ระบบไฮโดรโปนิกส์แบบอัจฉริยะ
Intelligent Hydroponics System

อรรฆพร อู่อุดมสุขทรัพย์
Akaporn U-udomsuksap
อัษฎาวุธ ฐานวิสัย
Adsadawut Thanwisai

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2560

ระบบไฮโดรโปนิกส์แบบอัจฉริยะ

Intelligent Hydroponics System

โดย

นายอรรฆพร อู่อุดมสุขทรัพย์

นายอัษฎาวุธ ฐานวิสัย

อาจารย์ที่ปรึกษา

ผศ.ดร.ยุทธนา คิดใจเดียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบไฮโดรโปนิกส์แบบอัจฉริยะ

Intelligent Hydroponics System

ผู้จัดทำ นายอรรฆพร อู่อุดมสุขทรัพย์ 57011495

นายอัษฎาวุธ ฐานวิสัย 57011532

รายงานนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ผศ.ดร.ยุทธนา คิดใจเดียว)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

Adstract II

กิตติกรรมประกาศ III

คำนำ IV

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตในการจัดทำโครงงาน 2

1.4 ระยะเวลาในการจัดทำรายงาน 2

บทที่ 2 แนวคิดทางทฤษฎี

2.1 ไฮโดรโปนิกส์ 4

2.2 การควบคุมค่า pH ของน้ำ 4

2.3 การควบคุมค่า EC ของน้ำ 5

2.4 อุณหภูมิและออกซิเจน 7

2.5 เซ็นเซอร์วัดความชื้นสัมพัทธ์และอุณหภูมิ (DHT22) 7

2.6 ความเข้มแสง 8

2.7 WIFI 8

2.8 ระบบ NFT 9

2.6 อุปกรณ์และวงจร 9

2.6.1 Analog pH Meter (pH Sensor) 9

2.6.2 Digital Temperature Sensor 10

2.6.3 Solenoid Valve 10

2.6.4 1-Channel Relay Module 11

2.6.5 Switching Power Supply 12

2.6.6 Arduino Mega 2560 12

2.6.7 TFT LCD Module Display 13

2.6.8 Float Switch Sensor 13

2.6.9) LED Grow Light 14

2.6.10) Photoresistor LDR Light Sensor Module (LDR) 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.6.11) DHT22 Temperature and Humidity Sensor Module 15
2.6.12) ESP8266 15
2.6.13) พัดลมระบายความร้อน 16

บทที่ 3 หลักการทำงานและการออกแบบ

3.1 Block Diagram 17
3.2 การออกแบบ 19
3.2.1 Design Hydroponic System 19
3.2.2 Schematic 19
3.3 การทำงานของวงจร 23
3.3.1 Arduino Mega2560 23
3.3.2 Arduino Uno 26
3.3.3 ESP8266 26
3.4 Flow chart 28
3.4.1 แผนผังการทำงานของ Pump Pressure 28
3.4.2 แผนผังการปรับค่า pH ของน้ำ 29
3.4.3 แผนผังการควบคุม Water Level 30
3.4.4 แผนผังการควบคุมหลอดไฟ 30

บทที่ 4 การทดลองและผลการทดลอง

4.1 การทดลอง 32
4.1.1 การทดลองสั่งการผ่านหน้าจอ LCD Touch Screen 32
4.1.2 การทดลองระบบเปิด-ปิดไฟอัตโนมัติตามความเข้มแสง 33
4.1.3 การทดลองสั่งการผ่านสมาร์ทโฟน 33
4.2 ผลการทดลอง 34
4.2.1 ผลการทดลองการสั่งการผ่านหน้าจอ LCD Touch Screen 34
4.2.2 ผลการทดลองระบบเปิด-ปิดไฟอัตโนมัติตามความเข้มแสง 36
4.2.3 ผลการทดลองสั่งการผ่านสมาร์ทโฟน 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลการทดลอง

5.1 สรุปผลการทดลอง..... 37

5.2 อุปสรรคและปัญหา..... 37

เอกสารอ้างอิง..... 38

ภาคผนวก

ภาคผนวก ก. การทดลองและโครงสร้างของระบบไฮโดรโปนิกส์..... 39

ภาคผนวก ข. โค้ดบนโปรแกรม Arduino..... 49

ภาคผนวก ค. Datasheet..... 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

4.1 แสดงการเจริญเติบโตของพืชในระบบไฮโดรโปนิกส์ในระบบอัตโนมัติ..... 34

4.2 แสดงการเปรียบเทียบค่า pH ระหว่าง pH Meter เทียบกับ Analog pH Sensor..... 35

4.3 แสดงความสัมพันธ์ระหว่างค่า pH ของสารชนิดต่างๆ กับ Voltage..... 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

รูปที่ 2.1 ระบบ NFT (Nutrient Film Technique : NFT) 9

รูปที่ 2.2 Analog pH Meter (pH Sensor) 9

รูปที่ 2.3 Digital Temperature Sensor 10

รูปที่ 2.4 Solenoid Valve 10

รูปที่ 2.5 Relay Module 11

รูปที่ 2.6 Switching Power Supply 12

รูปที่ 2.7 Arduino Mega 2560 12

รูปที่ 2.8 TFT LCD module display 13

รูปที่ 2.9 Float Switch Sensor 13

รูปที่ 2.10 LED Grow Light 14

รูปที่ 2.11 Photoresistor LDR Light Sensor Module (LDR) 14

รูปที่ 2.12 DHT22DHT22 15

รูปที่ 2.13 ESP8266 15

รูปที่ 2.14 พัดลมระบายความร้อน 16

รูปที่ 3.1 Block Diagram ของระบบไฮโดรโปนิกส์ 17

รูปที่ 3.2 โครงสร้างของระบบไฮโดรโปนิกส์ 19

รูปที่ 3.3 วงจรรวม 19

รูปที่ 3.4 วงจรควบคุมอุณหภูมิ 20

รูปที่ 3.5 วงจรควบคุมหลอดไฟ ผ่าน Arduino Mega2560 20

รูปที่ 3.6 วงจรควบคุมหลอดไฟ ผ่าน Arduino Uno 21

รูปที่ 3.7 วงจรควบคุมค่าความเป็นกรด-เบส 21

รูปที่ 3.8 วงจรควบคุมระดับน้ำ 22

รูปที่ 3.9 วงจรสำหรับควบคุมและแสดงผลผ่านสมาร์ทโฟน 22

รูปที่ 3.10 ขั้นตอนการทำงานของระบบ 23

รูปที่ 3.11 จอ LCD แสดงหน้าต่างในการเลือกปลูกผัก 24

รูปที่ 3.12 จอ LCD แสดงหน้าต่างค่าอุณหภูมิ ค่าพีเอช และค่าระดับน้ำในถังสารละลาย 25

รูปที่ 3.13 จอ LCD แสดงหน้าต่างระบบแมนนวล 25

รูปที่ 3.14 จอ LCD แสดงหน้าต่างระบบคาริเบท 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.15 การแสดงผลและควบคุมผ่านสมาร์ทโฟน..... 27
รูปที่ 3.16 แผนผังการควบคุมการทำงานของ Pressure Pump..... 28
รูปที่ 3.17 ค่าอุณหภูมิที่ส่งผลต่อปริมาณออกซิเจน..... 28
รูปที่ 3.18 แผนผังการควบคุมค่าpHของน้ำ..... 29
รูปที่ 3.19 แสดงค่า pH ที่พืชแต่ละชนิดต้องการ..... 29
รูปที่ 3.20 แผนผังควบคุม Water Level..... 30
รูปที่ 3.21 แผนผังควบคุม LED ผ่าน Arduino Mega2560..... 30
รูปที่ 3.22 แผนผังควบคุม LED ผ่าน Arduino Uno..... 31
รูปที่ 4.1 กราฟเปรียบเทียบค่า pH ที่ได้จาก pH Meter เทียบกับ Analog pH Sensor..... 35
รูปที่ 4.2 กราฟแสดงความเป็นเชิงเส้นของค่า pH เทียบกับ Voltage..... 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน ระบบไฮโดรโปนิกส์แบบอัจฉริยะ

นักศึกษา นายอรรฆพร อู่อุดมสุขทรัพย์ รหัสประจำตัว 57011495

นายอัษฎาวุธ ฐานวิสัย รหัสประจำตัว 57011532

ปริญญา วิศวกรรมศาสตรบัณฑิต

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

ปีการศึกษา 2560

อาจารย์ที่ปรึกษาโครงการ ผศ.ดร.ยุทธนา คิดใจเดียว

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ เพื่อออกแบบระบบไฮโดรโปนิกส์ ซึ่งเป็นระบบปลูกพืชโดยไม่ใช้ดิน โดยควบคุมการจ่ายน้ำและสารอาหารแบบอัตโนมัติ เป็นระบบที่ทำหน้าที่ประมวลผลค่า pH ค่าอุณหภูมิของน้ำ และระดับน้ำจากเซนเซอร์ โดยไมโครคอนโทรลเลอร์ (Arduino MEGA 2560) ซึ่งการควบคุมค่าพีเอชจะใช้โซลินอยด์วาล์วในการปล่อยกรดและเบสสำหรับควบคุมค่าความเป็นกรดเบส ให้อยู่ในช่วงที่ต้องการ มีการใช้สเปรย์หมอกในการลดอุณหภูมิ เมื่อค่าอุณหภูมิมีค่าสูงเกินช่วงที่กำหนด และควบคุมระดับน้ำจากเซนเซอร์ลูกลอยไฟฟ้า เมื่อพืชดูดน้ำไปใช้ทำให้น้ำในถังสารอาหารลดลง จะมีปั๊มดูดน้ำจากอีกถังเข้ามาเติมให้ระดับน้ำเท่าเดิม อีกทั้งมีการให้แสงที่มีช่วงแสงใกล้เคียงกับแสงในธรรมชาติมาช่วยในการเจริญเติบโตของพืช ในกรณีที่พื้นที่ในการเพาะปลูกนั้นได้รับแสงจากธรรมชาติไม่เพียงพอ มีการแสดงค่าผ่านจอทัชสกรีน (TFT) ซึ่งสามารถเลือกแบบอัตโนมัติ หรือแบบปรับค่าด้วยมือได้ โดยในแบบอัตโนมัตินั้น เราออกแบบให้สามารถเลือกปลูกพืชได้ 3 ชนิด คือ 1. ผักสลัด 2. กะหล่ำปลี 3. โรสแมรี่ ส่วนในแบบปรับค่าด้วยมือ เราสามารถปรับกรดหรือเบส และสามารถควบคุมสเปรย์หมอกได้เอง ในทั้งสองแบบจะแสดงค่าความเป็นกรดเบส และอุณหภูมิในขณะนั้นๆ อีกทั้งยังสามารถปรับกรดและเบส สั่งการเปิดและปิดพัดลมระบายความร้อน รวมไปถึงการแสดงค่าความชื้น, อุณหภูมิ และค่า pH ในแอพพลิเคชั่น Blynk ผ่านหน้าจอสมาร์ทโฟนได้อีกด้วย ระบบไฮโดรโปนิกส์แบบอัจฉริยะนี้เป็นการจำลองการปลูกพืชไร้ดิน เพื่อนำไปประยุกต์ใช้ในทางเกษตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project title	Intelligent Hydroponics System	
Student	Mr.Akaporn U-udomsuksap	Student ID 57011495
	Mr.Adsadawut Thanwisai	Student ID 57011532
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2017	
Project Advisor	Asst.Prof.Dr.Yuttana Kidjaidure	

ABSTRACT

This project is a microcontroller application designed for automatically control water contribution, amount of mineral in water and temperature to Hydroponics system. The system processes data from a pH sensor, temperature of water and water level. These data are sent to the microcontroller board (Arduino MEGA 2560). Then, the microcontroller will control solenoid valves, pressure pump and light for the plant growth. LCD touchscreen provides the status of the temperature, pH value and water level. In addition, the system can operate in two modes. In automatic mode, we design to have three kinds of plants, Salad, Cabbage and Rosemary. The manual mode allows the user to adjust pH values and temperature for the specific plants. Moreover, user can access intelligent hydroponics via smart phone by using Blynk application. Blynk will display humidity, temperature and pH value. It can also command solenoid valves and ventilation fan too. This intelligent hydroponics system is the model of the plant growth without soil utilized in agriculture.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการจัดทำโครงงานเรื่องระบบไฮโดรโปนิกส์แบบอัจฉริยะ (Intelligent Hydroponics System) นี้เสร็จสมบูรณ์ลงในส่วนหนึ่งได้นั้นกลุ่มของข้าพเจ้าขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ.ดร.ยุทธนา คิดใจเดียว ที่คอยชี้แนะให้ความรู้เกี่ยวกับวงจร ให้คำแนะนำและคำปรึกษาเกี่ยวกับข้องบกพร่องและแก้ไขปัญหาต่างๆ รวมถึงสถานที่และอุปกรณ์ที่ใช้ในการทำโครงงาน อีกทั้งคณะอาจารย์ทุกท่านที่เกี่ยวข้องในการให้ความรู้แก่คณะผู้จัดทำแล้วนำความรู้ที่ได้นำมาประยุกต์ใช้ในการทำโครงงาน ขอขอบพระคุณบิดามารดาที่ช่วยเหลือในเรื่องค่าใช้จ่ายต่างๆ อีกทั้งยังคอยสนับสนุนและเป็นกำลังใจให้เสมอ รวมถึงพี่ๆ เพื่อนๆ วิศวกรรมอิเล็กทรอนิกส์ที่ให้ยืมอุปกรณ์ รวมถึงให้คำแนะนำ

นายอรรฆพร อู่อุดมสุขทรัพย์

นายอัษฎาวุธ ฐานวิสัย

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

ปริญญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อประกอบการทำโครงงาน โดยเนื้อหาจะกล่าวถึงทฤษฎี ขั้นตอนในการออกแบบระบบปลูกผักไฮโดรโปนิกส์อัตโนมัติ โดยประกอบไปด้วยส่วนของ Hardware คือ โครงปลูกผักในระบบ NFT และวงจรควบคุมปัจจัยในการปลูกพืชอัตโนมัติได้แก่ การควบคุมอุณหภูมิ, ค่าความเป็นกรด-เบสของน้ำ, ระดับน้ำ และหลอดไฟ LED ที่ใช้สำหรับปลูกพืช ในส่วนของ Software ใช้โปรแกรม Arduino ในการรับค่าข้อมูลจากเซนเซอร์ เพื่อแสดงค่า pH, Temperature และระดับน้ำ ผ่านหน้าจอ TFT LCD Module อีกทั้งยังสามารถปรับกรดและเบส สั่งการเปิดและปิดพัดลมระบายความร้อน รวมไปถึงการแสดงค่าความชื้น, อุณหภูมิ และค่า pH ในแอพพลิเคชั่น Blynk ผ่านหน้าจอสมาร์ทโฟนได้อีกด้วย สุดท้ายนี้ ผู้จัดทำหวังเป็นอย่างยิ่งว่าปริญญานิพนธ์ฉบับนี้จะเป็นประโยชน์แก่ผู้ที่มีความสนใจอย่างสูงสุด

นายอรรฆพร อู่อุดมสุขทรัพย์

นายอัษฎาวุธ ฐานวิสัย

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ไฮโดรโปนิกส์ (Hydroponics) เป็นการปลูกพืชโดยไม่ใช้ดิน นับเป็นวิธีการใหม่ในการปลูกพืชโดยเฉพาะการปลูกผักและพืชที่ใช้เป็นอาหาร เนื่องจากประหยัดพื้นที่ และไม่ปนเปื้อนกับสารเคมีต่างๆ ในดิน ทำให้ได้พืชผักที่สะอาดเป็น ไฮโดรโปนิกส์ เป็นทางเลือกหนึ่งให้ผลผลิตที่มีคุณภาพ มีความสม่ำเสมอของผลผลิตสูง สามารถวางแผน กำหนดปริมาณการผลิตให้เป็นตามเป้าหมาย หรือความต้องการของตลาดได้ดีกว่า เนื่องจากสามารถควบคุมปัจจัยต่างๆ ที่ให้แก่พืชได้อย่างเหมาะสม ทำให้ปลูกพืชได้อย่างต่อเนื่อง สำหรับประเทศไทยมีความจำเป็นไปได้ที่จะนำวิธีการปลูกพืชแบบนี้มาใช้ทำให้สามารถปลูกพืชในพื้นที่ไม่มีดินหรือดินมีปัญหา บริเวณรอบๆ เมืองใหญ่ มีคนอาศัยแน่นหนา เพราะการปลูกแบบนี้จะให้พื้นที่น้อยแต่ให้ผลผลิตมาก จะเลือกปลูกพืชชนิดใดต้องศึกษาความต้องการของตลาดก่อน ความต้องการผลผลิตเพื่อใช้บริโภคภายในประเทศ หรือต้องการผลิตเพื่อส่งออกไปขายที่ต่างประเทศ จะผลิตพืชผัก ไม้ดอกหรือไม้ประดับ ควรจะเป็นพืชที่มีราคาแพงหรือให้ผลตอบแทนสูงถ้าทำเป็นการค้า เมื่อเลือกชนิดพืชที่มีตลาดรองรับแน่นอนแล้ว จึงเลือกระบบการปลูกพืชที่เหมาะสม ควรเป็นระบบที่ง่าย ไม่ยุ่งยาก วัสดุอุปกรณ์หาได้ภายในประเทศเพื่อลดต้นทุน

อย่างไรก็ตามไม่มีระบบไหนจะสมบูรณ์ที่สุด ยังต้องมีพัฒนาการต่อไป เพื่อให้ได้ระบบการปลูกที่เหมาะสมกับพืชแต่ละชนิดในสภาพภูมิอากาศของบ้านเรา ทั้งนี้ต้องไม่ลืมว่าหัวใจสำคัญของการปลูกพืชเป็นการค้า คือ ต้องผู้ซื้อหรือมีตลาดที่แน่นอน ผู้ซื้อและผู้ขายต้องมีความซื่อสัตย์ มีคุณธรรม ไม่เอาเปรียบกันในเรื่องราคาผลผลิต ถ้าทำได้เช่นนี้ทุกฝ่ายก็จะได้รับผลประโยชน์จากการใช้เทคโนโลยีนี้ร่วมกันสืบไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 ศึกษาแนวทางในการปลูกพืชไร้ดิน

1.2.2 เพื่อลดปัญหาเกี่ยวกับข้อจำกัดเรื่องพื้นที่ในการทำเกษตร และลดการใช้สารเคมีในพืช

1.2.3 ศึกษาการทำงานของ pH Sensor, Temperature Sensor, Humidity Sensor, LDR Light Intensity Sensor และ Float Switch Sensor เพื่อส่งข้อมูลให้ Arduino Mega2560, Arduino UNO และ ESP8266

1.2.4 เรียนรู้การใช้ Arduino Mega2560, Arduino UNO และ ESP 8266 เป็น Microcontroller เพื่อควบคุมระบบการทำงานของ Hydroponic System

1.2.5 ศึกษาการเขียนโค้ด เพื่อควบคุมการทำงานของระบบไฮโดรโปนิกส์แบบอัจฉริยะ

1.2.6 ทดลองปลูกพืชและบันทึกผลการเจริญเติบโต

1.3 ขอบเขตในการจัดทำโครงงาน

ศึกษาการปลูกพืชแบบไร้ดิน (ระบบไฮโดรโปนิกส์) รวมถึงปัจจัยที่มีผลต่อการควบคุมการเจริญเติบโตของพืช ได้แก่ ค่า pH ในน้ำ, ค่าอุณหภูมิในน้ำที่เหมาะสม, ค่าความชื้นที่เหมาะสมกับการเจริญเติบโตของพืช, ความเข้มแสงที่พืชต้องการ และการรักษาระดับน้ำในถังสารอาหาร เพื่อทำการควบคุมปัจจัยเหล่านี้โดยทำเป็นระบบอัตโนมัติที่ใช้ Arduino Mega2560, Arduino UNO และ ESP8266 ในการควบคุมการทำงานของระบบเพื่อปรับค่า pH, อุณหภูมิ, ความชื้น, ความเข้มแสงและระดับน้ำ ให้ได้ค่าที่เหมาะสมต่อการเจริญเติบโตของพืชแล้วทำการทดลองปลูกพืชและเก็บผลการเจริญเติบโตของพืช

1.4 ระยะเวลาในการจัดทำ

ตั้งแต่วันที่ 7 สิงหาคม 2560 ถึง 23 เมษายน 2561

กิจกรรม	ปี 2560			
	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน
1. กำหนดหัวข้อโครงงาน	■			
2. หาข้อมูลของโครงงาน	■			
3. วางแผนและออกแบบการทำงาน		■		
4. จัดเตรียมอุปกรณ์ในการทำงาน		■		
5. ทำโครงงาน			■	
6. ทดสอบและปรับปรุงโครงงาน				■
7. บันทึกผลการทดลองและจัดทำรายงาน				■

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรม	ปี 2561			
	มกราคม	กุมภาพันธ์	มีนาคม	เมษายน
1. ทำโครงสร้างของอุปกรณ์				
2. แก้ไขและพัฒนาระบบให้ดีขึ้น				
3. นำอุปกรณ์ทั้งหมดลงกล่องคอนโทรลเลอร์และติดตั้งอุปกรณ์				
4. ทดสอบการทำงานของระบบ				
5. ปลูกพืชและเก็บผลการเจริญเติบโต				
6. บันทึกผลและจัดทำรายงาน				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

แนวคิดทางทฤษฎี

2.1 ไฮโดรโปนิกส์

ไฮโดรโปนิกส์ (hydroponics) เป็นคำที่มาจากภาษากรีก 2 คำ คือคำว่า hydro ซึ่งแปลว่า น้ำ และคำว่า ponos แปลว่าทำงานหรือแรงงาน เมื่อรวมกันจึงมีความหมายว่าการทำงานที่เกี่ยวข้องกับน้ำ ประวัติความเป็นมาของการปลูกพืชโดยวิธีนี้นั้นเริ่มมาจากการศึกษาเกี่ยวกับการใช้ธาตุอาหารต่างๆ ในการปลูกพืช ซึ่งมีมาตั้งแต่หลายพันปีก่อนสมัยของอริสโตเติล จากหลักฐานทางประวัติศาสตร์พบว่านักวิทยาศาสตร์หลายท่านได้เขียนบันทึกต่างๆ ทางพฤกษศาสตร์ขึ้นและปรากฏอยู่จนทุกวันนี้ แต่การปลูกพืชตามหลักการทางวิทยาศาสตร์นั้นเริ่มขึ้นประมาณ 300 ปีมาแล้ว คือประมาณ ค.ศ. 1699 John Woodward นักพฤกษศาสตร์ชาวอังกฤษได้พยายามทำการทดลอง เพื่อหาคำตอบว่าอนุภาคของของแข็งและของเหลวที่อยู่ในดินมีความสำคัญต่อการเจริญเติบโตของพืชอย่างไร ต่อมาปี ค.ศ. 1860-1865 นักวิทยาศาสตร์ชื่อ Sachs และ Knop นับเป็นผู้ริเริ่มปลูกพืชด้วยวิธีไฮโดรโปนิกส์ตามหลักการทางวิทยาศาสตร์สมัยใหม่ โดยการปลูกพืชด้วยสารละลายเกลือ อนินทรีย์ต่างๆ เช่น โพแทสเซี่ยมฟอสเฟต โพแทสเซี่ยมไนเตรต ซึ่งให้ธาตุอาหารที่จำเป็นต่อการเจริญเติบโตของพืช คือ ไนโตรเจน ฟอสฟอรัส โพแทสเซี่ยม แมกนีเซียม กำมะถัน แคลเซียม และเหล็ก ภายหลังมีการพัฒนาสูตรธาตุอาหารพืชเรื่อยมา จนถึงปี ค.ศ. 1920-1930 William F.Gericke แห่งมหาวิทยาลัยแคลิฟอร์เนีย ประสบความสำเร็จในการปลูกมะเขือเทศในสารละลายธาตุอาหาร โดยพืชมีการเจริญเติบโตสมบูรณ์และให้ผลผลิตเร็ว นับเป็นจุดเริ่มต้นของการนำเทคนิคการปลูกพืชโดยวิธีนี้ไปประยุกต์ใช้เพื่อปลูกพืชเป็นการค้า และได้มีการพัฒนาเทคนิควิธีการและส่วนประกอบในสารละลายเรื่อยมาจนถึงปัจจุบัน

การปลูกพืชด้วยวิธีไฮโดรโปนิกส์ คือ การปลูกพืชในน้ำที่มีธาตุอาหารพืชละลายอยู่ หรือการปลูกพืชในสารละลายธาตุอาหาร ทดแทนการปลูกพืชผลการเกษตรทั่วไป ซึ่งจำเป็นต้องมีการควบคุมค่าเหนี่ยวนำกระแสไฟฟ้าในของเหลว (EC) และค่า pH ของสารละลายเพื่อให้พืชสามารถใช้ประโยชน์จากธาตุอาหารได้อย่างครบถ้วนในทุกฤดูกาล

2.2 การควบคุมค่า pH ของน้ำ

ค่า pH ในความหมายของการปลูกพืชไร้ดิน คือ ค่าความเป็นกรด-เบส ของสารละลาย (น้ำผสมธาตุอาหารที่ใช้ในการปลูกพืช) โดยค่า pH จะอยู่ช่วงการวัดอยู่ที่ 1 - 14 โดยจะนับค่าที่ 7 เป็นกลาง กล่าวคือ หากวัดค่าได้ต่ำกว่า 7 แสดงว่าของเหลวเป็นกรด หากวัดได้สูงกว่า 7 ขึ้นไปแสดงว่าเป็นเบส สำหรับการปลูกพืชด้วยน้ำนั้นค่า pH มีส่วนสำคัญเป็นอย่างมาก สำหรับการทำปฏิกิริยาทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคมีกับธาตุอาหารที่ใช้เลี้ยงพืช โดยธรรมชาติน้ำที่มีความเป็นกรดจะทำให้ธาตุอาหารพืชละลายตัวได้ดี และพืชสามารถดูดซึมไปใช้งานได้อย่างสะดวก แต่ถ้าหากน้ำที่ใช้ผสมธาตุอาหารพืชมีความเป็นเบสสูง จะทำให้ธาตุอาหารพืชตกตะกอนจนพืชไม่สามารถดูดซึมไปใช้งานได้ เพราะฉะนั้นจึงควรค่อยปรับ pH ด้วยความระวัง และค่อยปรับลดลง อย่าปรับค่า pH ให้ต่ำกว่า 4 จะทำให้รากพืชได้รับอันตรายจากการกัดกร่อนของกรด จนทำให้รากพืชอ่อนแอ และเชื้อโรคเข้าทำลายได้ง่ายขึ้น ค่า pH ที่ต่ำต่ำเกินไปยังส่งผลให้ความเข้มข้นของธาตุเหล็กในระบบปลูกมีสูงขึ้นถ้าธาตุเหล็กในระบบปลูกมีมากเกินไปจะเป็นพิษกับพืชได้ ในทางกลับกันถ้าปล่อยให้ค่า pH สูงเกินกว่า 7 เป็นระยะเวลาติดต่อกัน 2 – 3 วัน จะส่งผลต่อการดูดซึมธาตุอาหารพืช เช่น ฟอสฟอรัส,เหล็ก, แมงกานีส

การลดค่า pH นิยมใช้ กรดไนตริก (Nitric Acid) มีสูตรทางเคมี คือ HNO_3 ซึ่งกรดชนิดนี้เมื่อผสมกับน้ำจะแตกตัวเป็นอนุมูลย่อย เป็นกรดไนโตรเจน ซึ่งเป็นธาตุอาหารหลักของพืช และกรดที่นิยมใช้ อีกชนิดหนึ่ง คือ กรดฟอสฟอริก (Phosphoric Acid) ซึ่งมีสูตรทางเคมี คือ H_3PO_4 ซึ่งกรดชนิดนี้เมื่อผสมกับน้ำจะแตกตัวเป็นอนุมูลย่อย เป็นฟอสฟอรัส ซึ่งเป็นธาตุอาหารหลักของพืชเช่นกัน การใช้กรดทั้งสองชนิดนี้จึงมีผลพลอยได้จากการปรับลดค่า pH แล้วยังได้ธาตุอาหารพืชเพิ่มขึ้นมาในระบบอีกด้วย การเพิ่มค่า pH นิยมใช้ โพแทสเซียมคาร์บอเนต (Potassium Carbonate) หรือ โพแทสเซียมไฮดรอกไซด์ (Potassium Hydroxide) ซึ่งเคมีทั้ง 2 ชนิดนี้เมื่อผสมกับน้ำจะแตกตัวเป็นอนุมูลย่อยได้ โพแทสเซียม ซึ่งเป็นธาตุอาหารหลักของพืชเช่นกัน

ดังนั้น การปรับค่า pH ผู้ปลูกจะต้องปรับให้อยู่ในระดับที่เหมาะสมกับอายุการปลูกและชนิดของพืชนั้นๆ ด้วย โดยปกติค่า pH ที่ใช้ในการปลูกพืชจะมีค่าอยู่ในช่วง 5.5 - 7.0 แต่ที่ดีที่สุดต่อการละลายตัวธาตุอาหารพืชจะอยู่ที่ 5.8 – 6.3

2.3 การควบคุมค่า EC ของน้ำ

ค่า EC คือ ค่าเหนี่ยวนำกระแสไฟฟ้าในของเหลว ในการปลูกไฮโดรโปนิกส์ หมายถึง ปริมาณแร่ธาตุที่ละลายอยู่ในของเหลว โดยปกติน้ำบริสุทธิ์จะมีค่านำไฟฟ้าต่ำหรือมีค่าเป็นศูนย์ แต่เมื่อมีการเติมสารละลายต่างๆ ลงในน้ำนั้นจะทำให้ค่าสารละลาย หรือค่านำกระแสไฟฟ้าในน้ำนั้นๆ สูงขึ้นด้วย พืชแต่ละชนิดจะมีความต้านทานต่อค่า EC หรือ ความเข้มข้นของธาตุอาหารพืชที่ไม่เท่ากัน ทั้งนี้ขึ้นอยู่กับสายพันธุ์, อายุของพืช และสภาพแวดล้อมในการปลูกขณะนั้นด้วย หากเราให้ค่า EC ไม่เหมาะสมกับพืชแล้วจะทำให้พืชนั้นเติบโตไม่เป็นปกติหรือขาดความสมบูรณ์ได้

ปัจจัยที่เป็นตัวกำหนดค่า EC คือ

1. ชนิดและสายพันธุ์ กล่าวคือ พืชต้องการอาศัยการคายน้ำทางใบเพื่อให้เกิดแรงดันที่รากพืชเพื่อให้น้ำที่ผสมธาตุอาหารซึมผ่านจากรากไปยังส่วนต่างๆ ของพืชได้ หากค่า EC สูงกว่าค่ามาตรฐาน ของพืชชนิดนั้นๆพืชจะไม่สามารถนำพาน้ำที่มีธาตุอาหารไปยังส่วนต่างๆของพืชได้ส่งผลให้พืชเจริญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เติบโตได้ไม่ดีและเกิดขาดธาตุอาหารต่างๆได้

2. อายุของพืช กล่าวคือ พืชในช่วงแต่ละอายุจะมี การใช้ธาตุอาหารไม่เท่ากัน โดยจะแบ่งออกเป็น 3 ช่วงของการเจริญเติบโต ดังนี้

2.1 ช่วงต้นกล้า : ช่วงสัปดาห์แรกของการเจริญเติบโต เมื่อพืชงอกออกจากเมล็ดพืชจะใช้พลังงานและอาหารจากใบเลี้ยงเดี่ยวเป็นหลัก ทำให้การกำหนดค่า EC ในช่วงสัปดาห์แรกนี้จะอยู่ที่ประมาณ 30 – 50 % ของค่า EC ในพืชชนิดนั้นๆ และจะเพิ่มขึ้นเรื่อยๆ ในสัปดาห์ต่อไป

2.2 ช่วงเจริญเติบโต : ช่วงสัปดาห์ที่ 3 เป็นต้นไป ช่วงนี้เป็นช่วงที่พืชต้องกรใช้พลังงานและธาตุอาหารสูงมาก เพื่อใช้ในการสร้างส่วนต่างๆ ของใบ, ลำต้น, ดอก โดยจะใช้ธาตุอาหารประมาณ 80 – 100% ของค่า EC ในพืชชนิดนั้นๆ

2.3 ช่วงขยายพันธุ์ : เป็นช่วงที่พืชผ่านการเจริญเติบโตอย่างเต็มที่มาแล้วพืชได้ทำการสะสมอาหารและพลังงานมาได้อย่างเต็มที่แล้วพืชจะเริ่มใช้ธาตุอาหารใหม่น้อยลง โดยเฉลี่ยจะอยู่ที่ประมาณ 50-70%ของค่าECในพืชชนิดนั้นๆ

3. สภาพอากาศและฤดูกาล หากช่วงดังกล่าวมีปัจจัยที่ทำให้พืชต้องคายน้ำสูง เช่น แสงแดดจัด, อากาศ ร้อน พืชจำเป็นต้องดูดซึมน้ำมากเพื่อนำมาชดเชยน้ำที่สูญเสียไป หากมีการใช้ค่า EC ที่สูงในช่วงเวลาดังกล่าวแล้วพืชจะนำน้ำไปชดเชยน้ำที่สูญเสียไปได้ลำบากเราจึงเห็นพืชเหี่ยวเฉาในช่วงเวลาที่อากาศร้อนและแสงแดดจัด ดังนั้นช่วงเวลาที่อากาศร้อนมากๆ และแสงแดดแรงเกินไปเราต้องปรับค่า EC ลง พร้อมกับการกดกิจกรรมการคายน้ำของพืชลง เช่น พรางแสง, สเปรย์น้ำ หรือการใช้พัดลมระบายอากาศ เพื่อลดอุณหภูมิลงน้ำที่เหมาะสมที่สุดสำหรับนำมาใช้ปลูกผักไฮโดรโปนิกส์ได้แก่ น้ำฝน, น้ำประปาส่วนภูมิภาคและประปานครหลวง ฯลฯ เนื่องจากมีค่า EC ต่ำและเป็นแหล่งน้ำที่ประหยัด ส่วนน้ำที่ไม่แนะนำมาใช้ในการปลูก เช่น น้ำบาดาล เนื่องจากส่วนใหญ่น้ำบาดาลจะมีค่า EC สูง แล้วยังมีแคลเซียมคาบอเนต (หินปูน) สาเหตุของความกระด้างในน้ำ ทำให้ปุ๋ยตกตะกอนได้ง่าย หากไม่สามารถหาน้ำได้จากแหล่งดังกล่าวจริงอาจจะต้องมีการบำบัดด้วยวิธีกรองเพื่อลดค่าสารละลายในน้ำลงก่อนเพื่อให้มีค่า EC อยู่ ในระดับที่เหมาะสมที่จะนำมาปลูกพืชได้ โดยวิธีการกรองต้องใช้เครื่องกรองที่สามารถของกรองสารละลายในน้ำได้ เช่น ระบบกรอง Reverse Osmosis (R.O.) หรือการกรองด้วยระบบกรอง Softener ด้วยสารกรอง Resin เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 อุณหภูมิและออกซิเจน

พืชมีความจำเป็นใช้ก๊าซคาร์บอนไดออกไซด์(CO_2)ที่มีอยู่ประมาณ 0.033 เปอร์เซ็นต์ในบรรยากาศในการผลิตกลูโคส ($C_6H_{12}O_6$) ซึ่งเป็นสารอินทรีย์เริ่มต้น เหตุการณ์ที่พืชจะขาดคาร์บอนไดออกไซด์ เป็นไปได้ยากเนื่องจากมีแหล่งคาร์บอนไดออกไซด์อย่างเหลือเฟือ เช่น การเผาไหม้เชื้อเพลิงจากโรงงานและรถยนต์ตลอดจนการผลิตไฟฟ้า เป็นต้น ส่วนก๊าซออกซิเจน (O_2) พืชต้องการเพื่อใช้ในกระบวนการหายใจ(Respiration) เพื่อเปลี่ยนพลังงานแสงอาทิตย์ซึ่งถูกเก็บไว้ในรูปพลังงานเคมีในรูปของน้ำตาลกลูโคส และสามารถให้เป็นพลังงานเพื่อใช้ในการขับเคลื่อนกระบวนการเมตาโบลิซึม (Metabolism) ต่างๆ การหายใจของส่วนเหนือดินของพืชมักไม่มีปัญหา เพราะในบรรยากาศมีออกซิเจนเป็นองค์ประกอบอยู่ถึง 20 เปอร์เซ็นต์ สำหรับรากพืชมักขาดออกซิเจน โดยเฉพาะปลูกพืชไร้ดินด้วยเทคนิคการปลูกด้วยสารละลาย (Water Culture or Liquid Culture) จำเป็นต้องให้ออกซิเจนในจำนวนที่เพียงพอต่อความต้องการของพืช การให้ออกซิเจนแก่รากพืชจะให้ในรูปแบบของฟองอากาศที่แทรกอยู่ในสารละลายธาตุอาหารพืช ซึ่งให้โดยใช้เครื่องสูบลม หรือการใช้ระบบน้ำหมุนเวียน

ปริมาณออกซิเจนที่ละลายในน้ำ หรือดีโอ (Dissolved Oxygen : DO) ปริมาณ DO ในน้ำจะมีได้มากหรือน้อยเพียงใดขึ้นกับ 3 ปัจจัย ได้แก่ อุณหภูมิ, ค่า salinity และความดันบรรยากาศ
1. ปริมาณ DO จะเพิ่มขึ้นเมื่ออุณหภูมิลดลง (น้ำเย็นจะยอมให้ออกซิเจนละลายมากกว่าน้ำร้อน)
2. ปริมาณ DO จะเพิ่มขึ้นเมื่อค่า salinity ลดลง (น้ำจืดจะยอมให้ออกซิเจนละลายได้มากกว่าน้ำเค็ม)
3. ปริมาณ DO จะลดลงเมื่อค่าความดันลดลง (เมื่ออยู่ในระดับที่สูงขึ้นออกซิเจนจะละลายได้น้อยลง)

2.5 เซ็นเซอร์วัดความชื้นสัมพัทธ์และอุณหภูมิ (DHT22)

เซนเซอร์วัดความชื้นสัมพัทธ์ที่สามารถวัดได้ในช่วง 0 ถึง 100% เปอร์เซ็นความชื้นสัมพัทธ์ และ -40 ถึง 80 องศาเซลเซียส โดยมีความคลาดเคลื่อนอยู่ที่ +-2 ถึง+-5 เปอร์เซ็นความชื้นสัมพัทธ์ และ +-0.5 องศาเซลเซียส ตามลำดับ โดยมีหลักการทำงานดังนี้

เมื่อ Micro Controller Unit (MCU) ให้สัญญาณเริ่มการทำงาน DHT22 จะเปลี่ยนจากสภาวะเตรียมพร้อมเป็นเริ่มทำงาน โดยจะส่งข้อมูลอุณหภูมิและความชื้นกลับไปยัง MCU ในรูปแบบสัญญาณอนาลอก เมื่อส่งเสร็จแล้ว DHT22 จะกลับสู่สภาวะเตรียมพร้อมอีกครั้งเพื่อรอสัญญาณเริ่มการทำงานจาก MCU

การปลูกพืชไร้ดินในระบบไฮโดรโปนิกส์ แบบ NFT ในโรงเรือนเพาะปลูกระบบปิด จะต้องควบคุมอุณหภูมิ 25-35 °c ความชื้นสัมพัทธ์ 60-80 % อุณหภูมิภายในรางปลูก 18-30 °c (ดิเรก ทองอร่าม, 2560; นภดล เรียบเลิศหิรัญ, 2553; โสระยา ร่วมรังสี, 2544; อานัฐ ตันโช, 2552) การควบคุมดังกล่าวมีวัตถุประสงค์เพื่อให้ได้ผลผลิตที่คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ความเข้มแสง

แสงอาทิตย์ จะประกอบด้วย Color Spectrum ทุกสี เหมือนตอนเราฉายแสงผ่านปริซึม หรือตอนเกิดรุ้งบนท้องฟ้า โดยแสงแต่ละสีก็จะมีความยาวคลื่นที่แตกต่างกันโดยความยาวคลื่นแสงที่จำเป็น กับการเจริญเติบโตของต้นไม้จริงๆแล้วมีเพียงแสงสีแดงกับแสงสีน้ำเงินเท่านั้น ส่วนสีเหลือง ส้ม เขียวจะไม่จำเป็นเพราะจะโดนสะท้อนกลับหมด

แสงที่ใกล้เคียงแสงอาทิตย์มากที่สุดคือแสงจากหลอดไฟ Incandescent หรือหลอดไส้นั้นเอง แสงที่ผลิตจะเป็นแสงที่พืชต้องการ เน้นสี แดง(630-660 nm) และสีน้ำเงิน (430-460 nm) โดยมีความยาวคลื่นแสงที่พืชต้องการ(ค่าคลื่นแสงประมาณ 400-700 nm) จะเป็นช่วงที่พืชสามารถสังเคราะห์แสงได้ดีที่สุดและเจริญเติบโตดีที่สุด ช่วงแสง สีส้ม สีเหลือง และสีเขียวจะเป็นสีที่ไม่จำเป็นต่อการเจริญเติบโตของพืช หรือ LED สีแดง 80-85% กับ LED สีน้ำเงิน 15-20% รวมกันเป็นแสงที่มีความยาวคลื่นที่ต้นไม้ต้องการได้เช่นเดียวกับแสงอาทิตย์เหมาะจะใช้กับไม้ประดับที่ปลูกอยู่ในร่ม หรือการปลูกพืชแบบ Hydroponic ก็ได้

โดยธรรมชาติแล้ว ผักสลัดตระกูลสีแดง เช่น เรดโอ๊ค เรดคลอรัล จะต้องการความเข้มแสงอยู่ประมาณ 20,000 – 40,000 ลักซ์ ขณะที่ผักสลัดตระกูลสีเขียวสามารถเจริญเติบโตได้ดี แม้ว่าความเข้มแสงจะอยู่เกิน 50,000 – 70,000 ลักซ์ การที่ผักได้รับความเข้มแสงมากเกินไปหรือน้อยเกินไปจะทำให้เกิดผลเสียต่อการเจริญเติบโต กล่าวคือ ถ้าผักได้รับความเข้มแสงมากเกินไปจะทำให้ใบตก รากคล้ำ และเน่า แต่หากได้รับความเข้มแสงน้อยเกินไปจะทำให้ผักแคระ เพราะไม่สามารถสังเคราะห์แสงได้ และตายลงในที่สุด

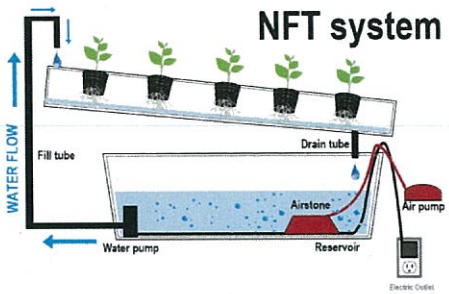
2.7 WIFI

Wi-Fi (Wireless Fidelity) หรือก็คือ Wireless LAN เป็นการสื่อสารด้วยระบบไร้สายบนเทคโนโลยี IEEE 802.11 โดยที่จะทำงานภายใต้คลื่นวิทยุ 2.4 GHz ซึ่งอุปกรณ์ทุกตัวซึ่งต่างยี่ห้อกันนั้นมันสามารถติดต่อสื่อสารกันได้โดยไม่มีปัญหา ภายใต้มาตรฐานเดียวกัน โดยจะมีการออกเป็น WIFI certified ซึ่งเป็นอันรู้กันว่า อุปกรณ์ชิ้นนั้นสามารถติดต่อสื่อสารกับอุปกรณ์ตัวอื่นที่มีตรา WIFI certified นี้ได้เช่นกัน ซึ่งในปัจจุบันเทคโนโลยีดังกล่าวได้ออกมา 3 ความเร็วด้วยกันคือ

1. 802.1a ทำงานด้วยความถี่ 5 GHz ที่อัตราความเร็วข้อมูล 54 Mbps (แต่ไม่นิยมใช้งานใน ประเทศไทย)
2. 802.1b ทำงานด้วยความถี่ 2.4 GHz ที่ความเร็ว 11 Mbps
3. 802.1g ทำงานด้วยความถี่ 2.4 GHz ที่ความเร็ว 54 Mbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ระบบ NFT (Nutrient Film Technique : NFT)



รูปที่ 2.1 ระบบ NFT (Nutrient Film Technique : NFT)

ระบบ NFT คือการปล่อยให้น้ำผสมธาตุอาหารพืชไหลลงไปในรางปลูก โดยสารละลายในรางปลูกจะมีความลึกประมาณ 0.5 cm การทำเช่นนี้จะช่วยให้ราก และน้ำมีการสัมผัสกับอากาศ เพื่อเพิ่มออกซิเจนในสารละลายธาตุอาหารให้มากขึ้น การปลูกแบบนี้จึงช่วยลดปัญหาการขาดอากาศของรากพืชได้ดี

ข้อดีของระบบนี้ คือ ระบบการให้สารละลายแก่พืชไม่ยุ่งยาก ง่ายต่อการดูแลรักษา พืชจะเจริญเติบโตได้ดี เพราะก้นรางแบนราบ (แต่จะมีร่องเล็กน้อย) ทำให้การกระจายตัวของออกซิเจนที่รากดีกว่าระบบอื่นๆ

ข้อเสียของระบบนี้ คือ เป็นระบบที่ต้องดูแลอย่างใกล้ชิด เพราะมีโอกาสที่ระบบจะเสียได้ง่าย และพืชจะถูกกระทบกระเทือนอย่างรุนแรงและรวดเร็ว และยังมีการแพร่กระจายของโรคพืชบางชนิดอย่างรวดเร็ว

2.9 อุปกรณ์และวงจร

2.6.1) Analog pH Meter (pH Sensor)

รูปที่ 2.2 Analog pH Meter (pH Sensor)

Analog pH Meter (pH Sensor) เป็นเซ็นเซอร์สำหรับวัดความเป็น กรด-เบส ของสารละลาย โดยค่าที่วัดได้จะอยู่ในช่วง 0 - 14pH output เป็นแบบ Analog (0-1023) ใช้ไฟเลี้ยง 5V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2) Temperature Sensor - Waterproof (DS18B20)

รูปที่ 2.3 Temperature Sensor

ไอซี DS 18B20 ใช้แรงดันไฟเลี้ยง V_{DD} (หรือ V_{CC}) ได้ในช่วง 3.0V ถึง 5.5V มี 3 ขา (สำหรับตัวถังTO-92) คือ Gnd (Pin 1), DQ (Pin 2), Vdd (Pin 3) ใช้งานได้สองแบบคือ normal mode (ใช้ทั้ง 3 ขา) และ parasite power mode (ใช้เพียง 2 ขา คือ DQ และ GND ในขณะที่ขา V_{DD} จะต่อกับขา Gnd) สามารถนำไอซีมาพ่วงต่อกันในบัสเดียว (เส้นสัญญาณ DQ) ได้หลายอุปกรณ์ ในการใช้งาน จะต้องต่อ pull-up 4.7 กิโลโอห์ม (หรือน้อยกว่าได้เล็กน้อย) ที่ขา DQ กับแรงดันไฟเลี้ยง วัดอุณหภูมิ ได้ในช่วง -55 °C ถึง +125 °C มีความแม่นย่า ±0.5 °C สำหรับอุณหภูมิในช่วง -10°C ถึง +85°C มี ความละเอียดของค่าที่อ่านได้ 12 บิต (Resolution) ใช้เวลาในการแปลงข้อมูลสำหรับ ADC ไม่เกิน 750 msec (มิลลิวินาที) สำหรับข้อมูล 12 บิต ไอซีแต่ละตัวมีหมายเลขเฉพาะตัว ขนาด 64 บิต (64-bit serial code)

2.6.3) Solenoid Valve

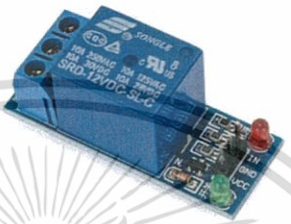
รูปที่ 2.4 Solenoid Valve

โซลินอยด์ (Solenoid) เป็นอุปกรณ์แม่เหล็กไฟฟ้าชนิดหนึ่งที่มีหลักการทำงานคล้ายกับรีเลย์ (Relay) ภายในโครงสร้างของโซลินอยด์จะประกอบด้วยขดลวดที่พันอยู่รอบแท่งเหล็กที่ภายในประกอบ ด้วยแม่เหล็กชุดบนกับชุดล่าง เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดที่พันรอบแท่งเหล็ก ทำให้แท่งเหล็ก ชุดล่างมีอำนาจแม่เหล็กดึงแท่งเหล็กชุดบนลงมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัมผัสกันทำให้ครบวงจรทำงาน เมื่อวงจรถูกตัด กระแสไฟฟ้าทำให้แท่งเหล็กส่วนล่างหมดอำนาจแม่เหล็ก สปริงก็จะดันแท่งเหล็กส่วนบนกลับสู่ ตำแหน่งปกติ จากหลักการดังกล่าวของโซลินอยด์ก็จะนำมาใช้ในการเลื่อนลิ้นวาล์วของระบบนิวเมติกส์ การปิด-เปิดการจ่ายน้ำหรือของเหลวอื่นๆ

2.6.4) 1-Channel Relay Module 12 V



รูปที่ 2.5 Relay Module

หลักการคือถ้าสั่ง digital Write (Relay, LOW) ไฟ LED จะไม่ติดเพราะ LED ใน Optocoupler ไปต่อกับ Ground เลยไม่มีกระแสไหล ตัวรับแสงไม่ได้รับแสงจึงทำหน้าที่เป็นวงจรเปิด ส่งผลให้ไม่มีกระแสจากแหล่งจ่ายไฟไหลไปที่BaseของTransistor (Ib=0) ทำให้ไม่มีกระแสที่ Collector (Ic = B x Ib = 0) และทำให้ไม่มีกระแสไปที่ Emitter (Ie = Ib + Ic = 0) และทำให้ไม่มี กระแสไปที่ขดลวดของ Relay ทำให้ Relay ไม่ทำงาน แต่พอเราสั่ง digitalWrite (Relay, HIGH) LED D1 บนบอร์ดโมดูลและ LED ใน Optocoupler U1 ก็จะติดด้วยเพราะแรงดันที่ขา IN เป็น 5V/3.3V แต่อีกด้านของ LED เป็น Ground และพอ LED ใน U1 ติดตัวรับแสงก็จะทำงาน ทีนี้ตัวรับแสงก็จะ ทำงานด้วยโดยทำ หน้าที่ปิดวงจรและมีกระแสจากแหล่งจ่ายไฟไหลไปที่ Base และมีกระแสที่ Emitterไหลไปที่ขดลวดของRelayด้วยด้วยเหตุนี้ Relay จึงทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5) Switching Power Supply 12 V 5 A

รูปที่ 2.6 Switching Power Supply 12V 5A

สวิตชิ่งเพาเวอร์ซัพพลาย (Switching Power Supply) เป็นแหล่งจ่ายไฟตรงคงค่าแรงดันแบบหนึ่ง และสามารถเปลี่ยนแรงดันไฟจากไปสลับโวลต์สูง ให้เป็นแรงดันไฟตรงค่าต่ำ เพื่อใช้ในงานอิเลคทรอนิกส์ได้เช่นเดียวกันแหล่งจ่ายไฟเชิงเส้น (Linear Power Supply) ถึงแม้เพาเวอร์ซัพพลาย ทั้งสองแบบจะต้องมีการใช้หม้อแปลงในการลดทอนแรงดันสูงให้เป็นแรงดันต่ำเช่นเดียวกัน แต่สวิตชิ่งเพาเวอร์ซัพพลายจะต้องการใช้หม้อแปลงที่มีขนาดเล็กและน้ำหนักน้อย เมื่อเทียบกับแหล่งจ่ายไฟเชิงเส้น อีกทั้งสวิตชิ่งเพาเวอร์ซัพพลายยังมีประสิทธิภาพสูงกว่าอีกด้วย

2.6.6) Arduino Mega 2560

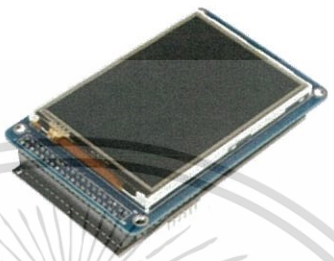
รูปที่ 2.7 Arduino Mega 2560

Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดุยโน่) เป็นบอร์ดไมโครคอนโทรเลอรต์ระกูล AVR ที่มี การพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยัง สามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วยความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจร อิเล็กทรอนิคสจ์ากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือเพื่อความสะดวกสามารถ เลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

2.6.7) TFT LCD module display



รูปที่ 2.8 TFT LCD module display

หน้าจอ TFT LCD (Thin Film Transistor) จะเป็นหน้าจอที่มีการตอบสนองต่อการแสดงผลที่ ค่อนข้างไว ประมวลผลการทำงานได้รวดเร็ว ทำให้การแสดงผลมีความคมชัดและสว่างสดใสกว่า หน้าจอแบบ STNLCD ซึ่งหน้าจอแบบนี้มักจะนำไปใช้กันอย่างแพร่หลายในอุปกรณ์อิเล็กทรอนิกส์ ราคาแพงต่างๆ เช่น Notebook, PDA, กล้องดิจิตอล รวมถึงโทรศัพท์มือถือราคาแพงหลายรุ่น แต่ ข้อเสียของหน้าจอ TFT LCD ก็คือจะใช้พลังงานไฟฟ้าค่อนข้างมาก ทำให้แบตเตอรี่ต้องมีการประจุไฟ ใหม่อยู่บ่อยครั้งหรืออาจจะต้องมีแบตเตอรี่ที่มีค่าความจุ (Ah) สูงๆ เอาไว้รองรับ

2.6.8) Float Switch Sensor

รูปที่ 2.9 Float Switch Sensor

Float Switch เป็นสวิทช์ลูกลอยไฟฟ้าใช้วัดระดับน้ำหรือของเหลว Switch จะ On/Off ขึ้นอยู่ กับระดับของเหลว สามารถต่อกับอุปกรณ์อื่นๆ เช่นต่อกับลำโพงให้ส่งเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

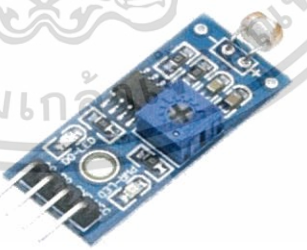
เมื่อระดับน้ำสูงเกินหรือต่ำ เกิน หรือต่อกับระบบปั๊มน้ำเพื่อให้ปั๊มทำงานหากระดับน้ำต่ำกว่าที่กำหนด และหยุดปั๊มหากระดับน้ำ สูงถึงจุดที่กำหนด

2.6.9) LED Grow Light

รูปที่ 2.10 LED Grow Light

หลอดไฟ LED Grow Light คือหลอดไฟที่ผลิตขึ้นมาสำหรับปลูกต้นไม้โดยเฉพาะ โดยใช้เทคโนโลยีสมัยใหม่มาใช้ในการผลิต โดยนำเอาคุณสมบัติพิเศษที่เราสามารถพบได้ในตัวหลอด LED มาใช้ นั่นคือให้ตัวหลอด LED ขับเฉพาะความยาวคลื่นของแสงที่เราต้องการได้ ในกรณีนี้เราจะเลือกคลื่นแสงประมาณ 430 – 460 nmและ 630 – 660 nm เนื่องจากความยาวคลื่นแสงช่วงนี้เหมาะแก่การสังเคราะห์แสงของพืชมากที่สุดและยังช่วยในการเจริญเติบโตของพืชได้มากที่สุดอีกด้วย

2.6.10) LDR Photoresistor Sensor Module

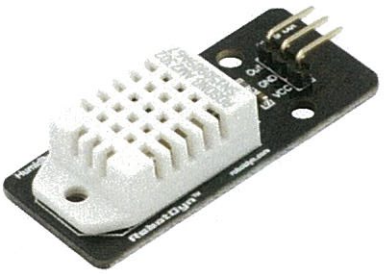


รูปที่ 2.11 LDR Photoresistor sensor module

LDR Photoresistor sensor module เป็นโมดูลสำหรับวัดความเข้มแสง ให้สัญญาณเอาต์พุตได้ทั้งแบบอนาลอกค่าระหว่าง 0-1024 และแบบดิจิตอลค่า 0 กับ 1 สามารถปรับความไวได้ที่โวลุ่มบอร์ด ใช้ไฟเลี้ยง 3.3-5V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.11) DHT22 Temperature and Humidity Sensor Module



รูปที่ 2.12 DHT22

เป็นโมดูลที่สามารถวัดอุณหภูมิและความชื้นสภาพแวดล้อม ใช้แรงดันไฟเลี้ยงได้ในช่วง 3.3-5.5V ให้สัญญาณเอาท์พุทแบบ Digital Output สามารถวัดอุณหภูมิได้ในย่าน -48 ถึง 80 องศาเซลเซียส

2.6.12) ESP8266

รูปที่ 2.13 ESP8266

SP8266 คือ โมดูล wifi ภายในมีเฟิร์มแวร์ทำงานในลักษณะ Serial-to-WiFi ที่ช่วยให้อุปกรณ์อื่นๆสามารถต่อเข้ากับอินเตอร์เน็ตได้ โดยใช้พอร์ตและใช้คำสั่ง AT ในการควบคุมการทำงาน ทำงานที่แรงดันไฟฟ้า 3.0-3.6V ทำงานโดยใช้กระแสเฉลี่ย 80 mA ภายในมี Low power MCU 32บิท ทำให้เราเขียนโปรแกรมสั่งงานได้ มีวงจร analog digital converter ทำให้สามารถอ่านค่าจาก analog ได้ความละเอียด 10 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.13) พัดลมระบายความร้อน

รูปที่ 2.14 พัดลมระบายความร้อน

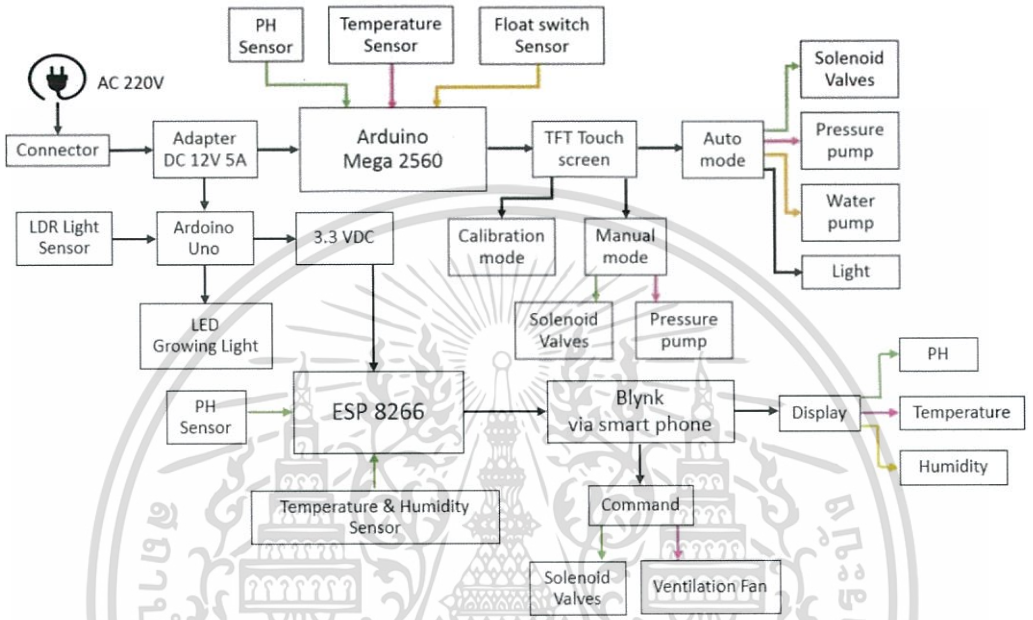
พัดลมระบายความร้อนขนาด 12V ใช้ดูดความชื้นออกจากระบบไฮโดรโปนิกส์ อีกทั้งยังช่วยลดอุณหภูมิด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการทำงานและการออกแบบ

3.1 Block Diagram



รูปที่ 3.1 Block Diagram ของระบบไฮโดรโปนิกส์

ระบบไฮโดรโปนิกส์นั้นเป็นการปลูกพืชโดยไม่ใช้ดินแต่จำเป็นต้องมีปัจจัยควบคุมต่างๆในการเจริญเติบโตของพืช เช่น ค่า pH ในน้ำ,ค่า Electro Conductivity ของน้ำ,แสง,ระดับน้ำและอุณหภูมิ จากรูปที่ 3.1 สามารถอธิบายการทำงานของวงจรได้ว่า Arduino Mega 2560 , Arduino Uno และ ESP8266 รับค่าปัจจัยควบคุมการเจริญเติบโตของพืชจากเซ็นเซอร์ต่างๆ ได้แก่ pH Sensor, Temperature Sensor, LDR light Sensor, Temperature and Humidity Sensor และ Float Switch Sensor แล้วแสดงผลผ่านทาง LCD และสมาร์ทโฟน โดยทางจอ LCD สามารถเลือกการทำงานได้ 3 ระบบ

1.) ระบบการคาลิเบทค่าความเป็นกรดและเบสสำหรับโพรบพีเอช

2.) ระบบอัตโนมัติจะสามารถควบคุมการเปิด-ปิด Solenoid Valve เพื่อปรับค่าpH ในน้ำ, ควบคุมการเปิด-ปิด Pump Pressure พ่นละอองน้ำเพื่อลดอุณหภูมิ, ควบคุม Water Pump เพื่อปรับระดับน้ำในถังสารละลาย และควบคุมการเปิด-ปิดของหลอดไฟ LED Grow Light เพื่อให้แสงแก่พืช

3.) ระบบปรับด้วยมือ ผู้ใช้สามารถควบคุมการเปิด-ปิด Solenoid Valve เพื่อปรับค่า pH ในน้ำและ

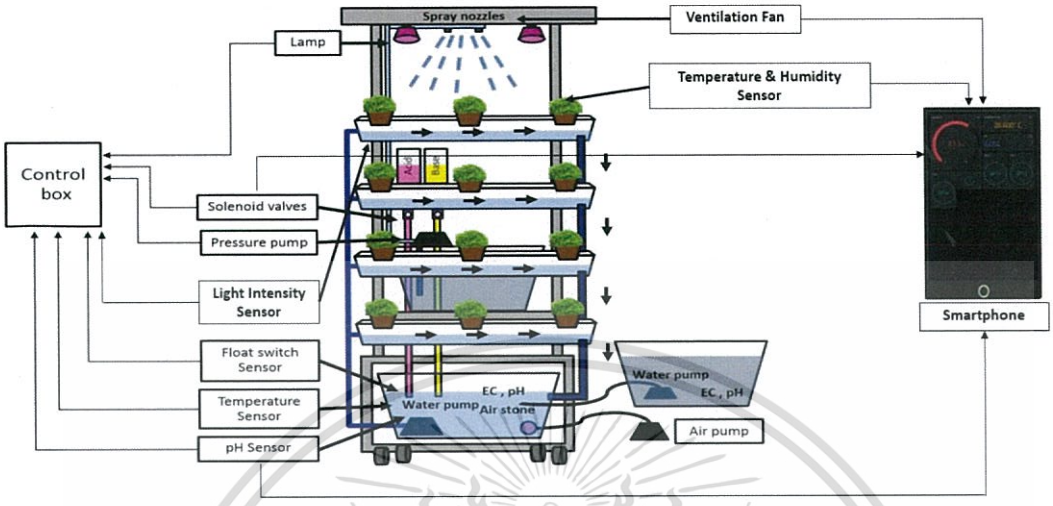
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถควบคุมการเปิด-ปิด Pump Pressure พ่นละอองน้ำเพื่อลดอุณหภูมิ ส่วนทางสมาร์ทโฟนสามารถทำการแสดงค่า pH, อุณหภูมิ และความชื้นในขณะนั้น พร้อมทั้งยังสามารถปรับค่าด้วยมือได้ โดยผู้ใช้สามารถควบคุมการเปิด-ปิด Solenoid Valve เพื่อปรับค่า pH ในน้ำและสามารถควบคุมการเปิด-ปิดพัดลมระบายอากาศเพื่อลดค่าความชื้นในอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

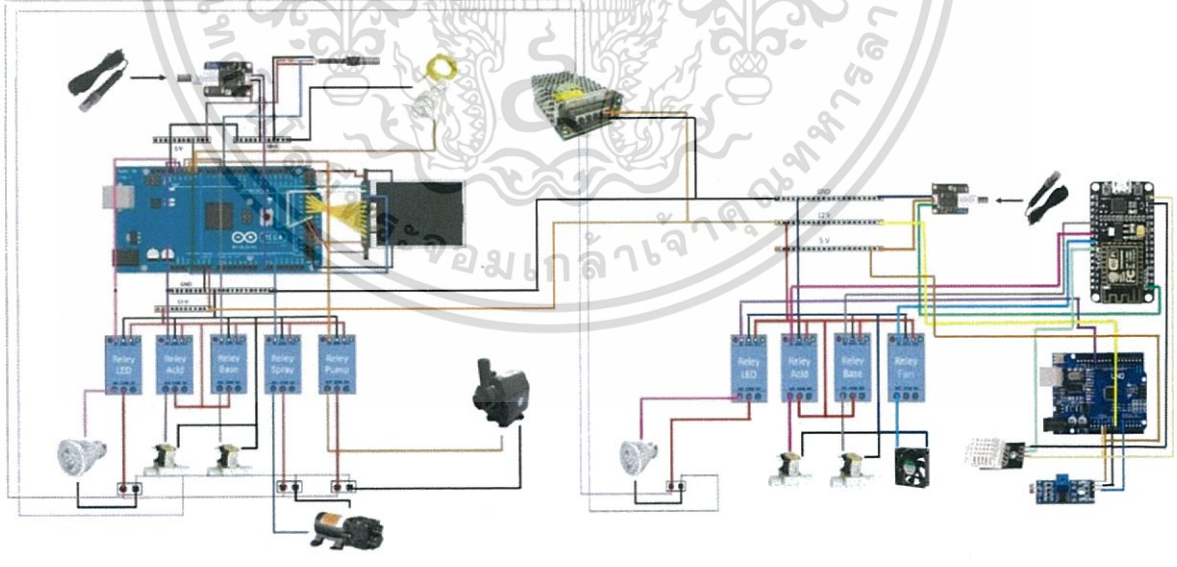
3.2 การออกแบบ

3.2.1) Design Hydroponic System



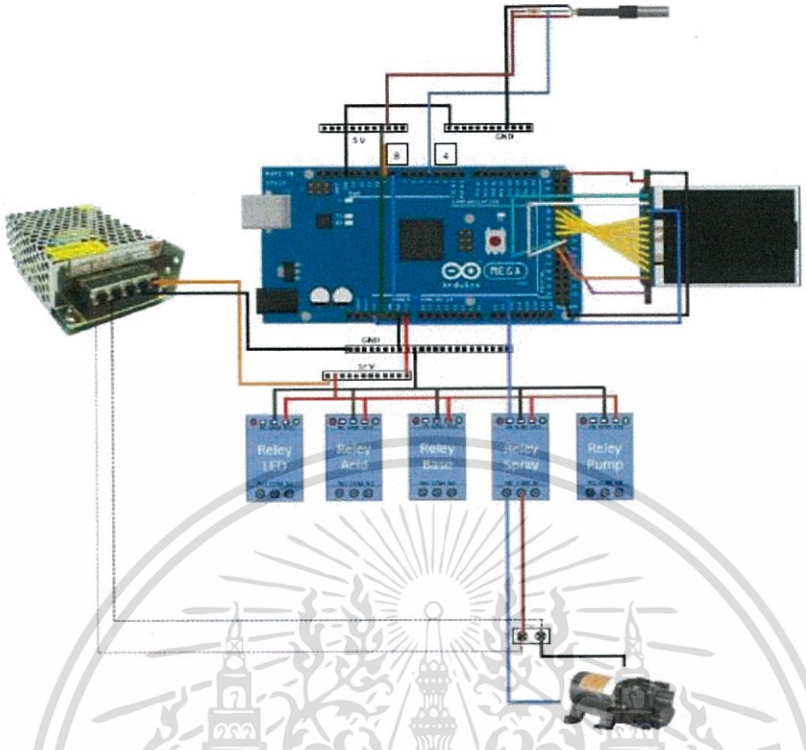
รูปที่ 3.2 โครงสร้างของระบบไฮโดรโปนิกส์

3.2.2) Schematic

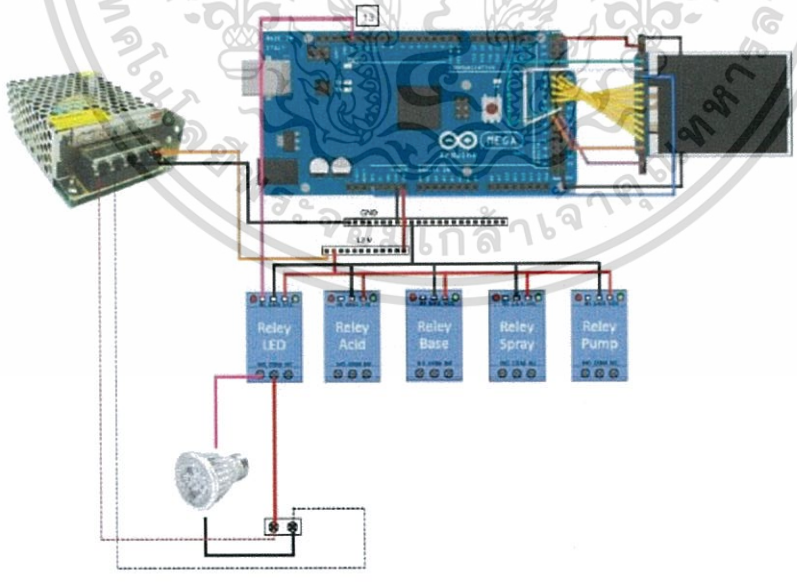


รูปที่ 3.3 วงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

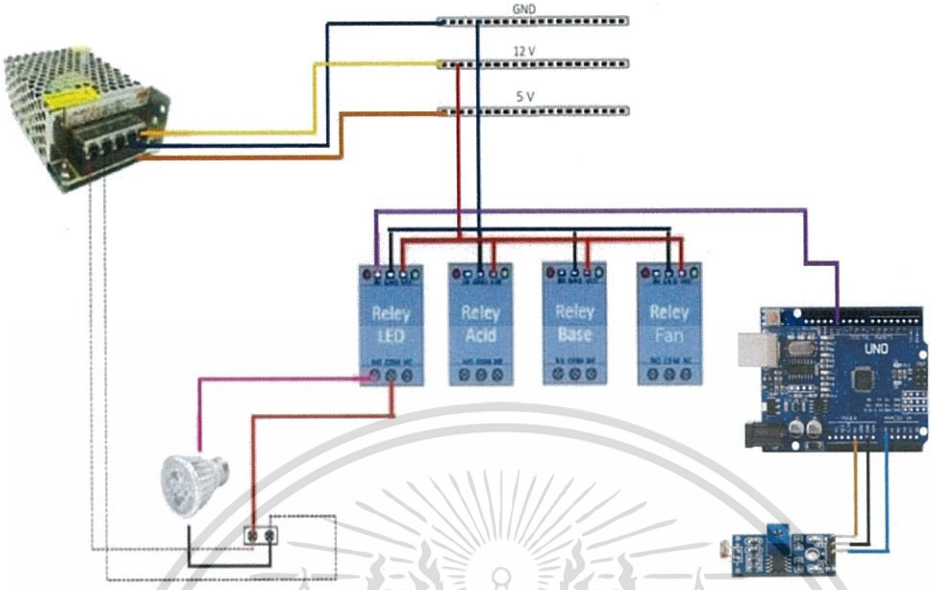


รูปที่ 3.4 วงจรควบคุมอุณหภูมิ

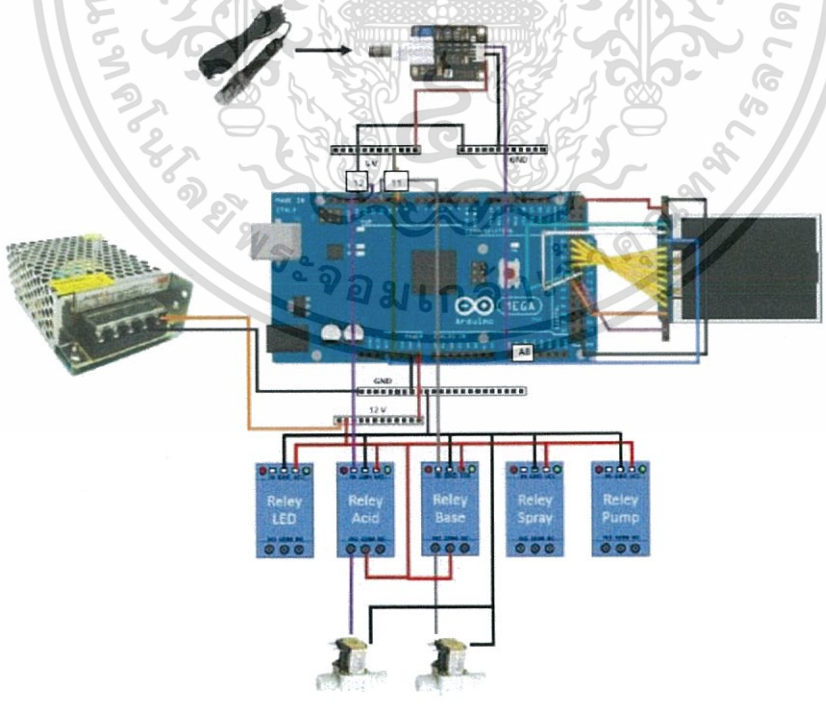


รูปที่ 3.5 วงจรควบคุมหลอดไฟ ผ่าน Arduino **Mega2560**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

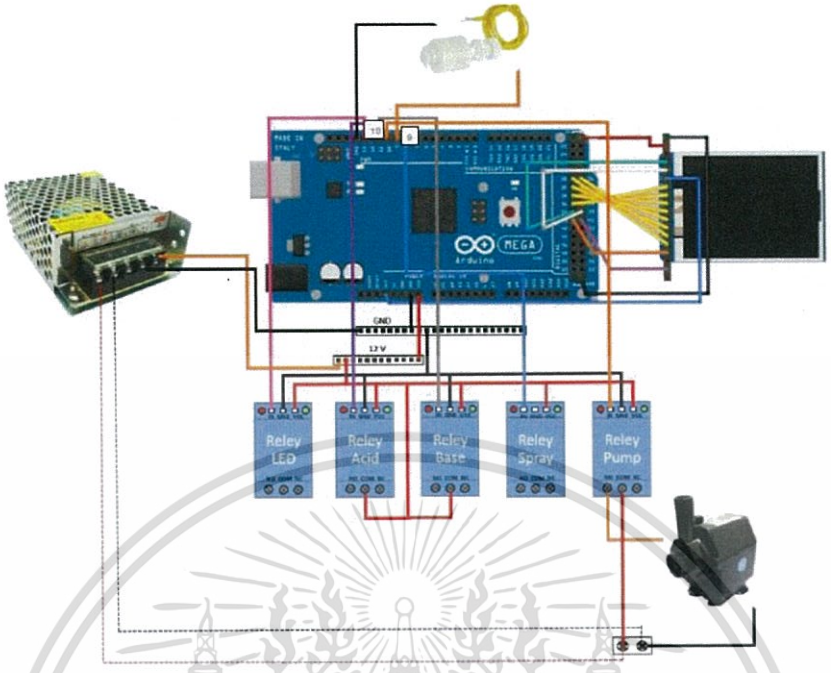


รูปที่ 3.6 วงจรควบคุมหลอดไฟ ผ่าน Arduino Uno

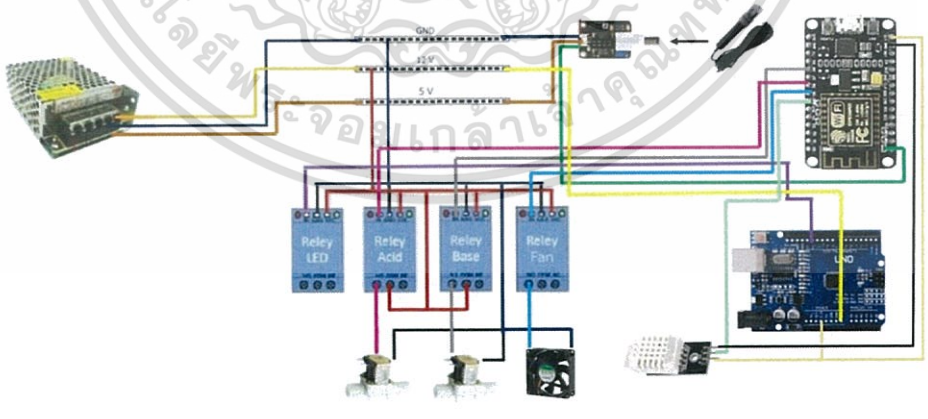


รูปที่ 3.7 วงจรควบคุมค่าความเป็นกรดเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



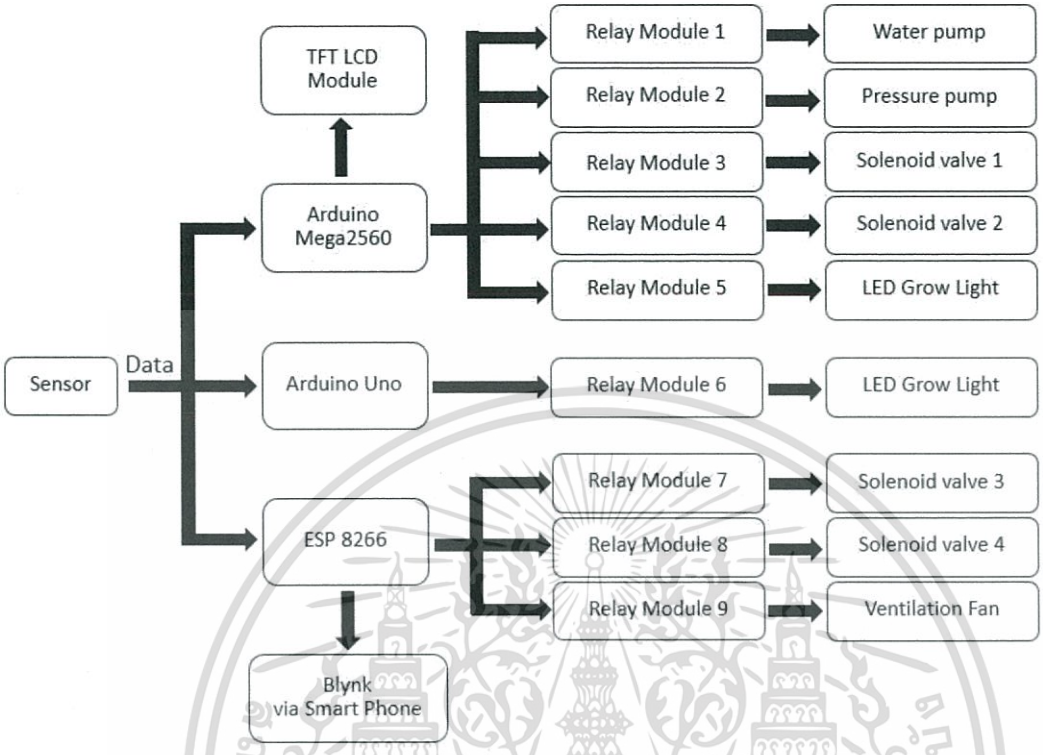
รูปที่ 3.8 วงจรควบคุมระดับน้ำ



รูปที่ 3.9 วงจรสำหรับควบคุมและแสดงผลผ่านสมาร์ทโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การทำงานของวงจร



รูปที่ 3.10 ขั้นตอนการทำงานของระบบ

3.3.1 Arduino Mega2560

เมื่อ Arduino Mega 2560 รับค่าจากเซนเซอร์ต่างๆ ได้แก่ Temperature Sensor, Float Switch Sensor และ pH Sensor จะทำการประมวลผลข้อมูลและแสดงผลผ่าน TFT LCD Module โดยการทำงานนั้นสามารถเลือกโหมดได้ 3 ระบบ คือ

3.3.1.1) ระบบแบบอัตโนมัติ แบ่งออกเป็น

1. การควบคุม pH แบ่งเป็นการเติมกรดและการเติมเบส การเติมกรดจะทำงานเมื่อค่า pH มีค่าสูงกว่าค่าที่กำหนด Arduino จะส่งสัญญาณ logic 1 ไปให้ Relay Module3 ทำให้ relay ทำงาน โซลินอยวาล์ว1 จึงทำงานเพื่อเติมกรด ส่วนการเติมเบสจะทำงานเมื่อค่า pH มีค่าต่ำกว่าค่าที่ กำหนด Arduino จะส่งสัญญาณ logic 1 ไปให้ Relay Module4 ซึ่งหน้าที่เป็นเหมือนสวิตช์ โซลินอยวาล์ว2 จึงทำงานเพื่อเติมเบส

2. การควบคุมอุณหภูมิจะทำงานเมื่ออุณหภูมิสูงกว่า 29 องศาเซลเซียส Arduino จะส่งสัญญาณ logic 1 ไปให้ Relay Module2 Pressure Pump จึงทำงานโดยพ่นละอองน้ำเพื่อลดอุณหภูมิ เป็นเวลา 15 วินาทีและจะหยุดทำงานเป็นเวลา15นาที จากนั้นเซนเซอร์จะทำ

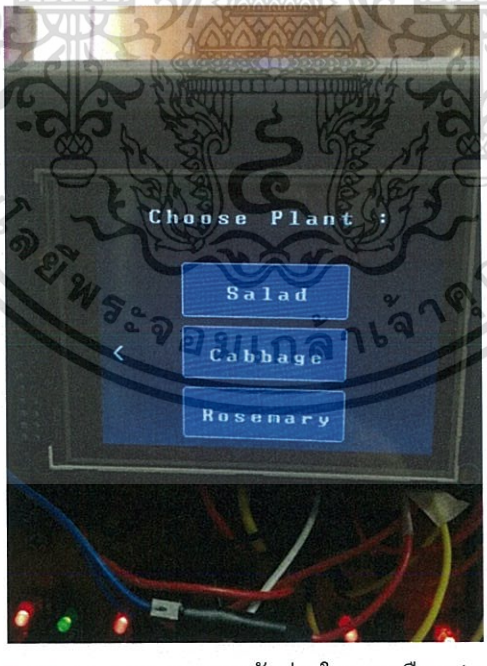
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเช็คค่า อุณหภูมิใหม่ ถ้าอุณหภูมิสูงกว่า 29 องศาเซลเซียส Pressure Pump ก็จะกลับมา ทำงานอีก

3. การควบคุมระดับน้ำในถังสารละลายจะทำงานเมื่อระดับน้ำในถังสารละลายลดต่ำลง Arduino จะส่งสัญญาณ logic 1 ไปให้ Relay Module1 Water Pump จึงทำงานโดยสูบน้ำจากถังสำรองเข้าสู่ถังสารละลาย

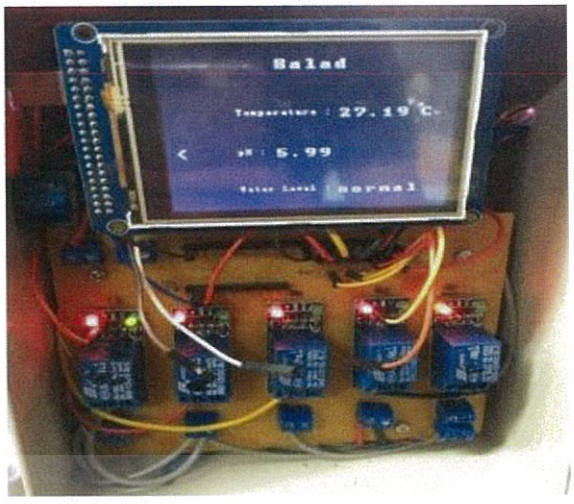
4. การควบคุมแสงสว่างทำได้โดยใช้ Arduino เป็นตัวจับเวลาในการเปิด-ปิดหลอด LED Grow Light โดยการเปิดหลอด LED Grow Light นั้น Arduino จะส่งสัญญาณ logic1 ไปให้ Relay Module5 เพื่อให้แสงสว่างเป็นเวลา 6 ชั่วโมง เมื่อครบ 6 ชั้วโมงแล้ว Arduino จะส่งสัญญาณ logic 0 ไปให้ Relay Module5 ทำให้หลอดไฟดับเป็นเวลา 1 ชั่วโมงตามเวลาที่ตั้งไว้

ในโครงงานนี้เราออกแบบให้ปลูกผักไว้ 3 ชนิดได้แก่ ผักสลัด, ผักกะหล่ำและโรสแมรี่ ซึ่งระบบอัตโนมัติจะควบคุมค่าความเป็นกรดเบสของสารละลายที่พืชแต่ละชนิดต้องการแตกต่างกันออกไป เมื่อเลือกปลูกผักชนิดใดแล้วก็จะมีหน้าจอ LCD แสดงผลบอกถึงอุณหภูมิ ค่าความเป็นกรดเบส และระดับน้ำในถังของผักนั้นๆ หากต้องการปลูกผักชนิดอื่นก็สามารถเพิ่มในส่วนของโปรแกรม



รูปที่ 3.11 จอ LCD แสดงหน้าต่างในการเลือกปลูกผัก

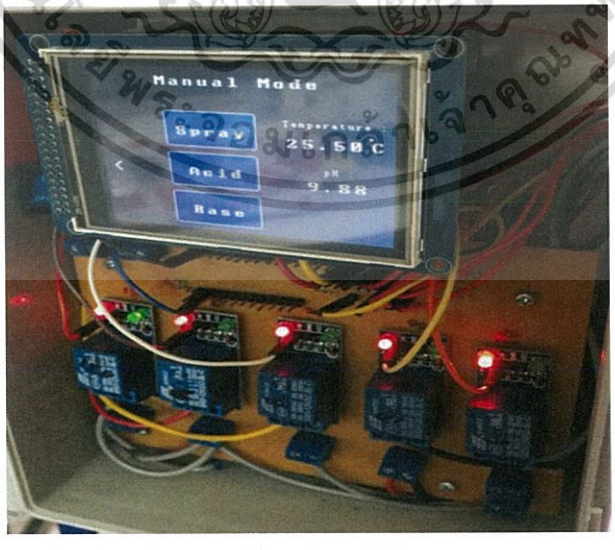
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 จอ LCD แสดงหน้าต่างค่าอุณหภูมิ ค่าพีเอช และค่าระดับน้ำในถังสารละลาย

3.3.1.2) ระบบปรับค่าด้วยมือ แบ่งออกได้เป็น

1. การปรับค่าความเป็นกรดเบสนั้นผู้ใช้สามารถเลือกเติมกรดหรือเบสได้ด้วยตัวเองโดยการเติมกรดนั้นจะกดปุ่ม Acid และหยุดเติมกรดโดยการกดปุ่มเดิมซ้ำอีกครั้ง ส่วนการเติมเบส จะกดปุ่ม Base และหยุดเติมเบสโดยการกดปุ่มเดิมซ้ำอีกครั้งเช่นกัน ระบบจะแสดงค่าความเป็นกรดเบสของสารละลายบนจอ LCD ควบคู่ไปด้วย
2. การปรับค่าอุณหภูมินั้นผู้ใช้สามารถสั่งให้ระบบพ่นละอองน้ำให้แก่ผักได้โดยการกดปุ่ม Spray และหยุดพ่นละอองน้ำโดยการกดปุ่มเดิมซ้ำอีกครั้งระบบจะแสดงค่าอุณหภูมิควบคู่ไปด้วย

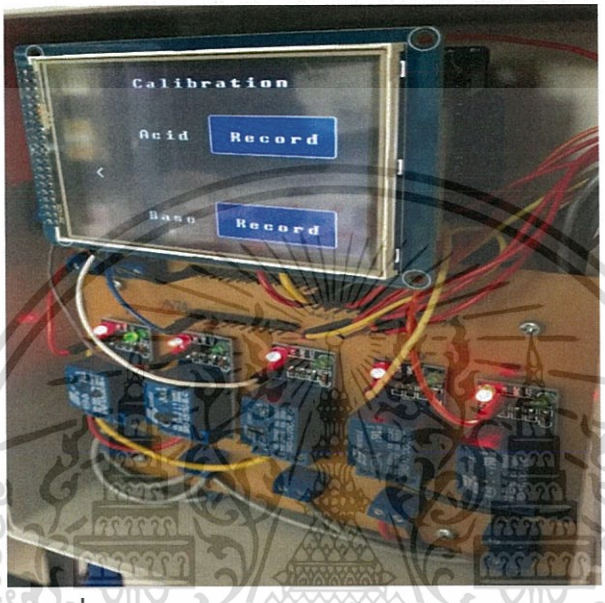


รูปที่ 3.13 จอ LCD แสดงหน้าต่างระบบแมนนวล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.3) ระบบการคาริเบทพีเอชเซนเซอร์

ในการคาริเบทพีเอชเซนเซอร์นั้นจะมีปุ่มแสดงบนหน้าจอ LCD อยู่สองปุ่มคือปุ่มบันทึกค่าเบสและกรดสามารถท่าได้โดยจุ่มโพรบ pH ที่สารละลายความเข้มข้น 4.00 แล้วบันทึกค่า voltage ที่ได้จากนั้นล้างหัวโพรบ pH แล้วน่ามาจุ่มสารละลายความเข้มข้น 6.86 แล้วบันทึกค่า voltage เพื่อเป็นการปรับเทียบค่า pH ให้สามารถอ่านค่าได้ถูกต้อง



รูปที่ 3.14 จอ LCD แสดงหน้าต่างระบบคาริเบท

3.3.2 Arduino Uno

เมื่อ Arduino Uno รับข้อมูลจาก LDR Light Sensor จะทำการประมวลผลข้อมูล หากมีความเข้มแสงน้อยกว่า 20,000 ลักซ์ Arduino จะส่งสัญญาณ logic 1 ไปที่ Relay Module6 เพื่อสั่งการให้เปิด LED Grow Light

3.3.3 ESP8266

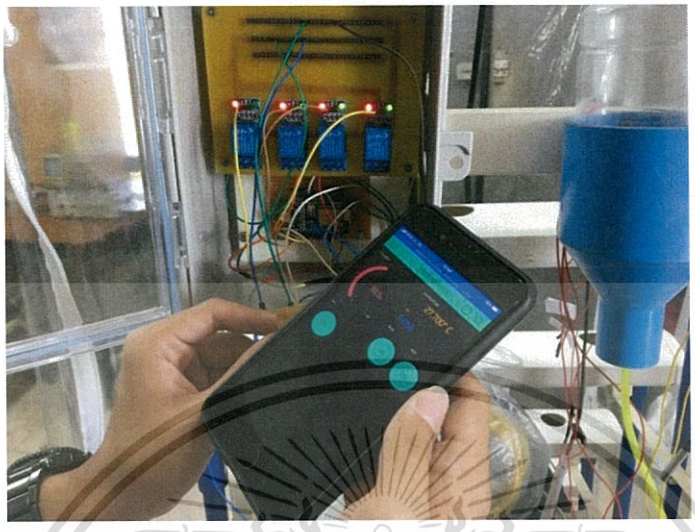
เมื่อ ESP8266 รับค่าจากเซนเซอร์ต่างๆ คือ Temperature & Humidity Sensor และ Analog pH Sensor จะทำการประมวลผลข้อมูลและแสดงผลผ่านแอพพลิเคชั่น Blynk ในสมาร์ทโฟน โดยในการทำงานนั้นแบ่งออกเป็น

1.การแสดงผลข้อมูล เป็นการแสดงค่าอุณหภูมิและความชื้นที่วัดได้จาก DHT22 และค่า pH ซึ่งวัดได้จาก Analog pH Sensor

2.ระบบปรับค่าด้วยมือ สามารถควบคุมการเปิด-ปิด โซลินอยด์วาล์ว เพื่อควบคุมค่า pH และพัดลมระบายอากาศ เพื่อควบคุมค่าความชื้นในอากาศ โดย ESP8266 จะส่งสัญญาณ logic 1 ไปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Relay Module 7และ8 เพื่อสั่งให้เปิดโซลินอยด์วาล์ว และจะส่งสัญญาณ logic 1 ไปที่ Relay Module 9 เพื่อสั่งการให้เปิดพัดลมระบายอากาศ

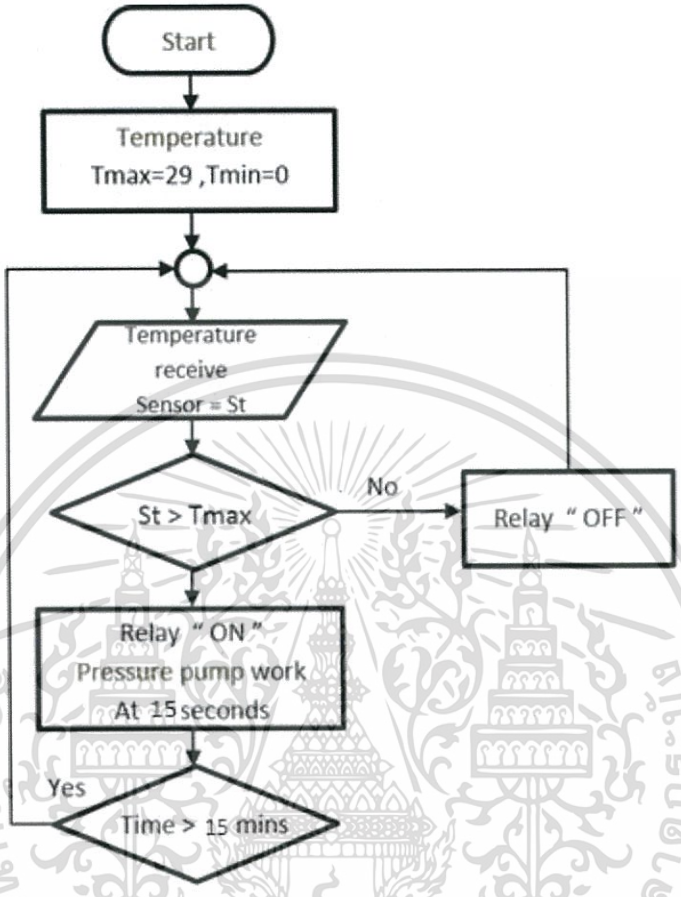


รูปที่ 3.15 การแสดงผลและควบคุมผ่านสมาร์ทโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 Flow chart

3.4.1) แผนผังการทำงานของ Pump Pressure



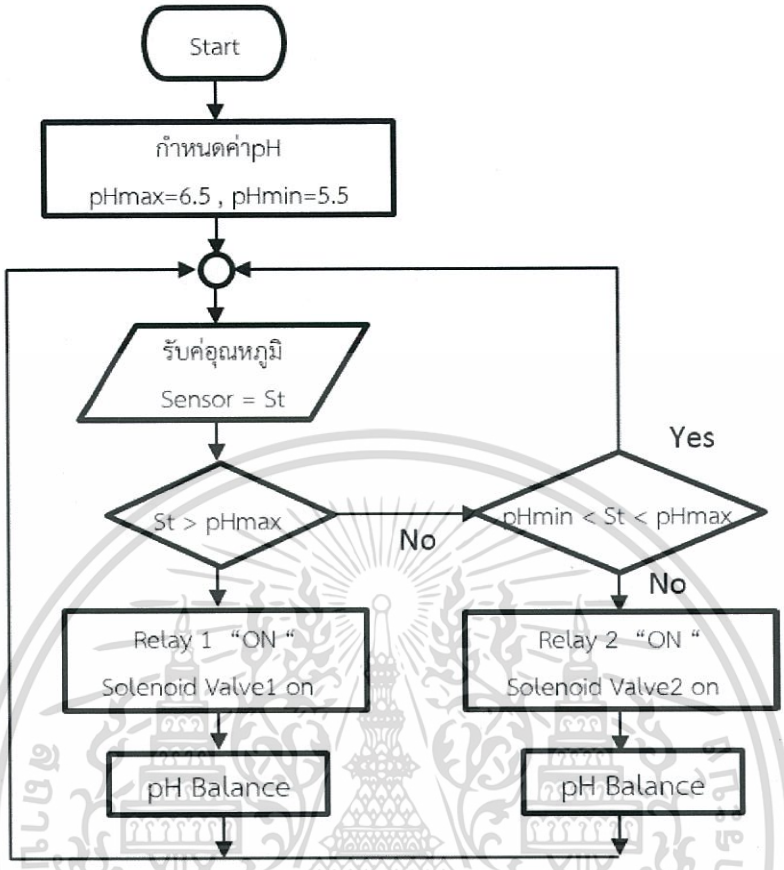
รูปที่ 3.16 แผนผังการควบคุมการทำงานของ Pressure Pump

Dissolved Oxygen (mg/L) at Saturation in freshwater		
Temperature (C^0)	(F^0)	Dissolved Oxygen at saturation (mg/L)
0^0	32^0	14.6
5^0	41^0	12.8
10^0	50^0	11.3
15^0	59^0	10.1
20^0	68^0	9.1
25^0	77^0	8.3
30^0	86^0	7.6

รูปที่ 3.17 ค่าอุณหภูมิที่ส่งผลต่อปริมาณออกซิเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2) แผนผังการปรับค่า pH ของน้ำ



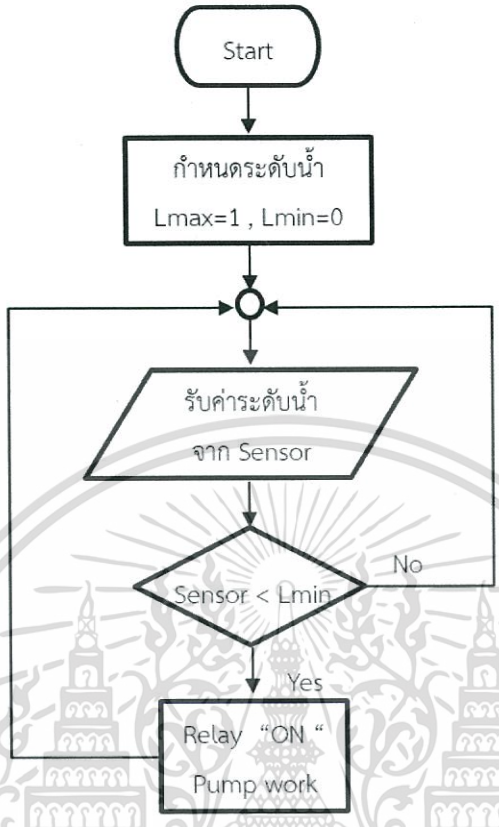
รูปที่ 3.18 แผนผังการควบคุมค่า pH ของน้ำ

พืช	pH	EC (ms/cm)
กระเจี๊ยบเขียว	6.0 - 6.5	2.0 - 3.0
กะหล่ำดอก	6.0 - 6.5	2.5 - 3.0
กะหล่ำปลี	6.5 - 7.0	2.5 - 3.0
กุหลาบ	5.0 - 6.0	1.8 - 2.2
แกลดิโอลัส	5.5 - 6.5	1.6 - 2.4
ขึ้นฉ่าย	6.0 - 6.5	2.0 - 3.0
คะน้า	6.0 - 6.5	3.5 - 4.5
แคนตาลูป	5.5 - 6.5	2.0 - 2.5
แครอท	5.8 - 6.3	1.8 - 2.2
จิปโซฟิลล่า	6.0 - 6.5	1.2 - 1.8
ซูกินี	6.0 - 6.5	1.2 - 2.0
เซอราลี	6.0 - 6.5	2.5 - 3.0
ดาวเรือง	5.5 - 6.5	1.6 - 2.4
แตงกวา	5.5 - 6.0	1.0 - 2.5
แตงโม	5.5 - 6.5	1.7 - 2.5
ถั่วทั่วไป	5.5 - 6.2	2.0 - 4.0
บล็อกโคลี	6.0 - 6.5	2.5 - 3.0
บีโกเนีย	6.0 - 6.5	1.4 - 1.8
ปวยเล้ง	5.0 - 7.0	1.5 - 2.0
ผักกวางตุ้งใบ, กวางตุ้งดอก	6.0 - 7.0	1.8 - 2.5
ผักกวางตุ้งฮ่องเต้	6.0 - 7.0	2.0 - 3.0
ผักกาดหอม (สลัดใบอ่อน)	5.5 - 6.5	1.0 - 1.5
ผักกาดหอมห่อ (สลัดแก้วและสลัดใบกรอบ)	5.5 - 6.5	1.0 - 1.6

รูปที่ 3.19 แสดงค่า pH ที่พืชแต่ละชนิดต้องการ

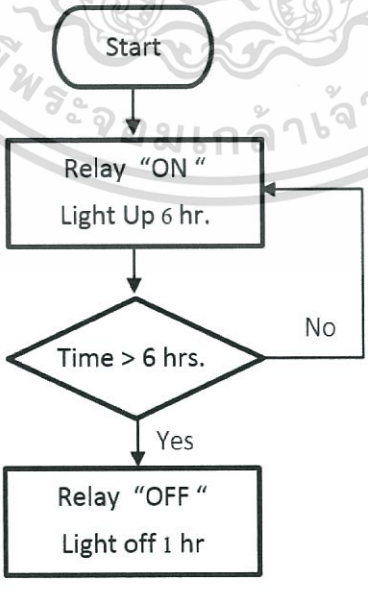
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3) แผนผังการควบคุม Water Level



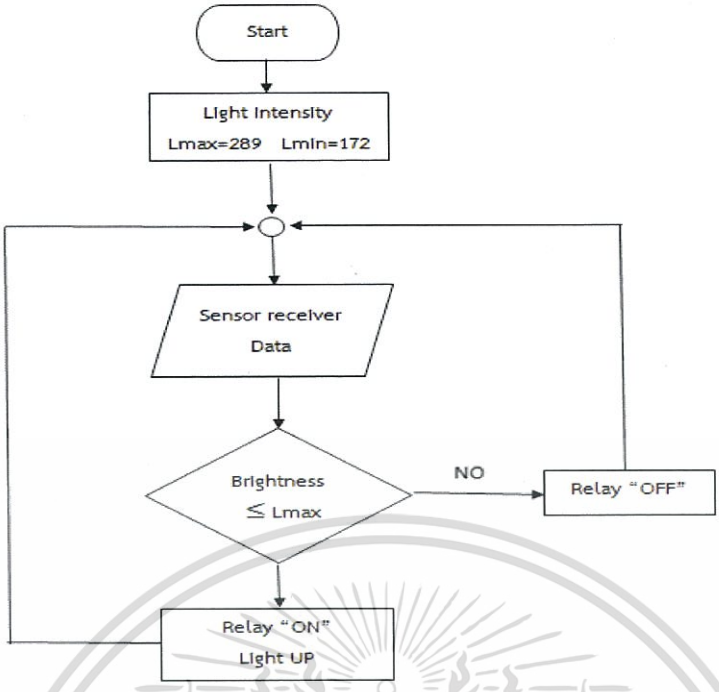
รูปที่ 3.20 แผนผังควบคุม Water Level

3.4.4) แผนผังการควบคุมหลอดไฟ



รูปที่ 3.21 แผนผังควบคุม LED ผ่าน Arduino Mega2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 แผนผังควบคุม LED ผ่าน Arduino Uno

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลอง

การทดลองนั้นแบ่งเป็น 3 ส่วนคือ

4.1.1) การทดลองสั่งการผ่านหน้าจอ LCD Touch Screen

4.1.1.1) การทดลองระบบอัตโนมัติ

1. การเปิดปิดของหลอดไฟ ซึ่งหลอดไฟจะสว่างเป็นเวลา 6 ชั่วโมง และดับเป็นเวลา 1 ชั่วโมง
2. การเปิดปิดของ Pump Pressure ตามอุณหภูมิ เมื่ออุณหภูมิมีค่ามากกว่า 29 องศาเซลเซียส Pump Pressure จะทำงานโดยพ่นละอองน้ำเพื่อลดอุณหภูมิเป็นเวลา 15 วินาทีและจะหยุดทำงานเป็นเวลา15นาที จากนั้นเซนเซอร์จะทำการเช็คค่าอุณหภูมิใหม่ ถ้าอุณหภูมิสูงกว่า 29 องศาเซลเซียส Pressure Pump ก็จะทำงานอีกครั้ง
3. การควบคุมโซลินอยวาล์ว เพื่อปรับค่าความเป็นกรดเบสของสารละลายโดย โซลินอยวาล์ว 3 จะทำงาน เมื่อ pH มีค่าสูงกว่าที่กำหนด โดยจะปล่อยกรดออกมาและหยุดทำงานเมื่อ pH ต่ำกว่าที่กำหนด โซลินอยวาล์ว 4 จะทำงานเมื่อ pH มีค่าต่ำที่กำหนด โดยจะปล่อยเบสออกมาและหยุดทำงานเมื่อ pH สูงกว่าที่กำหนด ซึ่งในผักแต่ละชนิดมีช่วง pH ในการปลูกต่างกันคือ ผักสลัด pH = 5.5-6.5, ผักกะหล่ำปลี pH = 6.5-7, โรสแมรี่ pH = 6.6-7.8
4. การควบคุมระดับน้ำในถังสารละลายเมื่อระดับน้ำลดลงต่ำ Water pump จะทำการสูบน้ำ จากถังสำรองเข้าสู่ถังสารละลาย

4.1.1.2) การทดลองระบบปรับค่าด้วยมือ

1. ผู้ใช้สามารถเลือกเติมกรดหรือเบสได้ด้วยตัวเอง ในกรณีที่ค่า pH ไม่ได้อยู่ในช่วงที่ที่ต้องการโดยการกดปุ่ม Acid หรือ Base และหยุดทำงานโดยการกดปุ่มเดิมซ้ำอีกครั้ง
2. ผู้ใช้สามารถเปิดการสเปรย์น้ำเองได้ ในกรณีที่ค่าอุณหภูมิไม่ได้อยู่ในช่วงที่กำหนด โดยการเลือกกดปุ่ม Spray และจะหยุดทำงานเมื่อกดปุ่มเดิมซ้ำอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.3) การทดลองระบบคาริเบท

การคาริเบท pH Electrode ทำได้โดยการนำ pH Electrode ไปจุ่มสารละลายที่มี pH = 4 กดปุ่น Record บันทึกค่า Voltage จากนั้นนำ pH Electrode ไปจุ่มสารละลายที่มี pH = 7 กดปุ่ม Record บันทึกค่า Voltage นำมาเปรียบเทียบกับสมการเส้นตรง

$y = mx + c$ จะได้สมการ $Voltage = m(pH) + c$ (1)

จากนั้นท่าการหาค่า m ได้จาก $m = \frac{Volt(pH6.86) - Volt(pH4)}{6.86 - 4}$ (2)

เมื่อได้ m แล้วน่าไปแทนค่าใน (1) จะได้ค่า c จากนั้น ท่าการย้ายข้างสมการเพื่อหา pH ในสารละลายอื่นๆจะได้สมการ $pH = \frac{(Voltage - c)}{m}$ (3)

4.1.2) การทดลองระบบเปิด-ปิดไฟอัตโนมัติตามความเข้มแสง

เปิด-ปิดไฟในห้องเพื่อทดสอบว่าหลอดไฟติดเมื่อปิดไฟในห้อง และดับเมื่อเปิดไฟในห้อง

4.1.3) การทดลองสั่งการผ่านสมาร์ทโฟน

4.1.3.1) การทดลองระบบแสดงผล

1. ทดสอบอุณหภูมิที่แสดงในหน้าจอสมาร์ทโฟนเทียบกับเทอร์โมมิเตอร์
2. ทดสอบค่าความชื้นที่แสดงในหน้าจอสมาร์ทโฟนเทียบกับไฮโกรมิเตอร์
3. ทดสอบค่า pH ที่แสดงในหน้าจอสมาร์ทโฟนเทียบกับ Digital pH Meter

4.1.3.2) การทดลองระบบบรับค่าด้วยมือ

1. ผู้ใช้สามารถเติมกรดและเบสได้เอง ในกรณีที่ค่า pH ไม่ได้อยู่ในช่วงที่ต้องการ โดยการกดปุ่ม ON ที่ Acid หรือ Base และจะหยุดทำงานเมื่อกดปุ่ม OFF
2. ผู้ใช้สามารถเปิด-ปิดพัดลมระบายอากาศได้ ในกรณีที่ค่าความชื้นไม่ได้อยู่ในช่วงที่ต้องการ โดยกดปุ่ม ON ที่ Fan และจะหยุดทำงานเมื่อกดปุ่ม OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง

4.2.1) ผลการทดลองการสั่งการผ่านหน้าจอ LCD Touch Screen

4.2.1.1) ผลการทดลองในระบบอัตโนมัติ

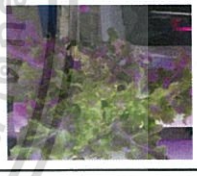
1.หลอดไฟจะเปิดเป็นเวลา 6 ชั่วโมง และดับเป็นเวลา 1 ชั่วโมง ตามที่เรากำหนด

2. เมื่ออุณหภูมิสูงกว่า 29 องศาเซลเซียส จะทำการสเปรย์น้ำเพื่อลดอุณหภูมิเป็นเวลา 15 วินาทีตามที่กำหนด

3. โซลินอยวาล์วในโมดูลตัวที่ 3 และ 4 ทั้ง 2 ตัวสามารถปล่อยกรดเบสได้ออกมาในปริมาณที่เหมาะสมตามความต้องการของผักแต่ละชนิด

4. การทดลองพบว่าเมื่อระดับน้ำลดต่ำลง สวิทช์ลูกลอยไฟฟ้าสามารถทำงานได้อย่างปกติ โดย water pump จะสูบน้ำจากถังสำรองเข้าสู่ถังสารละลายจนมีระดับน้ำปกติ

ตารางที่ 4.1 แสดงการเจริญเติบโตของพืชในระบบไฮโดรโปนิกส์ในระบบอัตโนมัติ

สัปดาห์ / ชนิดผัก	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3-4	สัปดาห์ที่ 4-5
สลัดกรีนโอ๊ค				
ระยะเวลาในการปลูก	ระยะเพราะต้นกล้าประมาณ 7 วัน	ลงในรางปลูกประมาณ 14 วัน	ลงในรางปลูกประมาณ 25 วัน	ลงในรางปลูกประมาณ 35 วัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.2) ผลการทดลองในระบบปรับค่าด้วยมือ

จากการทดลองพบว่าสามารถควบคุมการทำงานได้ทั้ง 3 ปุ่ม คือ Acid, Base และ Spray

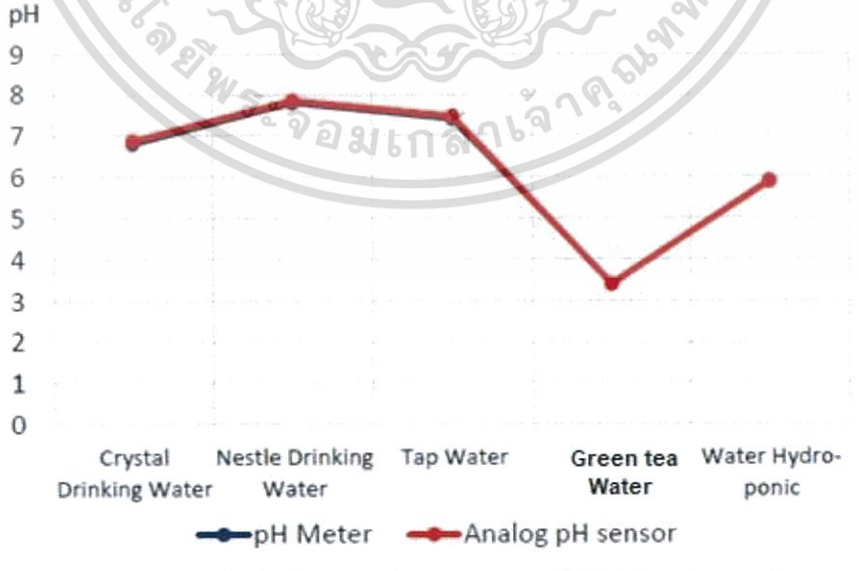
4.2.1.3) ผลการทดลองในระบบคาริเบท

ตารางที่ 4.2 แสดงการเปรียบเทียบค่า pH ระหว่าง pH Meter เทียบกับ Analog pH Sensor

ชนิดสารละลายตัวอย่าง	ค่า pH ที่ได้จาก pH Meter	ค่า pH ที่ได้จาก Analog pH Sensor	ค่าความคลาดเคลื่อน % Error
น้ำดื่มคริสตัล	6.8	6.88	1.18
น้ำดื่มเนสท์เล่	7.8	7.86	0.77
น้ำประปา	7.4	7.48	1.08
น้ำชาเขียว	3.4	3.39	0.29
น้ำจากระบบไฮโดรโปนิกส์	5.9	5.88	0.34

<u>ค่าความคลาดเคลื่อน (%Error)</u>

$$\%\text{Error} = \frac{\text{ค่าที่ได้จากพีเอชมิเตอร์} - \text{ค่าที่ได้จากพีเอชเซนเซอร์}}{\text{ค่าจากพีเอชมิเตอร์}} \times 100\% \quad (4)$$



รูปที่ 4.1 กราฟเปรียบเทียบค่า pH ที่ได้จาก pH Meter เทียบกับ Analog pH Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงความสัมพันธ์ระหว่างค่า pH ของสารชนิดต่างๆ กับ Voltage

ชนิดสาร	น้ำชาเขียว	ผงปรับ pH = 4.0	น้ำจากระบบไฮโดรโปนิกส์	ผงปรับ pH = 6.86	น้ำดื่มคริสตัล	น้ำประปา	น้ำดื่มเบสท์แล่
pH	3.39	4	5.88	6.86	6.88	7.48	7.86
Voltage	1.07	1.20	1.66	1.90	1.92	2.03	2.16

รูปที่ 4.2 กราฟแสดงความเป็นเชิงเส้นของค่า pH เทียบกับ Voltage

4.2.2) ผลการทดลองระบบเปิด-ปิดไฟอัตโนมัติตามความเข้มแสง

พบว่าทำงานได้ปกติดี คือ LED ติดเมื่อแสงน้อย และดับเมื่อมีแสงเพียงพอ

4.2.3) ผลการทดลองสั่งการผ่านสมาร์ทโฟน

4.2.3.1) ผลการทดลองระบบแสดงผล แสดงค่าอุณหภูมิ ความชื้น และ pH ได้อย่างแม่นยำ แต่มีความล่าช้าอยู่บ้างตามความเร็วของอินเตอร์เน็ตที่ใช้ในการเชื่อมต่อ

4.2.3.2) ผลการทดลองระบบปรับค่าด้วยมือ พบว่าสามารถควบคุมการทำงานได้ทั้ง 3 ปุ่ม คือ Acid, Base และ Fan แต่มีความล่าช้าอยู่บ้างตามความเร็วอินเตอร์เน็ตที่ใช้ในการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทดลองของระบบไฮโดรโปนิกส์สามารถสรุปได้ว่า

1.) ระบบอัตโนมัติซึ่งเป็นการควบคุมปัจจัยต่างๆ ที่มีผลต่อการเจริญเติบโตของพืชนั้นต้องมีการควบคุมค่าต่างๆให้เหมาะสม ได้แก่ การควบคุมการปล่อยกรดเบส ควรค่อยๆ ปล่อยสารในระบบหยด เพราะจะทำให้น้ำในถังค่อยๆปรับ pH ไปเรื่อยๆ จนอยู่ในช่วงที่ต้องการ หากปล่อยสารเร็วเกินไปจะทำให้การควบคุม pH ทำได้ยาก, การควบคุมอุณหภูมิไม่ควรเกิน 29 องศา เพราะจะทำให้พืชใบเหี่ยวเนื่องจากมีการคายน้ำมากเกินไป และทำให้ออกซิเจนในน้ำลดลงด้วย, ควรมีการเว้นระยะการสเปรย์ของน้ำให้เหมาะสม ดังนั้นจากการทดลองพบว่าความถี่ที่เหมาะสมคือทำการสเปรย์ทุกๆ 15 นาที เป็นเวลา 15 วินาที, นอกจากนั้นแล้ว ยังมีการนำ LED Grow Light มาใช้ในการทดลอง เพื่อทดแทนการใช้แสงจากแสงจากดวงอาทิตย์ จากผลการทดลองพบว่าแสงที่ได้จากหลอด LED นั้น ยังไม่เพียงพอเนื่องจากมีความเข้มแสง น้อยเกินไป ดังนั้นจึงมีการเพิ่มจำนวนหลอด LED เข้าไป

2.) ระบบปรับค่าด้วยมือ ซึ่งนำมาเป็นอีกทางเลือกหนึ่งที่นอกเหนือจากการควบคุมอัตโนมัติ ซึ่งสามารถปรับผ่าน LCD Touch Screen ได้แก่ การ ให้สเปรย์น้ำโดยผู้ใช้เอง โดยที่ไม่ต้องรอให้ถึงเวลาที่ตั้งหรืออุณหภูมิที่กำหนด และยังสามารถเลือกปล่อยกรดเบสตามความต้องการได้ และยังปรับผ่านหน้าจอสมาร์ทโฟน ได้แก่ สามารถสั่งการเปิดปิดของพัดลมระบายความร้อนได้ และยังสามารถเลือกปล่อยกรดเบสตามความต้องการได้ด้วย

3.) ระบบคาริเบทพีเอชเซนเซอร์ เพื่อให้อุปกรณ์สามารถอ่านค่าความเป็นกรดเบสได้ถูกต้อง จากการทดลองพบว่ายังมีความคลาดเคลื่อนอยู่เล็กน้อย เมื่อเทียบกับ pH meter สาเหตุเนื่องจากแช่หัวโพรบไว้ในสารละลายเป็นเวลานาน ทำให้เกิดคราบสกปรกเกาะติดอยู่ เป็นเหตุให้ค่าที่วัดนั้นคลาดเคลื่อนหรือไม่นิ่งแต่เกิดขึ้นปริมาณที่ยอมรับได้

5.2 อุปสรรคและปัญหา

จากการทำโครงงาน ผู้จัดทำพบปัญหาหลายประการ ได้แก่ อุปกรณ์บางชิ้นมีการชำรุดขณะทำการทดลองเนื่องจากการโดนกรดกัดกร่อน, มีความล่าช้าเกิดขึ้นในการควบคุมการสั่งงานผ่านทางสมาร์ทโฟนซึ่งขึ้นอยู่กับความเร็วของอินเทอร์เน็ตที่เชื่อมต่อ, รวมถึง การที่สวิทช์ลูกลอยทำงานได้ไม่สมบูรณ์ จึงต้องทำการเปลี่ยนและจัดวางตำแหน่งให้เหมาะสมใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] https://www.dfrobot.com/wiki/index.php/PH_meter(SKU:_SEN0161)

[2] http://www.arduinoall.net/arduino-tutor/courses/arduino-esp8266-%E0%B8%AA%E0%B8%AD%E0%B8%99

[3] http://www.instructables.com/id/Digital-Thermometer-With-Arduino-DS1307/ALLSTEPS

[4] http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app

[5] http://www.myhomeveg.com/index.php?lay=show&ac=article&Id=405383

[6] http://arduinotronics.blogspot.com/2010/12/two-ds18b20-temp-sensors-on-lcddisplay.html

[7] http://engineeringlearning.blogspot.com/2013/10/interfacing-lcd-withoutpotentiometer.html

[8] http://hydroponicscool.blogspot.com/2012/05/hydroponics_4901.html

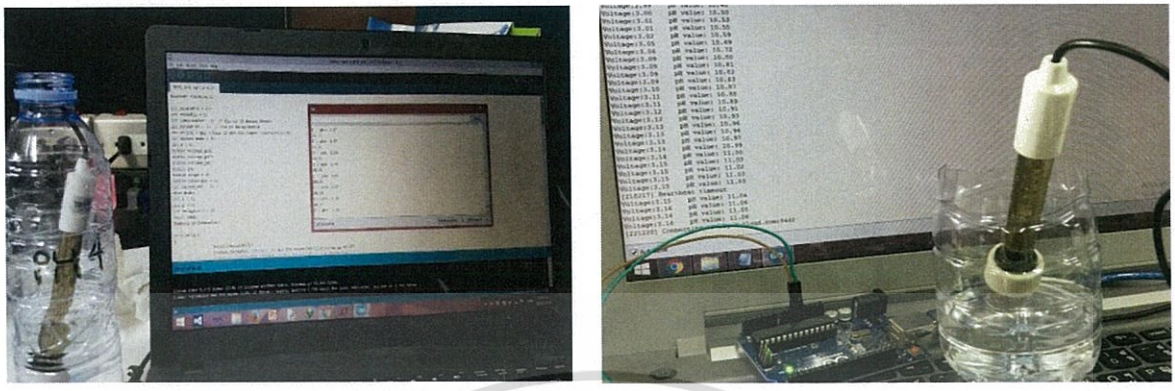
[9] https://www.ysi.com/ysi-blog/water-blogged-blog/2013/05/what-is-affecting-yourdissolved-oxygen-measurements-part-1-of-4

[10] https://www.cpe.ku.ac.th/~yuen/204471/power/switching_regulator/

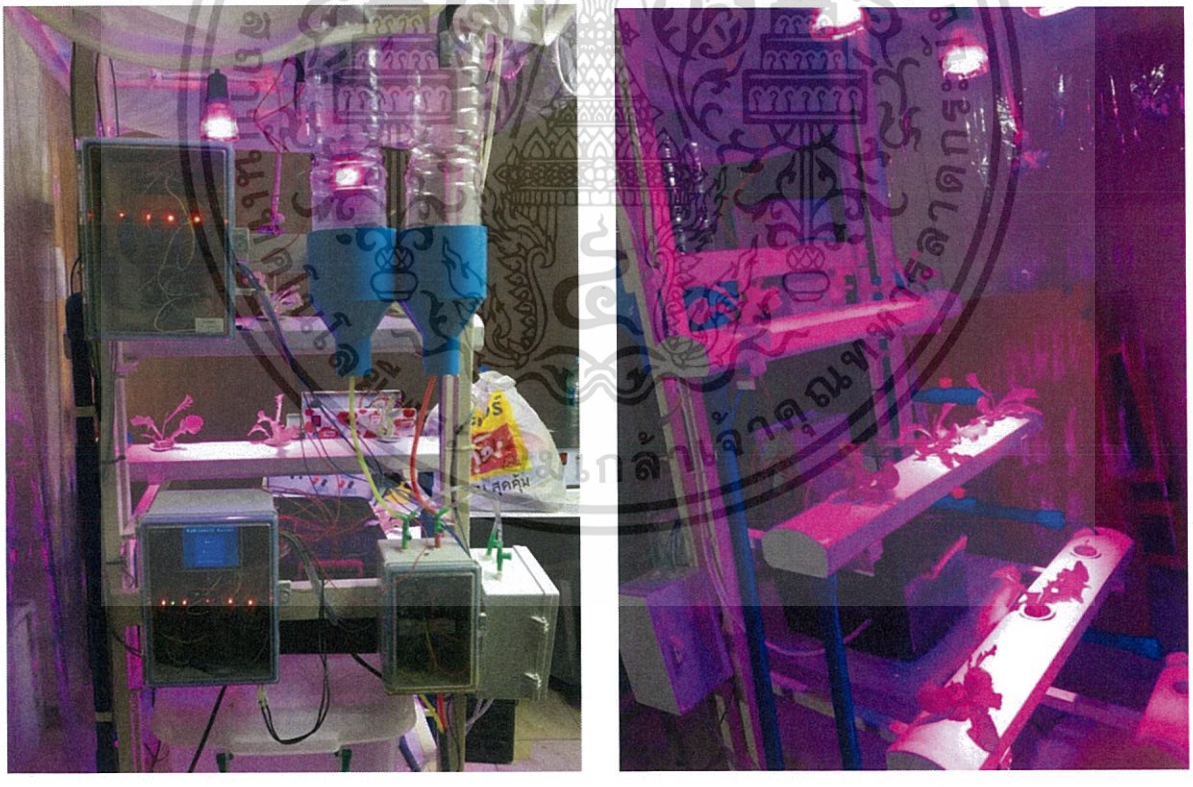
[11] https://goo.gl/av24yD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก



การทดลองคาริเบทค่าความเป็นกรดเบส



เครื่องปลูกผักในระบบไฮโดรโปนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โค้ดบนโปรแกรม Arduino

การสั่งการผ่านหน้าจอ LCD Touch Screen

```
#include <OneWire.h>
#include <URTouch.h>
#include <UTFT.h>
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];
UTFT myGLCD(ILI9341_16,38,39,40,41);
URTouch myTouch( 6, 5, 4, 3, 2);
int TempPin = A9;
int SprayPin = 8;
int floatswitchPin = 9;
int pumpPin = 10;
int solenoidValvePin1 = 11;
int solenoidValvePin2 = 12;
int ledPin = 13;
int irmotionPin = 4; // Pin of IR Motion
Sensor
int relayPin7 = 7; // Pin of Relay
Module
int ph_pin = A8; //This is the pin
number connected to Po
int switch_mode;
int k = 0;
int waterLevel;
double voltage_ph4 = 1.19;
double voltage_ph7 = 1.93;
double voltage_ph;
double ph;
double slope = (voltage_ph7 -
voltage_ph4)/(6.86-4.01);
double intercept = voltage_ph4-
(slope*(4.01));
double phStore = 0;
int count = 0;
int switch_cal = 5;
char mode;
int a = 0;
int b = 0;
float temp;
char page = 'h';
OneWire ds(TempPin);
int x,y;
String str_x,str_y;
String strTemp;
String strPH;
String strPHPH;
String strwaterLevel;
String str_voltage_ph4;
String str_voltage_ph7;
String str_slope;
String str_intercept;
int measure_ph4;
int measure_ph7;
int measure_ph;
bool ph4_calibrated = false;
bool ph7_calibrated = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned long _micro;
int spray_Hour = 0;
int LED_StartCommand = 1;
int LED_StopCommand = 2;

int LED_DelayCommand = 3;
int LED_Command;
int LED_NextCommand;
unsigned long LED_timeStart;
bool LED_DelayOneHour = false;
int LED_hourTotal = 0;
int LED_hourDesire;
int LED_StartCommandHour = 6;
int LED_StopCommandHour = 1;
static unsigned long OneSecond =
1000000;// 1 second
void Counter(bool* Flag, int
secondNum, unsigned long time_start)
{
unsigned long microSecondTotal =
OneSecond * secondNum;
if ( (_micro = micros()) - time_start >
microSecondTotal )
{
*Flag = true;
}
}
void LED()
{
if (LED_Command ==
LED_DelayCommand)
{
LED_DelayOneHour = false;
Counter(&LED_DelayOneHour,
3600, LED_timeStart);
if (LED_DelayOneHour == true)
{
LED_DelayOneHour = false;
LED_hourTotal++;
if (LED_hourTotal <
LED_hourDesire)
{
LED_timeStart = micros();
}
else
{
LED_Command =
LED_NextCommand;
LED_hourTotal = 0;
}
}
}
if (LED_Command ==
LED_StopCommand)
{
digitalWrite(ledPin, LOW);
LED_Command =
LED_DelayCommand;
LED_NextCommand =
LED_StartCommand;
LED_hourDesire =
LED_StopCommandHour;
LED_timeStart = micros();
}
if (LED_Command ==
LED_StartCommand)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
digitalWrite(ledPin, HIGH);
LED_Command =
LED_DelayCommand;

LED_NextCommand =
LED_StopCommand;
LED_hourDesire =
LED_StartCommandHour;
LED_timeStart = micros();
}
}
void HomeScreen()
{
myGLCD.clrScr();
myGLCD.setFont(BigFont);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Hydroponic System",
CENTER, 10);\
myGLCD.setBackColor(0, 0, 255);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 60, 235,
110);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (80, 60, 235,
110);
myGLCD.print("Auto", CENTER, 80);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 120, 235,
170);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (80, 120,
235, 170);
myGLCD.print("Manual", CENTER, 140);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 180, 235,
230);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (80, 180,
235, 230);
myGLCD.print("Calibrate", CENTER,
200);
myGLCD.setBackColor(0,0,0);
x = 0;
y = 0;
while (true)
{
LED();
if (myTouch.dataAvailable())
{
myTouch.read();
x=myTouch.getX();
y=myTouch.getY();
str_x = String(x);
str_y = String(y);
if ( (x!=-1) && (y!=-1) )
{
if ((x>=80) && (x<=235)) //
Upper row
{
if ((y>=60) && (y<=110)) //
Button: 1
{
waitForIt(80, 60, 235, 110);
page = 'a';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
break;
}
if ((y>=120) && (y<=170)) //
Button: 2
{
waitForIt(80, 120, 235, 170);
page = 'm';
break;
}
if ((y>=180) && (y<=230)) //
Button: 3
{
waitForIt(80, 180, 235, 230);
page = 'c';
break;
}
}
}
}
}
}
void AutoScreen()
{
myGLCD.clrScr();
myGLCD.setFont(BigFont);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Choose Plant :",
CENTER, 10);
myGLCD.setBackColor(0, 0, 255);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 60, 235,
110);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (80, 60, 235,
110);
myGLCD.print("Salad", CENTER, 80);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 120, 235,
170);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (80, 120,
235, 170);
myGLCD.print("Cabbage", CENTER,
140);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 180, 235,
230);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (80, 180,
235, 230);
myGLCD.print("Rosemary", CENTER,
200);
myGLCD.setBackColor (0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("<", 10, 140);
x = 0;
y = 0;
while (true)
{
LED();
if (myTouch.dataAvailable())
{
myTouch.read();
x=myTouch.getX();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
y=myTouch.getY();
str_x = String(x);
str_y = String(y);
if ( (x!=-1) && (y!=-1) )
{

if ((x>=80) && (x<=235)) //
Upper row
{
if ((y>=60) && (y<=110)) //
Button: 1
{
waitForIt(80, 60, 235, 110);
page = 's';
break;
}
if ((y>=120) && (y<=170)) //
Button: 2
{
waitForIt(80, 120, 235, 170);
page = 'b';
break;
}
if ((y>=180) && (y<=230)) //
Button: 3
{
waitForIt(80, 180, 235, 230);
page = 'l';
break;
}
}
if ((x>=0) && (x<=40) &&
(y>=120) && (y<=170))
{
waitForIt(0, 120, 40, 170);
page = 'h';
break;
}
}
}
}
}
void PlantScreen(String ScreenTitle,
char ScreenChar)
{
myGLCD.clrScr();
myGLCD.setFont(BigFont);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print(ScreenTitle, CENTER,
10);
myGLCD.setFont(SmallFont);
myGLCD.print("Temperature :", 80,
80);
myGLCD.print("pH :", 80, 140);
myGLCD.print("Water Level :", 80,
200);
myGLCD.setFont(BigFont);
myGLCD.print("<", 10, 140);
myGLCD.print(".",275,65);
myGLCD.print("C",280,80);
x = 0;
y = 0;
phStore = 0;
count = 0;
double ph_high;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
double ph_low;
double phMean;
double phStoreFirst;
double phStoreSecond;
int phNum = 40;
bool firstMean = true;

if (ScreenChar == 's')
{
ph_high = 6.5;
ph_low = 5.5;
}
else if (ScreenChar == 'b')
{
ph_high = 7;
ph_low = 6.5;
}
else if (ScreenChar == 'l')
{
ph_high = 7.8;
ph_low = 6.6;
}
unsigned long spray_timeStart;
int spray_Command;
int spray_StartCommand = 1;
int spray_StopCommand = 2;
int spray_DelayCommand = 3;
spray_Command =
spray_StartCommand;
int spray_NextCommand;
int spray_hourDesire;
int spray_secondDesire;
bool spray_DelayComplete = false;
int spray_hourTotal = 0;
while (true)
{
LED();
temp = getTemp();
if (temp<50)
{
strTemp = String(temp);
myGLCD.print(strTemp,190,80);
}
if (spray_Command ==
spray_DelayCommand)
{
spray_DelayComplete = false;
if (spray_hourDesire > 0)
{
Counter(&spray_DelayComplete,
3600, spray_timeStart);
if (spray_DelayComplete ==
true)
{
spray_DelayComplete =
false;
spray_hourTotal++;
if (spray_hourTotal <
spray_hourDesire)
{
spray_timeStart =
micros();
}
else
{
spray_Command =
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
spray_NextCommand;
spray_hourTotal = 0;
}
}
}
else
{

Counter(&spray_DelayComplete,
spray_secondDesire, spray_timeStart);
if (spray_DelayComplete ==
true)
{
spray_DelayComplete =
false;
spray_Command =
spray_NextCommand;
}
}
}
if (spray_Command ==
spray_StopCommand)
{
digitalWrite(SprayPin, LOW);
spray_Command =
spray_DelayCommand;
spray_NextCommand =
spray_StartCommand;
spray_hourDesire = 0;
spray_secondDesire = 900;
spray_timeStart = micros();
}
if (spray_Command ==
spray_StartCommand)
{
if (temp<50)
{
if (temp>29.0)
{
digitalWrite(SprayPin, HIGH);
spray_Command =
spray_DelayCommand;
spray_NextCommand =
spray_StopCommand;
spray_hourDesire = 0;
spray_secondDesire = 15;
spray_timeStart = micros();
}
else
{
digitalWrite(SprayPin, LOW);
}
}
}
measure_ph =
analogRead(ph_pin);
voltage_ph = 5 / 1024.0 *
measure_ph;
ph = (voltage_ph -
intercept)/slope;
phStore = phStore + ph;
count++;
if (count == phNum)
{
phMean = phStore/phNum;
strPHPH = String(phMean,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
myGLCD.print(strPHPH,120,140);
if (phMean > ph_high)
{
digitalWrite(solenoidValvePin2,HIGH);
digitalWrite(solenoidValvePin1,LOW);
}
else if (phMean < ph_low)

{
digitalWrite(solenoidValvePin1,HIGH);
digitalWrite(solenoidValvePin2,LOW);
}
else
{
digitalWrite(solenoidValvePin1,LOW);
digitalWrite(solenoidValvePin2,LOW);
}
phStore = 0;
count = 0;
}
waterLevel =
digitalRead(floatswitchPin);
if (waterLevel == 0)
{
myGLCD.print("low ",190,200);
digitalWrite(pumpPin,HIGH);
}
else
{
myGLCD.print("normal",190,200);
digitalWrite(pumpPin,LOW);
}
if (myTouch.dataAvailable())
{
myTouch.read();
x=myTouch.getX();
y=myTouch.getY();
str_x = String(x);
str_y = String(y);
if ( (x!=-1) && (y!=-1) )
{
if ((x>=0) && (x<=40) &&
(y>=120) && (y<=170))
{
waitForIt(0, 120, 40, 170);
digitalWrite(SprayPin, LOW);
page = 'a';
break;
}
}
}
}
void ManualScreen()
{
myGLCD.clrScr();
myGLCD.setFont(BigFont);
myGLCD.setBackColor(0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Manual Mode",
CENTER, 10);
myGLCD.setFont(SmallFont);
myGLCD.print("Temperature :", 210,
65);
myGLCD.print("pH :", 250, 130);
myGLCD.setFont(BigFont);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
myGLCD.print(".",290,75);
myGLCD.print("C",295,90);
myGLCD.setBackColor(0, 0, 255);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 60, 180,
110);
myGLCD.setColor(255, 255, 255);

myGLCD.drawRoundRect (80, 60, 180,
110);
myGLCD.print("Spray", 90, 80);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 120, 180,
170);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (80, 120,
180, 170);
myGLCD.print("Acid", 100, 140);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (80, 180, 180,
230);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (80, 180,
180, 230);
myGLCD.print("Base", 100, 200);
myGLCD.setBackColor (0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("<", 10, 140);
x = 0;
y = 0;
phStore = 0;
count = 0;
bool SprayButtonPressed = false;
bool SolenoidValve1Pressed = false;
bool SolenoidValve2Pressed = false;
while (true)
{
LED();
temp = getTemp();
strTemp = String(temp);
myGLCD.print(strTemp,210,90);
measure_ph =
analogRead(ph_pin);
voltage_ph = 5 / 1024.0 *
measure_ph;
ph = (voltage_ph -
intercept)/slope;
phStore = phStore + ph;
count++;
if (count == 40)
{
strPH = String(phStore/40);
myGLCD.print(strPH,230,150);
phStore = 0;
count = 0;
}
if (myTouch.dataAvailable())
{
myTouch.read();
x=myTouch.getX();
y=myTouch.getY();
str_x = String(x);
str_y = String(y);
if ( (x!=-1) && (y!=-1) )
{
if ((x>=80) && (x<=180)) //
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Upper row
{
if ((y>=60) && (y<=110)) //
Button: 1
{
waitForIt(80, 60, 180, 110);
if (SprayButtonPressed ==
false)

{
digitalWrite(SprayPin,HIGH);
SprayButtonPressed =
true;
}
else
{
digitalWrite(SprayPin,LOW);
SprayButtonPressed =
false;
}
}
if ((y>=120) && (y<=170)) //
Button: 2
{
waitForIt(80, 120, 180, 170);
if (SolenoidValve2Pressed
== false)
{
digitalWrite(solenoidValvePin2,HIGH);
SolenoidValve2Pressed =
true;
}
else
{
digitalWrite(solenoidValvePin2,LOW);
SolenoidValve2Pressed =
false;
}
}
if ((y>=180) && (y<=230)) //
Button: 3
{
myGLCD.print("3",RIGHT,200);
waitForIt(80, 180, 180, 230);
if (SolenoidValve1Pressed
== false)
{
digitalWrite(solenoidValvePin1,HIGH);
SolenoidValve1Pressed =
true;
}
else
{
digitalWrite(solenoidValvePin1,LOW);
SolenoidValve1Pressed =
false;
}
}
}
if ((x>=0) && (x<=40) &&
(y>=120) && (y<=170))
{
waitForIt(0, 120, 40, 170);
page = 'h';
break;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
}
}
}
void CalibrateScreen()
{
myGLCD.clrScr();
myGLCD.setFont(BigFont);
myGLCD.setBackColor(0, 0, 0);

myGLCD.setColor(255, 255, 255);
myGLCD.print("Calibration", CENTER,
10);
myGLCD.print("Acid",73,80);
myGLCD.print("Base",73,200);
myGLCD.setBackColor(0, 0, 255);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (160, 60, 290,
110);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (160, 60,
290, 110);
myGLCD.print("Record", 180, 80);
myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (160, 180, 290,
230);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (160, 180,
290, 230);
myGLCD.print("Record", 180, 200);
myGLCD.setBackColor (0, 0, 0);
myGLCD.setColor(255, 255, 255);
myGLCD.print("<", 10, 140);
if ((ph4_calibrated == true) &&
(ph7_calibrated == true))
{
str_voltage_ph4 =
String(voltage_ph4);
str_voltage_ph7 =
String(voltage_ph7);
str_slope = String(slope);
str_intercept = String(intercept);
myGLCD.setFont(SmallFont);
myGLCD.print(str_voltage_ph4,90,95);
myGLCD.print(str_voltage_ph7,90,215);
myGLCD.print("V",127,95);
myGLCD.print("V",127,215);
myGLCD.print("slope :",75,130);
myGLCD.print(str_slope,140,140);
myGLCD.print("intercept :",75,150);
myGLCD.print(str_intercept,150,150);
myGLCD.setFont(BigFont);
x = 0;
y = 0;
while (true)
{
LED();
if (myTouch.dataAvailable())
{
myTouch.read();
x=myTouch.getX();
y=myTouch.getY();
str_x = String(x);
str_y = String(y);
if ( (x!=-1) && (y!=-1) )
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ((x>=160) && (x<=290))
{
if ((y>=60) && (y<=110))
{
waitForIt(160, 60, 290, 110);
measure_ph4 = analogRead(ph_pin);
voltage_ph4 = 5 / 1024.0 *
measure_ph4;
ph4_calibrated = true;
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (90, 95, 120,
105);
myGLCD.setColor(255,255,255);
myGLCD.setFont(SmallFont);
str_voltage_ph4 = String(voltage_ph4);
myGLCD.print(str_voltage_ph4,90,95);
myGLCD.print("V",127,95);
myGLCD.setFont(BigFont);
}
if ((y>=180) && (y<=230))
{
waitForIt(160, 180, 290,
230);
measure_ph7 =
analogRead(ph_pin);
voltage_ph7 = 5 / 1024.0 *
measure_ph7;
ph7_calibrated = true;
myGLCD.setColor(0, 0, 0);
myGLCD.fillRoundRect (90,
215, 120, 225);
myGLCD.setColor(255,255,255);
myGLCD.setFont(SmallFont);
str_voltage_ph7 =
String(voltage_ph7);
myGLCD.print(str_voltage_ph7,90,215);
myGLCD.print("V",127,215);
myGLCD.setFont(BigFont);
}
}
if ((ph4_calibrated == true) &&
(ph7_calibrated == true))
{
slope = (voltage_ph7 -
voltage_ph4)/(6.86-4.01);
intercept = voltage_ph4-
(slope*(4.01));
myGLCD.setColor(0,0,0);
myGLCD.fillRoundRect (71, 138, 307,
154);
myGLCD.setColor(255,255,255);
myGLCD.setFont(SmallFont);
str_slope = String(slope);
str_intercept = String(intercept);
myGLCD.print("slope :",75,140);
myGLCD.print(str_slope,140,140);
myGLCD.print(",",173,140);
myGLCD.print("intercept :",180,140);
myGLCD.print(str_intercept,275,140);
myGLCD.setFont(BigFont);
}
if ((x>=0) && (x<=40) &&
(y>=120) && (y<=170))
{
waitForIt(0, 120, 40, 170);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
page = 'h';
break;
}
}
}
}
}
void waitForIt(int x1, int y1, int x2, int
y2)
{
myGLCD.setColor(255, 0, 0);
myGLCD.drawRoundRect (x1, y1, x2,
y2);
while (myTouch.dataAvailable())
myTouch.read();
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (x1, y1, x2,
y2);
}
void setup()
{
myGLCD.InitLCD();
myGLCD.clrScr();
myTouch.InitTouch();
myTouch.setPrecision(PREC_MEDIUM);
myGLCD.setFont(BigFont);
myGLCD.setBackColor(0,
0, 0);
pinMode(SprayPin,
OUTPUT);
digitalWrite(SprayPin,
LOW);
pinMode(floatswitchPin,INPUT_PULLUP
);
pinMode(pumpPin,OUTPUT);
digitalWrite(pumpPin,LOW);
pinMode(solenoidValvePin1,OUTPUT);
digitalWrite(solenoidValvePin1,LOW);
pinMode(solenoidValvePin2,OUTPUT);
digitalWrite(solenoidValvePin2,LOW);
pinMode(ledPin,OUTPUT);
digitalWrite(ledPin,HIGH);
ph4_calibrated = false;
ph7_calibrated = false;
LED_Command = LED_StartCommand;
page = 'h';
while(true)
{
if (page == 'h')
{
HomeScreen();
}
if (page == 'a')
{
AutoScreen();
}
if (page == 's')
{
PlantScreen("Salad",'s'); //Salad
}
if (page == 'b')
{
PlantScreen("Cabbage",'b'); //Cabbage
}
if (page == 'l')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
PlantScreen("Rosemary",'l'); //Rosemary
}
if (page == 'm')
{
ManualScreen();
}
if (page == 'c')
{
CalibrateScreen();
}

}
Serial.begin(9600);
}
float getTemp()
{
byte data[12];
byte addr[8];
if ( !ds.search(addr))
{
ds.reset_search();
return -1;
}
if ( OneWire::crc8( addr, 7)
!= addr[7])
{
Serial.println("CRC is
not valid!");
return -1;
}
if ( addr[0] != 0x10 &&
addr[0] != 0x28)
{
Serial.print("Device is
not recognized");
return -1;
}
ds.reset();
ds.select(addr);
ds.write(0x44,1);
byte present = ds.reset();
ds.select(addr);
ds.write(0xBE);
for (int i = 0; i < 9; i++)
{
data[i] = ds.read();
}
ds.reset_search();
byte MSB = data[1];
byte LSB = data[0];
float TRead = ((MSB <<8) |
LSB);
float Temperature =
TRead / 16;
return Temperature;
}
void loop()
{
temp = getTemp();
Serial.print(temp);
Serial.print(" , ");
if (temp>29.0)
{digitalWrite(solenoidValvePin1,
LOW);}
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{digitalWrite(solenoidValvePin1,
HIGH);}
waterLevel =
digitalRead(floatswitchPin);
Serial.println(waterLevel);
if (waterLevel == 0)
{
digitalWrite(pumpPin,LOW);
}
else
```

ระบบเปิด-ปิดไฟอัตโนมัติ

```
void setup() {
Serial.begin(9600);
pinMode(7,OUTPUT);
digitalWrite(7,HIGH);
delay(1000);
digitalWrite(7,LOW);
}
void loop()
{
int sensorValue = analogRead(A4);
Serial.println(sensorValue);
delay(100);
if(sensorValue > 400-){
digitalWrite(7,HIGH);
Serial.println("LED  ON");
}
else
{
digitalWrite(7,LOW);
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสั่งการผ่านสมาร์ทโฟน

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SimpleTimer.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN 2
#define DHTTYPE DHT22
#define SensorPin A0 //pH meter Analog output to Arduino Analog Input 0
#define Offset 0.00 //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth 40 //times of collection
int pHArray[ArrayLenth]; //Store the average value of the sensor feedback
int pHArrayIndex=0;
DHT_Unified dht(DHTPIN, DHTTYPE);
char auth[] = "f54d3719f28c475382af9b40054dd838";
SimpleTimer timer;
float voltage=0;
float pHValue=0;
void setup()
{
Serial.begin(9600);
Blynk.begin(auth, "Bosziio", "14121412");
dht.begin();
// Setup a function to be called every second
timer.setInterval(1000L, sendUptime);
{
pinMode(LED,OUTPUT);
Serial.begin(9600);
Serial.println("pH meter experiment!");
//Test the serial monitor
}
}
void sendUptime()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Blynk.virtualWrite(V5, millis() / 1000);
sensors_event_t event;
dht.temperature().getEvent(&event);
if (!isnan(event.temperature)) {
Blynk.virtualWrite(V1, event.temperature);
}
dht.humidity().getEvent(&event);
if (!isnan(event.relative_humidity))
{
Blynk.virtualWrite(V2,
event.relative_humidity);
}
}
void loop()
{
Blynk.run();
timer.run();
int sensorValue = analogRead(A0);
float volt = (sensorValue/1024.0) * 5;
float ph = volt*3.5 ;
static unsigned long samplingTime =
millis();
static unsigned long printTime = millis();
//static float pHValue,voltage;
 if(millis()-samplingTime >
samplingInterval)
 {
pHArray[pHArrayIndex++]=analogRead(Sen
sorPin);
if(pHArrayIndex==ArrayLenth)pHArrayIndex
=0;
voltage = avergearray(pHArray,
ArrayLenth)*5.0/1024;
pHValue = 3.5*voltage+Offset;
samplingTime=millis();
 }
 if(millis() - printTime > printInterval)
 //Every 800 milliseconds, print a
 numerical, convert the state of the LED
 indicator
 {
 Serial.print("Voltage:");
 Serial.print(voltage,2);
 Serial.print("    pH value: ");
 Serial.println(pHValue,2);
 digitalWrite(LED,digitalRead(LED)^1);
 printTime=millis();
 }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
double avergearray(int* arr, int number){
int i;
int max,min;
double avg;
long amount=0;
if(number<=0){
Serial.println("Error number for the array to
avraging!/n");
return 0;
 }
if(number<5){   //less than 5, calculated
directly statistics
for(i=0;i<number;i++){
amount+=arr[i];
}
avg = amount/number;
return avg;
}
else
{
if(arr[0]<arr[1]){
min = arr[0];max=arr[1];
}
else
{
min=arr[1];max=arr[0];
}
for(i=2;i<number;i++){
 if(arr[i]<min){
 amount+=min;        //arr<min
 min=arr[i];
 }
else
 {
  if(arr[i]>max){
  amount+=max;    //arr>max
  max=arr[i];
  }
 else
 {
amount+=arr[i]; //min<=arr<=max
 }
 } //if
  } //for
  avg = (double)amount/(number-2);
  } //if
  return avg;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

BLYNK_READ(V6) // Widget in the app
READs Virtal Pin V5 with the certain
frequency

{
Blynk.virtualWrite(V6,pHValue);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

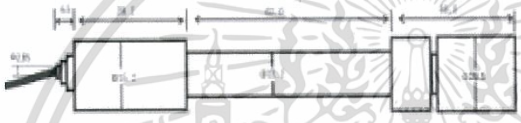
ภาคผนวก ค

pH Sensor Datasheet

Introduction

Need to measure water quality and other parameters but haven't got any low cost pH meter? Find it difficult to use with Arduino? Here comes an analog pH meter, specially designed for Arduino controllers and has built-in simple, convenient and practical connection and features. It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface. You can just connect the pH sensor with BNC connector, and plug the PH2.0 interface into any analog input on Arduino controller to read pH value easily.

Specification



SEN0161 dimension

- Module Power: 5.00V
- Circuit Board Size: 43mm×32mm
- pH Measuring Range: 0-14
- Measuring Temperature: 0-60 ℃
- Accuracy: ± 0.1pH (25 ℃)
- Response Time: ≤ 1min
- pH Sensor with BNC Connector
- PH2.0 Interface (3 foot patch)
- Gain Adjustment Potentiometer
- Power Indicator LED

Precautions

- Before and after use of the pH electrode every time, you need to use (pure)water to clean it.
- The electrode plug should be kept clean and dry in case of short circuit.
- **Preservation**: Electrode reference preservation solution is the **3N KCL** solution.
- Measurement should be avoided staggered pollution between solutions, so as not to affect the accuracy of measurement.
- Electrode blub or sand core is defiled which will make PTS decline, slow response. So, it should be based on the characteristics of the pollutant, adapted to the cleaning solution, the electrode performance recovery.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Electrode when in use, the ceramic sand core and liquid outlet rubber ring should be removed, in order to make salt bridge solution to maintain a certain velocity.

NOTE: Differences between the probes, SEN0161 and SEN0169

Their usages/ specifications are almost the same. The differences locates at

Long-firing Operation: SEN0169 supports, while SEN0161 NOT, i.e. you can not immerse SEN0161 in water for Continuous Testing.

Life Span: In 25 ℃, pure water, do Continuous Testing with them both, SEN0169 can work two years, while SEN0161 can only last for 6 months. And just for reference, if put them in turbid, strongly acid and alkali solution, 25℃, the life span would drop to one year (SEN0169), 1 month(or shorter, SEN0161).
Tempreture, pH, turbidity of the water effect the probe life span a lot.

Waterproof: You can immerse the whole probe SEN0169 into the water, while you can only immerse the front part of the probe SEN0161, the electrode glass bulb, into water, the rear part, from the white shell to the cable, MUST NOT be under water.

Strongly Acid and Alkali: SEN0169 are preferred for strongly acid and alkali test. And if your testing range is usually within pH6~8, then SEN0161 is capable for that.

pH Electrode Characteristics

The output of pH electrode is Millivolts, and the pH value of the relationship is shown as follows (25 ℃):

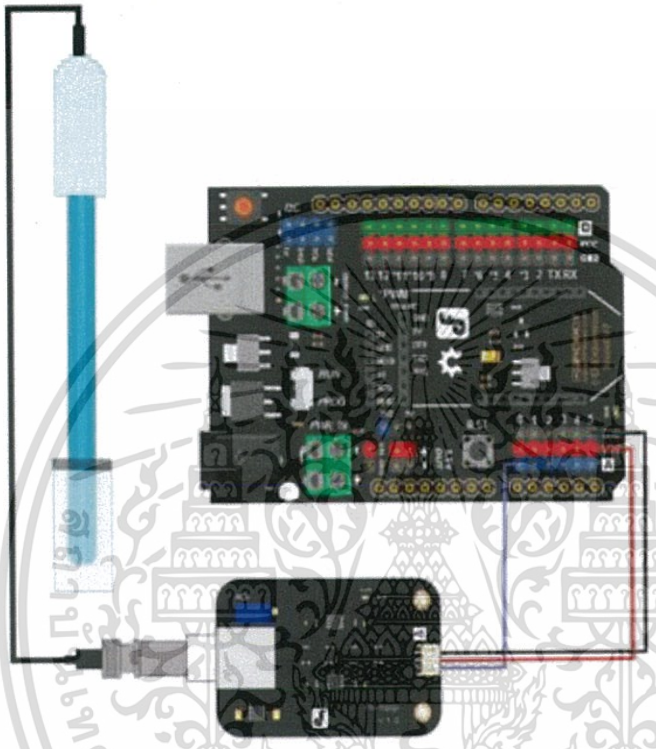
VOLTAGE （mV）	pH value	VOLTAGE （mV）	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTE: It is normal that if your reading is much different with the table since you are not reading from the electrode directly but from the voltage adapter, it has converted the original voltage (-5V ~ +5V) to Arduino compatible voltage, i.e. 0 ~ 5V. See the discussion on Forum.

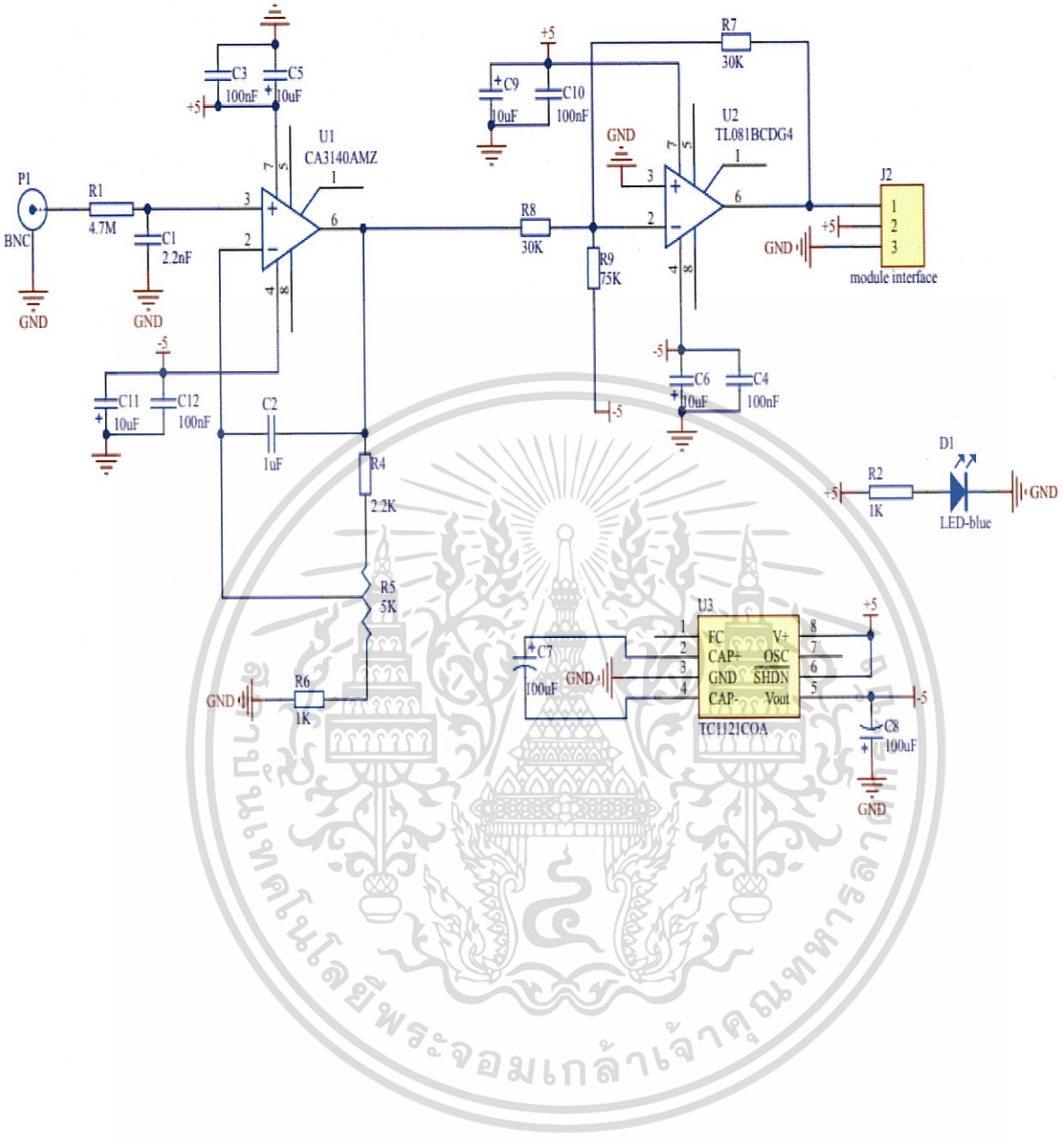
Usage

Connecting Diagram



NOTE:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temperature Sensor Datasheet

DALLAS MAXIM

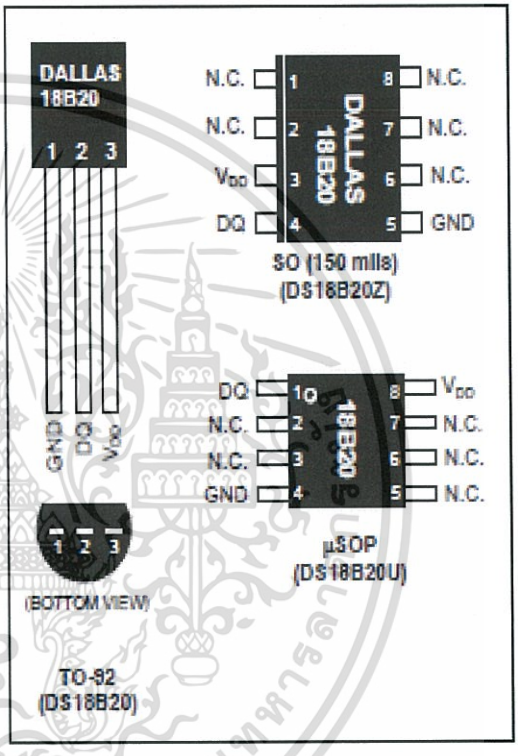
DS18B20
Programmable Resolution 1-Wire Digital Thermometer

www.maxim-ic.com

FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS



DESCRIPTION

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDERING INFORMATION

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 µSOP	18B20
DS18B20U+	-55°C to +125°C	8 µSOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 µSOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 µSOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

+Denotes a lead-free package. A "+" will appear on the top mark of lead-free packages.
T&R = Tape and reel.
**TO-92 packages in tape and reel can be ordered with straight or formed leads. Choose "SL" for straight leads. Bulk TO-92 orders are straight leads only.*

PIN DESCRIPTION

PIN			NAME	FUNCTION
SO	µSOP	TO-92		
1, 2, 6, 7, 8	2, 3, 5, 6, 7	—	N.C.	No Connection
3	8	3	V_{DD}	Optional V_{DD}. V_{DD} must be grounded for operation in parasite power mode.
4	1	2	DQ	Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the *Powering the DS18B20* section.)
5	4	1	GND	Ground

OVERVIEW

Figure 1 shows a block diagram of the DS18B20, and pin descriptions are given in the *Pin Description* table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The T_H, T_L, and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Solenoid Valve Datasheet

ASCO®

Engineering Information
Solenoid Valves

Solenoid Valves

Principles of Operation

A solenoid valve is a combination of two basic functional units:

- A solenoid (electromagnet) with its core
- A valve body containing one or more orifices

Flow through an orifice is shut off or allowed by the movement of the core when the solenoid is energized or de-energized. ASCO valves have a solenoid mounted directly on the valve body. The core is enclosed in a sealed tube, providing a compact, leaktight assembly.

Direct Acting Valves (Figures 1A, 1B)

When the solenoid is energized in a direct acting valve, the core directly opens the orifice of a Normally Closed valve or closes the orifice of a Normally Open valve. When de-energized, a spring returns the valve to its original position. The valve will operate at pressures from 0 psi to its rated maximum.

The force needed to open the valve is proportional to the orifice size and fluid pressure. As the orifice size increases, so does the force required. To open large orifices while keeping solenoid size small, a Pilot Operated construction is used.

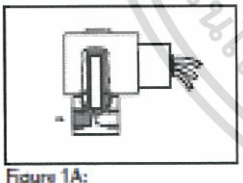


Figure 1A:
Direct Acting,
Normally Closed Valve,
De-Energized

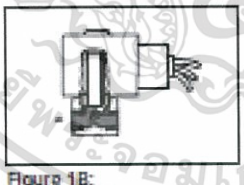


Figure 1B:
Direct Acting,
Normally Closed Valve,
Energized

Internal Pilot Operated Valves (Figures 2A, 2B)

Normally, these valves have a pilot and bleed orifice which enable them to use line pressure for operation.

When the solenoid is de-energized, the pilot orifice is closed and full line pressure is applied to the top of the piston or diaphragm through the bleed orifice, providing seating force for tight closure.

When the solenoid is energized, the core opens the pilot orifice, relieving pressure from the top of the piston or diaphragm via the outlet side of the valve. The line pressure then opens the valve by lifting the diaphragm or piston off the main orifice.

Two constructions are available for 2-way valves:

- Floating diaphragm or piston which requires a minimum pressure drop across the valve to remain in the open position (Figures 2A, 2B).
- Hung-type diaphragm or piston held open mechanically by the solenoid core. The valve opens and remains open with zero pressure drop (Figures 3A, 3B).

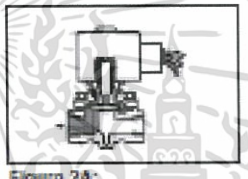


Figure 2A:
Pilot Operated, Normally Closed Valve, De-Energized

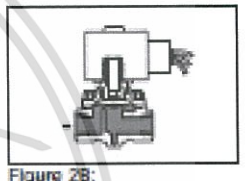


Figure 2B:
Pilot Operated, Normally Closed Valve, Energized

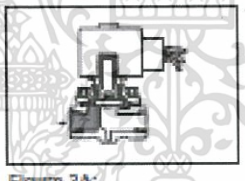


Figure 3A:
Pilot Operated, Normally Closed Valve, De-Energized

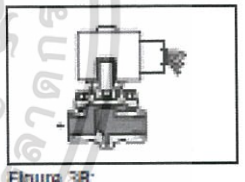


Figure 3B:
Pilot Operated, Normally Closed Valve, Energized

Manual Reset Valves (Figures 4A, 4B)

Manual reset valves must be manually latched into position and will return to their original position only when the solenoid has been energized or de-energized, depending on construction

Figure 4A:
No Voltage Release
Manual Reset Valve,
Un-Latched, De-Energized

Figure 4B:
No Voltage Release
Manual Reset Valve,
Latched, Energized

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Types of Solenoid Valves

2-Way Valves (Figures 1A, 1B, 2A, 2B, 3A, 3B)

Two-way valves have one inlet and one outlet pipe connection. They are used to allow or shut off fluid flow, and are available in either:

Normally Closed – closed when de-energized and open when energized.

Normally Open – open when de-energized and closed when energized.

3-Way Valves (Figures 5A, 5B)

Three-way valves have three pipe connections and two orifices (when one is open, the other is closed, and vice versa). They are commonly used to alternately apply pressure to and exhaust pressure from the diaphragm operator of a control valve, single-acting cylinder, or rotary actuator.

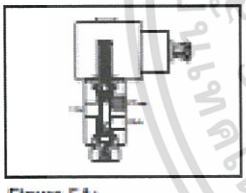


Figure 5A:
Three-Way
Normally Closed Valve,
De-Energized

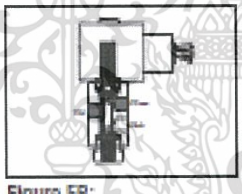


Figure 5B:
Three-Way
Normally Closed Valve,
Energized

Three modes of operation are available:

Normally Closed – when the valve is de-energized, the pressure port is closed and the cylinder port is connected to the exhaust port. When the valve is energized, the pressure port is connected to the cylinder port and the exhaust port is closed.

Normally Open – when the valve is de-energized, the pressure port is connected to the cylinder port and the exhaust port is closed. When the valve is energized, the pressure port is closed and the cylinder port is connected to the exhaust port.

Universal – allows the valve to be connected in either the Normally Closed or Normally Open position to select one of two fluids or to divert flow from one port to another.

4-Way Valves (Figures 6A, 6B)

Four-way valves are generally used to operate double-acting cylinders or actuators. They have four or five pipe connections: one pressure, two cylinder, and one or two exhausts. In Position A, pressure is connected to one cylinder port, the other is connected to exhaust. In Position B, pressure and exhaust are reversed at the cylinder ports.

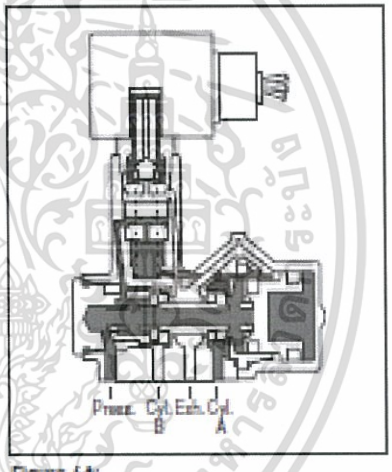


Figure 6A:
Four-Way Valve, De-Energized

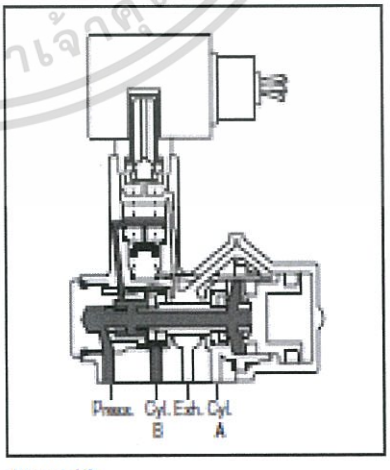


Figure 6B:
Four-Way Valve, Energized

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Engineering Information

Solenoid Valves

ASCO®

Coil Operating Voltage Ranges

All coils are designed for industrial operating voltages and can be used on the following voltage ranges:

AC		DC	
Nominal Voltage Rating	Normal Operating Range	Nominal Voltage Rating	Normal Operating Range
24	20-24	6	5.1-6.3
120	102-120	12	10.2-12.6
—	—	24	20-25
240	204-240	120	102-126
480	408-480	240	204-252

Note: Special coils are required for battery charging circuits where wider voltage ranges are typically encountered. For these applications, special continuous duty Class H coils are available that will accommodate a voltage range equivalent to 12% over nominal, 28% under nominal, and a 140°F (60°C) ambient. Standard nominal voltages are 125 and 250 DC, which translate to a voltage range of 90-140 and 180-280, respectively. Add prefix "HC" to the catalog number. "HC" prefix is only applicable to valves with coil classes FT and HT. *Consult factory for other constructions.*

Most ASCO valves, depending upon construction, will operate at 15% under nominal voltage and maximum operating pressure differential, and are capable of operating for short periods at 10% over nominal voltage. For coil classes other than FT and HT, over voltage is not recommended. *For wider voltage ranges than shown here or for operating voltage ranges for specific catalog numbers, please consult your local ASCO sales office.*

Power Consumption

Power consumption can be determined from the ratings shown on individual Series pages. For AC valves, the watts, volt-ampere "inrush" (the high momentary surge occurring at coil energization), and volt-ampere "holding" (the continuous draw following inrush) are given.

The current rating for inrush and holding may be determined by dividing the voltage into the volt-amp rating:

$$\text{Inrush Amps} = \frac{\text{volt-amp inrush}}{\text{voltage}}$$

$$\text{Holding Amps} = \frac{\text{volt-amp holding}}{\text{voltage}}$$

DC valves have no inrush current. The amp rating can be determined by dividing the voltage into the DC watt rating:

$$\text{Amps} = \frac{\text{watts (DC)}}{\text{voltage}}$$

Notes:

1. When a valve has been energized for a long period, the solenoid becomes hot and can be touched by hand for only an instant. This is a perfectly safe operating temperature. Any excessive heating will be indicated by smoke and the odor of burning coil insulation.
2. Valves for AC service can be converted to other AC voltages simply by changing the coil. Similarly, DC valves can be converted to other DC voltages. *When converting from AC to DC, or vice versa, consult your local ASCO sales office for instructions.*

Solenoid Constructions

Internal parts in contact with fluids are of non-magnetic 300 and magnetic 400 series stainless steel. In AC constructions, the shading coil is normally copper, except that silver is mostly used in valves with stainless steel bodies. Other materials are available, when required. In DC constructions, no shading coil is required. Typically, the core tubes are of 300 series stainless steel.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASCO®

Engineering Information

Flow Data

Sample Problems

Liquids: ①

To find Cv: What Cv is required to pass 20 GPM of oil, with a specific gravity of 0.9 and a pressure drop of 25 psi? The viscosity is less than 300 SSUs.②

Solution: Formula is:

$$Cv = \frac{GPM}{Fg \times Fsg}$$

To find Fg (Graph Factor), use Liquid Flow Graph on page 11.16. The Fg factor is that corresponding to 25 psi pressure drop and equals 5. The Fsg factor (Specific Gravity Factor) can be obtained from the Fsg Chart, and is that corresponding to .9 specific gravity and equals 1.05.

Therefore:

$$Cv = \frac{20}{5 \times 1.05} = 3.81$$

Air and Gases:

To find Cv: A valve is required to pass 500 SCFH at an inlet pressure of 60 psig and a Δp③ of 10 psi. Find Cv if the fluid is carbon dioxide at room temperature.

Solution: Refer to 10-100 psig graph on page 11.17. The formula to be used is:

$$Cv = \frac{SCFH}{Fg \times Fsg \times Ft}$$

Locate Fg at the intersection of 60 psig inlet pressure and 10 psi Δp③ (curved lines). Read down to Fg. Fg=1560.

Locate Fsg corresponding to specific gravity of carbon dioxide (S.G.=1.5). Fsg=0.81. (Refer to next page.) Since the gas is at room temperature, the Ft factor can be ignored.

Insert values into formula:

Steam:

To find Cv: A valve is required to pass 25 lb/hr of saturated steam at an inlet pressure of 7 psig and a Δp③ of 3 psi. What is the Cv?

Solution: Refer to the Steam Graph on page 11.18. Use formula:

$$Cv = \frac{lb/hr}{Fg}$$

Locate Fg on graph corresponding to 7 psig inlet pressure and 3 psi Δp③ (curved lines). Fg = 23.5.

Insert values into formula:

$$Cv = \frac{25}{23.5} = 1.06$$

For further information, consult your local ASCO sales office.

Notes:
① Liquid formulas and flow graphs are based on US gallons.
② If viscosity is less than 300 SSU, correction factors are not necessary.
③ Δp stands for pressure drop.

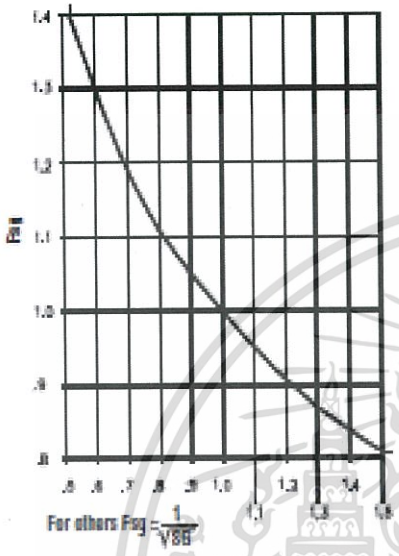
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Engineering Information

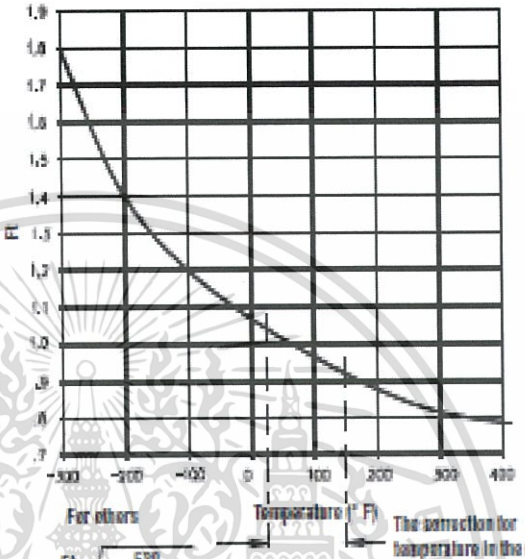
Flow Data

ASCO®

Fsg Chart

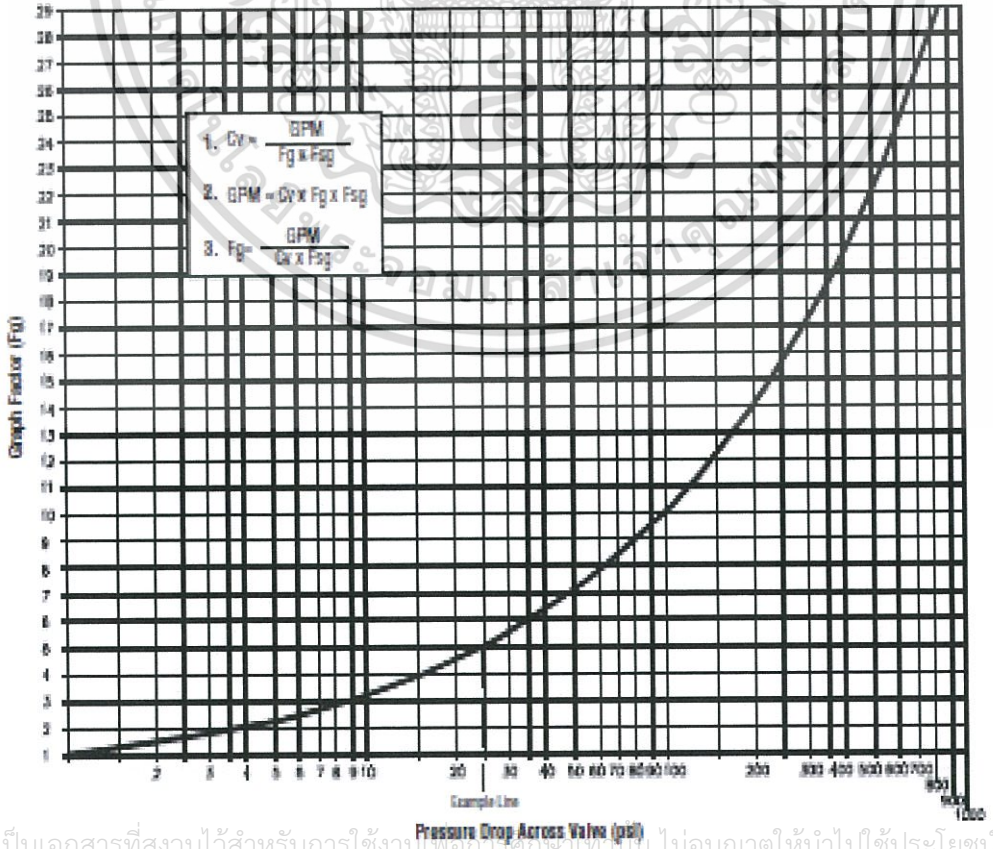


Ft Chart



The correction for temperature in the range of 20°F to 150°F is very small and, therefore, can be ignored in ordinary applications.

Liquid Flow Graph

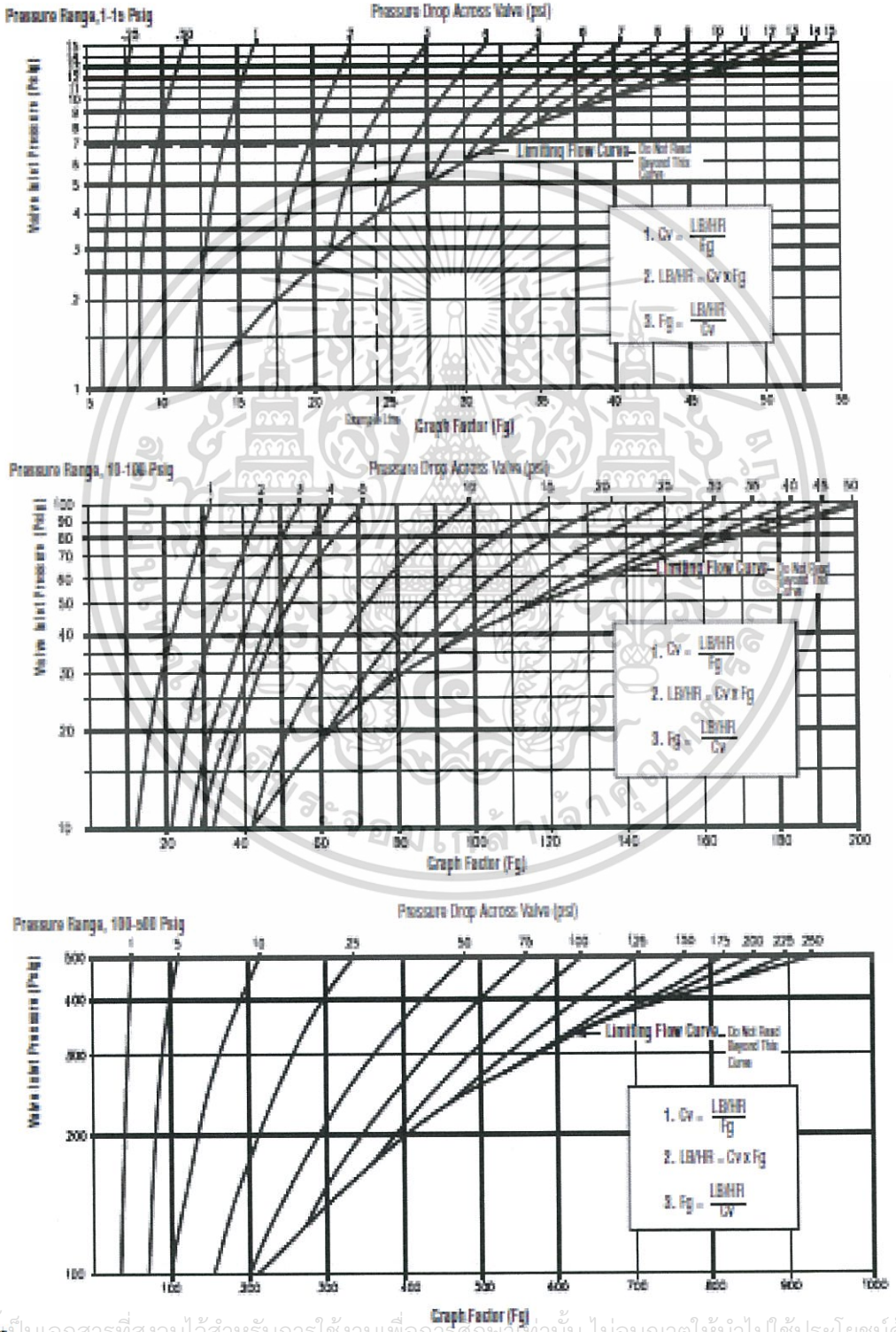


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Engineering Information

Flow Data

Steam Flow Graphs



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Relay Module Datasheet

Electronic Brick of Relay

Overview

What is an electronic brick? An electronic brick is an electronic module which can be assembled like Lego bricks simply by plugging in and pulling out. Compared to traditional universal boards and circuit modules assembled with various electronic components, electronic brick has standardized interfaces, plug and play, simplifying construction of prototype circuit on one' s own. There are many types of electronic bricks, and we provide more than twenty types with different functions including buttons, sensors, Bluetooth modules, etc, whose functions cover from sensor to motor drive, from Ethernet to wireless communication via Bluetooth, and so on. We will continue to add more types to meet the various needs of different projects.

Electronic brick of relay can be connected to digital I/O port, when the input of S port is low level, relay will be under 'disconnect' state; when the input of S port is high level, relay will be under 'connect' state. It can be connected in series to the LED lights, buzzer or even the loop of a small lamp so as to control the state of the switch.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

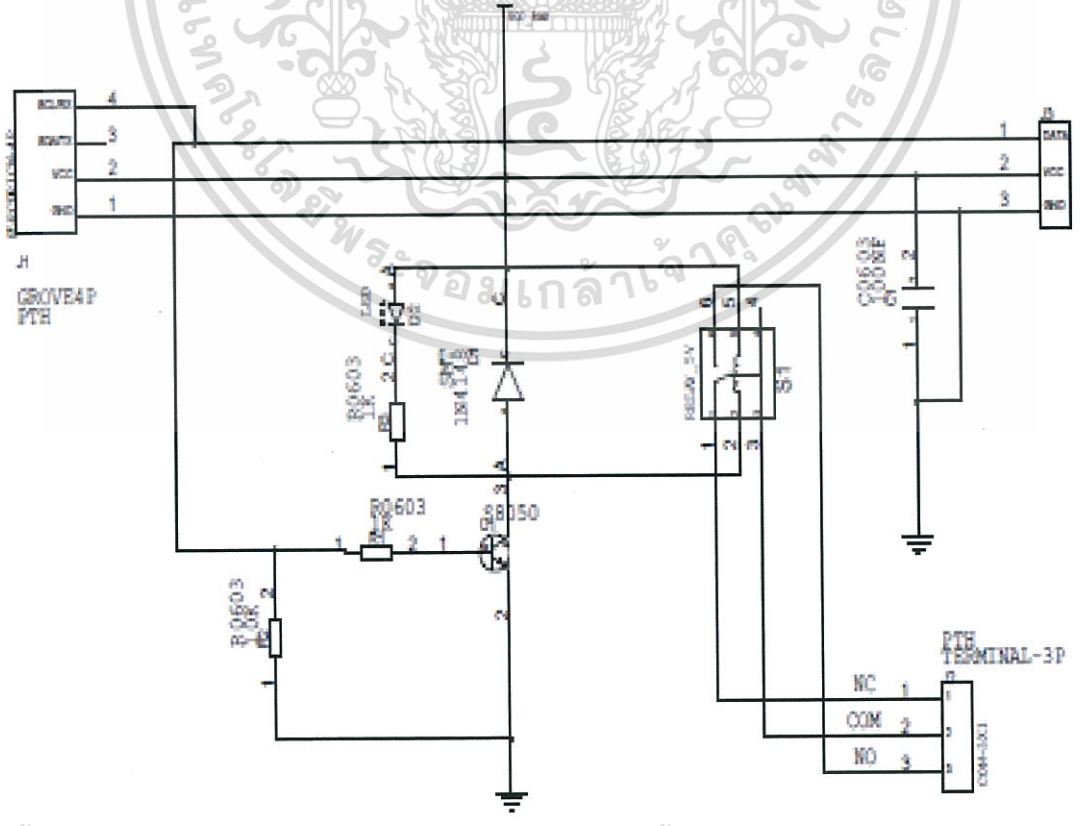
Specifications

PCB size	36.0mm X 24.0mm X 1.6mm
Working voltage	5V DC
Operating voltage	5V DC
Compatible interfaces	2.54 3-pin interface and 4-pin Grove interface[1]

Note 1 : 5 for digital output port, V and G for voltage at the common collector and ground respectively

Electrical characteristics

Parameter	Min.	Typical	Max.	Unit
Working voltage	4.5	5	5.5	VDC
Digital input voltage (VCC=5V)	0	-	5	V
Working current (VCC=5V)	-	34	-	mA
Maximum Switch current	-	2	-	A
Maximum Switch AC	-	120	-	VAC
Maximum Switch DC	-	24	-	VDC

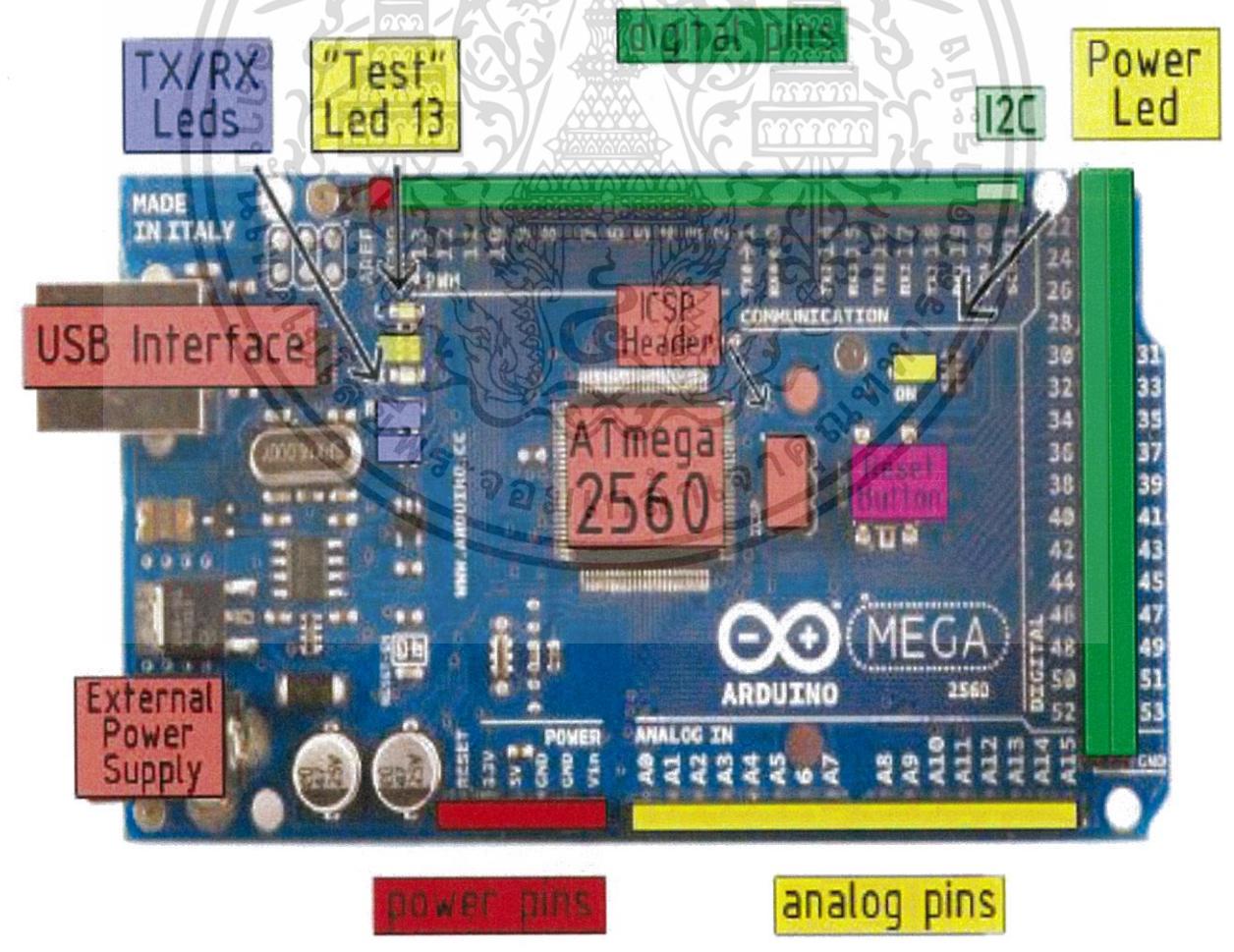


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Mega2560 Datasheet

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

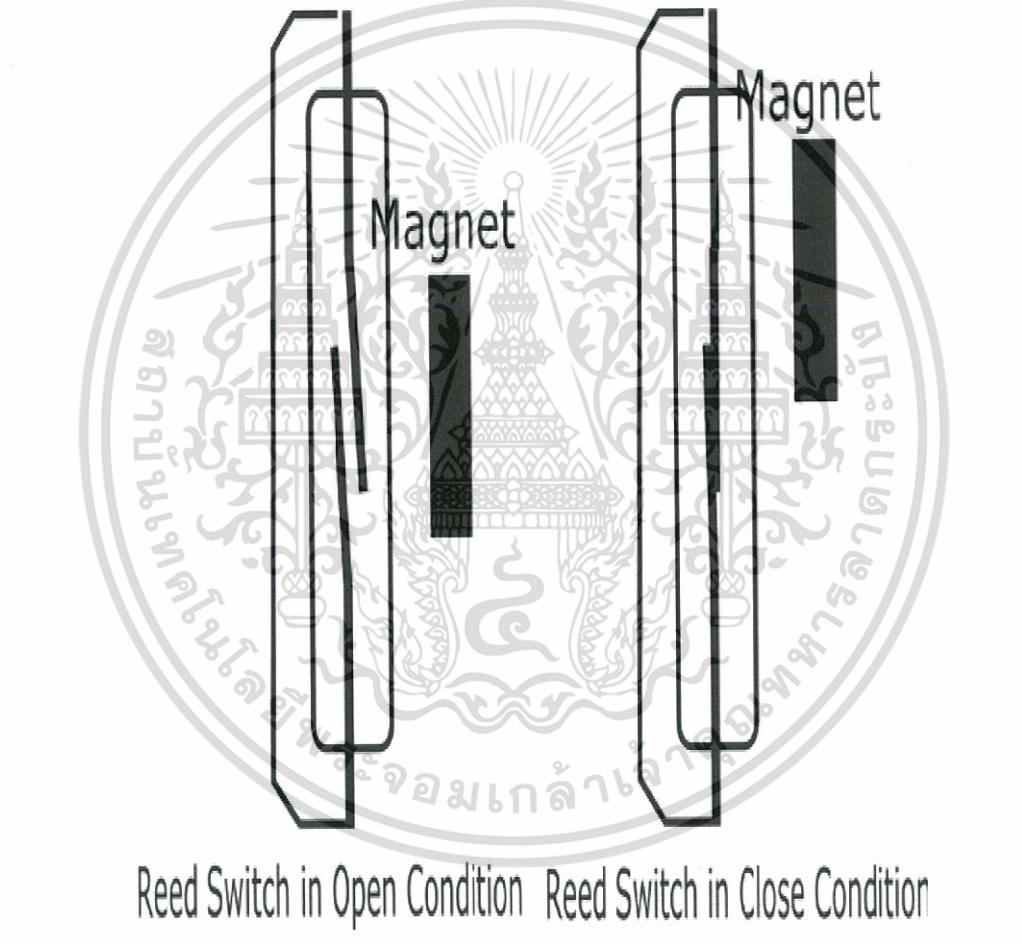
Float Switch Sensor Datasheet

1. Introduction:

A float sensor is a device used to detect the level of liquid within a tank. The switch may be used in a pump, an indicator, an alarm, or other devices. Magnetic float sensor is an electromagnetic ON/OFF switch. It helps to sense the level of water present in the overhead tank or sump.

1.1 Working:

To complete a circuit, float switches utilize a magnetic reed switch, which consists of two contacts sealed in a glass tube. When a magnet comes close to the contact, they become attracted to each other and touch, allowing current to pass through. When the magnet moves away, the contacts demagnetize and will separate (breaking the circuit).



Float Sensor which we have used is Normaly Opened (NO), that is when the float is at its low point, resting on its bottom clip circuit will be open and when float is at its high point, it will complete the circuit. So, when the water level goes down float sensor breaks the circuit and attached led will get off.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specifications:

Electrical Characteristics

Contact form (ASA)	: A.
Contact material	: Ru.
Maximum contact rating	: 10W/VA.
Maximum switching current	: 1.0A.
Maximum carry current	: 1.2A.
Maximum switching voltage	: 200V DC.
	: 140V AC.
Minimum breakdown voltage	: 240V DC.

Temperature Range

Operating temperature	: -20 to +80°C.
Storage temperature	: -30 to +80°C.

Material

Case/Nut	: PP.
Float	: PP.
Washer	: TEP.
Potting compound	: Epoxy.
Cable	: UL 1007, AWG 24.

Testing

Sensors tested 100% for electrical function and operating distance.

Remarks

When mounted near ferromagnetic parts switching points may vary.

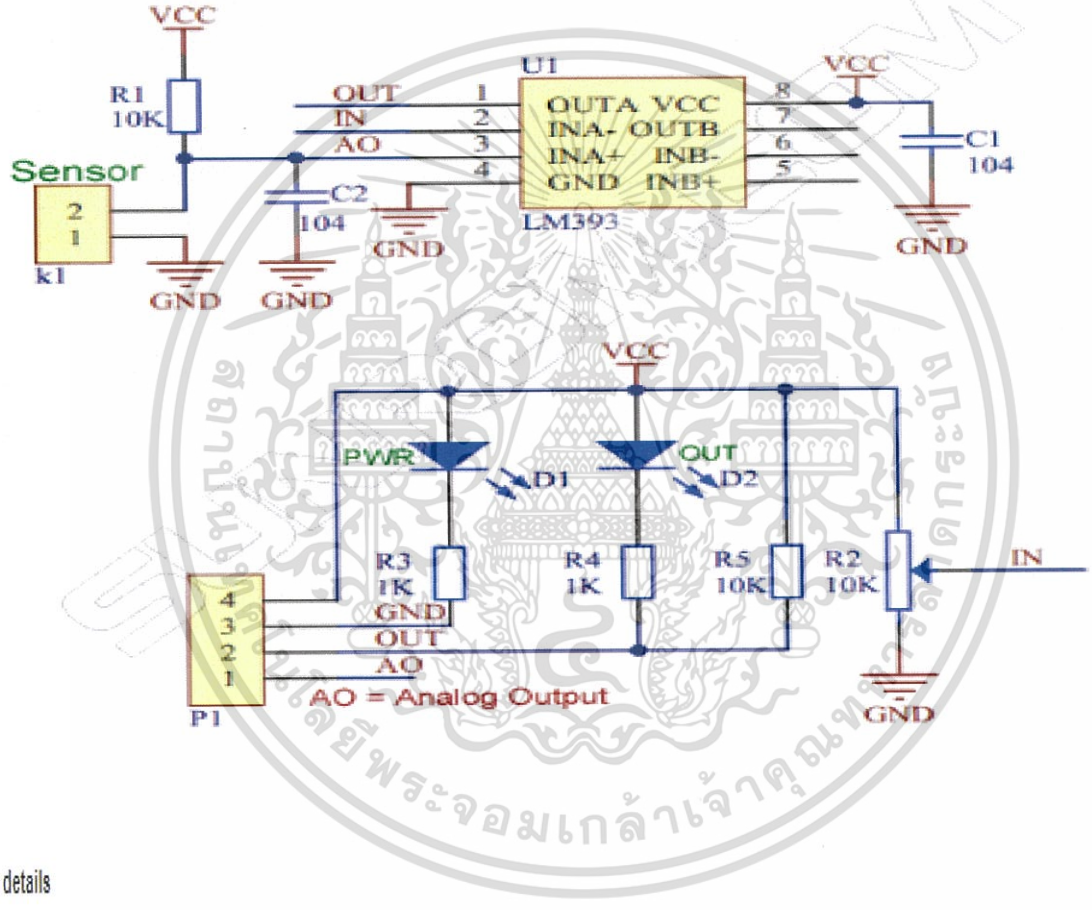
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LDR Photoresistor Sensor Module Datasheet

Specifications

- Operating Voltage: 3.3V to 5V DC
- Operating Current: 15ma
- Output Digital - 0V to 5V, Adjustable trigger level from preset
- Output Analog - 0V to 5V based on light falling on the LDR
- LEDs indicating output and power
- PCB Size: 3.2cm x 1.4cm
- LM393 based design

Board Schematic



Pin details

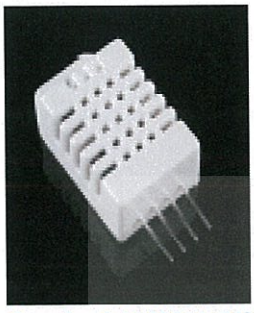
- VCC = 3.3V to 5V DC
- GND = Ground
- DO = Digital Output
- AO = Analog Output

How to use

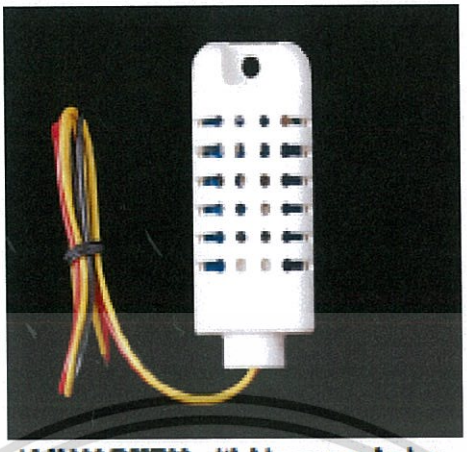
- Photosensitive resistor module most sensitive to environmental light intensity is generally used to detect the ambient brightness and light intensity.
- Module light conditions or light intensity reach the set threshold, DO port output high, when the external ambient light intensity exceeds a set threshold, the module D0 output low;
- Digital output D0 directly connected to the MCU, and detect high or low TTL, thereby detecting ambient light intensity changes;
- Digital output module DO can directly drive the relay module, which can be composed of a photoelectric switch;
- Analog output module AO and AD modules can be connected through the AD converter, you can get a more accurate light intensity value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น พร้อมทั้งขอสงวนสิทธิ์ในการฟ้องร้องดำเนินคดีตามกฎหมายกับผู้ที่นำไปใช้ในทางการค้าทุกกรณีที่มีการนำไปใช้

DHT22 Datasheet



Standard AM2302/DHT22



AM2302/DHT22 with big case and wires

Digital relative humidity & temperature sensor AM2302/DHT22

1. Feature & Application:

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

2. Description:

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

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

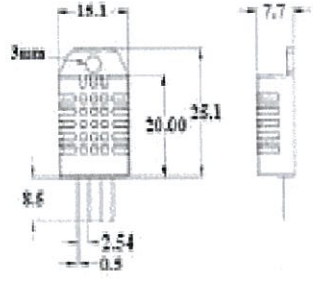
Small size & low consumption & long transmission distance(100m) enable AM2302 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

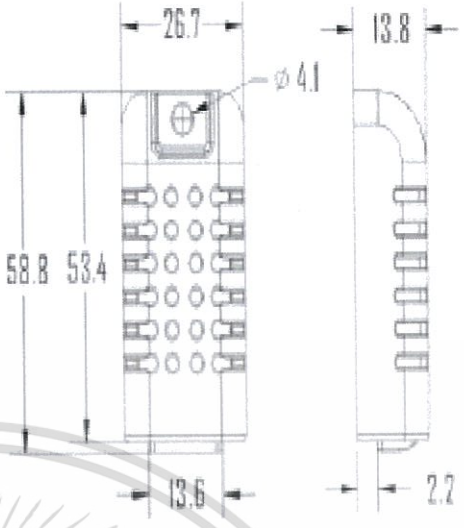
3. Technical Specification:

Model	AM2302	
Power supply	3.3-5.5V DC	
Output signal	digital signal via 1-wire bus	
Sensing element	Polymer humidity capacitor	
Operating range	humidity 0-100%RH;	temperature -40~80Celsius
Accuracy	**humidity +-2%RH**(Max +-5%RH);	temperature +-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius
Repeatability	humidity +-1%RH;	temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH	
Long-term Stability	+-0.5%RH/year	
Interchangeability	fully interchangeable	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Dimensions: (unit---mm)





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

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

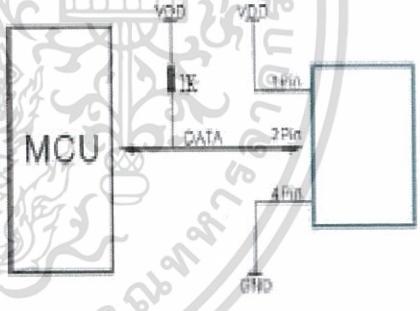
Standard AM2302's dimensions as above

Big case's dimensions as above

Red wire—power supply, Black wire—GND

Yellow wire—Data output

5. Electrical connection diagram:



6. Operating specifications:

(1) Power and Pins

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

(2) Communication and signal

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

Illustration of our 1-wire bus:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Example: MCU has received 40 bits data from AM2302 as

0000 0010 1000 1100	0000 0001 0101 1111	1110 1110
16 bits RH data	16 bits T data	check sum

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

0000 0010 1000 1100 → 652

Binary system Decimal system

RH=652/10=65.2%RH

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

0000 0001 0101 1111 → 351

Binary system Decimal system

T=361/10=36.1℃

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

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

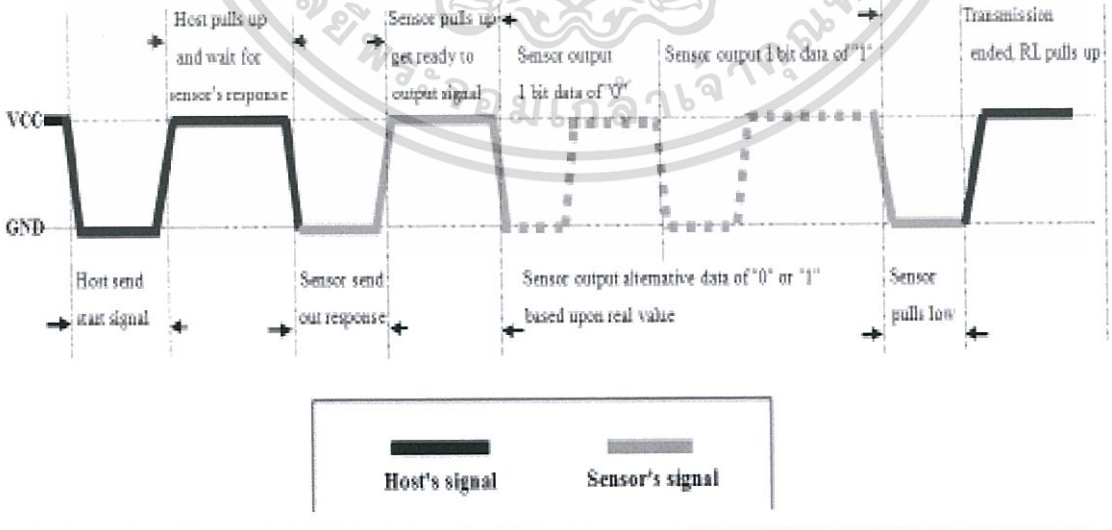
16 bits T data

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

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

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

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

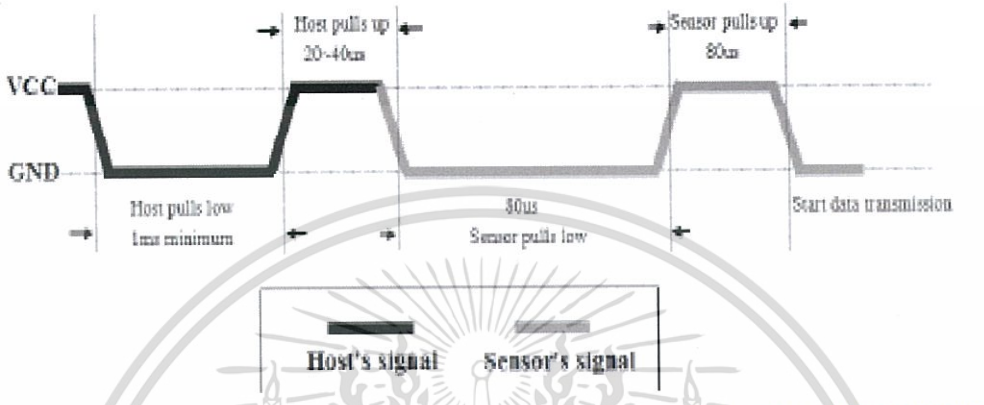


1) Step 1: MCU send out start signal to AM2302 and AM2302 send response signal to MCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

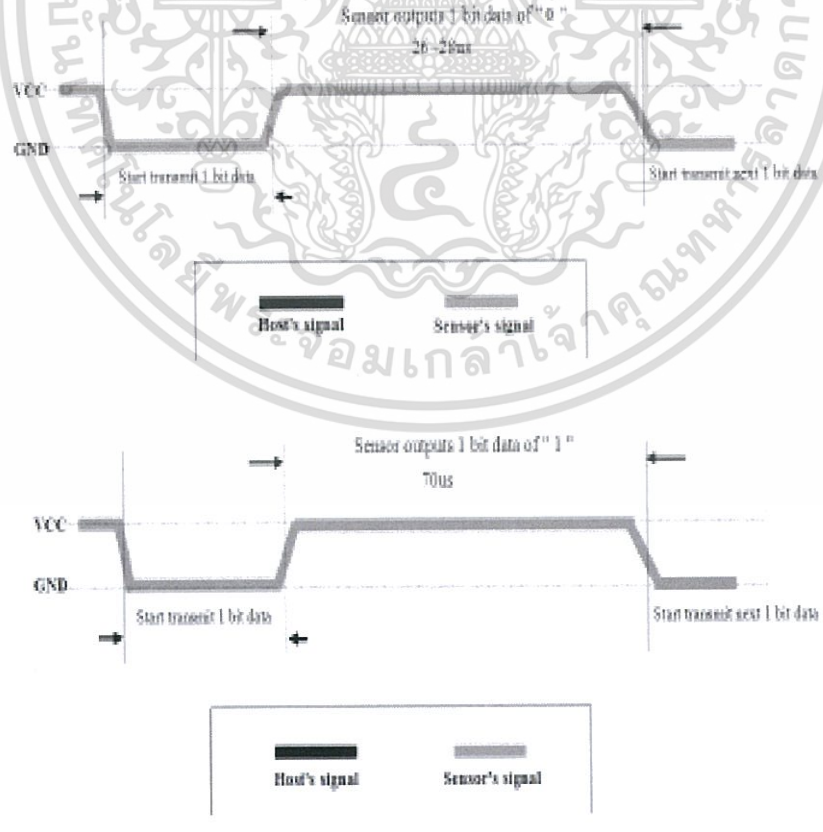
When AM2302 detect the start signal, AM2302 will pull low the bus 80us as response signal, then AM2302 pulls up 80us for preparation to send data. See below figure:



---

2). Step 2: AM2302 send data to MCU

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



Attention:

If signal from AM2302 is always high-voltage-level, it means AM2302 is not working properly, please check the electrical connection status.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. Electrical Characteristics:

Items	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	6	V
Current supply	Measuring	1		1.5	mA
	Stand-by	40	Null	50	uA
Collecting period	Second		2		Second

8. Attentions of application:

(1) Operating and storage conditions

We don't recommend the applying RH-range beyond the range stated in this specification. The AM2302 sensor can recover after working in abnormal operating condition to calibrated status, but will accelerate sensors' aging.

(2) Attentions to chemical materials

Vapor from chemical materials may interfere AM2302's sensitive-elements and debase AM2302's sensitivity.

(3) Disposal when (1) & (2) happens

Step one: Keep the AM2302 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the AM2302 sensor at condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Attention to temperature's affection

Relative humidity strongly depend on temperature, that is why we use temperature compensation technology to ensure accurate measurement of RH. But it's still be much better to keep the sensor at same temperature when sensing.

AM2302 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

Long time exposure to strong light and ultraviolet may debase AM2302's performance.

(6) Attentions to connection wires

The connection wires' quality will effect communication's quality and distance, high quality shielding-wire is recommended.

(7) Other attentions

* Welding temperature should be bellow 260Celsius.
* Avoid using the sensor under dew condition.
* Don't use this product in safety or emergency stop devices or any other occasion that failure of AM2302 may cause personal injury.

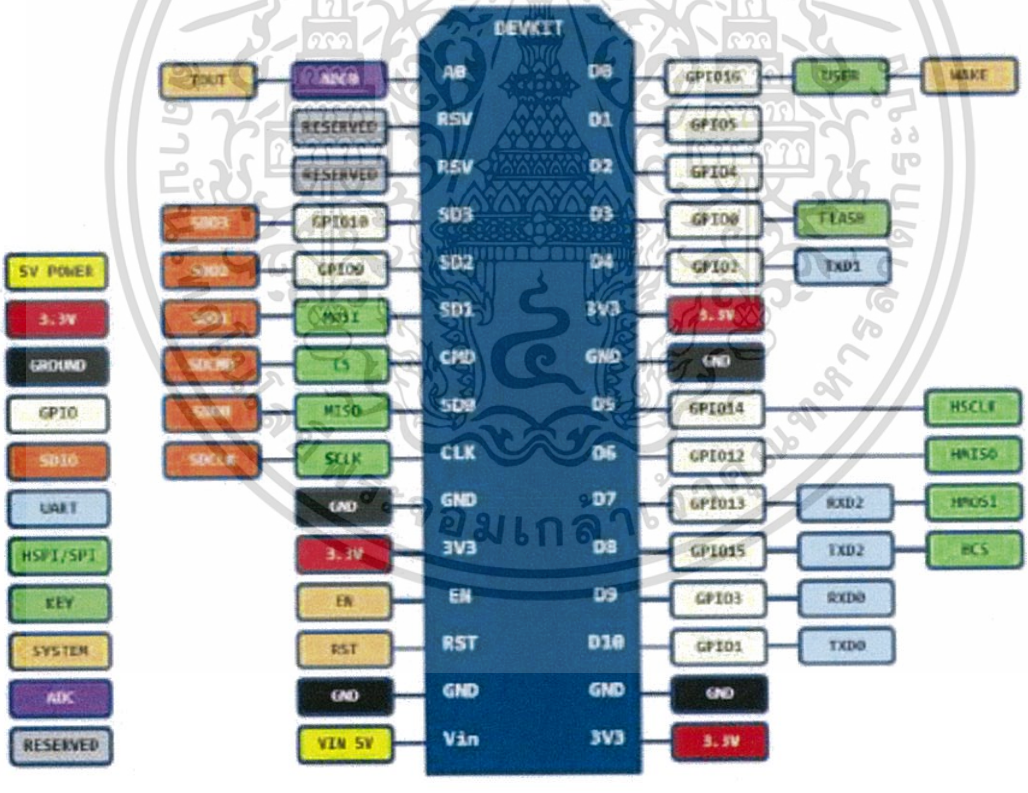
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ESP8266 Datasheet

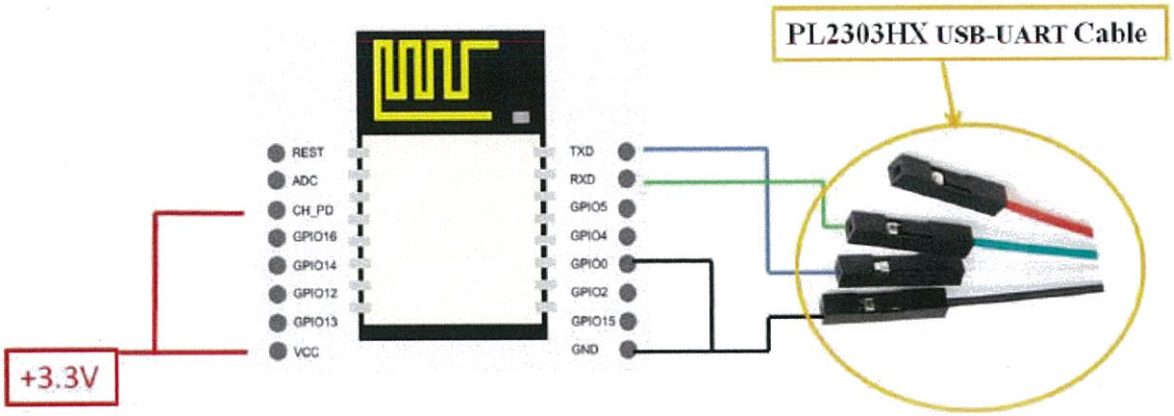
1. Specification:

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

2. Pin Definition:



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ESP8266 Pin	Description
CH_PD	Pull high, connect to Vcc +3.3V
Vcc	Power Supply +3.3V
TXD	Connect to RXD (white) of PL2303HX USB-Serial converter cable
RXD	Connect to TXD (Green) of PL2303HX USB-Serial converter cable
GPIO0	Pull low, connect to GND pin
GND	Power Supply ground

The GPIO(General Purpose Input/Output) allows us to access to pins of ESP8266 , all the pins of ESP8266 accessed using the command GPIO, all the access is based on the I/O index number on the NoddMCU dev kits, not the internal GPIO pin, for example, the pin 'D7' on the NodeMCU dev kit is mapped to the internal GPIO pin 13, if you want to turn 'High' or 'Low' that particular pin you need to called the pin number '7', not the internal GPIO of the pin. When you are programming with generic ESP8266 this confusion will arise which pin needs to be called during programming, if you are using NodeMCU devkit, it has come prepared for working with Lua interpreter which can easily program by looking the pin names associated on the Lua board. If you are using generic ESP8266 device or any other vendor boards please refer to the table below to know which IO index is associated to the internal GPIO of ESP8266.

Nodemcu dev kit	ESP8266 Pin	Nodemcu dev kit	ESP8266 Pin
D0	GPIO16	D7	GPIO13
D1	GPIO5	D8	GPIO15
D2	GPIO4	D9	GPIO3
D3	GPIO0	D10	GPIO1
D4	GPIO2	D11	GPIO9
D5	GPIO14	D12	GPIO10
D6	GPIO12		

D0 or GPIO16 can be used only as a read and write pin, no other options like PWM/I2C are supported by this pin.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้