# การออกแบบแอปพลิเคชันบนสมาร์ทโฟนสำหรับควบคุมระบบการให้น้ำพืช

# Design of a Smartphone application for Irrigation System Control

ธนวัฒน์ กมลจารุพิศุทธิ์
ปิยะณัฐ สูญพานิช
พงศ์ธร ญาติพร้อม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมเกษตร
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

# การออกแบบแอปพลิเคชันบนสมาร์ทโฟนสำหรับควบคุมระบบการให้น้ำพืช
# Design of a Smartphone application for Irrigation System Control

ธนวัฒน์ กมลจารุพิศุทธิ์
ปิยะณัฐ สูญพานิช
พงศ์ธร ญาติพร้อม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมเกษตร คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# DESING OF A SMARTPHONE APPLICATION FOR IRRIGATION SYSTEM CONTROL

THANAWAT KAMONCHARUPISUT
PIYANAT SOONPANICH
PONGTHORN YARTPROM

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN AGRICULTURAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2557
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

.....

**หัวข้อปริญญานิพนธ์** การออกแบบแอปพลิเคชันบนสมาร์ทโฟนสำหรับควบคุมระบบการให้น้ำพืช
Design of a Smartphone application for Irrigation System Control

**นักศึกษาผู้จัดทำ**

| | |
|---|---|
| นายธนวัฒน์ กมลจารุพิศุทธิ์ | รหัสประจำตัว 54010561 |
| นายปิยะณัฐ สูญพานิช | รหัสประจำตัว 54010821 |
| นายพงศ์ธร ญาติพร้อม | รหัสประจำตัว 54010849 |

**ปริญญา** วิศวกรรมศาสตรบัณฑิต (วิศวกรรมเกษตร)

**หลักสูตร** วิศวกรรมเกษตร

**สาขาวิชา** วิศวกรรมเครื่องกล

**ปีการศึกษา** 2557

| อาจารย์ผู้ควบคุมปริญญานิพนธ์ | ลายมือชื่อ |
|---|---|
| ผศ.ดร.ทรงวุฒิ แสงจันทร์ | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หัวข้อปริญญานิพนธ์** การออกแบบแอปพลิเคชันบนสมาร์ทโฟนสำหรับควบคุมระบบการให้น้ำพืช
Design of a Smartphone application for Irrigation System Control

**นักศึกษาผู้จัดทำ** นายธนวัฒน์ กมลจารุพิศุทธิ์ 54010561
นายปิยะณัฐ สูญพานิช 54010821
นายพงศ์ธร ญาติพร้อม 54010849

**อาจารย์ที่ปรึกษา** ผศ.ดร.ทรงวุฒิ แสงจันทร์

ปีการศึกษา 2557

## บทคัดย่อ

การศึกษาวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบแอปพลิเคชันสำหรับควบคุมการให้น้ำพืช โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวประมวลผล ในการควบคุมและสั่งการให้น้ำอัตโนมัติ ซึ่งสามารถปรับตั้งค่าได้ตามความต้องการ โดยระบบมีส่วนประกอบ 2 ส่วน คือ ส่วนควบคุมการทำงานหลัก และส่วนตรวจวัดความชื้น ซึ่งทำงานโดยระบบไร้สายระหว่างชุดเซนเซอร์ในแปลงกับชุดควบคุมหลัก ได้ทำการทดลองโดยใช้เซนเซอร์ตรวจจับความชื้นในดิน ซึ่งได้จากการวัดกระแสไฟฟ้าที่ส่งไปยังเซนเซอร์ โดยที่กระแสไฟฟ้าจะมากน้อยนั้นขึ้นอยู่กับความชื้นของดิน ไมโครคอนโทรลเลอร์ทำหน้าที่รับค่าความต่างศักย์ ($v_{OUT}$) จากกระแสไฟฟ้า DC ที่ส่งออกไปเพื่อเข้าสู่กระบวนการ ADC Conversion ซึ่งจะทำการแปลงค่าจาก Analog เป็น Digital และบันทึกค่าไว้ในไมโครคอนโทรลเลอร์ และยังได้ทำการวัดค่าตัวแปรอื่นๆ ที่อาจมีผลต่อการตรวจจับความชื้นในดิน เช่น อุณหภูมิ ความชื้นในอากาศ ทั้งนี้เพื่อนำข้อมูลไปใช้ อ้างอิงสำหรับควบคุมการให้น้ำ โดยสร้างระบบโครงข่ายที่สามารถทำงานประสานกันได้อย่างมีประสิทธิภาพ ซึ่งสามารถควบคุมสั่งการพร้อมทั้งแสดงค่าความชื้นในดิน, ความชื้นอากาศ และอุณหภูมิจากเซนเซอร์ผ่านแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|---|---|---|
| **Thesis Title** | Design of a Smartphone application for Irrigation System Control | |
| **Authors** | Thanawat Kamonchacupisut | 54010561 |
| | Piyanat Soonpanich | 54010821 |
| | Pongthorn Yartprom | 54010849 |
| **Thesis Advisor** | Asst. Prof. Dr. Songvoot Saengchan | |
| **Year** | 2014 | |


## Abstract

. The objective of this project was to design an application on smartphone for irrigation control. Automatic irrigation system was controlled by microcontroller which can be adjusted any requirements. The system consists of two main parts; control unit and soil moisture sensor which will be connected by using wireless system. The electric current of sensor is depend on amount of soil moisture. Microcontroller receives the voltage ($V_{out}$) from DC current and sends to ADC Conversion which can convert signal from analog to digital and record it in microcontroller. The other variables were measured which may effect to soil moisture sensor such as temperature and humidity. There will be used as references for irrigation control. The network system has done with good efficiency. It can control the irrigation system with show soil moisture value, humidity and temperature through an application on smartphone.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ ผศ.ดร.ทรงวุฒิ แสงจันทร์ อาจารย์ที่ปรึกษาที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก ขอขอบพระคุณคณาจารย์สาขาวิศวกรรมเกษตรทุกท่าน ที่เอื้อเฟื้อสถานที่ ให้คำปรึกษา รวมถึงเครื่องมืออุปกรณ์ต่างๆ ในการทดลองตลอดงานวิจัย

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นายธนวัฒน์ กมลจารุพิศุทธิ์
นายปิยะณัฐ สูญพานิช
นายพงศ์ธร ญาติพร้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ


| | หน้า |
|---|---|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญตาราง | VII |
| สารบัญภาพ | VIII |
| บทที่ 1 บทนำ | 1 |
| 1.1 ที่มาและความสำคัญ | 1 |
| 1.2 วัตถุประสงค์ | 1 |
| 1.3 ขอบเขตการศึกษา | 1 |
| 1.4 ผลที่คาดว่าจะได้รับ | 1 |
| บทที่ 2 หลักการและทฤษฎี | 2 |
| 2.1 ความชื้นในดิน | 2 |
| 2.1.1 ระดับความชื้นที่สำคัญของดิน | 2 |
| 2.1.2 การตรวจวัดความชื้นในดิน | 3 |
| 2.2 การให้น้ำแบบฉีดฝอย | 4 |
| 2.2.1 ระบบให้น้ำแบบฉีดฝอย | 5 |
| 2.2.2 อุปกรณ์การให้น้ำแบบฉีดฝอย | 5 |
| 2.2.3 การออกแบบระบบการให้น้ำ | 5 |
| 2.3 ไมโครคอนโทรลเลอร์ | 8 |
| 2.3.1 หลักการทั่วไปของไมโครคอนโทรลเลอร์ | 8 |
| 2.3.2 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์ | 8 |
| 2.3.3 Arduino UNO R3 | 9 |
| 2.3.4 Raspberry Pi | 11 |
| 2.4 DHT22 Humidity and Temperature Sensor | 12 |
| 2.5 ทฤษฎี Sensor สำหรับตรวจวัดความชื้นในดิน | 13 |
| 2.6 โมดูล nRF24L01 | 14 |
| 2.7 Relay | 14 |
| 2.8 Real Time Clock (RTC) | 15 |
| 2.9 โปรแกรม PuTTy | 16 |
| 2.10 โปรแกรม WinSCP | 17 |
| 2.11 ภาษาที่ใช้ในการเขียนโปรแกรมและคำสั่ง | 17 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)


| | หน้า |
|---|---|
| 2.9.1 ภาษา Python | 17 |
| 2.9.2 ภาษา C | 18 |
| 2.9.3 ภาษา HTML | 18 |
| 2.9.4 ภาษา JavaScript | 19 |
| 2.9.5 ภาษา SQLite | 19 |
| 2.9.7 Apache | 19 |
| บทที่ 3 วิธีและขั้นตอนการดำเนินการ | 20 |
| 3.1 องค์ประกอบของระบบ | 20 |
| 3.1.1 ส่วนฮาร์ดแวร์ | 20 |
| 3.1.2 ส่วนซอฟต์แวร์ | 21 |
| 3.2 การออกแบบฮาร์ดแวร์ | 21 |
| 3.2.1 ส่วนตรวจวัดความชื้น (Sensor Node) | 21 |
| 3.2.2 ส่วนควบคุมการทำงานหลัก (Base Station) | 22 |
| 3.3 ซอฟต์แวร์และระบบการทำงาน | 24 |
| 3.3.1 ซอฟแวร์ที่ใช้ในการพัฒนา | 25 |
| 3.3.2 การออกแบบซอฟแวร์ | 26 |
| 3.4 การทดลองและวิเคราะห์ข้อมูล | 31 |
| 3.4.1 การหาค่าความชื้นในดินที่ Field Capacity | 31 |
| 3.4.2 การหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า | 32 |
| บทที่ 4 ผลการทดลอง | 33 |
| 4.1 การต่อวงจรของส่วนตรวจวัดความชื้น | 33 |
| 4.1.1 การต่อโมดูล DHT 22 เข้ากับ Arduino | 33 |
| 4.1.2 การต่อ Soil moisture sensor เข้ากับ Arduino | 34 |
| 4.1.3 การต่อโมดูลnRF24L01 เข้ากับ Arduino | 34 |
| 4.2 การต่อวงจรของส่วนควบคุมการทำงานหลัก | 36 |
| 4.3 การออกแบบซอฟแวร์บน Raspberry Pi | 37 |
| 4.3.1 ซอฟแวร์ส่วนรับผลจากเซ็นเซอร์ | 37 |
| 4.3.2 ซอฟแวร์ส่วนควบคุมรีเลย์ | 40 |
| 4.3.3 ซอฟแวร์ส่วนฐานข้อมูล | 41 |
| 4.3.4 ซอฟแวร์ส่วนเซิฟเวอร์และแสดงผลเวปเพจ | 42 |
| 4.4 การหาค่าความชื้นในดินที่ Field Capacity | 44 |
| 4.5 การหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า | 45 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)


| | หน้า |
|---|---|
| บทที่ 5 สรุปผลการทดลอง | 47 |
| 5.1 การต่อวงจรของส่วนตรวจวัดความชื้นและส่วนควบคุมการทำงานหลัก | 47 |
| 5.2 การหาค่าความชื้นในดินที่ Field Capacity | 47 |
| 5.3 การหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า | 47 |
| 5.4 ผลการทดสอบระบบการให้น้ำพืชผ่านแอปพลิเคชัน | 47 |
| 5.5 ข้อเสนอแนะ | 48 |
| เอกสารอ้างอิง | 49 |
| ภาคผนวก | 51 |
| ภาคผนวก ก ตารางผลการทดลอง | 52 |
| ภาคผนวก ข แอปพลิเคชันควบคุมระบบ | 54 |
| ภาคผนวค ค โค้ดโปรแกรมการทำงาน | 58 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


| | หน้า |
|---|---|
| ตารางที่ 2.1 ตารางแสดงความสามารถในการอุ้มน้ำที่เป็นประโยชน์ต่อพืชของดินประเภทต่างๆ | 6 |
| ตารางที่ 3.1 ซอฟแวร์ที่ใช้ในการพัฒนา | 24 |
| ตารางที่ 3.2 ส่วนฐานของข้อมูล | 29 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ


| | หน้า |
|---|---|
| รูปที่ 2.1 เครื่องมือวัดความชื้นดินทางวิทยาศาสตร์ | 4 |
| รูปที่ 2.2 Arduino UNO R3 ด้านหน้า | 10 |
| รูปที่ 2.3 Arduino UNO R3 ด้านหลัง | 11 |
| รูปที่ 2.4 Raspberry Pi | 11 |
| รูปที่ 2.5 ส่วนประกอบของ Raspberry Pi | 12 |
| รูปที่ 2.6 DHT22/AM2302 Humidity and Temperature Sensor | 12 |
| รูปที่ 2.7 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีทั้งหมด 5 ไบต์ | 13 |
| รูปที่ 2.8 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH | 13 |
| รูปที่ 2.9 เซนเซอร์ตรวจจับความชื้นดิน | 13 |
| รูปที่ 2.10 โมดูล nRF24L01 | 14 |
| รูปที่ 2.11 ขาต่างๆ ของโมดูล nRF24L01 | 14 |
| รูปที่ 2.12 รูปร่างและลักษณะของรีเลย์ | 15 |
| รูปที่ 2.13 หลักการทำงานเบื้องต้นของรีเลย์ | 15 |
| รูปที่ 2.14 แสดงหลักการทำงานเบื้องต้นของรีเลย์ | 16 |
| รูปที่ 3.1 องค์ประกอบของระบบ | 19 |
| รูปที่ 3.2 การต่อโมดูล DHT 22 เข้ากับArduino | 20 |
| รูปที่ 3.3 การต่อ Soil moisture sensor เข้ากับArduino | 21 |
| รูปที่ 3.4 การต่อโมดูลnRF24L01 เข้ากับArduino | 21 |
| รูปที่ 3.5 การต่อวงจรวัดแบตเตอรี่โดยใช้วงจร voltage divider | 22 |
| รูปที่ 3.6 การต่อ RTC และ nRF24l01 กับ Raspberry Pi | 23 |
| รูปที่ 3.7 การต่อรีเลย์ กับ Raspberry Pi | 23 |
| รูปที่ 3.8 ลำดับการทำงานของโปรแกรมส่วนฮาร์ดแวร์ทั้งหมด | 24 |
| รูปที่ 3.9 ลำดับการทำงานของโปรแกรมส่วนตรวจวัดความชื้น | 26 |
| รูปที่ 3.10 ลำดับการทำงานของส่วนรับข้อมูลจากเซ็นเซอร์ | 27 |
| รูปที่ 3.11 ลำดับการทำงานของส่วนควบคุมรีเลย์ | 28 |
| รูปที่ 3.12 ลำดับการทำงานของซอฟแวร์ส่วนเซิฟเวอร์และแสดงผลเวปแอฟพลิเคชัน | 30 |
| รูปที่ 3.13 อุปกรณ์สำหรับเก็บตัวอย่างดิน | 31 |
| รูปที่ 3.14 ลำดับขั้นตอนการลดความชื้นในดินเพื่อวัดแรงดันไฟฟ้า | 32 |
| รูปที่ 4.1 การต่อโมดูล DHT 22 | 33 |
| รูปที่ 4.2 การทดสอบการทำงานของโมดูล DHT 22 | 33 |
| รูปที่ 4.3 การต่อ Soil moisture sensor | 34 |
| รูปที่ 4.4 การทดสอบการทำงานของ Soil moisture sensor | |


เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)


| | หน้า |
|---|---|
| รูปที่ 4.5 การต่อโมดูลnRF24L01 บนบอร์ดไข่ปลา | 35 |
| รูปที่ 4.6 การทดสอบการทำงานของโมดูลnRF24L01 | 35 |
| รูปที่ 4.7 การต่อวงจรวัดแบตเตอรี่โดยใช้วงจร voltage divider | 35 |
| รูปที่ 4.8 วงจรของส่วนตรวจวัดความชื้น | 36 |
| รูปที่ 4.9 วงจรของส่วนควบคุมการทำงานหลัก | 36 |
| รูปที่ 4.10 โปรแกรม PuTTy | 37 |
| รูปที่ 4.11 โปรแกรม WinSCP | 37 |
| รูปที่ 4.12 Arduino ส่งค่าให้ Raspberry Pi ผ่านnRF24L01 | 38 |
| รูปที่ 4.13 Raspberry Pi รับค่าจาก Arduino ผ่านnRF24L01 | 38 |
| รูปที่ 4.14 ซอฟแวร์ Supervisord | 39 |
| รูปที่ 4.15 การทำงานของซอฟแวร์ Supervisord | 39 |
| รูปที่ 4.16 โปรแกรม shell command | 40 |
| รูปที่ 4.17 ฟังก์ชั่นตรวจสอบสถานะความชื้นของ Arduino | 40 |
| รูปที่ 4.18 ตรวจสอบสถานะเซ็นเซอร์ที่รับมาตามเงื่อนไข | 41 |
| รูปที่ 4.19 ฐานข้อมูลของระบบการให้น้ำพืช | 41 |
| รูปที่ 4.20 หน้าแรกของเวปแอปพลิเคชันในโหมดต่างๆ | 42 |
| รูปที่ 4.21 การทำงานของเมนู Auto Mode ในการเปิด-ปิดงรีเลย์ | 42 |
| รูปที่ 4.22 การทำงานของเมนู Manual Mode ในการเปิด-ปิดงรีเลย์ | 43 |
| รูปที่ 4.23 หน้าของเวปแอปพลิเคชันในเมนูเซ็นเซอร์ | 43 |
| รูปที่ 4.24 หน้าของเวปแอปพลิเคชันในเมนู Zones | 44 |
| รูปที่ 4.25 ฟังก์ชั่นควบคุมการให้น้ำที่ใช้ความชื้นที่ Field Capacity เป็นเกณฑ์ | 44 |
| รูปที่ 4.26 ความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์กับความชื้นในดิน | 45 |
| รูปที่ 4.27 ความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์กับความชื้นในดิน | 45 |
| รูปที่ 4.28 ฟังก์ชั่นสำหรับแปลงค่าจากแรงดันไฟฟ้าเป็นความชื้นในดิน | 46 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

# บทนำ

## 1.1 ที่มาและความสำคัญ

ประเทศไทยเป็นประเทศเกษตรกรรม ทำให้มีการปลูกพืชอย่างแพร่หลาย ซึ่งการเจริญเติบโตของพืชมีน้ำเป็นปัจจัยที่สำคัญ จึงมีการคิดค้นระบบการให้น้ำพืชขึ้นมาโดยระยะแรกได้นำระบบจับเวลาเข้ามาใช้ในการให้น้ำ ซึ่งสามารถให้น้ำโดยอัตโนมัติได้ตามเวลาที่ตั้งค่าไว้ แต่ยังมีปัญหาในการให้น้ำที่ยังไม่ตรงตามความต้องการของพืช ซึ่งต่อมาได้มีการนำไมโครคอนโทรลเลอร์กับเซ็นเซอร์เข้ามาใช้ในระบบทำให้สามารถจ่ายน้ำได้ตรงตามความต้องการของพืช แต่ยังมีข้อจำกัดในเรื่องการควบคุม และการแสดงผลข้อมูล ที่ยังต้องทำจากตู้ควบคุมหรือคอมพิวเตอร์

ทั้งนี้ทางผู้จัดทำก็ได้เล็งเห็นการประยุกต์ใช้แอปพลิเคชันบนสมาร์ทโฟน เพื่อควบคุมระบบการให้น้ำ โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวประมวลผล ในการควบคุมและสั่งการให้น้ำ ด้วยการพิจารณาจากความชื้นที่มีอยู่ในดินและเมื่อค่าความชื้นในดินลดน้อยลงถึงเกณฑ์ที่ตั้งไว้ เซ็นเซอร์ก็จะส่งข้อมูลให้ไมโครคอนโทรลเลอร์เพื่อสั่งจ่ายน้ำให้แก่พืช และเมื่อทำการส่งจ่ายน้ำได้ตามปริมาณที่ตั้งค่าไว้แล้ว เซ็นเซอร์ก็จะส่งข้อมูลให้ไมโครคอนโทรลเลอร์สั่งให้หยุดการจ่ายน้ำ ซึ่งค่าความชื้นของดิน ความชื้นของอากาศ และอุณหภูมิ จะแสดงผ่านทางแอปพลิเคชันบนสมาร์ทโฟน

## 1.2 วัตถุประสงค์

1) เพื่อศึกษาระบบควบคุมการให้น้ำพืชอัตโนมัติแบบไร้สาย

2) เพื่อศึกษาปัจจัยที่มีผลต่อการให้น้ำแก่พืช อาทิเช่น ดิน สภาพแวดล้อม และองค์ประกอบอื่นๆ ที่เกี่ยวข้อง

3) เพื่อออกแบบ และสร้างอุปกรณ์ควบคุมการให้น้ำพืชอัตโนมัติแบบไร้สาย เพื่อจ่ายน้ำในปริมาณ และเวลาที่เหมาะสมแก่พืชโดยแสดงค่าผ่านสมาร์ทโฟน

## 1.3 ขอบเขตการศึกษา

ออกแบบแอปพลิเคชันบนสมาร์ทโฟนและสร้างอุปกรณ์ควบคุมระบบการให้น้ำพืชอัตโนมัติบนแปลงทดลอง โดยใช้ไมโครคอนโทรเลอร์ควบคุมและรับสัญญาณการเปลี่ยนแปลงความชื้น อุณหภูมิ และรายงานผลปัจจัยต่างๆ ของพืชบนแอปพลิเคชันในสมาร์ทโฟน

## 1.4 ผลที่คาดว่าจะได้รับ

อุปกรณ์ควบคุมระบบการให้น้ำพืชแบบอัตโนมัติ สามารถจ่ายน้ำให้แก่พืชในปริมาณที่พืชนั้นต้องการอย่างเหมาะสม และรายงานผลปัจจัยต่างๆ ของพืชแบบเรียลไทม์ ผ่านแอปพลิเคชันบนสมาร์ทโฟนได้ ซึ่งจะทำให้ประหยัดปริมาณน้ำ เวลา และ แรงงาน ในการให้น้ำแก่พืช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# หลักการและทฤษฎี

## 2.1 ความชื้นในดิน[1]

ความชื้นในดินเป็นสัดส่วนระหว่างมวลของน้ำในดินกับมวลของดินแห้ง การวัดความชื้นในดินช่วยบอกสภาพของดินที่อยู่ในบริเวณระบบนิเวศนั้น และยังใช้อธิบายความสามารถของดินในการให้ธาตุอาหารและน้ำสู่พืชซึ่งมีผลต่อการเจริญเติบโตและการอยู่รอดของพืช ความชื้นของดินประกอบด้วย 2 สถานะ คือ สถานะที่เป็นของเหลว เราเรียกว่า น้ำในดิน และสถานะที่เป็นก๊าซ เราเรียกว่า ไอน้ำในดิน ในประเทศที่มีอากาศหนาวจัด ความชื้นของดินอาจจะอยู่ในรูปของน้ำแข็ง ส่วนประเทศในเขตร้อน ส่วนใหญ่น้ำในดินจะอยู่ในรูปของของเหลว ดังนั้นความชื้นของดิน กับน้ำในดิน จึงมีความหมายเดียวกัน คือ ส่วนที่อยู่ในสถานะที่เป็นของเหลว ถ้าในส่วนของช่องว่างในดินมีน้ำอยู่เต็มไม่มีก๊าซอยู่เลยเรียกว่า ดินที่อิ่มตัวด้วยน้ำ (saturated soil) แต่ถ้าในช่องว่างของดินมีทั้งน้ำและก๊าซอยู่ด้วยเรียกว่า ดินที่ไม่อิ่มตัว (unsaturated soil) ดังนั้น ดินที่ใช้ในการทำการเกษตรส่วนใหญ่ คือดินที่ไม่อิ่มตัว ความชื้นในดินมีความสำคัญเป็นอย่างยิ่งสำหรับสิ่งมีชีวิตในดิน ได้แก่ สัตว์ พืช หรือจุลินทรีย์ เนื่องจากน้ำเป็นองค์ประกอบที่สำคัญของพืชและสัตว์ เพื่อใช้ในขบวนการเมทาบอลิซึม (metabolism) ต่าง ๆ เช่น ขบวนการสังเคราะห์แสงของพืชและจุลินทรีย์ในดินบางชนิด พืชสามารถที่จะนำเอาธาตุอาหารไปใช้ได้ ธาตุอาหารเหล่านั้นจะต้องอยู่ในรูปของสารละลาย น้ำเป็นตัวทำละลายที่ดีและมีปริมาณมาก หาได้ง่าย และสะดวก น้ำเป็นตัวกลางที่ดีในการเคลื่อนย้ายไอออนจากบริเวณหนึ่งไปยังอีกบริเวณหนึ่ง อีกทั้งยังลำเลียงธาตุอาหารที่อยู่ในรูปของไอออนจากดินเข้าสู่ภายในลำต้นของพืช และเข้าไปในจุลินทรีย์ นอกจากนี้น้ำยังมีความร้อนจำเพาะ และความร้อนแฝงที่สูง ทำให้เปลี่ยนอุณหภูมิได้ยาก ทำให้น้ำในดินมีอุณหภูมิไม่สูงหรือต่ำจนเกินไป ทำให้ดินมีสภาวะที่เหมาะสมต่อการเจริญเติบโตของพืช

### 2.1.1 ระดับความชื้นที่สำคัญของดิน

1. ระดับความชื้นที่ดินอิ่มตัวด้วยน้ำ (water saturated) คือปริมาณน้ำในดินที่เกิดขึ้นเมื่อช่องว่างระหว่างเม็ดดินทั้งหมดถูกแทนที่ด้วยน้ำ ถ้าดินมีความสามารถในการระบายน้ำได้ดี ปริมาณน้ำที่อยู่ในช่องว่างขนาดใหญ่จะเคลื่อนที่ลงสู่ด้านล่างเนื่องจากแรงโน้มถ่วงของโลกได้ง่าย

2. ความชื้นชลประทานหรือความจุความชื้นในสนาม (Field Capacity) หมายถึง ความชื้นของดินที่เหลืออยู่ในดินหลังจากที่น้ำอิสระได้ถูกระบายออกไปจากช่องว่างหมดแล้ว หรือเป็นปริมาณน้ำที่ดินสามารถดูดซับไว้ได้มากที่สุดโดยไม่ไหลลงสู่ด้านล่างตามแรงโน้มถ่วงของโลก น้ำในช่องว่างขนาดเล็กจะมีน้ำขังอยู่เต็ม ส่วนช่องว่างขนาดใหญ่จะมีอากาศอยู่เต็ม
การวัดค่าความชื้นที่ Field Capacity โดยตรงค่อนข้างจะยุ่งยากและไม่สะดวก ในทางปฏิบัติมักจะถือว่า ในดินที่มีการระบายน้ำได้ดี ปริมาณความชื้นหลังจากที่มีฝนตกหนักหรือหยุดให้น้ำแล้ว 2-3 วัน เป็นความชื้นที่ Field Capacity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ความชื้นที่จุดเหี่ยวเฉาถาวร (Permanent wilting point) คือความชื้นในดินเมื่อพืชไม่สามารถนำมาใช้ให้เพียงพอกับการคายน้ำและพืชเริ่มมีอาการเหี่ยวเฉาอย่างถาวร

อาการเหี่ยวเฉาของพืชนั้นขึ้นอยู่กับหลายๆปัจจัย เช่น สภาพอากาศที่ร้อนจัด ความชื้นของอากาศต่ำ ลมแรง พืชมีใบกว้าง ที่กล่าวมานี้จะทำให้พืชมีการสูญเสียน้ำโดยการคายน้ำออกทางใบมาก และเมื่ออัตราที่พืชดูดน้ำจากดินน้อยกว่าที่คายน้ำออกมาทางใบพืชก็จะเหี่ยวเฉา ถึงแม้ว่าดินจะมีความชื้นอยู่มากก็ตาม

4. ระดับความชื้นในดินที่เป็นประโยชน์ต่อพืช (Available moisture content) คือ ผลต่างระหว่างความชื้นของดินในขณะหนึ่งกับความชื้นที่จุดเหี่ยวเฉาเป็นสิ่งที่แสดงถึงปริมาณความชื้นในดินที่เป็นประโยชน์ต่อพืชของดินในขณะนั้น

**2.1.2 การตรวจวัดความชื้นในดิน[2]**

การให้น้ำแก่พืชจะต้องเริ่มทำเมื่อความชื้นของดินลดลงเหลือน้อยกว่าครึ่งหนึ่งของความจุความ ชื้นที่เป็นประโยชน์ต่อพืช แต่ต้องไม่ต่ำกว่าความชื้นที่จุดวิกฤตหรือต่ำกว่า 25ของความจุความชื้นที่เป็นประโยชน์ต่อพืช โดยเราจะต้องทราบค่าความชื้นที่ Field Capacity และความชื้นที่จุดเหี่ยวเฉาถาวร วิธีการหาความชื้นในดินทำได้ 3 วิธีคือ

- การวัดความชื้นในดินโดยตรงจากการชั่งน้ำหนักและอบแห้ง
- การวัดความชื้นในดินโดยทางอ้อมโดยใช้เครื่องมือทางวิทยาศาสตร์
- การวัดความชื้นในดินโดยดูลักษณะและความรู้สึกสัมผัส

1. การวัดความชื้นในดินโดยตรงจากการชั่งน้ำหนักและอบแห้ง

การตรวจวัดความชื้นของดินโดยการชั่งน้ำหนัก ทำได้โดยการเก็บตัวอย่างดินที่มีระดับความลึกต่างๆในเขตรากพืชและที่จุดต่างๆในแปลงเพาะปลูก บรรจุดินตัวอย่างในกระป๋องเก็บตัวอย่างซึ่งมีฝาปิดมิดชิด นำมาชั่งน้ำหนักแล้วอบให้แห้งในเตาอบซึ่งมีอุณหภูมิ 105-110 องศาเซลเซียส เป็นเวลา 24 ชั่วโมงหรือจนกว่าดินนั้นจะแห้งทั่วถึงกัน ตัวอย่างดินที่เก็บควรจะหนักไม่น้อยกว่า 100 กรัม น้ำหนักที่หายไปหลังจากที่อบให้แห้งแล้วก็คือ น้ำหนักของน้ำที่อยู่ในดิน การหาจำนวนความชื้นในดินอาจทำได้ 2 แบบ คือ คำนวณเป็นเปอร์เซ็นต์โดยน้ำหนักของดินอบแห้ง และเปอร์เซ็นต์โดยปริมาตร

1.1 เปอร์เซ็นต์ความชื้นโดยน้ำหนัก เหมาะสำหรับการหาจำนวนความชื้นเมื่อไม่ทราบปริมาตรของดินตัวอย่าง การคำนวณหาเปอร์เซ็นต์ความชื้นโดยน้ำหนักทำได้โดยใช้สมการ

$$P_w = ( W_w / W_s ) \times 100 \qquad (2.1)$$

เมื่อ $P_w$ = เปอร์เซ็นต์ความชื้นโดยเทียบกับน้ำหนักของดินแห้ง

$W_w$ = น้ำหนักของน้ำในดิน

$W_s$ = น้ำหนักของดินที่อบให้แห้งด้วยเตาอบ

1.2 เปอร์เซ็นต์ความชื้นโดยปริมาตร การหาจำนวนเปอร์เซ็นต์ความชื้นโดยปริมาตรทำได้ยากกว่าเปอร์เซ็นต์ความชื้นโดยน้ำหนัก เพราะจำเป็นต้องทราบปริมาตรของตัวอย่างดิน การหาจำนวนเปอร์เซ็นต์ความชื้นโดยปริมาตรทำได้โดยใช้สมการ

$$Pv = ( Vw / V ) \times 100 \qquad (2.2)$$

เมื่อ $P_v$ = เปอร์เซ็นต์ความชื้นโดยเทียบกับปริมาตรของก้อนดินทั้งก้อน

$V_w$ = ปริมาตรของน้ำในดิน

V = ปริมาตรของก้อนดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การวัดความชื้นในดินโดยทางอ้อมโดยใช้เครื่องมือทางวิทยาศาสตร์

เครื่องมือทางวิทยาศาสตร์ที่ใช้หาค่าความชื้นในดินจะทำให้ทราบค่าความชื้นในดินได้โดยทันที ที่วัด มีอยู่หลายชนิด เช่น เครื่องเทนซิโอมิเตอร์(tensiometer) เครื่องวัดความชื้นด้วยไฟฟ้า(Electrical resistance instruments) เครื่องวัดความชื้นแบบวัดการแผ่กระจายของนิวตรอน (Neutron moisture meter) เป็นต้น

(a) (b) (c)

รูปที่ 2.1 เครื่องมือวัดความชื้นดินทางวิทยาศาสตร์ (a) Tensiometer; (b) Electrical resistance instruments; (c) Neutron moisture meter [1]

3. การวัดความชื้นในดินโดยดูลักษณะและความรู้สึกสัมผัส

วิธีนี้เป็นวิธีการตรวจวัดความชื้นในดินอย่างง่าย รวดเร็วและสามารถทำได้ด้วยตนเอง ไม่สิ้นเปลืองค่าใช้จ่ายในการตรวจสอบ ตรวจสอบโดยการขุดเจาะดินด้วยสว่านเจาะดินหรือใช้พลั่วขุดดินที่ระดับความลึกต่างๆในเขตรากพืชและบริเวณแปลงเพาะปลูกจุดอื่นๆที่ต้องการทราบ ตรวจดูโดยการใช้ความรู้สึกจากการสัมผัสด้วยมือ ทำให้ทราบค่าความชื้นของดินโดยประมาณว่ามีความชื้นในดินเหลือสำหรับพืชเท่าใด ถ้ามีความชำนาญก็จะสามารถบอกได้ทันทีว่าดินในขณะนั้นควรจะเริ่มให้น้ำได้หรือยัง อย่างไรก็ตาม วิธีการนี้เป็นเพียงการประมาณค่าความชื้นของดินเท่านั้น แต่ก็สามารถกำหนดการให้น้ำแก่พืชได้พอสมควร

## 2.2 การให้น้ำแบบฉีดฝอย (Sprinkler Irrigation)[3]

การให้น้ำแบบฉีดฝอยจะทำโดยฉีดน้ำจากหัวฉีดขึ้นไปบนอากาศแล้วให้ตกลงมาบนพื้นที่เพาะปลูก โดยมีการแผ่กระจายของหยดน้ำหรือละอองน้ำที่สม่ำเสมอ และอัตราที่น้ำตกลงบนพื้นที่จะน้อยกว่าอัตราการซึมของน้ำเข้าสู่ดิน เนื่องจากการให้น้ำโดยวิธีนี้มีลักษณะคล้ายกับฝนตก บางครั้งจึงเรียกการให้น้ำแบบนี้ว่าการให้น้ำแบบฝนโปรยหรือฝนเทียม

การเลือกใช้การให้น้ำแบบฉีดฝอย

1. ดินมีอัตราการซึมของน้ำผ่านผิวดินสูงมาก กล่าวคือสูงกว่า 75 มิลลิลิตรต่อชั่วโมง ซึ่งจะทำให้การให้น้ำแบบอื่นมีประสิทธิภาพต่ำ
2. ความลึกของชั้นดินที่เป็นประโยชน์ต่อพืชตื้นมาก ไม่เหมาะกับการปรับพื้นที่
3. พื้นที่มีความลาดชันและดินถูกกัดพาได้ง่าย
4. อัตราการส่งน้ำจากแหล่งน้ำน้อยเกินไปที่จะส่งน้ำทางผิวดินได้อย่างมีประสิทธิภาพ
5. พื้นที่ขรุขระเป็นคลื่นซึ่งถ้าจะทำการปรับพื้นที่เพื่อการให้น้ำทางผิวดินนั้นมีค่าใช้จ่ายสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ผู้ให้น้ำไม่มีความชำนาญทางด้านการให้น้ำทางผิวดิน
7. การให้น้ำแบบฉีดฝอยสามารถออกแบบและติดตั้งได้รวดเร็ว

### 2.2.1 ระบบให้น้ำแบบฉีดฝอย (Sprinkler Irrigation System)

ระบบการให้น้ำแบบฉีดฝอยแบ่งออกได้เป็น 3 แบบด้วยกัน คือ

1. แบบติดอยู่กับที่ (Permanent Systems) เป็นแบบที่อุปกรณ์ทุกอย่างติดอยู่กับที่ไม่สามรถเคลื่อนย้ายได้ ระบบนี้มักจะใช้ในเรือนเพาะชำ หรือใช้กับพืชที่ต้องการน้ำบ่อยๆ มีค่าลงทุนสูงกว่าแบบอื่นๆ แต่จะประหยัดแรงงานในการให้น้ำได้มาก

2. แบบเคลื่อนย้ายได้เพียงบางส่วน (Semi-portable Systems) อุปกรณ์บางอย่างจะติดอยู่กับที่บางอย่างจะสามารถเคลื่อนย้ายได้ โดยส่วนมาก เครื่องสูบน้ำ ท่อประธาน ท่อรองประธานจะติดอยู่กับที่ ส่วนท่อแยกซึ่งเป็นท่อที่ติดตั้งหัวฉีดน้ำจะสามารถถอดออกเป็นท่อนๆแล้วไปติดตั้งใหม่ที่อื่นได้ หรือเคลื่อนที่ไปด้านข้างด้วยตัวของมันเองโดยอาศัยแรงดันของน้ำหรือมอเตอร์ หรือหมุนเป็นวงกลมรอบปลายข้างหนึ่งของท่อแยก

3. แบบเคลื่อนย้ายได้ทั้งหมด (Portable Systems) อุปกรณ์ทั้งหมดของระบบนี้สามารถเคลื่อนย้ายได้ทั้งหมด

### 2.2.2 อุปกรณ์การให้น้ำแบบฉีดฝอย

ระบบการให้น้ำแบบฉีดฝอยประกอบด้วยอุปกรณ์ที่สำคัญ 4 อย่างด้วยกันคือ

1. เครื่องสูบน้ำ (Pumping Unit) ทำหน้าที่สูบน้ำจากแหล่งน้ำ และเพิ่มความดันให้กับหัวจ่ายน้ำ เครื่องสูบน้ำอาจจะขับเคลื่อนด้วยเครื่องยนต์หรือไฟฟ้าก็ได้

2. ท่อประธาน (Mainline Pipe Unit) ทำหน้าที่ส่งน้ำจากเครื่องสูบน้ำไปสู่ท่อแยก ท่อประธานอาจเป็นท่ออ่อน ท่อโลหะที่สามารถถอดออกเป็นท่อนๆได้ หรือเป็นท่อที่ติดอยู่กับที่ก็ได้

3. ท่อแยก (Lateral Pipe Unit) ทำหน้าที่ส่งน้ำจากท่อประธานให้กับหัวจ่ายน้ำ ท่อแยกนี้มี 3 แบบเช่นเดียวกับท่อประธาน แต่มีขนาดเล็กกว่า มีอุปกรณ์สำหรับติดตั้งท่อตั้งเพื่อให้หัวฉีดน้ำอยู่สูงกว่าระดับของยอดพืช

4. หัวจ่ายน้ำ (Sprinkler Unit) ทำหน้าที่ฉีดน้ำ ซึ่งมี 2 แบบด้วยกันคือ แบบจ่ายน้ำโดยการหมุนหัวฉีดเป็นวงกลมในแนวราบ (Rotary Sprinkler) และแบบเป็นท่อซึ่งเจาะรูเล็กๆให้น้ำฉีดออกมาตลอดความยาวของท่อนั้น (Perforated Pipe) ซึ่งแบบนี้ไม่ค่อยได้รับความนิยมนัก

### 2.2.3 การออกแบบระบบการให้น้ำ[4]

การออกแบบระบบการให้น้ำแบบฉีดฝอยจะทฤษฎีต่างๆดั้งต่อไปนี้

1. เปอร์เซ็นต์พื้นที่เปียกน้ำเทียบกับพื้นที่ทั้งหมด เป็นการบอกให้ทราบว่ามีพื้นที่ที่เปียกน้ำหรือได้รับน้ำโดยหัวปล่อยน้ำ เทียบเป็นกี่เปอร์เซ็นต์ของพื้นที่ทั้งหมด

$$P_w = (A_w / A_p) \times 100 \qquad (2.3)$$

เมื่อ $P_w$ = เปอร์เซ็นต์พื้นที่เปียกน้ำของหัวปล่อยน้ำเทียบกับพื้นที่ทั้งหมด

$A_w$ = พื้นที่เปียกน้ำที่ได้รับจากหัวปล่อยน้ำทั้งหมดต่อต้น (ตร.เมตร)

$A_p$ = พื้นที่อาณาเขตต้นพืช(ระยะระหว่างต้น x ระยะระหว่างแถว) (ตร.เมตร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบ ถ้ากำหนดให้มีค่า Pw สูงๆ น้ำก็จะเก็บอยู่ในเขตรากได้มาก แนะนำว่าค่า Pw ที่ใช้ในการออกแบบนั้นควรมีพื้นที่เปียกน้ำอย่างน้อยที่สุดหนึ่งในสาม และอย่างมากก็ครึ่งหนึ่งของพื้นที่ทั้งหมดในอาณาเขตต้นพืชก็เพียงพอแล้ว (33% ≤ Pw ≤ 50%) โดยเฉพาะสำหรับพืชที่ปลูกในระยะห่างๆกัน ควรใช้ค่า Pw ไม่เกิน 50 % แต่อย่างไรก็ตาม เพื่อความปลอดภัยในการออกแบบ ขอแนะนำว่าให้ใช้ค่า Pw ¬อย่างน้อย 60% เผื่อว่าหัวปล่อยน้ำบางหัวเกิดการไหลไม่สะดวก ซึ่งเราอาจไม่ได้สังเกต อาจทำให้พืชดูดน้ำไปใช้ได้ไม่เพียงพอ

2. จำนวนหัวปล่อยน้ำที่ใช้ได้น้อยที่สุดต่อต้นพืช เป็นการพิจารณาว่าจะใช้จำนวนหัวปล่อยน้ำกี่หัวนั้น จะต้องทราบค่าความสามารถในการกระจายน้ำว่าแต่ละหัวจะสามารถครอบคลุมพื้นที่เปียกน้ำได้มากที่สุดเท่าไร วิธีที่ดีที่สุดและถูกต้องแม่นยำที่สุดคือต้องทำการทดลองจริงๆในพื้นที่ที่จะออกแบบ

$$e = (A_p \times P_w)/(A_w \times 100) \qquad (2.4)$$

เมื่อ $e$ = จำนวนหัวปล่อยน้ำที่น้อยที่สุดต่อต้น

$A_p$ = พื้นที่อาณาเขตต้นพืชถ้าพิจารณาต้นเดียวให้ใช้พื้นที่พุ่มไม้ (ตร.เมตร)

$P_w$ = เปอร์เซ็นต์พื้นที่เปียกน้ำที่กำหนด

$A_w$ = พื้นที่เปียกน้ำต่อหัวปล่อยน้ำ (ตร.เมตร)

3. ความสามารถในการอุ้มน้ำของดิน ในการให้น้ำเราจะพยายามให้น้ำไม่มากกว่าที่ความสามารถของดินในระยะรากพืชจะเก็บเอาไว้ได้ ถ้าให้มากกว่านั้น น้ำที่ให้จะกลายเป็นน้ำที่สูญเสีย ดินแต่ละชนิดมีความสามารถในการเก็บน้ำที่แตกต่างกัน ขึ้นอยู่กับลักษณะของเนื้อดินและลักษณะการจัดเรียงตัวของเม็ดดิน

เพื่อประโยชน์ในการคำนวณหาจำนวนน้ำที่มากที่สุดที่ให้ได้ในแต่ละครั้ง และจะใช้ได้กี่วันนั่น จำเป็นต้องทราบว่าดินในแปลงมีความสามารถอุ้มน้ำได้มากเท่าไร ซึ่งตามปกติ ความชื้นในดินที่พืชนำไปใช้ประโยชน์ได้ จะนิยมบอกเป็นหน่วยความลึกของน้ำที่เทียบกับความลึกของดิน(ระยะความลึกของราก)โดยหาได้จากการทดลอง หรืออาจจะใช้ค่าโดยประมาณจากตาราง

ตารางที่ 2.1 ตารางแสดงความสามารถในการอุ้มน้ำที่เป็นประโยชน์ต่อพืชของดินประเภทต่างๆ[4]

| | ความสามารถในการเก็บน้ำที่พืชนำไปใช้ประโยชน์ได้ ความลึกของน้ำเป็น มม./ความลึกของดิน 1 ซม. | | |
|---|---|---|---|
| ดินทราย | 0.50 – 1.00 | เฉลี่ย | 0.7 |
| ดินร่วนปนทราย | 1.00 – 1.50 | เฉลี่ย | 1.2 |
| ดินร่วน | 1.20 – 1.90 | เฉลี่ย | 1.5 |
| ดินร่วนปนดินเหนียว | 1.50 – 2.10 | เฉลี่ย | 1.8 |
| ดินเหนียว | 1.30 – 2.50 | เฉลี่ย | 1.9 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

4. ปริมาณน้ำชลประทานสูงสุดสุทธิที่ให้ได้แต่ละครั้ง (Ix) ปริมาณน้ำชลประทานสูงสุดในการให้น้ำแต่ละครั้ง เพื่อเก็บไว้ในดินให้พืชใช้ได้นานๆนั้น พิจารณาได้โดยการเพิ่มความชื้นจากความชื้นก่อนให้น้ำซึ่งไม่ควรต่ำกว่าจุดวิกฤต จนถึงความชื้นที่จุดสูงสุดคือความชื้นที่ Field capacity หรือ

เรียกว่า ความชื้นที่ยอมให้ลดลงได้ (Allowable Soil Moisture Deficiency, SMD) โดยคิดเป็นความลึกของน้ำที่ต้องให้สุทธิ

$$I_x = (SMD/100) \times W_a \times Z \times (P_w/100) \quad (2.5)$$

เมื่อ $I_x$ = ความลึกสูงสุดสุทธิต่อการให้น้ำ 1 ครั้ง (มม.)
$W_a$ = ความสามารถในการอุ้มดินของน้ำ (มม./ซม.)
$Z$ = ความลึกของดินในเขตราก (ซม.)
$P_w$ = เปอร์เซ็นต์พื้นที่เปียกน้ำ
SMD = ความชื้นที่ยอมให้ลดลงได้ คือ
สำหรับพืชรากตื้น (10-50 ซม.) SMD = 20%
สำหรับพืชรากลึกมากกว่า 50 ซม. SMD = 30%

5. การพิจารณาเลือกอัตราการจ่ายน้ำของหัวปล่อยน้ำ (qa) อัตราการจ่ายน้ำของหัวปล่อยน้ำนั้น จะกำหนดเป็นลิตรต่อชั่วโมง ซึ่งอัตราการจ่ายน้ำของหัวปล่อยน้ำชนิดต่างๆจะมีแสดงให้ดูในคู่มือของบริษัทผู้ผลิต หรือผู้จำหน่าย ซึ่งอัตราการจ่ายน้ำจะสัมพันธ์กับความดันของน้ำที่หัวจ่ายน้ำได้รับ ซึ่งจะต้องพิจารณาเลือกหัวจ่ายน้ำให้เหมาะสม อัตราการจ่ายน้ำของหัวจ่ายน้ำที่นิยมใช้กันมากคือ 2,4 และ 8 ลิตร/ชม.

6. การคำนวณออกแบบท่อแขนงโดยทั่วไป ท่อแขนงนิยมวางท่อไว้บนผิวดิน เพาะสะดวกและประหยัดค่าแรง ฉะนั้นท่อที่ใช้ส่วนมากจะทำด้วยพลาสติกที่ยืดหยุ่น เช่น ท่อพีอี ถ้าวางไว้บนผิวดิน สามารถใช้งานได้ไม่น้อยกว่า 3 ปี แต่ถ้าฝังดินอาจจะใช้งานไดถึง 50 ปี แต่ถ้าใช้ท่อพีวีซี ควรจะออกแบบเป็นแบบฝังดินและติดหัวปล่อยน้ำโผล่ยื่นสูงขึ้นเหนือดิน อายุใช้งานอาจนานถึง 5 – 10 ปี แต่ถ้าวางบนผิวดินท่ามกลางแสงแดดจัด ประมาณ 2-3ปีก็ชำรุดเสียหายแล้ว

ท่อแขนงโดยส่วนมากใช้กับความดันของน้ำไม่มากนัก ไม่เกิน 40 เมตร (ประมาณ 60 ปอนด์ต่อตร.นิ้ว) ขนาดของท่อที่ใช้กันส่วนมากจะไม่เกิน 22 มม. และความยาวไม่เกิน 300 เมตร

จำนวนหัวปล่อยน้ำและอัตราการไหลของน้ำผ่านท่อแขนง โดยการหาจำนวนหัวปล่อยน้ำทั้งหมดต่อความยาวท่อแขนงนั้น หาได้จากสมการ

$$Ne = Lee/St \quad (2.6)$$

เมื่อ Le = ความยาวของท่อแขนง (เมตร)
e = จำนวนหัวปล่อยน้ำ (หัว)
St = ระยะห่างระหว่างต้นพืชบนแถว (เมตร)

สำหรับอัตราการไหลของน้ำในท่อแขนง ขึ้นอยู่กับจำนวนของหัวปล่อยน้ำที่ติดตั้งตลอดความยาวของท่อ หาได้จากสมการ

$$Ql = (1/3600) \times Ne \times qa \quad (2.7)$$

เมื่อ Ql = อัตราการไหลรวมในเส้นท่อแขนง (ลิตร/วินาที)
Ne = จำนวนหัวปล่อยน้ำบนท่อแขนง (หัว)
qa = อัตราการไหลลี่ยของหัวปล่อย (ลิตร/ชม.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ขนาดของท่อแขนง เพื่อเป็นการประหยัดในการเลือกใช้ขนาดท่อที่เหมาะสม มีข้อแนะนำว่า ความเร็วของน้ำในท่อแขนงไม่ควรมากกว่า 2 เมตรต่อวินาที จะไม่ทำให้เกิดความเสียหายจากน้ำกระแทกอย่างฉับพลันตอนปิดเปิดน้ำเร็วๆ สามารถหาขนาดของท่อได้โดยอาศัยสมการ

$$D = 2.52 (Q - l)0.5 \qquad (2.8)$$

เมื่อ Ql = อัตราการไหลรวมในเส้นท่อแขนง (ลิตร/วินาที)

D = เส้นผ่าศูนย์กลางภายในของท่อที่เล็กที่สุดที่ใช้ได้ (ซม.)

## 2.3 ไมโครคอนโทรลเลอร์

### 2.3.1 หลักการทั่วไปของไมโครคอนโทรลเลอร์[5]

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ไอซี (IC: Integrated Circuit) ที่สามารถโปรแกรมการทำงานได้ซับซ้อน สามารถรับข้อมูลในรูปสัญญาณดิจิตอลเข้าไปทำการประมวลผลแล้วส่งผลลัพธ์ข้อมูลดิจิตอลออกมาเพื่อนำไปใช้งานตามที่ต้องการได้

ไมโครคอนโทรลเลอร์ภายในชิพจะมีหน่วยความจำ, Port อยู่ในชิพเพียงตัวเดียวซึ่งอาจจะเรียกได้ว่าเป็นคอมพิวเตอร์ชิพเดี่ยว ไมโครคอนโทรลเลอร์เป็นไมโครโพรเซสเซอร์ชนิดหนึ่ง เช่นเดียว กับหน่วยประมวลผลกลาง (CPU: Central Processing Unit) ที่ใช้ในคอมพิวเตอร์ แต่ได้รับการพัฒนาแยกออกมาภายหลังเพื่อนำไปใช้ในวงจรทางด้านงานควบคุม คือ แทนที่ในการใช้งานจะต้องต่อวงจรภายนอกต่าง ๆ เพิ่มเติมเช่นเดียวกับไมโครโปรเซสเซอร์ ก็จะทำการรวมวงจรที่จำเป็น เช่น หน่วยความจำ, ส่วนอินพุท/เอาท์พุท บางส่วนเข้าไปในตัว ไอซีเดียวกัน และเพิ่มวงจรบางอย่างเข้าไปด้วยเพื่อให้มีความสามารถเหมาะสมกับการใช้ในงานควบคุม เช่น วงจรตั้งเวลา, วงจรการสื่อสารอนุกรม วงจรแปลงสัญญาณ Analog เป็น Digital เป็นต้น สรุปคือ

Microcontroller = Microprocessor + Memory + I/O

ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานอย่างกว้างขวาง โดยมักจะเป็นการนำไปใช้ฝังในระบบของอุปกรณ์อื่น ๆ (Embedded Systems) เพื่อใช้ควบคุมการทำงานบางอย่างในอุปกรณ์ เพราะว่าไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้ในงานควบคุมหลายประการเช่น

-ชิพไอซีและระบบที่ได้มีขนาดเล็ก
-ระบบที่ได้มีราคาถูกกว่าการใช้ชิพไมโครโพรเซสเซอร์
-วงจรที่ได้จะมีความซับซ้อนน้อย ช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้ในการต่อวงจร
-มีคุณสมบัติเพิ่มเติมสำหรับงานควบคุมโดยเฉพาะซึ่งใช้งานได้ง่าย
-ช่วยลดระยะเวลาในการพัฒนาระบบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์[6]

โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้นสามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit) ทำหน้าที่เป็นศูนย์กลางควบคุมการทำงานของระบบคอมพิวเตอร์ทั้งหมด โดยนำข้อมูลจากอุปกรณ์รับข้อมูลมาทำงานประมวลผลข้อมูลตามคำสั่งของโปรแกรม และส่งผลลัพธ์ออกไปหน่วยแสดงผล

2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของ เครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Mempry) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์นั้น มีความเร็วในการประมวลผลสูงตามไปด้วย

ไมโครคอนโทรลเลอร์มีหลายยี่ห้อ หลายตระกูล และหลายเบอร์ด้วยกัน ซึ่งแต่ละเบอร์ก็จะมีโครงสร้างภายในและความสามารถในการทำงานที่แตกต่างกันทำให้เลือกใช้กับงานได้อย่างเหมาะสมซึ่งในกรณีนี้เราจะใช้ Arduino UNO R3 และ Raspberry Pi ในการศึกษา

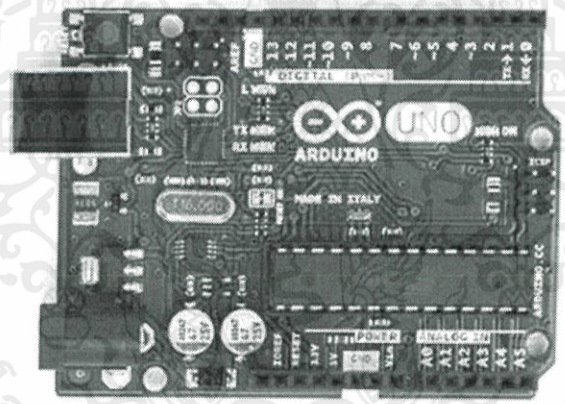
### 2.3.3 Arduino UNO R3[7]

Arduino คือ ไมโครคอนโทรลเลอร์ ชนิดหนึ่ง ซึ่งเป็นแบบที่เรียกว่า Open Hardware กล่าวคือ Arduino อุปกรณ์ที่มีแบบส่วนประกอบเป็นมาตรฐานที่เปิดเผย ซึ่งหมายความว่า เราสามารถทำเองโดยใช้แบบที่มีการเปิดเผยทั่วไปก็ได้ หรือสามารถซื้อหาก็ได้ เนื่องจากมีราคาถูก มีซอฟต์แวร์แวร์ให้ใช้งานฟรี สามารถนำไปใช้งานทั่วไปหรือแบบธุรกิจได้โดยไม่ต้องเสียค่าลิขสิทธิ์ เป็นรูปแบบที่มีข้อมูล

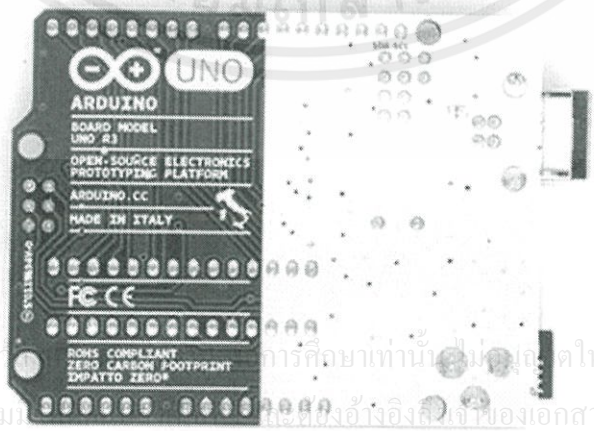
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากที่สุดบนอินเตอร์เน็ต การพัฒนาสามารถทำได้ง่าย เพราะไม่ต้องเขียนโปรแกรมในรูปแบบ Low Level หมายความว่า เราสามารถใช้คำสั่งเขียนโปรแกรมได้เสมือนโปรแกรมภาษาชั้นสูงทั่วไป ส่วนประกอบของ Arduino UNO R3 ประกอบด้วย[6]

-ไมโครคอนโทรลเลอร์ ATmega328
-แรงดัน 5V
-แรงดันขาเข้า (recommended) 7-12V
-แรงดันไฟฟ้าขา (imlits) 6-20V
-ดิจิตอล I / O Pins 14 (ซึ่ง 6 ให้ออก PWM)
-อะนาล็อกพินอินพุท 6
-ปัจจุบันซีต่อขา I / O 40 mA
-ปัจจุบัน DC สำหรับ 3.3V ขา 50 mA
-หน่วยความจำแฟลช 32 กิโลไบต์ (ATmega328) ที่ 0.5 กิโลไบต์ bootloader
-SRAM 2 กิโลไบต์ (ATmega328)
-EEPROM 1 กิโลไบต์ (ATmega328)
-ความเร็วสัญญาณนาฬิกา 16 MHz




รูปที่ 2.2 Arduino UNO R3 ด้านหน้า[7]




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

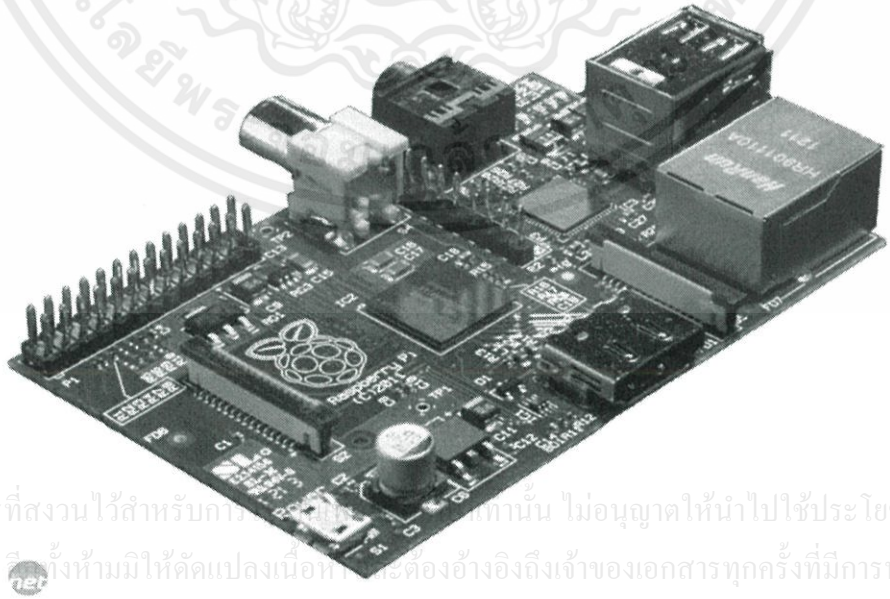
รูปที่ 2.3 Arduino UNO R3 ด้านหลัง[7]

### 2.3.4 Raspberry Pi[8]

Raspberry Pi เป็นแผงวงจรอิเล็กทรอนิกส์บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

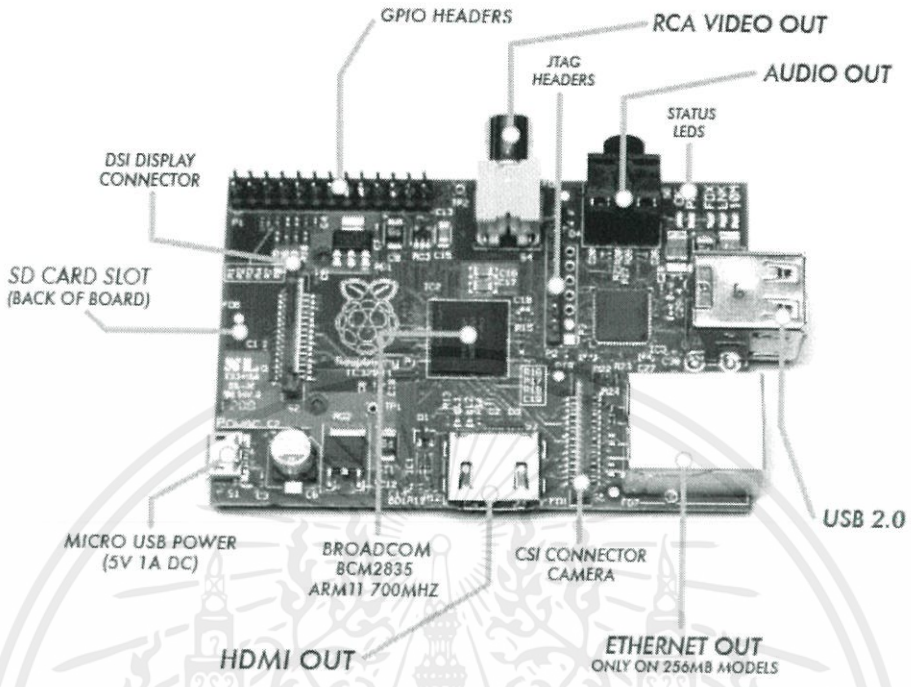
บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ คุณสมบัติของ Raspberry Pi ซึ่งประกอบไปด้วย

- Chip : Broadcom BCM2835 SoC full HD multimedia applications processor
- CPU : 700 MHz Low Power ARM1176JZ-F Applications Processor
- GPU : Dual Core VideoCore IV® Multimedia Co-Processor
- Memory : 512MB SDRAM
- Ethernet : onboard 10/100 Ethernet RJ45 jack
- USB 2.0 : Dual USB Connector
- Video Output : HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
- Audio Output : 3.5mm jack, HDMI
- Onboard Storage : SD, MMC, SDIO card slot
- Operating System : Linux
- Dimensions : 8.6cm x 5.4cm x 1.7cm
- Power requirements : 5V - 700mA



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

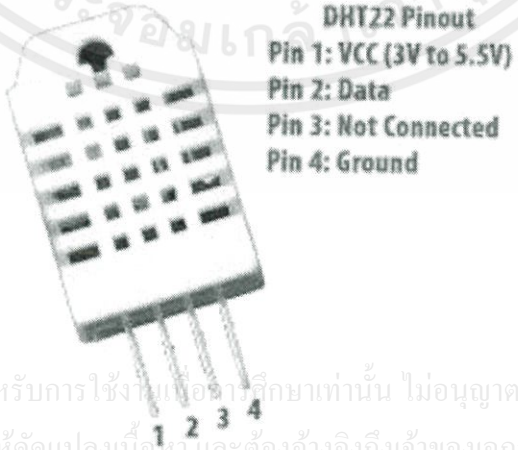
รูปที่ 2.4 Raspberry Pi[8]




รูปที่ 2.5 ส่วนประกอบของ Raspberry Pi[8]

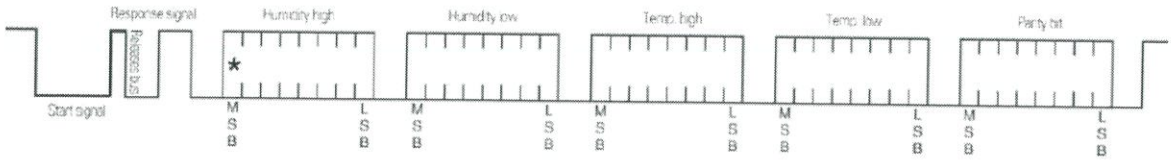
## 2.4 DHT22 Humidity and Temperature Sensor[9]

Module DHT22 หรือ AM2302 เป็นโมดูลที่ใช้ในการวัดอุณหภูมิกับความชื้นในอากาศ ใช้แรงดันไฟเลี้ยงในช่วง: 3.3V ถึง 5.5V DC ดังนั้นจึงใช้ได้กับ 3.3V และ 5V สามารถวัดอุณหภูมิได้ในช่วง -40 to 80 °C (±0.5 °C accuracy) และวัดความชื้นสัมพัทธ์ได้ในช่วง 0 - 100 RH% (2 - 5% accuracy) โดยมีอัตราการวัดสูงสุดอยู่ที่ 0.5Hz คอนเนกเตอร์แบบ 4 ขา ( 0.1" / 2.54mm spacing)

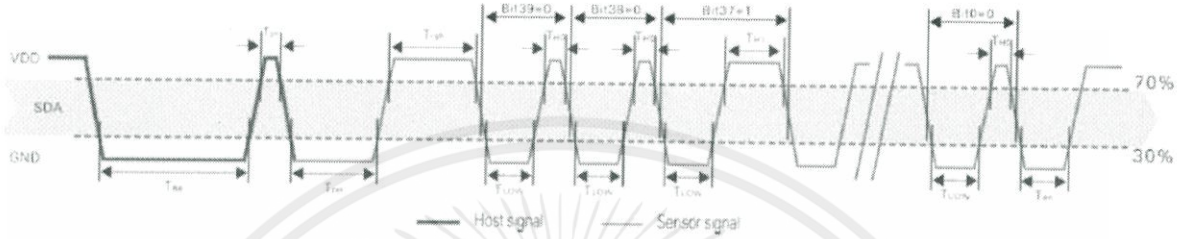



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.6 DHT22/AM2302 Humidity and Temperature Sensor[9]



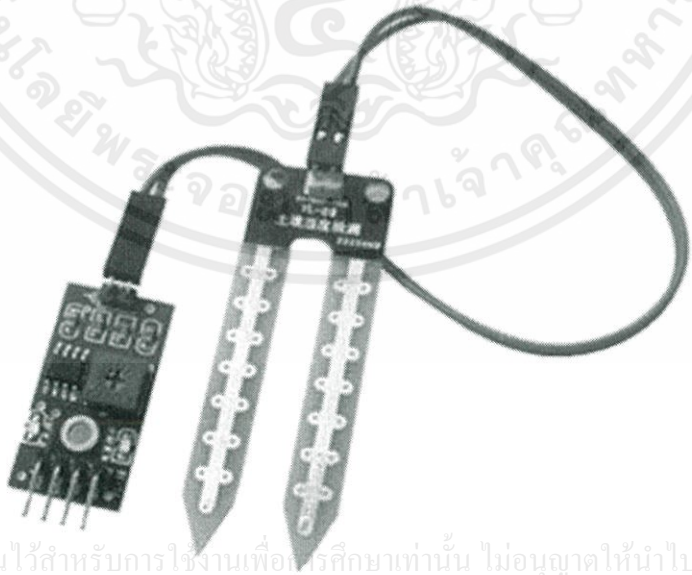

รูปที่ 2.7 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีทั้งหมด 5 ไบต์[9]




รูปที่ 2.8 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH[9]

**2.5 ทฤษฎี Sensor สำหรับตรวจวัดความชื้นในดิน**[10]

หลักการที่ใช้วัดความชื้นในดินโดย Sensor คือค่าความต้านทานในดินจะแปรผกผันกับค่าความชื้นภายในดิน ลักษณะของตัวตรวจวัดความชื้นในดินสำหรับโครงการนี้ Soil moisture sensor ใช้ไฟ 5V ในการเลี้ยงวงจร โดยเก็บค่า Analog to Digital Convertor หรือค่าความต่างศักย์ ( $V_{OUT}$ ) ที่ไมโครคอนโทรลเลอร์แปลงจาก Analog เป็น Digital เมื่อความต้านทานของดินเปลี่ยนแปลงไป (ความชื้นดินเปลี่ยนแปลงไป) จะส่งผลให้ค่าความต่างศักย์ ที่ออกมาเปลี่ยนแปลงไปด้วย ความสัมพันธ์ระหว่างความชื้นดินกับค่าความต่างศักย์ ค่าความต่างศักย์ที่ได้จะถูกส่งเข้าไปในไมโครคอนโทรลเลอร์เพื่อทำการแปลงค่าความต่างศักย์ไปเป็นค่าความชื้นในดิน

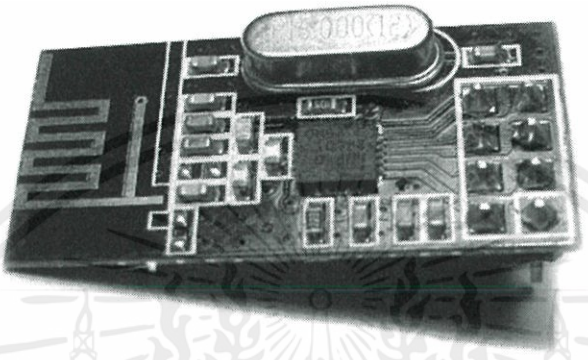


รูปที่ 2.9 เซนเซอร์ตรวจจับความชื้นดิน[10]

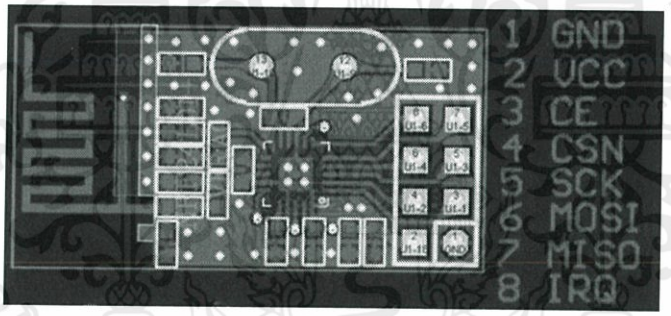
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 โมดูล nRF24L01[11]

nRF24L01 เป็นโมดูลเพื่อการสื่อสารแบบไร้สาย สื่อสารระหว่างกันด้วยคลื่นความถี่วิทยุ ที่ระดับความถี่ 2.4 GHz ออกแบบให้มีสายอากาศฝังอยู่บนตัวโมดูล โดยเอาท์พุตของไอซีนี้เป็นการติดต่อแบบ SPI ใช้พลังงานต่ำ มีระบบจัดการข้อมูลเช่นพวกการเข้าคิว, การแก้ความผิดพลาดของข้อมูล ฯลฯ ใช้ไฟเลี้ยงได้ตั้งแต่ 1.9 V ถึง 3.6 V แต่สามารถใช้ระดับแรงดันในการติดต่อได้ถึง 5V โมดูลตัวนี้ มีระยะการส่งในที่โล้งแจ้ง ได้สูงสุดถึง 200 ฟุต



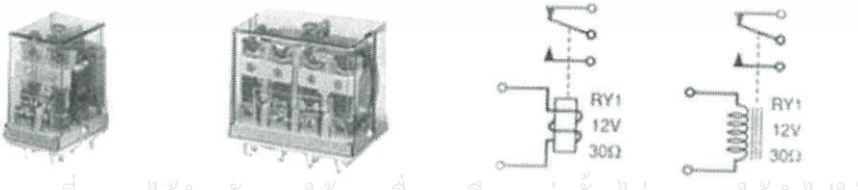
รูปที่ 2.10 โมดูล nRF24L01[11]




รูปที่ 2.11 ขาต่างๆ ของโมดูล nRF24L01[11]

## 2.7 Relay[12]

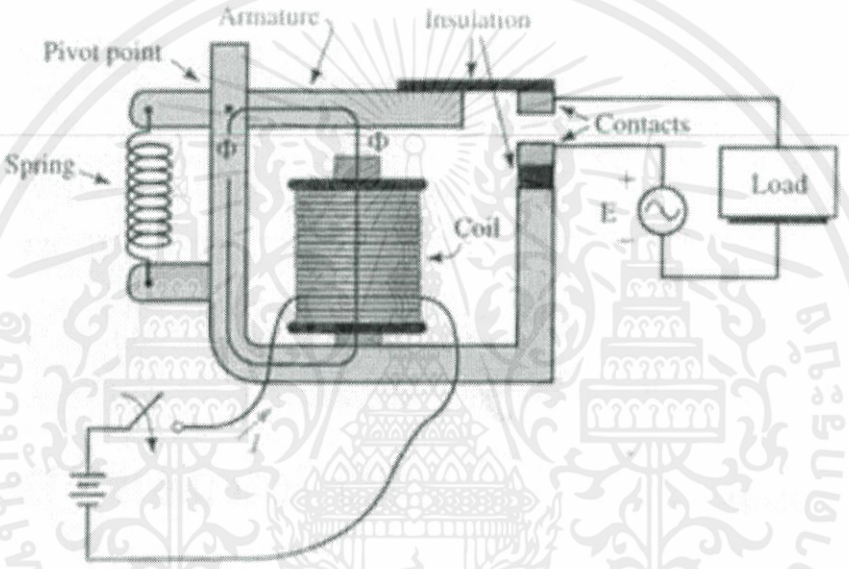
รีเลย์เป็นอุปกรณ์ที่นิยมนำมาทำเป็นสวิตช์ทางด้านอิเล็กทรอนิกส์ โดยจะต้องป้อนกระแสไฟฟ้าให้ไหลผ่านขดลวดจำนวนหนึ่ง เพื่อนำไปควบคุมวงจรกำลังงานสูง ๆ ที่ต่ออยู่กับหน้าสัมผัสหรือคอนแทกต์ของรีเลย์ รูปที่ 2.12 แสดงรูปร่างและสัญลักษณ์ของรีเลย์




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.12 รูปร่างและลักษณะของรีเลย์[12]

หลักการทำงานเบื้องต้นของรีเลย์แสดงดังรูปที่ 2.41 การทำงานเริ่มจากปิดสวิตช์ เพื่อป้อนกระแสให้กับขดลวด (Coil) โดยทั่วไปจะเป็นขดลวดพันรอบแกนเหล็ก ทำให้เกิดสนามแม่เหล็กไปดูดเหล็กอ่อนที่เรียกว่าอาร์เมเจอร์ (Armature) ให้ต่ำลงมา ที่ปลายของอาร์เมเจอร์ด้านหนึ่งมักยึดติดกับสปริง (Spring) และปลายอีกด้านหนึ่งยึดติดกับหน้าสัมผัส (Contacts) การเคลื่อนที่อาร์เมเจอร์ จึงเป็นการควบคุมการเคลื่อนที่ของหน้าสัมผัส ให้แยกจากหรือแตะกับหน้าสัมผัสอีกอันหนึ่งซึ่งยึดติดอยู่กับที่ เมื่อเปิดสวิตช์อาร์เมเจอร์ ก็จะกลับสู่ตำแหน่งเดิม เราสามารถนำหลักการนี้ไปควบคุมโหลด (Load) หรือวงจรอิเล็กทรอนิกส์ต่าง ๆ ได้ตามต้องการ ซึ่งในที่นี้เราจะนำเอา Relay มาใช้ในการเปิดปิดปั๊มเพื่อใช้ในการจ่ายน้ำให้แก่พืช




รูปที่ 2.13 หลักการทำงานเบื้องต้นของรีเลย์[12]

## 2.8 Real Time Clock (RTC)[13]

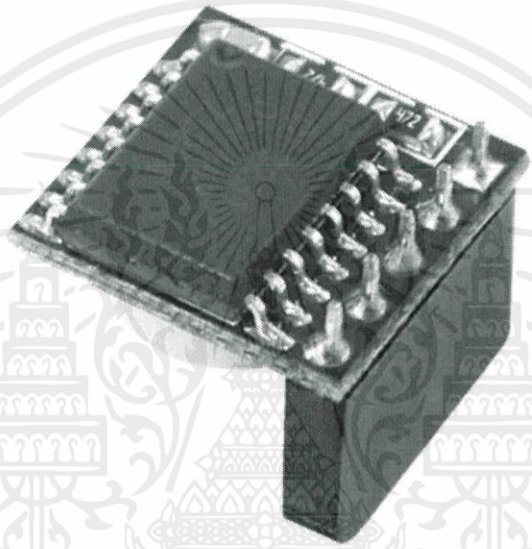
DS3231 เป็นไอซีประเภท RTC (Real-Time Clock) ของบริษัท Dallas Semiconductor / Maxim ทำหน้าที่เป็นระบบฐานเวลา (ทำหน้าที่เป็นเสมือนนาฬิกาของระบบ) เก็บข้อมูลอย่างเช่น วินาที นาที ชั่วโมง (แบบ 12 หรือ 24) วันเดือนและปีในปัจจุบัน เชื่อมต่อสื่อสารแบบบัส I2C ได้

ข้อมูลเชิงเทคนิคที่สำคัญของไอซี DS3231

- ใช้แรงดันไฟเลี้ยง (VCC) ในช่วง +2.5V .. +5.5V (+3.3V typ.)
- ใช้แบตเตอรี่สำรองได้ แรงดันในช่วง (VBAT) +2.5V .. +5.5V (+3V typ.)
- ใช้พลังงานต่ำ (Low-Power Consumption) ดังนั้นเมื่อปิดแรงดันไฟเลี้ยง VCC สามารถทำงานต่อเนื่องได้โดยใช้แรงดันไฟเลี้ยง VBAT ได้โดยอัตโนมัติ
- ใช้ตัวถังแบบ SO (Small Outline) จำนวน 16 ขา
- เชื่อมต่อแบบบัส I2C (สัญญาณ SDA และ SCL) และใช้ความเร็วได้ถึง 400kHz
- ภายในมีวงจรสร้างสัญญาณ clock (crystal oscillator) ความถี่ 32kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีความแม่นยำ (Accuracy) ±2ppm ในช่วงอุณหภูมิ 0°C + 40°C และ ±3.5ppm สำหรับ -40°C..+85°C
- สามารถตั้งค่าการแจ้งเตือนหรือ Alarm เลือกได้จาก 2 ชุด และสร้างสัญญาณอินเทอร์รัพท์ได้ (Interrupt)
- สามารถเลือกสร้างสัญญาณเอาต์พุตได้ (Programmable Square-Wave Output) ที่ขา #INT/SQW
- สามารถวัดค่าอุณหภูมิได้ ให้ข้อมูลดิจิทัลแบบ 10 บิต (2's complement) ความละเอียด 0.25°C แต่มีความแม่นยำ ±3°C



รูปที่ 2.14 แสดงหลักการทำงานเบื้องต้นของรีเลย์[13]

**2.9 โปรแกรม PuTTy[14]**

PuTTy เป็นโปรแกรมที่มีหน้าที่เอาไว้ ใช้ในการ Remote หรือติดต่อ เชื่อมต่อ ไปยังเครื่องคอมพิวเตอร์อีกเครื่องหนึ่ง โดยวิธีการ Telnet หรือ Secure Shell (SSH) จากเครื่องลูก (Client) เข้าไปจัดการ พิมพ์คำสั่ง หรือส่งคำสั่ง ในเครื่องแม่ (Server) ด้วยระบบ Command-line Interface (ใช้พิมพ์คำสั่ง) โดยให้ความรู้สึกเหมือนนั่งอยู่บนหน้าเครื่องเซริฟ์เวอร์ (เครื่องแม่) จริงๆ โดยไม่ต้องเดินทางไปนั่งหน้าเครื่องนั้นจริงๆ อยู่ที่ไหนในโลกก็สามารถเข้าไปได้ โดยผ่านกระบวนการการเข้าใช้ทรัพยากรระยะไกลทำได้ด้วยกัน 3 ทาง ได้แก่

1. SSH (Secure Shell, RFC 4250-4256)
   - เป็นโปรโตคอลการเชื่อมต่อที่มีความปลอดภัยมากกว่า Rlogin หรือ Telnet
   - การยืนยันตัวตน (Authentication) มีความปลอดภัยมากขึ้นเนื่องจากมีการเข้ารหัสที่เรียกว่า Public key cryptiography
   - สนับสนุนการทำงานอื่นๆ เช่น Tunneling, TCP port forwarding และ X11 Connection
   - สามารถโอนย้ายแฟ้มต่างๆ โดยใช้งานร่วมกับโปรโตคอล SFTP และ SCP ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Rlogin (RFC 1282)

- เป็นโปรโตคอลเชื่อมต่อที่ไม่มีการเข้ารหัสข้อมูล โดยใช้งานผ่าน port 513
- สามารถทำการ Login โดยไม่ต้องใส่รหัสผ่านได้

3. Telnet (RFC 854)

- ไม่มีการเข้ารหัสข้อมูลเช่นเดียวกันกับ Rlogin แต่ใช้งานบน port 23
- การ Login ต้องมีการใส่ชื่อผู้ใช้และรหัสผ่าน
- Telnet daemon มีช่องโหว่ให้โจมตี

ประโยชน์/ข้อดีของ PuTTy

- เป็นโปรแกรมฟรี และโปรแกรมมีขนาดเล็ก
- รองรับการเชื่อมต่อเครื่องผู้ให้บริการได้หลายรูปแบบ
- ง่ายต่อการใช้งาน และง่ายต่อการตั้งค่าต่างๆ ของการเชื่อมต่อ
- เหมาะสมสำหรับผู้ใช้งานทุกระดับ

## 2.10 **โปรแกรม** WinSCP**[15]**

WinSCP เป็นโปรแกรมประเภทที่เอาไว้ใช้รับส่งไฟล์ หรือที่เรียกว่า โปรแกรม FTP หรือ SFTP SCP มันเป็น โปรแกรมรับส่งไฟล์ ขนาดจิ๋ว พกพาได้ที่ดีอีกตัวหนึ่ง ที่รวมเอาคุณสมบัติของ การเชื่อมต่อข้อมูล หรือใช้ในการ รับส่งไฟล์ ต่างๆ ระหว่างเครื่องลูก (Client) ไปยังเครื่องแม่ข่าย (Server) แต่ว่ามันต้องทำงานร่วมกับ โปรแกรม Telnet ชื่อดังอย่าง โปรแกรม Putty ซึ่งไม่ได้ให้มันเป็นเพียงแค่โปรแกรมส่งคำสั่ง (Command) ไปยังเครื่องแม่ข่ายอย่างเดียว แต่ยังทำให้มันสามารถเป็น โปรแกรมรับส่งไฟล์ ที่เอาไว้อัพโหลด ดาวน์โหลดไฟล์ ระหว่างเครื่องลูกกับ เครื่องแม่ ได้อีกด้วย

โดย โปรแกรม FTP นี้ ได้รวมเอาความสามารถของโปรโตคอล การรับส่งไฟล์ ชื่อดัง และยอดนิยมของโลกอย่าง โปรโตคอล SFTP, FTP รวมไปถึง SCP (Secure Copy) Client ที่ได้รวมเอาคุณสมบัติการ รับส่งไฟล์ ไปใส่ไว้กับ โปรแกรม Putty ซึ่งเป็น โปรแกรม Telnet ที่รู้จักกันดีเอาไว้ด้วย บทสรุปคือ จึงสามารถทำให้ รับส่งไฟล์ รวมไปถึง ข้อมูล (Data) ไว้ได้โดย ตัวโปรแกรม FTP นี้โปรแกรมเดียว

## 2.11 ภาษาที่ใช้ในการเขียนโปรแกรมและคำสั่ง

### 2.11.1 ภาษา Python[16]

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

ในปัจจุบันภาษาที่ใช้ในการพัฒนา Web Application มีมากมายหลายภาษา อาทิเช่น ภาษา Perl, PHP, JAVA, ASP, Tcl, Python เป็นต้น สำหรับภาษา Python นับว่ายังใหม่ในวงการพัฒนาโปรแกรมบนเว็บ แต่ด้วยข้อดีหลายประการของภาษา Python ทำให้มีผู้นิยมใช้มากขึ้นเรื่อยๆ ซึ่งสามารถสรุปข้อดีของภาษา Python ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง และใช้ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ง่ายต่อการเรียนรู้ โดยภาษา Python มีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษา Python จะคล้ายกับภาษา C มาก เพราะภาษา Python สร้างขึ้นมาโดยใช้ภาษา C ทำให้ผู้ที่คุ้นเคยภาษา C อยู่แล้วใช้งานภาษา Python ได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี

2. ไม่ต้องเสียค่าใช้จ่ายใดๆ เพราะตัวแปรภาษา Python อยู่ภายใต้ลิขสิทธิ์ GNU

3. ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษา Python ถูกออกแบบใช้งานกับระบบ Unix อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษา Python ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ อาทิเช่น Linux, Windows 95/98/ME, Windows NT, Windows 2000, 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

### 2.11.2 ภาษา C[17]

ภาษา C เป็นภาษาคอมพิวเตอร์ที่ถูกค้นคิดขึ้นในปี ค.ศ. 1970 เพื่อใช้ในระบบปฏิบัติการยูนิกซ์ (UNIX) นับจากนั้นมาก็ได้รับความนิยมเพิ่มขึ้นจนถึงปัจจุบัน ภาษา C สามารถติดต่อในระดับฮาร์ดแวร์ได้ดีกว่าภาษาระดับสูงอื่น ๆ ไม่ว่าจะเป็นภาษาเบสิกฟอร์แทน ขณะเดียวกันก็มีคุณสมบัติของภาษาระดับสูงอยู่ด้วย ด้วยเหตุผลดังกล่าวจึงจัดได้ว่าภาษา C เป็นภาษาระดับกลาง (Middle –lever language) ภาษา C เป็นภาษาคอมพิวเตอร์ชนิดคอมไพล์ (compiled Language) ซึ่งมีคอมไพลเลอร์ (Compiler) ทำหน้าที่ในการคอมไพล์ (Compile) หรือแปลงคำสั่งทั้งหมดในโปรแกรมให้เป็นภาษาเครื่อง (Machine Language) เพื่อให้เครื่องคอมพิวเตอร์นำคำสั่งเหล่านั้นไปทำงานต่อไป

### 2.11.3 ภาษา HTML[18]

HTML (Hyper Text Markup Language) เป็นภาษาประเภท Markup Language ที่ใช้ในการสร้างเว็บเพจ โดยมีแม่แบบมาจากภาษา SGML (Standard Generalized Markup Language) ที่ตัดความสามารถบางส่วนออกไป เพื่อให้สามารถทำความเข้าใจและเรียนรู้ได้ง่าย ซึ่งปัจจุบันมีการพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C)

ภาษา HTML ได้ถูกพัฒนาขึ้นอย่างต่อเนื่องตั้งแต่ HTML Level 1, HTML 2.0, HTML 3.0, HTML 3.2 และ HTML 4.0 ในปัจจุบัน ทาง W3C ได้ผลักดัน รูปแบบของ HTML แบบใหม่ ที่เรียกว่า XHTML ซึ่งเป็นลักษณะของโครงสร้าง XML แบบหนึ่ง ที่มีหลักเกณฑ์ในการกำหนดโครงสร้างของโปรแกรมที่มีรูปแบบที่มาตรฐานกว่า มาทดแทนใช้ HTML รุ่น 4.01 ที่ใช้กันอยู่ในปัจจุบัน

HTML มีโครงสร้างการเขียนโดยอาศัย Tag ในการควบคุมการแสดงผลของข้อความ รูปภาพ หรือวัตถุอื่น ๆ แต่ละ Tag อาจจะมีส่วนขยาย เรียกว่า Attribute สำหรับจัดรูปแบบเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างเว็บเพจ โดยใช้ภาษา HTML สามารถทำโดยใช้โปรแกรม Text Editor ต่างๆ เช่น Notepad, EditPlus ส่วนการเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML จะใช้โปรแกรม Internet Web Browser เช่น Internet Explorer (IE), Mozilla Firefox, Safari, Opera, และ Google Chrome เป็นต้น

### 2.11.4 ภาษา JavaScript[19]

จาวาสคริปต์ (JavaScript) เป็นภาษาสคริปต์ ที่มีลักษณะการเขียนแบบโปรโตไทพ (Prototyped-based Programming) ส่วนมากใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน แต่ก็ยังมีใช้เพื่อเพิ่มเติมความสามารถในการเขียนสคริปต์โดยฝังอยู่ในโปรแกรมอื่นๆ

จาวาสคริปต์ เป็นภาษาในรูปแบบของภาษาโปรแกรมแบบโปรโตไทพ โดยมีโครงสร้างของภาษาและไวยกรณ์อยู่บนพื้นฐานของภาษาซี ปัจจุบันมีการใช้จาวาสคริปต์ที่ฝังอยู่ในเว็บเบราว์เซอร์ในหลายรูปแบบ เช่น ใช้เพื่อสร้างเนื้อหาที่เปลี่ยนแปลงเสมอภายในเว็บเพจ, ใช้เพื่อตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้กรอกก่อนนำเข้าระบบ, ใช้เพื่อเข้าถึงข้อมูลที่อยู่ภายใต้โครงสร้างแบบ Document Object Model (DOM)เป็นต้น

นอกจากนี้จาวาสคริปต์ยังถูกฝังอยู่ในแอปพลิเคชันต่างๆ นอกเหนือจากเว็บเบราว์เซอร์ได้อีกด้วย โดยรวมแล้วจาวาสคริปต์ถูกใช้เพื่อให้นักพัฒนาโปรแกรม สามารถเขียนสคริปต์เพื่อสร้างคุณสมบัติพิเศษต่างๆ เพิ่มเติมจากที่มีอยู่บนแอปพลิเคชันดั้งเดิม

### 2.11.5 ภาษา SQLite[20]

SQLite คือไลบรารีระบบจัดการฐานข้อมูลที่พัฒนาโดยภาษา C ทำงานในรูแบบของไฟล์ข้อมูลซึ่งแตกต่างจากระบบฐานข้อมูลตัวอื่นที่จะทำงานในระบบ Client-Server ที่จะฝังตัวอยู่ในระบบ โดยรูปแบบการทำงานของ SQLite เป็นแบบ Standalone ทำงานอยู่ใน Application นั้น ๆ SQLite มีโครงสร้างง่ายต่อการจัดเก็บและนำไปใช้ และไฟล์ที่จัดเก็บนั้นก็มีขนาดเล็กมาก เกือบเท่ากับการเก็บข้อมูลจริง เพราะฉะนั้น SQLite Database จึงเหมาะสมในการใช้พัฒนาร่วมกับ Embedded systems, Web browsers, และ Application softwares เป็นอย่างยิ่ง โดยเฉพาะ อันเนื่องจากข้อจำกัดทางด้าน Hardware และ Memory รวมทั้งความสามารถในการ Process ข้อมูลต่าง ๆ ใน Embedded systems หรือ Smartphone ย่อมน้อยกว่า PC Desktop เป็นธรรมดา

### 2.11.7 Apache[21]

Apache ทำหน้าที่เป็น web server ที่มีผู้ใช้กันทั่วโลก มีหน้าที่ในการจัดเก็บ Homepage ไปยัง Browser ที่มีการเรียกเข้า ยัง web server ที่เก็บ Homepage ซึ่งปัจจุบันจัดได้ว่าเป็น web server ที่น่าเชื่อถือเป็นอย่างมาก อาปาเช่เป็นซอฟต์แวร์ที่อยู่ในลักษณะของโอเพ่นซอร์ส ที่เปิดให้บุคคลทั่วไปได้สามารถเข้ามาร่วมพัฒนาให้เกิดประโยชน์มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

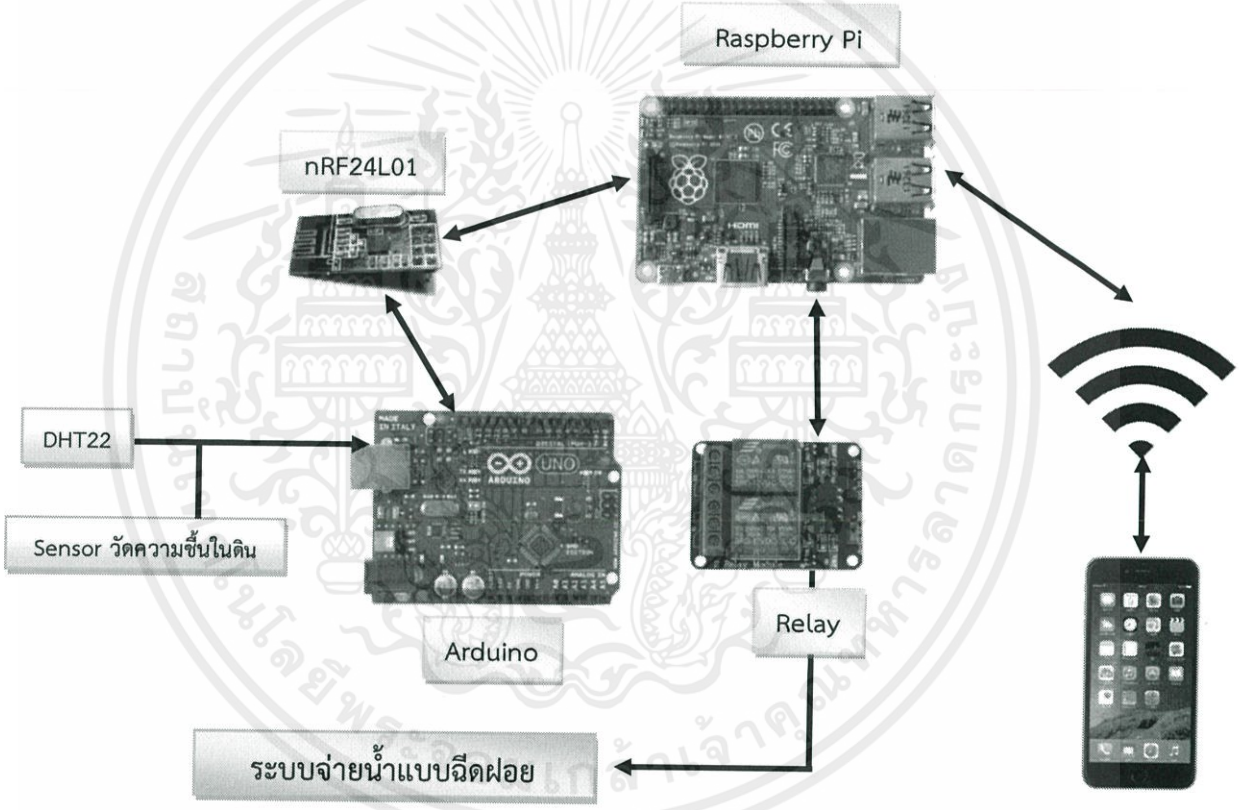
# บทที่ 3

## วิธีและขั้นตอนการดำเนินการ

การออกแบบและพัฒนาระบบควบคุมการให้น้ำพืชจะนำทฤษฎีจากบทที่ 2 มาใช้สำหรับออกแบบและพัฒนาระบบ โดยเนื้อหาในบทนี้จะกล่าวถึงภาพรวมของระบบและอธิบายรายละเอียดต่างๆในการออกแบบและพัฒนาระบบ

### 3.1 องค์ประกอบของระบบ

การออกแบบระบบนั้นมีองค์ประกอบโดยรวมแสดงได้ดังรูปที่ 3.1




รูปที่ 3.1 องค์ประกอบของระบบ

จากรูปองค์ประกอบของระบบการออกแบบและพัฒนาระบบควบคุมการให้น้ำพืชจะถูกแบ่งเป็น 2 ส่วนที่สำคัญดังนี้

#### 3.1.1 ส่วนฮาร์ดแวร์

- หน่วยประมวลผลกลาง เลือกใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino เป็นส่วนการติดต่อกับอุปกรณ์ภายนอกและเซ็นเซอร์ ส่วน Raspberry pi เป็นตัวประมวลผลทำงานเป็นเซิร์ฟเวอร์ โดยใช้โมดูล nRF24L01 ในการติดต่อกันระหว่าง Arduino กับ Raspberry pi
- เซ็นเซอร์วัดความชื้นและอุณหภูมิ (Humidity and Temperature Sensor) ได้เลือกใช้ DHT22 ซึ่งให้ค่าเป็นแบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เซ็นเซอร์สำหรับตรวจวัดความชื้นในดิน Soil moisture sensor
- วงจรรีเลย์ควบคุม เลือกใช้รีเลย์โมดูล 8 Channel ชนิด 5 ขา 2 หน้าสัมผัส แรงดันใช้งาน 5VDC และกระแสผ่านหน้าสัมผัสที่ 10 A 250 VAC

**3.1.2 ส่วนซอฟต์แวร์**

- ส่วนติดต่อกับผู้ใช้ (Mobile Application)
- ส่วนติดต่อประสานกับผู้ใช้ (GUI)
- ส่วนติดต่อกับฐานข้อมูล

## 3.2 การออกแบบฮาร์ดแวร์

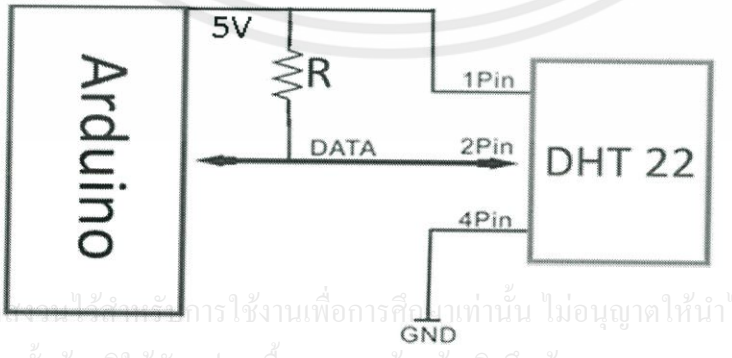
ส่วนของฮาร์ดแวร์จะประกอบไปด้วย บอร์ดไมโครคอนโทรลเลอร์ Arduino Raspberry pi เซ็นเซอร์วัดเซ็นเซอร์วัดความชื้นและอุณหภูมิ เซ็นเซอร์สำหรับตรวจวัดความชื้นในดิน โซลินอยวาล์ว ปั๊มน้ำ และวงจรรีเลย์ควบคุม ได้มีการออกแบบและพัฒนาโดยใช้ภาษา C ใน Arduino ติดต่อควบคุมและรับค่าจากเซ็นเซอร์ โดย Arduino จะรับค่าจากเซ็นเซอร์ DHT22 และเซ็นเซอร์สำหรับตรวจวัดความชื้นในดิน แล้วส่งให้ Raspberry pi เพื่อประมวลผลและแสดงค่าที่ได้ให้กับผู้ใช้แอปพลิเคชั่นบนมือถือ ผ่านทางอินเทอร์เน็ตและมีการรับค่าจากผู้ใช้กลับมายัง Raspberry pi เพื่อที่จะประมวลผลและควบคุมการการให้น้ำ โดยมีวงจรรีเลย์ทำหน้าที่เป็นสวิตช์ในการตัดต่อวงจร

ในส่วนของวงจรจะแบ่งออกเป็น 2 ส่วนใหญ่ๆคือ ส่วนตรวจวัดความชื้น (Sensor Node) และส่วนควบคุมการทำงานหลัก (Base Station)

**3.2.1 ส่วนตรวจวัดความชื้น (Sensor Node)**

ส่วนตรวจวัดความชื้นจะใช้ไมโครคอนโทรลเลอร์ Arduino ในการรับค่าจากเซ็นเซอร์ DHT22 ,เซ็นเซอร์ Soil moisture sensor และเช็คสถานะแบตเตอรี่ จากนั้นส่งเอาต์พุตไปยังไมโครคอนโทรลเลอร์ Raspberry pi เพื่อประมวลผลผ่านโมดูล nRF24L01 ซึ่งลำดับการทำงานของโปรแกรมในส่วนตรวจวัดความชื้น สามารถแบ่งการต่อวงจรของระบบควบคุมได้ดังนี้

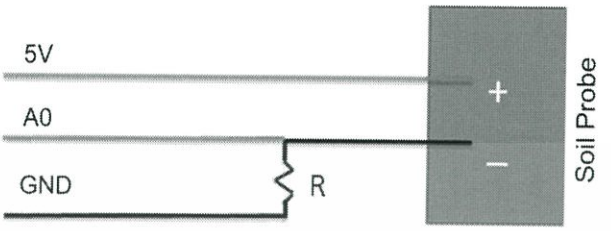
- การต่อโมดูล DHT 22 เข้ากับ Arduino จะแสดงในรูปที่ 3.2 โดยทำการต่อขาPin 1 ของDHT 22 เข้ากับขาVCC 5V ของArduino ต่อขาPin 4 ของDHT 22 เข้ากับขาGND ของArduino ต่อตัวต้านทานคร่อมขาPin1และPin2 จากนั้นต่อขาPin 2 ของDHT 22 เข้ากับขาDigital Pin 2 ของ Arduino เพื่ออ่านค่าที่วัดได้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

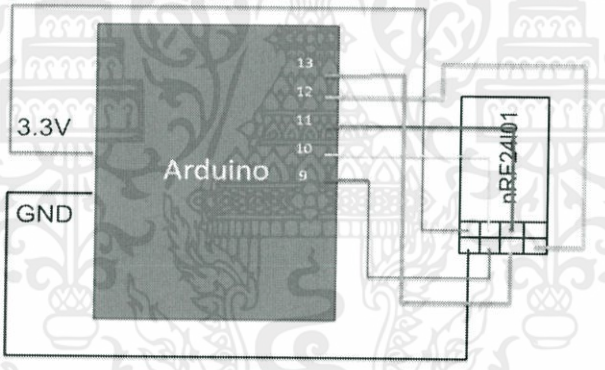
รูปที่ 3.2 การต่อโมดูล DHT 22 เข้ากับArduino

- การต่อ Soil moisture sensor เข้ากับ Arduino จะแสดงในรูปที่ 3.3 โดยทำการต่อขาบวก ของ Soil moisture sensor เข้ากับขาVCC 5V ของArduino ต่อตัวต้านทานคร่อมขาA0 และ GND จากนั้นต่อขาA0 และGND เข้ากับขาลบของ Soil moisture sensor อ่านค่าที่วัดได้จากขาA0 ของArduino



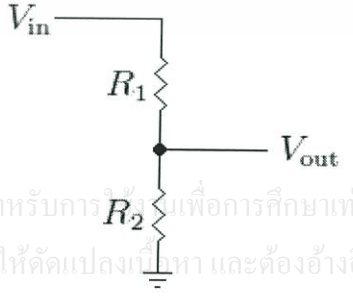

รูปที่ 3.3 การต่อ Soil moisture sensor เข้ากับArduino

- การต่อโมดูลnRF24L01 เข้ากับ Arduino จะแสดงในรูปที่ 3.4 โดยทำการต่อขาบวกของnRF24L01 เข้ากับขาVCC 3.3V ของArduino ต่อขาGND ของnRF24L01 เข้ากับขาGND ของ Arduino และต่อตัวเก็บประจุคร่อมขาVCC และ GND จากนั้นต่อขา SS กับขา 9 , ขาCSNกับขา 10, ขาMOSIกับขา 11, ขาMISOกับขา 12, ขาSCKกับขา 13 ของโมดูลnRF24L01 เข้ากับ Arduino ตามลำดับ




รูปที่ 3.4 การต่อโมดูลnRF24L01 เข้ากับArduino

- การต่อวงจรวัดแบตเตอรี่โดยใช้วงจร voltage divider แบ่งแรงดันไฟฟ้าจาก 9V เป็น 5V เนื่องจาก Arduino ใช้แรงดันไฟฟ้า 5V จึงต้องมีการแปลงค่าเพื่อวัดแรงดันไฟฟ้าที่เหลือของแบตเตอรี่โดยจะทำการต่อวงจรดังรูปที่ 3.5 โดย $R_1$ = 5kΩ และ $R_2$ = 4kΩ



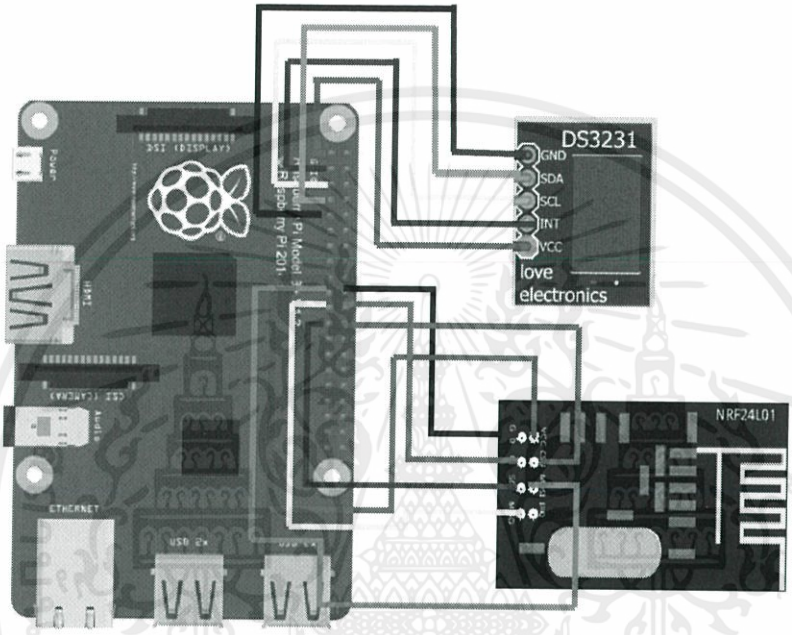

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 การต่อวงจรวัดแบตเตอรี่โดยใช้วงจร voltage divider

### 3.2.2 ส่วนควบคุมการทำงานหลัก (Base Station)

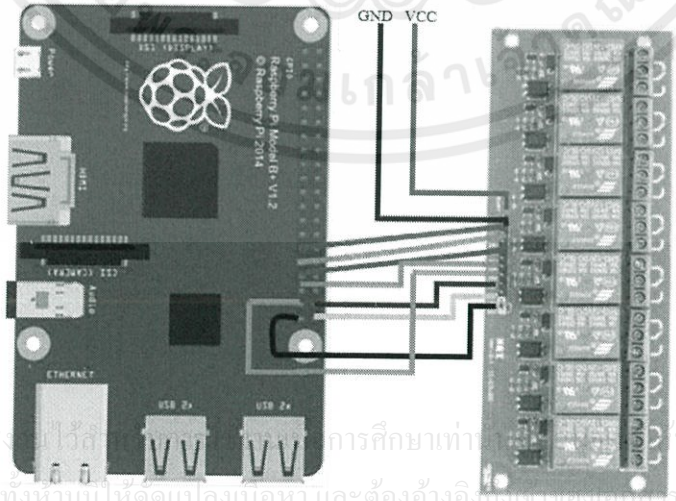
ในส่วนนี้จะใช้ Raspberry pi สั่งคำสั่งผ่าน Serial port เพื่อควบคุมปั้มน้ำ โซลินอยวาล์ว โดยมีการต่อร่วมกับวงจรรีเลย์ควบคุม ทำหน้าที่เป็นสวิตช์ในการเปิด-ปิดอุปกรณ์Hardware ในส่วนของ Raspberry Pi สามารถแบ่งการต่อวงจรของระบบควบคุมได้ดังนี้

- ในการต่อ RTC และ nRF24l01 กับ Raspberry Pi จะแสดงในรูปที่ 3.6 โดยหลักๆคือการเชื่อมต่อ Pin CE ให้ตรงกับที่ตั้งค่าไว้ในซอฟแวร์นั่นคือ Pin 25




รูปที่ 3.6 การต่อ RTC และ nRF24l01 กับ Raspberry Pi

- ในการต่อรีเลย์ กับ Raspberry Pi เพื่อควบคุมการทำงานของโซลินอยด์วาล์วจะแสดงในรูปที่ 3.7

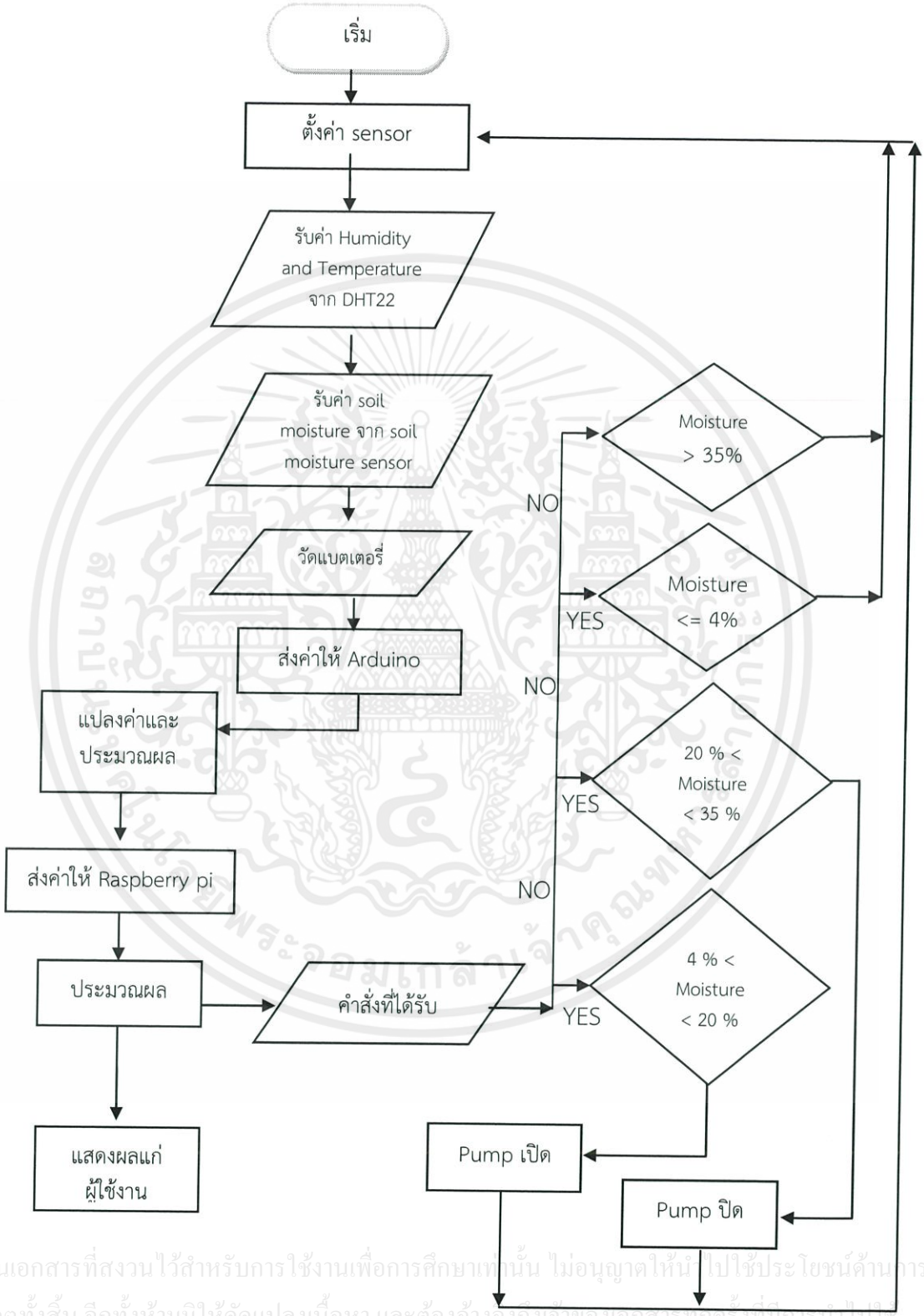



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.7 การต่อรีเลย์ กับ Raspberry Pi

### 3.3 ซอฟต์แวร์และระบบการทำงาน

ซึ่งลำดับการทำงานของโปรแกรมในส่วนฮาร์ดแวร์ทั้งหมดเป็นดังนี้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 ลำดับการทำงานของโปรแกรมส่วนฮาร์ดแวร์ทั้งหมด

จากรูปที่ 3.8 ลำดับการทำงานเริ่มต้นจากการกำหนดค่ารีจีสเตอร์ (Initial Register) ต่อมาไมโครคอนโทรลเลอร์ Arduino จะทำการรับค่าจากเซ็นเซอร์ที่ละตัวแล้วส่งค่าไปยัง Raspberry pi โดยใช้โมดูล nRF24L01 ในการติดต่อ จากนั้นจะทำการประมวลผลพร้อมส่งค่าไปยังผู้ใช้ หลังจากนั้นจะทำการตรวจสอบค่าHumidity หากน้อยกว่าค่าที่ตั้งไว้ ไมโครคอนโทรลเลอร์จะควบคุมปั๊มน้ำ (Pump On) เพื่อจ่ายน้ำให้แก่พืช แต่ถ้าไม่ใช่จะปิดปั๊มน้ำ (Pump Off) โดยสามารถแบ่งการออกแบบซอฟต์แวร์เป็น 2 ส่วนดังนี้

### 3.3.1 ซอฟแวร์ที่ใช้ในการพัฒนา

โดยซอฟแวร์ที่ใช้ในการเขียนและพัฒนาระบบทั้งในส่วนของ Arduino และ Raspberry Pi จะแสดงในตารางที่ 3.1

ตารางที่ 3.1 ซอฟแวร์ที่ใช้ในการพัฒนา

| ที่ | อุปกรณ์ | ซอฟแวร์ที่ต้องการ |
|---|---|---|
| 1. | Arduino | 1) Temperature<br>2) RF24master<br>3) LowPowermaster |
| 2. | Raspberry Pi | 1) Linux Operating System<br>2) Sqlite3<br>3) WiringPi2 and Wiringpi2-Python<br>4) Supervisord<br>5) Flask<br>6) Apache-wsgi |
| 3. | Laptop | 1) Arduino IDE<br>2) Notepad++<br>3) PuTTy<br>4) WinSCP |

จากตารางที่ 3.1 แสดงถึงซอฟแวร์ที่ต้องการในการพัฒนาระบบให้น้ำ เริ่มจาก Arduino รับค่าจากเซ็นเซอร์ DHT22 และเซ็นเซอร์ Soil moisture sensor จากนั้นส่งเอาต์พุตผ่านทางโมดูล nRF24L01 ไปยังไมโครคอนโทรลเลอร์ Raspberry pi เพื่อประมวลผลข้อมูลจากเซนเซอร์และบันทึกลงฐานข้อมูล ซึ่งต้องมีการติดตั้งระบบปฏิบัติการเพื่อให้ Computer Hardware ติดต่อกับ software ที่พัฒนาได้ สำหรับ Raspberry Pi จะใช้ระบบปฏิบัติการ Linux เป็นหลัก

ข้อมูลจะบันทึกอยู่ในรูปแบบของ SQLite ซึ่งเป็นรูปแบบไฟล์ที่เล็กและง่ายต่อการพัฒนา เหมาะกับการใช้งานในระบบที่มีการใช้งานขนาดเล็ก ซึ่งตรงกับความต้องการของระบบนี้

ในการเชื่อมต่ออุปกรณ์หรือสั่งการใช้งาน GPIO บน Raspberry Pi เพื่อให้สะดวกต่อการพัฒนาจำเป็นต้องใช้ไลบรารี่ wiringPi

เนื่องจากปกติในการใช้งาน GPIO นั้นจะต้องกระทำในขณะที่ผู้ใช้อยู่ในสถานะ Superuser ผ่านคำสั่ง sudo เท่านั้น ซึ่งจะทำให้ไม่สามารถใช้งานได้ผ่านทางอื่น เช่น ทางเว็ป ทางแอปพลิเคชั่น ไลบรารี่ตัวนี้ถูกพัฒนามาเพื่อลบข้อจำกัดนั้น พร้อมทั้งเสริมความสามารถในการใช้งาน GPIO เพิ่มขึ้นอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากระบบจะต้องพร้อมรับข้อมูลที่ส่งมาจากเซ็นเซอร์ตลอดเวลา จึงต้องมีการรันโปรแกรมพร้อมๆกันหลายตัว

Supervisord ช่วยในส่วนการจัดการสั่งรันโปรแกรมในรูปแบบ Background ซึ่งจะทำให้ไม่มีการทำงานซ้อนกันในแต่ละโปรแกรม พร้อมทั้งตรวจสถานะในการทำงานของโปรแกรมนั้นได้อีกด้วย

Flask เป็นซอฟแวร์ที่จัดการในเรื่องของการเชื่อมต่อกับส่วนแสดงผล พัฒนาโดยภาษา Python เมื่อมีการเรียกขอข้อมูลจากเซิฟเวอร์ Flask จะมีหน้าที่เป็น Server เสมือนในการดึงข้อมูลที่ถูกเรียกมาให้ส่วนแสดงผลประมวลผลต่อไป เนื่องจากมันถูกพัฒนามาในภาษา Python จึงสามารถพัฒนาร่วมกับส่วนอื่นได้โดยง่าย

ระบบนี้ใช้ Raspberry Pi เป็นตัวเซิฟเวอร์ในการส่งข้อมูลแก่ผู้ใช้ จึงต้องมีการติดตั้งซอฟแวร์ Apache และตั้งค่าให้มีการเชื่อมต่อในวง LAN เพื่อใช้ในการพัฒนา พร้อมทั้งติดตั้ง wsgi application เพื่อให้ Apache สามารถใช้งานร่วมกับ Flask ซึ่งเป็นเฟรมเวิคสำหรับการพัฒนาเวปได้

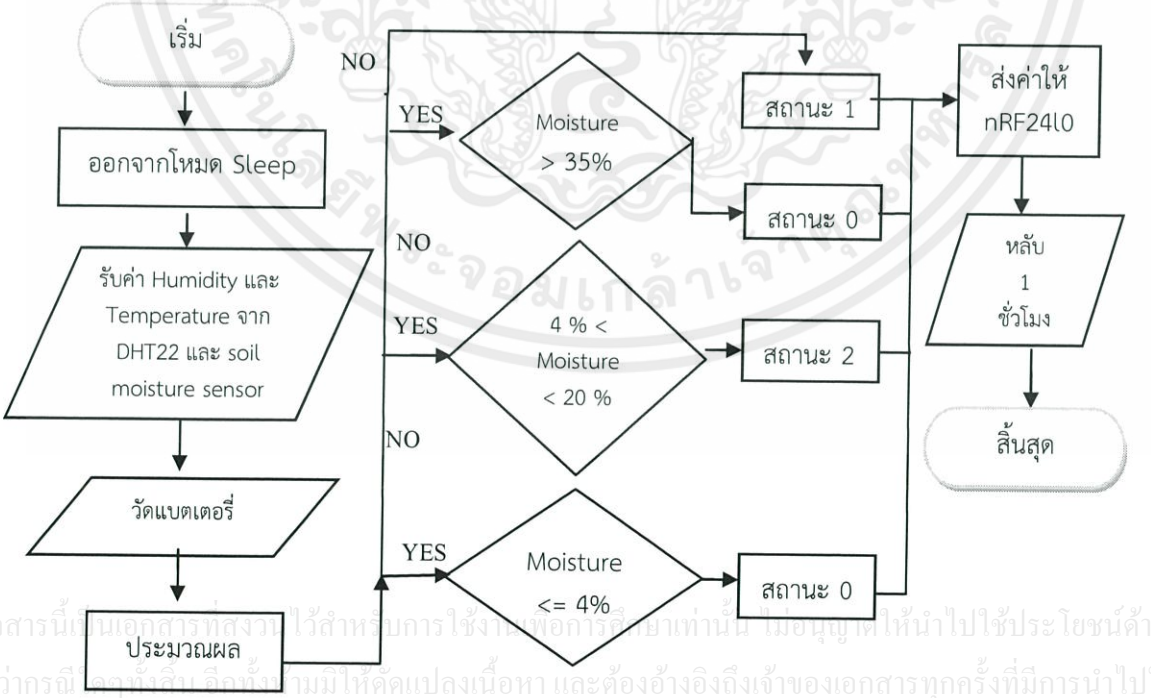
การเขียนโปรแกรม การติดตั้งโปรแกรมทุกอย่างจะกระทำบน Laptop เชื่อมต่อกับอุปกรณ์ต่างๆ เช่น Arduino, Raspberry Pi โดยผ่าน USB , LAN เขียนโปรแกรมผ่าน Arduino IDE, NotePad++

### 3.3.2 การออกแบบซอฟแวร์

ระบบการให้น้ำแบ่งเป็น 5 ส่วน ส่วนแรกคือส่วนเซ็นเซอร์วัดความชื้น ส่วนสองคือส่วนรับผลจากเซ็นเซอร์ ส่วนที่สามคือควบคุมรีเลย์ ส่วนที่สี่คือส่วนของฐานข้อมูล และส่วนสุดท้ายคือส่วนเซิฟเวอรกับส่วนแสดงผลเวปแอฟพลิเคชัน

- ซอฟแวร์ส่วนเซ็นเซอร์วัดความชื้น (Sensor Node)

ส่วนตรวจวัดความชื้นจะใช้ไมโครคอนโทรลเลอร์ Arduino ในการรับค่าจากเซ็นเซอร์ DHT22 และเซ็นเซอร์ Soil moisture sensor มาประมวณผลและส่งค่าเอาต์พุตไปยังไมโครคอนโทรลเลอร์ Raspberry pi ผ่านโมดูล nRF24L01 หลังจากส่งค่าเสร็จ Arduino โหมดSleep เพื่อประหยัดพลังงาน ซึ่งลำดับการทำงานของโปรแกรมในส่วนตรวจวัดความชื้นจะแสดงในรูปที่ 3.9

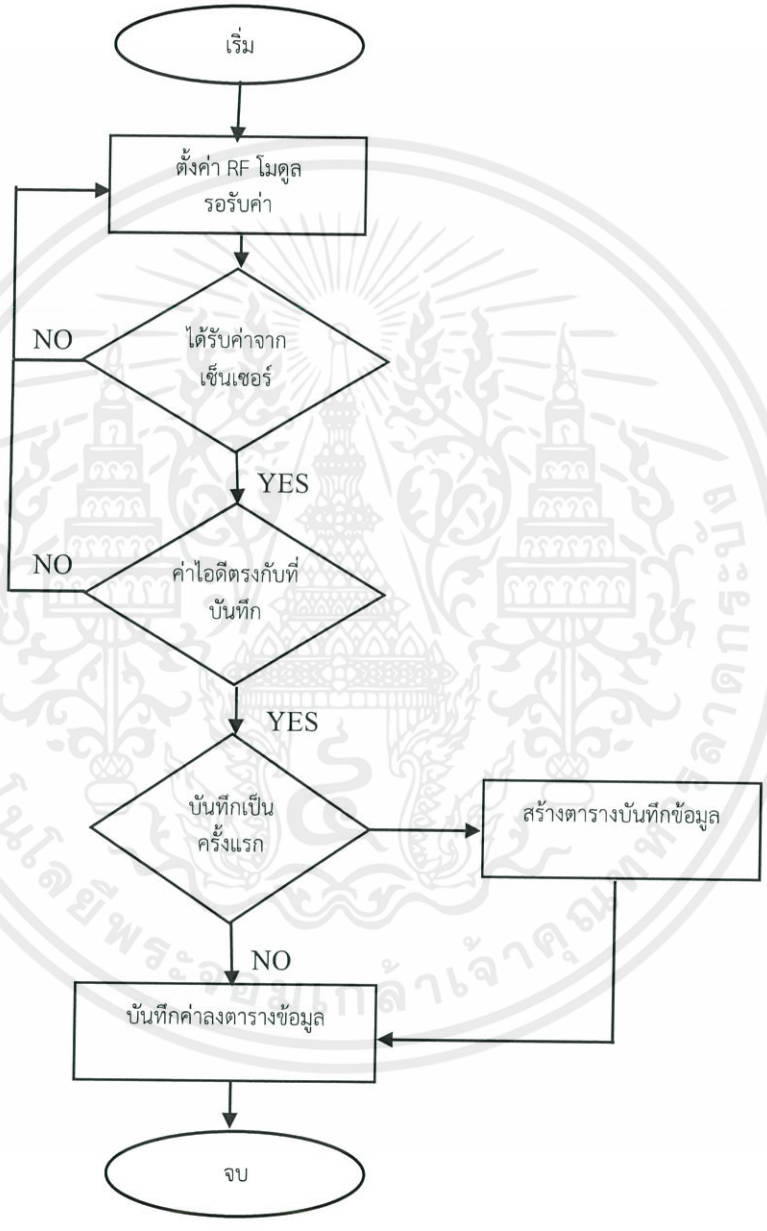



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 ลำดับการทำงานของโปรแกรมส่วนตรวจวัดความชื้น

- ซอฟแวร์ส่วนรับข้อมูลจากเซ็นเซอร์

เริ่มจากปรับค่าเริ่มต้นของ RF โมดูล ค่าที่รับมาจาก RF โมดูลจะถูกแบ่งเป็นส่วนๆ ซึ่งในแต่ละส่วนก็คือค่าที่วัดได้จากเซ็นเซอร์ต่างๆ นั่นก็คือค่า ไอดีเซ็นเซอร์, แบตเตอรี่, อุณหภูมิ, ความชื้นอากาศ, ความชื้นในดิน และสถานะของดิน ต่อมาทำการตรวจสอบว่า ไอดีเซ็นเซอร์ ตรงกับที่ผู้ใช้งานได้เปิดใช้หรือไม่หากตรงจะทำการบักทึกค่าเหล่านั้นลงบนฐานข้อมูลแต่หากเป็นการบันทึกครั้งแรกจะทำการสร้างตารางบันทึกข้อมูลก่อน



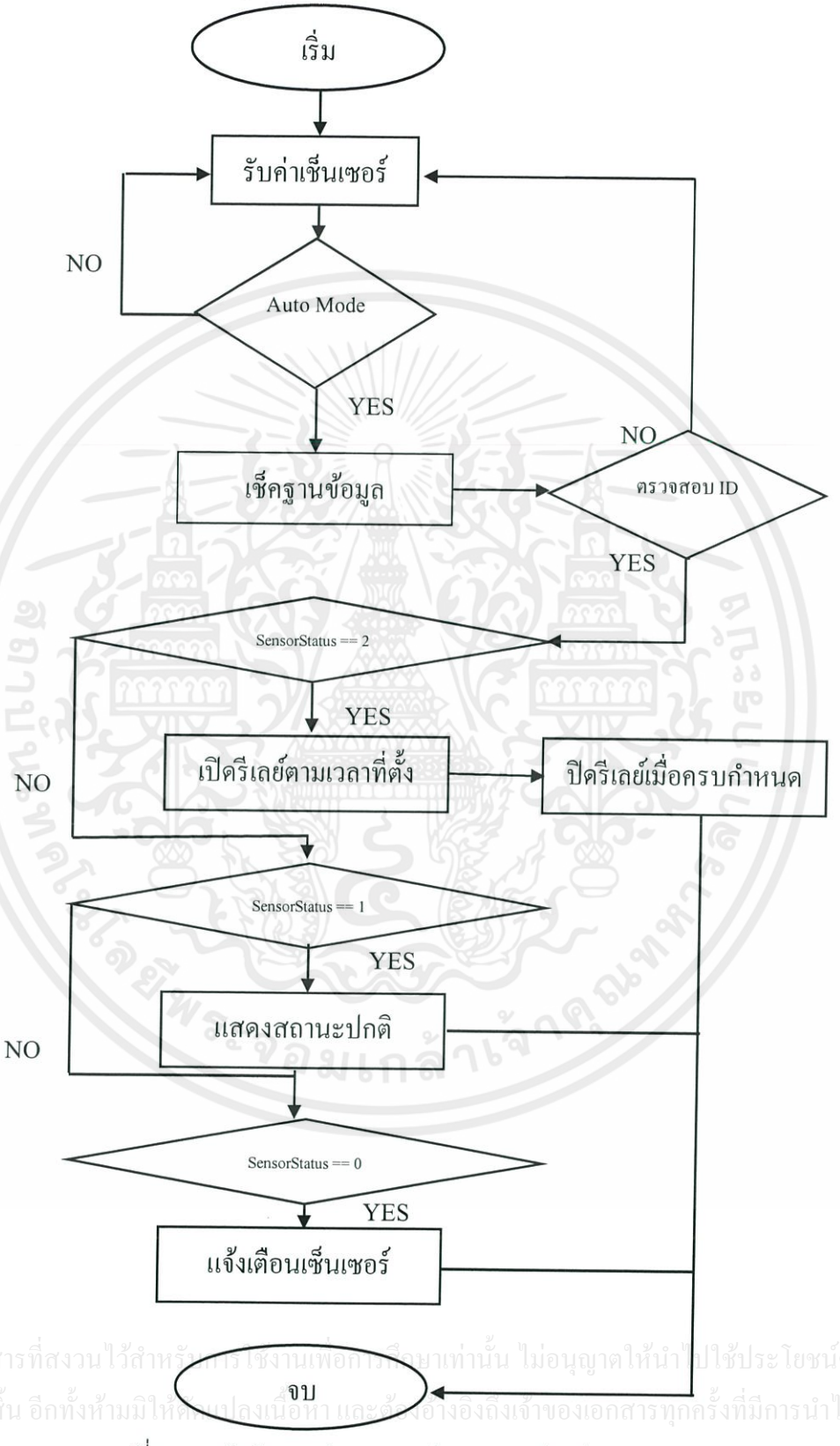

รูปที่ 3.10 ลำดับการทำงานของส่วนรับข้อมูลจากเซ็นเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ซอฟแวร์ส่วนควบคุมรีเลย์

ในการควบคุมรีเลย์ได้มีการเขียนโปรแกรมให้เปิด-ปิดได้ทีละตัวเนื่องจากต้องการรักระดับแรงดันปั๊มให้คงที่ การควบคุมจะมีทั้งแบบ เปิด-ปิดทีละโซน และเปิด-ปิดตามค่าความชื้นในดินที่

ได้รับจากเซ็นเซอร์ โดยสามารถกำหนดได้ว่า จะใช้ค่าจากเซ็นเซอร์ตัวไหนในการควบคุมโซนใดได้ เปิดเป็นเวลากี่นาที คำสั่งในการให้น้ำจะเรียงคิวตามค่าเซ็นเซอร์ที่อ่านได้




รูปที่ 3.11 ลำดับการทำงานของส่วนควบคุมรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ซอฟแวร์ส่วนฐานข้อมูล

ฐานข้อมูลในระบบนี้มีความสำคัญในการเชื่อมต่อการทำงานของซอฟแวร์อื่นๆในระบบ ในสร้างฐานข้อมูลนั้นใช้ภาษา Python ในการเขียนและใช้ library SQLite3 ในฐานข้อมูลจะมีตารางหลักอยู่ได้แก่ ตารางโซน, ตารางดัชนีเซ็นเซอร์ ,ตารางดัชนีโปรแกรมและตารางเช็คสถานะ 2 ตาราง ในตารางโซนจะใช้ในการบันทึกสถานะของโซนในระบบ ประกอบไปด้วยคอลัมน์ ไอดี, ชื่อโซน, สถานะโซน ตารางดัชนีเซ็นเซอร์จะใช้ในการบันทึกข้อมูลเซ็นเซอร์ที่มี ประกอบด้วยคอลัมน์ เซ็นเซอร์ไอดีที่มี, ชื่อเซ็นเซอร์และคำอธิบายเซ็นเซอร์ ตารางดัชนีโปรแกรมซึ่งจะใช้ในโหมดอัตโนมัติ ประกอบด้วยคอลัมน์เซ็นเซอร์ที่ใช้ข้อมูล, สถานะโปรแกรม, และลำดับโซนที่ควบคุม สำหรับตารางเช็คสถานะจะใช้ในการบันทึกโซนที่เปิดใช้ล่าสุด เวลาที่ทำงาน

ตารางที่ 3.2 ส่วนฐานของข้อมูล

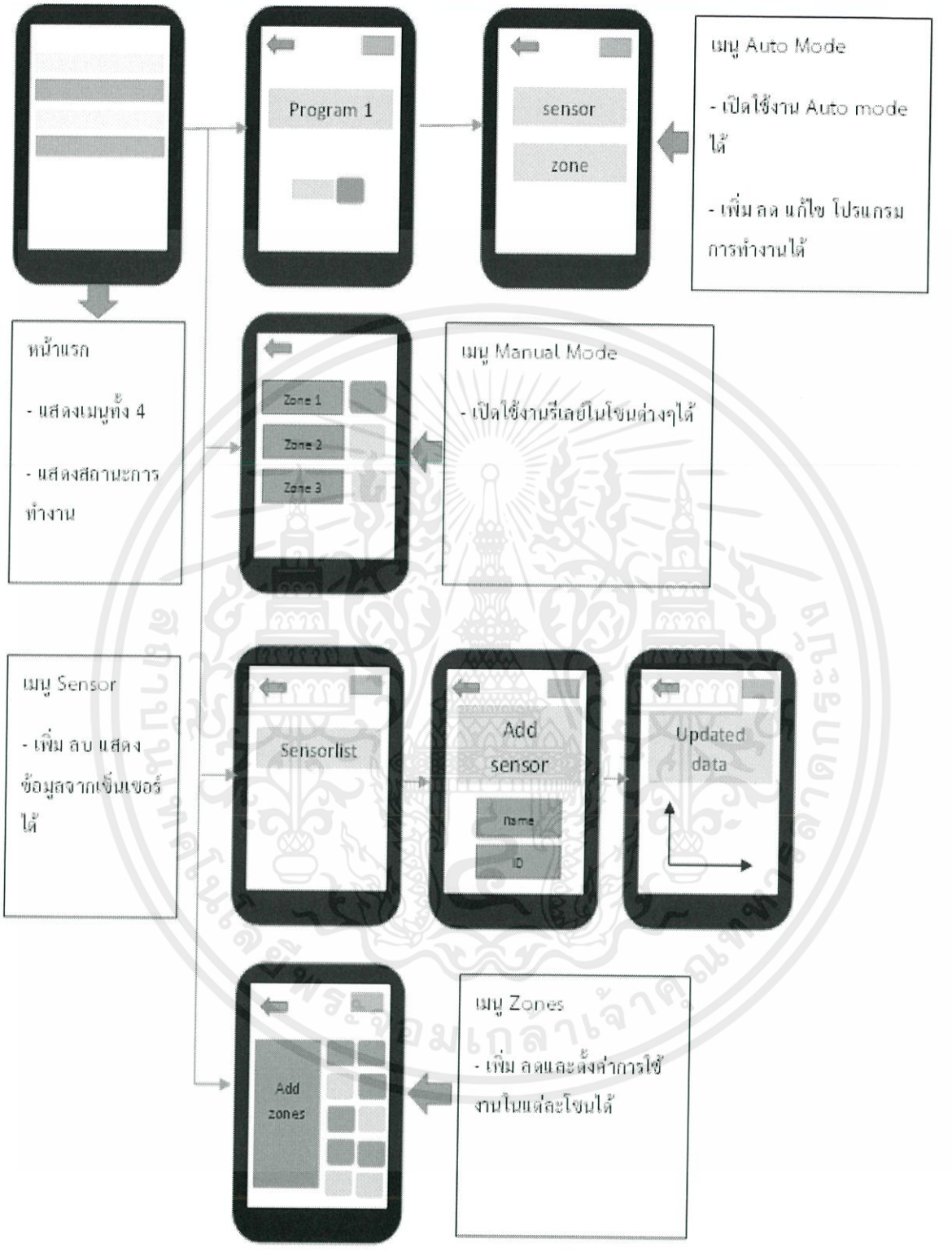
<table>
<tr><td colspan="5">ฐานข้อมูล</td></tr>
<tr><td>ตารางโซน</td><td>ดัชนีเซ็นเซอร์</td><td>ดัชนีโปรแกรม</td><td rowspan="4">เช็คสถานะ 1</td><td rowspan="4">เช็คสถานะ 2</td></tr>
<tr><td>ไอดี</td><td>เซ็นเซอร์ไอดี</td><td>เซ็นเซอร์ที่ใช้ข้อมูล</td></tr>
<tr><td>ชื่อโซน</td><td>ชื่อเซ็นเซอร์</td><td>สถานะโปรแกรม</td></tr>
<tr><td>สถานะโซน</td><td>คำอธิบายเซ็นเซอร์</td><td>ลำดับโซนที่ควบคุม</td></tr>
</table>

- ซอฟแวร์ส่วนเซิฟเวอร์และแสดงผลเวปแอฟพลิเคชัน

พัฒนาโดยใช้ Flask ซึ่งเป็นเฟรมเวิคที่พัฒนามาจากภาษา Python ทำหน้าที่เป็นตัวกำหนด URL ในการรับส่งข้อมูลระหว่างเวปแอฟพลิเคชันและเซิฟเวอร์ URL จะมีตั้งแต่การแสดงหน้าเวปแอฟพลิเคชัน, การส่งข้อมูลสั่งการรีเลย์, การบักทึกค่าบนฐานข้อมูลและการเรียกดูข้อมูลจากฐานข้อมูล โดยข้อมูลที่รับ-ส่งจะอยู่ในรูปแบบ JSON packet

ในการออกแบบหน้าเวปแอฟพลิเคชันเราใช้ Notepad++ ซึ่งเป็น Free Text Editor สามารถติดตั้งและใช้งานได้ฟรี สำหรับการทำเวปแบบตอบสนองรวดเร็วและสนันสนุนการเรียกดูในเบราเซอร์, แทปเล็ตรวมไปถึงสมาร์ทโฟนซึ่งมีขนาดหน้าจอการแสดงผลที่แตกต่าง เราได้นำ Jquery และ Jquery Mobile มาเป็นตัวช่วยใช้ในการพัฒนาร่วมกับภาษา HTML พร้อมทั้งเขียน javascript เพื่อเพิ่มฟังชั่นการแสดงผลผ่านหน้าเวปแอฟพลิเคชัน โดยเวปแอฟพลิเคชันแสดงผลทั้งหมดจะมีด้วยกัน 8 เทมเพลท แผนผังแสดงการทำงานของเวปแอฟพลิเคชันแสดงในรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 ลำดับการทำงานของซอฟแวร์ส่วนเซิฟเวอร์และแสดงผลเวปแอฟพลิเคชัน

## 3.4 การทดลองและวิเคราะห์ข้อมูล

### 3.4.1 การหาค่าความชื้นในดินที่ Field Capacity

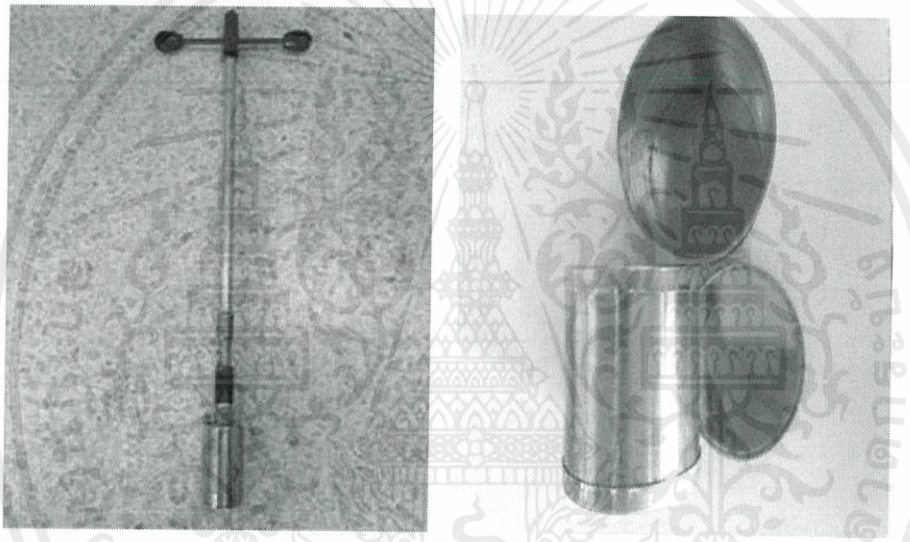
การทดลองนี้เป็นการหาค่าความชื้นในดินที่ Field Capacity โดยจะทำการเก็บตัวอย่างดินจากแปลงทดลอง จากนั้นใช้วิธีการวัดความชื้นในดินจากการชั่งน้ำหนักและอบแห้ง เพื่อนำค่าความชื้นในดินที่ Field Capacity ที่ได้จากการทดลองมาเป็นเกณฑ์สำหรับให้น้ำพืช ซึ่งการค่าเปอร์เซ็นต์ความชื้นโดยน้ำหนักทำได้โดยใช้สมการที่ 2.1

$$Pw = ( W_w / W_s ) \times 100$$

เมื่อ $P_w$ = เปอร์เซ็นต์ความชื้นโดยเทียบกับน้ำหนักของดินแห้ง

$W_w$ = น้ำหนักของน้ำในดิน

$W_s$ = น้ำหนักของดินที่อบให้แห้งด้วยเตาอบ



(a) (b)

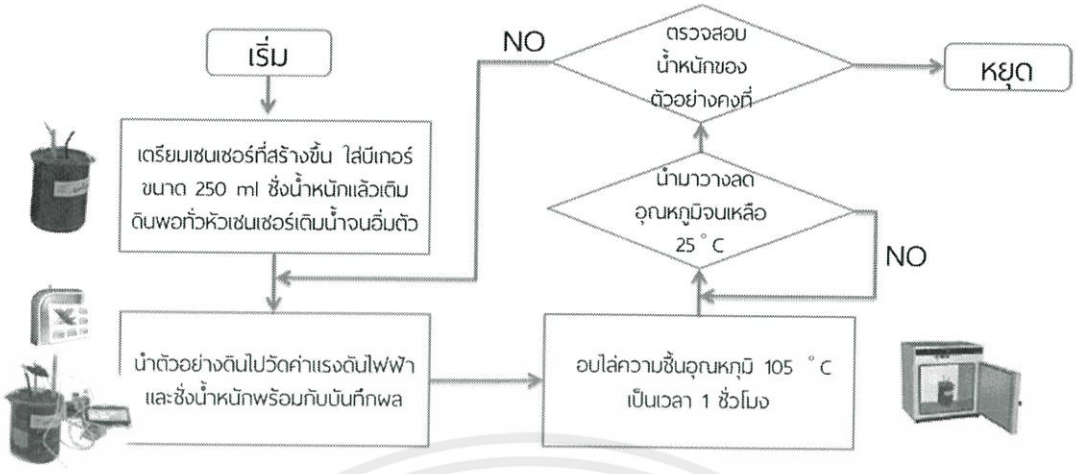
รูปที่ 3.13 อุปกรณ์สำหรับเก็บตัวอย่างดิน (a)แท่งขุดดิน (b)กระบอกเก็บตัวอย่างดิน

### 3.4.2 การหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า

การทดลองนี้จะทำการวัดความชื้นในดินและค่าแรงดันไฟฟ้า($V_{out}$) ที่ได้จากเซนเซอร์วัดความชื้นในดิน เพื่อนำค่าที่ได้มาสร้างสมการความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์ กับความชื้นในดิน จากนั้นนำสมการที่ได้ไปใช้ในการแปลงค่าจากแรงดันไฟฟ้าเป็นความชื้นในดิน โดยมีขั้นตอนดังนี้

ทดลองวัดค่าแรงดันไฟฟ้าและชั่งน้ำหนัก โดยวัดที่ความชื้นในแต่ละช่วง โดยนำเซนเซอร์ที่เตรียมไว้ใส่บีกเกอร์ขนาด 250 ml แล้วใส่ดินพอท่วมเซนเซอร์ เติมน้ำจนดินอิ่มตัว จากนั้นวัดค่าแรงดันไฟฟ้า เริ่มต้นและชั่งน้ำหนัก ต่อมานำดินไปอบไล่ความชื้นที่อุณหภูมิ 105°C เป็นเวลา 1 ชั่วโมง แล้วนำตัวอย่างดินมาลดอุณหภูมิจนเท่ากับอุณหภูมิห้องหรือ 25°C แล้วจึงวัดค่า ADC และชั่งน้ าหนักอีกครั้ง เก็บตัวอย่างอย่างนี้เรื่อยไป จนกว่าดินจะแห้ง โดยขั้นตอนการลดความชื้นเพื่อวัดเทียบเคียงค่า ADC กับแรงดันไฟฟ้าจะแสดงไว้ในรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.14 ลำดับขั้นตอนการลดความชื้นในดินเพื่อวัดแรงดันไฟฟ้า

นำค่าที่ได้จากการชั่งน้ำหนักมาหาความชื้นในดินโดยใช้สมก่ารที่ 2.1 จากนั้นนำค่าความชื้นในดินและแรงดันไฟฟ้าที่ได้มาสร้างกราฟความสัมพันธ์ระหว่าง แรงดันไฟฟ้าจากเซนเซอร์ กับความชื้นในดิน เพื่อหาสมการความสัมพันธ์ระหว่าง แรงดันไฟฟ้าจากเซนเซอร์ กับความชื้นในดิน สำหรับใช้ในการแปลงค่าแรงดันไฟฟ้าเป็นความชื้นในดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

## ผลการทดลอง

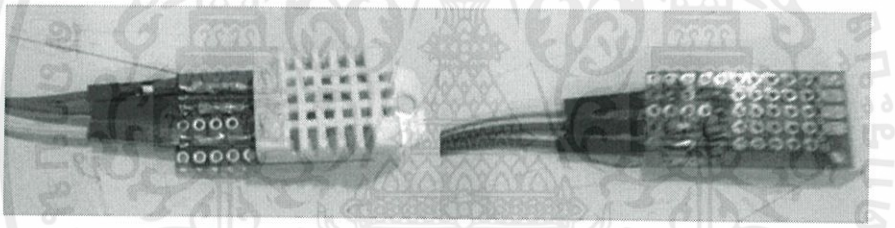
การออกแบบและพัฒนาระบบควบคุมการให้น้ำพืชสามารถแบ่งการศึกษาออกเป็นส่วนหลักๆ คือ การสร้างระบบควบคุม, การหาค่าความชื้นในดินที่ Field Capacity ,การหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า และการทดสอบการทำงานของระบบควบคุมการให้น้ำ ดั้งนั้นจึงแบบผลการศึกษาออกเป็นดังนี้

### 4.1 การต่อวงจรของส่วนตรวจวัดความชื้น

ในการต่อวงจรของส่วนตรวจวัดความชื้นจะแบ่งแยกการต่อออกแป็นส่วนต่างๆเพื่อให้สะดวกต่อการแก้ไขปรับปรุงและทดสอบ ซึ่งสามาถแบ่งออกได้ดังนี้

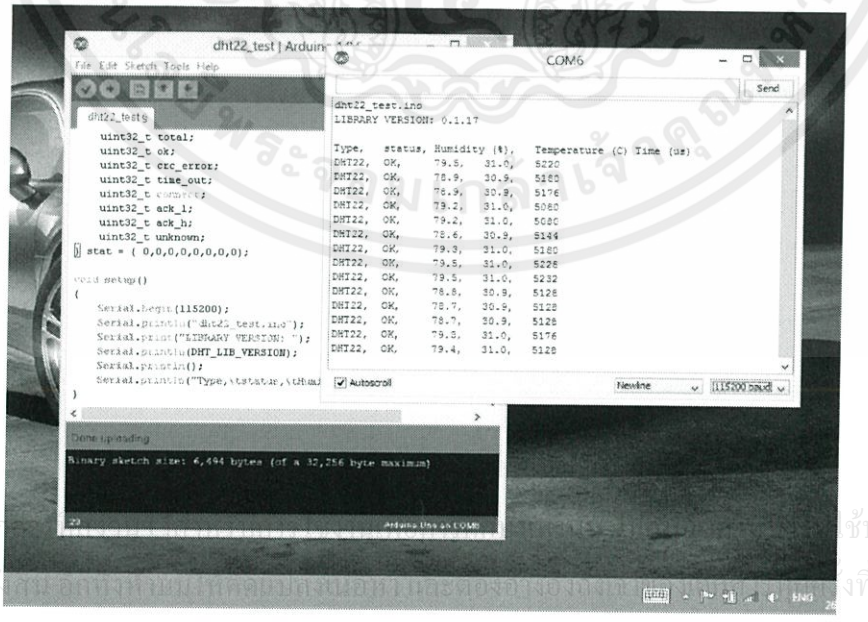
#### 4.1.1 การต่อโมดูล DHT 22 เข้ากับ Arduino

จากการต่อโมดูล DHT 22 เข้ากับ Arduino จะแสดงในรูปที่ 4.1 โดยทำการต่อขา DHT 22 เข้ากับขา ของArduino ตามวิธีการในบทที่ 3



(a) (b)

รูปที่ 4.1 การต่อโมดูล DHT 22 (a)ด้านหน้า (b)ด้านหลัง

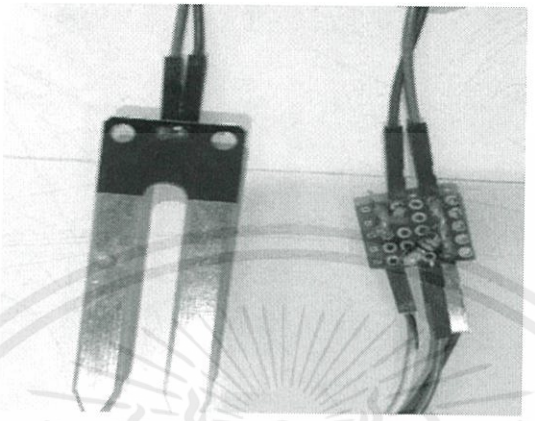



เอกสารนี้เป็นเอ... ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้ง... งที่มีการนำไปใช้

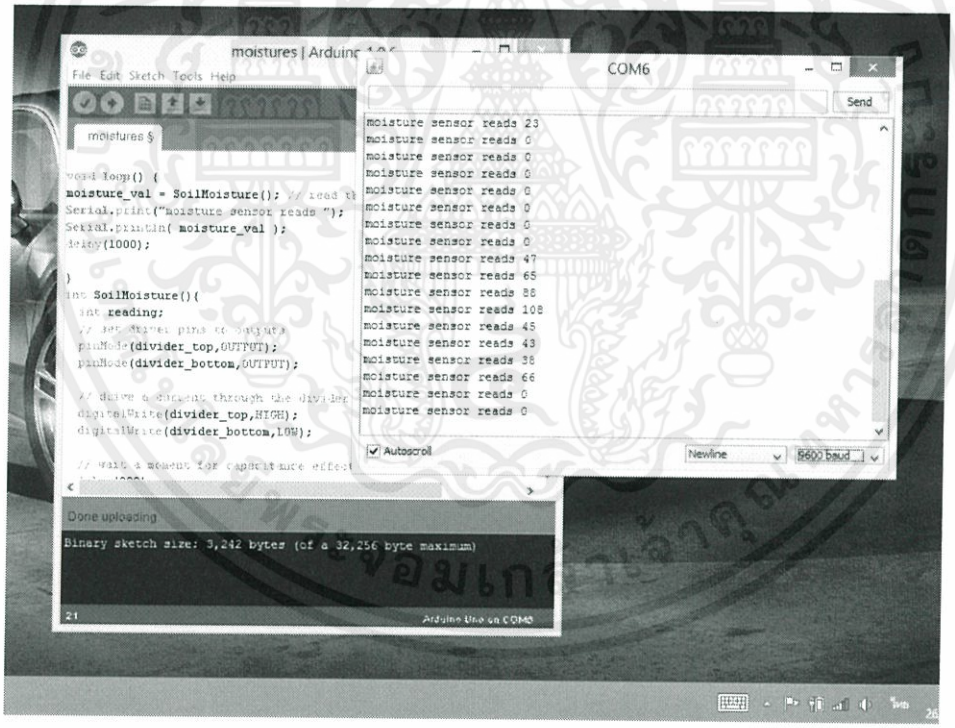
รูปที่ 4.2 การทดสอบการทำงานของโมดูล DHT 22

### 4.1.2 การต่อ Soil moisture sensor เข้ากับ Arduino

จากการต่อ Soil moisture sensor เข้ากับ Arduino จะแสดงในรูปที่ 4.2 โดยทำการต่อขาบวกและขาลบของSoil moisture sensor เข้ากับขาของArduino ตามวิธีการในบทที่ 3



รูปที่ 4.3 การต่อ Soil moisture sensor

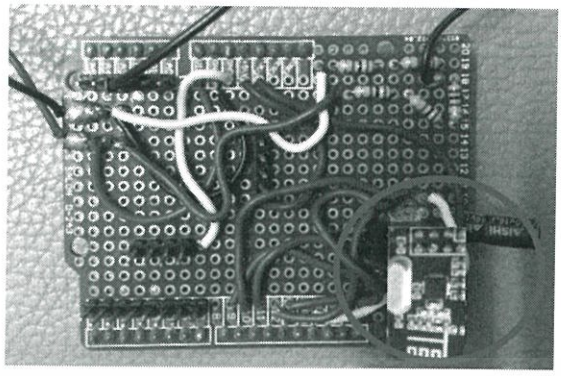



รูปที่ 4.4 การทดสอบการทำงานของ Soil moisture sensor

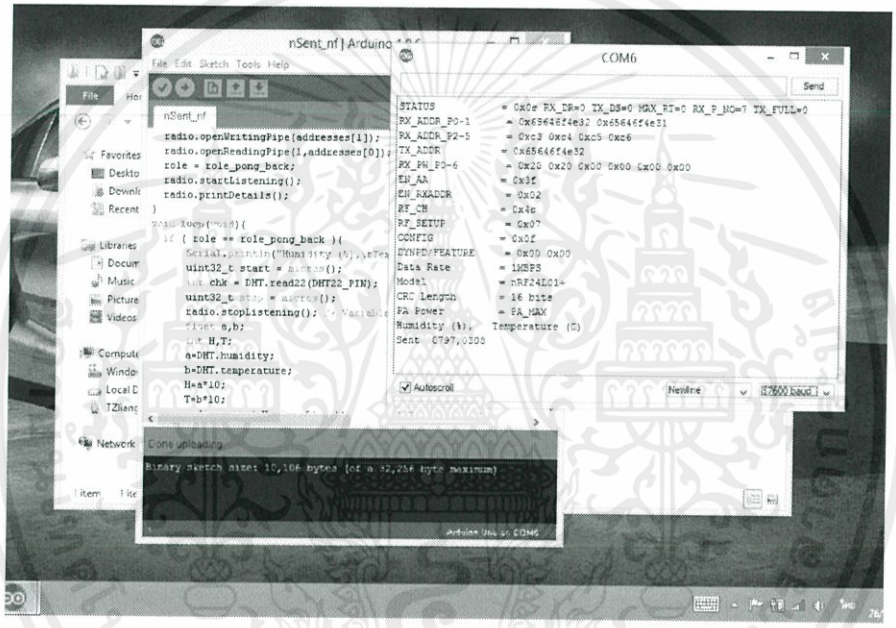
### 4.1.3 การต่อโมดูลnRF24L01 เข้ากับ Arduino

การต่อโมดูลnRF24L01 เข้ากับ Arduino จะแสดงในรูปที่ 4.3 โดยทำการต่อบนบอร์ดไข่ปลาที่ติดกับArduino และใช้ตัวเก็บประจุขนาด 10 µF โดยตามวิธีการในบทที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

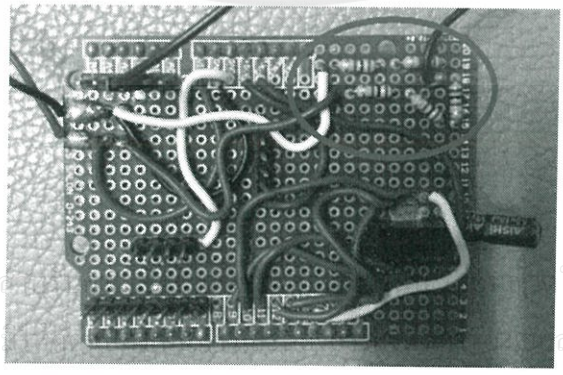


รูปที่ 4.5 การต่อโมดูลnRF24L01 บนบอร์ดไข่ปลา



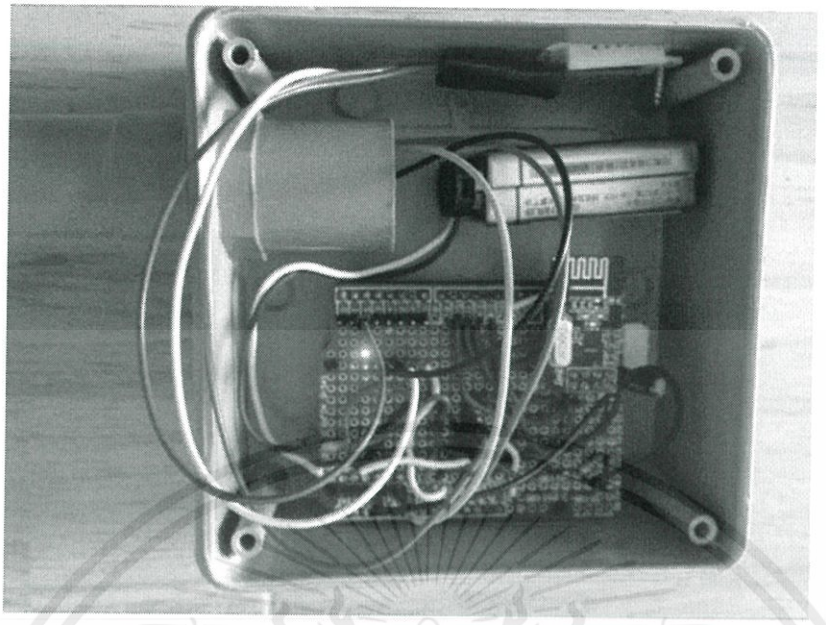

รูปที่ 4.6 การทดสอบการทำงานของโมดูลnRF24L01

- การต่อวงจรวัดแบตเตอรี่โดยใช้วงจร voltage divider แปลงแรงดันไฟฟ้าจาก 9V เป็น 5V เนื่องจาก Arduino โดยการต่ออนุกรมให้ R1 = 5kΩ และ R2 = 4kΩ แสดงการต่อวงจรในรูปที่ 4.7



เอกสารนี้เป็นเอกสารที่สงวนไว้ ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้าม สารทุกครั้งที่มีการนำไปใช้

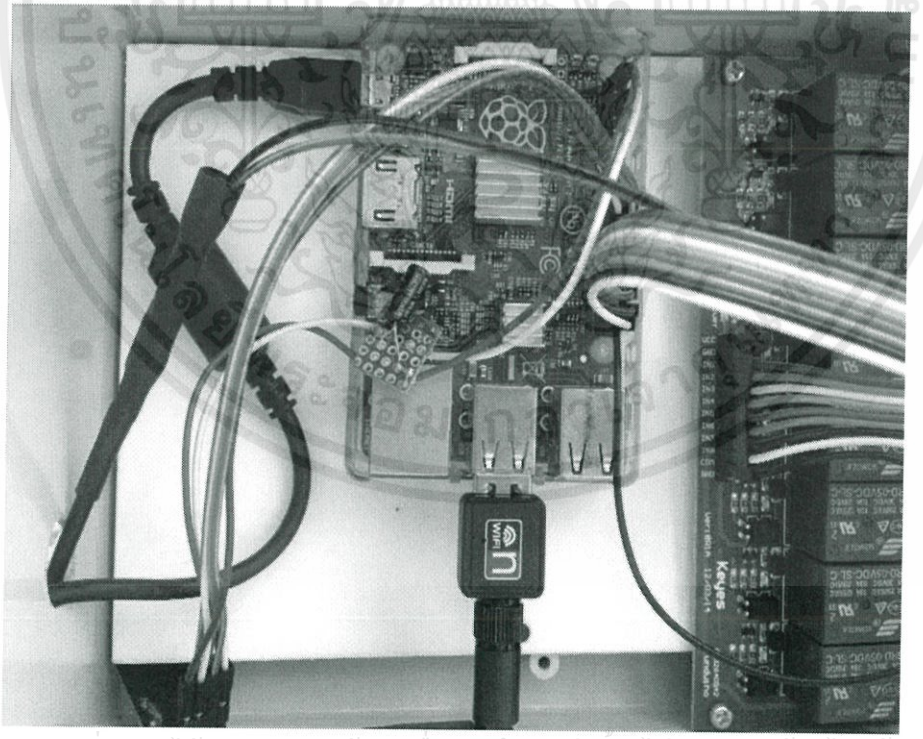
รูปที่ 4.7 การต่อวงจรวัดแบตเตอรี่โดยใช้วงจร voltage divider



รูปที่ 4.8 วงจรของส่วนตรวจวัดความชื้น

## 4.2 การต่อวงจรของส่วนควบคุมการทำงานหลัก

ในการต่อวงจรส่วนควบคุมการทำงานหลักจะประกอบได้ด้วย nRF24l01, DS3231 และรีเลย์บอร์ดกับ Raspberry Pi โดยใช้วิธีการต่อจากบทที่ 3 จะแสดงในรูปที่ 4.9

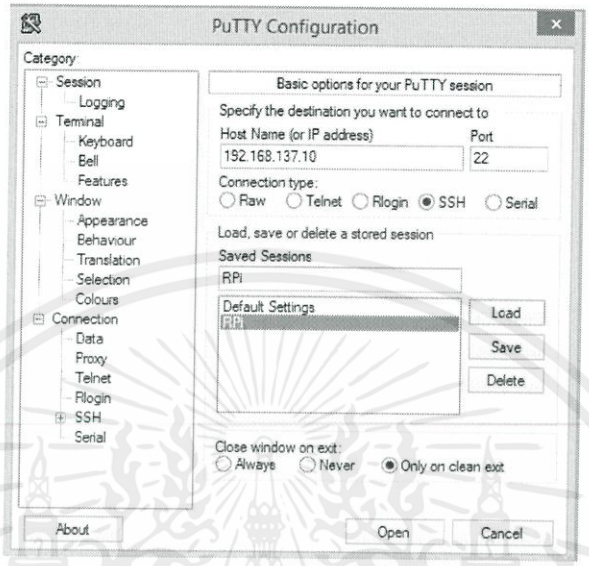


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

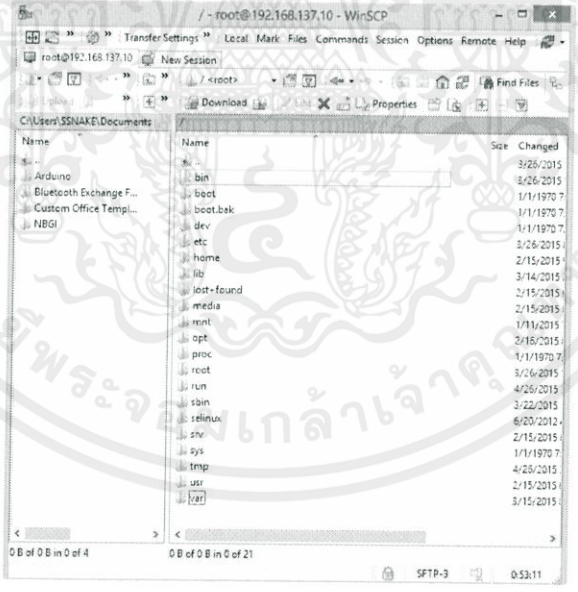
รูปที่ 4.9 วงจรของส่วนควบคุมการทำงานหลัก

## 4.3 การออกแบบซอฟแวร์บน Raspberry Pi

การติดตั้งระบบปฏิบัติการ Raspbian ที่เสร็จสมบูรณ์ ซอฟแวร์ในการทำงานต่างๆจะถูกเขียนโดยผ่าน PuTTy และ WinSCP จะแสดงในรูปที่ 4.10 และ 4.11 ตามลำดับ




รูปที่ 4.10 โปรแกรม PuTTy




รูปที่ 4.11 โปรแกรม WinSCP

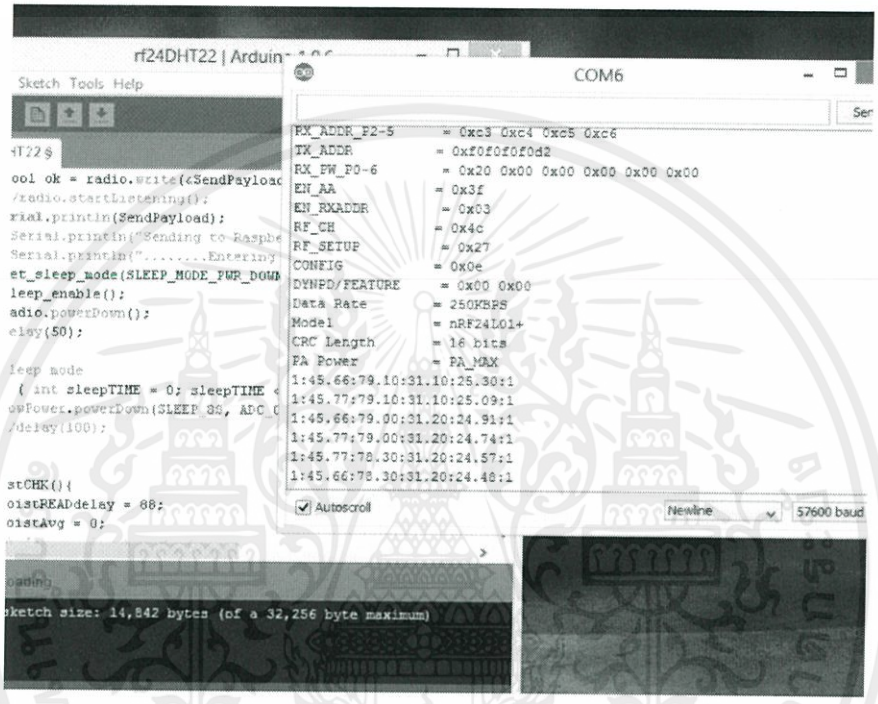
### 4.3.1 ซอฟแวร์ส่วนรับผลจากเซ็นเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งนี้ ห้ามมิให้ทำสำเนาเนื้อหาของเอกสารนี้ไม่ว่ากรณีใดๆ

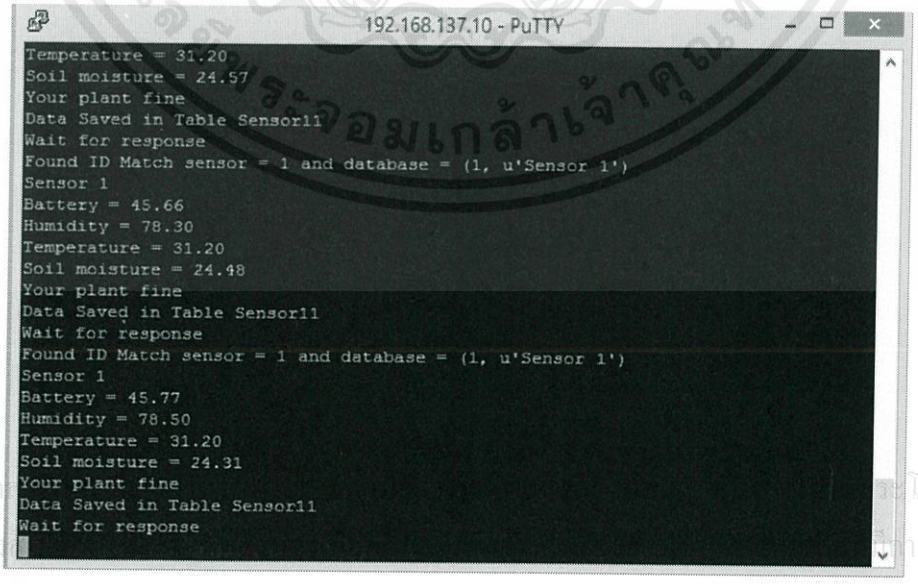
ซอฟแวร์ในส่วนนี้ พัฒนาโดยใช้ภาษา Python และไลบรารี่ lib_nrf24 เซ็นเซอร์ตรวจวัดความชื้นและ Base Station ถูกเขียนโปรแกรมขึ้นมาให้สื่อสารกันผ่านโมดูสื่อสารไร้สาย nRF24l01 ในการทดสอบการสื่อสารกันของทั้งสองส่วนนี้ ได้ใช้ Serial monitor ของ Arduino

ร่วมกับ PuTTy ที่เชื่อมต่อกับ Raspberry Pi รูปที่4.12 และ4.13 แสดงผลการส่งข้อมูลจาก Arduino ไปยัง Raspberry Pi

ในการบันทึกค่าที่รับจากเซนเซอร์ โปรแกรมจะตรวจสอบว่าผู้ใช้งานมีการเพิ่มเซ็นเซอร์ที่ใช้เข้ามาในระบบหรือไม่ หากมีการเพิ่มเข้ามาจะตรวจสอบอีกว่าเป็นการบันทึกครั้งแรกหรือไม่ หากเป็นการบันทึกครั้งแรกจะทำการสร้างตารางบันทึกขึ้นมาก่อน เพื่อใช้ในการบันทึกครั้งต่อๆไป



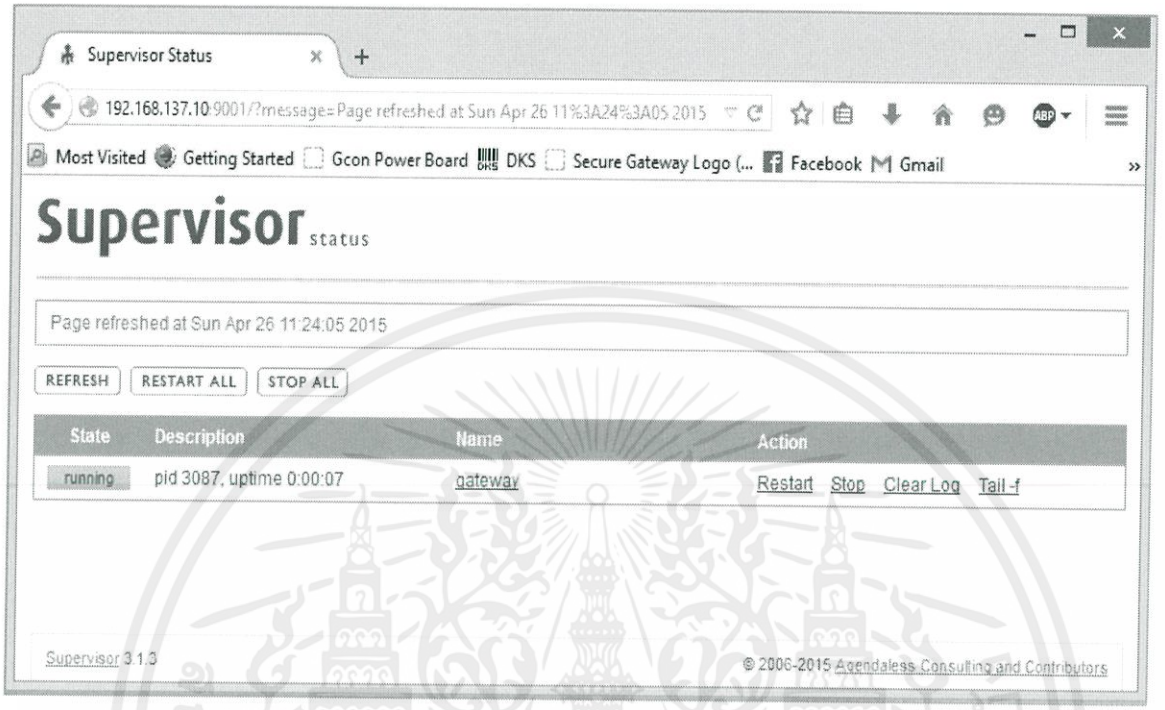

รูปที่ 4.12 Arduino ส่งค่าให้ Raspberry Pi ผ่านnRF24L01



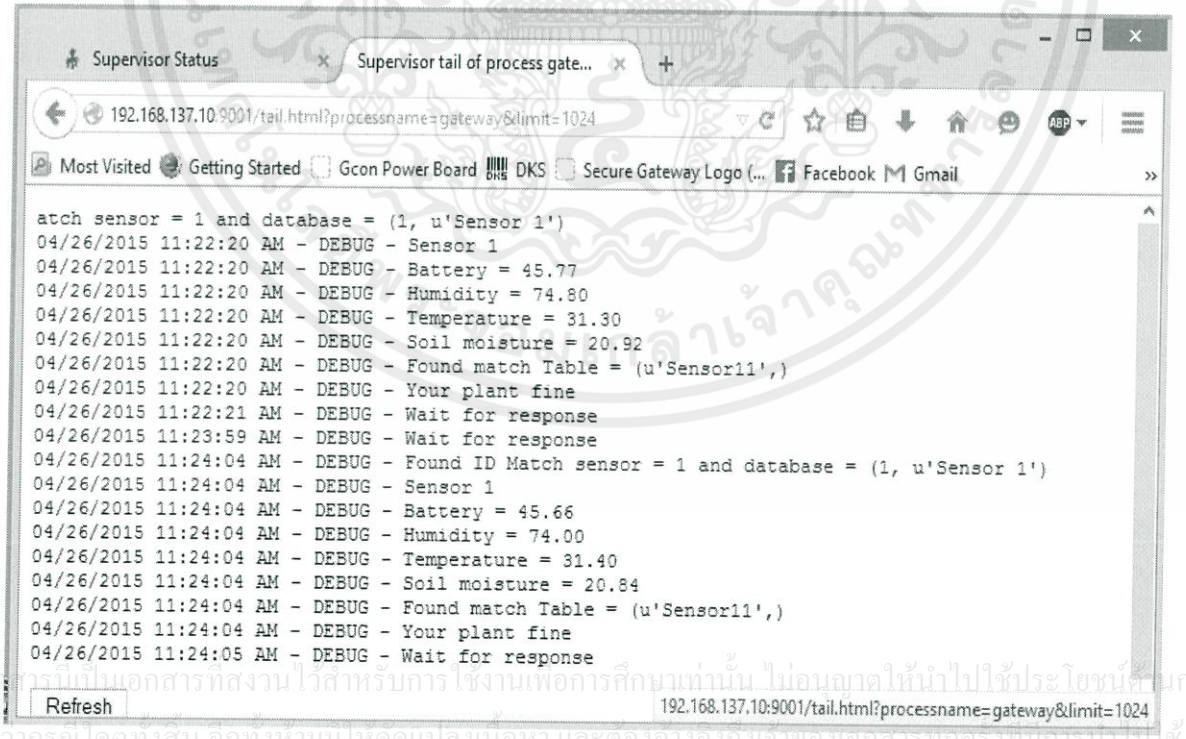

รูปที่ 4.13 Raspberry Pi รับค่าจาก Arduino ผ่านnRF24L01

เอกสารนี้เป็นเอก ระโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้ง การนำไปใช้

ในการใช้งานจริงจำเป็นต้องสั่งทำงานซอฟแวร์ตัวนี้เป็นแบคกราวด์ จึงได้นำซอฟแวร์ Supervisord มาช่วยในส่วนนี้ ผลการทำงานเป็นดังรูปที่ 4.14 และ 4.15




รูปที่ 4.14 ซอฟแวร์ Supervisord

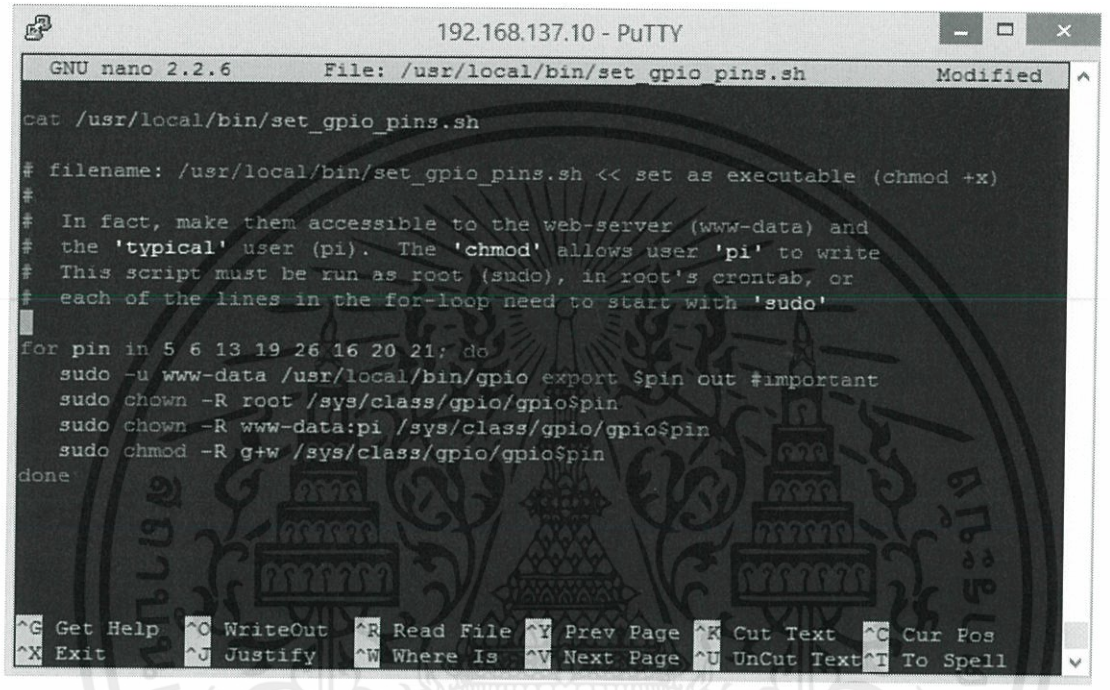



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.15 การทำงานของซอฟแวร์ Supervisord

### 4.3.2 ซอฟแวร์ส่วนควบคุมรีเลย์

ซอฟแวร์ในส่วนนี้ พัฒนาโดยใช้ภาษา Python และไลบราลี่ WiringPi เพราะเนื่องจากการสั่งใช้งาน GPIO ผ่านเวป หรือ แอปพลิเคชันบน Raspberry Pi ผู้ใช้งานจะอยู่ในสถานะ www-data จึงมีการแก้ปัญหาโดยการเขียน Shell command อนุญาติให้ ผู้ใช้ที่อยู่ในสถานะ www-data ใช้งาน GPIO ได้ และใช้คำสั่ง crontab สั่งการทำงานของ Shell command นี้ทุกครั้งเมื่อมีการเปิดเครื่อง ดังรูปที่ 4.16




รูปที่ 4.16 โปรแกรม shell command

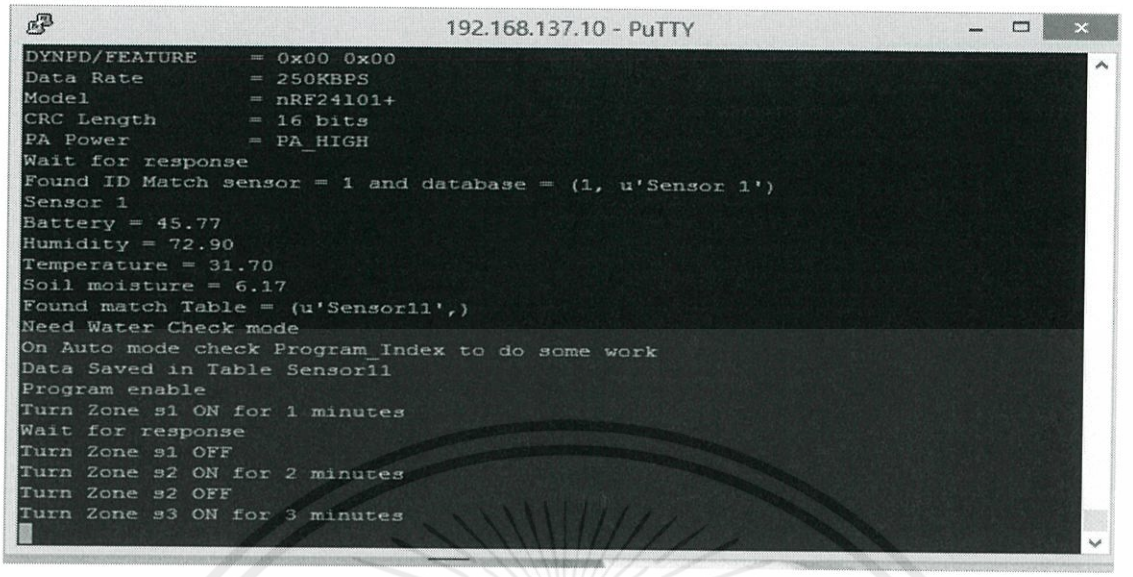
การใช้งานรีเลย์สามารถใช้งานได้ 2 โหมดคือ แมนนวลโหมดและออโต้โหมด ออกแบบมาให้ใช้งานรีเลย์ได้ทั้งหมด 8 ตัว ตัวแรกสำหรับระบบที่มีปั๊มหลักที่ต้องกำหนดให้ทำงานพร้อมกับรีเลย์ตัวอื่น แมลนวลโหมด ได้เขียนชุดคำสั่งแยกไว้เพื่อให้ซอฟแวร์ควบคุมส่วนอื่นเรียกใช้ได้ โดยสามารถเปิด-ปิด

การใช้งานรีเลย์ ในออโต้โหมด เขียนเป็นชุดคำสั่งรวมไว้กับส่วนรับผลจากเซ็นเซอร์ โดยจะตรวจสอบตาราง Mode Status ในฐานข้อมูลทุกๆ 2 วินาที หากมีการบันทึกค่าในตารางเป็น “auto” จึงจะทำการตรวจสอบสถานะเซ็นเซอร์ที่รับมาตามเงื่อนไขในรูปที่ 4.18

```
if(Moistureavg <= 4) {
  State=Error;
}else if(Moistureavg > 35){
   State=Error;
}else if(Moistureavg > 4 && Moistureavg < 20){
  State=need_water;
}else{
  State=Normal;
}
```

รูปที่ 4.17 ฟังก์ชั่นตรวจสอบสถานะความชื้นของ Arduino

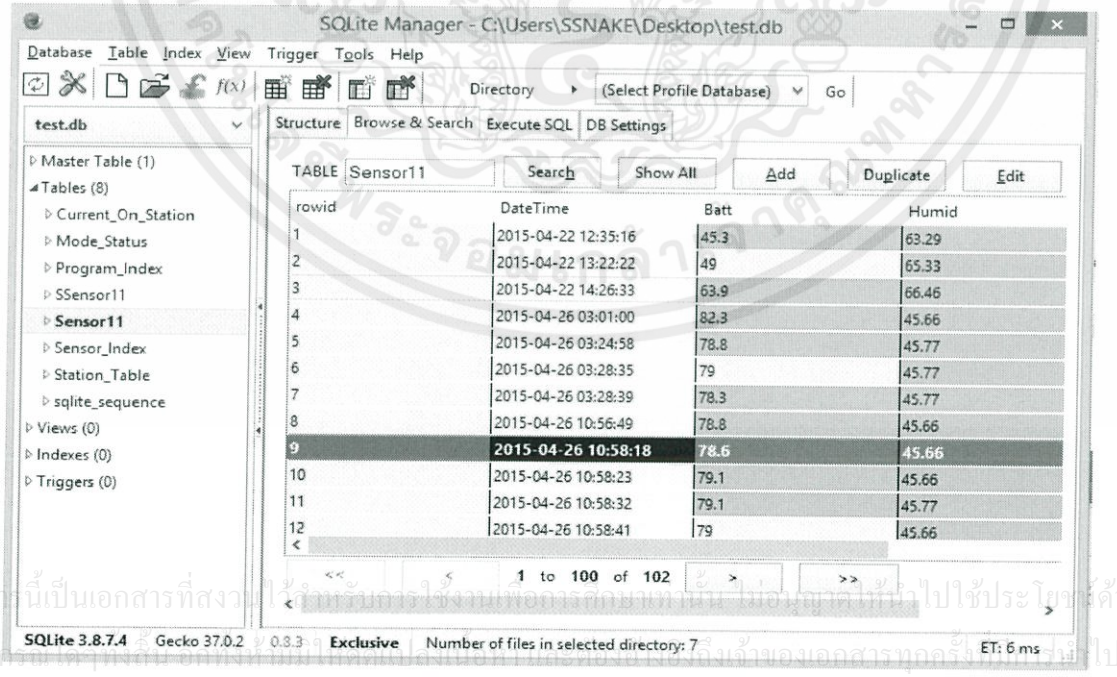
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.18 ตรวจสอบสถานะเซ็นเซอร์ที่รับมาตามเงื่อนไข

4.3.3 ซอฟแวร์ส่วนฐานข้อมูล

พัฒนาโดยใช้ SQLite3 และภาษา Python .ในการสร้างฐานข้อมูล และใช้ซอฟแวร์ SQLite Manager เพื่อความสะดวกในการจัดการฐานข้อมูลที่สร้างขึ้นมา รูปที่ 4.19 แสดงฐานข้อมูลของระบบ ค่าจากเซ็นเซอร์จะถูกบันทึกตามเซนเซอร์ที่ผู้ใช้งานเพิ่มเข้ามาในฐานข้อมูล โดยหากเป็นการบันทึกครั้งแรกจะสร้างตารางบันทึกขึ้นมาใหม่ชื่อของตารางจะอ้างอิงจากชื่อและไอดีของเซ็นเซอร์ที่ผู้ใช้เพิ่มเข้ามาในตาราง Sensor_Index และหากผู้ใช้มีการเพิ่มโปรแกรมการใช้งานออโต้โหมด จะมีการบันทึกข้อมูลลงตาราง Program_Index เพื่อเป็นข้อมูลให้ ออโต้โหมดทำงาน

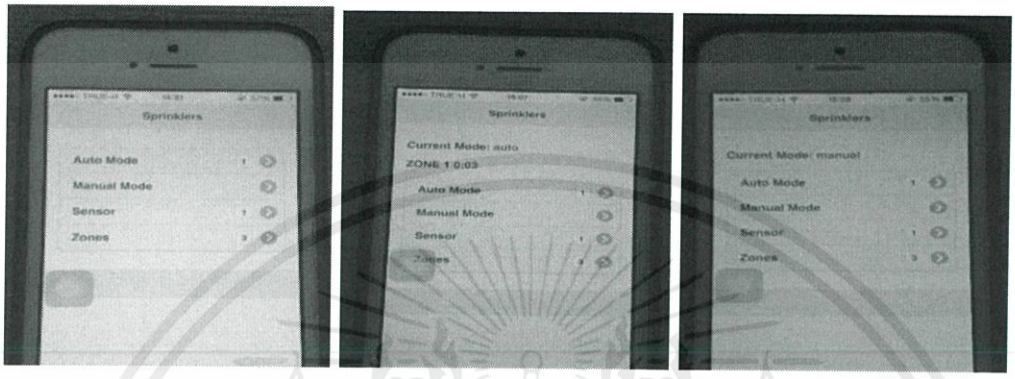



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.19 ฐานข้อมูลของระบบการให้น้ำพืช

4.3.4 ซอฟแวร์ส่วนเซิฟเวอร์และแสดงผลเวปเพจ

การทำงานของเซิฟเวอร์บน Raspberry Pi จะแสดงเมื่อผู้ใช้งานเชื่อมต่อมายังเซิฟเวอร์ผ่าน http://192.168.137.10 ในเครือข่ายเดียวกันกับ Raspberry Pi เซิฟเวอร์ก็จะตอบสนองโดยการทำคำสั่งแสดงหน้าแรกของเวปแอปพลิเคชัน ผู้ใช้งานสามารถเลือกใช้คำสั่งต่างๆผ่านทางเวปแอพพลิชันได้ โดยเซิฟเวอร์ก็จะตอบสนองคำสั่งนั้นๆตามฟังชั่นที่กำหนดไว้บนเซิฟเวอร์

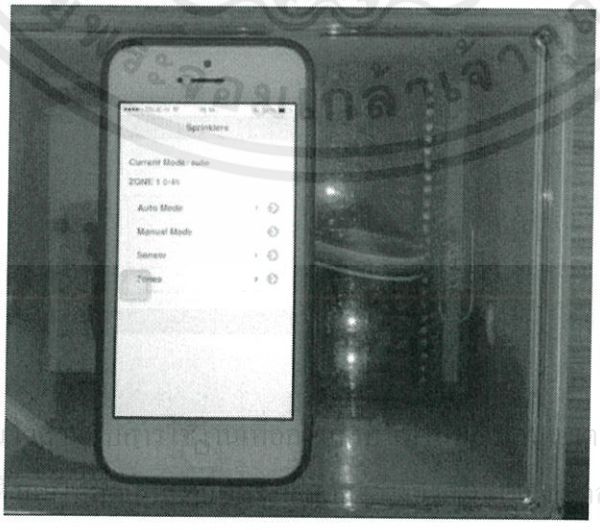



(a) (b) (c)

รูปที่ 4.20 หน้าแรกของเวปแอปพลิเคชันในโหมดต่างๆ (a) ไม่มีการใช้งานทั้ง 2 โหมด (b) Auto Mode (c) Manual Mode

จากรูปที่ 4.20 แสดงผลหน้าแรกของเวปแอพพลิเคชันบนสมาร์โฟน ในการใช้ซอฟแวร์ส่วนนี้จำเป็นต้องทำการเชื่อมต่ออินเตอร์เน็ตที่มี Base Station ในเครือข่าย หน้าแรกของเวปแอพพลิเคชันโดยแสดง 4 เมนูการใช้งาน

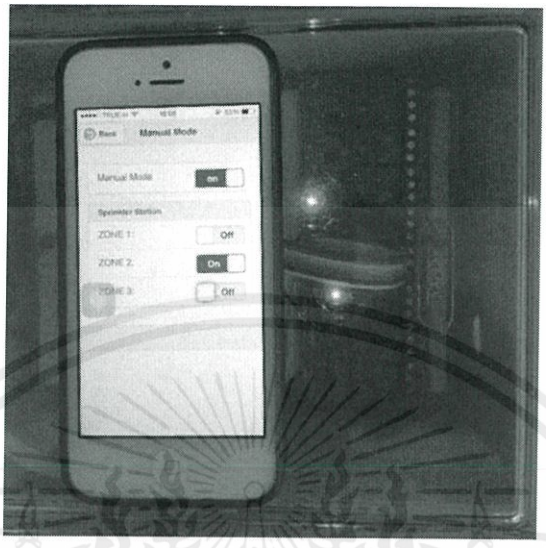
- เมนูแรก คือ Auto Mode เมนูนี้สามารถเปิดการใช้งาน ออโต้โหมด และสามารถเพิ่มโปรแกรมการควบคุมโซน โดยกำหนดได้ว่า จะใช้ค่าจากเซ็นเซอร์ตัวไหน เพื่อควบคุม โซนตัวไหนบ้าง และสามารถแก้ไข ลบ โปรแกรมเดิมได้พร้อมทั้งในขณะที่ใช้งานออโต้โหมดอยู่ หน้าแก้ไขโปรเกรมจะแสดงเวลาที่เซ็นเซอร์จะวัดค่าในครั้งต่อไป และเมื่อมีการเปิดรีเลย์ หน้าแรกจะแสดงเวลาการเปิดใช้งานรีเลย์ดังรูปที่ 4.21




เอกสารนี้เป็นเอกสารที่สงวน ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้ง สารทุกครั้งที่มีการนำไปใช้

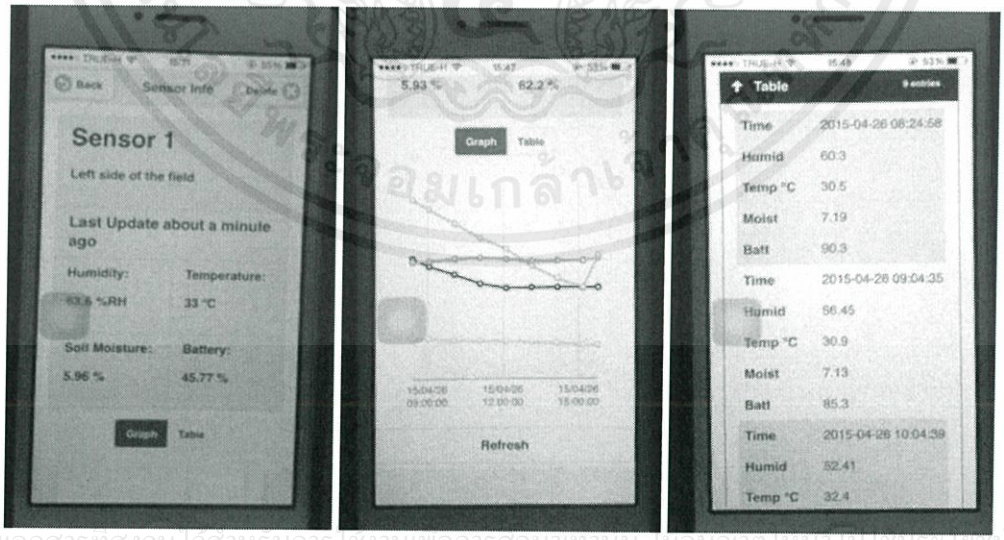
รูปที่ 4.21 การทำงานของเมนู Auto Mode ในการเปิด-ปิดรีเลย์

- เมนูที่สอง คือ เมนู Manual Mode เมื่อเปิดใช้งาน จะแสดงโซนที่สามารถใช้งานได้ สามารถกดปุ่ม ON-OFF ในแต่ละโซนเพื่อสั่งการรีเลย์ หน้าแรกจะแสดงสถานการณ์ทำงานในโหมดนี้ ดังรูปที่ 4.22




รูปที่ 4.22 การทำงานของเมนู Manual Mode ในการเปิด-ปิดรีเลย์

- เมนูที่สามคือ Sensor สามารถเพิ่มตารางบันทึกข้อมูลเซ็นเซอร์ โดยต้องกำหนดไอดีให้ตรงกับเซ็นเซอร์ที่จะใช้งาน ตั้งชื่อเซ็นเซอร์ เพิ่มคำอธิบายเซ็นเซอร์ เมื่อเพิ่มเสร็จและเซ็นเซอร์ทำการส่งข้อมูลมาแล้ว สามารถเรียกดูค่าต่างๆได้โดยเลือกไปที่ชื่อเซ็นเซอร์ จะแสดงข้อมูลอัพเดทล่าสุดของเซ็นเซอร์นั้นๆ ตั้งแต่ค่าความชื้นอากาศ, ความชื้นในดิน, อุณหภูมิและแบตเตอรี่ที่เหลือ พร้อมทั้งสามารถแสดงข้อมูลย้อนหลัง 10 วัน ในรูปแบบของตารางและกราฟ ดังรูปที่ 4.23

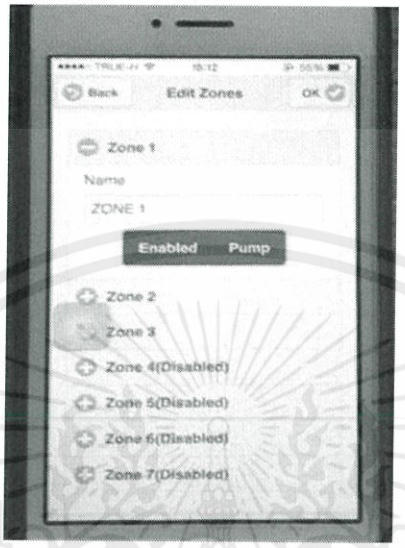



(a) (b) (c)

รูปที่ 4.23 หน้าของเวปแอปพลิเคชันในเมนูเซ็นเซอร์(a) แสดงข้อมูลอัพเดทล่าสุดของเซ็นเซอร์ (b) ข้อมูลย้อนหลังในรูปแบบกราฟ (c) ข้อมูลย้อนหลังในรูปแบบตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

- เมนูที่สี่คือ Zones สามารถเพิ่ม-ลดการใช้งานในแต่ละโซนและกำหนดให้โซนนั้นๆทำงานพร้อมกับปั้มหลักได้ดังรูปที่ 4.24 การตั้งค่าต่างๆในเมนูนี้จะส่งผลไปยังการแสดงผลโซนที่ใช้งานได้ทั้งใน Manual Mode และ Auto Mode




รูปที่ 4.24 หน้าของเวปแอปพลิเคชันในเมนู Zones

## 4.4 การหาค่าความชื้นในดินที่ Field Capacity

**การทดลองนี้เป็นการหาค่าความชื้นในดินที่** Field Capacity **โดยใช้วิธี**การวัดความชื้นในดินจากการชั่งน้ำหนักและอบแห้งโดยใช้สมการที่ 2.1 ในการคำนวนหาค่าความชื้นในดินซึ่งความชื้นในดินที่ Field Capacity เฉลี่ยเท่ากับ 31.86% ± 2.97 นำค่าความชื้นในดินที่ได้มาเป็นเกณฑ์ในเขียนฟังก์ชั่นควบคุมการให้น้ำแสดงในรูปที่ 4.25

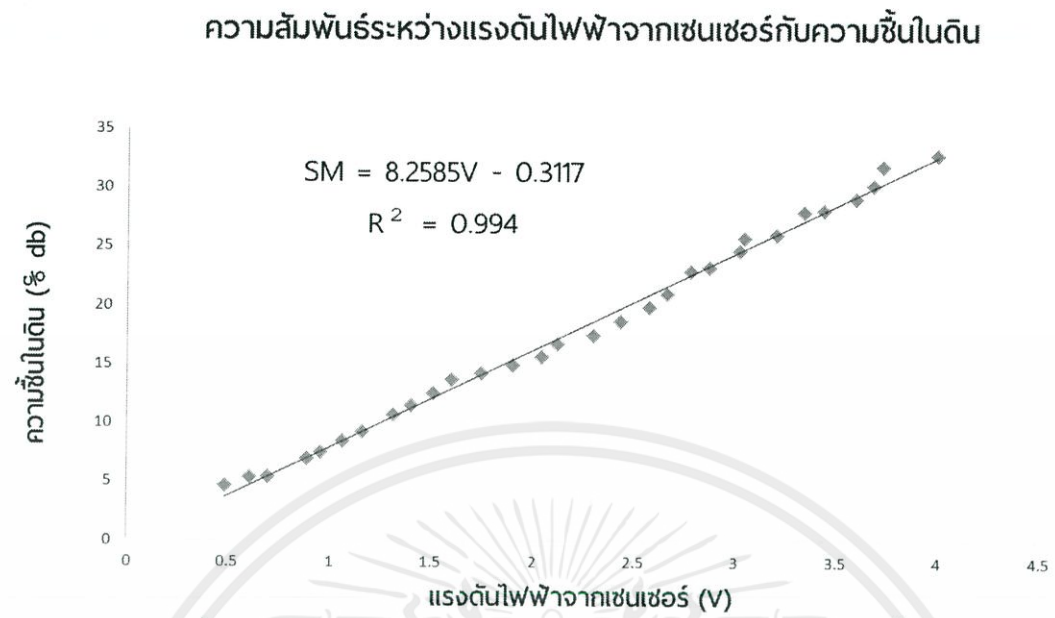
```
float a=Smavg*0.004888;
Moistureavg =  (0.3494*(pow(2,a)))+(6.719*a)+1.0114;
if(Moistureavg <= 4) {
 State=Error;
}else if(Moistureavg > 35){
   State=Error;
}else if(Moistureavg > 4 && Moistureavg < 20){
   State=need_water;
}else{
   State=Normal;
}
  digitalWrite(DHT22_PIN, HIGH);
  delay (38);
  //uint32_t start = micros();
```

รูปที่ 4.25 ฟังก์ชั่นควบคุมการให้น้ำที่ใช้ความชื้นที่ Field Capacity เป็นเกณฑ์

## 4.5 การหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า

การหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้าได้นำค่าที่ได้จากการทดลองมาสร้างกราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์ กับความชื้นในดินได้ตามรูปที่ 4.2 และรูปที่ 4.26

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้



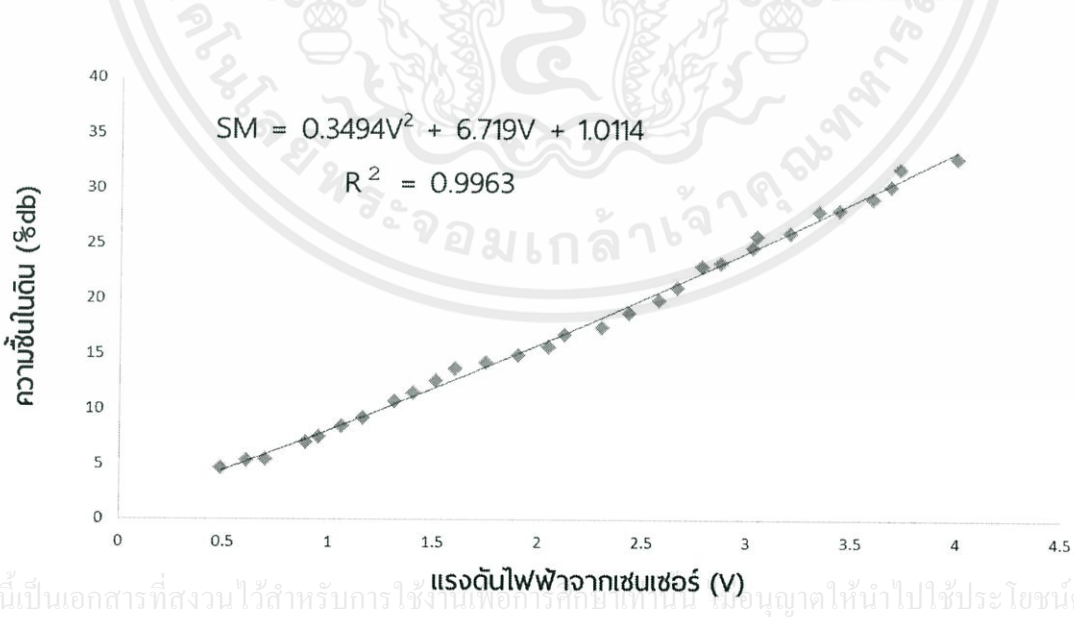

รูปที่ 4.26 ความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์กับความชื้นในดิน

จากกราฟในรูปที่ 4.26 จะได้สมการความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์กับความชื้นในดินดิน แบบสมการเชิงเส้น

$$SM = 8.2585V - 0.3117 \quad (4.1)$$

เมื่อ SM = ความชื้นดิน (%)

V = ค่าความต่างศักย์ (V)




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.27 ความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์กับความชื้นในดิน

จากกราฟในรูปที่ 4.27 จะได้สมการความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์กับความชื้นในดินแบบสมการโพลิโนเมียลกำลัง2

$$SM = 0.6262V2 + 4.6074V + 1.5913 \quad (4.2)$$

เมื่อ SM = ความชื้นดิน (%)

V = ค่าความต่างศักย์ (V)

จากการทดลองแทนค่าแรงดันไฟฟ้าในสมการที่ 4.1 และ 4.2 เพื่อหาค่าความชื้นในดินผลที่ออกมาคือสมการที่4.2 มีค่าความชื้นในดินใกล้เคียงกับการทดลองมากกว่าสมการที่ 4.1 ดั้งนั้นจึงนำสมการที่4.2 มาใช้ในการเขียนฟังก์ชั่นสำหรับแปลงค่าจากแรงดันไฟฟ้าเป็นความชื้นในดินดังรูปที่ 4.28

```
for ( int i = 0 ; i < 3 ; i++ ) {
 Moisture += moistCHK();
}
Smavg = (Moisture/3);
float a=Smavg*0.004888;
Moistureavg =  (0.3494*(pow(2,a)))+(6.719*a)+1.0114;
if(Moistureavg <= 4) {
 State=Error;
}else if(Moistureavg > 35){
 State=Error;
}else if(Moistureavg_      && Moistureavg < 20){
 State=need_water.
}else{
```

$$SM = 0.3494V^2 + 6.719V + 1.0114$$

```
int chk = DHT.read22(DHT22_PIN);
//uint32_t stop = micros();
float DH Temp;
```

รูปที่ 4.28 ฟังก์ชั่นสำหรับแปลงค่าจากแรงดันไฟฟ้าเป็นความชื้นในดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

# สรุปผลการทดลอง

## 5.1 การต่อวงจรของส่วนตรวจวัดความชื้นและส่วนควบคุมการทำงานหลัก

จากการต่อวงจรส่วนตรวจวัดความชื้นและส่วนควบคุมการทำงานหลัก เพื่อสร้างระบบควบคุมการให้น้ำพืช โดยการต่อวงจรในบางวงจรประสบปัญหาในเรื่องการจ่ายไฟที่ไม่เสถียร หรือจแรงดันไฟฟ้าสูงเกินไปทำให้อุปกรณ์บางชนิดเสียหาย ทางผู้วิจัยจึงได้นำเอาตัวต้านทานและตัวเก็บประจุเข้ามาช่วยในการต่อวงจรของระบบ เพื่อลดแรงดันไฟฟ้าและทำให้จ่ายไฟเสถียรมากขึ้น โดยวงจรที่ต่อสามารถวัดค่า รับค่า ส่งค่า และประมวณผลข้อมูลได้

## 5.2 การหาค่าความชื้นในดินที่ Field Capacity

จากการทดลองหาค่าความชื้นดินที่ Field Capacity ของแปลงทดลองโดยใช้วิธีการวัดความชื้นในดินจากการชั่งน้ำหนักและอบแห้ง ซึ่งจากการทดลองได้เก็บตัวอย่างดิน 6 ตัวอย่าง มาทำการอบแห้งและชั่งน้ำหนัก ซึ่งจากการทดลองได้ค่าความชื้นในดินที่ Field Capacity ของแปลงทดลองมีค่าเฉลี่ยเท่ากับ 31.86% ± 2.97

## 5.3 การหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า

จากการทดลองหาค่าความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า($V_{out}$) ที่ได้จากเซนเซอร์วัดความชื้นในดิน พบว่าค่าที่อ่านได้สูงสุดอยู่ที่ในช่วงแรงดันไฟฟ้าเท่ากับ 4V ทั้งที่ Microcontroller สามารถรองรับค่าได้สูงสุดถึง 5V ซึ่งอาจเกิดจากการเลือกใช้ตัวต้านทานที่มีค่าความต้านทานสูงเกินไปดังนั้นแม้เปียกน้ำมากเท่าใดค่าสูงสุดที่ได้จะไม่ถึง 5V เมื่อนำค่าที่ได้มาสร้างสมการความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากเซนเซอร์กับความชื้นในดิน จะได้ความสมการความสัมพันธ์แบบเชิงเส้นและแบบโพลิโนเมียลกำลัง2 ซึ่งสมการแบบโพลิโนเมียลกำลัง2 มีความแม่นยำในการแปลงค่ามากกว่าสมการแบบเชิงเส้น จึงนำสมการแบบโพลิโนเมียลกำลัง2 ไปใช้ในการเขียนฟังก์ชั่นสำหรับแปลงค่าความชื้นในดิน

## 5.4 ผลการทดสอบระบบการให้น้ำพืชผ่านแอปพลิเคชัน

- ระบบสามารถสั่งการรีเลย์เพื่อเปิด-ปิดโซลินอยด์วาล์ว สำหรับจ่ายน้ำให้แก่พืชได้ตามค่าที่ได้จากเซนเซอร์

- ระบบสามารถควบคุมสั่งการและแสดงค่าความชื้น-อุณหภูมิของอากาศ และความชื้นในดินผ่านทางแอปพลิเคชันได้แบบ Real-Time

- แอปพลิเคชันและระบบควบคุมการให้น้ำพืชที่ออกแบบ สร้าง และทดสอบสามารถทำงานได้ตามคำสั่งที่ตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5.5 ข้อเสนอแนะ**

จากการออกแบบ สร้างและทดสอบควบคุมระบบการให้น้ำพืชผ่านแอปพลิเคชันบนสมาร์ทโฟน ยังมีส่วนที่สามารถนำไปพัฒนาต่อได้ดังนี้

- จัดทำในส่วนระบบการจ่ายน้ำโดยการติดตั้งโซลินอยด์วาล์วและสปริงเกอร์
- พัฒนาในเรื่องของแบตเตอรี่ให้สามารถใช้ได้นานขึ้น โดยใช้บอร์ดที่ประหยัดพลังงาน
- จัดทำในส่วนของ Application โดยการพัฒนาให้สามารควบคุมผ่านอินเตอร์เน็ตได้ ปัจจุบันควบคุมได้เพียงจากWi-Fi เท่านั้น
- เพิ่มในส่วนของตัวตรวจวัดความชื้น (Sensor Node) เพื่อให้สามารถควบคุมระบบการให้น้ำให้มีประสิทธิภาพมากยิ่งขึ้น ปัจจุบันมีเพียง 1 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# เอกสารอ้างอิง


[1] ดนัย บุณยเกียรติ. ความสัมพันธ์ของน้ำและพืช. [ออนไลน์]. เข้าถึงได้จาก : http://web.agri.cmu.ac.th/hort/course/359311/PPHY2.htm (วันที่ค้นข้อมูล : 29 สิงหาคม 2557)

[2] บุญแสน เตียวนุกูลธรรม. ความชื้นของดิน. [ออนไลน์]. เข้าถึงได้จาก : http://elearning.nsru.ac.th/web_elearning/soil/lesson.php. (วันที่ค้นข้อมูล : 9 กันยายน 2557)

[3] การให้น้ำแบบฉีดฝอย. [ออนไลน์]. เข้าถึงได้จาก : http://www.natres.psu.ac.th/Department/PlantScience/510-111web/book/book%20content.htm/chapter09/agri_09.htm (วันที่ค้นข้อมูล : 9 กันยายน 2557)

[4] วินัย กล้าจริง. การออกแบบชลประทานแบบฉีดฝอยและแบบหยด สาขาวิชาวิศวกรรมเกษตร คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

[5] ไมโครคอนโทรลเลอร์. (Microcontroller). [ออนไลน์]. เข้าถึงได้จาก : http://www.ee.eng.cmu.ac.th/~tharadol/teach/ee490/490_44/project4/rp_mid.doc (วันที่ค้นข้อมูล : 31 สิงหาคม 2557)

[6] ไมโครคอนโทรลเลอร์. [ออนไลน์]. เข้าถึงได้จาก : http://th.wikipedia.org/wiki/ไมโครคอนโทรลเลอร์ (วันที่ค้นข้อมูล : 31 สิงหาคม 2557)

[7] Arduino คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : http://arduino-thai.blogspot.com/2013/05/arduino.html (วันที่ค้นข้อมูล : 30 สิงหาคม 2557)

[8] Raspberry Pi. [ออนไลน์]. เข้าถึงได้จาก : http://raspberry-pi-thai.blogspot.com/2013/07/raspberry-pi.html (วันที่ค้นข้อมูล : 9 กันยายน 2557)

[9] DTH22-AM2302. [ออนไลน์]. เข้าถึงได้จาก : http://www.arduitronics.com/product/177/dht22-am2302-high-accuracy-digital-temperature-and-humidity-sensor (วันที่ค้นข้อมูล : 30 ตุลาคม 2557)

[10] ธีรวัฒน์ ดิษฐ์ประสพ, พรนรินทร์ ตันกระหาด และ พฤติวัชร จันทร์เศรษฐี. 2554 การออกแบบระบบการให้น้ำพืชอัตโนมัติไร้สาย. ปริญญานิพนธ์วิศวกรรมศาสตร์บัณฑิต, สาขาวิชาวิศวกรรมเกษตร คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

[11] โมดูล nRF24L01. [ออนไลน์]. เข้าถึงได้จาก : http://thaisensormodule.com/index.php/data-rf/nrf24l01 (วันที่ค้นข้อมูล : 30 ตุลาคม 2557)

[12] Relay. [ออนไลน์]. เข้าถึงได้จาก : http://kpp.ac.th/elearning/elearning3/book-09.html (วันที่ค้นข้อมูล : 1 พฤศจิกายน 2557)

[13] โมดูล DS3231 RTC. [ออนไลน์]. เข้าถึงได้จาก : http://cpre.kmutnb.ac.th/esl/learning/index.php?article=ds3231-i2c-rtc (วันที่ค้นข้อมูล : 7 กันยายน 2557)

[14] PuTTy คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : http://www.msit.mut.ac.th/newweb/phpfile/show.php?Qid=6961 (วันที่ค้นข้อมูล : 6 กันยายน 2557)

[15] โปรแกรม WinSCP [ออนไลน์]. เข้าถึงได้จาก : http://software.thaiware.com/10606-WinSCP-Download.html (วันที่ค้นข้อมูล : 6 กันยายน 2557)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[16] Python คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : http://python.cmsthailand.com/basic_python.html (วันที่ค้นข้อมูล : 6 กันยายน 2557)

[17] สุธี พงศาสกุลชัย. (2553). การโปรแกรมภาษา C++. ใน กิตติ ภักดีวัฒนะกุล (บรรณาธิการ), ตัวแปลและชนิด ข้อมูล (หน้า 19-34). กรุงเทพฯ: เคทีพีคอมพ์ แอนด์ คอนซัลท์.

[18] ภาษาเอชทีเอ็มแอล. [ออนไลน์]. เข้าถึงได้จาก : http://th.wikipedia.org/wiki/เอชทีเอ็มแอล (วันที่ค้นข้อมูล : 6 กันยายน 2557)

[19] ภาษาจาวาสคริปต์. [ออนไลน์]. เข้าถึงได้จาก : http://th.wikipedia.org/wiki/จาวาสคริปต์ (วันที่ค้นข้อมูล : 9 กันยายน 2557)

[20] ภาษา SQLite. [ออนไลน์]. เข้าถึงได้จาก : http://www.thaicreate.com/mobile/android-sqlite-database.html (วันที่ค้นข้อมูล : 1 พฤศจิกายน 2557)

[21] Apache. [ออนไลน์]. เข้าถึงได้จาก : lecture.cs.buu.ac.th/~f53370/ass07/51535525/hw07.odt (วันที่ค้นข้อมูล : 1 พฤศจิกายน 2557)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก
# ตารางผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 ตารางบันทึกผลการทดลองการหาค่าความชื้นในดิน

| ครั้งที่ (ซ้ำ) | น้ำหนักกระป๋อง (g) | น้ำหนักตัวอย่างเปียก+น้ำหนักกระป๋อง(g) | น้ำหนักตัวอย่างเปียก (g) | น้ำหนักตัวอย่างแห้ง+น้ำหนักกระป๋อง(g) | น้ำหนักตัวอย่างแห้ง (g) | ปริมาณน้ำ (g) | ความชื้น (%) |
|---|---|---|---|---|---|---|---|
| 1 | 99.42 | 281.45 | 182.03 | 235.09 | 135.67 | 46.36 | 34.17 |
| 2 | 99.08 | 269.29 | 170.21 | 223.52 | 124.44 | 45.77 | 36.78 |
| 3 | 98.93 | 244.89 | 145.96 | 210.65 | 111.72 | 34.24 | 30.65 |
| 4 | 99.42 | 263.61 | 164.19 | 225.72 | 126.30 | 37.89 | 30.00 |
| 5 | 99.37 | 264.37 | 165.00 | 225.83 | 126.46 | 38.54 | 30.48 |
| 6 | 99.23 | 264.23 | 165.00 | 227.06 | 127.83 | 37.17 | 29.08 |
| เฉลี่ย | 99.24 | 264.64 | 165.40 | 224.65 | 125.40 | 39.99 | 31.86 |

ตารางที่ ก.2 ความชื้นในดินที่ความชื้นต่างๆเทียบกับแรงดันไฟฟ้า และAnalog to Digital Converters

| ADC | แรงดันไฟฟ้า(V) | ความชื้นในดิน(%db) |
|---|---|---|
| 100 | 0.49 | 4.64 |
| 125 | 0.61 | 5.32 |
| 143 | 0.70 | 5.38 |
| 182 | 0.89 | 6.98 |
| 195 | 0.95 | 7.51 |
| 217 | 1.06 | 8.42 |
| 238 | 1.16 | 9.23 |
| 269 | 1.31 | 10.72 |
| 287 | 1.40 | 11.47 |
| 310 | 1.52 | 12.55 |
| 328 | 1.60 | 13.67 |
| 358 | 1.75 | 14.23 |
| 389 | 1.90 | 14.91 |
| 418 | 2.04 | 15.62 |
| 434 | 2.12 | 16.72 |
| 470 | 2.30 | 17.45 |
| 498 | 2.43 | 18.65 |
| 527 | 2.58 | 19.89 |
| 545 | 2.66 | 21.01 |
| 569 | 2.78 | 22.93 |

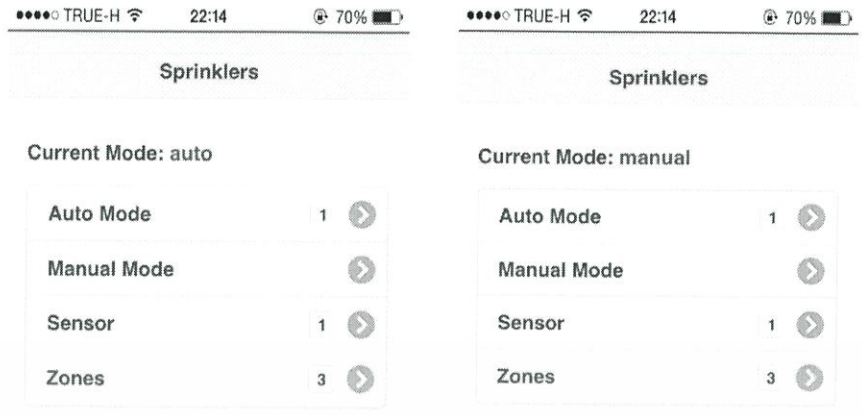
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ADC | แรงดันไฟฟ้า(V) | ความชื้นในดิน(%db) |
|---|---|---|
| 587 | 2.87 | 23.25 |
| 618 | 3.02 | 24.68 |
| 622 | 3.04 | 25.71 |
| 654 | 3.20 | 26.01 |
| 682 | 3.33 | 27.93 |
| 702 | 3.43 | 28.09 |
| 735 | 3.59 | 29.12 |
| 753 | 3.68 | 30.25 |
| 762 | 3.72 | 31.86 |
| 817 | 3.99 | 32.78 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

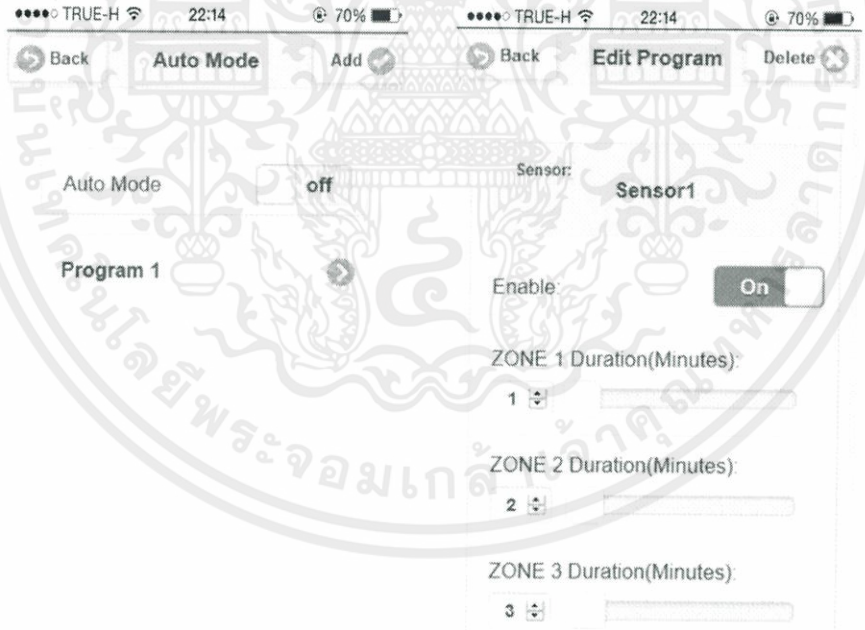
# ภาคผนวก ข
# แอปพลิเคชันควบคุมระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




(a) (b)

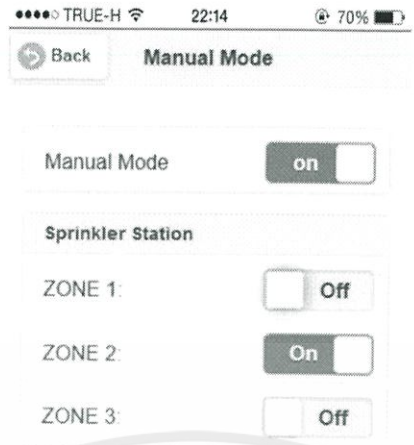
รูปที่ ข.1 หน้าแรกของเวปแอปพลิเคชันในโหมดต่างๆ (a) Auto Mode (b) Manual Mode



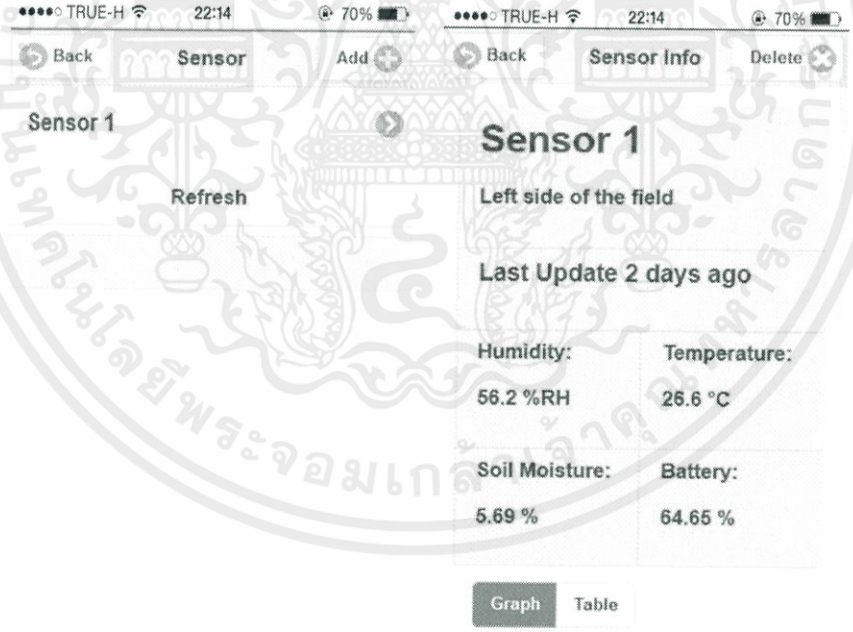

(a) (b)

รูปที่ ข.2 หน้าของเวปแอปพลิเคชันในของเมนู Auto Mode
(a) หน้าเปิด-ปิด การทำงาน Auto Mode (b) หน้าปรับแต่งแก้ไขโปรเกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



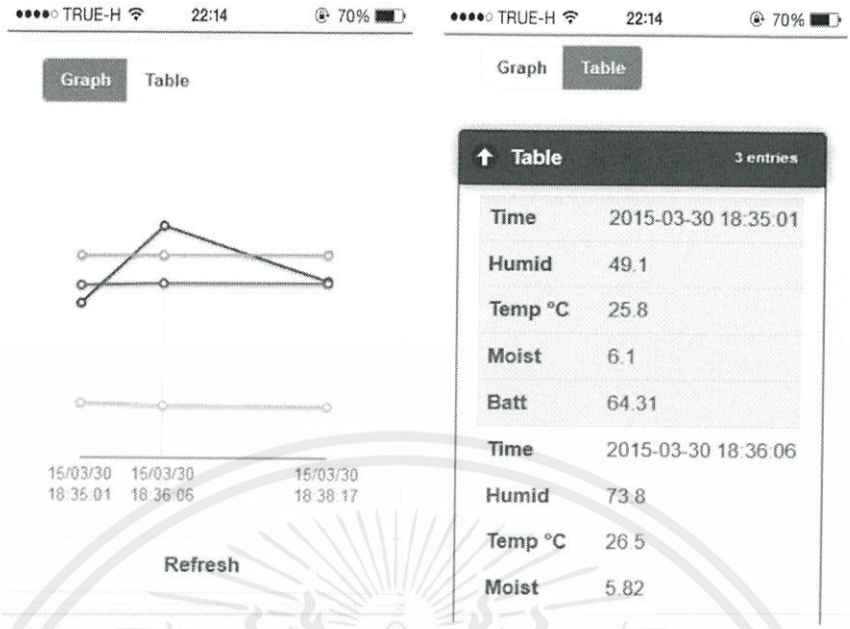

รูปที่ ข.3 หน้าของเวปแอปพลิเคชันในของเมนู Manual Mode แสดงการเปิด-ปิดรีเลย์




(a) (b)

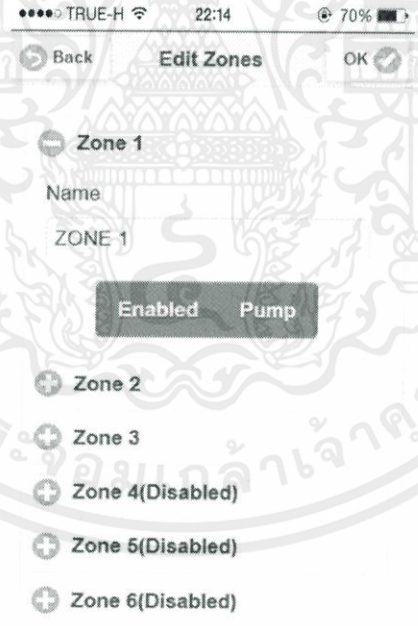
รูปที่ ข.4 หน้าของเวปแอปพลิเคชันในเมนู Sensor (a) แสดงจำนวน Sensor ที่ใช้งาน (b) แสดงข้อมูลอัพเดทล่าสุดของ Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




(a) (b)

รูปที่ ข.5 หน้าของเวปแอปพลิเคชันในเมนู Sensor (a) ข้อมูลย้อนหลังในรูปแบบกราฟ (b) ข้อมูลย้อนหลังในรูปแบบตาราง




รูปที่ ข.6 หน้าของเวปแอปพลิเคชันในเมนู Zones สามารถเพิ่ม-ลดการใช้งานในแต่ละโซนและกำหนดให้โซนนั้นๆทำงานพร้อมกับปั๊มหลักได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวค ค
# โค้ดโปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**1. โค้ดโปรแกรมส่วนตรวจวัดความชื้น**

```
*********************************************************************************
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include "printf.h"
#include <avr/power.h>
#include <avr/sleep.h>
#include "LowPower.h"

#define moisture_input 0 // Analog pin 0
#define divider_top 5
#define divider_bottom 4
int NodeID=1;
float Moistureavg;
int Smavg;
int State;
int need_water=2;
int Normal=1;
int Error=0;

// For using the temperature sensor DHT22
#include <dht.h>
dht DHT;
#define DHT22_PIN 2
struct
//nRF24 set the pin 10 to CE and 9 to CSN/SS
// Cables are:
//     SS      -> 9
//     MOSI    -> 11
//     MISO    -> 12
//     SCK     -> 13

RF24 radio(10,9);
//gain address to a write pipe
const uint64_t pipes[2] = { 0xF0F0F0F0E1LL,0xF0F0F0F0D2LL };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// here we can send up to 8 chars
char Send1[8];
char Send2[8];
char Send3[8];
char Send4[8];
char Send5[4];

int sleepStatus = 0;          // variable to store a request for sleep
int count = 450;               // counter
int sleepTIME=0;

//battery check
int VPin = 2;
int val = 0;
float pinVoltage = 0;
float batteryVoltage = 0;
float battery = 0;
float ratio = 2.316;

void setup(void) {
 Serial.begin(57600); //Debug
 printf_begin();
 //nRF24 configurations
 radio.begin();
 radio.setChannel(0x4c);
 radio.setAutoAck(1);
 radio.setRetries(15,15);
 radio.setDataRate(RF24_250KBPS);
 radio.setPayloadSize(32);
 radio.openWritingPipe(pipes[1]);
 radio.printDetails(); //for Debugging
 radio.powerDown();
 pinMode(DHT22_PIN, INPUT_PULLUP);
 pinMode(VPin, INPUT_PULLUP);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void loop() {
    //Check voltage battery
    digitalWrite(VPin, HIGH);
    for ( int i = 0 ; i < 3 ; i++ ) {
      delay(50);
       val = analogRead(VPin);              // read the voltage 3 times. keep last reading
     }
    pinVoltage = val * 0.00488;             // Calculate the voltage on the A/D pin
    batteryVoltage = pinVoltage * ratio;    // Use the ratio calculated for the voltage
divider
    battery = batteryVoltage*10;
    digitalWrite(VPin, LOW);

 //Get Soil moisture Voltage from soil probe
     int Moisture=0;
     for ( int i = 0 ; i < 3 ; i++ ) {
      Moisture += moistCHK();
     }
     Smavg = (Moisture/3);
     float a=Smavg*0.004888;
     Moistureavg =  (0.3494*(pow(2,a)))+(6.719*a)+1.0114;
     if(Moistureavg <= 4) {
      State=Error;
   }else if(Moistureavg > 35){
       State=Error;
   }else if(Moistureavg > 4 && Moistureavg < 20){
       State=need_water;
   }else{
      State=Normal;
   }
     digitalWrite(DHT22_PIN, HIGH);
     delay (38);
     //uint32_t start = micros();
     int chk = DHT.read22(DHT22_PIN);
     //uint32_t stop = micros();
     float RH,Temp;
     RH=DHT.humidity;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Temp=DHT.temperature;
    delay (18);
    digitalWrite(DHT22_PIN, LOW);

// Prepare Reading Values for Transmission
    dtostrf(battery,2,2,Send1);
    dtostrf(RH,2,2,Send2);
    dtostrf(Temp,2,2,Send3);
    dtostrf(Moistureavg,4,2,Send4);
    dtostrf(State,1,0,Send5);

//add a str
    String str;
    str += String(NodeID);
    str += ":";
    str += String(Send1);
    str += ":";
    str += String(Send2);
    str += ":";
    str += String(Send3);
    str += ":";
    str += String(Send4);
    str += ":";
    str += String(Send5);
    byte Size = str.length();
    Size = Size + 1;
    char SendPayload[Size];
    str.toCharArray(SendPayload,Size); // convert string to char array for transmission
    sleep_disable();
    radio.powerUp();
    delay(50);
    bool ok = radio.write(&SendPayload,strlen(SendPayload));
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); // sleep mode nrf
    sleep_enable();
    radio.powerDown();
    delay(50);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  // sleep mode
   for ( int sleepTIME = 0; sleepTIME < count; sleepTIME++ ) {
     LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);  //sleep duration is 8
seconds multiply by the sleep cycle variable.
   }
}
int moistCHK(){
 int moistREADdelay = 88;
 int moistAvg = 0;
 // set driver pins to outputs
 pinMode(divider_top,OUTPUT);
 pinMode(divider_bottom,OUTPUT);

 // drive a current through the divider in one direction
 digitalWrite(divider_top,HIGH);
 digitalWrite(divider_bottom,LOW);

 // wait a moment for capacitance effects to settle
 delay(moistREADdelay);

 // take a reading
 int moistVal1 =analogRead(moisture_input);
 // reverse the current
 digitalWrite(divider_top,LOW);
 digitalWrite(divider_bottom,HIGH);

 // give as much time in 'revers'e as in 'forward'
 delay(moistREADdelay);
 int moistVal2 = analogRead(moisture_input);
 digitalWrite(divider_bottom,LOW);
 moistVal2 = 1023 - moistVal2;
 moistAvg = (moistVal1 + moistVal2) / 2;
 return moistAvg;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โค้ดโปรแกรมส่วนรับผลและควบคุม

```
**************************************************************************
#!/usr/bin/python                                          // import ไลบราลี่ที่ใช้
ทั้งหมด
import threading
from Queue import Queue
import RPi.GPIO as GPIO
import sys
import sqlite3
from lib_nrf24 import NRF24
import time, datetime
import spidev
import logging
import PinControl as pin

GPIO.setmode(GPIO.BCM)                                     // ตั้งค่าเริ่มต้น GPIO
GPIO.setwarnings(False)
pin.Init()
pin.TurnOffChannels()

LOG_FILENAME = '/var/log/sprinkler/gateway.out.log'                    // เชื่อมต่อ
ไฟล์บันทึกการทำงาน
logging.basicConfig(filename=LOG_FILENAME,
                    level=logging.DEBUG,
                    format='%(asctime)s - %(levelname)s - %(message)s',
                    datefmt='%m/%d/%Y %I:%M:%S %p')

con = sqlite3.connect("//var//www//FlaskApp//FlaskApp//db//test.db",
check_same_thread=False)
// เชื่อมต่อฐานข้อมูล
q =Queue(maxsize=10)                                                   // ตั้งค่าตัวแปร
ในฟังชั่น Low moist
aflag = [True]
prev_nid = ''
prev_time = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pipes = [[0xf0, 0xf0, 0xf0, 0xf0, 0xe1], [0xf0, 0xf0, 0xf0, 0xf0, 0xd2]]  //ตั้งค่าการใช้งาน
nRF24l01
radio = NRF24(GPIO, spidev.SpiDev())
radio.begin(0, 25)
radio.setRetries(15,15)
radio.setPayloadSize(32)
radio.setChannel(0x4c)
radio.setDataRate(NRF24.BR_250KBPS)
radio.setPALevel(NRF24.PA_MAX)
radio.setAutoAck(True)
radio.openWritingPipe(pipes[0])
radio.openReadingPipe(1, pipes[1])
radio.startListening()
radio.stopListening()
radio.printDetails()
radio.startListening()

def Low_moist(flag=[False]):                                    // ฟังชั่นการจ่ายน้ำ
while True:
        if flag[0]:
            try:
                if not q.empty():
                    totalswork_time = 0
                    with con:
                        cur = con.cursor()
                        cur.execute("SELECT Stationlist, Enable FROM Program_Index WHERE
SensorNameID = '{}'".format(q.get()));
                        programs = [dict(Slist=row[0], enable=row[1]) for row in cur.fetchall()]
                        #print programs
                        check_ena = iter(programs)
                        for pro in check_ena:
                            #print pro['enable']
                            logging.debug(pro['enable'])
                            if pro['enable'] == 'on':
                                #do some work
                                cur.execute("SELECT Station, Duration FROM {} ORDER BY Station
ASC".format(pro['Slist']));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
                works = [dict(S=row[0], du=row[1]) for row in cur.fetchall()]
                worklist = iter(works)
                for work in worklist:
                    cur.execute("SELECT Mode FROM Mode_Status");
                    mode = cur.fetchone()
                    logging.debug(mode[0])
                    if (mode[0] == "auto") and (work['du'] != 0):
                        cur.execute("SELECT Name FROM Station_Table WHERE ID
='{}'".format(work['S']));
                        sname  = cur.fetchone()
                        worktime = work['du'] * 60
                        cur.execute("UPDATE Current_On_Station SET DateTime =
datetime(CURRENT_TIMESTAMP, 'localtime'), Station = '{}', Duration =
'{}'".format(sname[0], worktime));
                        con.commit()
                        pin.TurnOnChannel(work['S'].replace("s",""))
                        logging.debug("{} ON!!".format(work['S']))
                        while worktime > 0:
                            cur.execute("SELECT Mode FROM Mode_Status");
                            mode = cur.fetchone()
                            if mode[0] != "auto":
                                logging.debug("TURN OFF!!")
                                worktime = 0
                            else:
                                time.sleep(4)
                                worktime -= 4
                        pin.TurnOffChannels()
            cur.execute("UPDATE Current_On_Station SET DateTime =
datetime(CURRENT_TIMESTAMP, 'localtime'), Station = '{}', Duration =
'{}'".format('None','None'));
                con.commit()
            q.task_done()
        time.sleep(2)

    except sqlite3.Error, e:
        if con:
            con.rollback()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            logging.debug("Error {}:".format(e.args[0]))
            sys.exit(1)
            con.close()

def Sensorcheck(SensorID):                                    //ฟังชันตรวจสอบไอดี
เซ็นเซอร์
    global idcheck
    idcheck = -1
    for x in SensorMem:
        idcheck = idcheck + 1
        if SensorID == x[0]:
            return True

def loadSensors():                                            //ฟังชันตรวจสอบไอดีเซ็นเซอร์
จากฐานข้อมูล
    global SensorMem
    with con:
        cur = con.cursor()
        cur.execute("SELECT SensorID, Name FROM Sensor_Index");
        SensorMem = cur.fetchall()

t = threading.Thread(target=Low_moist, args=(aflag,))         // เริ่มต้น loop การ
ทำงานของ Low moist
t.setDaemon(True)
t.start()

loadSensors()                                                 //ตรวจสอบไอดีเซ็นเซอร์จาก
ฐานข้อมูล

try:
    while True:
        pipe = [0]
        time.sleep(1)
        logging.debug('Wait for response')
        #print 'Wait for response'
        while not radio.available(pipe):                      // รอรับข้อมูลจากเซ็นเซอร์
            time.sleep(1000/1000000.0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        recv_buffer = []
        radio.read(recv_buffer)
        #logging.debug("retrived: {}".format(recv_buffer))
        #print "retrived: {}".format(recv_buffer)
        message = ''.join(chr(i) for i in recv_buffer)
        #print message
        Sensordata = message.strip()
        #print Sensordata
        Sensordata = Sensordata.split(':')
        #print Sensordata
        Sensordata = [x.strip('\x00') for x in Sensordata]
        logging.debug("Data are: {}".format(Sensordata))
        #print "Data are: {}".format(Sensordata)
        SensorID = Sensordata[0]
        SensorID = int(SensorID)

        #if sensorID doesnt exist!!
        if not Sensorcheck(SensorID):                    // เทียบความเหมือน
ของไอดี
            #print "ID not Match!!"
            logging.debug("ID not Match!! {}".format(SensorID))
            loadSensors()

        if Sensorcheck(SensorID):
            #print "Found ID Match!!"
            logging.debug("Found ID Match!! {}".format(SensorID))
            with con:
                cur = con.cursor()
                cur.execute("UPDATE Sensor_Index SET Enable='on' WHERE
SensorID='{}'".format(SensorID));
                con.commit()

            logging.debug(SensorMem[idcheck][1])
            Name = SensorMem[idcheck][1]
            logging.debug("Found Name Match!! {}".format(Name))
            NameID = Name + str(SensorID)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
            logging.debug("Found NameID Match!! {}".format(NameID))
            NameID = NameID.strip()
            logging.debug("Found NameID Match!! {}".format(NameID))
            NameID = NameID.replace(" ","")
            logging.debug("Found NameID Match!! {}".format(NameID))
            Batt = Sensordata[1]
            Humid = Sensordata[2]
            TempC = Sensordata[3]
            Moist = Sensordata[4]
            sen_status = Sensordata[5]
            logging.debug("Found NameID Match!! {}".format(NameID))


            with con:
                cur = con.cursor()
                cur.execute("SELECT name FROM sqlite_master WHERE type='table' AND
name='{}'".format(NameID));
                tableResult = cur.fetchone()

            if not tableResult:                                    // สร้างตารางบันทึกหากยังไม่มี
                logging.debug("No match tableName Create new one")
                with con:
                    cur = con.cursor()
                    cur.execute("CREATE TABLE {}(DateTime DATE DEFAULT
(datetime('now','localtime')),Batt INT, Humid INT, TempC INT, Moist
INT)".format(NameID));
                    cur.execute("INSERT INTO {}(DateTime, Batt, Humid, TempC, Moist)
VALUES (datetime(CURRENT_TIMESTAMP, 'localtime'), 0, 0, 0, 0)".format(NameID));
                    con.commit()

            with con:                                              // ตรวจสอบสถานะความ
ต้องการน้ำจากเซ็นเซอร์
                logging.debug("Found match tableName {}".format(tableResult))
                cur = con.cursor()
                if sen_status == '2':
                    cur.execute("SELECT Mode FROM Mode_Status");
                    mode = cur.fetchone()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if mode[0] == "auto":
                    logging.debug(prev_nid)
                    logging.debug(NameID)
                    logging.debug(time.mktime(datetime.datetime.now().timetuple()) -
prev_time)
                    if ((prev_nid == NameID) and
(time.mktime(datetime.datetime.now().timetuple()) - prev_time < 30)):
                        logging.debug("Prev id = name id and time drifferent < 1 Sensor
Error")
                    else:
                        logging.debug("Auto mode")
                        logging.debug(NameID)
                        q.put(NameID)
                        prev_nid = NameID
                        prev_time = (time.mktime(datetime.datetime.now().timetuple()))
                else:
                    logging.debug("another mode")
            elif sen_status == '0':
                #print "Moisture below 100 or Higher than 800"
                logging.debug("Moisture below 100 or Higher than 800")
            else:
                #print "Your plant fine"
                logging.debug("Your plant fine")
            cur.execute("INSERT INTO {} (DateTime, Batt, Humid, TempC,  Moist) VALUES
(datetime(CURRENT_TIMESTAMP, 'localtime'), {}, {}, {},
{})".format(NameID,Humid,Batt,TempC,Moist));
            con.commit()                                                    // บันทึกข้อมูล

except sqlite3.Error, e:
    if con:
        con.rollback()
        logging.debug("Error {}:".format(e.args[0]))
        sys.exit(1)
        con.close()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โค้ดโปรแกรมฟังชันส่วนควาบคุม

```
**********************************************************************************
import wiringpi2 as wiring                                          // import ไลบราลี่ที่ใช้
import sqlite3
from time import sleep

con = sqlite3.connect("//var//www//FlaskApp//FlaskApp//db//test.db",
check_same_thread=False)
// เชื่อมต่อกับฐานข้อมูล
MAX_CHANNEL = 7                                                     // ประกาศตัวแปร
PinDict = {
        'R0': 21,
        'R1': 20,
        'R2': 16,
        'R3': 26,
        'R4': 19,
        'R5': 13,
        'R6': 6,
        'R7': 5
}

def Init():                                                 // ฟังชันเริ่มต้นการใช้งาน GPIO
        wiring.wiringPiSetupSys()
        for key, value in PinDict.iteritems():
                wiring.pinMode(value,1)

def CheckPump(id):                                  // ฟังชันตรวจสอบการทำงานร่วมกับปั้มหลัก
        Name = 's' + str(id)
        with con:
                cur = con.cursor()
                cur.execute("SELECT Pump FROM Station_Table WHERE ID =
'{}'".format(Name));
                pump = cur.fetchone()
        return pump

def MainChannel(state):                                     // ฟังชันเปิดใช้งานปั้มหลัก
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if state == 'on':
        wiring.digitalWrite(PinDict['R0'], 1)

def TurnOffChannels():                                  // ฟังชันปิดทุกโซน
    for key, value in PinDict.iteritems():
        wiring.digitalWrite(value, 0)


def TurnOnChannel(idChan):                              // ฟังชันเปิดโซน
    num = int(idChan)
    if num <= 0 or num > MAX_CHANNEL:
        return
    TurnOffChannels()
    for key, value in PinDict.iteritems():
        if int(key[1:]) == num:
            Pump = CheckPump(num)
            wiring.digitalWrite(value, 1)
            MainChannel(Pump[0])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โค้ดโปรแกรมส่วนเซิฟเวอร์

```
**********************************************************************************
import os
from flask import Flask, render_template, g, jsonify, request
import sqlite3
import threading, datetime, time, random
import PinControl

app = Flask(__name__)

PinControl.Init()

app.database = "//var//www//FlaskApp//FlaskApp//db//test.db"

def Date_To_Sec(date):
        dateType = datetime.datetime.strptime(date, '%Y-%m-%d %H:%M:%S')
        sec_rec = time.mktime(dateType.timetuple())
        return sec_rec

timenow_sec = time.mktime(datetime.datetime.now().timetuple())

def connect_db():
        return sqlite3.connect(app.database)

@app.route("/")
def index():
        title = "index"
        return render_template("index.htm", title=title)

@app.route("/auto")
def auto():
        return render_template("auto.htm")

@app.route("/stations")
def page_zone():
        return render_template("stations.htm")
```

เอกสารนี้สงวนสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
@app.route("/manual")
def page_man():
        title = "Manual"
        return render_template("manual.htm", title=title)

@app.route("/addsen")
def Show_Add_Sensor():
        return render_template("addsen.htm")

@app.route("/sen")
def list_sensors():
        return render_template("sen.htm")

@app.route("/shsen")
def Show_sensor():
        return render_template("showsen.htm")

@app.route("/addpro")
def Add_pro():
        ID = request.args.get('id')
        if ID == None:
                return render_template("addpro.htm", title='Add Program')
        else:
                return render_template("addpro.htm", title='Edit Program')

@app.route("/json/index", methods=["GET"])
def Show_stuff():
        g.db = connect_db()
        cur = g.db.execute("SELECT COUNT(*) FROM Program_Index");
        pro_count = cur.fetchone()
        cur = g.db.execute("SELECT COUNT(*) FROM Sensor_Index");
        sen_count = cur.fetchone()
        cur = g.db.execute("SELECT COUNT(Enable) FROM Station_Table WHERE Enable
= 'on'");
        s_count = cur.fetchone()
        cur = g.db.execute("SELECT Mode FROM Mode_Status");
        cur_mode = cur.fetchone()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
        cur = g.db.execute("SELECT * FROM Current_On_Station");
        current = cur.fetchone()
        On_time = current[0]
        On_station = current[1]
        On_duration = current[2]
        if On_duration == 'None':
                return jsonify(pro=pro_count[0], sen=sen_count[0], stations=s_count[0],
mode=cur_mode[0])
        elif On_duration == 99999:
                return jsonify(pro=pro_count[0], sen=sen_count[0], stations=s_count[0],
mode=cur_mode[0], duration=On_duration, onstation=On_station)
        else:
                time_left = float(On_duration) -
(time.mktime(datetime.datetime.now().timetuple()) - Date_To_Sec(On_time))
                return jsonify(pro=pro_count[0], sen=sen_count[0], stations=s_count[0],
mode=cur_mode[0], duration=time_left, onstation=On_station)

@app.route("/json/mode", methods=["GET"])
def mode_status():
        g.db = connect_db()
        cur = g.db.execute("SELECT * FROM Mode_Status")
        data = cur.fetchone()
        return jsonify(mode=data[0], state=data[1])

@app.route("/json/slist", methods=["GET"])
def send_slist():
        g.db = connect_db()
        cur = g.db.execute("SELECT SensorID, Name, Enable FROM Sensor_Index")
        data = [dict(id=row[0], name=row[1], e=row[2]) for row in cur.fetchall()]
        return jsonify(list=data)

@app.route("/json/plist", methods=["GET"])
def send_plist():
        g.db = connect_db()
        cur = g.db.execute("SELECT ProID, Enable FROM Program_Index")
        data = [dict(id=row[0], e=row[1]) for row in cur.fetchall()]
        return jsonify(list=data)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
@app.route("/json/pro", methods=["GET"])
def send_pro():
        ID = request.args.get('id')
        g.db = connect_db()
        cur = g.db.execute("SELECT SensorNameID, Enable, StationList FROM
Program_Index WHERE ProID='{}'".format(ID));
        result = cur.fetchone()
        cur = g.db.execute("SELECT * FROM '{}' ORDER BY Station ASC".format(result[2]));
        data = [dict(stations=row[0], name=row[1], duration=row[2]) for row in
cur.fetchall()]
        return jsonify(sname=result[0], enable=result[1], stations=data)

@app.route("/json/next_chk", methods=["GET"])
def check():
        ID = request.args.get('id')
        g.db = connect_db()
        cur = g.db.execute("SELECT Name FROM Sensor_Index WHERE
SensorID='{}'".format(ID));
        result = cur.fetchone()
        nameID = result[0] + ID
        nameID = nameID.replace(" ","")
        cur = g.db.execute("SELECT DateTime FROM '{}' ORDER BY rowid DESC LIMIT
1;".format(nameID));
        date = cur.fetchone()
        last_chk = Date_To_Sec(date[0])
        return jsonify (time=last_chk)

@app.route("/json/sensor", methods=["GET"])
def send_status():
        ID = request.args.get('id')
        g.db = connect_db()
        cur = g.db.execute("SELECT Desc, Name FROM Sensor_Index WHERE
SensorID='{}'".format(ID));
        result = cur.fetchone()
        desc = result[0]
        name = result[1]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        nameID = name + ID
        nameID = nameID.replace(" ","")
        cur = g.db.execute("SELECT * FROM '{}' ORDER BY rowid DESC LIMIT
1;".format(nameID));
        data = [dict(date=row[0], humid=row[1], batt=row[2], temp=row[3],
moist=row[4]) for row in cur.fetchall()]
        return jsonify(name=name, desc=desc, data=data)

@app.route("/json/table", methods=["GET"])
def send_table():
        ID = request.args.get('id')
        g.db = connect_db()
        cur = g.db.execute("SELECT Name FROM Sensor_Index WHERE
SensorID='{}'".format(ID));
        result2 = cur.fetchone()
        name = result2[0]
        nameID = name + ID
        nameID = nameID.replace(" ","")
        cur = g.db.execute("SELECT * FROM '{}' WHERE DateTime BETWEEN
datetime('now', '-7 day') AND datetime('now', 'localtime')".format(nameID));
        table = [dict(date=row[0], humid=row[1], batt=row[2], temp=row[3],
moist=row[4]) for row in cur.fetchall()]
        return jsonify(entries=table)

@app.route("/json/station")
def station():
        g.db = connect_db()
        cur = g.db.execute("SELECT * FROM Station_Table")
        table = [dict(ID=row[0], Name=row[1], Enabled=row[2], Pump=row[3],
State=row[4]) \
        for row in cur.fetchall()]
        g.db.close()
        return jsonify(station=table)

@app.route("/bin/mode", methods=["GET"])
def Set_Mode():
        mode = request.args.items()
```

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mode = dict(mode)
        CurrentMode = 'None'
        SetMode – ''
        ModeState = ''
        g.db = connect_db()
        for key, value in mode.iteritems():
                if key == 'mode':
                        SetMode = value
                elif key == 'state':
                        ModeState = value
        if SetMode == 'manual' and ModeState == 'on':
                PinControl.TurnOffChannels()
                cur = g.db.execute("UPDATE Station_Table SET State ='off'");
                cur = g.db.execute("UPDATE Mode_Status SET Mode = '{}', State =
'{}'".format(SetMode,ModeState));
        elif SetMode == 'auto' and ModeState == 'on':
                PinControl.TurnOffChannels()
                cur = g.db.execute("UPDATE Station_Table SET State ='off'");
                cur = g.db.execute("UPDATE Mode_Status SET Mode = '{}', State =
'{}'".format(SetMode,ModeState));
                cur = g.db.execute("UPDATE Current_On_Station SET Station = 'None',
Duration = 'None'");
        else:
                PinControl.TurnOffChannels()
                cur = g.db.execute("UPDATE Station_Table SET State ='off'");
                cur = g.db.execute("UPDATE Mode_Status SET Mode = 'None', State =
'None'");
                cur = g.db.execute("UPDATE Current_On_Station SET Station = 'None',
Duration = 'None'");
        g.db.commit()
        g.db.close()
        return ''

@app.route("/bin/addsen", methods=["GET"])
def Add_Sensor():
        New_sensor = request.args.items()
        New_sensor = dict(New_sensor)
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        g.db = connect_db()
        cur = g.db.execute("INSERT INTO Sensor_Index (SensorID, Name, Desc, Enable)
VALUES
('{}','{}','{}','{}')".format(New_sensor['id'],New_sensor['name'],New_sensor['descript'],New_se
nsor['enable']));
        g.db.commit()
        g.db.close()
        return ''

@app.route("/bin/addAuto", methods=["GET"])
def Add_Auto():
        New_Program = request.args.items()
        New_Program = dict(New_Program)
        g.db = connect_db()
        if 'namesen' in New_Program:
                names = New_Program['namesen'].replace(" ","")
                table_name = 'S' + names
                cur = g.db.execute("CREATE TABLE {} (Station TEXT, StationName TEXT,
Duration INT)".format(table_name));
                for key, value in New_Program.iteritems():
                        if key[0:1] == 's':
                                cur = g.db.execute("SELECT Name FROM Station_Table
WHERE ID = '{}'".format(key));
                                name = cur.fetchone()
                                cur = g.db.execute("INSERT INTO '{}' (Station, StationName,
Duration) VALUES ('{}','{}','{}')".format(table_name, key, name[0], value));
                cur = g.db.execute("INSERT INTO Program_Index (SensorNameID, Enable,
StationList) VALUES ('{}','{}','{}')".format(names,New_Program['enable'],table_name));
        else:
                for key, value in New_Program.iteritems():
                        if key[0:1] == 's':
                                ename = New_Program['ename'].replace(" ","")
                                cur = g.db.execute("UPDATE '{}' SET Duration = '{}' WHERE
Station = '{}'".format(ename, value, key));
                cur = g.db.execute("UPDATE Program_Index SET Enable = '{}' WHERE
StationList = '{}'".format(New_Program['enable'], ename));
        g.db.commit()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        g.db.close()
        return ''

@app.route("/bin/DelSen", methods=["GET"])
def Del_Sensor():
        SID = request.args.get('id')
        g.db = connect_db()
        cur = g.db.execute("SELECT Name FROM Sensor_Index WHERE SensorID =
'{}'".format(SID));
        name = cur.fetchone()
        Sensor_table = name[0] + SID
        cur = g.db.execute("DELETE FROM Sensor_Index WHERE SensorID =
'{}'".format(SID));
        cur = g.db.execute("DROP TABLE IF EXISTS '{}'".format(Sensor_table));
        g.db.commit()
        g.db.close()
        return ''

@app.route("/bin/DelAuto", methods=["GET"])
def Del_Pro():
        PID = request.args.get('id')
        g.db = connect_db()
        cur = g.db.execute("SELECT SensorNameID FROM Program_Index WHERE ProID =
'{}'".format(PID));
        name = cur.fetchone()
        Pro_table = 'S'+ name[0]
        cur = g.db.execute("DELETE FROM Program_Index WHERE ProID =
'{}'".format(PID));
        cur = g.db.execute("DROP TABLE IF EXISTS '{}'".format(Pro_table));
        g.db.commit()
        g.db.close()
        return ''

@app.route("/bin/station", methods=["GET"])
def set_station():
        New_station = request.args.items()
        New_station = dict(New_station)
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        g.db = connect_db()
        for key, value in New_station.iteritems():
                ID = key[0:2]
                if key[2:] == 'Name':
                        cur = g.db.execute("UPDATE Station_Table SET Name='{}' WHERE
ID='{}'".format(value,ID));
                elif key[2:] == 'e':
                        if value == 'on':
                                cur = g.db.execute("UPDATE Station_Table SET Enable='{}'
WHERE ID='{}'".format(value,ID));
                        elif value == 'off':
                                cur = g.db.execute("UPDATE Station_Table SET Enable='{}'
WHERE ID='{}'".format(value,ID));
                elif key[2:] == 'p':
                        if value == 'on':
                                cur = g.db.execute("UPDATE Station_Table SET Pump='{}'
WHERE ID='{}'".format(value,ID));
                        elif value == 'off' :
                                cur = g.db.execute("UPDATE Station_Table SET Pump='{}'
WHERE ID='{}'".format(value,ID));
        g.db.commit()
        g.db.close()
        return ''


@app.route("/bin/manual", methods=["GET"])
def ManualChannel():
        data = request.args.items()
        data = dict(data)
        g.db = connect_db()
        for key, value in data.iteritems():
                if key == 'station':
                        Channel = value[1:]
                elif key == 'state':
                        State = value
        Id = 's' + (Channel)
        if State == 'on':
                PinControl.TurnOnChannel(Channel)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                cur = g.db.execute("UPDATE Station_Table SET State ='off'");
                cur = g.db.execute("UPDATE Station_Table SET State ='on' WHERE ID
='{}'".format(Id));
                cur = g.db.execute("SELECT Name FROM Station_Table WHERE ID =
'{}'".format(Id));
                name = cur.fetchone()
                cur = g.db.execute("UPDATE Current_On_Station SET Station = '{}',
Duration = 99999".format(name[0]));
        else:
                PinControl.TurnOffChannels()
                cur = g.db.execute("UPDATE Station_Table SET State ='off'");
                cur = g.db.execute("UPDATE Current_On_Station SET Station = 'None',
Duration = 'None'");
        g.db.commit()
        g.db.close()
        return ''

if __name__ == '__main__':
        PinControl.Init()
        app.run()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้