



ระบบสารสนเทศสำหรับเซลล์แสงอาทิตย์
(Data Information for Solar Cell Systems)

จัดทำโดย

นายพีรวิชญ์ เพียรเจริญ 63010701

นางสาวภัควรินทร์ พงศ์ธนาบุลย์ 63010726

นายศิลา จูสิงห์ 63010917

อาจารย์ที่ปรึกษา

ผศ.ดร.วีระ เพ็งจันทร์

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของวิชา 01046729 PROJECT 2

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

ภาคเรียนที่ 2 ปีการศึกษา 2566

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสารสนเทศสำหรับเซลล์แสงอาทิตย์
(Data Information for Solar Cell Systems)

จัดทำโดย

นาย พีรวิชญ์ เพียรเจริญ 63010701

นางสาวภัควรินทร์ พงศ์ธนาบุลย์ 63010726

นายศิลา จูสิงห์ 63010917

อาจารย์ที่ปรึกษา

ผศ.ดร.วีระ เพ็งจันทร์

ปฏิญานិพนธ์ฉบับนี้เป็นส่วนหนึ่งของวิชา 01046729 PROJECT 2

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

ภาคเรียนที่ 2 ปีการศึกษา 2566

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของวิชา 01046729 PROJECT 2

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสารสนเทศสำหรับเซลล์แสงอาทิตย์

ผู้จัดทำ นายพีรวิษณุ เพียรเจริญ รหัสนักศึกษา 63010701

นางสาวภัควรินทร์ พงศ์ธนาบุญ รหัสนักศึกษา 63010726

นายศิลา จูสิงห์ รหัสนักศึกษา 63010917

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(ผศ.ดร.วิระ เพ็งจันทร์)

อาจารย์ที่ปรึกษา

หัวข้อปริญญานิพนธ์	ระบบสารสนเทศสำหรับเซลล์แสงอาทิตย์	
นักศึกษา	นายพีรวิษณุ เพียรเจริญ	รหัสนักศึกษา 63010701
	นางสาวภัคกรินทร์ พงศ์ธนาบุญ	รหัสนักศึกษา 63010726
	นายศิลา จูสิงห์	รหัสนักศึกษา 63010917
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์	
ปีการศึกษา	2566	
อาจารย์ที่ปรึกษา	ผศ.ดร.วีระ เพ็งจันทร์	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อศึกษาการจ่ายแรงดัน โดยอิงจากสภาพแวดล้อมโดยรอบของแผงโซลาร์เซลล์ โดยจะทำการเก็บข้อมูลค่าพารามิเตอร์ต่างๆ เช่น แรงดัน , กระแส , อุณหภูมิ , ความชื้น และความเข้มแสงแล้วจะนำข้อมูลที่ได้อมาเก็บไว้แล้วแสดงผลกราฟ โดยจะนำ Microcontroller ESP8266 มาเป็นตัวรับค่า ประมวลผล และส่งข้อมูลไปเก็บไว้ที่ ThingSpeak ซึ่งการทำงานก็จะแบ่งเป็น 2 ส่วน คือ ส่วนของ Hardware จะเป็นส่วนของ Module sensor ที่ใช้ทำการวัดค่าพารามิเตอร์ ซึ่งประกอบด้วย ACS712 , Voltage sensor , DHT22 และ BH1750FVI จากนั้นจะนำค่าส่งไปยัง ESP8266 เพื่อที่ส่งข้อมูลไปเก็บ โดยส่วนของ Software จะเป็นการเขียนโปรแกรมคำสั่งการทำงานในโปรแกรม Arduino IDE และเขียนโปรแกรมในส่วนของเว็บไซต์แล้วนำมาจากคลาวด์แล้วนำค่าที่ได้ไปแสดงผลเป็นกราฟผ่านเว็บไซต์ Thingspeak และหน้าจอ LCD

Project Title	Data Information for Solar Cell Systems	
Student	Mr. Peerawit Peanjarern	Student ID 63010701
	Miss. Pakwarin Pongthanabool	Student ID 63010726
	Mr. Sila Jusing	Student ID 63010917
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2023	
Project Advisor	Asst. Prof. Dr. Weera Pengchan	

ABSTRACT

This project is designed to study pressure distribution. Based on the surrounding environment of the solar panel. It will collect data on various parameters such as voltage, current, temperature, humidity and light intensity and then store the obtained data and display a graph. The Microcontroller ESP8266 will be used to receive the values, process and send the data. Stored at Thingspeak, the work is divided into 2 parts : The Hardware part is the Module sensor that is used to measure parameters. Which consists of ACS712 , Voltage sensor , DHT22 and BH1750FVI. Then the values will be sent to the ESP8266 for data to be sent and stored. The software part will be writing the work command program in the Arduino IDE program and writing the website program. It is then taken from the cloud and the obtained values are displayed as graphs on the Thingspeak website and LCD screen.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงบรรลุผลตามวัตถุประสงค์ได้ด้วยดีนั้น ผู้จัดทำขอกราบ
ขอบพระคุณ ผศ.ดร. วีระ เพ็งจันทร์ อาจารย์ที่ปรึกษาปริญญาานิพนธ์ฉบับนี้ ที่ให้ความช่วยเหลือคณะ
ผู้จัดทำอย่างเต็มที่ในทุกๆด้าน ขอกราบขอบพระคุณคณะกรรมการ อาจารย์และเพื่อนๆ ภาควิชา
อิเล็กทรอนิกส์ชั้นปีที่ 4 ที่กรุณาให้คำปรึกษา คำแนะนำ ข้อคิดเห็น ช่วยแก้ไขปัญหาต่างๆ และเสียสละ
เวลาส่วนตัว ดูแลการดำเนินการจัดทำ ตลอดจนตรวจสอบและแก้ไขข้อผิดพลาดต่างๆ ของปริญญาานิพนธ์
นี้ให้เสร็จสมบูรณ์

สุดท้ายนี้กลุ่มของข้าพเจ้าหวังอย่างยิ่งว่าปริญญาานิพนธ์เรื่องระบบสารสนเทศสำหรับเซลล์
แสงอาทิตย์ ฉบับนี้จะมีประโยชน์สำหรับผู้ที่เกี่ยวข้อง อีกทั้งเป็นแนวทางในการนำไปประยุกต์และต่อยอด
ต่อไป

พีรวิชญ์ เพียรเจริญ
ภักควรินทร์ พงศ์ธนาบุลย์
ศิลา จูสิงห์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VI
สารบัญตาราง.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตในการจัดทำ.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ระยะเวลาในการดำเนินการ.....	2
บทที่ 2 หลักการและทฤษฎี.....	4
2.1 Solar cell.....	4
2.2 NodeMCU ESP8266.....	7
2.3 DHT 22.....	8
2.4 MCP3208.....	10
2.5 ACS712 Current Sensor.....	11
2.6 Voltage Sensor Module.....	12
2.7 BH1750FVI.....	13
2.8 LCD.....	15
2.9 ตัวต้านทานไวร์วาวด์หุ้มฮีทซิงค์.....	17
บทที่ 3 หลักการทำงานและการออกแบบ.....	19
3.1 วัสดุ อุปกรณ์ เครื่องมือหรือโปรแกรมที่ใช้.....	19

สารบัญ

	หน้า
3.2 ขั้นตอนการดำเนินการ.....	20
3.3 การออกแบบและจัดทำระบบ.....	21
3.4 ฝังงาน.....	31
บทที่ 4 ผลการทดลอง.....	32
4.1 วัดค่าพารามิเตอร์ต่างๆ โดยแสดงผลผ่านหน้าจอ Serial Monitor.....	32
4.2 การแสดงผลผ่านเว็บไซต์.....	32
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	35
5.1 สรุปผลการทดลอง.....	35
5.2 ข้อเสนอแนะ.....	36
เอกสารอ้างอิง.....	37
ภาคผนวก.....	39

สารบัญรูป

รูปที่	หน้า
2.1 Solar cell.....	4
2.2 คุณสมบัติของแผง Solar cell.....	4
2.3 NodeMCU ESP8266.....	7
2.4 DHT22.....	8
2.5 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีทั้งหมด.....	9
2.6 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH.....	10
2.7 MCP3208.....	10
2.8 ขาของ MCP3208.....	11
2.9 ACS712.....	11
2.10 ขาของ ACS712.....	12
2.11 Voltage Sensor Module.....	13
2.12 BH1750FVI.....	13
2.13 BH1750 Block Diagram.....	14
2.14 LCD I2C.....	15
2.15 ขาของ LCD I2C.....	16
2.16 การรับ-ส่งข้อมูลแบบ I2C.....	17
2.17 ตัวต้านทานไวร์วาวด์.....	17
3.1 แผนภาพ Flowchart ขั้นตอนการทำงาน.....	20
3.2 Schematics.....	21
3.3 การออกแบบแผ่น PCB.....	21
3.4 แผ่น PCB ก่อนทำการกัดปรินต์.....	22
3.5 หน้าแรกของโปรแกรม Arduino IDE.....	23
3.6 หน้าต่าง File ของโปรแกรม Arduino IDE.....	23
3.7 ส่วนที่ทำการเขียนโปรแกรม.....	24
3.8 หน้าต่าง Tools ของโปรแกรม Arduino IDE.....	24
3.9 หน้าต่าง Library Manager.....	25

สารบัญรูป

รูปที่	หน้า
3.10 หน้าโหลด libralies ของเว็บไซต์.....	25
3.11 เลือก Board โดยการไปที่เมนู Tools > Board.....	26
3.12 เมนู Pages.....	27
3.13 กด Edit using Pagelayer เพื่อทำการแก้ไข.....	27
3.14 ตัวอย่างการแก้ไขหน้า Home.....	28
3.15 หน้า Menu.....	28
3.16 หน้าเข้าสู่ระบบของเว็บไซต์ Thingspeak.....	29
3.17 หน้าสร้าง My Channels.....	29
3.18 หน้าตั้งชื่อ Channel และ Field.....	29
3.19 หน้าดู API Keys ของ ThingSpeak.....	30
3.20 หน้าต่างปรับ Timescale และสีของกราฟ.....	30
3.21 หน้าต่าง Add Widgets.....	30
3.22 ผังงานแสดงขั้นตอนการทำงาน.....	31
4.1 หน้า Serial Monitor ในโปรแกรม Arduino IDE.....	32
4.2 หน้าต่างแสดงผลค่า ณ ปัจจุบัน และกราฟแสดงค่าก่อนหน้า.....	32
4.3 หน้าต่าง Export recent data.....	33
4.4 ตัวอย่าง Data ที่เก็บใน Excel.....	33
4.5 หน้าจอหลักของเว็บไซต์ Wordpress.....	34
4.6 ลองนำไปต่อกับ Power supply ก่อนใช้งาน.....	34
4.7 นำไปใช้งานจริง.....	34

สารบัญตาราง

ตารางที่	หน้า
1.1 ระยะเวลาในการดำเนินโครงการตั้งแต่วันที่ 10 สิงหาคม 2566 ถึง 18 ตุลาคม 2566.....	2
1.2 ระยะเวลาในการดำเนินโครงการตั้งแต่วันที่ 14 ธันวาคม 2566 ถึง 6 มีนาคม 2567.....	3



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันเราไม่สามารถปฏิเสธได้ว่า พลังงานเป็นสิ่งที่มีความจำเป็นต่อชีวิตประจำวันของเรามาก ทั้งในส่วนของภาครัฐและภาคเอกชน ต่างก็ได้มีการรณรงค์ให้มีการประหยัดพลังงาน ซึ่งทางเลือกหนึ่งที่นิยมใช้ก็คือ พลังงานแสงอาทิตย์ เนื่องจากเป็นพลังงานทดแทนที่เราสามารถหาได้ง่าย และเป็นพลังงานจากธรรมชาติที่ไม่มีวันหมด และในปัจจุบันตามบ้านเรือนต่างๆ ก็มีความนิยมในการติดตั้งแผงเซลล์แสงอาทิตย์ (Solar Module) เพื่อช่วยในการลดค่าใช้จ่ายไฟฟ้ามากยิ่งขึ้น แต่พลังงานไฟฟ้าที่ผลิตได้แต่ละวันนั้นมีการเปลี่ยนแปลงอยู่ตลอดเวลา ขึ้นอยู่กับปัจจัยภายนอกต่างๆ เช่น ปริมาณความส่องสว่างของแสงแดด อุณหภูมิโดยรอบ ชนิดและขนาดของแผงเซลล์แสงอาทิตย์ เป็นต้น จึงจำเป็นที่จะต้องมีการจดบันทึกข้อมูล เพื่อดูผลที่ได้รวมถึงแนวโน้มการเพิ่มขึ้น - ลดลงของพลังงานที่ได้จากแผงเซลล์แสงอาทิตย์ ซึ่งโครงการนี้ก็เป็นส่วนหนึ่งที่จะช่วยให้การจดบันทึกค่าที่ทำการตรวจวัดนั้นมีประสิทธิภาพมากขึ้น และสามารถนำไปวิเคราะห์ผลต่อไปได้จึงได้ทำการศึกษาและจัดทำเครื่องมือวัดและบันทึกค่าเพื่อวิเคราะห์ประสิทธิภาพของโซล่าเซลล์ โดยการใช้ไมโครคอนโทรลเลอร์ในการแก้ไขปัญหาความผิดพลาด และลดเวลาการทำงานในการจดบันทึกของผู้ใช้งาน โดยเราจะบันทึกค่าอุณหภูมิโดยรอบบริเวณ Solar Module บันทึกค่าความสว่างบริเวณ Solar Module และแสดงผลผ่านสัญญาณอินเตอร์เน็ตไร้สาย

1.2 วัตถุประสงค์

- 1.2.1 เพื่อศึกษาเกี่ยวกับไมโครคอนโทรลเลอร์
- 1.2.2 เพื่ออำนวยความสะดวกในการจดบันทึกข้อมูล
- 1.2.3 เพื่อให้ง่ายต่อการนำข้อมูลมาใช้ให้เกิดประโยชน์

1.3 ขอบเขตในการจัดทำ

- 1.3.1 เก็บบันทึกค่าพลังงานที่ได้จากแผงเซลล์แสงอาทิตย์แล้วนำไปแสดงผลผ่านทางเว็บไซต์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถทราบได้ว่าแต่ละช่วงเวลานั้นมีพลังงานที่ได้รับจากแผงเซลล์แสงอาทิตย์มากน้อย
อย่างไร

1.4.2 ทำให้สามารถวัดค่าพลังงานที่ได้จากแผงเซลล์แสงอาทิตย์ง่ายขึ้น

1.4.3 สามารถดูข้อมูลค่าพารามิเตอร์ต่างๆของแผงเซลล์แสงอาทิตย์ได้ โดยไม่จำเป็นต้องไปวัด
ค่าที่อยากรู้ด้วยตนเอง

1.5 ระยะเวลาในการดำเนินการ

1.5.1 ภาคเรียนที่ 1

รายละเอียด /สัปดาห์ที่	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน				
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3		
1. ปรึกษาอาจารย์ที่ปรึกษา เกี่ยวกับหัวข้อของโครงการ	←→																
2. ศึกษาหาข้อมูลที่เกี่ยวข้อง			←→														
3. จัดซื้ออุปกรณ์					←→												
4. ออกแบบโครงสร้างของตัว อุปกรณ์						←→											
5. ทดสอบวงจร/อุปกรณ์					←→												
6. ประกอบโครงสร้าง									←→								
7. จัดทำรูปเล่มโครงการ								←→									

ตารางที่ 1.1 ระยะเวลาในการดำเนินโครงการตั้งแต่วันที่ 10 สิงหาคม 2566 ถึง 18 ตุลาคม 2566

บทที่ 2

หลักการและทฤษฎี

2.1 Solar Cell

โซลาร์เซลล์ (Solar Cell) หรือ เซลล์แสงอาทิตย์ และที่หลายๆ คนรู้จักในชื่อ เซลล์โฟโตโวลตาอิก (Photovoltaic cell) เป็นอุปกรณ์อิเล็กทรอนิกส์ มีถูกนำมาใช้กันอย่างแพร่หลาย ในการผลิตกระแสไฟฟ้าจากพลังงานแสงอาทิตย์ ที่ทำจากสารกึ่งตัวนำชนิดพิเศษ ที่มีคุณสมบัติในการเปลี่ยนพลังงานแสงอาทิตย์ ให้เป็นพลังงานไฟฟ้า (Solar Cell) โดยพลังงานกระแสไฟฟ้าที่ผลิตได้จากโซลาร์เซลล์นั้น เป็นไฟฟ้ากระแสตรง หรือ DC ที่สามารถนำมาใช้ประโยชน์ได้ทันที



รูปที่ 2.1 Solar cell



รูปที่ 2.2 คุณสมบัติของแผง Solar cell

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการการทำงานของ Solar Cell

การทำงานของโซลาร์เซลล์จะเป็นกระบวนการเปลี่ยนจากพลังงานแสงให้กลายเป็นพลังงานไฟฟ้าได้โดยตรง โดยการใช้แสงซึ่งเป็นคลื่นแม่เหล็กไฟฟ้า และมีพลังงานไปกระทบกับสารกึ่งตัวนำ จะทำให้เกิดการถ่ายเทพลังงานระหว่างกัน โดยพลังงานจากแสงจะทำให้เกิดอิเล็กตรอน หรือ การเคลื่อนที่ของกระแสไฟฟ้าขึ้นในสารกึ่งตัวนำ จึงทำให้เกิดกระแสไฟฟ้าง่ายๆที่สามารถนำไปใช้งานได้

แผงโซลาร์เซลล์สามารถแบ่งออกได้เป็น 3 ชนิด ซึ่งแต่ละชนิดมีความแตกต่าง ข้อดีและเสียแตกต่างกันไป

1. แผงโซลาร์เซลล์ชนิด โมโนคริสตัลไลน์ (Monocrystalline Silicon Solar Cells)

แผงโซลาร์เซลล์ ชนิดที่ทำมาจาก ผลึกซิลิคอนเชิงเดี่ยว (mono-Si) หรือบางทีก็เรียกว่า single crystalline (single-Si) สังเกตดูค่อนข้างง่ายกว่าชนิดอื่น เพราะจะเห็นแต่ละเซลล์ลักษณะเป็นสี่เหลี่ยมตัดมุมทั้งสี่มุม และมีสี่เข็ม

แผงโซลาร์เซลล์ ชนิดที่ทำมาจาก ผลึกซิลิคอนเชิงเดี่ยว (mono-Si) หรือบางทีก็เรียกว่า single crystalline (single-Si) สังเกตดูค่อนข้างง่ายกว่าชนิดอื่น เพราะจะเห็นแต่ละเซลล์ลักษณะเป็นสี่เหลี่ยมตัดมุมทั้งสี่มุม และมีสี่เข็ม

แผงโซลาร์เซลล์ชนิด โมโนคริสตัลไลน์ นั้น เป็นชนิดที่ทำมาจากซิลิคอนที่มีความบริสุทธิ์สูง โดยเริ่มมาจากแท่งซิลิคอนทรงกระบอก อันเนื่องมาจาก เกิดจากกระบวนการ กวนให้ผลึกเกาะกันที่แกนกลาง ที่เรียกว่า Czochralski process จึงทำให้เกิดแท่งทรงกระบอก จากนั้นจึงนำมาตัดให้เป็นสี่เหลี่ยม และลบมุมทั้งสี่ออก เพื่อที่จะให้ได้ประสิทธิภาพสูงสุด และลดการใช้วัตถุดิบโมโนซิลิคอนลง ก่อนที่จะนำมาตัดเป็นแผ่นอีกที จึงทำให้เซลล์แต่ละเซลล์หน้าตาเป็นอย่างที่เราเห็นในแผงโซลาร์เซลล์

ข้อดี

- แผงโซลาร์เซลล์ชนิด โมโนคริสตัลไลน์ มีประสิทธิภาพสูงสุด เพราะผลิตมาจาก ซิลิคอนเกรดดีที่สุด โดยมีประสิทธิภาพเฉลี่ยอยู่ที่ 15-20%
- แผงโซลาร์เซลล์ชนิด โมโนคริสตัลไลน์ มีประสิทธิภาพต่อพื้นที่สูงสุด เพราะว่าให้กำลังสูงจึงต้องการพื้นที่น้อยที่สุดในการติดตั้งแผงโซลาร์เซลล์ชนิดนี้ โมโนคริสตัลไลน์ สามารถผลิตกระแสไฟฟ้าได้เกือบ 4 เท่า ของชนิด ฟิล์มบางหรือ thin film
- แผงโซลาร์เซลล์ชนิด โมโนคริสตัลไลน์ มีอายุการใช้งานยาวนานที่สุด โดยเฉลี่ยแล้วประมาณ 25 ปีขึ้นไป
- แผงโซลาร์เซลล์ชนิด โมโนคริสตัลไลน์ ผลิตกระแสไฟฟ้าได้มากกว่าชนิด โพลีคริสตัลไลน์ เมื่ออยู่ในภาวะแสงน้อย

ข้อเสีย

- แผงโซลาร์เซลล์ชนิด โมโนคริสตัลไลน์ เป็นชนิดที่มีราคาแพงที่สุด ในบางครั้งการติดตั้งด้วย แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ หรือชนิด thin film อาจมีความคุ้มค่าน่ามากกว่า
- ถ้าหาก แผงโซลาร์เซลล์ชนิด โมโนคริสตัลไลน์ มีความสกปรกหรือถูกบังแสงในบางส่วนของแผง อาจทำให้วงจรหรือ inverter ใหม่ได้ เพราะอาจจะทำให้เกิดโวลต์สูงเกินไป

2. แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ (Polycrystalline Silicon Solar Cells)

แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ เป็นแผงโซลาร์เซลล์ชนิดแรก ที่ทำมาจากผลึกซิลิคอน โดยทั่วไปเรียกว่า โพลีคริสตัลไลน์ (polycrystalline, p-Si) แต่บางครั้งก็เรียกว่า มัลติ-คริสตัลไลน์ (multi-crystalline, mc-Si) โดยในกระบวนการผลิต สามารถที่จะนำเอา ซิลิคอนเหลว มาเทใส่โมลด์ที่เป็นสี่เหลี่ยมได้เลย ก่อนที่จะนำมาตัดเป็นแผ่นบางอีกที จึงทำให้เซลล์แต่ละเซลล์เป็นรูปสี่เหลี่ยมจัตุรัส ไม่มีการตัดมุม สีของแผงจะออก น้ำเงิน ไม่เข้มมาก

ข้อดี

- แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ มีขั้นตอนกระบวนการผลิตที่ง่าย ไม่ซับซ้อน จึง ใช้ปริมาณซิลิคอน ในการผลิตน้อยกว่า เมื่อเทียบกับ ชนิด โมโนคริสตัลไลน์
- แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ มีประสิทธิภาพในการใช้งาน ในที่อุณหภูมิสูง ดีกว่า ชนิด โมโนคริสตัลไลน์ เล็กน้อย
- แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ มีราคาถูกกว่าเมื่อเทียบกับ ชนิด โมโนคริสตัลไลน์

ข้อเสีย

- แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ มีประสิทธิภาพโดยเฉลี่ยอยู่ที่ 13-16% ซึ่งต่ำกว่า เมื่อเทียบกับชนิด โมโนคริสตัลไลน์
- แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ มีประสิทธิภาพต่อพื้นที่ต่ำกว่า ชนิด โมโนคริสตัลไลน์
- แผงโซลาร์เซลล์ชนิด โพลีคริสตัลไลน์ มีสีน้ำเงิน ทำให้บางครั้งอาจดูไม่สวยงาม เมื่อเทียบกับ ชนิด โมโนคริสตัลไลน์ และชนิด thin film ที่มีสีเข้ม เข้ากับสิ่งแวดล้อม เช่น หลังคาบ้านได้ดีกว่า

3. แผงโซลาร์เซลล์ชนิด ฟิล์มบาง (Thin Film Solar Cells) แผงโซลาร์เซลล์ อะมอร์ฟัส เป็นหนึ่งในหลายชนิด ของแบบฟิล์มบาง

หลักการโดยทั่วไปของการผลิต โซลาร์เซลล์ ชนิดฟิล์มบาง (Thin Film Solar Cell, TFSC) คือ การนำเอาสารที่สามารถแปลงพลังงานจากแสงเป็นกระแสไฟฟ้า มาฉาบเป็นฟิล์มหรือชั้นบางๆ ซ้อนกันหลายชั้น จึงเรียก โซลาร์เซลล์ชนิดนี้ว่า ฟิล์มบาง หรือ thin film ซึ่งสารฉาบที่วานี้ก็มีด้วยกันหลายชนิด

ชื่อเรียกของ แผงโซลาร์เซลล์ ชนิดฟิล์มบางจึงแตกต่างกันออกไป ขึ้นอยู่กับชนิดวัสดุที่นำมาใช้ ได้แก่ อะมอร์ฟัส Amorphous silicon (a-Si), Cadmium telluride (CdTe), Copper indium gallium selenide (CIS/CIGS) และ Organic photovoltaic cells (OPC)

ด้านประสิทธิภาพของ แผงโซลาร์เซลล์ ชนิดฟิล์มบางนั้น มีประสิทธิภาพเฉลี่ยอยู่ที่ 7-13% ทั้งนี้ขึ้นอยู่กับชนิดของวัสดุที่นำมาทำเป็นฟิล์มฉาบ แต่สำหรับบ้านเรือนโดยทั่วไปแล้ว มีเพียงประมาณ 5% เท่านั้น ที่ใช้ แผงโซลาร์เซลล์ ที่เป็นแบบชนิดฟิล์มบาง

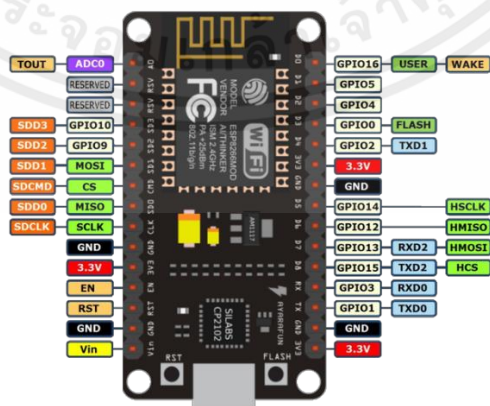
ข้อดี

- แผงโซลาร์เซลล์ ชนิดฟิล์มบาง มีราคาถูกกว่า เพราะสามารถผลิตจำนวนมากได้ง่ายกว่า ชนิดผลึกซิลิคอน
- ในที่อากาศร้อนมากๆ แผงโซลาร์เซลล์ ชนิด ฟิล์มบาง มีผลกระทบน้อยกว่า
- ไม่มีปัญหาเรื่อง เมื่อแผงสกปรกแล้วจะทำให้วงจรไหม้
- ถ้าคุณมีที่เหลือเพื่อ แผงโซลาร์เซลล์ ชนิด ฟิล์มบาง ก็เป็นทางเลือกที่ดี

ข้อเสีย

- แผงโซลาร์เซลล์ชนิดฟิล์มบาง มีประสิทธิภาพต่ำ
- แผงโซลาร์เซลล์ ชนิด ฟิล์มบาง มีประสิทธิภาพต่อพื้นที่ต่ำ
- สิ้นเปลืองค่าโครงสร้างและอุปกรณ์อื่นๆ เช่น สายไฟ
- ไม่เหมาะนำมาใช้ตามหลังคาบ้าน เพราะมีพื้นที่จำกัด
- การรับประกันสั้นกว่าชนิด ผลึกซิลิคอน

2.2 NodeMCU ESP8266



รูปที่ 2.3 NodeMCU ESP8266

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ESP8266 เป็นชื่อของชิปไอซีบนบอร์ดของโมดูล ซึ่งไอซี ESP8266 ไม่มีพื้นที่โปรแกรม (flash memory) ในตัว ทำให้ต้องใช้ไอซีภายนอก (external flash memory) ในการเก็บโปรแกรม ที่ใช้การเชื่อมต่อผ่านโปรโตคอล SPI ซึ่งสาเหตุนี้เองทำให้โมดูล ESP8266 มีพื้นที่โปรแกรมมากกว่าไอซีไมโครคอนโทรลเลอร์เบอร์อื่นๆ ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3V - 3.6V การนำไปใช้งานร่วมกับเซ็นเซอร์อื่นๆที่ใช้แรงดัน 5V ต้องใช้วงจรแบ่งแรงดันมาช่วย เพื่อไม่ให้โมดูลพังเสียหาย กระแสที่โมดูลใช้งานสูงสุดคือ 200mA ความถี่คริสตอล 40MHz ทำให้เมื่อนำไปใช้งานอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำให้การแสดงผลข้อมูลรวดเร็วกว่าไมโครคอนโทรลเลอร์ยี่ห้ออื่นมาก

ขาของโมดูล ESP8266 แบ่งได้ดังนี้

- VCC เป็นขาสำหรับจ่ายไปเข้าเพื่อให้โมดูลทำงานได้ ซึ่งแรงดันที่ใช้งานได้คือ 3.3 - 3.6V
- GND
- Reset และ CH_PD (หรือ EN) เป็นขาที่ต้องต่อเข้าไฟ + เพื่อให้โมดูลสามารถทำงานได้ ทั้ง 2 ขานี้สามารถนำมาใช้รีเซ็ตโมดูลได้เหมือนกัน แตกต่างตรงที่ขา Reset สามารถลอยไว้ได้ แต่ขา CH_PD (หรือ EN) จำเป็นต้องต่อเข้าไป + เท่านั้น เมื่อขานี้ไม่ต่อเข้าไฟ + โมดูลจะไม่ทำงานทันที
- GPIO เป็นขาดิจิตอลอินพุต / เอาต์พุต ทำงานที่แรงดัน 3.3V
- GPIO15 เป็นขาที่ต้องต่อลง GND เท่านั้น เพื่อให้โมดูลทำงานได้
- GPIO0 เป็นขาสำหรับการเลือกโหมดทำงาน หากนำขานี้ลง GND จะเข้าโหมดโปรแกรม หากลอยไว้ หรือนำเข้าไฟ + จะเข้าโหมดการทำงานปกติ
- ADC เป็นขาแอนาล็อกอินพุต รับแรงดันได้สูงสุดที่ 1V ขนาด 10 บิต การนำไปใช้งานกับแรงดันที่สูงกว่าต้องใช้อุปกรณ์แบ่งแรงดันเข้าช่วย

2.3 DHT 22



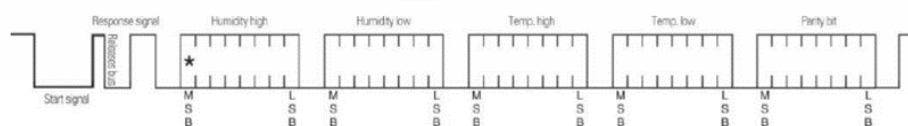
รูปที่ 2.4 DHT 22

DHT 22 เป็นอุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ (Temperature & Relative Humidity Sensor) การทดลองใช้งานโมดูล DHT22 มีราคาถูก ให้ค่าเป็นแบบดิจิทัล ใช้ขาสัญญาณดิจิทัลเพียงเส้นเดียว โดยในการเชื่อมต่อแบบบิตอนุกรมสองทิศทาง (Serial Data, Bi-Directional) โดยจะนำมาเชื่อมต่อกับ Arduino เพื่ออ่านค่าจากเซนเซอร์

- ใช้แรงดันไฟเลี้ยงได้ในช่วง 3.3 V ถึง 5.5 V
- วัดอุณหภูมิได้ในช่วง -40 to 80 °C (± 0.5 °C accuracy)
- วัดความชื้นสัมพัทธ์ได้ในช่วง 0 - 100 RH% (2 - 5% accuracy)
- คอนเนคเตอร์แบบ 4 ขา
- Pin 1 = VCC
- Pin 2 = SDA (Serial data, bidirectional)
- Pin 3 = ไม่ได้งาน
- Pin 4 = GND

หลักการการทำงานของ DHT22

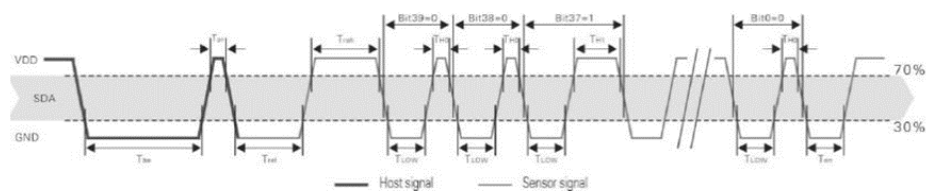
ในการอ่านข้อมูลจากไอซีนั้น จะใช้ขาสัญญาณเพียงเส้นเดียวคือ DATA (หรือ SDA) แบบสองทิศทาง และ ในสถานะปกติสัญญาณ DATA จะเป็น HIGH ในการอ่านข้อมูลแต่ละครั้ง ไมโครคอนโทรลเลอร์จะต้องกำหนดให้ ขา DATA เป็นเอาต์พุต และสร้างบิต START ซึ่งจะต้องเป็น LOW อย่างน้อย 800 μ sec จากนั้นจึงให้เป็น HIGH อย่างน้อย 20 μ sec หลังจากนั้นจะเป็นการรอการตอบกลับ (Response) และจากไอซีขา DATA จะถูกต้อง เปลี่ยนเป็นอินพุต เริ่มต้นของการตอบกลับไอซี จะดึงสัญญาณลงเป็น LOW และปล่อยให้เป็น HIGH ช่วงละ 80 μ sec โดยประมาณ (เรียกว่า Response Bit) จากนั้นจึงจะเป็นการส่งข้อมูลที่ละบิต รวม 40 บิต (ช่วง LOW ตามด้วยช่วง HIGH) ช่วง LOW ของแต่ละบิต จะกว้างเท่ากัน แต่จะต่างกันในช่วง HIGH สำหรับบิตที่มีค่าเป็น 0 หรือ 1 (ใช้ความกว้างช่วง HIGH ในการจำแนกค่าของบิต)



รูปที่ 2.5 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีทั้งหมด

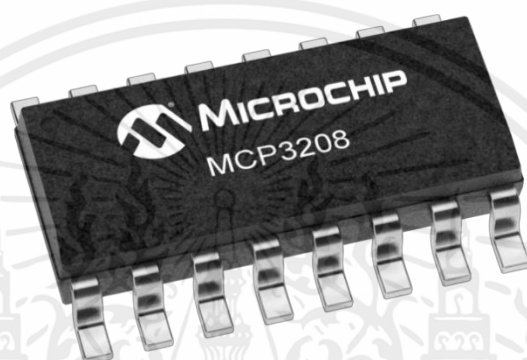
สองไบต์แรกสำหรับความชื้น สองไบต์ต่อมาสำหรับอุณหภูมิและไบต์สุดท้ายเป็น Checksum หรือ Parity Bits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH

2.4 MCP3208



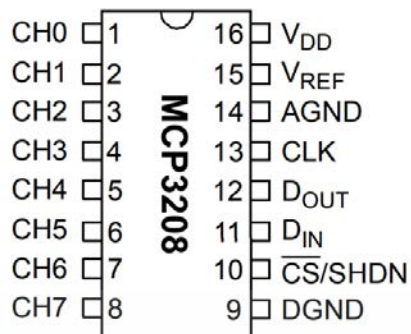
รูปที่ 2.7 MCP3208

เซ็นเซอร์ต่าง ๆ ที่ต่อเข้ากับไมโครคอนโทรลเลอร์ มีการรับ - ส่งข้อมูลแบ่งเป็น 2 แบบใหญ่ ๆ คือ แบบดิจิทัล และแบบอนาล็อก แม้ในปัจจุบันเซ็นเซอร์ต่าง ๆ จะเปลี่ยนการสื่อสารไปใช้แบบดิจิทัลแทบจะทั้งหมดแล้ว แต่ก็ยังมีเซ็นเซอร์บางชนิดที่ยังต้องสื่อสารแบบอนาล็อกอยู่ เช่น เซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์วัดอุณหภูมิ TMP35 เซ็นเซอร์แสงใช้ LDR เซ็นเซอร์วัดกระแสไฟฟ้า เป็นต้น

MCP3208 เป็นไอซี ADC ความละเอียด 12 บิต 8 ช่อง ตัวถังแบบ DIP-16 สามารถเสียบลงโปรโตบอร์ดได้ มีสเปคร่าว ๆ ดังนี้

- รองรับแรงดันไฟเลี้ยงได้ 2.7V ถึง 5.5V
- สื่อสารกับไมโครคอนโทรลเลอร์ผ่านบัส SPI
- กำหนดค่าแรงดันอ้างอิงได้
- มีกราวด์แยกสำหรับวงจรฝั่งอนาล็อก และดิจิทัล
- มีโหมด Differential

ขาต่อใช้งานมีทั้งหมด 16 ขา เสียบลงโปรโตบอร์ด หรือแผ่นปริ้น์นอกประสงค์ได้พอดี ขาของไอซีจะแยกเป็น 2 ฝั่งชัดเจน คือฝั่งด้านอินพุต เริ่มตั้งแต่ขา 1 ช่องที่ 0 ไปจนถึงขา 8 ช่องที่ 7 และฝั่งด้านเอาต์พุต + จ่ายไฟเลี้ยง



รูปที่ 2.8 ขาของ MCP3208

การเชื่อมต่อเป็นแบบ SPI เมื่อ

- ขา CLK ให้ต่อเข้ากับ SCK หรือ CLK ของไมโครคอนโทรลเลอร์
- ขา DOUT ให้ต่อเข้ากับ MISO ของไมโครคอนโทรลเลอร์
- ขา DIN ให้ต่อเข้ากับ MOSI ของไมโครคอนโทรลเลอร์
- ขา CS ให้ต่อเข้ากับขา GPIO ใด ๆ ของไมโครคอนโทรลเลอร์ก็ได้ (ต้องกำหนดในโค้ดโปรแกรมว่าต่อขา CS ไว้ที่ขาอะไร)

การใช้เซ็นเซอร์โดยทั่วไปจะใช้แหล่งจ่ายไฟร่วมกับแหล่งจ่ายของไมโครคอนโทรลเลอร์ และใช้ร่วมกับไอซี ADC ด้วย ดังนั้นช่อง AGND และ DGND สามารถต่อร่วมกัน แล้วต่อเข้ากับขา GND ปกติได้เลย ขา VDD และ Vref (ขาแรงดันอ้างอิง) มักต่อใช้ร่วมกัน ดังนั้นจึงจ่ายไฟเข้าขาที่ต่อรวมนี้ได้เลย

2.5 ACS712 Current Sensor



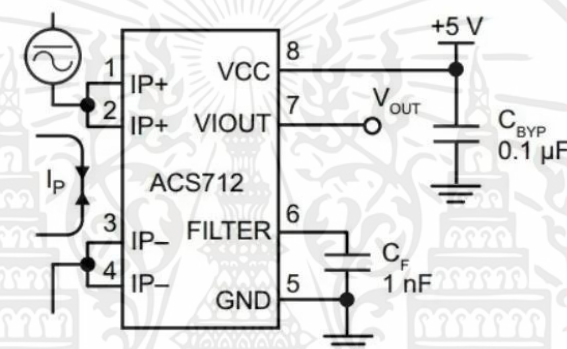
รูปที่ 2.9 ACS712

ACS712 คือเซ็นเซอร์ที่อาศัยหลักการวัดการเปลี่ยนแปลงของสนามแม่เหล็ก Hall effect สามารถวัดได้ทั้งกระแสตรง DC และกระแสสลับ AC ให้เอาต์พุตออกมาเป็นโวลต์สัมพันธ์กับกระแสที่วัด

ได้ มีฟังก์ชันป้องกันสัญญาณรบกวน , ตอนสนองต่อการวัดได้รวดเร็ว , เอาต์พุตมีความแม่นยำคลาดเคลื่อนประมาณ 1.5% สามารถเขียนโปรแกรมเพื่ออ่านค่าได้ง่าย

การทำงานของเซนเซอร์ ACS712 Current Sensor เซนเซอร์ ACS712 ประกอบด้วยวงจรถ่าย Hall Effect เอาต์พุตแบบเชิงเส้น มีวงจรถ่ายนำไฟฟ้าทองแดงตั้งอยู่รอบ ๆ พื้นผิวของตัวชิพ ACS712 เมื่อกระแส AC หรือ DC ผ่านเส้นทองแดงจะเกิดการสร้างสนามแม่เหล็ก สนามแม่เหล็กไฟฟ้านี้ทำปฏิกิริยากับเซนเซอร์ Hall Effect แปลงแม่เหล็กไฟฟ้านี้ เป็นแรงดันตามสัดส่วนของกระแสไฟฟ้าได้ทั้ง AC หรือ DC ขึ้นอยู่กับชนิดของกระแสไฟฟ้าอินพุต เราสามารถใช้บอร์ด Arduino หรือไมโครคอนโทรลเลอร์รุ่นอื่น ๆ วัดค่าแรงดันนี้ได้โดยตรงเพื่อแปลงค่ากลับมาเป็นกระแสไฟฟ้า

รูปขาของ ACS712 Pin Diagram ACS712 Hall Effect เซนเซอร์วัดกระแส



รูปที่ 2.10 ขาของ ACS712

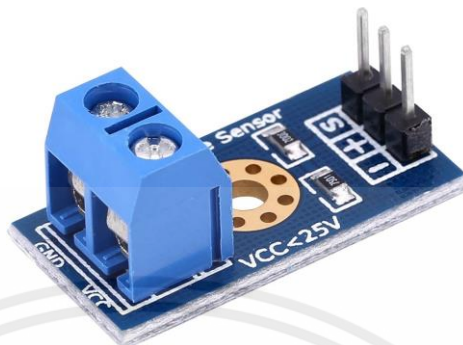
ขาของ ACS712 เซนเซอร์วัดกระแส ดังรูปนี้

- ขา 1,2 และ 3,4 ใช้สำหรับวัดกระแส เราสามารถเชื่อมต่อที่ขานี้กับโหนดที่ต้องการวัดได้
- ขา 6 กับ 5 ต่อ C สำหรับ filter ป้องกันสัญญาณรบกวน
- ขา 5 ไฟกราวด์
- ขา 8 ไฟเลี้ยง 5V
- ขา 7 เป็นขาเอาต์พุตที่วัดได้ หน่วยเป็นโวลต์

2.6 Voltage Sensor Module

Voltage Sensor Module โมดูลวัดแรงดันไฟฟ้า โวลต์เซ็นเซอร์ (DC 0-25V) โมดูลนี้ขึ้นอยู่กับหลักการออกแบบของตัวแบ่งตัวต้านทานทำให้แรงดันไฟฟ้าขาเข้าของอินเทอร์เฟซเทอร์มินัลแคบลงเท่าเท่า แรงดันไฟฟ้าอินพุตแบบอนาล็อก Arduino สูงถึง 5V จากนั้นแรงดันไฟฟ้าอินพุตของโมดูลตรวจจับแรงดันไฟฟ้าต้องไม่เกิน 25V เนื่องจากชิป Arduino AVR ใช้ใน 10 AD ดังนั้นความละเอียดการจำลอง

ของโมดูลนี้คือ 0.00489 V ดังนั้นโมดูลตรวจจับแรงดันไฟฟ้าอินพุตขั้นต่ำจึงตรวจจับแรงดันไฟฟ้า 0.02445V



รูปที่ 2.11 Voltage Sensor Module

โมดูลเซ็นเซอร์แรงดันไฟฟ้าในขณะที่วัดคุณสมบัตินี้

- ช่วงแรงดันไฟฟ้าขาเข้า DC 0-25V
- ช่วงการตรวจจับแรงดันไฟฟ้า DC 0.02445V-25V
- ความละเอียดอนาล็อกแรงดันไฟฟ้า 0.00489V

การต่อขาเซนเซอร์วัดแรงดันไฟ Voltage Sensor Module วิธีการต่อใช้งาน

- ขา - ต่อกับ กราวนด์
- ขา + ต่อกับไฟ 5 หรือ 3.3 โวลต์
- ขา S ต่อกับขา A1
- ขา Vcc ต่อกับไฟที่ต้องการวัด

ถ้าบอร์ด arduino ใช้ไฟเลี้ยง 3.3V ไฟที่วัดได้สูงสุดคือ 16.5V

ถ้าบอร์ด arduino ใช้ไฟเลี้ยง 5V ไฟที่วัดได้สูงสุดคือ 24.9V

2.7 BH1750FVI



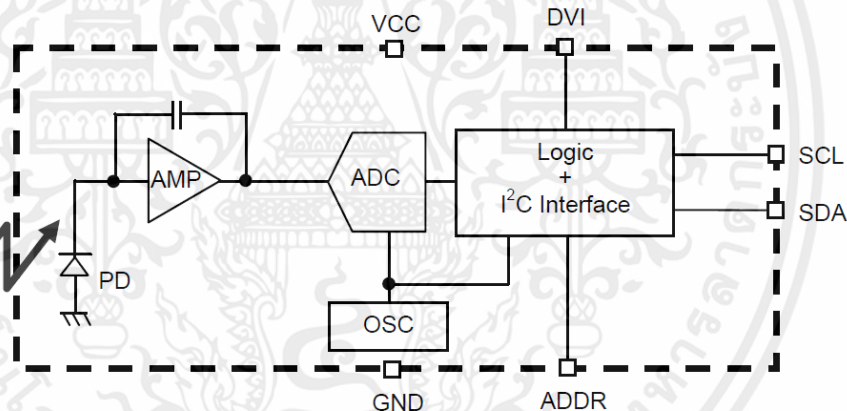
รูปที่ 2.12 BH1750FVI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BH1750FVI เป็น IC เซนเซอร์ตรวจจับแสงโดยรอบแบบดิจิทัลสื่อสารแบบ I2C เซนเซอร์นี้เหมาะสำหรับการใช้วัดค่าแสงแวดล้อม ข้อมูลสำหรับปรับไฟหน้าจอ LCD แบบอัตโนมัติตามความเข้มของแสงแวดล้อม เป็นเซนเซอร์วัดความเข้มแสง (Light Intensity) มีหน่วยการวัดเป็น "ลักซ์" (Lux หรือ Lumen / m²) สามารถวัดแสงที่สามารถมองเห็นได้ (Visible Light) และตอบสนองได้ดีสำหรับแสงในช่วงประมาณ 500 ถึง 600 นาโนเมตรไอซีสื่อสารข้อมูลได้ด้วยบัส I2C (ความเร็วได้ถึง 400kHz) ให้ข้อมูลจากการวัดแสงที่มีความละเอียด 16 บิต

มีคุณสมบัติดังนี้

- ใช้แรงดันไฟเลี้ยงได้ในช่วง 3 V ถึง 5 V
- ตรวจจับความสว่างได้ที่ 1 ถึง 65,535 lux
- กินกระแสไฟน้อยลง 0.12 mA
- ความแม่นยำ +/- 20%
- เชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ผ่าน I2C



รูปที่ 2.13 BH1750 Block Diagram

ภายในไอซี BH1750 ประกอบด้วยส่วนต่าง ๆ เริ่มต้นด้วย PD (Photo Diode) สำหรับรับแสง เมื่อได้รับแสงก็จะมีกระแสไฟฟ้าไหล มีการแปลงกระแสให้เป็นแรงดันไฟฟ้าโดยใช้วงจร Opamp-based Integrator แล้วแปลงให้เป็นข้อมูลดิจิทัล โดยใช้วงจร ADC (Analog-to-Digital Converter) และมีหน่วยประมวลผลแบบดิจิทัลสำหรับเชื่อมต่อสื่อสารข้อมูลด้วยบัส I2C ไอซีมีขา ADDR หนึ่งขาสำหรับกำหนดแอดเดรสของอุปกรณ์ (0x23: ADDR=0 / floating และ 0x5C: ADDR=1 ต่อกับ +3.3V)

ในการวัดค่าแสง แบ่งออกได้เป็นสองรูปแบบการทำงาน คือ

- One-Time Measurement เป็นการวัดค่าแสงเพียงหนึ่งครั้งแล้วไอซีจะเข้าสู่โหมดประหยัดพลังงาน (Powerdown) โดยอัตโนมัติ
- Continuous Measurement เป็นการวัดค่าแสงต่อเนื่องเข้าไปเรื่อย ๆ

2.8 LCD



รูปที่ 2.14 LCD I2C

LCD ย่อมาจากคำว่า Liquid Crystal Display ซึ่งเป็นจอที่ทำมาจากผลึกคริสตอลเหลว หลักการคือด้านหลังจอจะมีไฟส่องสว่าง หรือที่เรียกว่า Backlight อยู่ เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไป กระตุ้นที่ผลึก ก็จะทำให้ผลึกโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสดงขึ้นมาบนหน้าจอ ส่วนอื่นที่ โดนผลึกปิดกั้นไว้ จะมีสีที่แตกต่างกันตามสีของผลึกคริสตอล เช่น สีเขียว หรือ สีฟ้า ทำให้เมื่อมองไปที่จอ ก็จะพบกับตัวหนังสือสีขาว แล้วพบกับพื้นหลังสีต่างๆกัน

จอ LCD จะแบ่งเป็น 2 แบบใหญ่ๆตามลักษณะการแสดงผลดังนี้

1. Character LCD เป็นจอที่แสดงผลเป็นตัวอักษรตามช่องแบบตายตัว เช่น จอ LCD ขนาด 16x2 หมายถึงใน 1 แถว มีตัวอักษรใส่ได้ 16 ตัว และมีทั้งหมด 2 บรรทัดให้ใช้งาน ส่วน 20x4 จะหมายถึงใน 1 แถว มีตัวอักษรใส่ได้ 20 ตัว และมีทั้งหมด 2 บรรทัด
2. Graphic LCD เป็นจอที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอขึ้นแสง หรือปล่อยแสง ออกไป ทำให้จอนี้สามารถสร้างรูปขึ้นมาบนหน้าจอได้ การระบุขนาดจะระบุในลักษณะของ จำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 จุด และมีจุดตามแนวตั้ง 64 จุด

การเชื่อมต่อกับจอ Character LCD การเชื่อมต่อจะมีด้วยกัน 2 แบบ คือ

- การเชื่อมต่อแบบขนาน - เป็นการเชื่อมต่อจอ LCD เข้ากับบอร์ด Arduino โดยตรง โดยจะแบ่งเป็นการเชื่อมต่อแบบ 4 บิต และการเชื่อมต่อแบบ 8 บิต ใน Arduino จะนิยมเชื่อมต่อแบบ 4 บิต เนื่องจากใช้สายในการเชื่อมต่อน้อยกว่า
- การเชื่อมต่อแบบอนุกรม - เป็นการเชื่อมต่อกับจอ LCD ผ่านโมดูลแปลงรูปแบบการเชื่อมต่อกับจอ LCD จากแบบขนาน มาเป็นการเชื่อมต่อแบบอื่นที่ใช้สายน้อยกว่า เช่น การใช้โมดูล I2C Serial Interface จะเป็นการนำโมดูลเชื่อมเข้ากับตัวจอ LCD แล้วใช้บอร์ด Arduino เชื่อมต่อกับบอร์ดโมดูลผ่านโปรโตคอล I2C ทำให้ใช้สายเพียง 4 เส้น ก็ทำให้หน้าจอแสดงผลข้อความต่างๆออกมาได้

Pin No	Symbol	Description
1	GND	Ground
2	VCC	+5VDC
3	SDA	Serial Data
4	SCL	Serial Clock

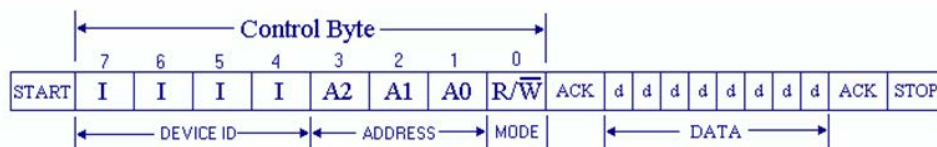
รูปที่ 2.15 ขาของ LCD I2C

การควบคุมการแสดงผลของจอ LCD (I2C)

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) อยู่ในตัวแล้ว ผู้ใช้สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD (I2C) เช่นเดียวกับกับจอ LCD แบบธรรมดา พุดง่ายๆ คือรหัสคำสั่งที่ใช้ในการควบคุมนั้นเหมือนกัน แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล จอ LCD จะมีการส่งข้อมูลรูปแบบ I2C ที่ใช้ขาเพียง 4 ขาที่ใช้ในการเชื่อมต่อเท่านั้น

- GND เป็น Ground ใช้ต่อระหว่าง Ground ของระบบ Microcontroller กับ LCD
- VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับ LCD มีขนาด +5VDC
- SDA (Serial Data) เป็นขาที่ใช้ในการรับส่งข้อมูล
- SCL (Serial Clock) เป็นขาสัญญาณนาฬิกาในการรับส่งข้อมูล

การรับ-ส่งข้อมูลแบบ I2C BUS



รูปที่ 2.16 การรับ-ส่งข้อมูลแบบ I2C

MCU จะทำการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส แล้วตามด้วยรหัสควบคุม (Control Byte) ซึ่งประกอบด้วยรหัสประจำตัวอุปกรณ์ Device ID, Device Address และ Mode ในการเขียนหรืออ่านข้อมูล เมื่ออุปกรณ์รับทราบว่า MCU ต้องการจะติดต่อกับ ก็ต้องส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง และเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่า สิ้นสุดการใช้บัส

2.9 ตัวต้านทานไว้วาวด์หุ้มอิทซิงค์ (Aluminium Housed Resistor)



รูปที่ 2.17 ตัวต้านทานไว้วาวด์

ตัวต้านทานหุ้มอิทซิงค์เป็นตัวต้านทานไว้วาวด์ชนิดหนึ่งเรียกชื่อตามวัสดุที่ใช้หุ้มคืออิทซิงค์ อิทซิงค์ช่วยให้ตัวต้านทานระบายความร้อนได้ดีขึ้นมาก ทำให้วัตต์ตัวต้านทานเพิ่มขึ้นประมาณ 40-50% เช่น ตัวต้านทานหุ้มอิทซิงค์ 10W ถ้าไม่หุ้มอิทซิงค์จะลดเหลือเพียง 5.5W ตัวต้านทานหุ้มอิทซิงค์ 50W ถ้าไม่หุ้มอิทซิงค์จะลดเหลือเพียง 20W เป็นต้น การใช้งานมีใช้งานทั่วไป และใช้งานในระบบควบคุมอุตสาหกรรมสีของตัวต้านทานชนิดนี้ส่วนใหญ่เป็นสีทองและอาจมีสีเงินอยู่บ้าง ตัวต้านทานหุ้มอิทซิงค์วัตต์ต่ำจะมีขาแบบบัดกรีถ้าวัตต์สูงจะมีขาแบบขันน็อต ที่ตัวต้านทานจะพิมพ์ค่าความต้านทาน ค่า

วัตต์ และ % ความคลาดเคลื่อนไว้ทำให้ง่ายในการเลือกใช้งานและหารุ่นทดแทน

คุณสมบัติของตัวต้านทานหุ้มอิทซิงค์

- มีค่าให้เลือกใช้งานตั้งแต่ค่าโอห์มต่ำจนถึงค่า K ohm
 - มีค่าวัตต์ให้เลือกใช้งาน 5W - 300W
 - ค่าสัมประสิทธิ์ความต้านทานต่ออุณหภูมิเสถียรดีมาก $<\pm 1\%$ $\pm 1\%$ $\pm 3\%$ $\pm 5\%$ $\pm 10\%$
- อิทซิงค์และสารนำความร้อนที่อยู่ข้างในช่วยให้ตัวต้านทานระบายความร้อนได้อย่างรวดเร็ว
- ป้องกันตัวต้านทานจากแวดล้อมได้ดีมาก โครงสร้างและชิ้นส่วนที่เชื่อมและหุ้มปิดช่วยป้องกันฝุ่นและความชื้นไม่ให้เข้าไปข้างในได้ (กรณีป้องกันไม่ดีความชื้นจะทำให้คุณสมบัติของวัสดุตัวต้านทานข้างในเสื่อมค่าและเปลี่ยนค่า)
 - ทนสัญญาณพัลส์ได้ดีเยี่ยม

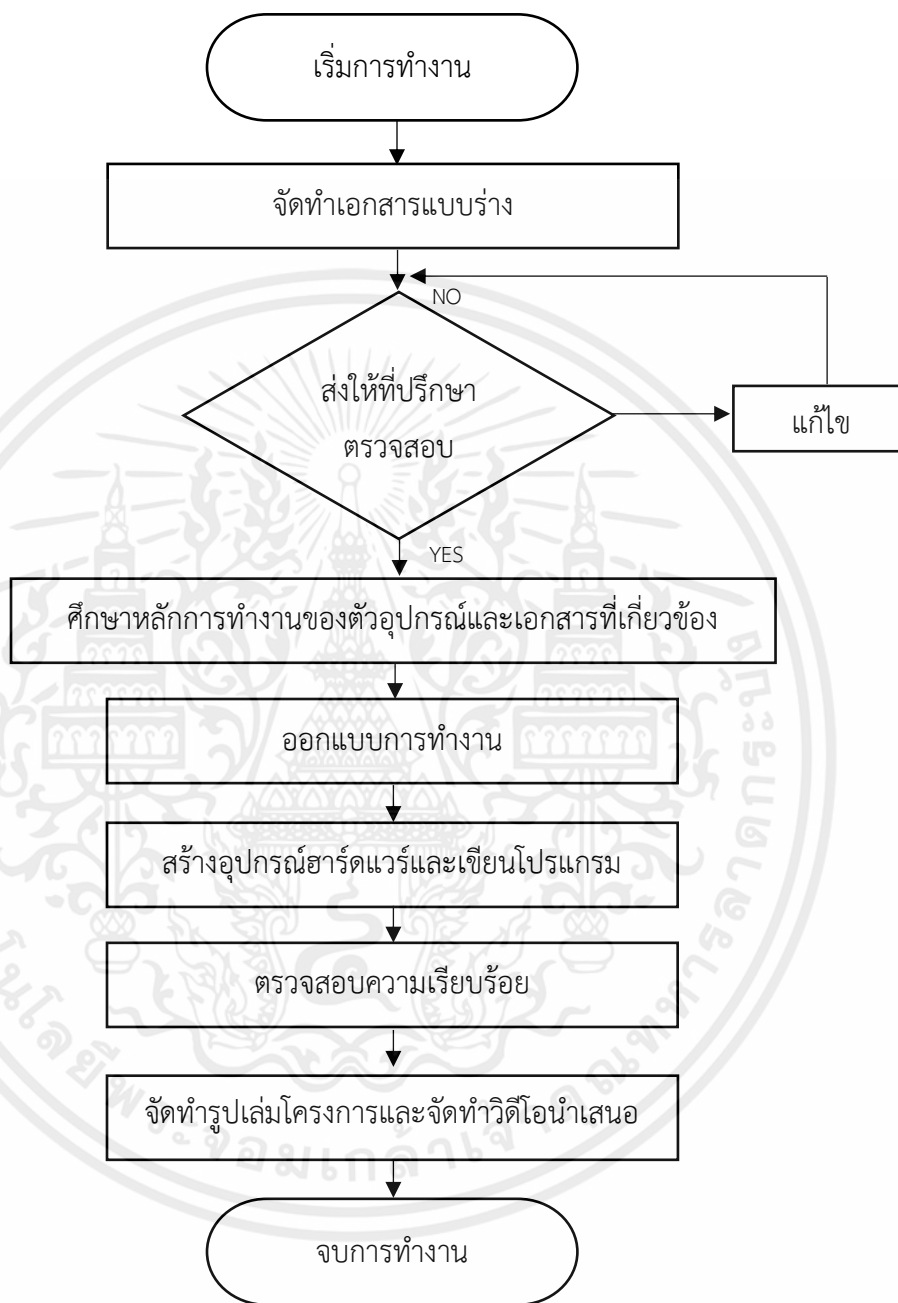
บทที่ 3

หลักการงานและการออกแบบ

3.1 วัสดุ อุปกรณ์ เครื่องมือหรือโปรแกรมที่ใช้

- NodeMCU ESP8266
- DHT 22
- BH1750FVI
- MCP3208
- LCD I2C ขนาด 20*4
- ตัวต้านทานไวร์วาวด์ 4Ω 100W
- ACS712 Current Sensor
- Voltage Sensor Module
- Resistor
- กล่องบรรจุอุปกรณ์
- เว็บไซต์ Thingspeak
- เว็บไซต์ Wordpress
- โปรแกรม Arduino IDE

3.2 ขั้นตอนการดำเนินการ



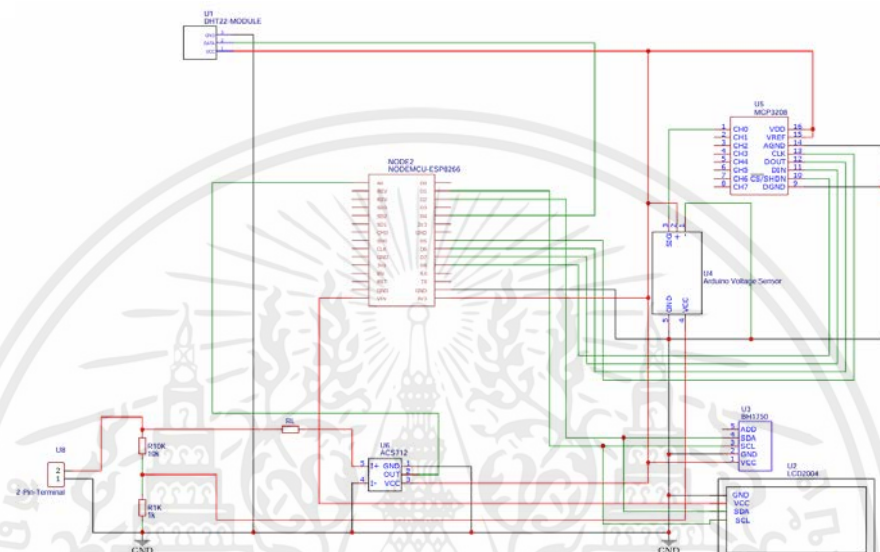
รูปที่ 3.1 แผนภาพ Flowchart ขั้นตอนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

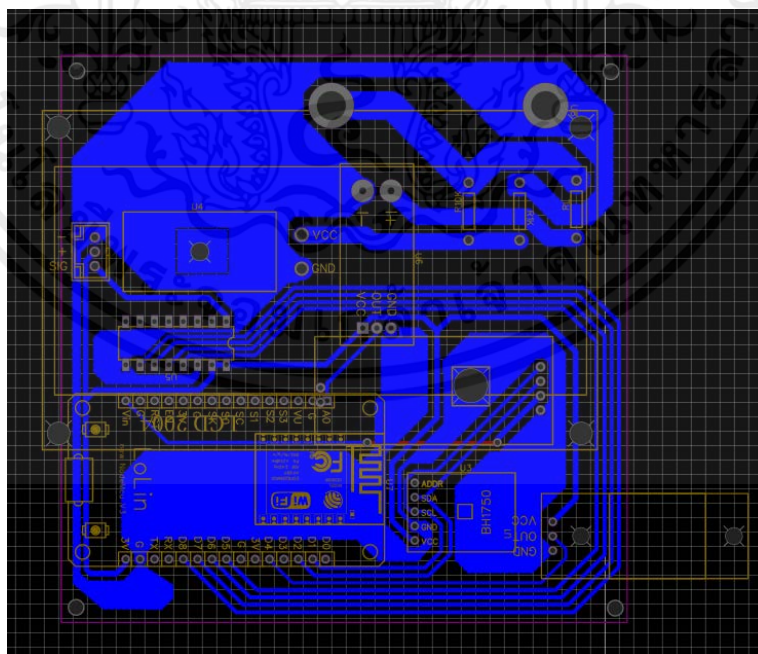
3.3 การออกแบบและจัดทำระบบ

การออกแบบวงจรนี้แบ่งออกเป็น 2 ส่วน คือ การออกแบบทางด้าน Hardware และการออกแบบทางด้าน Software

3.3.1 การออกแบบทางด้าน Hardware

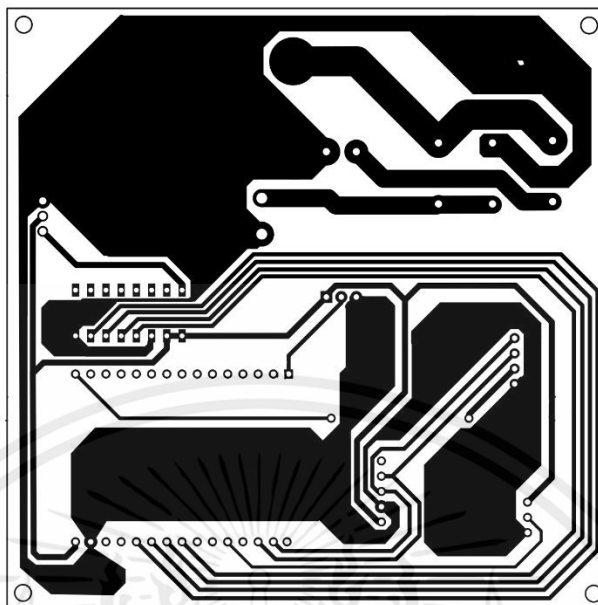


รูปที่ 3.2 Schematics



รูปที่ 3.3 การออกแบบแผ่น PCB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แผ่น PCB ก่อนทำการกัดปริน

3.3.2 การออกแบบทางด้าน Software

3.3.2.1 Arduino IDE

Arduino IDE เป็นชื่อเรียกของแพลตฟอร์มการเขียนโปรแกรมลง ไมโครคอนโทรลเลอร์ (Arduino) ที่เราสามารถสร้างสรรค์สิ่งต่าง ๆ จากการออกแบบได้ตามความคิด แล้วสร้างออกมาเป็นชิ้นงานที่ใช้ได้จริง โดยไม่จำเป็นต้องมีความรู้เรื่องการต่อวงจรที่ซับซ้อน การเขียนโปรแกรมที่เข้าใจง่ายด้วยภาษา C , C++ มีโค้ดตัวอย่างและโปรเจกต์ให้ศึกษาจำนวนมาก บอร์ดทดลองราคาไม่แพง มีให้เลือกใช้งานตามความเหมาะสม มีเซนเซอร์โมดูล และอุปกรณ์เสริมจำนวนมาก แพลตฟอร์มเป็นแบบโอเพ่นซอร์ส สามารถใช้งานได้ฟรี ด้วยความง่ายครบถ้วนและฟรีนี้ จึงทำให้มีผู้ใช้งานทั่วโลก และมีการพัฒนาให้ดีขึ้นอย่างต่อเนื่อง

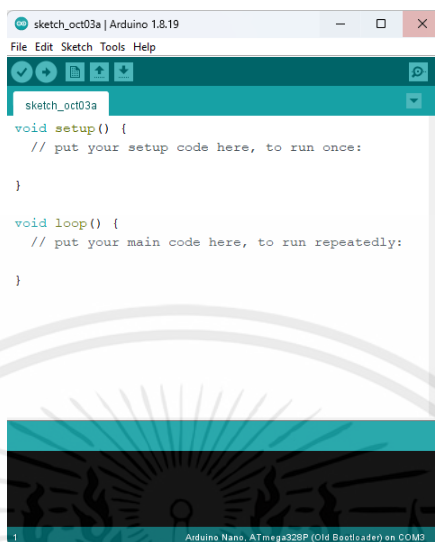
แนวคิดการใช้งานโปรแกรม Arduino IDE

1. เขียนโปรแกรมด้วยภาษา C/C++ สำหรับ Arduino
2. คอมไพล์หรือแปลโปรแกรมภาษา C/C++ ให้เป็นภาษาสำหรับไมโครคอนโทรลเลอร์และบันทึกเป็น Intel Hex File
3. อัปโหลด Intel Hex File ลงบนไมโครคอนโทรลเลอร์ซึ่งอยู่บนบอร์ด Arduino ผ่านสาย USB หรือผ่าน Programmer

วิธีใช้งานโปรแกรม Arduino IDE

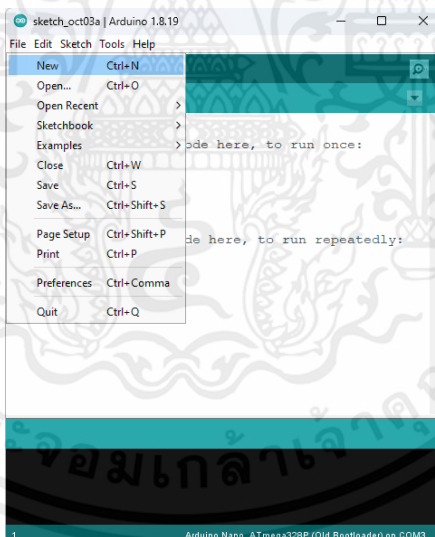
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เปิดโปรแกรม Arduino IDE ขึ้นมา



รูปที่ 3.5 หน้าแรกของโปรแกรม Arduino IDE

2. สร้าง Sketch ใหม่ โดยคลิกที่เมนู File > New



รูปที่ 3.6 หน้าต่าง File ของโปรแกรม Arduino IDE

3. เขียนโปรแกรมลงไปตรงพื้นที่สีขาวๆ ครับ ซึ่งการเขียนโปรแกรมสำหรับ Arduino จะประกอบไปด้วยฟังก์ชัน setup และ ฟังก์ชัน loop ซึ่งมีความหมายดังนี้

- ฟังก์ชัน setup จะเริ่มทำงานเป็นอันดับแรกเมื่อ Arduino เริ่มทำงาน และคำสั่งที่ถูกเขียนลงไป ในนี้ จะทำงานเพียงครั้งเดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชัน loop จะเริ่มทำงานทันทีเมื่อฟังก์ชัน setup ทำงานเสร็จ และคำสั่งที่ถูกเขียนลงไปนั้น จะทำงานตั้งแต่คำสั่งแรก ไล่ไปจนถึงคำสั่งสุดท้าย แล้วกลับมาที่คำสั่งแรก ไล่ไปจนถึงคำสั่งสุดท้าย วนซ้ำแบบนี้ไปเรื่อยๆ ตลอดการทำงานของ Arduino

```

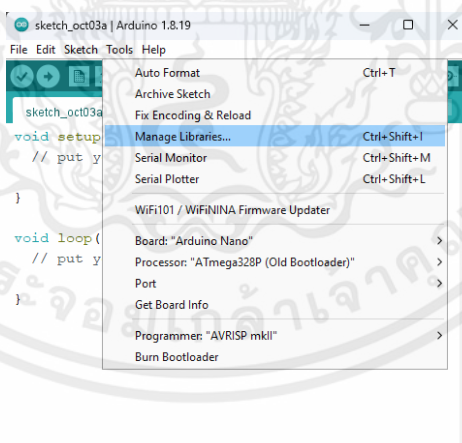
sketch_oct03a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

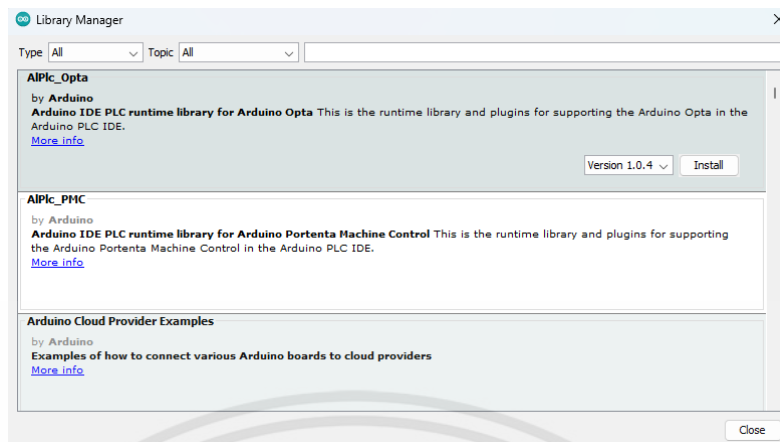
```

รูปที่ 3.7 ส่วนที่ทำการเขียนโปรแกรม

4. ในการ Programming บางครั้งเราอาจจะต้องเพิ่ม Library ให้กับตัวโปรแกรมเพื่อที่จะสามารถเรียกใช้โค้ดเฉพาะของอุปกรณ์ตัวต่างๆ ได้ ซึ่งบางอันสามารถหาโหลดได้จากโปรแกรม Arduino IDE ได้เลย โดยไปที่เมนู Tools > Manage Libraries แต่บางอันที่ไม่มีก็จะต้องหาโหลดจากเว็บไซต์ต่างๆ เช่น GitHub เป็นต้น

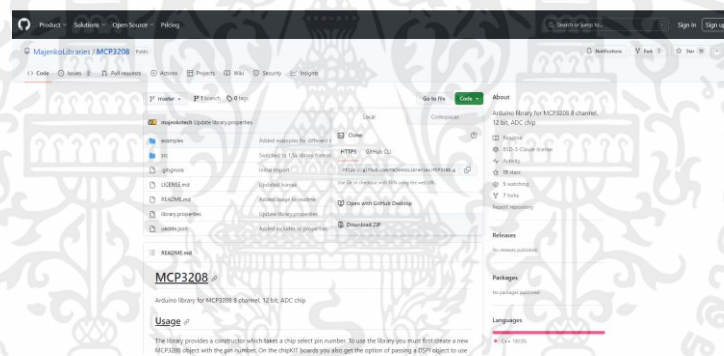


รูปที่ 3.8 หน้าต่าง Tools ของโปรแกรม Arduino IDE



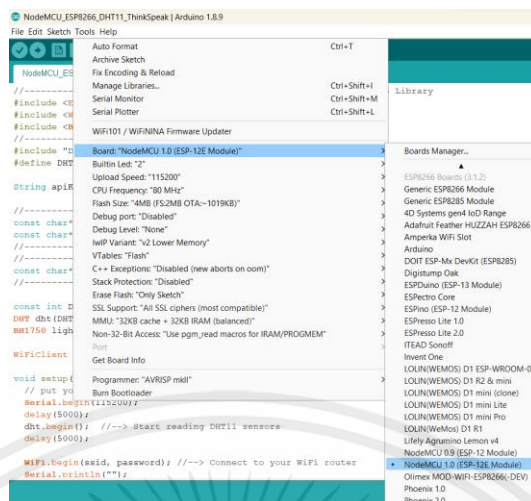
รูปที่ 3.9 หน้าต่าง Library Manager

หลังจากเราเจอ Library ที่เราต้องการใน GitHub ก็ทำการดาวน์โหลดโดยการกดที่เมนู Code > Download ZIP จากนั้นทำการแตกไฟล์และทำไปใส่ในโฟลเดอร์ Libraries ของ Arduino IDE



รูปที่ 3.10 หน้าโฮสต์ Libraries ของเว็บไซต์ GitHub

5. หลังจากที่เราทำการลง Libraries และเขียนโค้ดเสร็จแล้ว ก็ทำการเลือก Board ที่เราจะทำการอัปโหลดโค้ดลงไปโดยไปที่เมนู Tools > Board โดยจะเป็น Board ที่เราได้ดาวน์โหลดเอาไว้



รูปที่ 3.11 เลือก Board โดยการไปที่เมนู Tools > Board

6. การนั้นก็ทำการ Verify และอัปโหลดลง Arduino ก็เป็นการเสร็จสิ้น

3.3.2.2 WordPress

WordPress คือ เครื่องมือสร้างเว็บไซต์ แบบ Zero code เราสามารถออกแบบเว็บ เพิ่มเติมบทความ รูปภาพ ได้เองแบบง่ายๆ โดยไม่ต้องมีความรู้เรื่อง Code หรือความรู้ด้านโปรแกรมมิ่งเลย ทำให้ถึงควรใช้ WordPress

○ ประหยัดค่าใช้จ่าย

ใครๆ ก็สามารถโหลด WordPress มาใช้ได้ฟรี เพราะเป็น Opensource ดังนั้นเราจึงประหยัดค่าใช้จ่ายในการจ้างคนเขียนโปรแกรม เราสามารถเรียนรู้และทำด้วยตนเองได้เลย

○ ไม่ต้องเขียนโค้ดให้ยุ่งยาก

การสร้างเว็บด้วย WordPress เราไม่จำเป็นต้องมีทักษะ coding แต่เขาจะมีชิ้นส่วนต่างๆ เตรียมไว้ให้เรา เราสามารถนำชิ้นส่วนเหล่านั้น มาลากและวาง ประกอบร่างเป็นเว็บได้อย่างที่ใจต้องการเลย

○ มีฟังก์ชันต่างๆ มากมาย

ทุกระบบ ทุกฟังก์ชันที่เราต้องการ จะมีผู้สร้างปลั๊กอิน WordPress จากทั่วโลก ทำเตรียมไว้ให้ เราก็แค่ไปค้นหา และหยิบมาใส่บนเว็บ WordPress ของเราได้เลย เช่น ระบบ Ecommerce ระบบ Booking เป็นต้น

○ อัปเดตเนื้อหาต่างๆ ได้ง่าย

เพราะ WordPress ใช้งานง่าย ใช้งานได้บนทุกอุปกรณ์ คุณสามารถเพิ่มข้อมูลหรืออัปเดตเนื้อหาได้ตลอดทุกที่ ทุกเวลา แก่จุดไหนผิด หรือเผลอลบข้อมูลแบบไม่ตั้งใจ ตัว WordPress ยังมีความสามารถ restore ภู้เนื้อหาเดิมกลับมาได้อีกด้วย

○ มีธีมให้เลือกเยอะ ดีไซน์สวยงาม

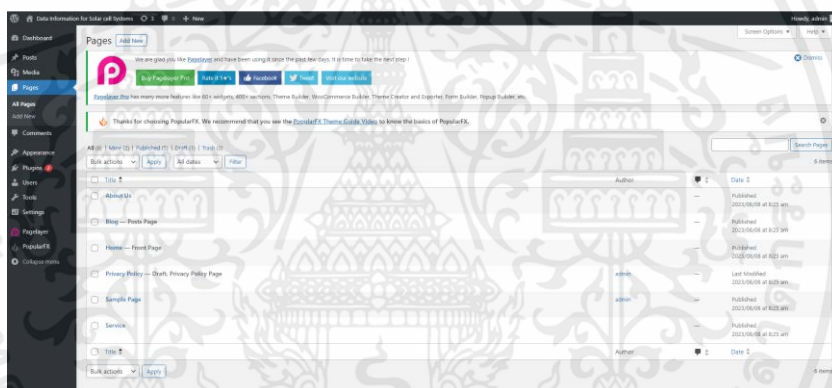
เราสามารถใส่ WordPress ออกแบบเว็บให้สวยๆ ได้ทุกจินตนาการ หรือหากคุณไม่มีไอเดียในการออกแบบเว็บ แต่ละธีมของ WordPress เขาจะมี Template การดีไซน์เว็บ ที่ออกแบบไว้อย่างสวยงาม มาให้เราเลือกใช้อีกด้วย

○ ทำ SEO ง่าย

WordPress มีฟังก์ชันสำหรับทำ SEO เตรียมไว้แบบครบวงจร และมีโครงสร้างเว็บที่เป็นมิตรกับการให้ Google เก็บข้อมูล ซึ่ง Google เองยังแนะนำให้คนทั่วไปที่ไม่เก่ง SEO ควรเลือกใช้ WordPress ในการสร้างเว็บ

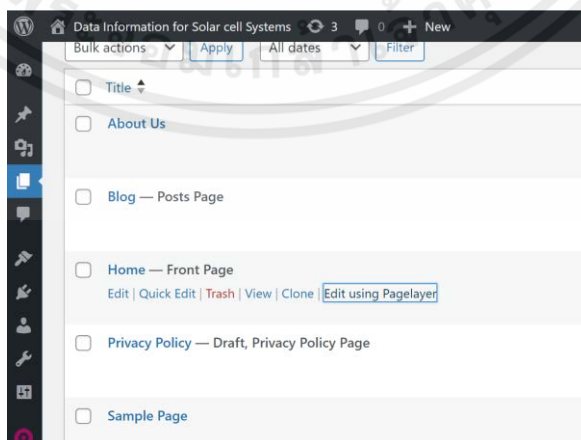
วิธีการใช้งาน WordPress

1. เมื่อ Login เข้ามาแล้ว ก็กดไปที่เมนู Pages ตรง Control panel ของ WordPress ก็จะสามารถเพิ่ม Pages หรือแก้ไข Pages ต่างๆ ได้ดังรูป



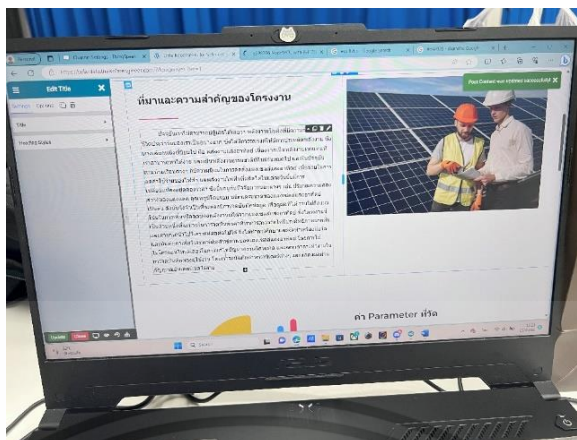
รูปที่ 3.12 เมนู Pages

2. หากเราต้องการแก้ไขหน้า Home ก็ทำการกด Edit using Pagelayer ก็จะสามารถเปลี่ยนได้หมดทุกอย่างไม่ว่าจะเป็นข้อความ รูปภาพ หรือพื้นหลัง



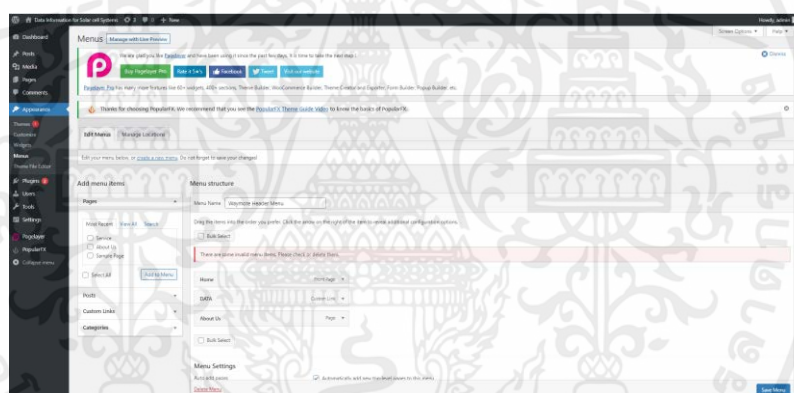
รูปที่ 3.13 กด Edit using Pagelayer เพื่อทำการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 ตัวอย่างการแก้ไขหน้า Home

3. และเมื่อทำ Pages ที่ต้องการเสร็จหมดแล้วสามารถมาเลือก Pages ที่ต้องการแสดงได้ที่ Appearance > Menu เลือก Pages ที่ต้องการ แล้วกด Add to Menu



รูปที่ 3.15 หน้า Menu

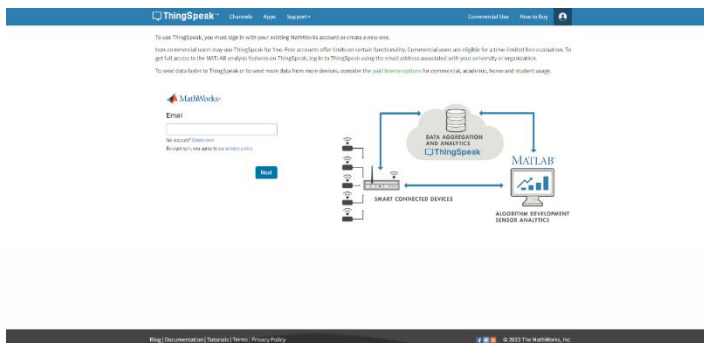
3.3.2.3 ThingSpeak

ThingSpeak คือผู้ให้บริการ Cloud ชนิดหนึ่ง ที่อนุญาตให้เราสามารถส่งค่าต่างๆขึ้นเพื่อไปเก็บบนพื้นที่ ที่เปิดให้เราใช้บริการ และยังเปิดให้เราสามารถเข้าถึงข้อมูลเหล่านี้ได้จากเว็บเบราว์เซอร์ต่างๆไปจากที่ไหนก็ได้ที่มี Internet จึงนับเป็นการเข้าถึงข้อมูลในรูปแบบ IOT (Internet of Things)

วิธีการใช้งาน ThingSpeak

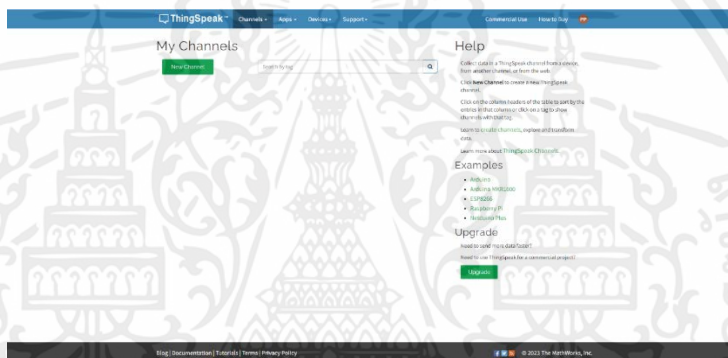
1. ต้องสมัครใช้งาน ThingSpeak โดยเข้าไปที่ <https://thingspeak.com/> แล้วกดที่ปุ่ม “Get Started For Free” เพื่อสมัครใช้งาน
2. หลังจากทำการสมัครเสร็จทำการเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



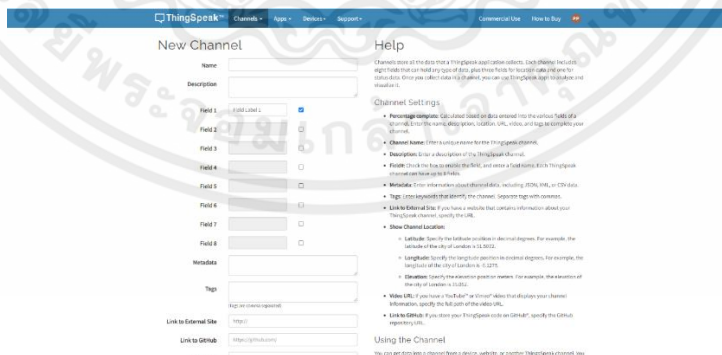
รูปที่ 3.16 หน้าเข้าสู่ระบบของเว็บไซต์ ThinkSpeak

3. ทำการสร้าง Channel ใหม่ด้วยการกด New Channel



รูปที่ 3.17 หน้าสร้าง My Channels

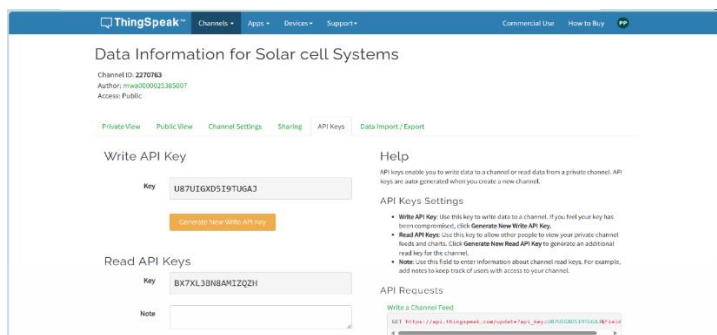
4. ตั้งชื่อ Channel และเลือกกับตั้งชื่อ Field ที่เราต้องการใช้งาน โดยเราสามารถเลือกได้สูงสุด 8 Fields



รูปที่ 3.18 หน้าตั้งชื่อ Channel และ Field

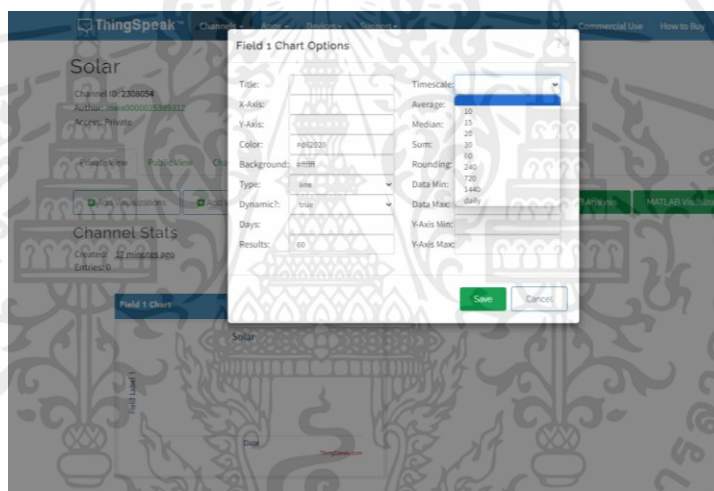
5. เมื่อสร้างเสร็จ เข้าไปในส่วนของ API Keys ซึ่งในแต่ละ Channel จะมี API key เฉพาะตัว ทำให้เราสามารถส่งข้อมูลไปยัง Channel ของเราได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

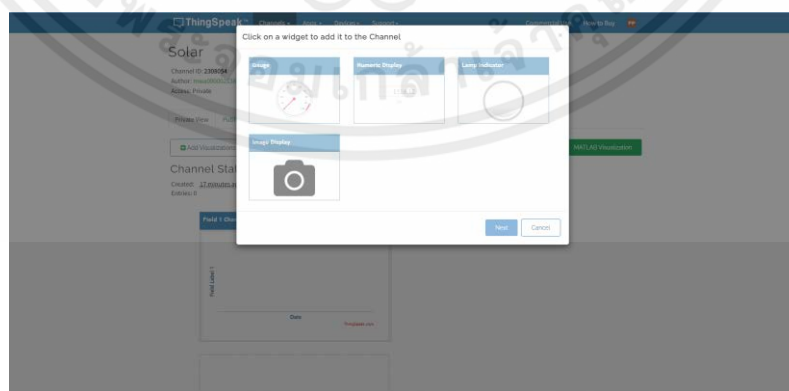


รูปที่ 3.19 หน้าดู API Keys ของ ThingSpeak

6. สามารถปรับ Timescale ปรับสีต่างๆ ของกราฟและสามารถเพิ่ม Widget ต่างๆ ได้ผ่าน Add Widgets



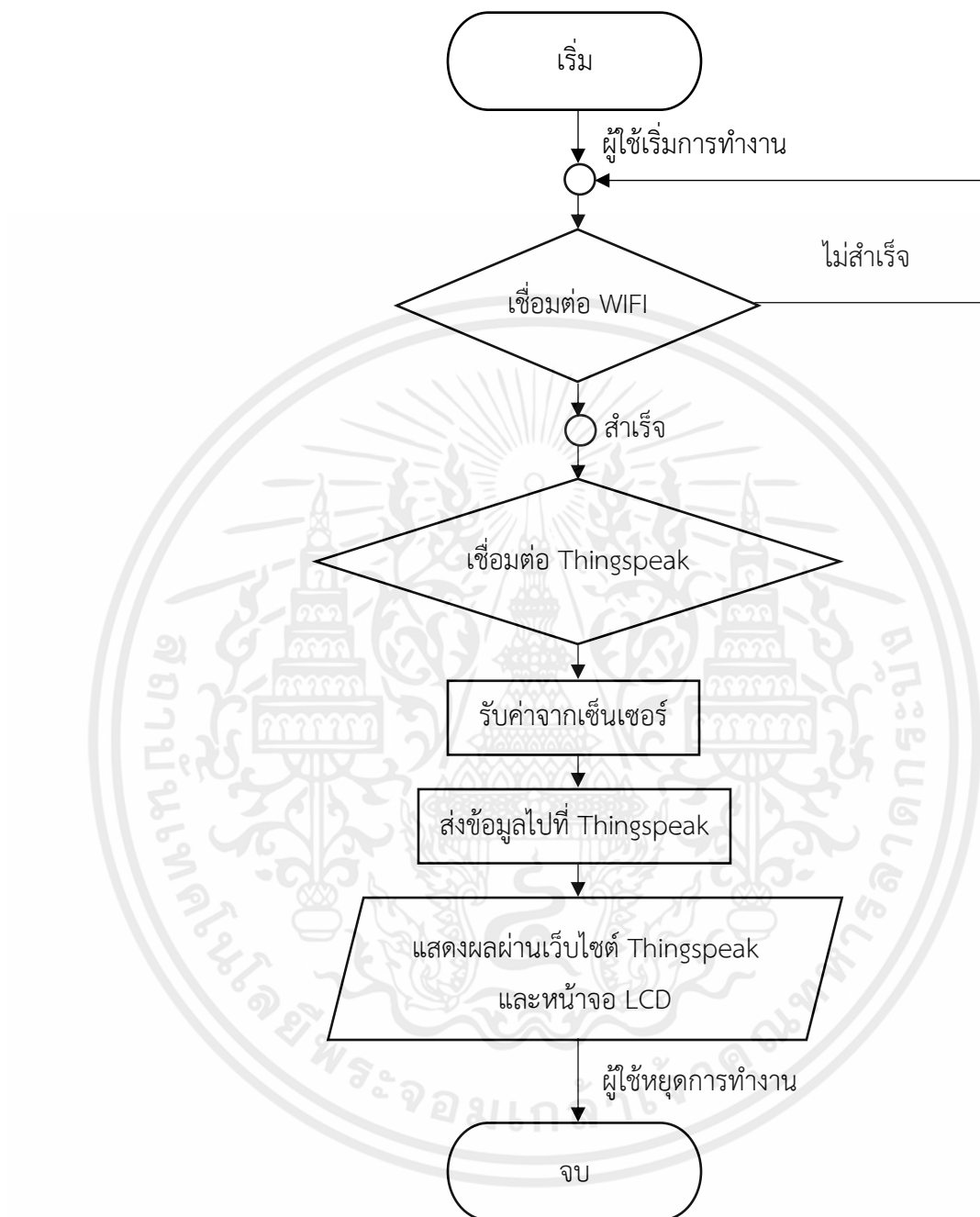
รูปที่ 3.20 หน้าต่างปรับ Timescale และสีของกราฟ



รูปที่ 3.21 หน้าต่าง Add Widgets

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ผังงาน (Flowchart)



รูปที่ 3.22 ผังงานแสดงขั้นตอนการทำงาน

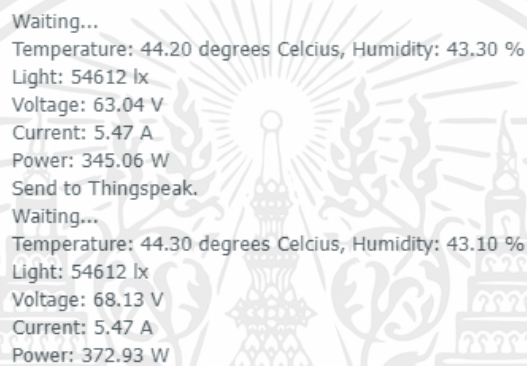
บทที่ 4

ผลการทดลอง

4.1 การแสดงผลด้าน Software

4.1.1 วัดค่าพารามิเตอร์ต่างๆ โดยแสดงผลผ่านหน้าจอ Serial Monitor

เปิดโปรแกรม Arduino IDE ที่เขียนคำสั่งไว้ จากนั้นเรียกใช้ Serial Monitor โดยคลิกที่ Tools > Serial Monitor หรือกดปุ่ม Ctrl + Shift + M



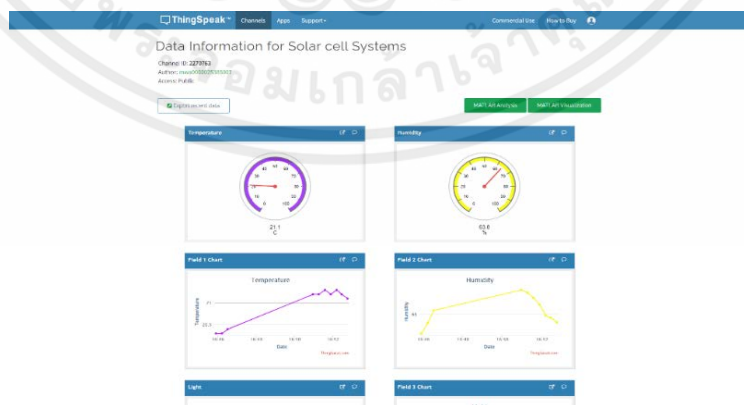
Waiting...
Temperature: 44.20 degrees Celcius, Humidity: 43.30 %
Light: 54612 lx
Voltage: 63.04 V
Current: 5.47 A
Power: 345.06 W
Send to Thingspeak.
Waiting...
Temperature: 44.30 degrees Celcius, Humidity: 43.10 %
Light: 54612 lx
Voltage: 68.13 V
Current: 5.47 A
Power: 372.93 W

รูปที่ 4.1 หน้า Serial Monitor ในโปรแกรม Arduino IDE

4.2 การแสดงผลจากเว็บไซต์

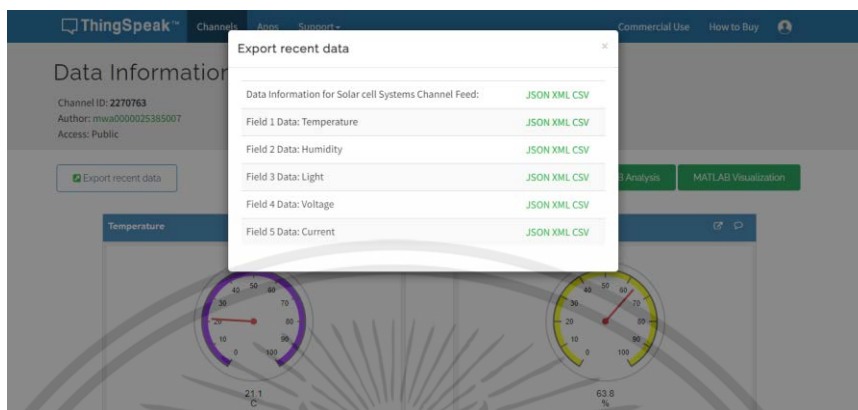
4.2.1 อ่านค่าพารามิเตอร์ต่างๆ จาก Serial Monitor และดูค่าที่ได้จากเว็บไซต์

Thingspeak



รูปที่ 4.2 หน้าต่างแสดงผลค่า ณ ปัจจุบัน และกราฟแสดงค่าก่อนหน้า

4.2.2 นอกจากนี้ยังสามารถ Export Data ใน Field ต่างๆ หรือทั้งหมดออกมาเป็นไฟล์ Excel ได้



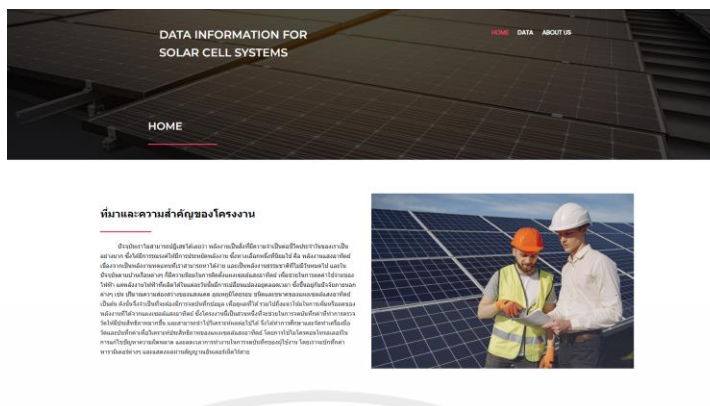
รูปที่ 4.3 หน้าต่าง Export recent data

84	2023-10-10	191	20.8	61.9	133.33	0.19
85	2023-10-10	192	20.8	61.6	131.67	0.15
86	2023-10-10	193	20.7	61.3	130	0.19
87	2023-10-10	194	20.5	62.9	104.17	0.19
88	2023-10-10	195	20.5	62.4	105	0.27
89	2023-10-10	196	20.5	62.7	101.67	12.36
90	2023-10-10	197	20.4	62.4	95.83	14.96
91	2023-10-10	198	20.3	62.5	100	14.96
92	2023-10-10	199	20.3	62.1	100	15.00
93	2023-10-10	200	20.3	63.7	92.5	0.19
94	2023-10-10	201	20.4	65.6	104.17	0.27
95	2023-10-10	202	21.2	68.7	114.17	14.77
96	2023-10-10	203	21.2	68.3	90.83	14.73
97	2023-10-10	204	21.3	67.5	93.33	14.80
98	2023-10-10	205	21.2	66.5	100	14.80
99	2023-10-10	206	21.3	64.8	90.83	14.88
100	2023-10-10	207	21.2	64.5	90.83	15.30
101	2023-10-10	208	21.1	63.8	93.33	15.42

รูปที่ 4.4 ตัวอย่าง Data ที่เก็บใน Excel

4.2.3 เว็บไซต์ที่ใช้ในการแสดงผลเกี่ยวกับประวัติความเป็นมา ที่มาและความสำคัญ และการแสดงข้อมูลต่างๆ ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

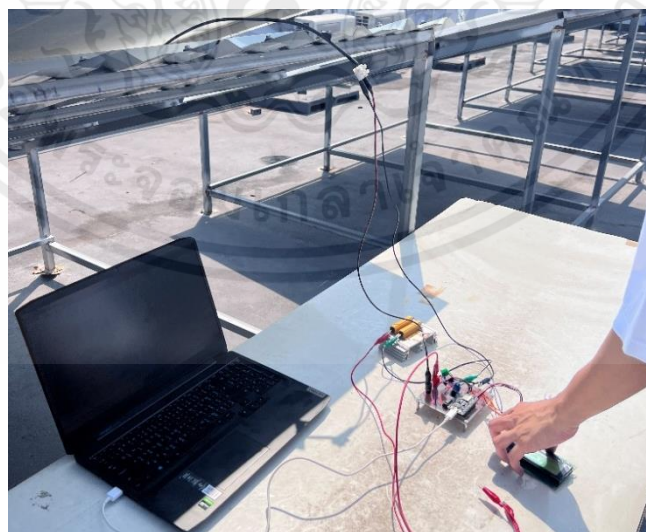


รูปที่ 4.5 หน้าจอหลักของเว็บไซต์ Wordpress

<https://solardata.thaisolarengineer.com/>



รูปที่ 4.6 ลองนำไปต่อกับ Power supply ก่อนใช้งานจริง



รูปที่ 4.7 นำไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผล

ในปฏิญญาฉบับนี้มีการศึกษาและออกแบบการใช้ Volt sensor module , Current sensor module , Digital Temperature and Humidity Sensor และ Light sensor ร่วมกับ Node mcu esp8266 เพื่อมาประยุกต์ใช้ในการวัดแรงดันไฟฟ้า , กระแสไฟฟ้า , อุณหภูมิ , ความชื้น และความเข้มแสง เพื่อแสดงผลผ่านเว็บไซต์โดยการออกแบบจะแบ่งเป็น 2 ส่วน คือ ส่วนที่ 1 คือด้าน software เป็นการสร้างเว็บไซต์ Website และนำ data จาก data base มาแสดงบนเว็บไซต์ ส่วนที่ 2 จะเป็น ส่วนตัวบอร์ดของวงจร หรือ hardware โดยเราจะนำตัวบอร์ดนี้ไปบรรจุในกล่องและนำไปวางใกล้ๆกับ Solar cell เพื่อทำการวัดค่าต่างๆ โดยจะนำสายไฟปากคียบไปจับกับสายไฟ ของ Solar cell จากนั้นแรงดันจาก solar cell จะผ่าน voltage divider ลดแรงดันเพื่อเข้าไปใน Volt sensor module และค่าที่ออกจาก module ที่เป็น Analog จะไปเข้า mcp3208 ที่ต้องใช้ mcp3208 เพื่อเพิ่มจำนวนบิตเป็น 12 บิต และค่าที่ออกจาก mcp3208 เป็น digital ไปเข้าขาที่เป็น digital ของ esp8266 ส่วน current วัดได้จากแรงดันไฟที่ออกจาก solar cell ไปต่อกับ Load ที่เป็น R Shunt และเอา current sensor module มาวัดโดยต่ออนุกรมกับ load จากนั้นค่าที่ออกมาจาก Current sensor module เป็น Analog ไปเข้าขา Analog ของ esp8266 และค่าพารามิเตอร์ ความชื้นแสง, อุณหภูมิ, ความชื้น ที่ออกมาจาก sensor เข้าขา digital ของ esp8266 จากนั้นทำการเขียน โปรแกรมในส่วนของ software เพื่อนำ data แรงดันไฟ กระแสไฟ กำลังไฟฟ้า อุณหภูมิ ความชื้น ความเข้มแสง ไปเก็บใน excel และแสดงผลออกมาเป็นกราฟผ่านเว็บไซต์ Thingspeak

5.2 ข้อเสนอแนะ

1. เพื่อให้การวัดมีความแม่นยำที่มากขึ้นควรจะใช้ โมดูลวัดแรงดันที่เป็น PZEM-017 ซึ่งสามารถรับแรงดันได้ในช่วงที่มากขึ้นสามารถรองรับแรงดันจากโซลาร์เซลล์ได้ โดยไม่ต้องผ่าน การ divider ที่อาจเกิดค่าความคลาดเคลื่อนได้
2. ในส่วนของ Website ควรจะดู/เช็คข้อมูลเป็นช่วงๆ เพราะข้อมูลที่เข้ามาจะเข้ามาทุกๆ 10 นาที เพราะฉะนั้นการที่เข้ามาดูก็จะไม่ตอบสนองต่อจุดประสงค์ของการทดลอง
3. อาจจะต้องหาโหลดที่สามารถทนทานต่ออุณหภูมิและการใช้งานได้มากกว่านี้



เอกสารอ้างอิง

- [1] Artronshop. (2564). “ESP8266 ตอนที่ 1 รู้จักกับ ESP และรุ่นที่นิยมใช้งาน” สืบค้นเมื่อ 12 กันยายน 2566, จาก <https://www.artronshop.co.th/article/11/esp8266-ตอนที่-1-รู้จักกับ-esp-และรุ่นที่นิยมใช้งาน>
- [2] Artronshop. (2564). “เพิ่มช่องอนาล็อก 8 ช่อง 12 บิต ให้ ESP8266 ด้วย MCP3208” สืบค้นเมื่อ 14 กันยายน 2566, จาก <https://www.artronshop.co.th/article/77/เพิ่มช่องอนาล็อก-8-ช่อง-12-บิต-ให้-esp8266-ด้วย-mcp3208>
- [3] Cybertice. (2561). “โมดูลแปลงแรงดันลง XL7015 Step-Down” สืบค้นเมื่อ 14 กันยายน 2566, จาก <https://www.cybertice.com/product/1915/โมดูลแปลงแรงดันลง-xl7015-step-down-module-5v-80v-เป็น-5-20v-กระแส-0-8a>
- [4] Allnewstep. (2562). “สอนวิธีใช้ ACS712” สืบค้นเมื่อ 14 กันยายน 2566, จาก <https://www.allnewstep.com/article/350/107-arduino-สอนวิธีใช้งาน-ac-และ-dc-ac-712-arduino-current-sensor-เซนเซอร์วัดกระแสไฟฟ้า-ac-และ-dc-ac-712-arduino>
- [5] Price. (2564). “การต่อโวลต์+แอมป์มิเตอร์กับโซล่าเซลล์” สืบค้นเมื่อ 14 กันยายน 2566, จาก <https://www.price.in.th/การต่อโวลต์-แอมป์-มิเตอร์/>
- [6] Mrtoolshop. (2558). “Digital DC Voltmeter Ammeter” สืบค้นเมื่อ 14 กันยายน 2566, จาก http://www.mrtoolshop.com/index.php?route=news/article&news_id=34
- [7] Irradiance. (2562). “แผงโซล่าเซลล์มีกี่ชนิด ต่างกันอย่างไร” สืบค้นเมื่อ 3 ตุลาคม 2566, จาก <http://www.irradiance.co.th/th/newsdetail-18-แผงโซล่าเซลล์%20มีกี่ชนิด%20ต่างกันอย่างไร>
- [8] Arduino4. (2560). “Voltage Sensor Module” สืบค้นเมื่อ 5 ตุลาคม 2566, จาก <http://www.arduino4.com/product/327/voltage-sensor-module-โมดูลวัดแรงดันไฟฟ้า-โวลต์เซ็นเซอร์-dc-0-25v>
- [9] Padveeweb school. (2564). “สอน Wordpress สำหรับผู้เริ่มต้น” สืบค้นเมื่อ 5 ตุลาคม 2566, จาก <http://padveeweb school.com/learn-wordpress/>

- [10] Analogread. (2564). “เริ่มต้นใช้งาน Thingspeak” สืบค้นเมื่อ 16 ตุลาคม 2566, จาก <http://www.analogread.com/article/192/เริ่มต้นใช้งาน-thingspeak-เว็บไซต์-iot-วิธีสมัครและการประยุกต์ใช้งานด้วย-esp32>
- [11] Analogread. (2564). “สอนการใช้งานบอร์ด Arduino กับ BH1750FVI” สืบค้นเมื่อ 15 ธันวาคม 2566, จาก <http://www.analogread.com/article/82/สอนการใช้งานบอร์ด-arduino-กับ-bh1750fvi-วัดความเข้มแสง-พร้อมโค้ดตัวอย่าง>
- [12] IOT Engineering Education. (2565). “BH1750 Ambient Light Sensor Module” สืบค้นเมื่อ 27 กุมภาพันธ์ 2567, จาก <http://iot-kmutnb.github.io/blogs/about/>
- [13] Thaeasyelec. (2561). “การใช้งาน Character LCD Display กับ Arduino” สืบค้นเมื่อ 27 กุมภาพันธ์ 2567, จาก blog.thaeasyelec.com/how-to-use-character-lcd-display-arduino-ch2-i2c/
- [14] Rapidtables. (2564). “สูตรการคำนวณ Lux เป็นวัตต์ สืบค้นเมื่อ 28 กุมภาพันธ์ 2567, จาก <http://www.rapidtables.org/th/calc/light/lux-to-watt-calculator.html>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----Include the NodeMCU ESP8266 Library
#include <ESP8266WiFi.h>
#include <Wire.h> //BH1750
#include "BH1750FVI.h" //BH1750
#include <MCP3208.h>
#include <SPI.h>
#include "LiquidCrystal_I2C.h"
//-----
#include "DHT.h" //--> Include the DHT Sensor Library
#define DHTTYPE DHT22 //--> Defines the type of DHT sensor used (DHT11, DHT21, and DHT22),
in this project the sensor used is DHT11.

String apiKey = "U87UIGXD5I9TUGAJ"; //--> Enter your Write API key from ThingSpeak

//-----SSID and Password of your WiFi router
const char* ssid = "Parn";
const char* password = "ppakrinn";
//-----
//-----ThingSpeak Server
const char* server = "api.thingspeak.com";
//-----

const int DHTPin = 2; //--> The pin used for the DHT11 sensor is Pin D1=Pin 5
DHT dht(DHTPin, DHTTYPE); //--> Initialize DHT sensor, DHT dht(Pin_used,
Type_of_DHT_Sensor);
BH1750FVI LightMeter(BH1750FVI::k_DevModeContLowRes); //BH1750FVI
MCP3208 adc(D8);
LiquidCrystal_I2C lcd(0x27, 20, 4);
WiFiClient client;

// Sensitivity for ACS712
int Sens = 66; // Sensitivity in mV/A for the 5A
// int Sens = 100; // Sensitivity in mV/A for the 20A
const int analogIn = A0; // Analog input pin
int OffsetVoltage = 2700; // 0 Current offset Voltage in mV @ 0 amps
int RawValue = 0;
double Voltage = 0;
double Current = 0;
double Current2 = 0;
double Power;
void setup() {
  // put your setup code here, to run once:
  lcd.begin();
  Serial.begin(115200);
  dht.begin(); //--> Start reading DHT22 sensors
  delay(5000);

  WiFi.begin(ssid, password); //--> Connect to your WiFi router

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("");

// Initialize the I2C bus (BH1750 library doesn't do this automatically)
Wire.begin(); //BH1750
// On esp8266 you can select SCL and SDA pins using Wire.begin(D4, D3);
// For Wemos / Lolin D1 Mini Pro and the Ambient Light shield use Wire.begin(D2, D1);

LightMeter.begin(); //BH1750
adc.begin();

//-----Wait for connection
Serial.print("Connecting");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}

//-----If connection successful show IP address in serial monitor
Serial.println("");
Serial.print("Successfully connected to : ");
Serial.println(ssid);
}

void loop() {
  // put your main code here, to run repeatedly:
  float h = dht.readHumidity(); //DHT22 Humidity
  float t = dht.readTemperature(); //DHT22 Temperature
  int lux = LightMeter.GetLightIntensity(); //BH1750
  double val = mapf(adc.analogRead(0), 0, 4095, 0, 25);
  double volt = val*6; //ทำค่ากลับเป็น Input จากการ Divider ประมาณ 6.2 เท่า
  //-----find current-----
  RawValue = analogRead(analogIn); // Read voltage from ASC712
  Voltage = (RawValue / 1024.0) * 5000; // Convert to mV
  Current2 = -(Voltage - OffsetVoltage) / Sens; // Convert to amps
  Current = Current2/3;
  Power = Current * volt;
  delay(5000);
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
  }
  if (client.connect(server,80)) { //--> "184.106.153.149" or api.thingspeak.com
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(t);
    postStr += "&field2=";
    postStr += String(h);
    postStr += "&field3="; //BH1750
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
postStr += String(lux); //BH1750
postStr += "&field4="; //Voltage sensor
postStr += String(volt); //Voltage sensor
postStr += "&field5="; //Voltage sensor
postStr += String(Current); //Voltage sensor
postStr += "&field6="; //Voltage sensor
postStr += String(Power); //Voltage sensor
postStr += "\r\n\r\n";
```

```
client.print("POST /update HTTP/1.1\n");
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(postStr.length());
client.print("\n\n");
client.print(postStr);
//LCD
```

```
lcd.setCursor(1, 0);
lcd.print("Solar Informations");
lcd.setCursor(1, 1);
lcd.print("P=" + String(Power) + "W");
lcd.setCursor(10, 1);
lcd.print("V=" + String(volt) + "V");
lcd.setCursor(1, 2);
lcd.print("I=" + String(Current) + "A");
lcd.setCursor(10, 2);
lcd.print("L=" + String(lux) + "lx");
lcd.setCursor(1, 3);
lcd.print("T=" + String(t) + "C");
lcd.setCursor(10, 3);
lcd.print("H=" + String(h) + "%");
```

```
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" degrees Celcius, Humidity: ");
Serial.print(h);
Serial.println(" %");
Serial.print("Light: "); //BH1750
Serial.print(lux); //BH1750
Serial.println(" lx"); //BH1750
Serial.print("Voltage: ");
Serial.print(volt);
Serial.println(" V");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

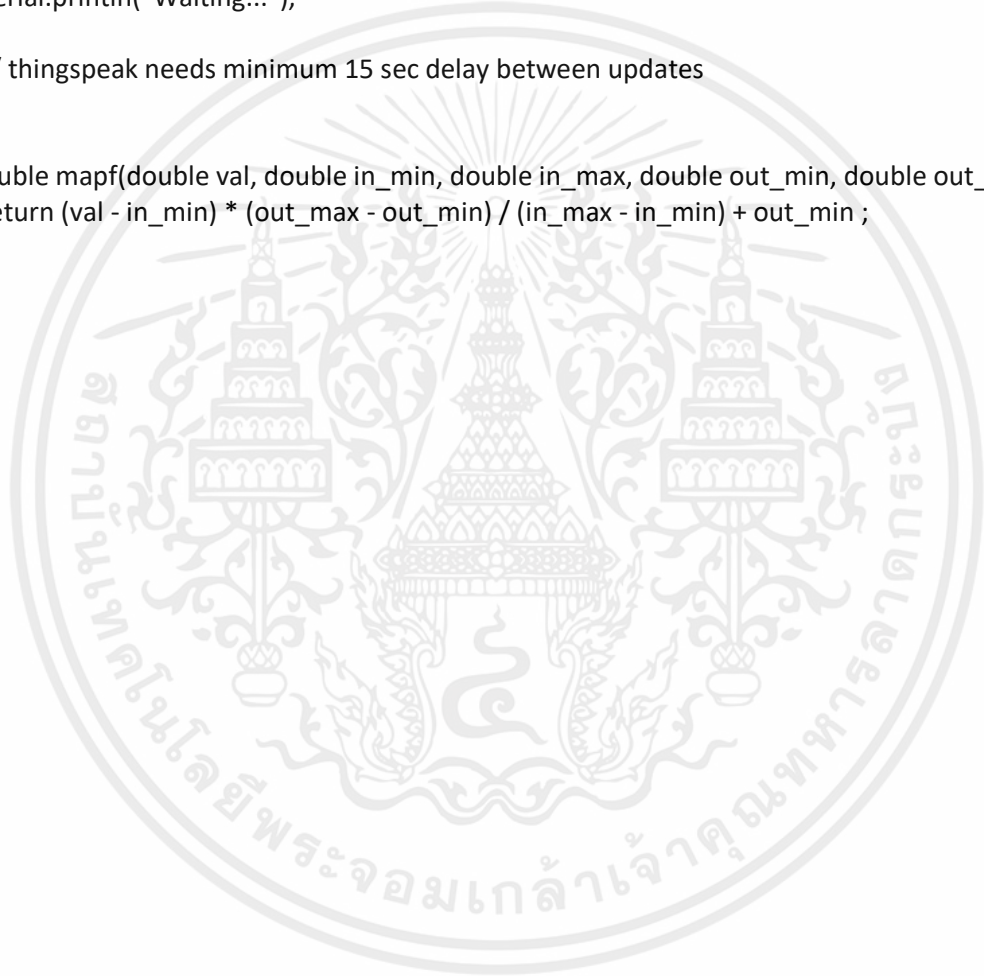
```

Serial.print("Current: ");
Serial.print(Current);
Serial.println(" A");
Serial.print("Power: ");
Serial.print(Power);
Serial.println(" W");
Serial.println("Send to Thingspeak.");
delay(1000); //BH1750
}
client.stop();
Serial.println("Waiting...");

// thingspeak needs minimum 15 sec delay between updates

}
double mapf(double val, double in_min, double in_max, double out_min, double out_max) {
return (val - in_min) * (out_max - out_min) / (in_max - in_min) + out_min ;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้