



ระบบควบคุมคุณภาพคัลงยาโดยใช้บอร์ด Raspberry Pi

โดย

นางสาวอริรัตน์	วงศ์จันทร์ชมภู	รหัสนักศึกษา	B5513345
นายวัชรพัฒน์	กองนาค	รหัสนักศึกษา	B5538713
นายภาณุวัฒน์	โชติโร	รหัสนักศึกษา	B5539628

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 527499 โครงการวิศวกรรมโทรคมนาคม
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2554
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
ประจำภาคการศึกษาที่ 3 ปีการศึกษา 2558

ระบบควบคุมคุณภาพคลังยาโดยใช้บอร์ด Raspberry Pi

คณะกรรมการสอบโครงการ

.....
(อาจารย์เศรษฐวิทย์ ภูญา)

กรรมการ / อาจารย์ที่ปรึกษาโครงการ

.....
(รองศาสตราจารย์ ดร. พีระพงษ์ อุฑารสกุล)

กรรมการ

.....
(ผู้ช่วยศาสตราจารย์ ดร. ปิยาภรณ์ มีสวัสดิ์)

กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับรายงานโครงการฉบับนี้ เป็นส่วนหนึ่งของการศึกษา
ระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม วิชา 527499 โครงการวิศวกรรมโทรคมนาคม ประจำปี
การศึกษา 2558

โครงการ	ระบบควบคุมคุณภาพคลังยาโดยใช้บอร์ดRaspberry Pi		
จัดทำโดย	นางสาวอริรัตน์ วงศ์จันทร์ชมภู	รหัสนักศึกษา	B5513345
	นายวัชรพัฒน์ กองนาถ	รหัสนักศึกษา	B5538713
	นายภาณุวัฒน์ โชติโรโส	รหัสนักศึกษา	B5539628
อาจารย์ที่ปรึกษา	อาจารย์เศรษฐวิทย์ ภูญาษา		
สาขาวิชา	วิศวกรรมโทรคมนาคม		
ภาคการศึกษาที่	3/2558		

บทคัดย่อ

เนื่องจากคลังยามีมูลค่าสูงและมีการควบคุมคุณภาพคลังยา ปัจจัยที่ส่งผลต่อคุณภาพคลังยา ได้แก่ อุณหภูมิและความชื้น ด้วยมูลค่าของคลังยามีมูลค่าสูงจึงจำเป็นต้องมีระบบรักษาความปลอดภัยเพื่อรายงานไปยังเภสัชกรผู้ควบคุมคลังยาเพื่อให้ทราบว่าบุคคลใดที่เข้าและออกคลังยา เมื่อพิจารณาถึงระบบแจ้งเตือนอุณหภูมิคลังยาและการรักษาความปลอดภัยของคลังยาระบบเดิมมีราคาสูง จึงทำให้ผู้จัดทำโครงการเกิดแนวคิดในการนำเสนอระบบแจ้งเตือนอุณหภูมิคลังยาและการรักษาความปลอดภัยของคลังยาดำรงงบประมาณจำกัดและมีราคาถูก เพื่อให้สามารถใช้งานกับโรงพยาบาลที่มีงบประมาณอยู่อย่างจำกัดได้

ดังนั้นทางผู้จัดทำโครงการจึงนำเสนอการใช้อินเทอร์เน็ต Raspberry Pi ร่วมกับตัวตรวจรู้อุณหภูมิและความชื้น กล้องWebcam และตัวตรวจรู้การเปิด-ปิดประตู สำหรับการควบคุมคุณภาพคลังยา และการรักษาความปลอดภัยของคลังยา

กิตติกรรมประกาศ

โครงการฉบับนี้สามารถสำเร็จลงไปได้ด้วยดี เนื่องจากได้รับความกรุณาจากอาจารย์ที่ปรึกษาอาจารย์เศรษฐวิทย์ ภูญาษา ที่ได้ให้ความช่วยเหลือเกี่ยวกับแนวคิด การดูแลเอาใจใส่ ติดตามงาน ชี้แนะข้อบกพร่อง ตลอดจนช่วยฝึกฝนและให้การสนับสนุนคณะผู้จัดทำให้มีความสามารถในการทำโครงการ และนำเสนอผลงานให้เป็นที่รู้จักและยอมรับได้

ขอขอบพระคุณผู้อำนวยการโรงพยาบาลมหาวิทยาลัยเทคโนโลยีสุรนารี หัวหน้าแผนกเภสัชกรรมโรงพยาบาลมหาวิทยาลัยเทคโนโลยีสุรนารีร่วมไปถึงเจ้าหน้าที่แผนกเภสัชกรรมโรงพยาบาลมหาวิทยาลัยเทคโนโลยีสุรนารีที่ให้ความช่วยเหลือในการใช้สถานที่จริงทำการทดสอบอุปกรณ์ และขอขอบพระคุณคณาจารย์และบุคลากรสาขาวิชาวิศวกรรมโทรคมนาคมทุกท่านที่ให้ความช่วยเหลือแก่คณะผู้จัดทำมาโดยตลอด รวมทั้งพี่บัณฑิตศึกษาปริญญาโทและพี่บัณฑิตปริญญาเอกสาขาวิชาวิศวกรรมโทรคมนาคมและเพื่อนนักศึกษาสาขาวิชาวิศวกรรมโทรคมนาคมทุกคนที่เป็นกำลังใจให้ตลอดมา

คณะผู้จัดทำใคร่ขอขอบพระคุณทุกท่านที่ได้กล่าวมาไว้ ณ ที่นี้ สำหรับส่วนดีของโครงการชิ้นนี้ ขออุทิศแก่อาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้แก่คณะผู้จัดทำ

ผู้จัดทำ

นางสาวอริรัตน์ วงศ์จันทร์ชมภู

นายวัชรพัฒน์ กองนาค

นายภาณุวัฒน์ โชติรโส

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูปภาพ	ฅ
สารบัญตาราง	ซ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตการดำเนินงาน	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	4
2.1 กล่าวนำ	4
2.2 วิธีการเก็บรักษาผลิตภัณฑ์ยา	4
2.2.1 ปัจจัยที่มีผลต่อการสลายตัวของยา	4
2.2.2 คำแนะนำเกี่ยวกับสถานะการเก็บยา	5
2.2.3 การสังเกตความไม่คงตัวของเภสัชภัณฑ์	6
2.2.4 การเก็บรักษาบางชนิดที่สำคัญ	6
2.3 Raspberry Pi (model B+)	7
2.3.1 คุณสมบัติของบอร์ด Raspberry Pi	9
2.3.2 ระบบปฏิบัติการ Raspberry Pi	9
2.3.3 โครงสร้าง Raspberry Pi	10
2.3.4 การควบคุม GPIO ของ Raspberry Pi	11
2.3.5 ภาษา Python	12
2.3.6 การตั้งค่าและการเริ่มต้นใช้งานโปรแกรมบอร์ด Raspberry Pi	14
2.3.7 การเชื่อมต่ออินเทอร์เน็ตผ่าน Wi-Fi ให้บอร์ด Raspberry Pi	22
2.3.8 การ Remote ใช้งานบอร์ด Raspberry Pi	23
2.4 Temperature & Humidity Sensor Pro - DHT22 Features	27

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4.1 คุณสมบัติ Temperature & Humidity Sensor Pro - DHT22 Features	27
2.4.2 โครงสร้าง Temperature & Humidity Sensor Pro - DHT22 Features	28
2.4.3 การทำงาน Temperature & Humidity Sensor Pro - DHT22 Features	28
2.5 I Module – Magnetic Switch	30
2.5.1 คุณสมบัติ i Module – Magnetic Switch	30
2.5.2 คุณลักษณะ i Module – Magnetic Switch	30
2.5.3 การนำไปประยุกต์ใช้งาน	31
2.5.4 โครงสร้าง i Module – Magnetic Switch	31
2.5.5 การใช้งาน i Module – Magnetic Switch	32
2.6 Application LINE	33
2.6.1 คุณสมบัติ Application LINE	34
2.6.2 ระบบปฏิบัติการ Application LINE	35
บทที่ 3 การออกแบบและการจำลอง	40
3.1 กล่าวนำ	40
3.2 การเขียนโปรแกรม	42
3.2.1 การส่งข้อความและรูปภาพ	42
3.2.2 การเปิด-ปิดประตู	44
3.2.3 การวัดค่าอุณหภูมิและความชื้น	45
3.2.4 การทำงานของระบบโดยรวม	47
บทที่ 4 ผลการทดลองและทดสอบการใช้งานจริง	53
4.1 กล่าวนำ	53
4.2 อุปกรณ์การทดสอบชิ้นงาน	53
4.3 ขั้นตอนการทดสอบ	54
4.4 ผลการทดลอง	55
4.5 วิเคราะห์ผลการทดลอง	56
4.6 สรุปผลการทดลอง	57
บทที่ 5 บทสรุปและข้อเสนอแนะ	58
5.1 บทสรุป	58

สารบัญ (ต่อ)

เรื่อง	หน้า
5.2 ปัญหาและอุปสรรค	58
5.3 แนวทางในการแก้ไขปัญหา	58
5.4 ข้อเสนอแนะเพิ่มเติม	59
5.5 สิ่งที่ได้รับจากการทำโครงการ	59
บรรณานุกรม	60
ภาคผนวก	61
ประวัติผู้เขียน	66



สารบัญรูปภาพ

เรื่อง	หน้า
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	4
รูปที่ 2.1 ลักษณะบอร์ด raspberry pi model B+	8
รูปที่ 2.2 Eben Christopher Upton	8
รูปที่ 2.3 โครงสร้างบอร์ด Raspberry Pi	10
รูปที่ 2.4 อุปกรณ์สำหรับ raspberry pi	11
รูปที่ 2.5 คุณสมบัติของขา GPIO	12
รูปที่ 2.6 คอมไพเลอร์ (Compiler)	13
รูปที่ 2.7 อินเตอร์พรีเตอร์ (Interpreter)	13
รูปที่ 2.8 หน้าต่าง Download ระบบปฏิบัติการ Raspbian "Wheezy"	15
รูปที่ 2.9 วิธีการติดตั้งโปรแกรม Win32DiskIma	15
รูปที่ 2.10 วิธีการติดตั้งโปรแกรม Win32DiskImage	16
รูปที่ 2.11 วิธีการติดตั้งโปรแกรม Win32DiskImage	16
รูปที่ 2.12 บอร์ด Raspberry Pi ก็จะเริ่มทำงาน	17
รูปที่ 2.13 วิธีตั้งค่าขยายพื้นที่บน SD Card ให้เต็มความจุ	17
รูปที่ 2.14 วิธีตั้งค่าขยายพื้นที่บน SD Card ให้เต็มความจุ	18
รูปที่ 2.15 วิธีกำหนดรูปแบบการใช้งานระบบปฏิบัติการ	18
รูปที่ 2.16 วิธีกำหนดรูปแบบการใช้งานระบบปฏิบัติการ	19
รูปที่ 2.17 วิธีเปิดใช้งาน โมดูลกล้อง	19
รูปที่ 2.18 วิธีเปิดใช้งาน โมดูลกล้อง	20
รูปที่ 2.19 วิธีออกจากหน้าเมนูการตั้งค่าระบบและ Reboot ระบบใหม่	20
รูปที่ 2.20 วิธีออกจากหน้าเมนูการตั้งค่าระบบและ Reboot ระบบใหม่	21
รูปที่ 2.21 หน้า Desktop ระบบปฏิบัติการ Raspbian	21
รูปที่ 2.22 โปรแกรม Wi-Fi config	22
รูปที่ 2.23 การค้นหาเครือข่าย Wi-Fi	22
รูปที่ 2.24 การกรอกรหัสผ่านเชื่อมต่อ Wi-Fi	23
รูปที่ 2.25 หน้ายืนยันการเปิดใช้งาน SSH	24
รูปที่ 2.26 การเปิดโปรแกรม Remote Desktop Connection	24

สารบัญรูปภาพ (ต่อ)

เรื่อง	หน้า
รูปที่ 2.27 การ Remote เข้าบอร์ด Raspberry Pi	25
รูปที่ 2.28 ยืนยันการเชื่อมต่อ Remote	25
รูปที่ 2.29 หน้า Log in xrdp	26
รูปที่ 2.30 หน้า Desktop ระบบปฏิบัติการ Raspbian ผ่าน Remote Desktop Connection	26
รูปที่ 2.31 ลักษณะตัวเซนเซอร์ DHT22	27
รูปที่ 2.32 ขาสัญญาณและการเชื่อมต่อของ DHT22	28
รูปที่ 2.33 กราฟแสดงการทำงานของเซนเซอร์ Temperature & Humidity	29
รูปที่ 2.34 ตัวอย่างการคำนวณหาค่า RH	30
รูปที่ 2.35 ลักษณะตัวเซนเซอร์ i Module – Magnetic Switch	31
รูปที่ 2.36 ขาสัญญาณและการเชื่อมต่อของ i Module – Magnetic Switch	32
รูปที่ 2.37 การเชื่อมต่อฮาร์ดแวร์ของ i Module – Magnetic Switch	32
รูปที่ 2.38 การเขียนโปรแกรมของ i Module – Magnetic Switch	33
รูปที่ 2.39 สัญลักษณ์ของ Application LINE	34
รูปที่ 2.40 แสดงหน้าจอการสนทนาด้วยเสียง	35
รูปที่ 2.41 แสดงหน้าจอการส่งข้อความแบบวิดีโอและเสียง	36
รูปที่ 2.42 แสดงหน้าจอการสนทนาแบบกลุ่ม	36
รูปที่ 2.43 แสดงหน้าจอการสนทนาด้วยสติ๊กเกอร์	37
รูปที่ 2.44 แสดงหน้าจอการปรับแต่งภาพพื้นหลัง	37
รูปที่ 2.45 แสดงหน้าจอ Timeline	38
รูปที่ 2.46 แสดงวิธีการเพิ่มเพื่อน	38
บทที่ 3 การออกแบบและการจำลอง	40
รูปที่ 3.1 โครงสร้างการทำงานโดยรวม	40
รูปที่ 3.2 ขั้นตอนการทำงานของระบบโดยรวม	41
รูปที่ 3.3 ค่าที่ได้จากการวัดอุณหภูมิและความชื้นจากเซนเซอร์	44
บทที่ 4 ผลการทดลองและทดสอบการใช้งานจริง	50
รูปที่ 4.1 โมเดลจำลองสำหรับการวัดค่าอุณหภูมิและความชื้น	50
รูปที่ 4.2 รูปแบบการต่อวงจร	51
รูปที่ 4.3 ผลการทดลองที่ส่งเข้าแอปพลิเคชัน LINE	52

สารบัญตาราง

เรื่อง	หน้า
บทที่ 1 บทนำ	1
ตารางที่ 1.1 ตารางการดำเนินงาน	2
บทที่ 4 ผลการทดลองและทดสอบการใช้งานจริง	50
ตารางที่ 4.1 ตารางแสดงผลการทดลองบน Serial Monitor และที่เครื่อง DIGICON TH-02	53



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันคลังขามีมูลค่าสูงและมีการควบคุมคุณภาพคลังยา ปัจจัยที่ส่งผลต่อคุณภาพคลังยา ได้แก่ อุณหภูมิและความชื้น ด้วยมูลค่าของคลังขามีมูลค่าสูงจำเป็นต้องมีระบบรักษาความปลอดภัยเพื่อรายงานไปยังเภสัชกรผู้ควบคุมคลังยา เมื่อพิจารณาถึงระบบแจ้งเตือนอุณหภูมิคลังยาและการรักษาความปลอดภัยของคลังขาระบบเดิมมีราคาสูง จึงทำให้ผู้จัดทำโครงการเกิดแนวคิดในการนำเสนอระบบแจ้งเตือนอุณหภูมิคลังยาและการรักษาความปลอดภัยของคลังยาด้วยงบประมาณจำกัดและมีราคาถูก เพื่อให้สามารถใช้งานกับโรงพยาบาลที่มีงบประมาณอยู่อย่างจำกัดได้

โครงการนี้จึงได้มีศึกษาค้นคว้าเกี่ยวกับเรื่องการใช้บอร์ด Raspberry Pi ร่วมกับตัวตรวจรู้ อุณหภูมิและความชื้น กล้องWebcam และตัวตรวจรู้การเปิดปิดประตู สำหรับการควบคุมคุณภาพคลังยาและการรักษาความปลอดภัยของคลังยา

1.2 วัตถุประสงค์

1. เพื่อรายงานอุณหภูมิและความชื้นของคลังยาให้แก่เภสัชกรผู้ควบคุมคลังยาและเจ้าหน้าที่ที่มีความเกี่ยวข้อง
2. เพื่อให้เกิดระบบควบคุมคุณภาพคลังยาที่สามารถรายงานอุณหภูมิและความชื้น ได้อย่างทันทั่วถึงที่เภสัชกรผู้ควบคุมคลังยา
3. เพื่อให้เภสัชกรผู้ควบคุมคลังยาสามารถรู้ได้ว่ามีบุคคลใดเข้า-ออกคลังยา

1.3 ขอบเขตของการดำเนินงาน

1. ระบบที่นำเสนอสามารถรายงานอุณหภูมิและความชื้นของคลังยาได้ และสามารถปรับปรุงแก้ไขให้สอดคล้องกับความต้องการของเภสัชกรผู้ควบคุม
2. ระบบที่นำเสนอสามารถรายงานบุคคลผู้เข้าและออกคลังยาได้
3. ระบบที่นำเสนอสามารถรายงานอุณหภูมิและความชื้นเมื่อเกินขีดจำกัดที่เภสัชกรผู้ควบคุมได้ตั้งไว้

1.4 ขั้นตอนดำเนินงาน

1. ปรึกษาอาจารย์ที่ปรึกษาโครงการเกี่ยวกับขอบเขตของโครงการที่จะทำ
2. ศึกษาข้อมูลและอุปกรณ์ที่เกี่ยวข้องกับการทำโครงการ
3. ศึกษาทฤษฎีและหลักการการทำงานของเซนเซอร์และบอร์ด Raspberry Pi
4. ศึกษาการใช้งานโปรแกรม Advanced IP Scanner โปรแกรม PUTTY และ โปรแกรม

Raspberry Pi (Remote Desktop Connection)

5. ออกแบบอุปกรณ์ต้นแบบ
6. สร้างอุปกรณ์ต้นแบบ
7. ทดสอบและวัดผลการทำงานของอุปกรณ์ต้นแบบ
8. สรุปผลการทดลอง เขียนรายงาน โครงการ และนำเสนอโครงการ

กิจกรรม	พ.ศ. 2558						
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ศึกษาค้นคว้าหาข้อมูล	↔						
2. เขียนโครงการและเสนอโครงการกับอาจารย์ที่ปรึกษา		↔					
3. ศึกษาการใช้งานของบอร์ด Raspberry Pi เพื่อใช้ในการเขียนโค้ดคำสั่ง			↔				
4. หาซื้ออุปกรณ์ที่ใช้ในการทำโครงการ				↔			
5. เขียนโค้ดคำสั่งเพื่อให้ตัวตรวจรู้ทำงานเพื่อให้ได้ตามวัตถุประสงค์					↔		
6. ทดสอบการทำงานของตัวตรวจรู้และบอร์ด Raspberry Pi					↔		
7. จัดทำรูปเล่มรายงาน และสื่อที่ใช้ในการนำเสนอ พร้อมทั้งเตรียมตัวนำเสนอชิ้นงาน							↔

ตารางที่ 1.1 ตารางการดำเนินงาน

1.5 ผลที่คาดว่าจะได้รับ

1. เกสัชกรได้รับความสะดวกในการตรวจสอบข้อมูลแบบ Real Time
2. สามารถควบคุมคุณภาพยาให้อยู่ภายใต้อุณหภูมิที่เหมาะสมได้หลังจากที่ได้รับข้อความแจ้งเตือนทั้งความชื้นและอุณหภูมิ
3. เกสัชกรสามารถตรวจสอบบุคคลที่เข้า-ออกคลังยาได้



บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 กล่าวนำ

เนื้อหาในบทนี้จะกล่าวถึงการเขียนโปรแกรมและการตั้งค่าโปรแกรม Raspberry Pi ด้วยภาษา Python การใช้งานเซนเซอร์ Temperature & Humidity Sensor Pro - DHT22 Features ในการบอกค่าอุณหภูมิและความชื้นสัมพัทธ์ การใช้งานเซนเซอร์ I Module – Magnetic Switch ในการบอกสถานะการทำงานของประตูโดยการแสดงผลผ่านทาง Application LINE และวิธีการเก็บรักษาผลิตภัณฑ์ยาในคลังยา

2.2 วิธีการเก็บรักษาผลิตภัณฑ์ยา

ผลิตภัณฑ์ยาที่เก็บรักษาไม่ถูกวิธีอาจทำให้ยาเกิดการสลายตัวและเกิดการเปลี่ยนแปลงได้ ซึ่งจะอันตรายอย่างมากต่อผู้ที่ได้รับยา ไม่ว่าจะเป็นการเปลี่ยนแปลงจากการสลายตัวทางเคมีของตัวยาที่ทำให้ปริมาณยาในเภสัชภัณฑ์ลดลง สารที่เกิดจากการสลายตัวของยามีพิษมากกว่ายา เช่น เตตราซัยคลิน ความไม่คงตัวของยาทำให้ประสิทธิภาพของยาลดลง

2.2.1 ปัจจัยที่มีผลต่อการสลายตัวของยา

1. อุณหภูมิ การเก็บยาจะมีวิธีการเก็บที่อุณหภูมิแตกต่างกัน ควรสังเกตข้อแนะนำที่บ่งไว้ในเอกสารยาหรือที่ขวดยาว่าต้องเก็บในสภาวะใด โดยเฉพาะยาที่ต้องเก็บในตู้เย็น เนื่องจากยาบางอย่างสลายตัวได้เร็วมากหากเก็บไว้ในที่อุณหภูมิห้อง ดังนั้นการเก็บยาควรเก็บตามสภาวะที่แนะนำอย่างเคร่งครัด
2. ความชื้น ทำให้ยารูปของแข็งไม่คงตัว และสารอื่นๆ ในตำรับยาอาจเป็นแหล่งของความชื้นได้ เช่น แป้งข้าวโพด เจลาติน sorbital เป็นต้น
3. แสง อาจเร่งให้ยาสลายตัวเร็วหลีกเลี่ยงโดยใช้ภาชนะสีชาซึ่งสามารถดักรังสี UV ได้

2.2.2 คำแนะนำเกี่ยวกับสภาวะการเก็บยา

1) Freezer คือ สถานที่ที่มีการควบคุมอุณหภูมิไว้ระหว่าง -20 องศาเซลเซียส ถึง -10 องศาเซลเซียส

2) Cold คือ อุณหภูมิไม่เกิน 8 องศาเซลเซียส

3) Refrigerator คือ สถานที่เย็นที่มีการควบคุมอุณหภูมิระหว่าง 2 องศาเซลเซียส ถึง 8 องศาเซลเซียส

4) Cool คือ อุณหภูมิระหว่าง 8 องศาเซลเซียส ถึง 15 องศาเซลเซียส ยาเตรียมที่ระบุให้เก็บไว้ใน cool place อาจเก็บไว้ในตู้เย็นได้ ยกเว้นกรณีระบุเป็นอย่างอื่น

5) Room temperature คือ อุณหภูมิในสภาวะการทำงานปกติ

6) Controlled room temperature คือ อุณหภูมิระหว่าง 20 องศาเซลเซียส ถึง 25 องศาเซลเซียส ยาเตรียมที่มีการระบุให้เก็บที่อุณหภูมิไม่เกิน 25 องศาเซลเซียส อาจเก็บใน cool place ได้นอกจากจะระบุเป็นอย่างอื่น

7) Warm คือ อุณหภูมิระหว่าง 30 องศาเซลเซียส ถึง 40 องศาเซลเซียส

8) Excessive heat คือ อุณหภูมิสูงกว่า 40 องศาเซลเซียส

9) Protection from freezing จะระบุในกรณีที่ภาชนะบรรจุมีการเสี่ยงต่อการแตก หรือยาเตรียมที่อาจสูญเสียความแรง หรือมีการทำลายลักษณะของยาเตรียม เมื่อเก็บในสภาวะที่แช่แข็ง เช่น วัคซีนบางชนิด tetanus toxoid เป็นต้น

10) Protection from moisture ให้เก็บในที่สภาวะความชื้นไม่เกิน 60% ในกรณีที่ไม่ระบุสภาวะในการเก็บ ควรเก็บโดยป้องกันความชื้น การแช่แข็ง และเก็บในอุณหภูมิไม่เกิน 40 องศาเซลเซียส

นอกจากการเก็บยาตามสภาวะของการเก็บรักษา ไม่ได้หมายความว่ายาจะมีความคงตัวได้ตลอดไป ให้ดูวันหมดอายุที่ฉลากยาด้วย โดยบริษัทผู้ผลิตยาจะเป็นผู้ระบุไว้ที่ฉลากยา โดยอาจใช้คำว่า expiration date หรือ beyond-use date หรือคำอื่น ๆ ที่มีความหมายคล้ายกัน หากไม่มีการระบุวันหมดอายุให้คิดจากวันผลิต โดยทั่วไปยาเม็ดจะนับวันหมดอายุจากวันผลิต 5 ปี ยาน้ำประมาณ 1 ปี

2.2.3 การสังเกตความไม่คงตัวของเภสัชภัณฑ์

โดยทั่วไปจะไม่สามารถตรวจสอบการสลายทางเคมีได้ แต่การสลายทางเคมีที่เกิดขึ้นในปริมาณมากจะส่งผลให้เกิดการเปลี่ยนแปลงทางกายภาพได้ ให้ตั้งสมมุติฐานว่ายาดังกล่าวมีการสลายตัวทางเคมี เช่น เกิดการเปลี่ยนแปลงของสี กลิ่น มีการตกตะกอน เป็นต้น

การเปลี่ยนแปลงทางกายภาพของยาที่พบบ่อยและบ่งถึงความไม่คงสภาพของยา เช่น เภสัชภัณฑ์ในรูปของแข็งอาจสังเกตได้จากการที่มีไอน้ำหรือหยดน้ำหรือยาจัดเป็นก้อนภายในภาชนะบรรจุ capsule ที่ห่อหุ้มเปลี่ยนแปลง เช่น อาจนิ่มติดกัน หรือแข็งและแตกแม้มีแรงกดอ่อนๆ หรือเม็ดยาเปลี่ยนสี มีรอยค่าง เม็ดยาเกาะติดกัน เป็นต้น ส่วนเภสัชภัณฑ์ที่เป็นรูปของเหลว อาจสังเกตได้จากความไม่คงตัวของสารละลาย มีการตกตะกอนแยกชั้น หรือยาน้ำแขวนตะกอนไม่สามารถแขวนลอยได้ หลังจากเขย่าขวด มีการเปลี่ยนแปลงของสี กลิ่น สำหรับยาน้ำใส (solution, elixir, syrups) จะมีการขุ่นและมีการเจริญของเชื้อหรือเกิดก๊าซ เภสัชภัณฑ์กึ่งแข็งกึ่งเหลว เช่น ครีม ยาขี้ผึ้ง ยาเหน็บ จะมีลักษณะของ สี กลิ่น หรือความหนืดที่เปลี่ยนไป

2.2.4 การเก็บรักษายาบางชนิดที่สำคัญ

1) การเก็บรักษาภายใต้สภาวะที่กำหนด

สภาวะการเก็บยาที่แนะนำมักจะระบุไว้บนฉลาก โดยอาจกำหนดช่วงอุณหภูมิหรือสถานที่หรือเงื่อนไขในการเก็บเภสัชภัณฑ์ เช่น “เก็บในตู้เย็น” “เก็บที่อุณหภูมิห้อง” ควรปฏิบัติตามคำแนะนำเสริมด้วย เช่น “เก็บยาให้พ้นแสง” ถ้าเภสัชภัณฑ์ที่ต้องเก็บให้พ้นแสงบรรจุในภาชนะใสหรือโปร่งแสงแล้วใส่ในกล่องทึบแสง ผู้ป่วยไม่ควรแกะกล่องทิ้งจนกว่าจะใช้ยาหมดหรือทิ้งยา ถ้าไม่ระบุสภาวะในการเก็บอย่างเฉพาะเจาะจง ควรเก็บเภสัชภัณฑ์นั้นในอุณหภูมิห้องที่ควบคุม (controlled room temperature, 20-25 °C) ควรหลีกเลี่ยงเภสัชภัณฑ์จากบริเวณที่ร้อน เย็น หรือมีแสงมากเกินไป หรือมีการเปลี่ยนแปลงอุณหภูมิมากเกินไป

ยาที่แบ่งบรรจุควรเก็บในสิ่งแวดล้อมที่ควบคุมความชื้นและตามอุณหภูมิที่กำหนดในเภสัชตำรับหรือตามฉลาก ในกรณีที่ไม่มีระบุอุณหภูมิและความชื้น ควรเก็บที่อุณหภูมิห้อง (controlled room temperature) แต่ไม่เกิน 23 °C ที่ความชื้นสัมพัทธ์ไม่มากกว่า 75% ในตู้เย็นและในช่องแช่แข็งไม่ถือเป็นบริเวณที่ควบคุมความชื้น ถ้ายานั้นถูกกำหนดให้เก็บในที่เย็นจัด (cold temperature) ให้เก็บยาในตู้เย็นหรือช่องแช่แข็งโดยใส่ยาในภาชนะบรรจุชั้นนอก (outer container) ที่เข้ามาตรฐานที่บังคับไว้สำหรับยานั้นๆ ได้แก่

1. อุณหภูมิ เกษษณกัณฑ์หลายชนิดต้องเก็บในที่เย็น โดยเฉพาะที่อุณหภูมิต่ำกว่า 15 °C เช่นยาเหน็บที่ต้องหลอมละลายที่อุณหภูมิของร่างกาย ยาเตรียมเหล่านี้อาจเสียได้ถ้าวางไว้ในที่ร้อน ยาฉีดอินซูลินควรเก็บที่อุณหภูมิ 2 – 8 (หรือ 10) °C คือในตู้เย็นแต่ไม่ใช่ช่องแช่แข็ง
2. ยาปฏิชีวนะในรูปแบบยาน้ำแขวนตะกอน ผลิตภัณฑ์ที่มีน้ำมันหอมระเหย และครีมบางชนิดต้องเก็บในที่เย็น
3. ยาที่มีไซรับในปริมาณสูงหรือมีปัญหาการละลายควรหลีกเลี่ยงการเปลี่ยนแปลงอุณหภูมิ
4. ความชื้น ยาในรูปแบบของแข็งควรป้องกันความชื้น ผลิตภัณฑ์เหล่านี้ควรบรรจุในภาชนะที่ป้องกันอากาศและความชื้นได้ และควรชี้แจงให้ผู้ป่วยปิดฝาหลังจากใช้ยา ยาผงที่แบ่งบรรจุในกล่อง/ห่อกระดาษควรเก็บในที่แห้ง
5. แสง ขวดสีชาใช้ป้องกันผลิตภัณฑ์ที่ไวต่อแสง การใส่กล่องช่วยเพิ่มการป้องกันแสงได้ สารบางชนิดเช่น paraldehyde ต้องเก็บในที่มืดสนิท ภาชนะบรรจุไม่ควรสัมผัสแสงอาทิตย์โดยตรงแม้จะเป็นภาชนะป้องกันแสงก็ตาม
6. การติดไฟ ผลิตภัณฑ์ที่มีตัวทำละลายอินทรีย์ที่ติดไฟได้ควรมีฉลาก “ติดไฟได้ เก็บให้ห่างจากเปลวไฟ” ตัวอย่าง ยาเตรียม Salicylic Acid Lotion BP ที่ใช้กับหนังสือควรแนะนำให้หลีกเลี่ยงการเป่าผมให้แห้งในบริเวณใกล้ไฟหรือเปลวไฟ

2.3 Raspberry Pi (model B+)

Raspberry Pi เป็นคอมพิวเตอร์ในบอร์ดเดียว (Single Board Computer) ขนาดเล็กเท่าบัตรเครดิต พัฒนาโดยทีมของ Raspberry Pi Foundation สหรัฐอเมริกา ภายใต้การทางานขององคัไม่หวังผลกำไร สร้างขึ้นเพื่อส่งเสริมการเรียนการสอนพื้นฐานวิทยาศาสตร์คอมพิวเตอร์ในสาขา Computer Science ปัจจุบัน Raspberry Pi ผลิตออกมา 4 แบบดังนี้ model A+, model A, model B+ และ model B ซึ่งสามารถรองรับระบบปฏิบัติการลินุกซ์และยูนิกซ์ที่หลากหลาย การจัดหาหน่วย Raspberry Pi ลิขสิทธิ์ร่วมระหว่างบริษัท Newark elemental 4 (Premier Famell), บริษัท RS Components และบริษัท Egoman สำหรับจำหน่ายออนไลน์



รูปที่ 2.1 ลักษณะบอร์ด raspberry pi model B+

วิศวกรผู้สร้างคือ Eben Christopher Upton ทำงานที่บริษัท Broadcom ในตำแหน่ง Technical Director และ ASIC Architect รับผิดชอบระบบซอฟต์แวร์ทั้งหมดและสถาปัตยกรรมฮาร์ดแวร์ของอุปกรณ์ Raspberry Pi เดิมสร้าง Raspberry Pi นี้มาเพื่อสอนในสาขา Computer Science



รูปที่ 2.2 Eben Christopher Upton

ตัวบอร์ด Raspberry Pi ใช้ชิป SoC ของ Broadcom BCM2835 ซึ่งบรรจุ ARM1176JZF-S (ARMv6k) ทำงานที่ความถี่สัญญาณนาฬิกา 700MHz ใช้หน่วยประมวลผลกราฟิก VideoCore IV GPU, หน่วยความจำ 512 MB ผู้สร้าง Raspberry Pi ได้ใช้ Debian และ Arch Linux ARM และวางแผนให้สามารถทำงานในภาษาไพธอนเป็นภาษาหลักในการควบคุม Raspberry Pi นอกจากนั้นยังเขียนภาษา

อื่นเพื่อสั่งการทำงานได้ เช่น ภาษาเบสิก BBC Basic บนระบบปฏิบัติการ RISC OS, ภาษาซี, ภาษา Perl ได้อีกด้วย

2.3.1 คุณสมบัติ Raspberry Pi (model B+)

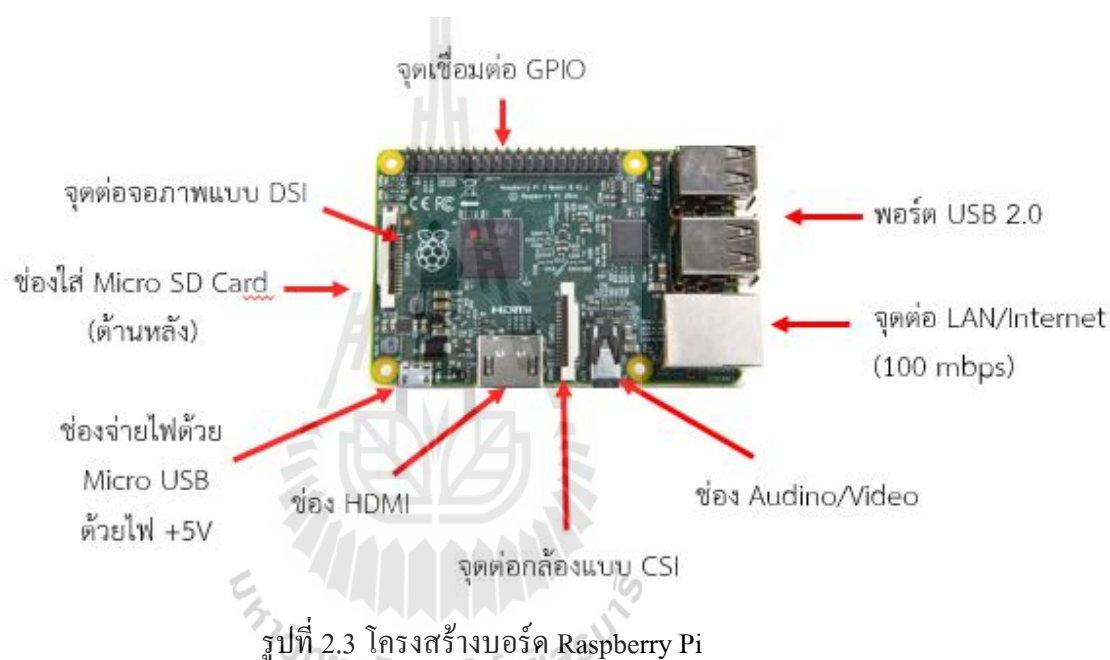
- 1) Chip ควบคุมหลัก Broadcom BCM2835 เทียบเท่าซึ่งรวม CPU, หน่วยประมวลกราฟฟิก หรือ GPU และหน่วยความจำ SD RAM ไว้ภายในตัวเดียวกัน
- 2) หน่วยประมวลผลกลางหรือ CPU ARM11 Core ARM1176JZF-S ความเร็ว 700MHz
- 3) หน่วยประมวลกราฟฟิกหรือ GPU Broadcom Video core IV หรือเทียบเท่ารองรับการ
- 4) แสดงผลผ่านจอภาพที่ใช้จุดต่อแบบ HDMI
- 5) หน่วยความจำ SDRAM 512 MB
- 6) USB 2.0 (2 พอร์ต)
- 7) เอาต์พุต RCA และ HDMI เอาต์พุตสัญญาณวิดีโอสำหรับต่อกับโทรทัศน์ หรือ
- จอแสดงผล
- 8) เอาต์พุตเสียง แจ็คหูฟังขนาด 3.5 มิลลิเมตร
- 9) พอร์ต Ethernet หรือ LAN
- 10) พอร์ตอินพุตเอาต์พุต GPIO (General Purpose Input/Output) ที่มีขาต่อแบบบัส SPI (Serial Peripheral Interface Bus), I2C, I2S
- 11) ขาสัญญาณรับส่งข้อมูลอนุกรม หรือ UART
- 12) Socket ของ SD การ์ด
- 13) ไฟเลี้ยง 5 โวลต์ 700 มิลลิแอมป์
- 14) ขนาด 85.60 x 53.93 มิลลิเมตร

2.3.2 ระบบปฏิบัติการ Raspberry Pi

ระบบปฏิบัติการสำหรับ Raspberry Pi ใช้บนพื้นฐานของระบบปฏิบัติการ Linux kernel ชิป ARM11 (ARMv6k) ที่ใช้ในบอร์ดเวอร์ชันกับระบบปฏิบัติการลินุกซ์ รวมถึง Ubuntu ด้วย

แต่ชิป ARM11 นี้ทำให้ไม่สามารถรันระบบปฏิบัติการวินโดวส์บนบอร์ด Raspberry Pi ได้ แต่สามารถติดตั้งระบบปฏิบัติการสำหรับ Raspberry Pi ได้โดยผ่านตัวจัดการ NOOBS ซึ่งมีรายชื่อดังนี้ Archlinux ARM, OpenELEC, Pidora (Fedora Remix), Puppy Linux, Raspbmc, RISC OS และ Raspbian ที่เป็นที่นิยม

2.3.3 โครงสร้าง Raspberry Pi (model B+)



ทั้งนี้ยังต้องมีอุปกรณ์เพิ่มเติมสำหรับเปิดใช้งานบอร์ด ได้แก่ แหล่งจ่ายไฟ 5 Volt DC ที่สามารถจ่ายไฟแรงดันไม่ต่ำกว่า 700 mA สาย micro USB สำหรับเชื่อมต่อระหว่างอะแดปเตอร์กับบอร์ดสาย AV ต่อกับพอร์ต RAC Video ดูเพื่อส่งสัญญาณภาพออกสู่จอแสดงผลหรือสาย HDMI และ SD-card ความจุไม่ต่ำกว่า 4 GB เพื่อใช้รันระบบปฏิบัติการ รวมทั้งสาย LAN หรือ WiFi USB adapter สำหรับเชื่อมต่อเครือข่าย



รูปที่ 2.4 อุปกรณ์สำหรับ raspberry pi

2.3.4 การควบคุม GPIO ของ Raspberry Pi

สำหรับระบบปฏิบัติการของ Raspbian จะติดตั้งโมดูล GPIO พร้อมในตัว หากไม่มี การติดตั้งสามารถติดตั้งได้จากคำสั่ง

```
Sudo apt-get update
```

```
Sudo apt-get install python-dev
```

```
Sudo apt-get install python-rpi.gpio
```

เมื่อทำการจ่ายไฟให้บอร์ด Raspberry Pi แล้ว ตามมาตรฐานของ Model B ซึ่งมี แรงดันไฟฟ้าขนาด 3.3V ออกทางขาหมายเลข 1,17 ส่วนแรงดันไฟฟ้าขนาด 5V ออกทางขา 2, 4 และ Ground ที่ขา 6, 9, 20, 25 ตามรูป 2.14 ลำดับหมายเลขคุณสมบัติของขา GPIO

Raspberry Pi B Rev 1 P1 GPIO Header				Raspberry Pi A/B Rev 2 P1 GPIO Header			
		Pin No.				Pin No.	
3.3V	1	2	5V	3.3V	1	2	5V
GPIO0	3	4	5V	GPIO2	3	4	5V
GPIO1	5	6	GND	GPIO3	5	6	GND
GPIO4	7	8	GPIO14	GPIO4	7	8	GPIO14
GND	9	10	GPIO15	GND	9	10	GPIO15
GPIO17	11	12	GPIO18	GPIO17	11	12	GPIO18
GPIO21	13	14	GND	GPIO27	13	14	GND
GPIO22	15	16	GPIO23	GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24	3.3V	17	18	GPIO24
GPIO10	19	20	GND	GPIO10	19	20	GND
GPIO9	21	22	GPIO25	GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8	GPIO11	23	24	GPIO8
GND	25	26	GPIO7	GND	25	26	GPIO7

Key	
Power +	UART
GND	SPI
I ² C	GPIO

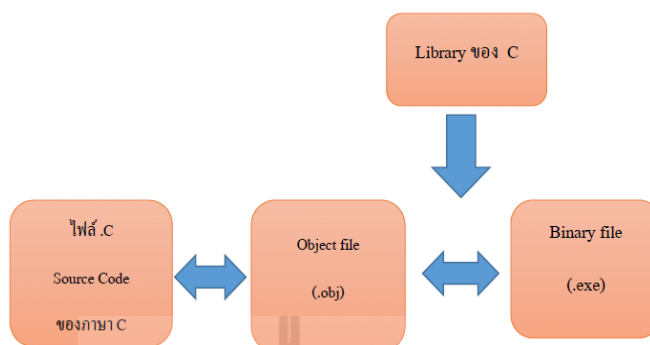
รูปที่ 2.5 คุณสมบัติของขา GPIO

2.3.5 ภาษา Python

Python เป็นภาษาระดับสูงภาษาหนึ่ง ที่มีความสามารถสูงถูกสร้างขึ้นในปี 1989 โดย Guido van Rossum ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux, Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษานี้เป็นภาษาลิขสิทธิ์ Open Source เหมือนอย่าง PHP

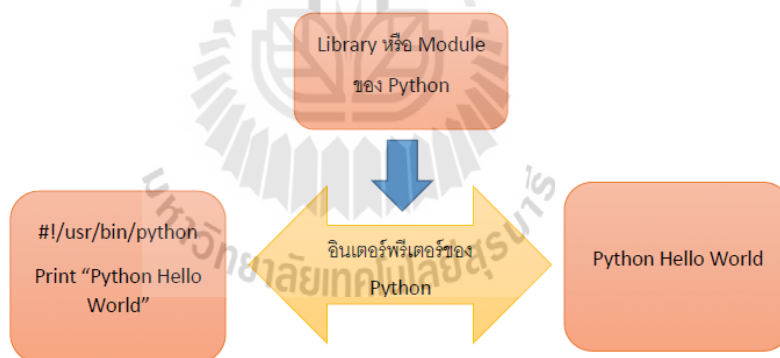
หลักการทำงานของภาษา Python โดยลักษณะของตัวแปลภาษานั้นแบ่งได้ 2 ประเภท ใหญ่ ๆ คือ

- 1) คอมไพเลอร์ (Compiler) เป็นตัวแปลภาษาสำหรับภาษา C, C++, Pascal การทำงาน ก็คือจะตรวจสอบความผิดพลาดของโค้ดคำสั่งตั้งแต่ต้นจนจบก่อน หรือเรียกว่าการคอมไพล์ ถ้าไม่มีข้อผิดพลาดก็จะทำการแปลโค้ดคำสั่งของเราให้เป็นไฟล์นามสกุล .obj (object file) จากนั้นก็ทำการแปลไฟล์ .obj ให้เป็นไบนารีไฟล์ .exe เพื่อทำงานต่อไป ดังตัวอย่างการทำงานของคอมไพเลอร์ ภาษา C ดังรูปที่ 2.6



รูปที่ 2.6 คอมไพเลอร์ (Compiler)

2) อินเตอร์พรีเตอร์ (Interpreter) จะทำงานเป็นบรรทัดต่อบรรทัด คือ อ่านโค้ดคำสั่งมาบรรทัดหนึ่งแล้วก็ทำงานให้ผลออกมาเลย ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 อินเตอร์พรีเตอร์ (Interpreter)

จากรูปตัวอย่างในกรณีที่มีการเรียกใช้ฟังก์ชันจากไลบรารี (Library) หรือ โมดูล (Module) ของภาษา Python อินเตอร์พรีเตอร์ของภาษา Python ก็จะไปทำการเรียก ฟังก์ชันเหล่านั้นให้ทำงานแล้วจึงแสดงผลการทำงานออกมา ในส่วนของประสิทธิภาพการทำงานนั้นตัวแปลภาษาแบบคอมไพเลอร์จะทำงานได้เร็วกว่าตัวแปลภาษาแลอินเตอร์พรีเตอร์ เพราะโค้ดคำสั่งถูกคอมไพล์และ

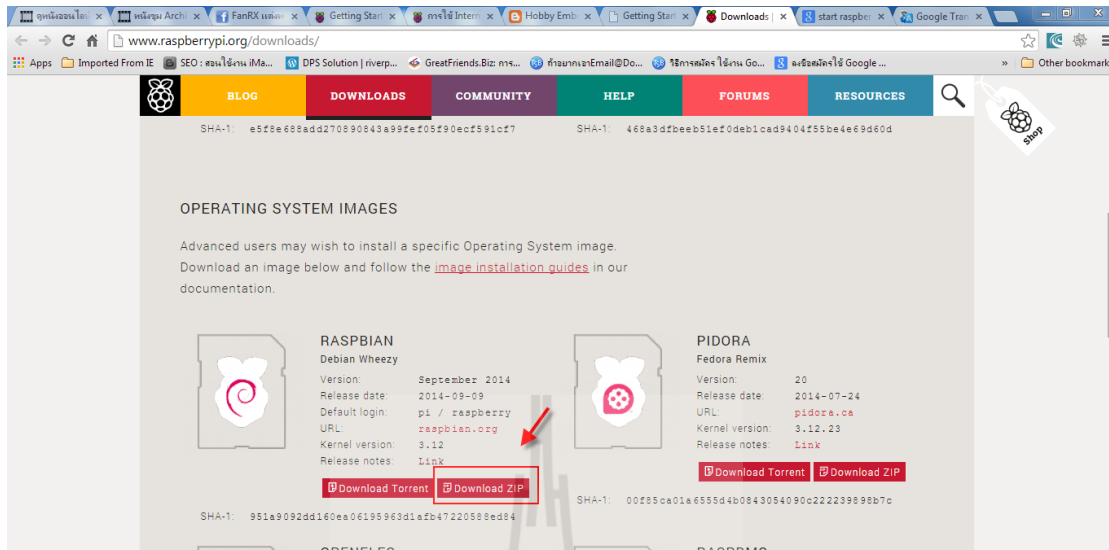
ลิงค์โดยตัวแปลภาษาแบบคอมไพเลอร์ผ่านแล้วได้เป็นไฟล์ .exe ออกมา จากนั้นก็เป็นขั้นตอนการทำงานอย่างเดียว

สรุปข้อดีของภาษา Python ได้ดังนี้

- 1) ง่ายต่อการเรียนรู้โดยภาษา Python มีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษา Python จะคล้ายกับภาษา C มาก เพราะภาษา Python สร้างขึ้น มา โดยใช้ภาษา C ทำให้ผู้ที่คุ้นเคยภาษา C อยู่แล้วใช้งานภาษา Python ได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี
- 2) ไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้น เพราะตัวแปลภาษา Python อยู่ภายใต้ลิขสิทธิ์ GNU
- 3) ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษา Python ถูกออกแบบใช้งานกับระบบ Unix อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษา Python ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ อาทิเช่น Linux, Windows 95/98/ME, Windows NT, Windows2000, OS/2
- 4) ภาษา Python ถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษา C, C++, Java, Perl
- 5) ภาษา Python เป็นภาษาประเภท Server side Script คือการทำงานของภาษา Python จะทำงานด้านฝั่ง Server แล้วส่งผลลัพธ์กลับมายัง Client ทำให้มีความปลอดภัยสูง
- 6) ใช้พัฒนา Web Service โดยที่ภาษา Python สามารถนำมาพัฒนาเว็บเซอร์วิส รวมทั้งใช้บริหารการสร้างเว็บไซต์สำเร็จรูปที่เรียกว่า Content Management Framework (CMF) ตัวอย่าง CMF ที่มีชื่อเสียงมากและเบื้องหลังทำงานด้วย python คือ plone

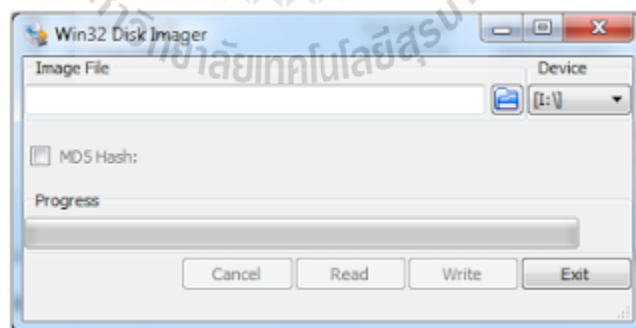
2.3.6 การตั้งค่าและการเริ่มต้นใช้งานโปรแกรมบอร์ด Raspberry Pi

- 1) เข้าไป Download ระบบปฏิบัติการ Raspbian "Wheezy" ได้ที่ <http://www.raspberrypi.org/downloads> แดกไฟล์ Zip แล้วจะได้ไฟล์ image 2014-09-09-wheezy-raspbian.img หรือ Version ที่สูงกว่า ดูความแตกต่างระหว่าง Version ได้จาก http://downloads.raspberrypi.org/raspbian/release_notes.txt ดังปรากฏหน้าต่างดังรูปที่ 2.8



รูปที่ 2.8 หน้าต่าง Download ระบบปฏิบัติการ Raspbian "Wheezy"

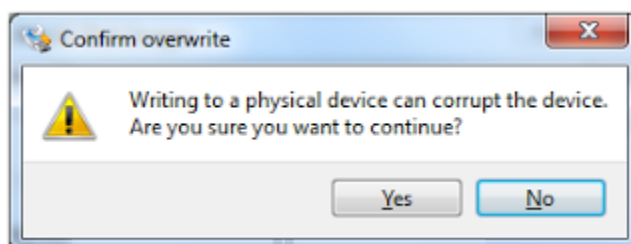
2) ต้องทำการติดตั้งโปรแกรม Win32DiskImage สำหรับเขียนไฟล์ image ก่อนโดย Download ได้ที่ <http://sourceforge.net/projects/win32diskimager/> เมื่อติดตั้งโปรแกรมเสร็จแล้วให้รันโปรแกรมขึ้นมาดังปรากฏหน้าต่างดังรูปที่ 2.9



รูปที่ 2.9 วิธีการติดตั้งโปรแกรม Win32DiskImage

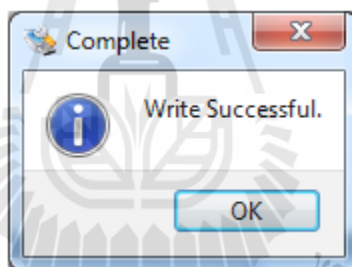
ช่อง Device ให้เลือก Drive ที่เป็น SD Card จากนั้นคลิกที่ Folder สีน้ำเงินเพื่อเลือกไฟล์ 2014-09-09-wheezy-raspbian.img ที่ Extract เอาไว้กด Write เพื่อทำการเขียน Disk Image ของ

raspbian ลง SD Card ได้เลย ระบบจะถามยืนยันอีกครั้งให้ตอบ Yes ไปเพื่อเริ่มเขียนไฟล์ หลังจากเขียนเสร็จเรียบร้อยแล้วจะปรากฏหน้าต่างดังรูปที่ 2.10



รูปที่ 2.10 วิธีการติดตั้งโปรแกรม Win32DiskImage

กด OK แล้วก็ Exit โปรแกรม ก็ถือว่า SD Card นี้มีระบบปฏิบัติการ Raspbian ของ Raspberry Pi ดังรูปที่ 2.11



รูปที่ 2.11 วิธีการติดตั้งโปรแกรม Win32DiskImage

3) หลังจากที่ได้ SD Card ที่มีข้อมูลของ Raspbian Wheezy เรียบร้อยแล้ว ก็สามารถนำไปเสียบในช่องเสียบ SD Card ของ Board Raspberry Pi หลังจากนั้นบอร์ด Raspberry Pi ก็จะเริ่มทำงาน และเริ่ม Boot ดังรูปที่ 2.12



```

Using makefile-style concurrent boot in runlevel 2.
Network Interface Plugging Daemon...skip eth0...done.
Starting NFS common utilities: statd.
Starting enhanced syslogd: rsyslogd.
Starting system message bus: dbus.
Starting periodic command scheduler: cron.
Starting NTP server: ntpd.
Starting portmap daemon...Already running..
Starting Hardware abstraction layer: hald.
Starting internet superserver: xinetd.
My network IP address is 10.0.2.15

Debian GNU/Linux 6.0 raspberrypi tty1

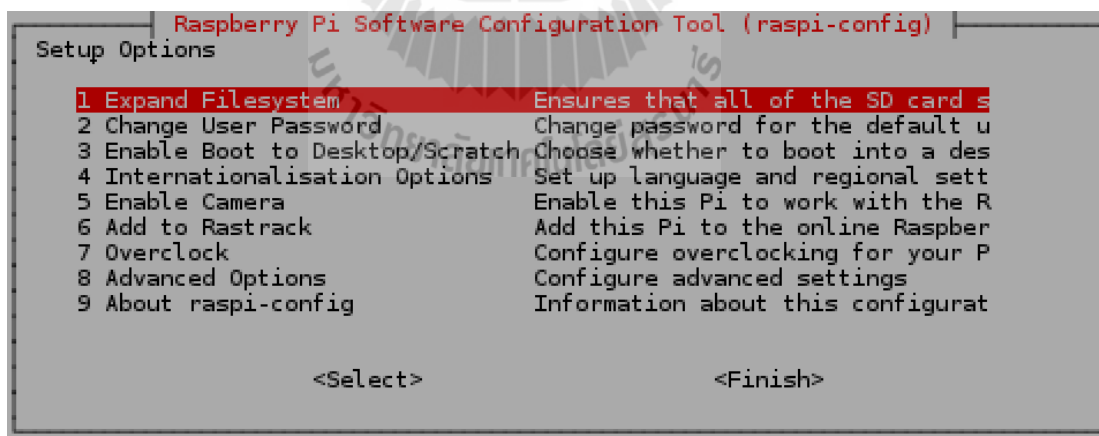
raspberrypi login: pi
Password:
Linux raspberrypi 3.1.9+ #2 Mon Apr 16 04:53:15 EST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

```

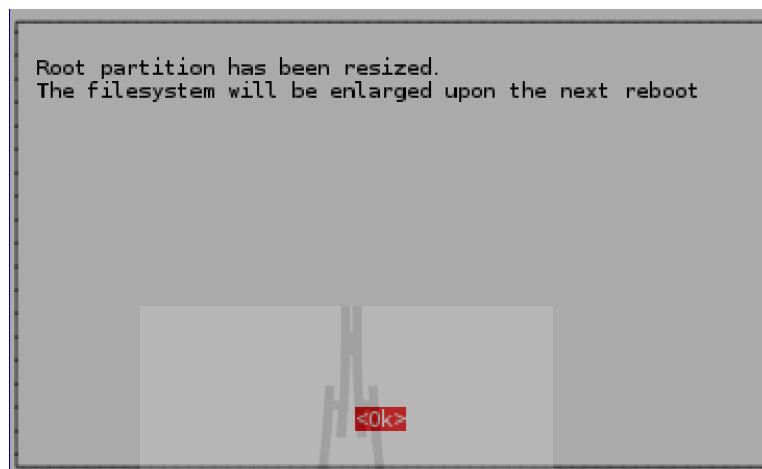
รูปที่ 2.12 บอร์ด Raspberry Pi ก็จะเริ่มทำงาน

4) หลังจากระบบปฏิบัติการ Boot เสร็จเรียบร้อยแล้วจะปรากฏหน้าต่างแสดงดังรูปที่ 2.13 ให้เลือกเมนู Expand Filesystem เพื่อขยายพื้นที่บน SD Card ให้ใช้งานได้เต็มความจุ เลือกด้วยลูกศรขึ้นลงแล้วกด Enter



รูปที่ 2.13 วิธีตั้งค่าขยายพื้นที่บน SD Card ให้เต็มความจุ

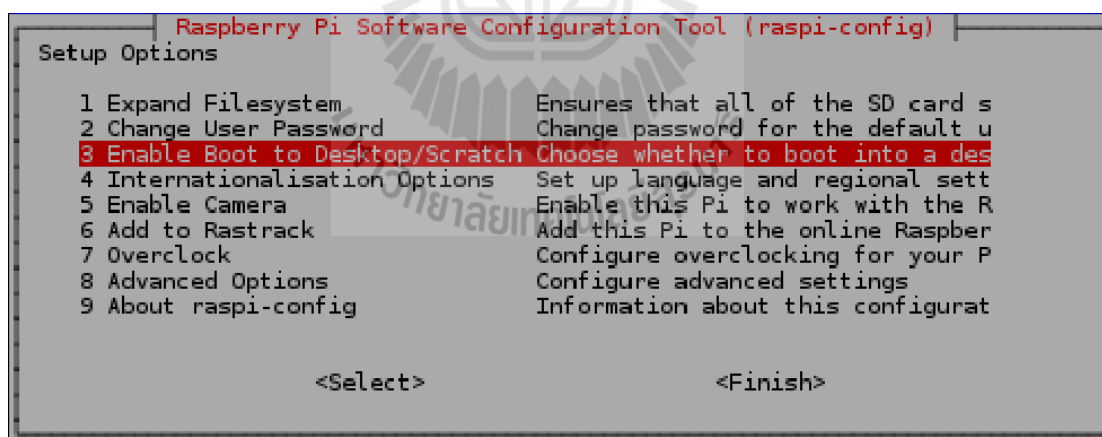
จะปรากฏหน้าต่างแสดงดังรูปที่ 2.14 ให้กด Enter อีกครั้ง



รูปที่ 2.14 วิธีตั้งค่าขยายพื้นที่บน SD Card ให้เต็มความจุ

5) กำหนดรูปแบบการใช้งานระบบปฏิบัติการให้ใช้งานในโหมด Graphic ให้เลือก

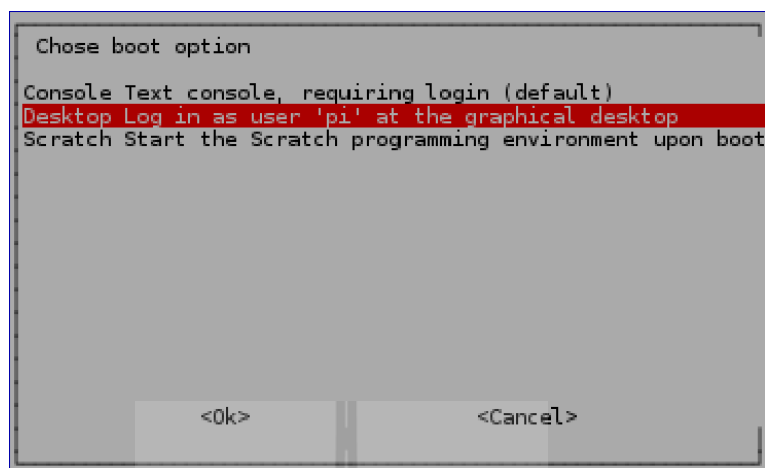
เมนู 3 Enable Boot to Desktop/Scratch แล้วกดแป้นพิมพ์ Enter ดังรูปที่ 2.15



รูปที่ 2.15 วิธีกำหนดรูปแบบการใช้งานระบบปฏิบัติการ

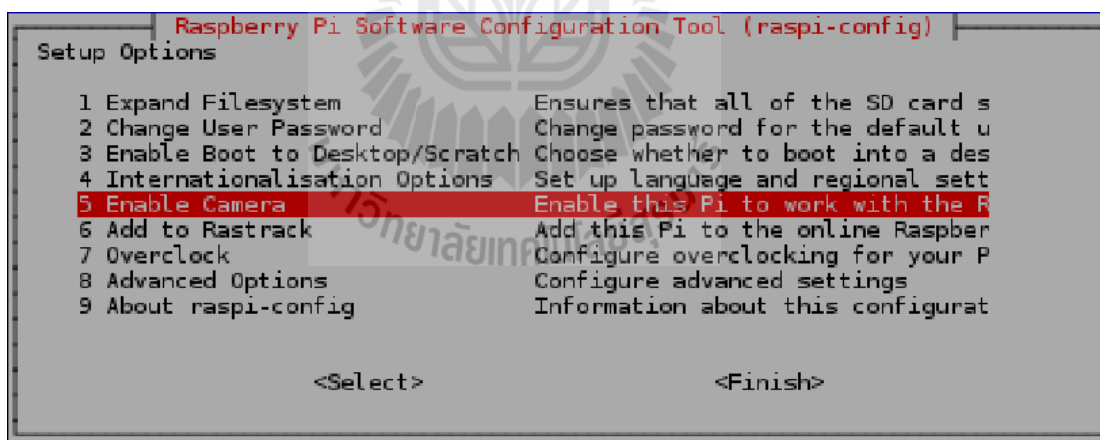
หลังจากนั้นจะปรากฏตัวเลือกมาทั้งหมด 3 ตัวเลือก ให้เลือก Desktop Log in as user

'pi' at the graphical desktop แล้วกดแป้นพิมพ์ Enter ดังรูปที่ 2.16



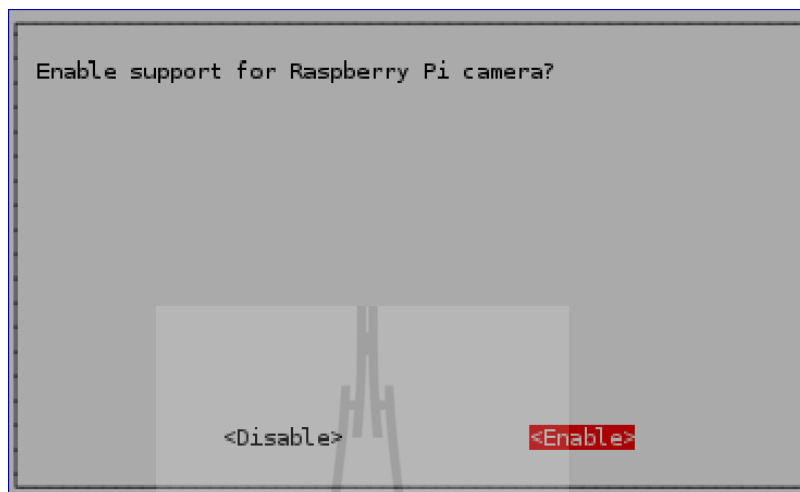
รูปที่ 2.16 วิธีกำหนดรูปแบบการใช้งานระบบปฏิบัติการ

6) หากต้องการใช้งานโมดูลกล้อง (Raspberry Pi Camera Module) ต้องเปิดฟังก์ชันการใช้งานนี้ด้วย โดยเลือกเมนู 5 Enable Camera แล้วกด Enter แต่ถ้าหากไม่ต้องการใช้ให้ข้ามขั้นตอนนี้ไป ดังรูปที่ 2.17



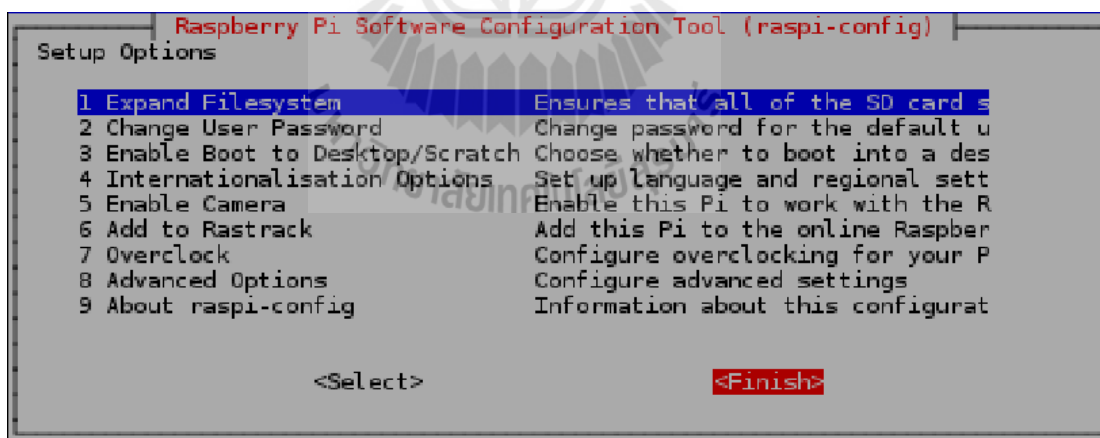
รูปที่ 2.17 วิธีเปิดใช้งานโมดูลกล้อง

หลังจากนั้นให้เลือก Enable แล้วกดแป้นพิมพ์ Enter ดังรูปที่ 2.18



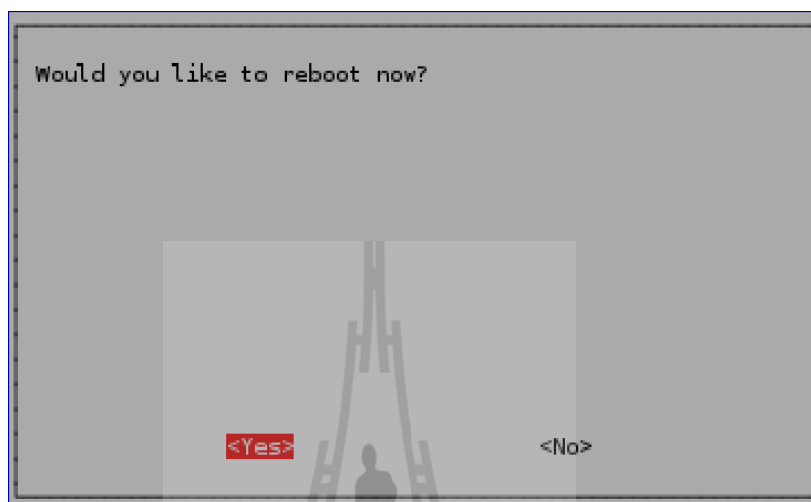
รูปที่ 2.18 วิธีเปิดใช้งานโมดูลกล้อง

7) สุดท้ายให้เลื่อนไปที่ Finish แล้วกด Enter เพื่อจบการตั้งค่าระบบและ Reboot ระบบใหม่ดังรูปที่ 2.19



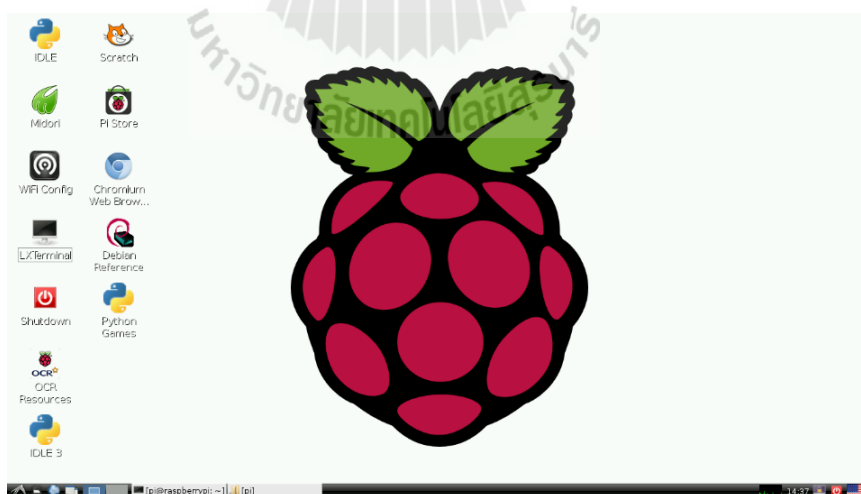
รูปที่ 2.19 วิธีออกจากหน้าเมนูการตั้งค่าระบบและ Reboot ระบบใหม่

ระบบจะถามซ้ำอีกครั้งว่าต้องการจะ Reboot ระบบใหม่นี้หรือไม่ เลือก Yes แล้วกด Enter ระบบก็จะ Reboot ใหม่ทันทีดังรูปที่ 2.20



รูปที่ 2.20 วิธีออกจากหน้าเมนูการตั้งค่าระบบและ Reboot ระบบใหม่

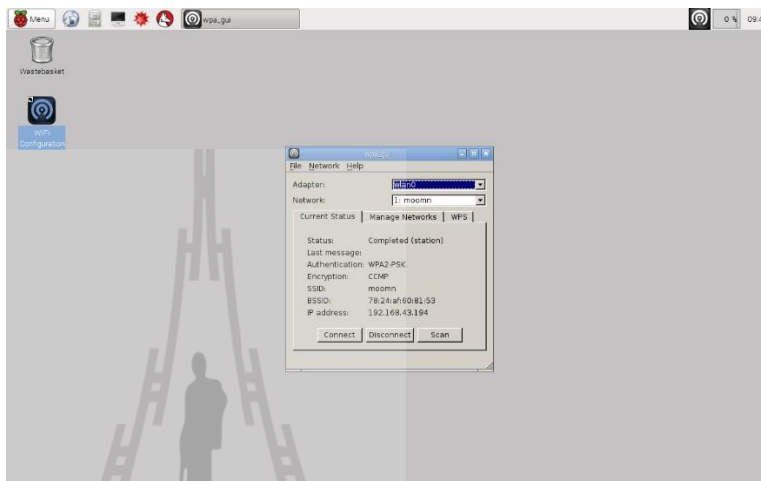
8) หลังจากทีระบบ Reboot ใหม่เสร็จเรียบร้อยแล้ว จะแสดงหน้า Desktop ของ Raspbian ดังรูปที่ 2.21



รูปที่ 2.21 หน้า Desktop ระบบปฏิบัติการ Raspbian

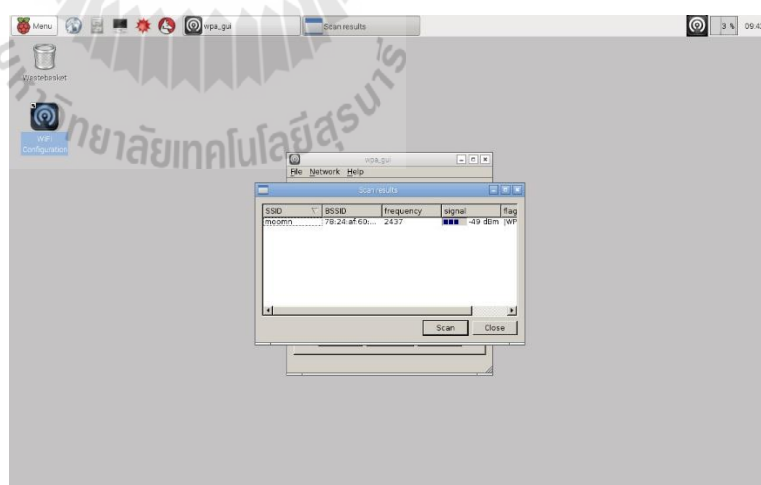
2.3.7 การเชื่อมต่ออินเทอร์เน็ตผ่าน Wi-Fi ให้บอร์ด Raspberry Pi

1) เปิดโปรแกรม Wi-Fi config ตามรูปที่ 2.22 ถ้ามีการเชื่อมต่ออุปกรณ์ Wi-Fi adapter แล้ว จะปรากฏ wlan0 ในช่อง Adapter



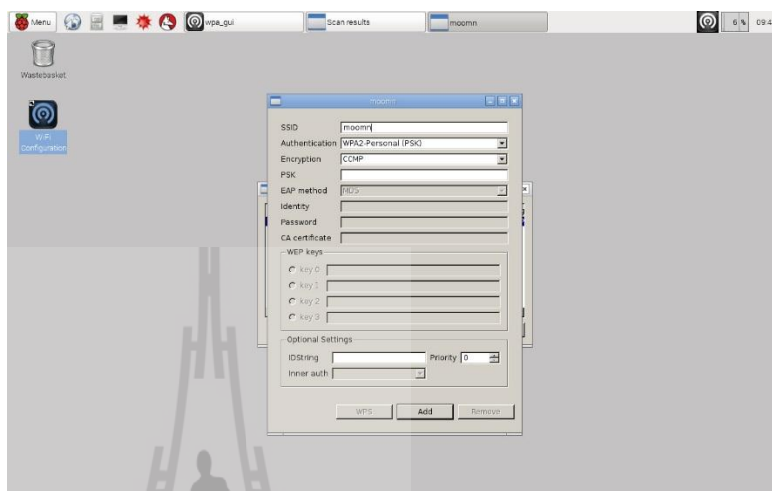
รูปที่ 2.22 โปรแกรม Wi-Fi config

2) กดปุ่ม Scan ในการหาเครือข่ายในการเชื่อมต่อครั้งแรก ตามรูปที่ 2.23



รูปที่ 2.23 การค้นหาเครือข่าย Wi-Fi

3) ดับเบิลคลิกเลือกและใส่รหัสผ่าน Wi-Fi ที่ช่อง PSK กด Add เพิ่มการเชื่อมต่อตามรูปที่ 2.24



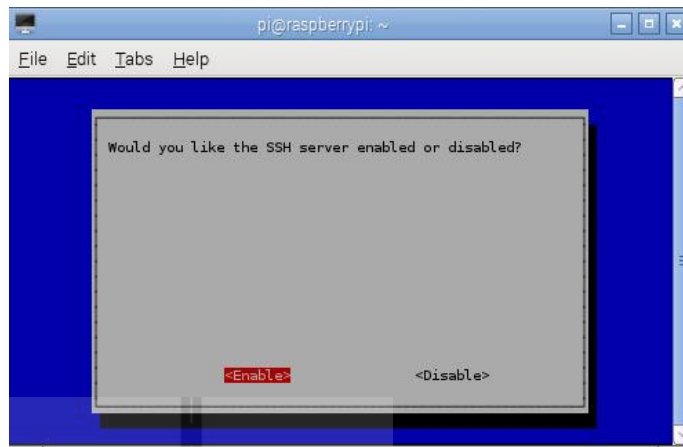
รูปที่ 2.24 การกรอกรหัสผ่านเชื่อมต่อ Wi-Fi

2.3.8 การ Remote เข้าใช้งานบอร์ด Raspberry Pi

1) เปิดการใช้งาน SSH (Secure Shell) ของบอร์ด Raspberry Pi โดยเข้าผ่าน LXterminal แล้วใช้คำสั่ง

```
$ sudo raspi-config
```

จะปรากฏหน้า Configuration Tools แล้วเลื่อน Cursor เลือก Advanced Options > SSH เลือก Enable เพื่อยืนยันการเปิดใช้งานตามรูปที่ 2.25



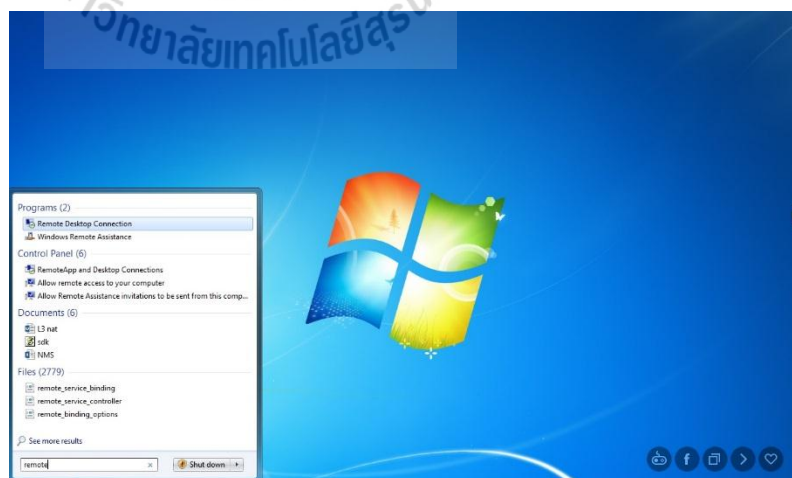
รูปที่ 2.25 หน้ายืนยันการเปิดใช้งาน SSH

ดังนี้

- 2) ติดตั้ง xrdp เพื่อเป็นตัวกลางในการเชื่อมต่อ remote ผ่าน LXterminal โดยใช้คำสั่ง

```
$ sudo apt-get install xrdp
```

- 3) เปิดโปรแกรม Remote Desktop บนอุปกรณ์ปลายทาง ในที่นี้คือคอมพิวเตอร์ระบบปฏิบัติการ Windows 7 เข้า Start ค้นหาโปรแกรม Remote Desktop Connection ตามรูปที่ 2.26



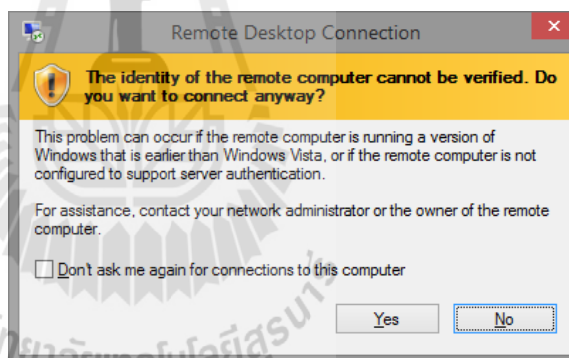
รูปที่ 2.26 การเปิดโปรแกรม Remote Desktop Connection

4) กรอกรหัสเลข IP ของบอร์ด Raspberry Pi ที่ได้จากหน้า Connection Wi-Fi
ข้างต้น ลงในโปรแกรมแล้ว Connect ตามรูปที่ 2.27



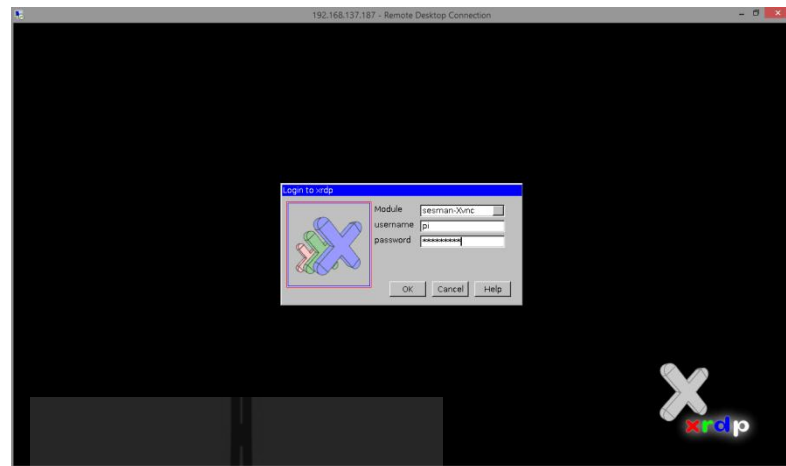
รูปที่ 2.27 การ Remote เข้าบอร์ด Raspberry Pi

5) ยืนยันการเชื่อมต่อ กดปุ่ม Yes ตามรูปที่ 2.28

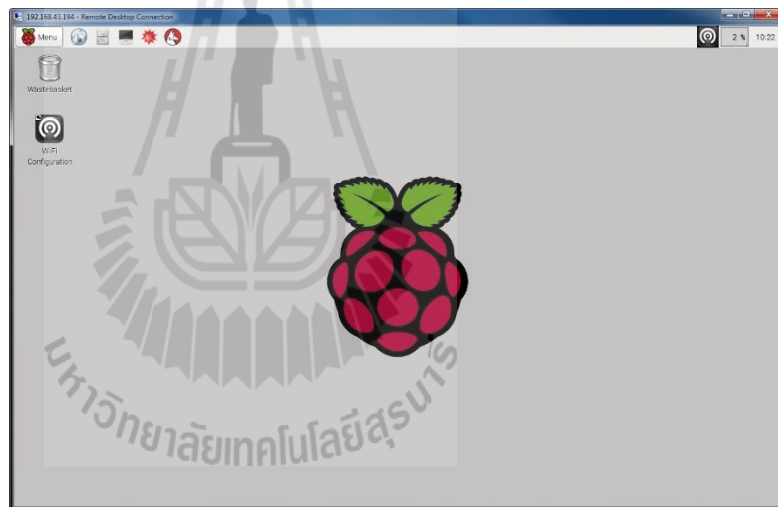


รูปที่ 2.28 ยืนยันการเชื่อมต่อ Remote

6) จะเข้าสู่หน้า Log in xrdp เข้าใช้งานบอร์ด Raspberry Pi ดังรูป 2.29 โดยใช้
Username: pi, Password:raspberry เหมือนการ Log in ใช้งานปกติ จากนั้นจะเข้าสู่หน้า Desktop
ของระบบปฏิบัติการ Raspbian ดังรูปที่ 2.30 ตามลำดับ



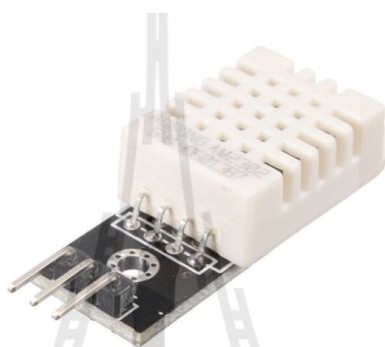
รูปที่ 2.29 หน้า Log in xrdp



รูปที่ 2.30 หน้า Desktop ของระบบปฏิบัติการ Raspbian ผ่าน Remote Desktop Connection

2.4 Temperature & Humidity Sensor Pro - DHT22 Features

อุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ (Temperature & Relative Humidity Sensor) เป็นอุปกรณ์ที่สามารถนำมาประยุกต์ใช้งานทางด้านระบบสมองกลฝังตัวได้หลากหลาย เช่น การวัดและควบคุมอุณหภูมิและความชื้น ระบบบันทึกข้อมูลเกี่ยวกับอุณหภูมิและความชื้นในห้อง เป็นต้น อุปกรณ์ประเภทนี้แตกต่างกันตามผู้ผลิต ราคา ความแม่นยำ ความละเอียดในการวัด การให้ค่าแบบดิจิตอลหรือแบบอนาล็อก เป็นต้น

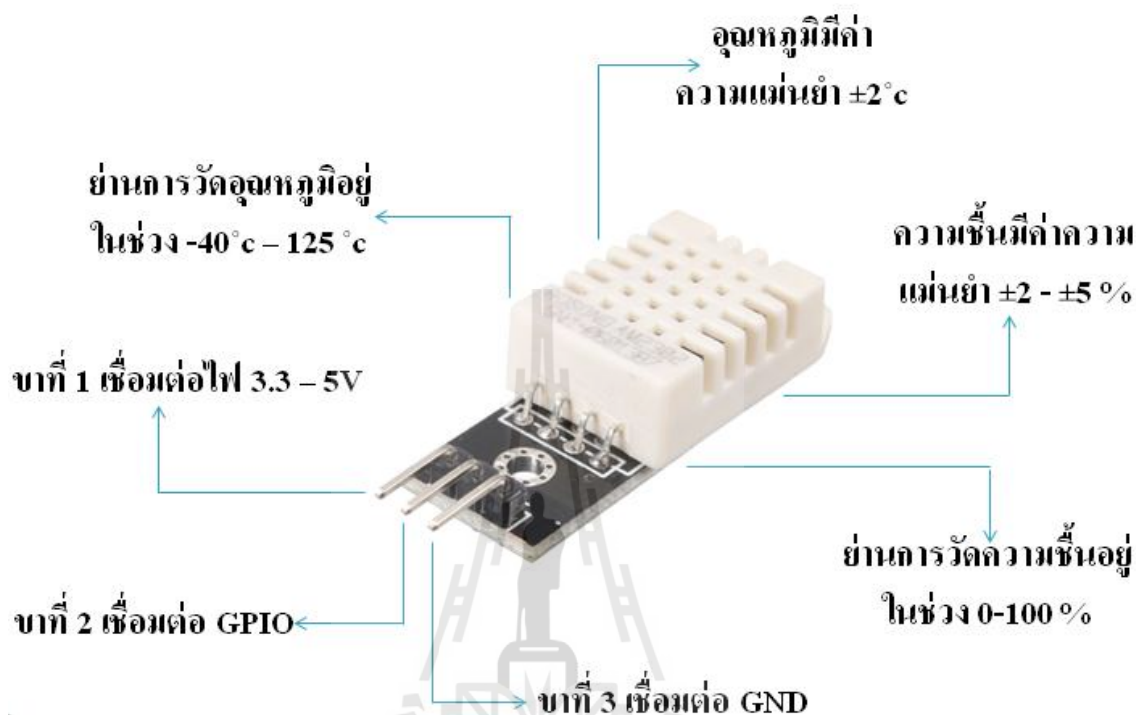


รูปที่ 2.31 ลักษณะตัวเซนเซอร์ DHT22

2.4.1 คุณสมบัติ Temperature & Humidity Sensor Pro - DHT22 Features

- 1) ไฟเลี้ยง 3.3-6 โวลต์
- 2) ย่านการวัดความชื้นอยู่ในช่วง 0-100 % RH โดยมีค่าความแม่นยำ $\pm 2 - \pm 5\%$ RH ความละเอียดในการวัด 1 % แสดงผลแบบ 8 บิต
- 3) ย่านการวัดอุณหภูมิอยู่ในช่วง $-40 \sim 125^{\circ}\text{C}$ โดยมีค่าความแม่นยำ ± 2 องศาเซลเซียสความละเอียดในการวัด 1 องศาเซลเซียส แสดงผลแบบ 8 บิต
- 4) อ่านค่าสัญญาณ (Sample Rate) ทุก 2 วินาที

2.4.2 โครงสร้าง Temperature & Humidity Sensor Pro - DHT22 Features



รูปที่ 2.32 ขาสัญญาณและการเชื่อมต่อของ DHT22

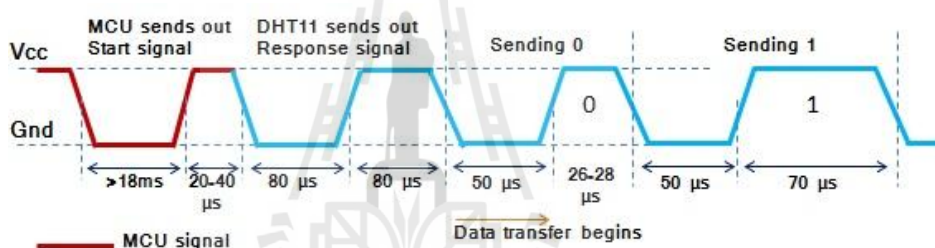
2.4.3 การทำงาน Temperature & Humidity Sensor Pro - DHT22 Features

วิธีการส่งข้อมูลของ DHT22 มีการทำงานดังจะกล่าวต่อไปนี้ อุปกรณ์นี้ใช้การสื่อสารกับ MCU ด้วยวิธี Single-wire Two-way Serial interface คือ การสื่อสารอนุกรมสองทางโดยใช้สายเส้นเดียวหรือการสื่อสารนี้ใช้สายสื่อสารเพียงเส้นเดียวและส่งข้อมูลจาก MCU ไปที่ตัว DHT22 และส่งข้อมูลจาก DHT22 ไปที่ตัว MCU

การใช้ pull up resistor หมายถึงต้องการให้เวลาทำงานปกติในขณะที่ไม่มีอุปกรณ์ต่ออยู่กับสายสื่อสาร ระดับแรงดันของสัญญาณในสาย Data คือแรงดันระดับ “สูง” และจะมีแรงดันในระดับ “ต่ำ” เมื่อมีอุปกรณ์ (จะเป็น บอร์ด Raspberry Pi หรือตัวเซนเซอร์ DHT22) ดึงสัญญาณลงใน

ระดับ “ต่ำ” ดังนั้นหากเราทำการวัดสัญญาณด้วยScopeได้ค่าเป็น “สูง” ตลอดเวลานั้นอาจจะเป็นไปได้ว่าอุปกรณ์อาจจะผิดปกติ

ในการสื่อสารโดยใช้สายเส้นเดียวนั้นจำเป็นต้องใช้โปรโตคอลที่ตกลงกันไว้ระหว่างตัว MCU และอุปกรณ์ที่ต้องการสื่อสารด้วย (DHT22) ดังนั้นขั้นตอนแรก Raspberry Pi จะต้องส่ง Start signal ที่เป็นแรงดันไฟฟ้าระดับต่ำอย่างน้อย 18 ไมโครวินาที ไปที่ DHT22 เพื่อให้ DHT22 เข้าใจว่าจะเริ่มส่งแล้วและรอ 20-40 ไมโครวินาทีเพื่อรอ DHT22 ตอบกลับ และเพื่อให้ Raspberry Pi รู้ว่า DHT22 พร้อมแล้ว DHT22 จะส่งแรงดันระดับต่ำกลับไป การส่งแรงดันจาก DHT22 กลับไปใช้เวลา 80 ไมโครวินาที จากนั้นจะรออีก 80 ไมโครวินาที ก่อนที่จะส่งข้อมูลบิตแรก



รูปที่ 2.33 กราฟแสดงการทำงานของเซนเซอร์ **Temperature & Humidity**

การส่งบิต "0" เป็นของ DHT22 โดยตัว DHT22 จะดึงระดับแรงดันลงต่ำนาน 50 ไมโครวินาที และปล่อยเป็นระดับ "สูง" นาน 26-28 ไมโครวินาที และการส่งข้อมูลเป็น "1" ตัวส่งจะดึงสายสัญญาณระดับต่ำ 50 ไมโครวินาที และปล่อยให้เป็นระดับสูงนาน 70 ไมโครวินาที (ดูช่วง Sending 1) ดังนั้น DHT22 ส่งมาเป็นบิต "0" หรือ "1" มาจนครบข้อมูลหนึ่งชุด ในแต่ละชุดของข้อมูลที่ส่งมาจาก DHT22 นั้นตัว MCU จะรับข้อมูลแล้วนำไปแปลงค่าต่อว่า DHT22 ข้อมูลที่ส่งมานั้นมีความหมายแปลว่าอย่างไร ใน 1 ชุดข้อมูลจะมีความยาว 40 บิต และใช้เวลาส่งประมาณ 40 มิลลิวินาที

ใน 40 บิตที่ส่งมา ประกอบด้วย " 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum" ดังตัวอย่างการคำนวณต่อไปนี้

Example: MCU has received 40 bits data from RHT03 as

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

16 bits RH data 16 bits T data check sum

Check sum = 0000 0010 + 1000 1100 + 0000 0001 + 0101 1111 = 1110 1110

$RH = (0000\ 0010\ 1000\ 1100) / 10 = 65.2\%RH$

$T = (0000\ 0001\ 0101\ 1111) / 10 = 35.1$

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

รูปที่ 2.34 ตัวอย่างการคำนวณหาค่า RH

2.5 i Module – Magnetic Switch

i Module – Magnetic Switch V1.0 เป็นโมดูลที่ถูกออกแบบมาเพื่อใช้เชื่อมต่อร่วมกับ i Module - Arduino Connector Base Shield รวมถึงรองรับการใช้งานร่วมกับบอร์ดทดลองหรือวงจรอื่นๆ โดยนำมาใช้เป็น Interface Input โดยทำงานร่วมกับแม่เหล็กเพื่อเปิดและปิดสวิทช์

2.5.1 คุณสมบัติ i Module – Magnetic Switch

- 1) ทำหน้าที่เป็นอุปกรณ์ที่ให้สัญญาณด้าน INPUT
- 2) สามารถใช้งานได้กับบอร์ดที่ใช้แรงดัน 3.3V – 5V
- 3) ให้สัญญาณออกมาเป็นระดับแรงดัน TTL โดยเมื่อตรวจพบสนามแม่เหล็กจะให้สัญญาณเป็น“1”- Active HIGH

2.5.2 คุณลักษณะ i Module – Magnetic Switch

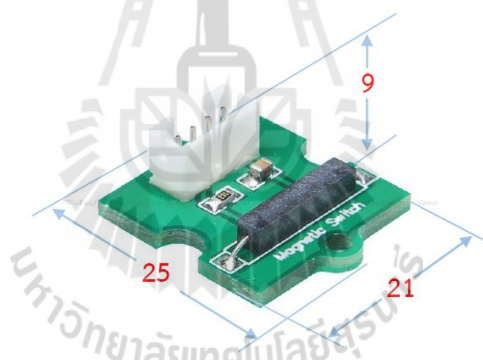
- 1) สวิทช์ถูกต่ออยู่ในลักษณะแบบ pull-down กับตัวต้านทานค่า 10k Ω
- 2) สามารถทำงานได้ที่อุณหภูมิ -40°C ถึง + 125°C
- 3) วงจรสวิทช์เป็นแบบ SPST - NO

- 4) สวิตช์จะทำงานเมื่อมีวัตถุที่มีสนามแม่เหล็กแผ่ออกมาในบริเวณใกล้เคียง
- 5) ขนาด (W x L x H) : 25 x 21 x 9 mm

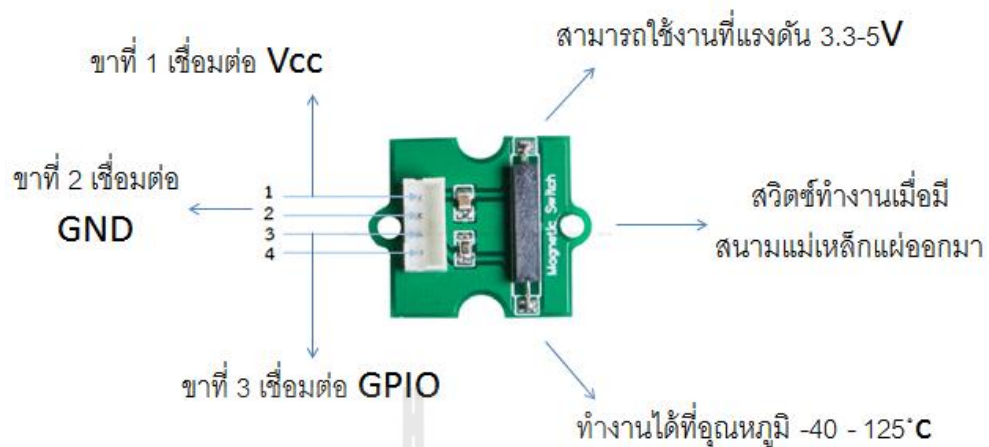
2.5.3 การนำไปประยุกต์ใช้งาน

- 1) ใช้เป็นโมดูลทำงานร่วมกับแม่เหล็ก
- 2) นำไปใช้งานเป็น Proximity Sensor
- 3) ใช้ในระบบ Security
- 4) ใช้ในการวัด Level Sensor

2.5.4 โครงสร้าง i Module – Magnetic Switch



รูปที่ 2.35 ลักษณะตัวเซนเซอร์ i Module – Magnetic Switch

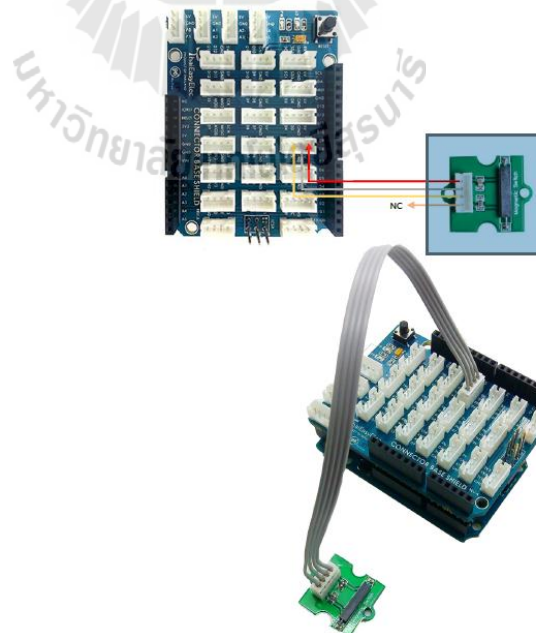


รูปที่ 2.36 ขาสัญญาณและการเชื่อมต่อของ i Module – Magnetic Switch

2.5.5 การใช้งาน i Module – Magnetic Switch

1. การเชื่อมต่อฮาร์ดแวร์ (Hardware Installation)

การเชื่อมต่อ จากตัวอย่างจะทำการเชื่อมต่อไปยัง i Module - Arduino Connector Base Shield แสดงดังภาพ หรือสามารถใช้สาย 4 Wire Cable(ELNK039) เพื่อใช้ในการเชื่อมต่อได้เช่นเดียวกัน



รูปที่ 2.37 การเชื่อมต่อฮาร์ดแวร์ของ i Module – Magnetic Switch

2. การเขียนโปรแกรม (Programming)

จากโปรแกรมด้านล่าง จะเป็นการทดสอบเมื่อนำแม่เหล็กมาแตะ โคนที่ตัวเซนเซอร์ แล้วทำการแสดงผลออกทางหน้าจอด้วยค่า “1”

รูปที่ 2.38 การเขียน โปรแกรมของ i Module – Magnetic Switch

2.6 Application LINE

LINE Application เป็นโปรแกรมที่ถูกสร้างขึ้นในช่วงกลางปี 2010 โดยการร่วมมือของ บริษัท Naver Japan Corporation และบริษัท livedoor โดยมี NHN Japan เป็นผู้พัฒนาฟีเจอร์ต่างๆ ของ ไลน์ และในส่วนของตลาดด้านธุรกิจนั้นยกให้บริษัทแม่ที่เกาหลี NHN Corporation จัดการ หลังจากที่เปิดตัวได้เพียงไม่นาน ก็ได้รับการตอบรับถึงหลายสิบล้านยูสเซอร์ในญี่ปุ่น ประเด็นแรกที่ใช้ ในการสร้างโปรแกรมแชท LINE ขึ้นมาก็มีสาเหตุมาจากเหตุการณ์แผ่นดินไหวที่ภูมิภาค Tohoku เมื่อ ต้นปี 2011 นั่นเอง ในตอนนั้นระบบการติดต่อทางการโทรศัพท์ล่ม ทำให้ NHN Japan ตัดสินใจ ออกแบบ App ที่สามารถใช้ได้ทั้งบนมือถือ บนแท็บเล็ต และคอมพิวเตอร์พีซี ซึ่งจะทำงานบนเครือข่าย ข้อมูลที่สามารถแชทตอบโต้ได้รวดเร็วและต่อเนื่อง



รูปที่ 2.39 สัญลักษณ์ของ Application LINE

ด้วยความที่ไลน์มีคุณสมบัติของโปรแกรมแชทครบถ้วน ตั้งแต่ แชท ส่งไฟล์รูป ไฟล์วิดีโอ ไฟล์เสียง ระบบการค้นหาเพื่อนด้วย QR Code หรือจะเป็นเกมที่เอาไว้เล่นคลายเครียด ยังมีอีกสิ่งหนึ่งที่ถือได้ว่าเป็นจุดเด่นของแอปพลิเคชันนี้ก็คือ “Sticker”

2.6.1 คุณสมบัติ Application LINE

- 1) รองรับหลายระบบปฏิบัติการ
- 2) ชิงก์ข้อมูลจากสมุดรายชื่อในโทรศัพท์
- 3) ข้อความที่ถูกจะส่ง จะถูกส่งในทันที
- 4) ส่งและแบ่งปันรูปภาพ, วิดีโอ, เพลง และอื่น ๆ
- 5) ส่งพิกัดสถานที่ที่อยู่
- 6) ส่ง อีโมจิ, สติกเกอร์ และ ไลน์อาร์ต
- 7) สร้างและร่วมกลุ่มเมื่อต้องการพูดคุยหรือแบ่งปันสื่อ
- 8) บอร์ดในกลุ่มสามารถจุผู้คนได้มากที่สุด 200 คน
- 9) เพิ่มเพื่อนโดยใช้ คิวอาร์โค้ด
- 10) เพิ่มเพื่อนโดยใช้ NFC ในการเชื่อมต่อโทรศัพท์ที่อยู่ใกล้ ๆ กัน
- 11) ข้อความเตือนแบบกล่อง สะดวกต่อการอ่านและตอบสนทนา
- 12) หน้าเส้นเวลา (Timeline) และหน้าแรก (Home Page) สำหรับ ไอโอ

เอส และ แอนดรอยด์

- 13) วินโดวส์และแมคสามารถใช้รุ่นของเดสก์ท็อปหรือสมาร์ทแท็บเล็ตผ่านเบราว์เซอร์

2.6.2 ระบบปฏิบัติการ Application LINE

LINE นั้นมีฟีเจอร์หลาย ๆ อย่างที่เป็นจุดเด่นทำให้ไม่เหมือนแอปฯ อื่น ๆ โดยเฉพาะสติ๊กเกอร์ตัวการ์ตูนน่ารัก ๆ ที่มีให้เลือกใช้ได้นับร้อยแบบเลยทีเดียว และที่สำคัญ LINE เป็นแอปฯ ที่สามารถดาวน์โหลดมาใช้ได้ฟรี ไม่ต้องมีค่าใช้จ่ายใด ๆ ในการใช้งานอีกด้วย

LINE คือแอปพลิเคชันที่ผสมผสานบริการ Messaging และ Voice Over IP นำมาผนวกเข้าด้วยกัน จึงทำให้เกิดเป็นแอปพลิเคชันที่สามารถแชท สร้างกลุ่ม ส่งข้อความ โพสต์รูปต่างๆ หรือจะโทรคุยกันแบบเสียงก็ได้ โดยข้อมูลทั้งหมดไม่ต้องเสียเงิน หากเราใช้งานโทรศัพท์ที่มีแพ็คเกจอินเทอร์เน็ตอยู่แล้ว แล้วยังสามารถใช้งานร่วมกันระหว่าง iOS และ Android รวมทั้งระบบปฏิบัติการอื่น ๆ ได้อีกด้วย การทำงานของ LINE นั้น มีลักษณะคล้าย ๆ กับ WhatsApp ที่ต้องใช้เบอร์โทรศัพท์เพื่อยืนยันการใช้งาน แต่ LINE ได้เพิ่มลูกเล่นอื่นๆ เข้ามา ทำให้ LINE มีจุดเด่นที่เหนือกว่า WhatsApp ได้แก่

- 1) Free Voice Calls (สนทนาด้วยเสียง ฟรี)



รูปที่ 2.40 แสดงหน้าจอการสนทนาด้วยเสียง

จุดเด่นอย่างหนึ่งของ LINE ก็คือบริการ Free Voice Calls ที่ให้ใช้งานสามารถโทรหาผู้ใช้ LINE ด้วยกัน โดยใช้งานผ่านเครือข่าย 3G และ Wi-Fi เพื่อส่งข้อมูลรูปแบบเสียง โดยไม่มีค่าใช้จ่ายใด ๆ

2) Send Videos & Voice Message (ส่งข้อความแบบวิดีโอและเสียง)



รูปที่ 2.41 แสดงหน้าจอการส่งข้อความแบบวิดีโอและเสียง

นอกจากการแชทด้วยการส่งข้อความแบบปกติแล้ว LINE ยังสามารถอัดภาพวิดีโอหรือเสียงแล้วส่งไปให้เพื่อน ๆ ได้อีกด้วย โดยสามารถส่งได้เป็นคลิปวิดีโอหรือเสียงในแบบสั้น ๆ ความยาวไม่กี่วินาที

3) Group Chat (แชทแบบกลุ่ม)



รูปที่ 2.42 แสดงหน้าจอการสนทนาแบบกลุ่ม

LINE สามารถสร้างกลุ่มเพื่อพูดคุยกันได้ หากมีก๊วนเพื่อนสนิท ต้องการความเป็นส่วนตัว อยากคุยเฉพาะกลุ่ม LINE เราก็สามารถสร้างกลุ่มเอาไว้พูดคุยได้

4) Stickers and Emoticons (สติ๊กเกอร์การ์ตูนน่ารัก ๆ)



รูปที่ 2.43 แสดงหน้าจอการสนทนาด้วยสติ๊กเกอร์

อีกหนึ่งความสนุกของแอปฯ แชททั่วไปที่ขาดไม่ได้ก็คืออีโมติคอนน่ารัก ๆ ที่ช่วยเพิ่มสีสันให้การแชทสนุกสนานยิ่งขึ้น และสำหรับ LINE มีทั้ง Stickers และ Emoticons รูปแบบต่าง ๆ และยังสามารถดาวน์โหลดเพิ่มเติมได้อีกด้วย ทำให้ผู้ใช้งานหลายคนติดอกติดใจกับ Stickers และ Emoticons น่ารัก ๆ ของ LINE

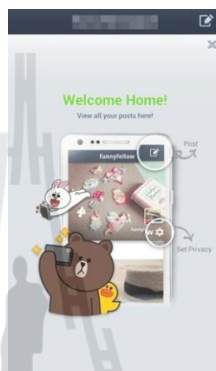
5) Customizable Wallpaper (ปรับแต่งภาพวอลเปเปอร์)



รูปที่ 2.44 แสดงหน้าจอการปรับแต่งภาพพื้นหลัง

LINE สามารถเปลี่ยน Wallpaper ในหน้าต่างแชทได้ โดยแอปฯ จะมีภาพ Wallpaper มาให้ทั้งหมด 23 แบบ และสามารถเพิ่ม Wallpaper ที่ต้องการ โดยนำรูปที่อยู่ในโทรศัพท์มือถือมาใช้งานเป็น Wallpaper ได้

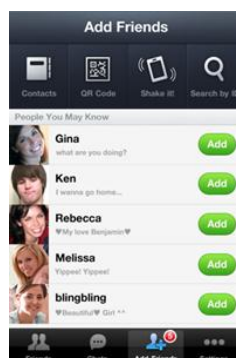
6) Timeline



รูปที่ 2.45 แสดงหน้าจอ Timeline

LINE มีความเป็นโซเชียลเน็ตเวิร์กในตัว มี Timeline ให้สามารถอัปเดตสเตตัส, โพสต์รูป, คอมเมนต์ หรือกดไลค์ได้เหมือนกับเฟซบุ๊กเลยทีเดียว

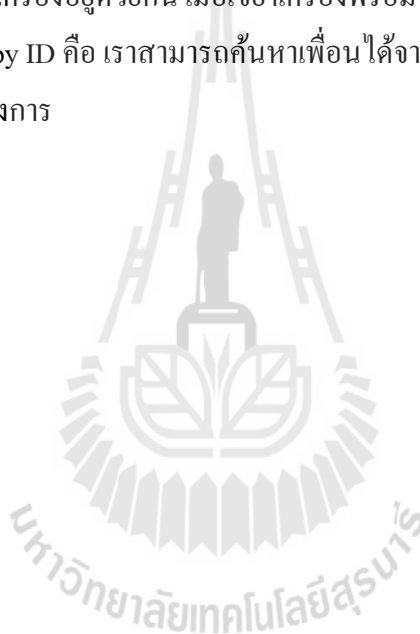
7) Add Friends / Contacts



รูปที่ 2.46 แสดงวิธีการเพิ่มเพื่อน

อีกหนึ่งลูกเล่นที่ทำให้ LINE แตกต่างจาก WhatsApp นั่นก็คือการเพิ่ม Contacts ที่เลือกได้ 4 รูปแบบ

1. เพิ่ม Contacts จากรายชื่อในโทรศัพท์หากมีเพื่อนคนไหนใช้แอปฯ นี้อยู่ จะมีสัญลักษณ์ LINE แสดงให้เห็นและสามารถเพิ่มเป็นเพื่อนได้ทันที
2. QR Code สามารถสแกน QR Code ของเพื่อนเราเพื่อเพิ่มเป็นเพื่อนและสามารถสร้าง QR Code ของเราเอง เพื่อใช้สำหรับให้เพื่อน ๆ คนอื่น มาสแกน QR Code เพื่อเพิ่มเพื่อนใน LINE
3. Shake it! เขย่าโทรศัพท์มือถือ เป็นวิธีการแอดเพื่อนที่เจ๋งสุด ๆ ของ LINE ใช้ในกรณีที่ทั้งสองโทรศัพท์สองเครื่องอยู่ด้วยกัน เมื่อเขย่าเครื่องพร้อม ๆ กัน ก็สามารถเพิ่มเป็นเพื่อนกันได้
4. Search by ID คือ เราสามารถค้นหาเพื่อนได้จาก ID (คล้าย ๆ กับ PIN ของ BB) โดยการพิมพ์ ID ของเพื่อนที่ต้องการ

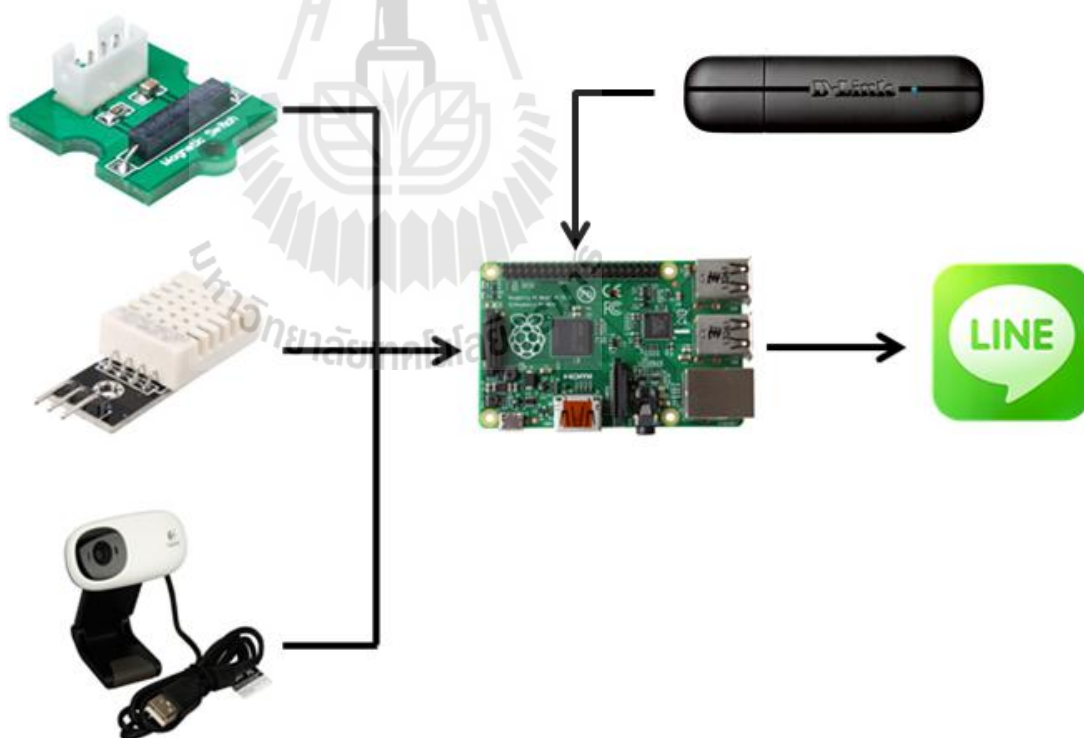


บทที่ 3

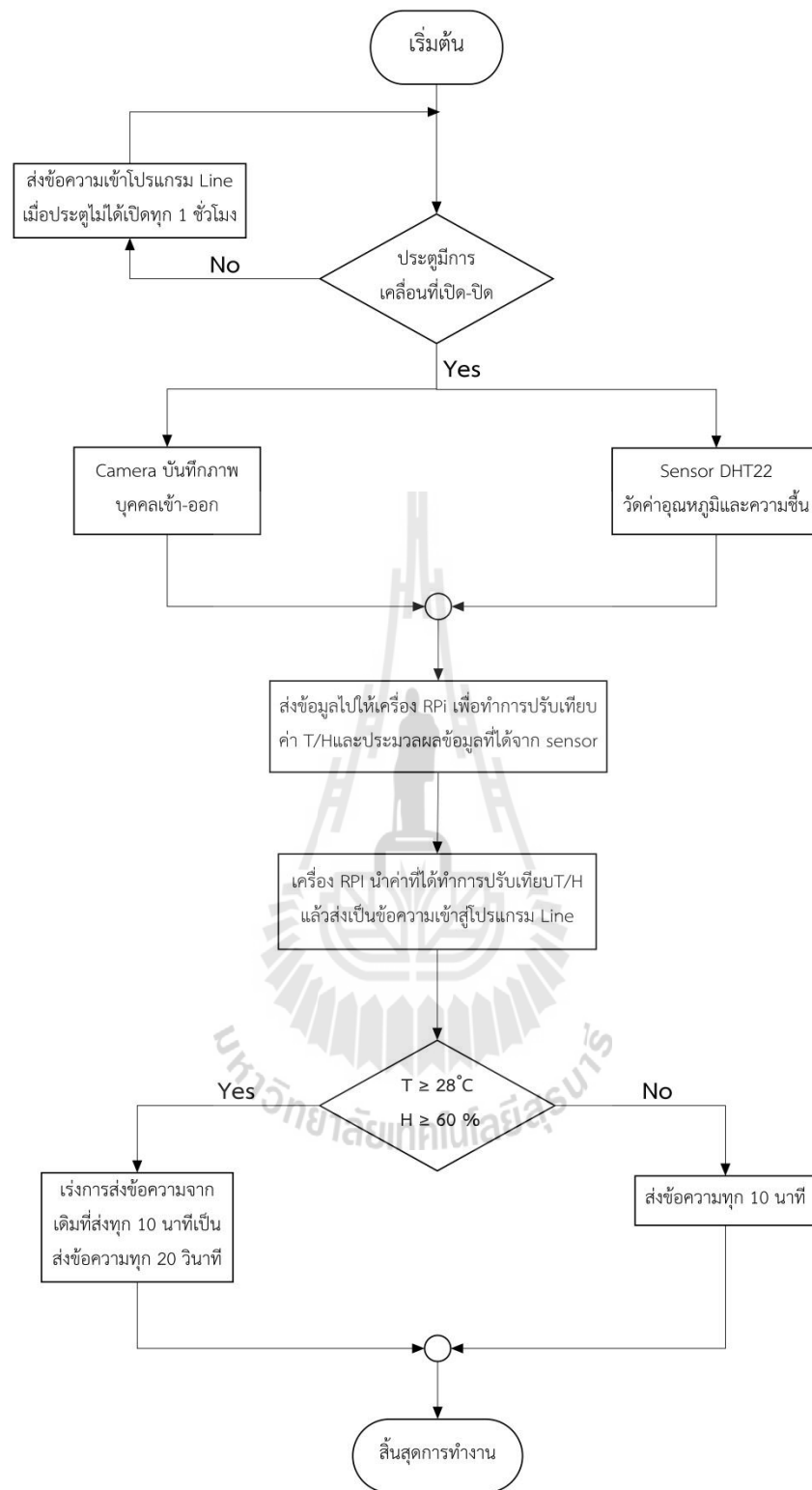
การออกแบบและการจำลอง

3.1 กล่าวนำ

ในบทนี้จะกล่าวถึงวิธีการออกแบบแบบจำลองชิ้นงาน เพื่อสามารถนำไปเป็นแนวทางในการสร้างชิ้นงานจริง โดยเริ่มจากการเตรียมอุปกรณ์ การใช้งานโปรแกรมที่ใช้สำหรับออกแบบควบคุมบอร์ด Raspberry Pi โดยบอร์ด Raspberry Pi ทำหน้าที่เป็นอุปกรณ์เชื่อมต่อ Line Server ที่สามารถตรวจสอบอุณหภูมิและความชื้น แล้วส่ง Alert ไปที่ Client แสดงผลผ่าน Line Application ดังรูป 3.1



รูปที่ 3.1 โครงสร้างการทำงานโดยรวม



รูปที่ 3.2 ขั้นตอนการทำงานของระบบโดยรวม

3.2 การเขียนโปรแกรม

3.2.1 การส่งข้อความและรูปภาพ

```
"""
from line import LineClient, LineGroup, LineContact
try:
    client = LineClient("happiness_you@hotmail.com", "gai036300182")
    #client = LineClient(authToken='AUTHTOKEN')
except:
    print "Login Failed"
#authToken = client.authToken
#print authToken

#print client.contacts
#friend = client.contacts[2]
#friend.sendMessage( "Hello" )

#friend = client.contacts[2]
#friend.sendImage( ' ./pic/wan.jpg' )
#friend.sendImage( ' ./pic/zuchi.jpg' )
#print friend

print client.groups
group = client.groups[2]
group.sendMessage( "Hello" )
group.sendImage( ' C:\Python27\Image.jpg' )
#group.sendImage( ' C:\Python27\sus.jpg' )
```

คำสั่งในโค้ดนี้จะทำการ Login และส่งรูปภาพกับข้อความเข้าระบบแอปพลิเคชันของ Line โดยใช้อีเมลในการ Login การที่จะส่งรูปภาพและข้อความนั้นก่อนอื่นจะต้องทำการเลือกกลุ่มในแอปพลิเคชัน Line ก่อนที่จะทำการส่งรูปภาพและข้อมูลไปที่กลุ่มไหนและเมื่อเลือกกลุ่มได้แล้วดังคำสั่ง `group = client.groups(2)` บอร์ด Raspberry Pi จะทำการส่งรูปภาพและข้อความไปที่กลุ่มในแอปพลิเคชัน Line



3.2.2 การเปิด-ปิดประตู

```
#!/usr/bin/python
import wiringpi
import ssl
import sys
import time
import wiringpi
import wiringpi2 as wiringpi
import sys
from time import sleep
import os

wiringpi.wiringPiSetup()
wiringpi.pinMode(5,0)
wiringpi.pinMode(6,1)

while True:
    status = wiringpi.digitalRead(5)
    if status == 1 :
        print status
        sleep(5)
    else :
        print status
        sleep(5)
```

คำสั่งในโค้ดนี้จะทำการเปิดและปิดประตูโดยการรับค่า 0 และ 1 จากการทำงานของเซนเซอร์ Magnetic Switch ถ้าค่าที่รับเข้ามาเป็น “1” หมายความว่า “ประตูปิด” ซึ่งแสดงว่ามีแม่เหล็กมาแตะโดนเซนเซอร์ Magnetic Switch แต่ถ้าค่าที่รับเข้ามาเป็น “0” หมายความว่า “ประตูเปิด” ซึ่งแสดงว่าไม่มีแม่เหล็กมาแตะโดนเซนเซอร์ Magnetic Switch

3.2.3 การวัดค่าอุณหภูมิและความชื้น

```
#!/usr/bin/python

import sys

import time

import Adafruit_DHT

sensor = Adafruit_DHT.DHT22

pin = 4

while True:

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

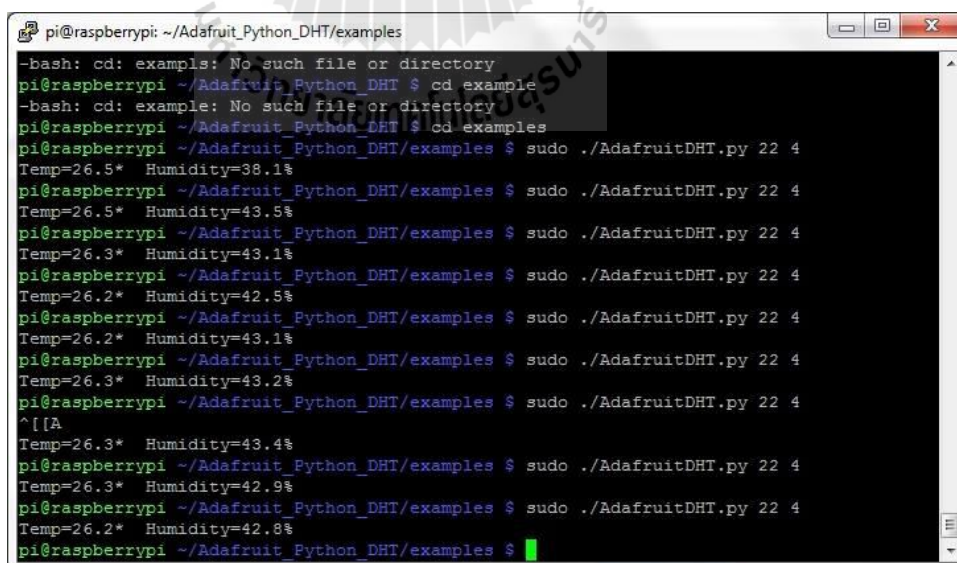
    if humidity is not None and temperature is not None:

        print 'Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature, humidity)

        time.sleep(2)

    else:

        print 'Failed to get reading. Try again!'
```



```
pi@raspberrypi: ~/Adafruit_Python_DHT/examples
-bash: cd: example: No such file or directory
pi@raspberrypi ~/Adafruit_Python_DHT $ cd example
-bash: cd: example: No such file or directory
pi@raspberrypi ~/Adafruit_Python_DHT $ cd examples
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=26.5* Humidity=38.1%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=26.5* Humidity=43.5%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=26.3* Humidity=43.1%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=26.2* Humidity=42.5%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=26.2* Humidity=43.1%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=26.3* Humidity=43.2%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
^[[A
Temp=26.3* Humidity=43.4%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=26.3* Humidity=42.9%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=26.2* Humidity=42.8%
pi@raspberrypi ~/Adafruit_Python_DHT/examples $
```

รูปที่ 3.3 ค่าที่ได้จากการวัดอุณหภูมิและความชื้นจากเซนเซอร์

คำสั่งในโค้ดนี้จะทำการวัดค่าอุณหภูมิและความชื้นด้วยเซนเซอร์ Temperature & Humidity – DHT22 โดยให้แสดงผลขึ้นที่หน้าจอว่าอุณหภูมิและความชื้น ณ เวลาที่วัดมีค่าอุณหภูมิและความชื้นเท่าไร



3.2.4 การทำงานของระบบโดยรวม

```
#!/usr/bin/python

from line import LineClient
import ssl
from line import LineClient
import sys
import time
import Adafruit_DHT
import wiringpi2 as wiringpi
import sys
from time import sleep
import os

sensor = Adafruit_DHT.DHT22
pin = 4

try:
    client = LineClient("happiness_you@hotmail.com","gai036300182")
    #authToken = 'AUTHTOKEN'
    #client = LineClient(authToken="authToken")
except:
    print "Login Failed"
```

```

wiringpi.wiringPiSetup()
wiringpi.pinMode(5,0)
wiringpi.pinMode(6,1)

while True:
    status = wiringpi.digitalRead(5)
    if status == 1 :
        os.system('fswebcam image.jpg')
        wiringpi.digitalWrite(6,1)
        print status
        humidity,temperature = Adafruit_DHT.read_retry(sensor, pin)
        if humidity is not None and temperature is not None:
            if temperature >25.00 :
                newT = temperature-0.3
                newH = humidity-6.51
                print 'Temp={0:0.1f}* Humidity={1:0.1f}%'.format(temperature, humidity)
                print 'TempC={0:0.1f}* HumidityC={1:0.1f}%'.format(newT, newH)
                group = client.groups[2]
                print client.groups
                group.sendMessage("DOOR CLOSE")
                group.sendImage('./image.jpg')
                group.sendMessage("Temp={0:0.1f}*C Humidity={1:0.1f}% \n".format(temperature,humidity))
                group.sendMessage("TempC={0:0.1f}*C HumidityC={1:0.1f}% \n".format(newT,newH))
                time.sleep(20)
            if temperature <25.00 :
                newT = temperature+0.3
                newH = humidity+6.51
                print 'Temp={0:0.1f}* Humidity={1:0.1f}%'.format(temperature, humidity)
                print 'TempC={0:0.1f}* HumidityC={1:0.1f}%'.format(newT, newH)
                group = client.groups[2]
                print client.groups
                group.sendMessage("DOOR CLOSE")
                group.sendImage('./image.jpg')
                group.sendMessage("Temp={0:0.1f}*C Humidity={1:0.1f}% \n".format(temperature,humidity))
                group.sendMessage("TempC={0:0.1f}*C HumidityC={1:0.1f}% \n".format(newT,newH))
                time.sleep(20)

```

```

while temperature >=28 :
    humidity,temperature = Adafruit_DHT.read_retry(sensor, pin)
    if temperature >25.00 :
        newT = temperature-0.3
        newH = humidity-6.51
        print "Alert"
        print 'Temp={0:0.1f} * Humidity={1:0.1f}%'.format(temperature, humidity)
        print 'TempC={0:0.1f} * HumidityC={1:0.1f}%'.format(newT, newH)
        group = client.groups[2]
        print client.groups
        group.sendMessage("Still Close")
        group.sendMessage("Temp={0:0.1f} *C Humidity={1:0.1f}% \n Please check your device".format(temperature,humidity))
        group.sendMessage("TempC={0:0.1f} *C HumidityC={1:0.1f}% \n".format(newT,newH))
        humidity,temperature = Adafruit_DHT.read_retry(sensor, pin)
    if temperature <25.00 :
        newT = temperature+0.3
        newH = humidity+6.51
        print "Alert"
        print 'Temp={0:0.1f} * Humidity={1:0.1f}%'.format(temperature, humidity)
        print 'TempC={0:0.1f} * HumidityC={1:0.1f}%'.format(newT, newH)
        group = client.groups[2]
        print client.groups
        group.sendMessage("Still Close")
        group.sendMessage("Temp={0:0.1f} *C Humidity={1:0.1f}% \n Please check your device".format(temperature,humidity))
        group.sendMessage("TempC={0:0.1f} *C HumidityC={1:0.1f}% \n".format(newT,newH))
        humidity,temperature = Adafruit_DHT.read_retry(sensor, pin)
        time.sleep(20)

if temperature <27.9:

    exit

```

คำสั่งใน Loop นี้จะทำการเช็คว่ามีคนเปิดประตูเข้า ออกหรือไม่-และเมื่อมีคนเปิดประตูเข้าออกค่าอุณหภูมิและความชื้นภายในคลั่งขามีการเปลี่ยนแปลงเกิน $\text{Temperature} \geq 28^{\circ}\text{C}$, $\text{Humidity} \geq 60\%$ หรือไม่ ถ้าไม่มีคนเปิดประตูเข้ามาแสดงว่าค่าสถานะของประตูเป็น “1” และจะทำการส่งรูปภาพ สถานะของประตู ว่า “DOOR CLOSE” และค่าอุณหภูมิความชื้น-ที่ผ่านการปรับเทียบแล้วเข้าแอปพลิเคชัน Line ทุกๆ 20 วินาที ถ้าไม่มีคนเปิดประตูเข้ามาแล้วค่า $\text{Temperature} \geq 28^{\circ}\text{C}$, $\text{Humidity} \geq 60\%$ ระบบจะทำการส่งข้อความ [“Alert” บอกค่าอุณหภูมิ ขึ้น และความ-“Please check your device”] เข้าแอปพลิเคชัน Line ทุกๆ 20 วินาที

else :

```

    os.system('fswebcam image.jpg')

    wiringpi.digitalWrite(6,0)

    print status

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

    if humidity is not None and temperature is not None:

if temperature >25.00 :

    newT = temperature-0.3
    newH = humidity-6.51
    print 'Temp={0:0.1f} * Humidity={1:0.1f}%'.format(temperature, humidity)
    print 'TempC={0:0.1f} * HumidityC={1:0.1f}%'.format(newT, newH)
    group = client.groups[2]
    print client.groups
    group.sendMessage("DOOR OPEN")
    group.sendImage('./image.jpg')
    group.sendMessage("Temp={0:0.1f} *C Humidity={1:0.1f}% \n ".format(temperature,humidity))
    group.sendMessage("TempC={0:0.1f} *C HumidityC={1:0.1f}% \n ".format(newT,newH))
    time.sleep(20)

if temperature <25.00 :

    newT = temperature+0.3
    newH = humidity+6.51

    print 'Temp={0:0.1f} * Humidity={1:0.1f}%'.format(temperature, humidity)
    print 'TempC={0:0.1f} * HumidityC={1:0.1f}%'.format(newT, newH)

    group = client.groups[2]

    print client.groups

    group.sendMessage("DOOR OPEN")

    group.sendImage('./image.jpg')

    group.sendMessage("Temp={0:0.1f} *C Humidity={1:0.1f}% \n ".format(temperature,humidity))
    group.sendMessage("TempC={0:0.1f} *C HumidityC={1:0.1f}% \n ".format(newT,newH))

    time.sleep(20)

```

```

while temperature >=28 :
    humidity,temperature = Adafruit_DHT.read_retry(sensor, pin)
    if temperature >25.00 :
        newT = temperature-0.3
        newH = humidity-6.51
        print "Alert"
        print 'Temp={0:0.1f} * Humidity={1:0.1f}%'.format(temperature, humidity)
        print 'TempC={0:0.1f} * HumidityC={1:0.1f}%'.format(newT, newH)
        group = client.groups[2]
        print client.groups
        group.sendMessage("Still Open")
        group.sendMessage("Temp={0:0.1f} *C Humidity={1:0.1f}% \n Please check your device".format(temperature,humidity))
        group.sendMessage("TempC={0:0.1f} *C HumidityC={1:0.1f}% \n".format(newT,newH))
        humidity,temperature = Adafruit_DHT.read_retry(sensor, pin)
        time.sleep(20)
    if temperature <25.00 :
        newT = temperature+0.3
        newH = humidity+6.51
        print "Alert"
        print 'Temp={0:0.1f} * Humidity={1:0.1f}%'.format(temperature, humidity)
        print 'TempC={0:0.1f} * HumidityC={1:0.1f}%'.format(newT, newH)
        group = client.groups[2]
        print client.groups
        group.sendMessage("Still Close")
        group.sendMessage("Temp={0:0.1f} *C Humidity={1:0.1f}% \n Please check your device".format(temperature,humidity))
        group.sendMessage("TempC={0:0.1f} *C HumidityC={1:0.1f}% \n".format(newT,newH))
        humidity,temperature = Adafruit_DHT.read_retry(sensor, pin)
        time.sleep(20)

if temperature <27.9:

    exit

```

คำสั่งใน Loop นี้จะทำการเช็คว่ามีคนเปิดประตูเข้าออกหรือไม่ และ-เมื่อมีคนเปิดประตูเข้าออกค่าอุณหภูมิและความชื้นภายในคลังยามีการเปลี่ยนแปลงเกิน $\text{Temperature} \geq 28^{\circ}\text{C}$, $\text{Humidity} \geq 60\%$ หรือไม่ ถ้ามีคนเปิดประตูเข้ามาแสดงว่าค่าสถานะของประตูเป็น “0” และค่าอุณหภูมิความชื้น-ไม่เปลี่ยนแปลงที่กำหนดระบบจะทำการส่งรูปภาพ สถานะของประตูว่า “DOOR OPEN” และค่าอุณหภูมิความชื้น-ที่ผ่านการปรับเทียบแล้วเข้าแอปพลิเคชัน Line ทุกๆ 20 วินาที แต่ถ้าค่า $\text{Temperature} \geq 28^{\circ}\text{C}$, $\text{Humidity} \geq 60\%$ ระบบจะทำการส่งข้อความ [“Alert” บอกค่าอุณหภูมิ ความชื้น และ-“Please check your device”] เข้าแอปพลิเคชัน Line ทุกๆ 20 วินาที



บทที่ 4

ผลการทดลองและทดสอบการใช้งานจริง

4.1 กล่าวนำ

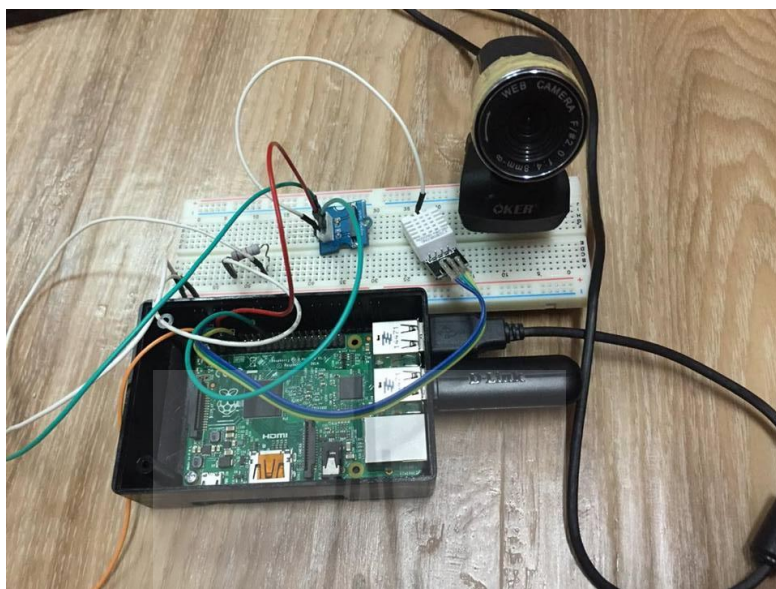
บทนี้จะกล่าวถึงการทดสอบอุปกรณ์วัดค่าอุณหภูมิและความชื้นที่วัดได้จากเซนเซอร์DHT22จะมีค่าเทียบเคียงกับค่าอุณหภูมิและความชื้นที่ได้จากเครื่องDIGICON TH-02 หรือไม่ อย่างไร และถ้าอุณหภูมิและความชื้นที่วัดได้มีค่าสูงจะทำการปรับเทียบให้มีค่าใกล้เคียงอย่างไร

4.2 อุปกรณ์การทดสอบชิ้นงาน

ผู้จัดทำได้สร้างโมเดลจำลองสำหรับการวัดค่าอุณหภูมิและความชื้น ตามรูปที่ 4.1 โดยภายในกล่องโมเดลมีวงจรการทำงานคอยควบคุมการวัดค่าอุณหภูมิและความชื้นตามรูปที่ 4.2 จากรูปเห็นว่าทางผู้จัดทำได้เจาะรูโมเดลเพื่อสอดสายไฟที่ต่อเข้ากับเซนเซอร์วัดค่าอุณหภูมิและความชื้น)DHT22 (ขึ้นมาด้านบนกล่องเพื่อความเป็นระเบียบเรียบร้อยและสังเกตได้ง่ายอีกด้วย



รูปที่ 4.1 โมเดลจำลองสำหรับการวัดค่าอุณหภูมิและความชื้น



รูปที่ 4.2 รูปแบบการต่อวงจร

4.3 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์การทดลองดังรูปที่ 4.2
2. ทดลองเปิด-ปิดประตูตามขั้นตอนโดยดูที่สถานะของการเปิด-ปิดประตูว่ามีสถานะเป็น “0” หรือ “1” หรือไม่ ต่อมาทำการวัดค่าอุณหภูมิและความชื้นว่ามีค่าเป็นอย่างไร ตรงตามที่ได้กำหนดเงื่อนไขหรือไม่
3. บันทึกค่าอุณหภูมิและความชื้นที่แสดงบน Serial Monitor และที่แสดงบนหน้าจอของเครื่อง DIGICON TH-02
4. ทำการทดลองซ้ำอีก 20 ครั้ง
5. บันทึกผลการทดลองและนำมาหาผลเฉลี่ย

4.4 ผลการทดลอง



รูปที่ 4.3 ผลการทดลองที่ส่งเข้าแอปพลิเคชัน LINE

ครั้งที่	ค่าที่บันทึกได้จากเครื่อง DIGICON TH-02		ค่าที่บันทึกได้จาก Serial Monitor		%error	
	Temperature	Humidity	Temperature	Humidity	Temperature	Humidity
1	20.7°	65%	21.3°	56%	2.89%	13.84%
2	20.7°	65%	21.8°	63%	5.31%	3.07%
3	20.7°	65%	21.5°	61.5%	3.86%	5.38%
4	20.8°	65%	21.6°	62.9%	3.84%	3.23%
5	20.9°	65%	21.5°	61.8%	2.87%	4.92%
6	20.9°	65%	21.3°	63.2%	1.91%	2.76%
7	20.9°	65%	21.2°	64.1%	1.43%	1.38%
8	20.9°	65%	21.2°	64.1%	1.43%	1.38%
9	20.9°	65%	21.2°	63.6%	1.43%	2.15%
10	21.0°	65%	21.2°	63.3%	0.952%	2.61%
11	21.0°	65%	21.2°	63%	0.952%	3.07%
12	21.0°	65%	21.2°	63.2%	0.952%	2.76%
13	21.0°	65%	21.1°	63.5%	0.476	2.31%
14	21.0°	65%	21.2°	63.8%	0.952%	1.48%
15	21.0°	65%	21.2°	64.2%	0.952%	1.23%
16	21.0°	65%	21.2°	63.5%	0.952%	2.31%
17	21.0°	65%	22.0°	67%	4.76%	3.07%
18	21.0°	65%	22.3°	61.3%	6.19%	5.69%
19	21.0°	65%	22.3°	61.2%	6.19%	5.84%
20	21.0°	65%	22.2°	61.2%	5.71%	5.85%
ค่าเฉลี่ยของค่าที่บันทึกได้จากSerial Monitor			21.48°	62.77%	2.70%	3.71%

ตารางที่ 4.1 ตารางแสดงผลการทดลองบน Serial Monitor และที่เครื่อง DIGICON TH-02

4.5 วิเคราะห์ผลการทดลอง

จากรูปแสดงผลการทดลองที่ 4.2 เป็นตัวอย่างการแสดงผลทาง Serial Monitor และที่เครื่อง DIGICON TH-02 โดยค่าอุณหภูมิและความชื้นที่ได้จากการแสดงผลทาง Serial Monitor นั้นจะปรากฏค่าให้เห็น คือ

- สถานะของการเปิด-ปิดประตู
- ค่าอุณหภูมิและความชื้นที่ยังไม่ผ่านการปรับเทียบ
- ค่าอุณหภูมิและความชื้นที่ผ่านการปรับเทียบแล้ว

ซึ่งค่าอุณหภูมิและความชื้นที่ผ่านการปรับเทียบแล้ว มาจากการคำนวณดังนี้

อุณหภูมิ ;

ถ้าอุณหภูมิ	21.48°	ทำการปรับเทียบ	0.3
ถ้าอุณหภูมิ	21.00°	ทำการปรับเทียบ	0.31

ความชื้น ;

ถ้าความชื้น	62.77%	ทำการปรับเทียบ	6.75
ถ้าความชื้น	65.00%	ทำการปรับเทียบ	6.51

ซึ่งค่าอุณหภูมิและความชื้นที่ผ่านการปรับเทียบแล้วมีความใกล้เคียงกับอุณหภูมิและความชื้นที่อ่านได้จากเครื่อง DIGICON TH-02

4.6 สรุปผลการทดลอง

จากตารางผลการทดลอง สรุปว่าค่าอุณหภูมิและความชื้นที่ผ่านการปรับเทียบแล้วมีความใกล้เคียงกับค่าอุณหภูมิและความชื้นที่อ่านได้จากเครื่อง DIGICON TH-02

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ระบบควบคุมคุณภาพคลังยาโดยใช้บอร์ด Raspberry Pi จะทำงานเริ่มต้นโดยการตรวจสอบสถานะของการเปิด-ปิดประตูว่ามีค่าเป็น “0” หรือ “1” และเมื่อได้ค่าสถานะของการเปิด-ปิดประตูขึ้นตอนต่อมากล้องจะทำการบันทึกภาพบุคคลที่ผ่านเข้าออกไว้และทำการส่งกลับมาให้บอร์ด Raspberry Pi ขึ้นตอนต่อมาระบบจะทำการวัดและตรวจสอบว่าค่าอุณหภูมิและความชื้นที่ได้จากเซนเซอร์ DHT22 เป็นตามเงื่อนไข $\text{Temperature} \geq 28^{\circ}\text{C}$, $\text{Humidity} \geq 60\%$ หรือไม่ เมื่อเซนเซอร์ DHT22 วัดค่าแล้วจะส่งค่าที่วัดได้กลับมาที่บอร์ด Raspberry Pi เพื่อให้บอร์ด Raspberry Pi ทำการเปรียบเทียบค่าที่ได้และเมื่อบอร์ด Raspberry Pi ทำการเปรียบเทียบค่าอุณหภูมิและความชื้นแล้วระบบจะส่งข้อมูลที่ได้ทั้งหมดเข้ากลุ่มที่ได้กำหนดไว้แล้วในแอปพลิเคชัน LINE

5.2 ปัญหาและอุปสรรค

1. การเชื่อมต่อกับ SUT Mobile ยังคงไม่สามารถเชื่อมต่อได้อย่างเสถียรเนื่องจากภายในห้องคลังยาเป็นจุดอับสัญญาณ
2. ความแม่นยำในการวัดจากสภาพแวดล้อมจริงและเซนเซอร์วัดอุณหภูมิและความชื้นยังมีความคลาดเคลื่อนอยู่เพียงเล็กน้อย

5.3 แนวทางการแก้ไขปัญหา

1. ติดตั้งเครื่องส่งสัญญาณอินเทอร์เน็ตภายในห้องคลังยาหรือจุดที่ง่ายต่อการเชื่อมต่อ
2. ทำการปรับเทียบค่าใหม่ให้มีความใกล้เคียงค่าจริงมากที่สุดเท่าที่จะทำได้

5.4 ข้อเสนอแนะเพิ่มเติม

ในอนาคตจะทำการสั่งงานจากแอปพลิเคชัน LINE กลับมายังบอร์ด Raspberry Pi เพื่อให้ระบบทำการปรับหรือเพิ่มค่าอุณหภูมิและความชื้น เมื่อเกิดเหตุการณ์ที่ว่าค่าอุณหภูมิและความชื้นที่ค่าต่ำหรือสูงเกินกว่าที่ได้ตั้งเงื่อนไขไว้เพื่อเพิ่มความสะดวกรวดเร็วแก่เกษตรกรที่ควบคุมดูแลคังยา

5.5 สิ่งที่ได้รับจากการทำโครงการ

1. เข้าใจหลักการทำงานของบอร์ด Raspberry Pi
2. เข้าใจหลักการทำงานของเซนเซอร์ที่ใช้ในการทำโครงการ
3. เข้าใจการเขียนโปรแกรมและโหลดโปรแกรมให้สามารถใช้งานได้
4. สามารถใช้งานระบบควบคุมคุณภาพคังยาโดยใช้บอร์ด Raspberry Pi ได้จริงในห้องคังยาของโรงพยาบาลมหาวิทยาลัยเทคโนโลยีสุรนารี



บรรณานุกรม

[1]*Raspberry Pi 2 Model*, สืบค้นจาก: <http://www.arduitronics.com/product/722/raspberry-pi-2-model-b>

[2]*Raspberry Pi 2 Model*, สืบค้นจาก: <http://www.thaieasyelec.com/products/development-boards/raspberry-pi-2-model-b-detail.html>

[3]*i Module - Magnatic Switch*, สืบค้นจาก : <http://www.thaieasyelec.com/products/sensors/i-module-magnatic-switch-v1-0-detail.html>

[4]*Temperature&Humidity Sensor Pro - DHT22*, สืบค้นจาก:
<http://www.thaieasyelec.com/products/sensors/humidity-temperature/grove-temperature-humidity-sensor-pro-dht22-detail.html>

[5]กลุ่มงานเภสัชกรรม โรงพยาบาลเมตตาประชารักษ์(วัดไร่ขิง).การเก็บรักษายา.11 มิถุนายน 2556,
สืบค้นจาก: http://mettapharmacy.blogspot.com/2013/06/blog-post_2128.html

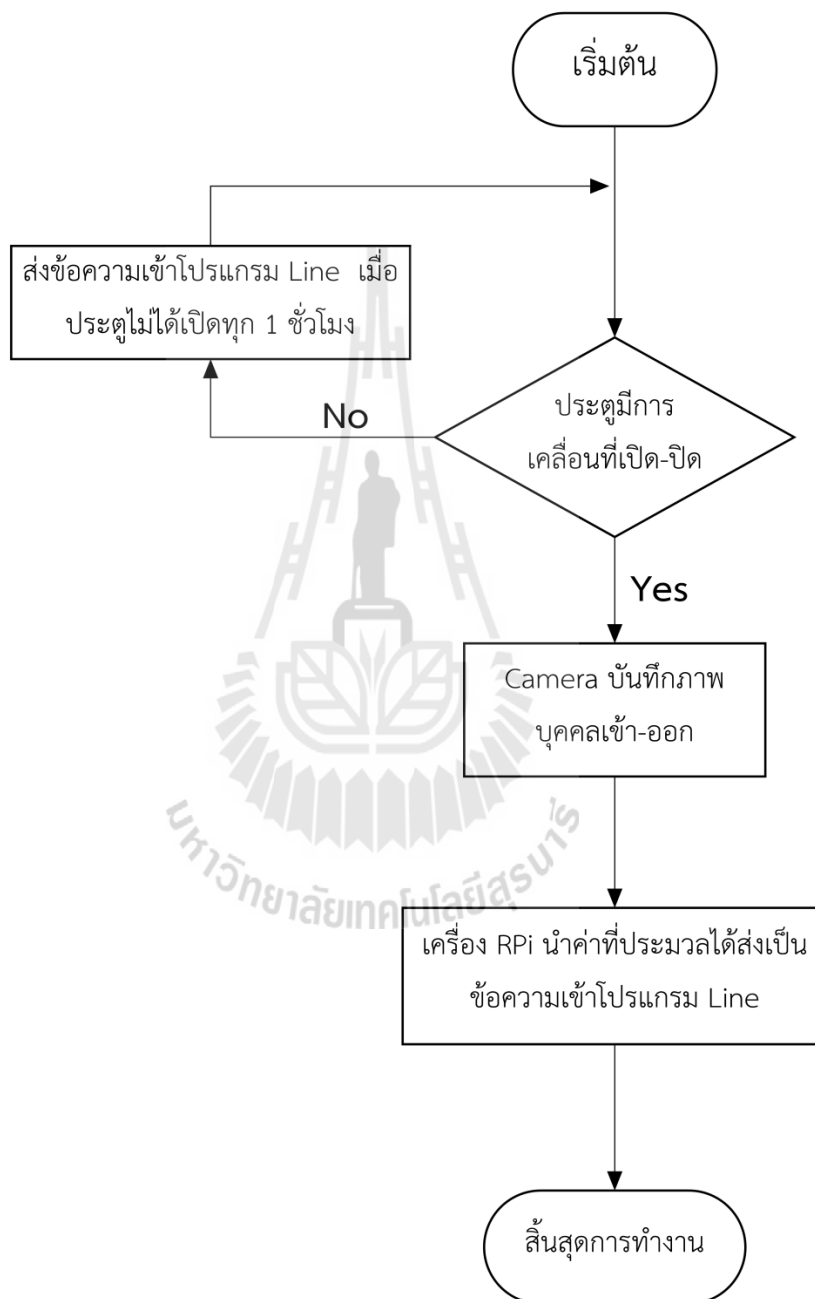
[6]ฝ่ายการพยาบาล โรงพยาบาลมหาราชนครเชียงใหม่.วิธีการเก็บรักษาผลิตภัณฑ์ยา.3 กรกฎาคม 2550,
สืบค้นจาก:

<http://www.med.cmu.ac.th/hospital/nis/km/cops/detail.php?sid=35&Room=warfarin&search=&page=1>

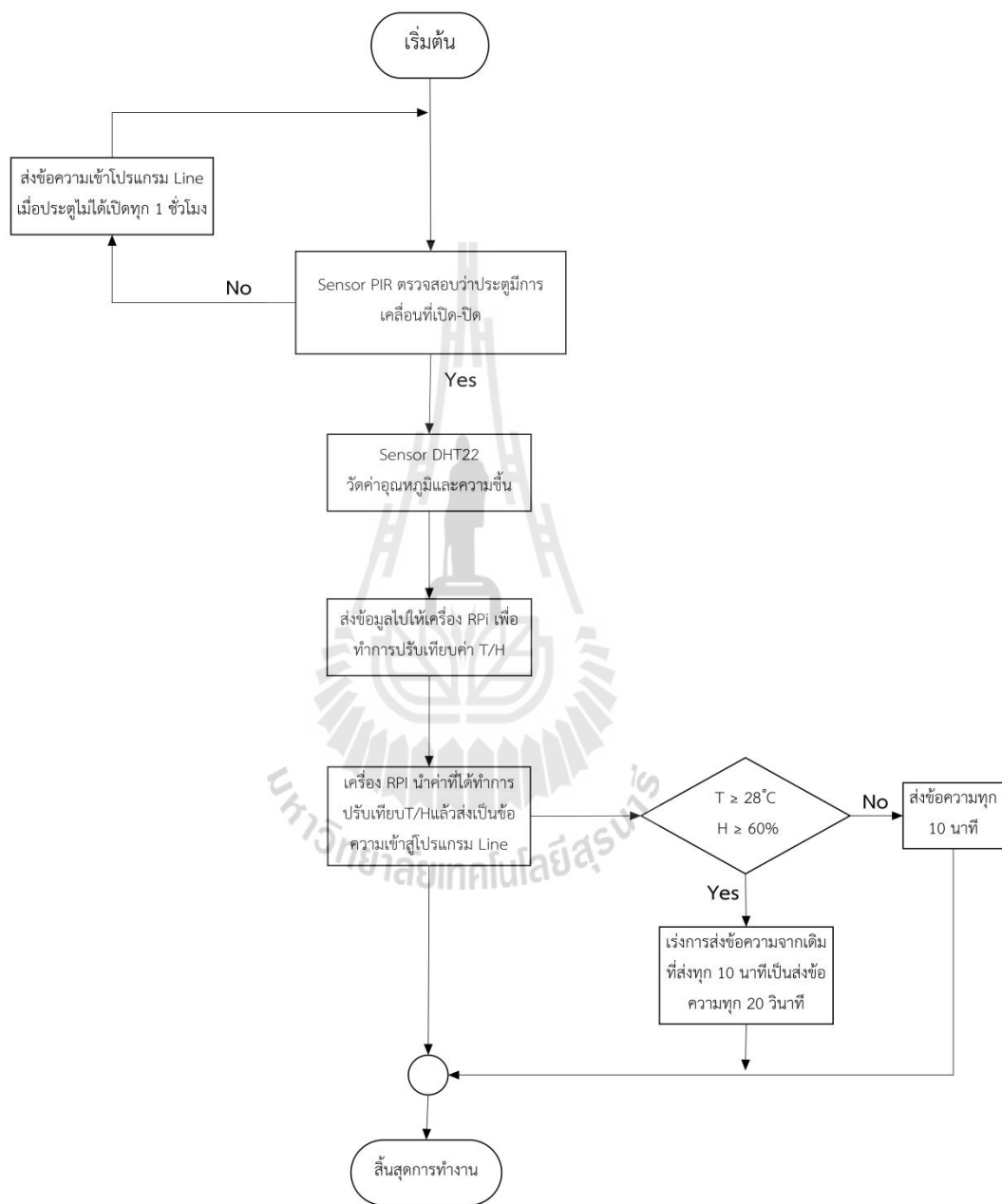
ภาคผนวก



ขั้นตอนการทำงานของระบบเปิด-ปิดประตู



ขั้นตอนการทำงานของการวัดค่าอุณหภูมิและความชื้น



คำสั่งที่ใช้ในการทดลองวัดค่าอุณหภูมิและความชื้น ณ สถานที่จริง

```
#!/usr/bin/python

from line import LineClient
import ssl

from line import LineClient
import sys
import time
import Adafruit_DHT
#import wiringpi2 as wiringpi
import sys
from time import sleep
import os

sensor = Adafruit_DHT.DHT22
pin = 4

try:
    client = LineClient("happiness_you@hotmail.com","gai036300182")
    #authToken = 'AUTHTOKEN'
    #client = LineClient(authToken="authToken")
except:
    print "Login Failed"

while True:
    os.system('fswebcam image.jpg')
    humidity,temperature = Adafruit_DHT.read_retry(sensor, pin)
    if humidity is not None and temperature is not None:
```

```

print 'Temp={0:0.1f}* Humidity={1:0.1f}%'.format(temperature, humidity)

    group = client.groups[2]

print client.groups

group.sendImage('./image.jpg')

group.sendMessage("Temp={0:0.1f}*C Humidity={1:0.1f}%
\n".format(temperature,humidity))

time.sleep(20)

while temperature >=28 :

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

    #time.sleep(20)

    print "Alert"

    print 'Temp={0:0.1f}* Humidity={1:0.1f}%'.format(temperature, humidity)

    group = client.groups[2]

    print client.groups

    group.sendMessage("Alert \n Temp={0:0.1f}*C Humidity={1:0.1f}% \n Please check your
device".format(temperature,humidity))

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

    time.sleep(20)

    if temperature <27.9:

        exit

```

ประวัติผู้เขียน



นางสาวอริรัตน์ วงศ์จันทร์ชมภู รหัสนักศึกษา B5513345
เกิดเมื่อวันที่ 16 มกราคม พ.ศ. 2537
ภูมิลำเนาอยู่ที่ ตำบลหม่ม่น อำเภอเมือง จังหวัดอุดรธานี
สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนสตรีราชินูทิศ
อำเภอเมือง จังหวัดอุดรธานี เมื่อปีการศึกษา 2554
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



นายวัชรพัฒน์ กองนาค รหัสนักศึกษา B5538713
เกิดเมื่อวันที่ 28 กุมภาพันธ์ พ.ศ. 2537
ภูมิลำเนาอยู่ที่ ตำบลบางขโมด อำเภอบ้านหมอ จังหวัดสระบุรี
สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนวินิตศึกษาในพระราชูปถัมภ์
อำเภอเมือง จังหวัดลพบุรี เมื่อปีการศึกษา 2554
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



นาย ภาณุวัฒน์ โชติรโส รหัสนักศึกษา B553962
เกิดเมื่อวันที่ 28 ตุลาคม พ.ศ. 2536
ภูมิลำเนาอยู่ที่ ตำบลพิชัย อำเภอเมือง จังหวัดลำปาง
สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนอัสสัมชัญลำปาง
อำเภอเมือง จังหวัด ลำปาง เมื่อปีการศึกษา 2554
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี