Mannitol." http://haamor.com/th/แมนนิทอล/.

5] "[Raspberry Pi] ขั้นตอนการทำ R-Pi ให้เป็น Web Server." http://doc.inex.co.th/r-pi-web-server-installation/

6] วสันต์ คุณดิลกเศวต. "*RASPBERRY PI HOME SERVER.*" http://www.poeclub.org/raspberry-pi-home-server-toc/.

[7] "JBtek 4 Channel DC 5V Relay Module for Arduino Raspberry Pi DSP AVR PIC ARM." https://www.amazon.com/JBtek-Channel-Module-Arduino-Raspberry/dp/B00KTEN3TM/.

[8] Aspencore. "AM2302 / DHT22 Datasheet." http://www.electroschematics.com/11293/am2302-dht22-datasheet/

[9] "DHT22 Temperature and humidity sensor module." https://i-esan.com/shop/arduino/dht22-temperature-and-humidity-sensor-module/.

[10] "GY-302 DIGITAL LIGHT INTENSITY SENSOR MODULE" http://www.robotpark.com/GY-302-Digital-Light-Intensity-Sensor-Module/.

[11] "Datasheet BH1750FVi." http://www.datasheetlib.com/datasheet/398045/bh1750fvi_rohm/download.html/.

[12] "BH1750" http://www.semi-journal.com/สร้างลักซ์มิเตอร์/.

[13] กิตติ แซ่เอี้ยว. “GY-302 BH1750 BH1750FVI Digital Light intensity Sensor Module for Arduino.” https://www.mcucity.com/product/1045/gy-302-bh1750-bh1750fvi-digital-light-intensity-sensor-module-for-arduino/.

[14] II-VI Technologies Beijing, Co., Ltd. “HOW DO THERMOELECTRIC COOLERS (TECS) WORK?” http://www.marlow.com/resources/general-faq/6-how-do-thermoelectric-coolers-tecs-work.html/.

[15] “Ultrasonic Mist Maker Water Fountain Pond Atomizer Air Humidifier No LED.” https://www.amazon.co.uk/Ultrasonic-Maker-Fountain-Atomizer-Humidifier/dp/B00MYOCGNE/.

[16] ทัศนีย์ กรองทอง. “ภาษาไพทอนกับการจัดการเรียนรู้.” https://library.ipst.ac.th/.

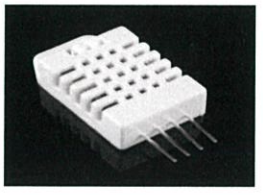
[17] Peerapas Phongeratiyut. “ภาษาPHP.” http://pasaphp.blogspot.com/.

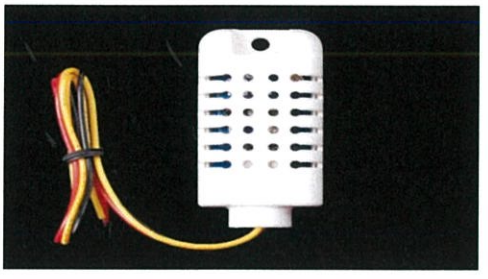
[18] วิรพล สิงหาอาจ. “ความหมายและความสำคัญของภาษา HTML.” https://sites.google.com/a/moeipit.ac.th/wirapons/hnwy-kar-reiyn-ru4/.

[19] ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. “คู่มือการใช้งาน NETPIE.” http://www.innoobec.com/wp-content/uploads/2017/09/NETPIE-WS_v23.pdf/.

ภาคผนวก ก

Datasheet Humidity & Temperature Sensor AM2302/DHT22





Standard AM2302/DHT22 AM2302/DHT22 with big case and wires Digital

relative humidity & temperature sensorAM2302/DHT22

1. Feature & Application:

*High precision
*Capacitive type
*Full range temperature compensated
*Relative humidity and temperature measurement
*Calibrated digital signal *4 pins packaged and fully interchangeable
*Outstanding long-term stability
*Extra components not needed
*Long transmission distance, up to 100 meters
*Low power consumption

2. Description:

AM2302 output calibrated digital signal. It applys exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

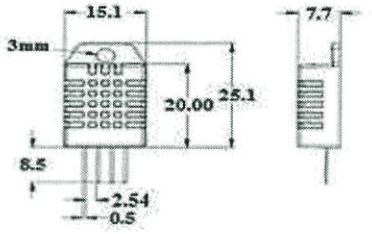
Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(100m) enable AM2302 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

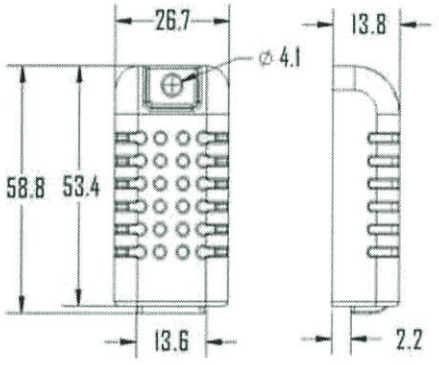
Model	AM2302	
Power supply	3.3-5.5V DC	
Output signal	digital signal via 1-wire bus	
Sensing element	Polymer humidity capacitor	
Operating range	humidity 0-100%RH;	temperature -40~80Celsius
Accuracy	**humidity +-2%RH**(Max +-5%RH);	temperature +-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius

Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH
Long-term Stability	+-0.5%RH/year
Interchangeability	fully interchangeable



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD—power supply
2	DATA—signal
3	GND
4	GND



4.Dimensions: (unit----mm)

Standard AM2302's dimensions as above

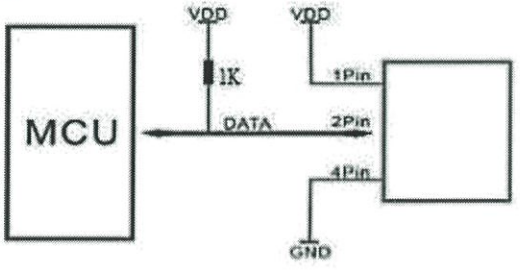
Big case's dimensions as above

Red wire--power supply,

Black wire--GND

Yellow wire--Data output

5.Electrical connection diagram:



6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-5.5V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

1-wire bus is used for communication between MCU and AM2302. (Our 1-wire bus is specially designed, it's different from Maxim/Dallas 1-wire bus, so it's incompatible with Dallas 1-wire bus.) **Illustration of our 1-wire bus:**

DATA=16 bits RH data+16 bits Temperature data+8 bits check-sum

Example: MCU has received 40 bits data from AM2302 as

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

16 bits RH data 16 bits T data check sum

Here we convert 16 bits RH data from binary system to decimal system,

0000 0010 1000 1100 ⟶ 652

Binary system Decimal system

RH=652/10=65.2%RH

Here we convert 16 bits T data from binary system to decimal system,

0000 0001 0101 1111 ⟶ 351

Binary system Decimal system

T=351/10=35.1℃

When highest bit of temperature is 1, it means the temperature is below 0 degree Celsius.

Example: 1000 0000 0110 0101, T= minus 10.1℃

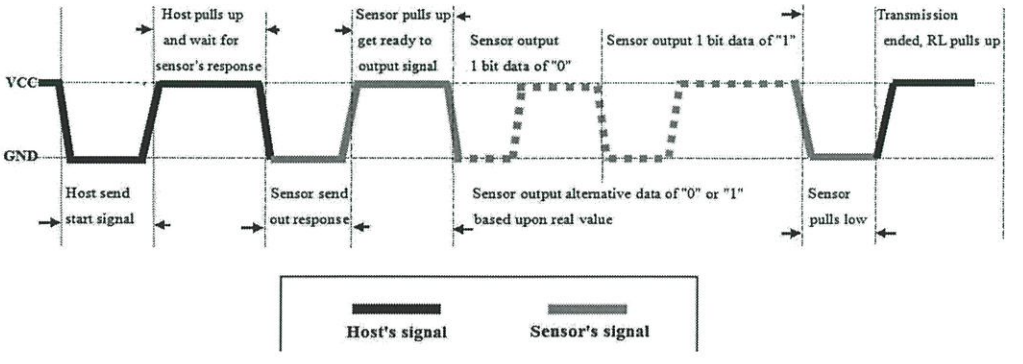
16 bits T data

Sum=0000 0010+1000 1100+0000 0001+0101 1111=1110 1110

Check-sum=the last 8 bits of Sum=1110 1110

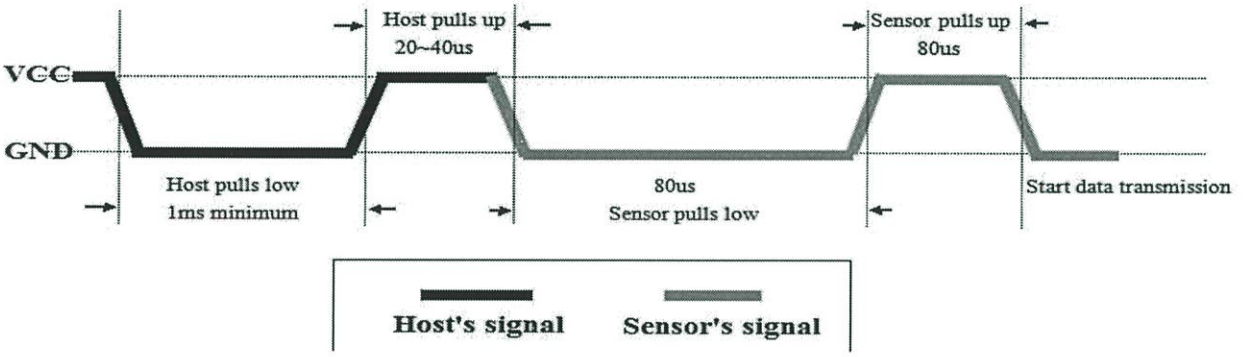
When MCU send start signal, AM2302 change from standby-status to running-status. When MCU finishs sending the start signal, AM2302 will send response signal of 40-bit data that reflect the relative humidity and temperature to MCU. Without start signal from MCU, AM2302 will not give response signal to MCU. One start signal for one response data from AM2302 that reflect the relative humidity and temperature. AM2302 will change to standby status when data collecting finished if it don't receive start signal from MCU again.

See below figure for overall communication process, the interval of whole process must beyond 2 seconds.



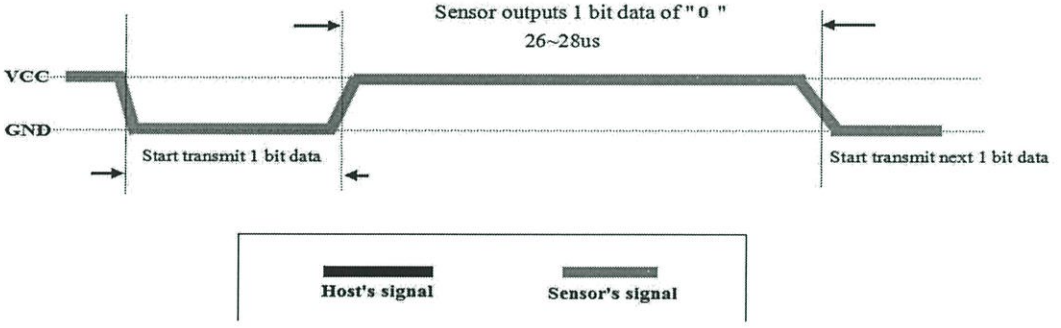
1) Step 1: MCU send out start signal to AM2302 and AM2302 send response signal to MCUData-bus's free status is high voltage level. When communication between MCU and AM2302 begins, MCU will pull low data-bus and this process must beyond at least 1~10ms to ensure AM2302 could detect MCU's signal, then MCU will pulls up and wait 20-40us for AM2302's response.

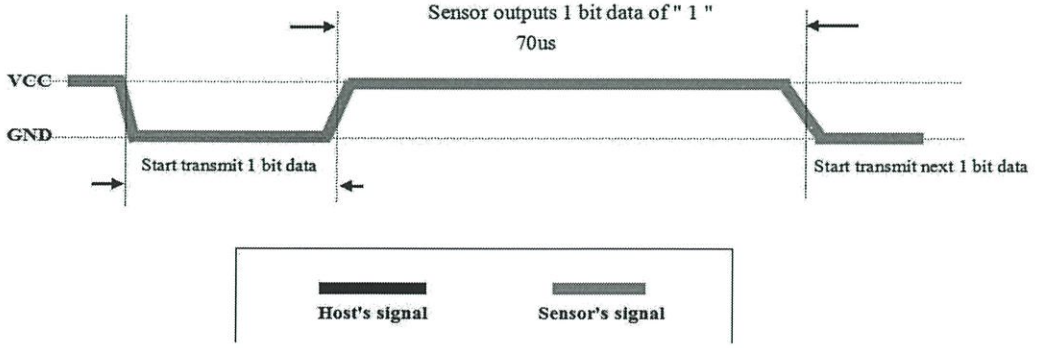
When AM2302 detect the start signal, AM2302 will pull low the bus 80us as response signal, then AM2302 pulls up 80us for preparation to send data. See below figure:



2). Step 2: AM2302 send data to MCU

When AM2302 is sending data to MCU, every bit's transmission begin with low-voltage-level that last 50us, the following high-voltage-level signal's length decide the bit is "1" or "0". See below figures:





ภาคผนวก ข

Datasheet Light Intensity Sensor IC BH1750FVI

ROHM SEMICONDUCTOR

TECHNICAL NOTE

Pb Free ROHM Electronic Components RoHS Directive Compliance

Ambient Light Sensor IC Series

Digital 16bit Serial Output Type Ambient Light Sensor IC

BH1750FVI

● Descriptions

BH1750FVI is an digital Ambient Light Sensor IC for I²C bus interface. This IC is the most suitable to obtain the ambient light data for adjusting LCD and Keypad backlight power of Mobile phone. It is possible to detect wide range at High resolution. (1 - 65535 lx).

● Features

1) I²C bus Interface (f / s Mode Support)
2) Spectral responsibility is approximately human eye response
3) Illuminance to Digital Converter
4) Wide range and High resolution. (1 – 65535 lx)
5) Low Current by power down function
6) 50Hz / 60Hz Light noise reject-function
7) 1.8V Logic input interface
8) No need any external parts
9) Light source dependency is little. (ex. Incandescent Lamp. Fluorescent Lamp. Halogen Lamp. White LED. Sun Light)
10) It is possible to select 2 type of I²C slave-address.
11) Adjustable measurement result for influence of optical window
(It is possible to detect min. 0.11 lx, max. 100000 lx by using this function.)
12) Small measurement variation (+/- 20%) 13) The influence of infrared is very small.

● Applications

Mobile phone, LCD TV, NOTE PC, Portable game machine, Digital camera, Digital video camera, Car navigation, PDA, LCD display

● Absolute Maximum Ratings

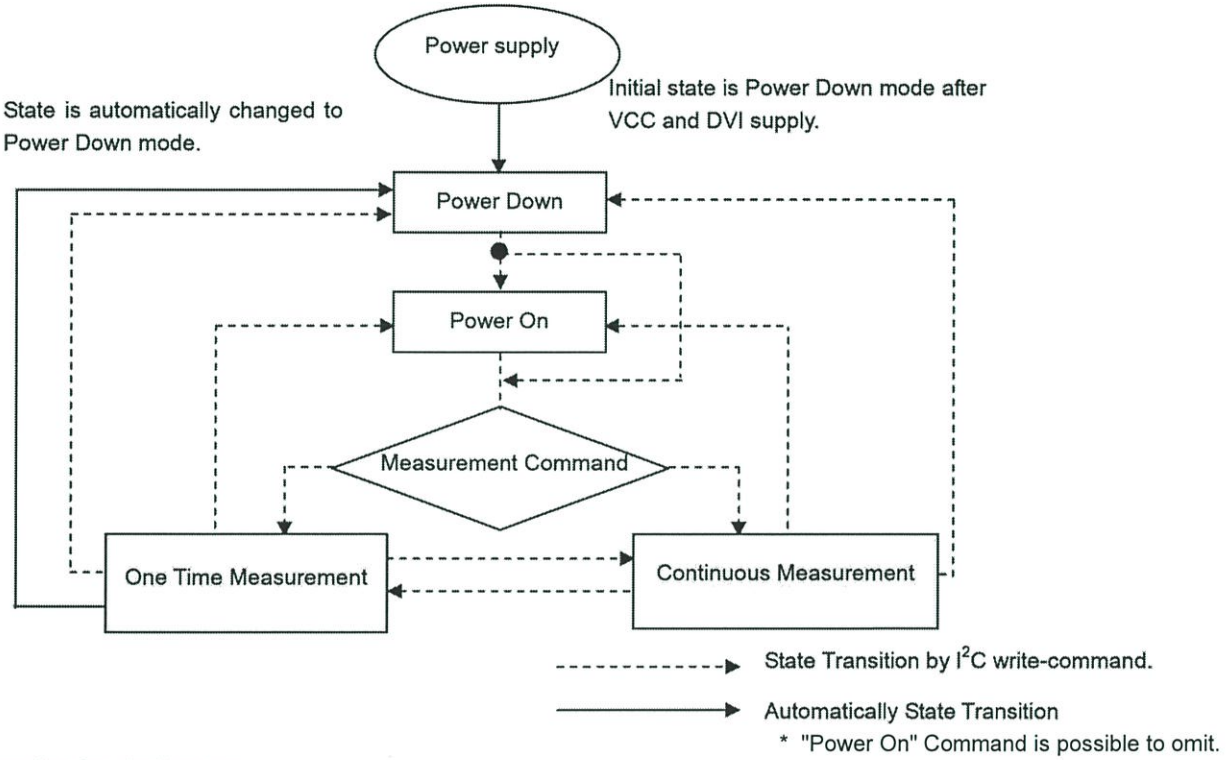
Parameter	Symbol	Limits	Units
Supply Voltage	Vmax	4.5	V
Operating Temperature	Topr	-40～85	℃
Storage Temperature	Tstg	-40～100	℃
SDA Sink Current	Imax	7	mA
Power Dissipation	Pd	260※	mW

※ 70mm × 70mm × 1.6mm glass epoxy board. Derating in done at 3.47mW/℃ for operating above Ta=25℃.

● Operating Conditions

Parameter	Symbol	Min.	Typ.	Max.	Units
VCC Voltage	Vcc	2.4	3.0	3.6	V
I^2C Reference Voltage	VDVI	1.65	–	VCC	V

● Measurement Procedure



Instruction Set Architecture

Instruction	Opecode	Comments
Power Down	0000_0000	No active state.
Power On	0000_0001	Waiting for measurement command.
Reset	0000_0111	Reset Data register value. Reset command is not acceptable in Power Down mode.
Continuously H-Resolution Mode	0001_0000	Start measurement at 1lx resolution. Measurement Time is typically 120ms.
Continuously H-Resolution Mode2	0001_0001	Start measurement at 0.5lx resolution. Measurement Time is typically 120ms.
Continuously L-Resolution Mode	0001_0011	Start measurement at 4lx resolution. Measurement Time is typically 16ms.
One Time H-Resolution Mode	0010_0000	Start measurement at 1lx resolution. Measurement Time is typically 120ms. It is automatically set to Power Down mode after measurement.
One Time H-Resolution Mode2	0010_0001	Start measurement at 0.5lx resolution. Measurement Time is typically 120ms. It is automatically set to Power Down mode after measurement.

One Time L-Resolution Mode	0010_0011	Start measurement at 4lx resolution. Measurement Time is typically 16ms. It is automatically set to Power Down mode after measurement.
Change Measurement time (High bit)	01000_MT[7,6,5]	Change measurement time. ※ Please refer "adjust measurement result for influence of optical window."
Change Masurement time (Low bit)	011_MT[4,3,2,1,0]	Change measurement time. ※ Please refer "adjust measurement result for influence of optical window."

※ Don't input the other opecode.

● Measurement mode explanation

Measurement Mode	Measurement Time.	Resolurtion
H-resolution Mode2	Typ. 120ms.	0.5 lx
H-Resolution Mode	Typ. 120ms.	1 lx.
L-Resolution Mode	Typ. 16ms.	4 lx.

We recommend to use H-Resolution Mode.

Measurement time (integration time) of H-Resolution Mode is so long that some kind of noise(including in 50Hz / 60Hz noise) is rejected. And H-Resolution Mode is 1 l x resolution so that it is suitable for darkness (less than 10 lx) H-resolution mode2 is also suitable to detect for darkness.

● Explanation of Asynchronous reset and Reset command "0000_0111"

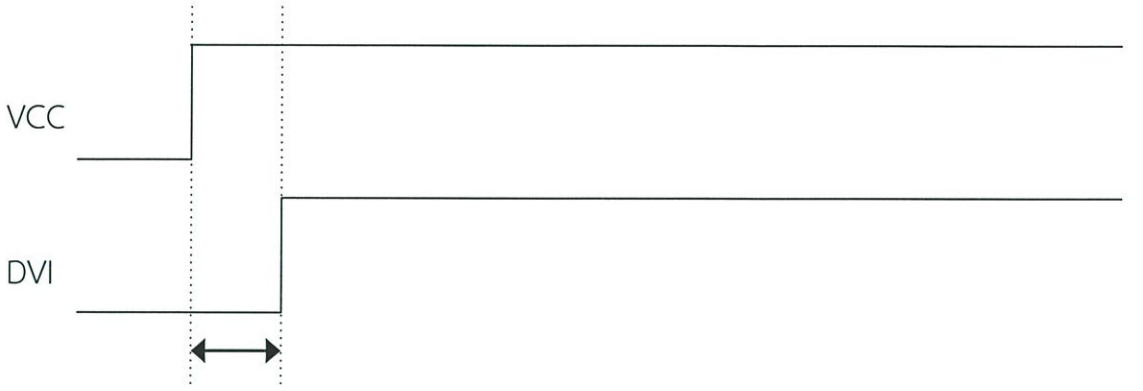
1) Asynchronous reset
   All registers are reset. It is necessary on power supply sequence. Please refer "Timing chart for VCC and DVI power supply sequence" in this page. It is power down mode during DVI = 'L'.

2) Reset command
   Reset command is for only reset Illuminance data register. (reset value is '0') It is not necessary even power supply sequence.It is used for removing previous measurement result. This command is not working in power down mode, so that please set the power on mode before input this command.

Timing chart for VCC and DVI power supply sequence

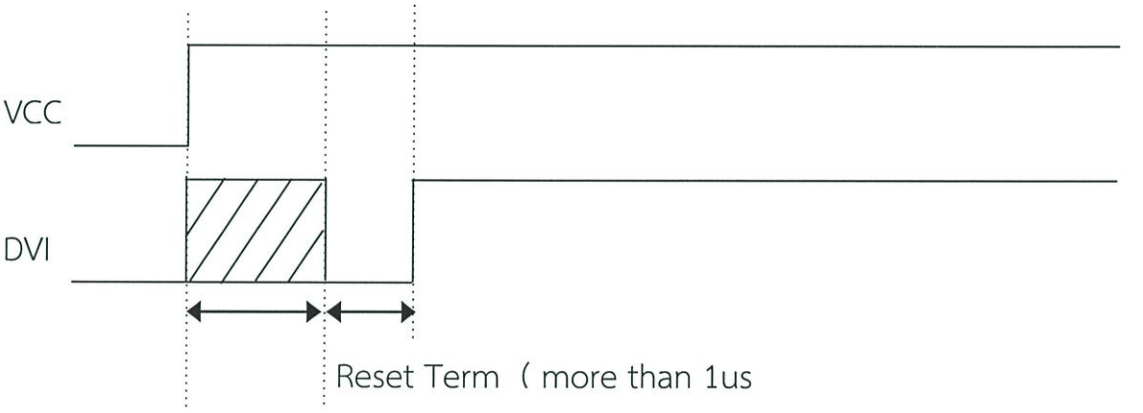
DVI is I^2C bus reference voltage terminal. And it is also asynchronous reset terminal. It is necessary to set to 'L' after VCC is supplied. In DVI 'L' term, internal state is set to Power Down mode.

1） Recommended Timing chart1 for VCC and DVI supply.



2） Timing chart2 for VCC and DVI supply.
3）

If DVI rises within 1us after VCC



Don't care state

ADDR, SDA, SCL is not stable if DVI 'L' term (1us) is not given by systems.

In this case, please connect the resisters (approximately 100kOhm) to ADDR without directly connecting to VCC or GND, because it is 3 state buffer for Internal testing.

Measurement sequence example from "Write instruction" to "Read measurement result"

ex1) Continuously H-resolution mode (ADDR = 'L')

from Master to Slave from Slave to Master

① Send "Continuously H-resolution mode " instruction

ST	0100011	0	Ack	00010000	Ack	SP

② Wait to complete 1st H-resolution mode measurement.(max. 180ms.)

③ Read measurement result.

ST	0100011	1	Ack	High Byte [15:8]	Ack

Low Byte [7:0]	$\overline{\text{Ack}}$	SP

How to calculate when the data High Byte is "10000011" and Low Byte is "10010000"
($2^{15} + 2^9 + 2^8 + 2^7 + 2^4$) / 1.2 ≒ 28067 [lx]

The result of continuously measurement mode is updated.(120ms.typ at H-resolution mode, 16ms.typ at L-resolution mode)

ex2) One time L-resolution mode (ADDR = 'H')

① Send "One time L-resolution mode " instruction

ST	1011100	0	Ack	00100011	Ack	SP

② Wait to complete L-resolution mode measurement.(max. 24ms.)

③ Read measurement result

ST	1011100	1	Ack	High Byte [15:8]	Ack

Low Byte [7:0]	$\overline{\text{Ack}}$	SP

How to calculate when the data High Byte is "00000001" and Low Byte is "00010000"
($2^8 + 2^4$) / 1.2 ≒ 227 [lx]

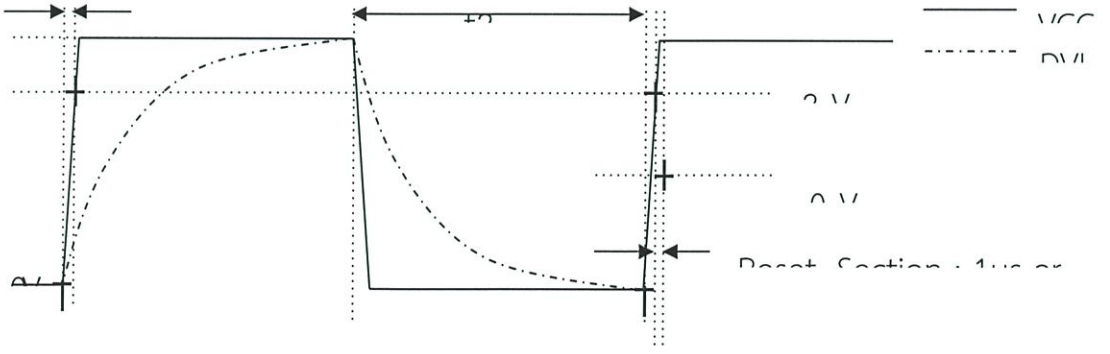
Notes when CR is inserted between VCC and

※ Please note that there is a possibility that reset section (1us) can not be satisfied because the power supply is turned on when the rise time of VCC is slow

※ When VCC is turned off, the DVI voltage becomes higher than VCC voltage but IC destruction is not occred if recommended constant (R1 = 1kOhm, C1 = 1uF) is used.

※ Please note that there is a possibility that Reset section (1usec) cannot be satisfied if wait time is not enough long after turning off VCC.(It is necessary to consider DVI voltage level after turning off VCC.)

t1

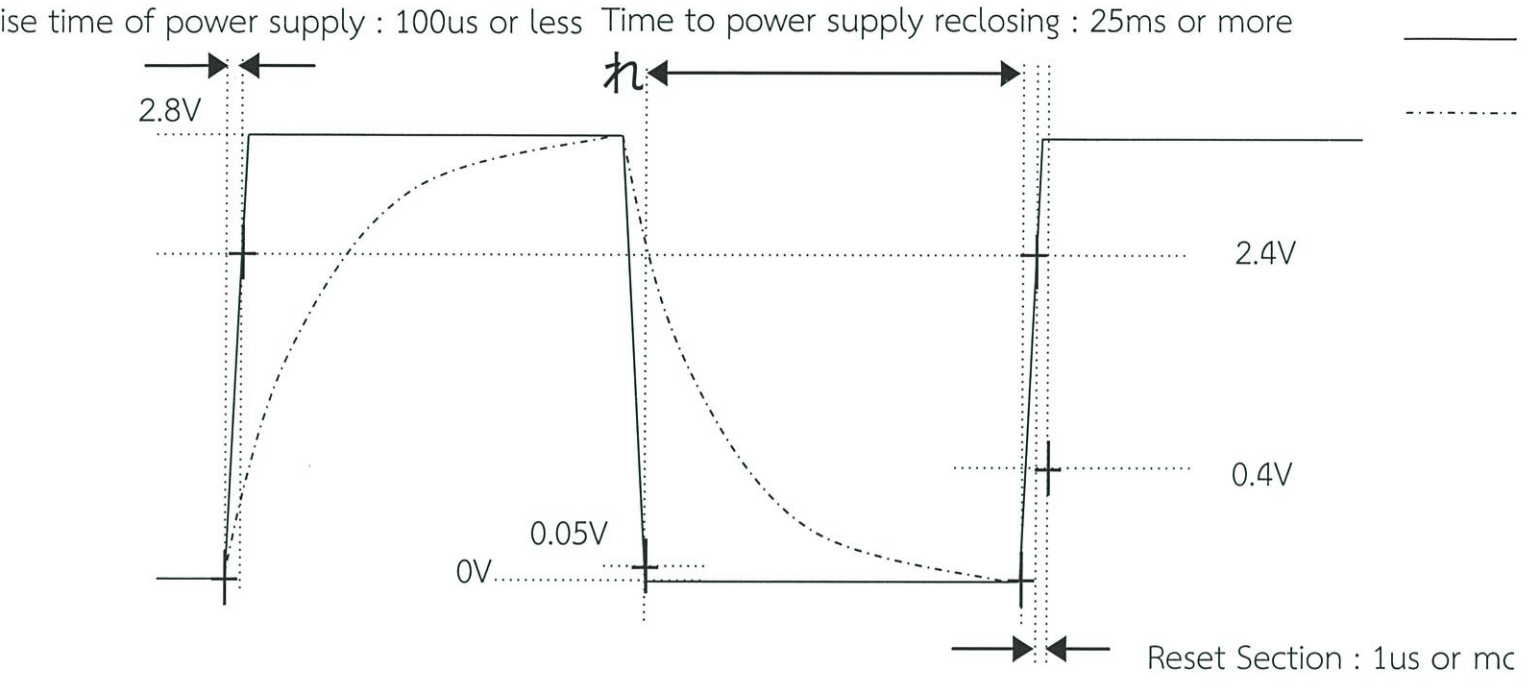


- Please do the application design to secure Reset section 1us or more after the reclosing of the power supply.

□ Example of designing set when CR (C = 1uF, R = 1kΩ) is inserted between VCC and DVI with VCC=2.8V

① The rise time to 0→2.4V of VCC must use the power supply of 100us or less.

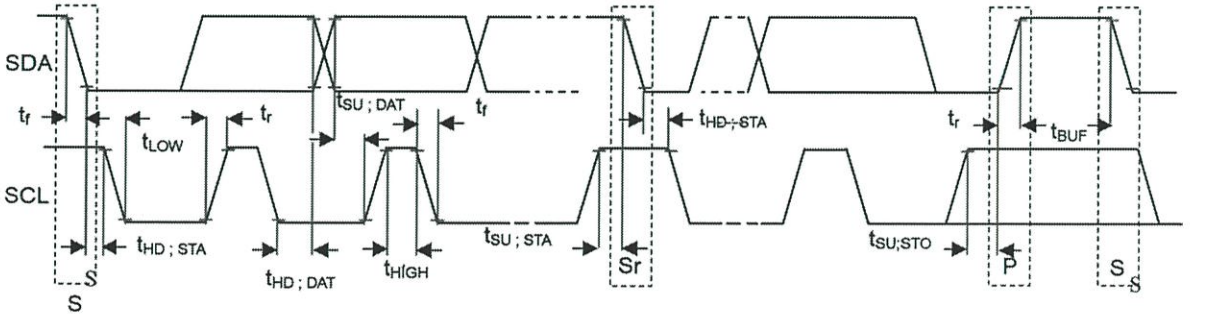
② Please wait 25ms or more after VCC turn off (VCC <= 0.05V), because it is necessary to secure reset section (1us or more).



・ Please do the application design to secure Reset section 1us or more after the reclosing of the power supply. • I²C Bus Access

1) I²C Bus Interface Timing chart

Write measurement command and Read measurement result are done by I²C Bus interface. Please refer the formally specification of I²C Bus interface, and follow the formally timing chart.



2) Slave Address

Slave Address is 2 types, it is determined by ADDR Terminal

ADDR = 'H' (ADDR ≧ 0.7VCC) → "1011100"

ADDR = 'L' (ADDR ≦ 0.3VCC) → "0100011"

3) Write Format

BH1750FVI is not able to accept plural command without stop condition. Please insert SP every 1 Opecode.

ST	Slave Address	R/W 0	Ack	Opecode	Ack	SP

4) Read Format

ST	Slave Address	R/W 1	Ack	High Byte [15:8] 215 214 213 212 211 210 29 28	Ack

Low Byte [7:0] 27 26 25 24 23 22 21 20	Ack	SP

from Slave to Master
from Master to Slave

ex)
High Byte = "1000_0011"
Low Byte = "1001_0000"
($2^{15} + 2^9 + 2^8 + 2^7 + 2^4$) / 1.2 ≒ 28067 [lx] ※2

* I²C BUS is trademark of Phillips Semiconductors. Please refer formality specification.

Adjust measurement result for influence of optical window. (sensor sensitivity adjusting)
BH1750FVI is possible to change sensor sensitivity. And it is possible to cancel the optical window influence (difference with / without optical window) by using this function. Adjust is done by changing measurement time. For example, when transmission rate of optical window is 50% (measurement result becomes 0.5 times if optical window is set), influence of optical window is ignored by changing sensor sensitivity from default to 2 times

Sensor sensitivity is shift by changing the value of MTreg (measurement time regisiter). MTreg value has to set 2 times if target of sensor sensitivity is 2 times. Measurement time is also set 2 times when MTreg value is changed from default to 2 times.

ex) Procedure for changing target sensor sensitivity to 2 times.

Please change Mtreg from "0100_0101" (default) to "1000_1010" (default * 2).

1) Changing High bit of MTreg

ST	Slave Address	R/W 0	Ack	01000_100	Ack	SP

2) Changing Low bit of MTreg

ST	Slave Address	R/W 0	Ack	011_01010	Ack	SP

3) Input Measurement Command

ST	Slave Address	R/W 0	Ack	0001_0000	Ack	SP

* This example is High Resolution mode, but it accepts the other measurement.
4) After about 240ms, measurement result is registered to Data Register. (High Resolution mode is typically 120ms, but measurement time is set twice.)

The below table is seeing the changable range of MTreg.

		Min.	Typ.	Max.
changable range of MTreg	binary	0001_1111 (sensitivity : default * 0.45)	0100_0101 default	1111_1110 (sensitivity : default * 3.68)
	decimal	31 (sensitivity : default * 0.45)	69 default	254 (sensitivity : default * 3.68)

It is possilbe to detect 0.23lx by using this function at H-resolution mode. And it is possilbe to detect 0.11lx by using this function at H-resolution mode2.

The below formula is to calculate illuminance per 1 count.

H-reslution mode : Illuminance per 1 count (lx / count) = 1 / 1.2 *(69 / X)
H-reslution mode2 : Illuminance per 1 count (lx / count) = 1 / 1.2 *(69 / X) / 2

1.2 : Measurement accuracy
69 : Default value of MTreg (dec)
X : MTreg value

The below table is seeing the detail of resolution.

Mtreg の値	lx / count at H-resolutin mode	lx / count at H-resolution mode2
0001_1111	1.85	0.93
0100_0101	0.83	0.42
1111_1110	0.23	0.11

H-Resolution Mode2

H-resolution mode2 is 0.5lx (typ.) resolution mode. It is suitable if under less than 10 lx measure ment data is necessary. This measurement mode supports " Adjust measurement result for influence of optical window ". Please refer it. It is possible to detect min. 0.11 lx by using H-resolution mode2.

○ Instruction set architecture for H-resolution mode2

Instruction	Opecode	Comments
Continuously H-Resolution Mode2	0001_0001	Start measurement at 0.5lx resolution. Measurement Time is typically 120ms.
One Time H-Resolution Mode2	0010_0001	Start measurement at 0.5lx resolution. Measurement Time is typically 120ms. It is automatically set to Power Down mode after measurement.

○ Measurement sequence example from "Write instruction" to "Read measurement result"

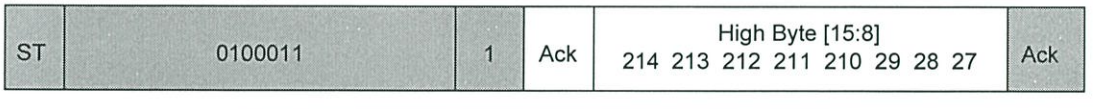
ex) Continuously H-resolution mode2 (ADDR = 'L')

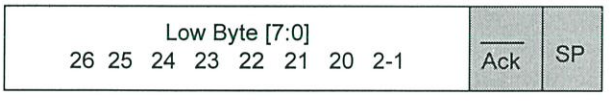
from Master to Slave from Slave to Master

① Send "Continuously H-resolution mode2 " instruction

② Wait to complete 1st H-resolution mode2 measurement.(max. 180ms.)

③ Read measurement result.





How to calculate when the data High Byte is "00000000" and Low Byte is "00010010"

$(2^3 + 2^0) / 1.2 \fallingdotseq 7.5$ [lx]

ภาคผนวก ค

โค้ดการทำงานของระบบควบคุมสภาพแวดล้อมแบบปิดสาหรับเพาะเลี้ยงเห็ดถั่งเช่าสีทอง

```
import microgear.client as client
import logging
import time
import Adafruit_DHT
import smbus
import time
import RPi.GPIO as GPIO
import MySQLdb
import subprocess

subprocess.Popen(["python",'camerastep.py'])

db = MySQLdb.connect(host="localhost",
                     user="root",
                     passwd="0819479947",
                     db="tabledb")
cur = db.cursor()

DEVICE = 0x23
POWER_DOWN = 0x00
POWER_ON = 0x01
RESET = 0x07
ONE_TIME_HIGH_RES_MODE = 0x20
CONTINUOUS_HIGH_RES_MODE_1 = 0x10
CONTINUOUS_HIGH_RES_MODE_2 = 0x11
ONE_TIME_HIGH_RES_MODE_1 = 0x20
ONE_TIME_HIGH_RES_MODE_2 = 0x21
ONE_TIME_LOW_RES_MODE = 0x23
bus = smbus.SMBus(1)
sensor = Adafruit_DHT.DHT22
pin = 4

appid = 'Testsystem'
gearkey = 'mdpqYKnVx06vOXb'
gearsecret = 'paWvucXGadSDK1jJlXQl1Wyru'

client.create(gearkey,gearsecret,appid,{'debugmode': True})
```

```
GPIO.setmode(GPIO.BCM)

pinList17 = [17]
pinList27 = [27]
pinList22 = [22]

GPIO.setup(pinList17, GPIO.OUT)
GPIO.output(pinList17, GPIO.HIGH)
GPIO.setup(pinList27, GPIO.OUT)
GPIO.output(pinList27, GPIO.HIGH)
GPIO.setup(pinList22, GPIO.OUT)
GPIO.output(pinList22, GPIO.HIGH)

def connection():
    logging.info("Now I am connected with netpie")

def subscription(topic,message):
    logging.info(topic+" "+message)

def disconnect():
    logging.debug("disconnect is work")

client.setalias("dataH")
client.setalias("dataT")
client.setalias("dataL")
client.setalias("dataS")
client.on_connect = connection
client.on_message = subscription
client.on_disconnect = disconnect
client.subscribe("/mail")
client.connect(False)

def convertToNumber(data):
    return ((data[1] + (256*data[0]))/1.2)
def readLight(addr=DEVICE):
    data = bus.read_i2c_block_data(addr,ONE_TIME_HIGH_RES_MODE)
```

```
    return convertToNumber(data)w,j
def main():
    while True:
        dt = list(time.localtime())
        month = dt[1]
        day = dt[2]
        hour = dt[3]
        minute = dt[4]
        second = dt[5]
        humidity,temperature = (Adafruit_DHT.read_retry(sensor,pin))
        light = readLight()
        data= {"humidity":humidity,"temperature":temperature,"light":light}
        if (client.connected):
            client.chat("dataH",str(humidity))
            client.chat("dataT",str(temperature))
            client.chat("dataL",str(light))
          if month == month:
              client.chat("dataS","White mycelium cover the food")
         if (day >= 1) & (day <= 7) & (month == month + 1):
              client.chat("dataS","Yellow mycelium cover the food ")
         if (day >= 8) & (day <= 25) & (month == month + 1):
              client.chat("dataS","Yellow stroma appear")

         client.writeFeed("Feedmysensor",data,"GukPr3B3DRUMzNCWGEAoMHKMZU8zmDKQ")
         cur.execute("""INSERT INTO systemdatabase
VALUES(NULL,%s,%s,%s,NULL)""",(temperature,humidity,light))
            db.commit()
            time.sleep(5)

            if humidity < 80 :
                GPIO.output(pinList27, GPIO.LOW)
                GPIO.output(pinList22, GPIO.LOW)
            if humidity > 90 :
                GPIO.output(pinList27, GPIO.HIGH)
                GPIO.output(pinList22, GPIO.HIGH)
        else:
            print 'Failed'
```

```
dt = list(time.localtime())
hour = dt[3]
minute = dt[4]
second = dt[5]

if (month == month + 1) && (day == 1)
    if hour == 6 :
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 7:
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 8 :
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 9:
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 10:
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 11:
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 12:
        GPIO.output(pinList17, GPIO.HIGH)
    elif hour == 18 :
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 19:
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 20 :
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 21:
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 22:
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 23:
        GPIO.output(pinList17, GPIO.LOW)
    elif hour == 0:
        GPIO.output(pinList17, GPIO.HIGH)
        time.sleep(2)
```

```
        if (month == month + 1) && (day == 8)
            if hour == 0 :
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 1:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 2:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 3:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 4:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 5:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 6:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 7:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 8:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 9:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 10:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 11:
                GPIO.output(pinList17, GPIO.LOW)
            elif hour == 12:
                GPIO.output(pinList17, GPIO.HIGH)

        if (month == month + 1) && (day == 25)
            GPIO.output(pinList17, GPIO.HIGH)


    cur.close()

if __name__=="__main__":
    main()
```