

เซนเซอร์ควบคุมผ่านอินเทอร์เน็ต
IoT SENSOR



โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาฟิสิกส์ประยุกต์
ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เซนเซอร์ควบคุมผ่านอินเทอร์เน็ต
IoT SENSOR



T149330



เลขหมู่.....
เลขทะเบียน.....149330
วัน เดือน ปี.....12 ก.พ. 2561

b. 12881843
l.....

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาฟิสิกส์ประยุกต์
ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IoT SENSOR



A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF SCIENCE (APPLIED PHYSICS)
DEPARTMENT OF PHYSICS, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ เซนเซอร์ควบคุมผ่านอินเทอร์เน็ต
IoT Sensor
ชื่อนักศึกษา นายวีรชัย อุดมสมุทร รหัสนักศึกษา 56051215
ปริญญา วิทยาศาสตร์บัณฑิต (ฟิสิกส์ประยุกต์)
ภาควิชา ฟิสิกส์
ปีการศึกษา 2559
อาจารย์ที่ปรึกษา รองศาสตราจารย์ วิชิต ศิริโชติ

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้
โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต(ฟิสิกส์ประยุกต์)
ประจำปีการศึกษา 2559

คณะกรรมการสอบ	ลายมือชื่อ
ดร.ภาณุพล โขลกกระโทก ประธานกรรมการ	
อ.ภูมินทร์ จินดาจิราวัฒน์ กรรมการ	
อ.ธรรมรัตน์ แต่งตั้ง กรรมการ	
รองศาสตราจารย์ วิชิต ศิริโชติ กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ	เซนเซอร์ควบคุมผ่านอินเทอร์เน็ต
ชื่อนักศึกษา	นายวีรชัย อุดมสมุทร รหัสนักศึกษา 56051215
ปริญญา	วิทยาศาสตร์บัณฑิต (ฟิสิกส์ประยุกต์)
ภาควิชา	ฟิสิกส์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2559
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ วิจิต ศิริโชติ

บทคัดย่อ

อุปกรณ์ IoT ที่ใช้วัดค่าอุณหภูมิและความชื้นควบคุมผ่านระบบอินเทอร์เน็ตได้รับการพัฒนาขึ้น อุปกรณ์ที่สร้างขึ้นประกอบไปด้วยบอร์ด NodeMCU เชื่อมต่อกับเซนเซอร์ DHT22 โดยบอร์ด NodeMCU จะเชื่อมต่อกับระบบอินเทอร์เน็ตโดยใช้โมดูล WIFI ในตัว เฟิร์มแวร์ของบอร์ด NodeMCU ที่ใช้ในการควบคุมเซนเซอร์และการเชื่อมต่ออินเทอร์เน็ตได้รับการพัฒนาด้วยภาษาซี เว็บไซต์ที่ใช้แสดงค่าอุณหภูมิและความชื้นได้รับการพัฒนาขึ้นโดยใช้ภาษา HTML ซึ่งค่าที่วัดได้จากเซนเซอร์จะถูกส่งไปที่ IoT Cloud Platform ทุกๆ 10 นาที และจะทำการบันทึกข้อมูลไว้ในฐานข้อมูล Could'sever โดยใช้โปรโตคอล MQTT เชื่อมต่อระหว่างเซนเซอร์และเว็บไซต์ ซึ่งเว็บไซต์สามารถแสดงค่าแบบเรียลไทม์และแสดงผลเป็นกราฟย้อนหลังได้ 5 ชั่วโมง จากผลการทดลองอ่านค่าอุณหภูมิและความชื้นจากเซนเซอร์ DHT22 เทียบกับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro พบว่าค่าความคลาดเคลื่อน 2%

คำสำคัญ : อุปกรณ์ IoT เซนเซอร์ DHT22 บอร์ด NodeMCU MQTT โปรโตคอล

Title	IoT Sensor
Students	Mr. Weerachai Udomsamuth Student ID 56051215
Degree	Bachelor of Science (Applied Physics)
Department	Physics
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic	2016
Advisor	Assoc.Prof. Wichit Sirichote

Abstract

The IoT device used for measuring temperature and humidity controlled over the internet has been developed. The device was built with the NodeMCU microcontroller board and the DHT22 sensor. The NodeMCU connects to the Internet using a built-in WIFI module. The NodeMCU's firmware that controls the sensor and internet connection has been developed with C language. The website used for displaying the temperature and humidity was written with HTML. The sensor readings are sent to the IoT Cloud Platform every 10 mins. The readings are saved in the Cloud's server database. The MQTT protocol is used to connect between sensor and website. The website is capable of displaying real-time readings and the 5 hours recorded data in graphical format. The sensor was tested with the BT-1 Pro humidity/temperature meter. The readings of DHT22 sensor were 2% higher than the BT-1 Pro.

Keywords : IoT device, DHT22 Sensor, NodeMCU board, MQTT protocol

กิตติกรรมประกาศ

โครงการพิเศษเล่มนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์จาก รองศาสตราจารย์ วิชิต ศิริโชติ (อาจารย์ที่ปรึกษา) ที่ได้ให้ความรู้ความเข้าใจ คำแนะนำ และ คำปรึกษาในการแก้ปัญหาต่างๆ จึงทำให้โครงการพิเศษเล่มนี้มีความสมบูรณ์และก่อให้เกิดประโยชน์ ต่อผู้ที่จะได้ศึกษาต่อไป

ขอขอบคุณผู้ช่วยศาสตราจารย์ ดร. ปานวิทย์ ชูระนุติ คณะเทคโนโลยีสารสนเทศ ที่ได้ให้ คำปรึกษาด้านการทำงานของ NodeMCU และโปรแกรม Arduino IDE

ขอขอบคุณคณะอาจารย์คณะเทคโนโลยีสารสนเทศ และวิทยาการที่ได้จัดโครงการอบรมการ พัฒนาความรู้ทักษะด้าน Internet of Things ซึ่งเป็นกิจกรรมดีๆ ที่ได้มาให้ความรู้และพัฒนา ศักยภาพของผู้จัดทำให้มีความเข้าใจเกี่ยวกับ Internet of Things มากยิ่งขึ้น

ขอขอบคุณ ดร.ภาณุพล โขลกกระโทก อ.ภูมินทร์ จินดาจิธาวัฒน์ และอาจารย์ธรรมรัตน์ แต่งตั้ง ที่ได้เสียสละเวลาอันมีค่าและให้เกียรติอย่างมากในการเป็นคณะกรรมการสอบรวมไปถึงให้ คำแนะนำในการทำรูปเล่มรายงานของวิจัยนี้

ขอขอบคุณผู้ที่เขียนบทความเชิงวิชาการทุกท่านที่ผู้จัดทำได้นำมาอ้างอิงข้อมูลประกอบการ ทำวิจัยเล่มนี้

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา ที่คอยให้กำลังใจและสนับสนุนในการศึกษาและ การทำงานตลอดมา รวมถึงรุ่นพี่และเพื่อนภาควิชาฟิสิกส์ประยุกต์ และภาควิชาวิทยาการ คอมพิวเตอร์ที่ช่วยเหลือเอื้อเฟื้อ แบ่งปันข้อมูลที่เป็นประโยชน์ต่อโครงการพิเศษเล่มนี้ หากมีข้อตก บกพร้อมนี้ได้กล่าวถึงผู้มีพระคุณท่านใด ผู้จัดทำต้องขอภัยมา ณ ที่นี้ด้วย

วีรชัย อุดมสมุทร

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของงานวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 ความหมายของ Internet of Things	3
2.2 แนวคิดของ Internet of Things	5
2.2.1 A wireless sensor network (WSN)	6
2.2.2 Access Technology	6
2.2.3 Gateway Sensor Nodes	7
2.2.4 การแบ่งกลุ่ม Internet of Things	8
2.2.5 IPv6	8
2.2.6 โพรโตคอล (Message Queue Telemetry Transport, MQTT)	9
2.2.7 IoT Reference Model	9
2.3 เซนเซอร์วัดอุณหภูมิและความชื้น DHT22	10
2.4 NodeMCU Board	12
2.5 HTML และ Dreamweaver CS6	13
2.6 งานวิจัยที่เกี่ยวข้อง	13
บทที่ 3 วิธีการดำเนินงานวิจัย	15
3.1 ขั้นตอนการออกแบบวงจร	15
3.1.1 วัสดุอุปกรณ์ที่ใช้ในการทดลอง	16
3.2 วิธีการทดลอง	19
3.3 การประยุกต์ใช้โปรแกรม	29
3.3.1 ขั้นตอนการสร้างเว็บไซต์	29
3.2.2 ขั้นตอนการอัปโหลด Web Server	30
3.4 การทดสอบการตอบสนองระหว่าง NodeMCU กับเว็บไซต์ และการบันทึกข้อมูล	32

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการวิจัยและการอภิปรายผล	35
4.1 คุณสมบัติของเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro กับเซนเซอร์วัดอุณหภูมิ และความชื้น DHT22	35
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	40
เอกสารอ้างอิง	41
ภาคผนวก	42
ภาคผนวก ก	43
ภาคผนวก ข	59



สารบัญตาราง

ตารางที่	หน้า
1.1 ขั้นตอนการดำเนินงาน	2
2.1 เทคโนโลยีสำหรับการเชื่อมต่อสำหรับ Access Technology	7
2.2 คุณสมบัติของ Sensor วัดอุณหภูมิและความชื้น DHT22	12
3.1 การควบคุมการแสดงผลของจอ LCD (I2C)	18
4.1 การเปรียบเทียบค่าของอุณหภูมิระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro	35
4.2 การเปรียบเทียบค่าของความชื้นระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro	36



สารบัญรูป

รูปที่	หน้า
2.1 Internet of Things	3
2.2 ตู้เย็นอัจฉริยะ	4
2.3 เครื่องซักผ้าอัจฉริยะ	4
2.4 นาฬิกาอัจฉริยะ	4
2.5 ผลสำรวจการประยุกต์ใช้งาน Internet of Things ที่ถูกพูดถึงมากที่สุด	5
2.6 Wireless Sensor Network	6
2.7 WSN Nodes	7
2.8 ไดอะแกรมอธิบายการเชื่อมต่อ Gateway หลายตัวเข้ากับ local network	7
2.9 Network Layers Internet of Things ของ IBM	8
2.10 IoT Reference Model	9
2.11 เซนเซอร์วัดอุณหภูมิและความชื้น DHT22	11
2.12 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีทั้งหมด 5 ไบต์ (40 บิต) สองไบต์แรกสำหรับความชื้น สองไบต์ต่อมาสำหรับอุณหภูมิและไบต์สุดท้ายเป็น checksum หรือ parity Bits	11
2.13 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH	11
2.14 NodeMCU Board	12
3.1 แผนภาพไดอะแกรมของการออกแบบวงจร	15
3.2 NodeMCU	16
3.3 ขบวนการทำงานของ NodeMCU V3	16
3.4 DHT22 Temperature & Relative Humidity Sensor	17
3.5 จอแสดงผล LCD	17
3.6 การเชื่อมต่อ I ² C	18
3.7 การต่อวงจรกับอุปกรณ์	19
3.8 หน้าต่างดาวน์โหลด Arduino IDE	19
3.9 การเพิ่ม Board ของ ESP8266	20
3.10 หน้าต่างฟังก์ชันเครื่องมือ	20
3.11 หน้าต่าง Boards Manager	21
3.12 การเลือกบอร์ด NodeMCU	21
3.13 การทดสอบการเชื่อมต่อของ NodeMCU กับ WIFI	22
3.14 หน้าต่างของโค้ดตัวอย่างในการทดสอบการเชื่อมต่อของ NodeMCU กับ WIFI	22
3.15 หน้าต่าง Serial Monitor	23
3.16 หน้าต่างเว็บไซต์ NETPIE	23
3.17 การ create application	24
3.18 การเลือกรูปแบบของการ create application	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.19 หน้าต่างของรหัส Key และ Secret	25
3.20 หน้าต่างของเว็บไซต์ ThingSpeak	25
3.21 การสร้าง Channel	26
3.22 หน้าต่างของรหัส API Key	26
3.23 หน้าต่างการ Upload	27
3.24 หน้าต่างการทำงานของโค้ด Arduino IDE	27
3.25 หน้าต่างแสดงการเชื่อมต่อระหว่าง NodeMCU กับ NETPIE	28
3.26 หน้าต่างการบันทึกอุณหภูมิและความชื้น	28
3.27 แผนภาพไดอะแกรมแสดงการเชื่อมต่อของอุปกรณ์ในระบบ IoT Sensor	29
3.28 หน้าต่างการทำงานโปรแกรม Dreamweaver	29
3.29 หน้าต่างการแสดงผลของโปรแกรม Dreamweaver	30
3.30 หน้าต่างของเว็บไซต์ hostinger	30
3.31 หน้าต่างของรหัส IP ADDRESS และ PASSWORD	31
3.32 หน้าต่างของโปรแกรม FileZilla Client	31
3.33 วงจรการทำงาน	32
3.34 อุปกรณ์ IoT Sensor	32
3.35 เว็บไซต์ที่สร้างสำเร็จแล้ว	33
3.36 เปรียบเทียบค่าอุณหภูมิและความชื้นจากจอ LCD กับบนเว็บไซต์	33
3.37 เมื่อสั่งการให้สวิตช์เปิด (On) จะทำให้ไฟ LED ติด	33
3.38 เมื่อสั่งการให้สวิตช์ปิด (Off) จะทำให้ไฟ LED ดับ	33
3.39 กราฟความสัมพันธ์ระหว่าง Temperature (°C) กับ time	34
3.40 กราฟความสัมพันธ์ระหว่าง Humidity (%) กับ time	34
4.1 กราฟเปรียบเทียบค่าของอุณหภูมิระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro	35
4.2 กราฟเปรียบเทียบค่าของความชื้นระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro	36
4.3 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่สภาพแวดล้อมทั่วไป	37
4.4 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่อุณหภูมิห้อง	37
4.5 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่ห้องน้ำ	38
4.6 ค่าของอุณหภูมิและความชื้นเมื่อวัดตู้เย็น	38
4.7 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่ตู้ดูดความชื้น	39
4.8 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่กลางแจ้ง	39

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของงานวิจัย

ในปัจจุบัน Internet of Things หรือ IoT เป็นเทคโนโลยีที่ได้เข้ามามีบทบาทสำคัญต่อการดำรงชีวิตของมนุษย์อย่างมาก ทุกวันนี้ความต้องการของมนุษย์คือความสะดวกสบาย รวดเร็ว และปลอดภัยในการใช้ชีวิตประจำวัน มนุษย์จึงได้คิดค้นและสร้างสรรค์อุปกรณ์หรือเทคโนโลยีที่สามารถเชื่อมต่อกับอินเทอร์เน็ตได้โดยการเชื่อมต่ออุปกรณ์ต่างๆ เป็นเครือข่ายและเชื่อมโยงอุปกรณ์เข้าไว้ด้วยกันผ่านระบบเครือข่ายอินเทอร์เน็ต

อุปกรณ์ที่มีวงจรมicrocontroller หรือเซนเซอร์ จะสามารถสื่อสารและเชื่อมต่อกับเครือข่าย IoT ได้ โดยจะเชื่อมต่อสื่อสารหรือแลกเปลี่ยนข้อมูลกันได้โดยผ่านระบบเครือข่ายอินเทอร์เน็ต ทำให้อุปกรณ์นั้นสามารถรับรู้สภาพแวดล้อมและถูกควบคุมได้จากระยะไกล และมีปฏิสัมพันธ์ได้ต่อกันจึงจะสามารถทำงานร่วมกันได้ ยกตัวอย่างเช่น เซนเซอร์ภายในบ้าน ซึ่งใช้ตรวจจับการเคลื่อนไหวของผู้อยู่อาศัย และส่งสัญญาณเพื่อสั่งเปิดหรือปิดสวิตช์ไฟตามห้องต่างๆ อุปกรณ์วัดสัญญาณชีพของผู้ป่วย ซึ่งส่งข้อมูลไปยังบุคลากรทางการแพทย์ หรือส่งข้อความเรียกหน่วยกู้ชีพหรือรถฉุกเฉิน เป็นต้น

ดังนั้นผู้วิจัยจึงเล็งเห็นประโยชน์ของเครือข่าย IoT ในการนำมาประยุกต์ใช้กับเซนเซอร์ให้สามารถรับและส่งข้อมูลผ่านอินเทอร์เน็ตได้ เพื่อความสะดวกสบาย ปลอดภัย และมีประสิทธิภาพในการใช้งานสูงสุด โดยเซนเซอร์ที่ใช้จะเป็นเซนเซอร์วัดอุณหภูมิและความชื้น และใช้ NodeMCU ในการควบคุมเซนเซอร์และเชื่อมต่อกับอินเทอร์เน็ต เมื่อมีการเชื่อมต่อกับอินเทอร์เน็ตแล้วข้อมูลจะถูกส่งไปยังเว็บไซต์ เพื่อแสดงข้อมูลที่รับมา ซึ่งจะสามารถควบคุมอุปกรณ์ได้จากระยะไกลผ่านเว็บไซต์ เพื่อให้เกิดประโยชน์สูงสุดต่อไป

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อศึกษาหลักการทำงานของเครือข่าย Internet of Things (IoT)
- 2) เพื่อศึกษาหลักการทำงานของ NodeMCU และการเขียนโปรแกรม Arduino IDE
- 3) เพื่อศึกษาหลักการทำงานของเซนเซอร์
- 4) เพื่อศึกษาการเขียน HTML ด้วยภาษา C โดยใช้โปรแกรม Dreamweaver CS6

1.3 ขอบเขตของงานวิจัย

- 1) สามารถแสดงค่าอุณหภูมิและความชื้นผ่านจอ LCD ได้
- 2) สามารถเชื่อมต่ออินเทอร์เน็ตได้
- 3) สามารถส่งค่าอุณหภูมิและความชื้นไปยังเว็บไซต์ได้
- 4) สามารถบันทึกค่าอุณหภูมิและความชื้นบนเว็บไซต์ได้
- 5) สามารถควบคุมอุปกรณ์ผ่านเว็บไซต์ได้

1.4 ขั้นตอนการดำเนินงาน.

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ระยะเวลา						
	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
ศึกษาค้นคว้าเรื่องเซนเซอร์ชนิดต่างๆ และการทำงานของ NodeMCU	↔						
ศึกษาวิธีการเขียนโปรแกรม Arduino IDE และ Dreamweaver		↔					
ทดสอบการทำงานของโปรแกรม Arduino IDE และ Dreamweaver			↔				
แก้ไขปรับปรุงข้อผิดพลาดของโปรแกรม Arduino IDE และ Dreamweaver				↔			
ทดสอบการทำงานของเว็บไซต์					↔		
สรุปผลและวิจารณ์ผลการทดลอง						↔	↔

1.5 ประโยชน์ที่คาดว่าจะได้รับ

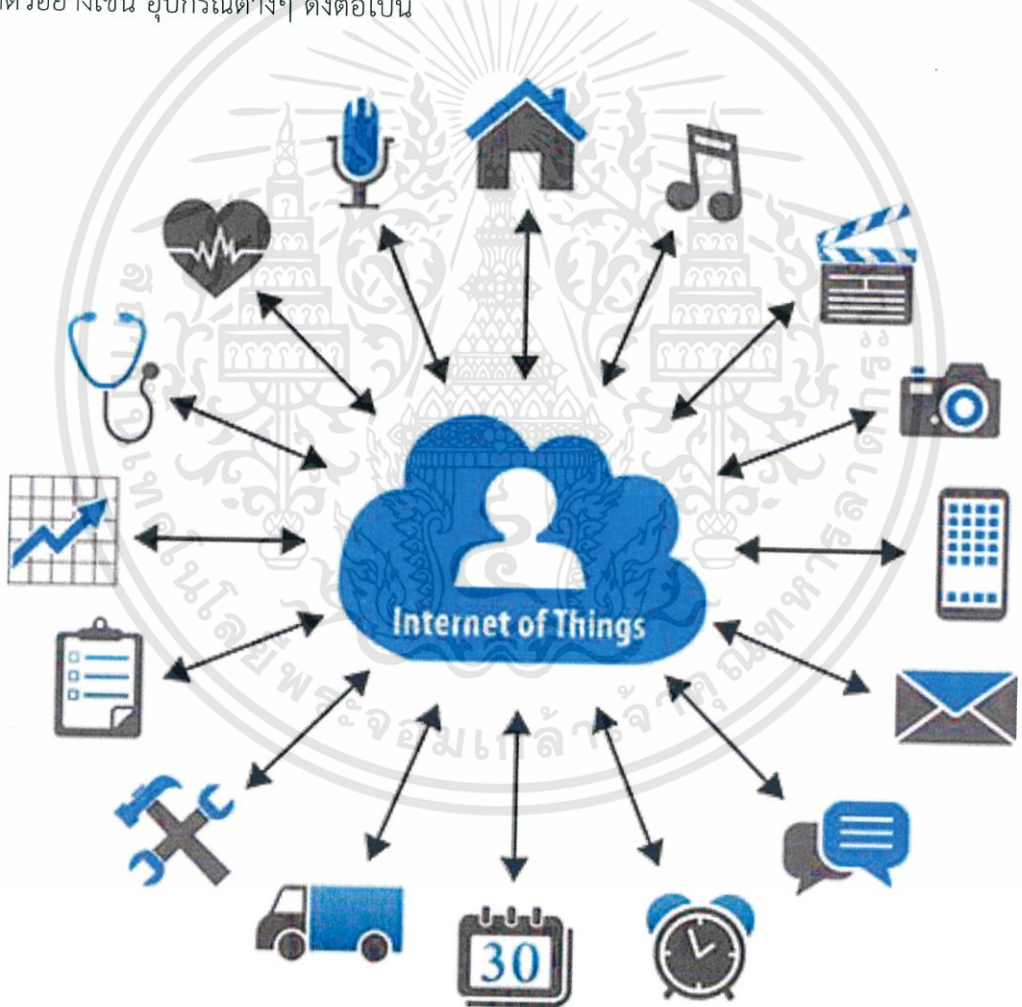
- 1) สามารถนำความรู้จากเรื่อง Internet of Things ไปประยุกต์ใช้ได้ในอนาคต
- 2) สามารถเขียนโปรแกรม Arduino IDE ได้
- 3) สามารถเขียนเว็บไซต์เบื้องต้นด้วยภาษา C ได้
- 4) เข้าใจหลักการทำงานของ NodeMCU และสามารถต่อวงจรกับเซนเซอร์ได้
- 5) อำนวยความสะดวกให้แก่ผู้ใช้งานเว็บไซต์นี้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ความหมายของ Internet of Things

เทคโนโลยี Internet of Things (IoT) บางแห่งเรียก M2M ย่อมาจาก Machine to Machine คือ เทคโนโลยีอินเทอร์เน็ตที่เชื่อมต่ออุปกรณ์กับเครื่องมือต่างๆ เช่น โทรศัพท์มือถือ วิทยุ ทรานซ์มิเตอร์ ตู้เย็น โทรทัศน์ และอื่นๆ เข้าไว้ด้วยกัน โดยการเชื่อมโยงจะช่วยให้สื่อสารกันได้ผ่านระบบอินเทอร์เน็ตและในอนาคตสิ่งต่างๆ จะสามารถเชื่อมต่อกันได้ด้วยระบบเครือข่าย IoT ซึ่งส่งผลให้ผู้บริโภคทั่วไปเริ่มจะคุ้นเคยกับเทคโนโลยีที่ทำให้พวกเขาสามารถควบคุมสิ่งของต่างๆ ทั้งจากในบ้านและสำนักงาน หรือจากที่ไหนก็ได้ทั้งนั้น เช่น การควบคุมอุณหภูมิภายในบ้าน ไปจนถึงการสั่งการเปิดปิดไฟ เป็นต้น ยกตัวอย่างเช่น อุปกรณ์ต่างๆ ดังต่อไปนี้



รูปที่ 2.1 Internet of Things

ตู้เย็นอัจฉริยะ มีตัวตรวจจับจำนวนสิ่งของต่างๆ ในตู้ เมื่ออาหารในตู้เย็นใกล้จะหมดอายุหรือสิ่งใดหมดก็จะมีข้อความแจ้งเตือนไปยังสมาร์ทโฟนให้เราสามารถสั่งซื้อของได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ตู้เย็นอัจฉริยะ

เครื่องซักผ้าอัจฉริยะ เครื่องซักผ้าสามารถรายงานสถานการณ์การทำงานของเครื่องไปยังสมาร์ตโฟนได้ กรณีที่เราใช้เครื่องซักผ้าในขณะที่เราไม่อยู่บ้าน สามารถตั้งค่าเครื่องซักผ้าในโหมดพิเศษได้ โดยเมื่อเครื่องซักผ้าทำงานเสร็จแล้ว เครื่องจะปั่นผ้าเบาๆ ทุกสองนาทีก่อน เพื่อให้มีอากาศไหลผ่าน ช่วยให้ผ้าไม่อับชื้น และเมื่อเรากลับถึงบ้าน เครื่องซักผ้าจะตรวจจับได้ว่าเรากลับบ้านแล้ว เครื่องซักผ้าจะจบการทำงานและมีการแจ้งเตือน



รูปที่ 2.3 เครื่องซักผ้าอัจฉริยะ

นาฬิกาอัจฉริยะ มีความสามารถมากกว่าใช้ดูเวลาเท่านั้น เช่น ถ่ายรูป บันทึกวิดีโอ รับส่งอีเมล จับเวลา นับก้าวเดิน คำนวณระยะทางและพลังงานที่ร่างกายใช้ นอกจากนี้ยังใช้เป็นรีโมตคอนโทรลของโทรทัศน์ได้อีกด้วย

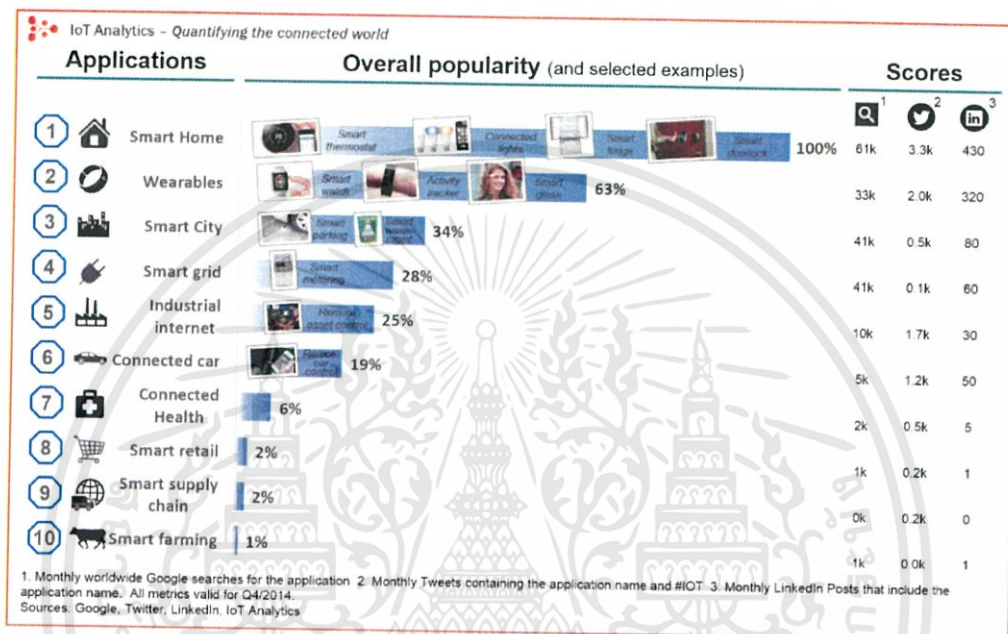


รูปที่ 2.4 นาฬิกาอัจฉริยะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากอุปกรณ์เหล่านี้แล้วก็ยังมีการใช้งานในรูปแบบอื่นๆ อีก เช่น การจอดรถในห้างสรรพสินค้าที่มีเซนเซอร์ตรวจจับที่ว่างแล้วแสดงผลให้ลูกค้าทราบ ในทางการแพทย์ใช้ในการวิเคราะห์ ตรวจสอบอาการของคนไข้ผ่านอุปกรณ์ที่ติดตั้งไว้ที่บ้านของคนไข้ จากผลสำรวจแนวทางการนำ Internet of Things ไปใช้โดยสำรวจจากการค้นหาใน Google การแชร์ผ่าน Twitter และ LinkedIn ดังรูปที่ 2.5 หัวข้อที่พบมากที่สุดคือ Smart Home หรือ ระบบบ้านอัจฉริยะ

10 อันดับการประยุกต์ใช้ Internet of Things ยอดนิยมในปัจจุบันนี้



รูปที่ 2.5 ผลสำรวจการประยุกต์ใช้งาน Internet of Things ที่ถูกพูดถึงมากที่สุด

จะเห็นได้ว่าปัจจุบันนี้ Internet of Things เป็นสิ่งที่ผู้คนให้ความสนใจและเริ่มคุ้นเคยกับเทคโนโลยีและอินเทอร์เน็ตมากขึ้น ทำให้เกิดความสะดวกสบาย อีกทั้งยังมีการพัฒนาด้านของเทคโนโลยีทำให้สามารถใช้และควบคุมสิ่งของต่างๆ ได้ง่ายขึ้น นอกจากนี้ยังมีการรับส่งข่าวสารบนอินเทอร์เน็ตที่มากมาย และเส้นทางของข้อมูลมีการเปลี่ยนแปลงเนื่องจากข้อมูลได้กลายเป็นส่วนหนึ่งของระบบข้อมูลโดยใช้ตัวเซนเซอร์หรือตัวเก็บข้อมูลฝังอยู่ในวัตถุต่างๆ ทำให้เกิดความได้เปรียบและเกิดความแม่นยำ ซึ่งส่งผลให้ในอนาคตสามารถควบคุมสิ่งของทั้งจากในบ้านและนอกบ้านได้ เช่น การเปิดปิดเครื่องทำอากาศภายในบ้าน รวมทั้งเทคโนโลยีอีกมากมายที่ช่วยให้เกิดความสะดวกสบาย

2.2 แนวคิดของ Internet of Things

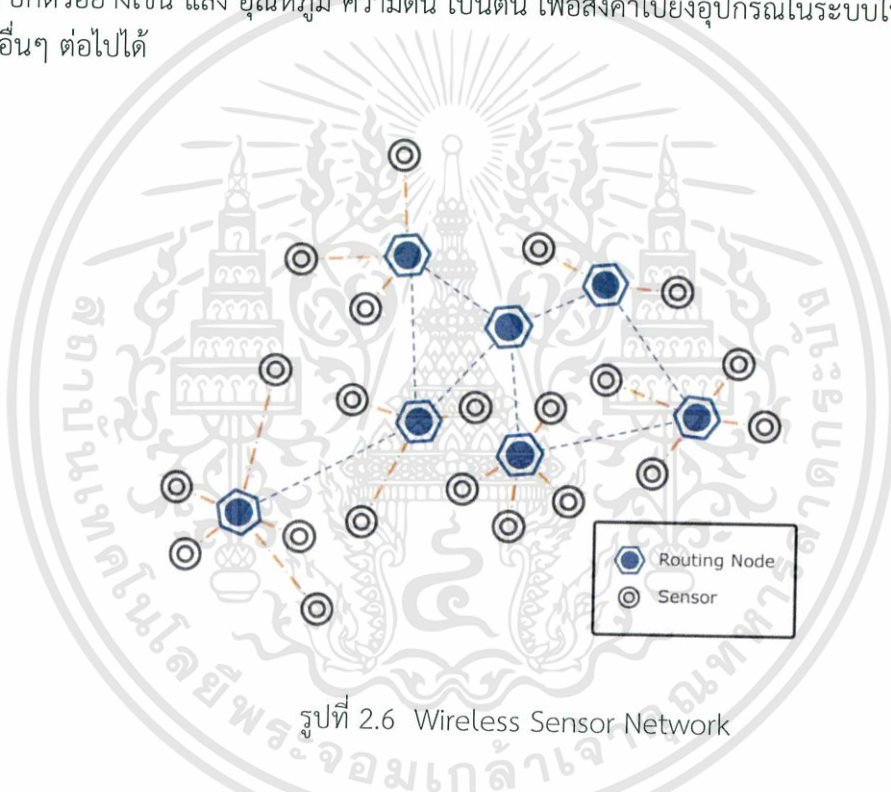
แนวคิด Internet of Things นั้นถูกคิดขึ้นโดย Kevin Ashton ในปี ค.ศ. 1999 ซึ่งเขาเริ่มต้นโครงการ Auto-ID Center ที่มหาวิทยาลัย Massachusetts Institute of Technology หรือ MIT จากเทคโนโลยี RFID ที่จะทำให้เป็นมาตรฐานระดับโลกสำหรับ RFID Sensors ต่างๆ ที่จะเชื่อมต่อกันได้ ต่อมาในยุคหลังปี ค.ศ. 2000 โลกมีอุปกรณ์อิเล็กทรอนิกส์ออกมาเป็นจำนวนมากและมีการใช้คำว่า Smart ซึ่งในที่นี้คือ smart device, smart grid, smart home, smart network และ smart intelligent transportation ต่างๆ เหล่านี้ล้วนมีโครงสร้างพื้นฐานที่สามารถเชื่อมต่อกับโลกอินเทอร์เน็ตได้ ซึ่งการเชื่อมต่อเหล่านี้เองกลายมาเป็นแนวคิดที่ว่าอุปกรณ์เหล่านี้ก็ย่อมสามารถ

ไม่ว่าการณ์ใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อสารกันได้ด้วยเช่นกัน โดยอาศัยตัว Sensor ในการสื่อสารถึงกัน นั่นแปลว่านอกจาก Smart devices ต่างๆ จะเชื่อมต่ออินเทอร์เน็ตได้แล้ว มันยังสามารถเชื่อมต่อไปยังอุปกรณ์ตัวอื่นได้ด้วย โดย Kevin นิยามมันไว้ตอนนั้นว่าเป็น “internet-like” หรือพูดง่ายๆ ก็คือ อุปกรณ์อิเล็กทรอนิกส์สามารถสื่อสารพูดคุยกันเองได้ ซึ่งศัพท์คำว่า “Things” ก็แทนอุปกรณ์อิเล็กทรอนิกส์ที่กล่าวมาก่อนหน้านี้นั่นเอง

2.2.1 A wireless sensor network (WSN)

เป็นตัวแปรสำคัญสำหรับ Internet of Things ที่ใช้ในการสื่อสาร นั้นไม่เพียงแต่ Internet network เท่านั้น แต่ยังมีตัวแปรอื่นเข้ามาเกี่ยวข้องอีก ซึ่งก็คือ Sensor node ต่างๆ จำนวนมากที่ทำให้เกิด wireless sensor network (WSN) ให้กับอุปกรณ์ต่างๆ นั้นสามารถเชื่อมต่อเข้ามาได้ ซึ่ง WSNs นี้เองสามารถตรวจจับปรากฏการณ์ต่างๆ (physical phenomena) ในเครือข่ายได้ด้วย ยกตัวอย่างเช่น แสง อุณหภูมิ ความดัน เป็นต้น เพื่อส่งค่าไปยังอุปกรณ์ในระบบให้ทำงานหรือสั่งงานอื่นๆ ต่อไปได้



รูปที่ 2.6 Wireless Sensor Network

2.2.2 Access Technology

การพัฒนา Internet of Things นั้น นอกจากจะพัฒนาเทคโนโลยีในฝั่ง Hardware ได้แก่ processors, radios และ sensors ซึ่งจะถูกรวมเข้าด้วยกันเรียกว่า a single chip หรือ system on a chip (SoC) แล้ว มันยังพัฒนา WSN ไปพร้อมๆ กันด้วย และเมื่อพูดถึงการเชื่อมต่อในปัจจุบันได้มีการพัฒนาเทคโนโลยีสำหรับการเชื่อมต่อสำหรับ Access Technology ทั้งหมด 3 เรื่อง ได้แก่

- 1) Bluetooth 4.0
- 2) IEEE 802.15.4e (มาตรฐานการทำงานของเครือข่ายไร้สาย)
- 3) WLAN IEEE 802.11™ (Wi-Fi)

โดยในแต่ละ Access Technology นั้นมีการส่งข้อมูลที่แตกต่างกันตามตารางที่ 2.1

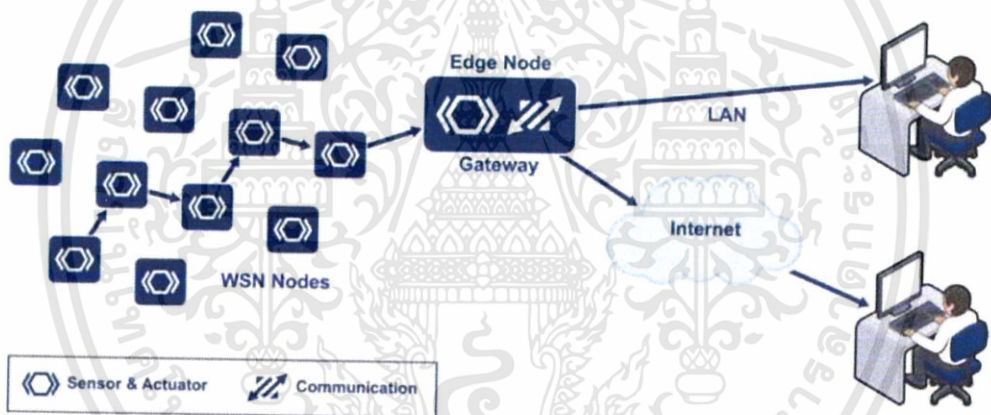
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 เทคโนโลยีสำหรับการเชื่อมต่อสำหรับ Access Technology

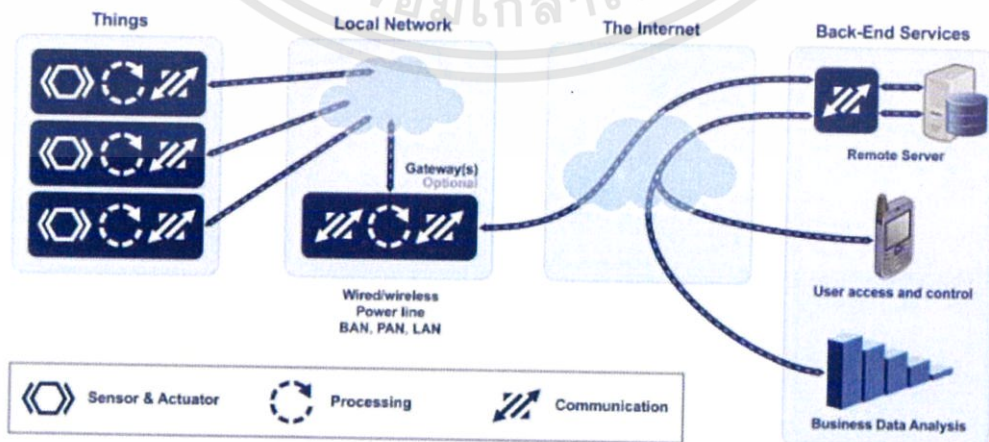
	IEEE 802.15.4e	Bluetooth	WLAN IEEE 802.11
Frequency	868/915 MHz 2.4 GHz	2.4 GHz	2.4, 5.8 GHz
Data rate	250 Kbps	723 Kbps	11 – 105 Mbps
Power	Very low	Low	High

2.2.3 Gateway Sensor Nodes

เมื่อมีโครงข่าย Sensor nodes แล้ว ก็จำเป็นจะต้องมี Gateway Sensor Nodes เพื่อที่จะเชื่อมต่อไปยังโลกอินเทอร์เน็ตด้วย โดยตัว Gateway นี้จะทำหน้าที่เชื่อมต่อไปยังเครือข่าย Internet โดยให้อุปกรณ์ทั้งหมดในโครงข่าย Sensor nodes ส่งข้อมูลเข้าสู่อินเทอร์เน็ตได้นั่นเอง และ Gateway ก็จะมีอยู่ภายใต้ Local network ซึ่งจะมีการกำหนดกันต่อไปว่า Gateway ภายใต้ Local network ที่ว่านั้นจะให้เชื่อมต่อไปยัง Internet ได้ด้วยหรือไม่ ถ้าเชื่อมต่อไปยัง Internet ไม่ได้ อุปกรณ์ที่เชื่อมต่อเข้ามาใน Gateway ก็อาจจะสื่อสารกันได้เฉพาะภายใน Local network เท่านั้น



รูปที่ 2.7 WSN Nodes



รูปที่ 2.8 ไดอะแกรมอธิบายการเชื่อมต่อ Gateway หลายตัวเข้ากับ local network

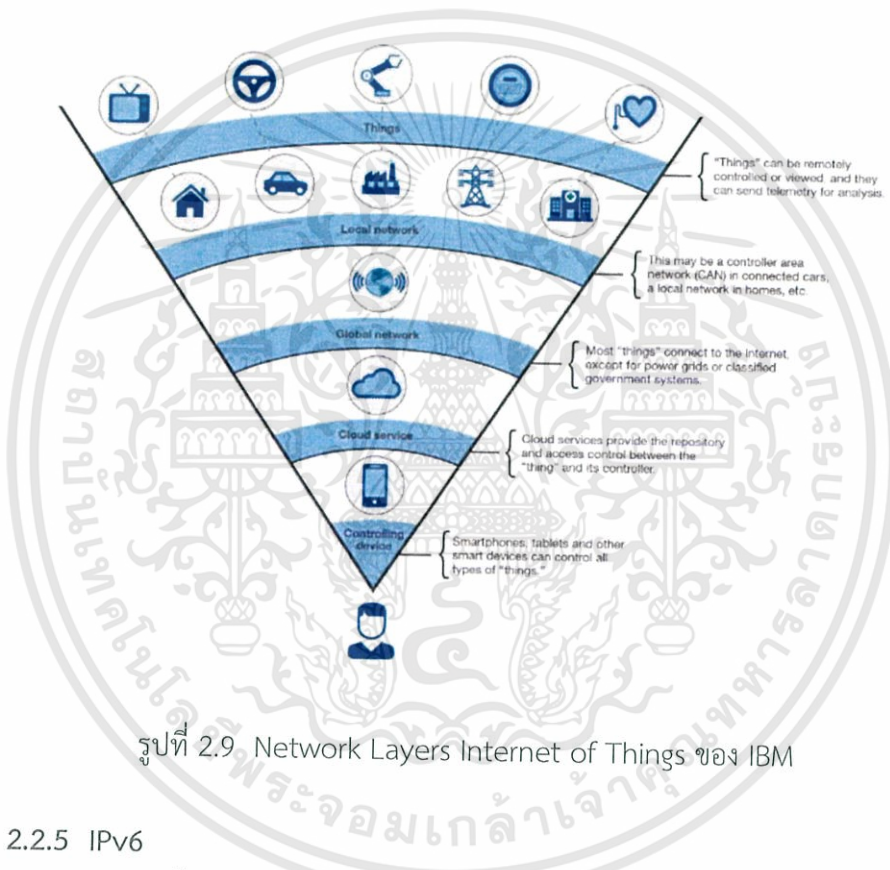
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การแบ่งกลุ่ม Internet of Things

ปัจจุบันมีการแบ่งกลุ่ม Internet of Things ออกตามตลาดการใช้งานเป็น 2 กลุ่ม ได้แก่

1) Industrial IoT คือ แบ่งจาก local network ที่มีหลายเทคโนโลยีที่แตกต่างกันในโครงข่าย Sensor nodes โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะเชื่อมต่อแบบ IP network เพื่อเข้าสู่อินเทอร์เน็ต

2) Commercial IoT คือ แบ่งจาก local communication ที่เป็น Bluetooth หรือ Ethernet (wired or wireless) โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะสื่อสารภายในกลุ่ม Sensor nodes เดียวกันเท่านั้นหรือเป็นแบบ local devices เพียงอย่างเดียว อาจไม่ได้เชื่อมสู่อินเทอร์เน็ต



รูปที่ 2.9 Network Layers Internet of Things ของ IBM

2.2.5 IPv6

IPv6 หรือ IP Address Version 6 คือ ส่วนสำคัญของ Internet of Things โดยตัวอุปกรณ์ IoT devices ต่างๆ นั้นจะต้องมีหมายเลขระบุเพื่อใช้ในการสื่อสารเปรียบเสมือนที่อยู่บ้านของเรานั่นเอง และการที่จะทำให้อุปกรณ์เหล่านั้นที่มีอยู่เป็นจำนวนมาก (รวมถึงอนาคตที่จะผลิตกันออกมา) สามารถใช้ในการสื่อสารได้จึงจำเป็นต้องใช้ IPv6 มากำกับเพื่อให้ได้หมายเลขที่ไม่ซ้ำกันและต้องใช้ได้ทั้ง

- IoT network ที่เป็น LAN, PAN และ BAN: Body Area Network หรือการสื่อสารของตัว Sensor กับร่างกายมนุษย์
- Internet network (protocols) ที่เป็น IP, UDP, TCP, SSL, HTTP, HTTPS และอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 โพรโทคอล (Message Queue Telemetry Transport, MQTT)

MQTT เป็นโพรโทคอลหรือมาตรฐานการสื่อสารข้อมูลระหว่างเครื่องจักรกับเครื่องจักร (machine-to-machine) ที่ส่งผ่านข้อมูลกันบน Internet of Things ซึ่งมีวิธีการพื้นฐานเหมือนกับมาตรฐานการส่งข้อความสั้น (Message Queue) ปกติ แต่ได้รับการพัฒนาให้รองรับงานด้าน Internet of Things มากยิ่งขึ้น อาทิ ขณะมีการรอรับและอ่านค่าแล้วยังสามารถสั่งงานอุปกรณ์ที่ใช้ MQTT ได้ด้วย จุดเด่นที่แตกต่างจากโพรโทคอลอื่นๆ คือ โพรโทคอลนี้จะต้องมีอุปกรณ์ที่ทำหน้าที่ฝั่งแม่ข่าย เรียกว่า MQTT Broker เพื่อทำหน้าที่รับส่งข้อมูลไปยังอุปกรณ์อื่นที่ใช้ MQTT เหมือนกัน

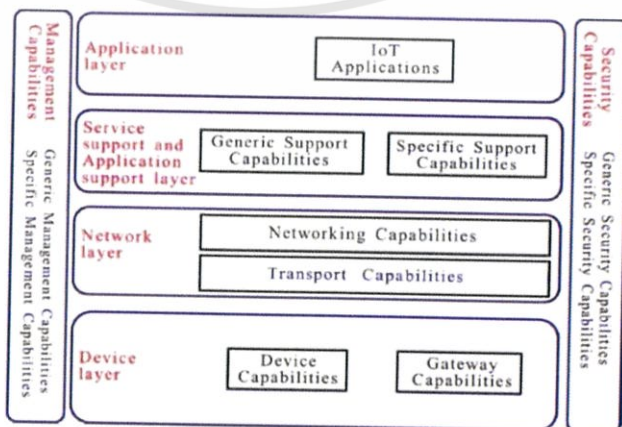
MQTT Protocol จะประกอบไปด้วย Broker, Publisher และ Subscriber ซึ่งแต่ละตัวมีหน้าที่ดังนี้

- 1) Broker จะทำหน้าที่เป็นตัวกลางคอยจัดการ message โดยอ้างอิงด้วย topic ซึ่งโปรแกรมที่ใช้ทำ MQTT Broker มีหลายค่ายให้ใช้งาน หนึ่งในนั้นคือโปรแกรม Mosquitto เป็น OpenSource MQTT Broker สนับสนุน MQTT Broker V3.1/3.1.1
- 2) Subscriber จะทำหน้าที่คอยดูการเปลี่ยนแปลงของ message ที่อ้างอิงด้วย topic เช่น สมมติว่าถ้าหัวข้อที่สนใจมีการเปลี่ยนแปลงก็จะดึง data มาใช้งานได้
- 3) Publisher จะทำหน้าที่ส่งข้อมูลไปยังหัวข้ออื่นๆ

โพรโทคอล MQTT นี้สามารถนำไปพัฒนาอุปกรณ์ที่เกี่ยวข้องต่างๆ กับ Internet of Things และ Cloud service ต่างๆ ได้อีกมากมาย อาทิ smart home, smart office, smart farm เป็นต้น ที่กล่าวมาทั้งหมดนี้คือส่วนสำคัญต่างๆ ของ Internet of Things ที่กำลังเกิดขึ้นและเป็น trend ที่กำลังมาแรงอยู่ในขณะนี้ สิ่งสำคัญคือ IoT ไม่ได้หมายถึงอุปกรณ์ Smart device เช่น นาฬิกาอัจฉริยะ (Apple Watch) หรือสายรัดข้อมือเพื่อสุขภาพเท่านั้น แต่มันยังครอบคลุมไปถึงอุปกรณ์ต่างๆ อีกหลากหลายล้านตัวกว้างไกลไปยังหลากหลายอุตสาหกรรมอีกด้วย โดยในอนาคตเราจะได้เห็นไมโครเวฟคุยกับตู้เย็นให้สั่งอาหารมาเติม เครื่องซักผ้าคุยกับทีวีบอกคุณว่าผ้าซักเสร็จแล้ว สายรัดข้อมือจะคุยกับรถพยาบาลแจ้งให้ปรับตัวผู้ป่วยที่กำลังหัวใจวาย สิ่งต่างๆ เหล่านี้คืออนาคตของ Internet of Things ที่กำลังจะสามารถคุยกันได้

2.2.7 IoT Reference Model

IoT Reference Model สามารถแบ่งได้เป็น 4 Layers ตามรูปที่ 2.10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการรูปที่ 2.10 IoT Reference Model ให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแต่ละ Layer แบ่งได้ดังนี้ คือ

Layer 1 Application layer ประกอบไปด้วย IoT applications.

Layer 2 Service support and application support layer ประกอบไปด้วย

- Generic support capabilities เป็นความสามารถทั่วไปใช้กับ IoT applications เช่น data processing หรือ data storage โดย Generic capabilities อาจถูกเรียกใช้โดย specific support capabilities เพื่อสร้างความสามารถเฉพาะอื่นๆ ขึ้นใหม่

- Specific support capabilities เป็นความสามารถเฉพาะตามความต้องการของ IoT applications บางชนิด

Layer 3 Network layer ประกอบไปด้วย

- Networking capabilities: ใช้ควบคุม functions การเชื่อมต่อของ network เช่น access and transport resource control functions, mobility management or authentication และ authorization and accounting เป็นต้น

- Transport capabilities: ใช้ควบคุมการส่งผ่านของ IoT service และข้อมูลเฉพาะของ application รวมทั้งการส่งผ่านของข้อมูล IoT-related control and management

Layer 4 Device layer ประกอบไปด้วย

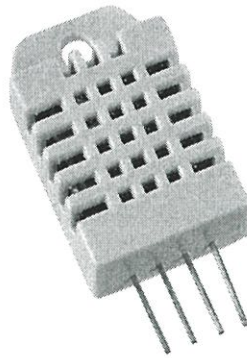
- Device capabilities: ความสามารถของตัวอุปกรณ์ต่างๆ

- Gateway capabilities: เป็นอุปกรณ์ที่มีความสามารถในการเชื่อมต่อเครือข่ายต่างๆ เข้าด้วยกัน โดยสามารถเชื่อมต่อ LAN หลายๆ เครือข่ายที่ใช้โปรโตคอลต่างกัน และใช้สื่อส่งข้อมูลต่างชนิดกันได้อย่างไม่มีขีดจำกัด [1]

2.3 เซนเซอร์วัดอุณหภูมิและความชื้น DHT22

อุปกรณ์ตรวจจับสัญญาณหรือปริมาณทางฟิสิกส์ต่างๆ เช่น อุณหภูมิ ความชื้น เสียง แสง และการสัมผัส เป็นต้น แล้วเปลี่ยนให้อยู่ในรูปของสัญญาณหรือข้อมูลที่สอดคล้องและเหมาะสมกับส่วนของการกำหนดเงื่อนไขทางสัญญาณ โดยเซนเซอร์ที่ใช้ในงานวิจัยนี้คือ DHT22 ซึ่งเป็น High Accuracy Digital Temperature and Humidity Sensor คือสามารถวัดได้ทั้งอุณหภูมิและความชื้น

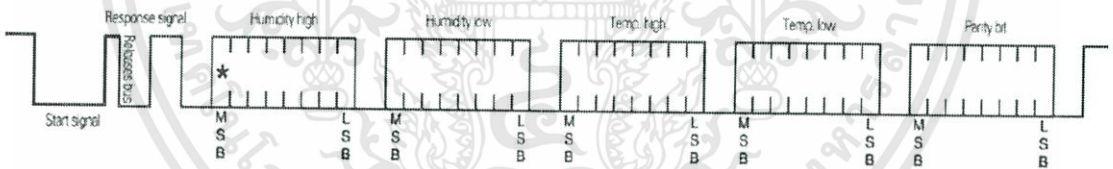
DHT22 เป็นอุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ (Temperature & Relative Humidity Sensor) ซึ่งสามารถนำมาประยุกต์ใช้งานทางด้านระบบสมองกลฝังตัวได้หลากหลาย เช่น การวัดและการควบคุมอุณหภูมิและความชื้น ระบบบันทึกข้อมูลเกี่ยวกับอุณหภูมิและความชื้นในห้อง ให้ค่าเป็นแบบดิจิทัล ใช้สัญญาณดิจิทัลเพียงเส้นเดียวในการเชื่อมต่อกับ Arduino เพื่ออ่านค่าจากเซนเซอร์



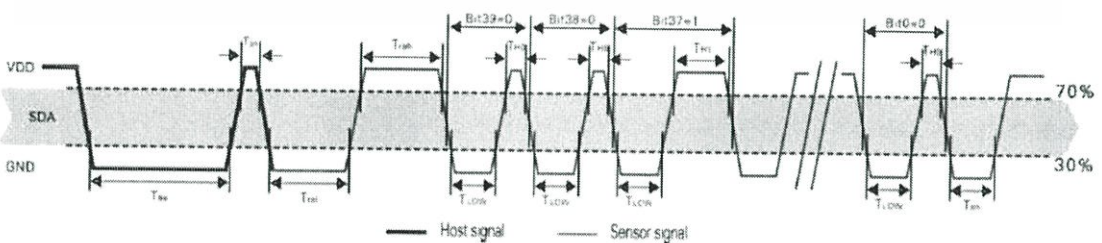
รูปที่ 2.11 เซนเซอร์วัดอุณหภูมิและความชื้น DHT22

ในการอ่านข้อมูลจากไอซีนั้น จะใช้ขาสัญญาณเพียงเส้นเดียวคือ DATA (หรือ SDA) แบบสองทิศทาง และในสถานะปกติสัญญาณ DATA จะเป็น HIGH ในการอ่านข้อมูลแต่ละครั้งไมโครคอนโทรลเลอร์จะต้องกำหนดให้ขา DATA เป็นเอาต์พุต และสร้างบิต START ซึ่งจะต้องเป็น LOW อย่างน้อย 800 μ s จากนั้นจึงให้เป็น HIGH อย่างน้อย 20 μ s หลังจากนั้นเป็นการรอการตอบกลับ (response) ของไอซี ซึ่งขา DATA จะต้องถูกเปลี่ยนเป็นอินพุต

เมื่อการตอบกลับเริ่มต้น ไอซีจะมีลักษณะสัญญาณเป็น LOW และ HIGH สลับกันช่วงละ 80 μ s โดยประมาณ (เรียกว่า Response Bit) จากนั้นจึงจะเป็นการส่งข้อมูลที่ละบิต รวม 40 บิต (ช่วง LOW ตามด้วยช่วง HIGH) ช่วง LOW ของแต่ละบิตจะกว้างเท่ากัน แต่จะต่างกันในช่วง HIGH สำหรับบิตที่มีค่าเป็น 0 หรือ 1 (ใช้ความกว้างช่วง HIGH ในการจำแนกค่าของบิต) [8]



รูปที่ 2.12 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีทั้งหมด 5 ไบต์ (40 บิต) สองไบต์แรกสำหรับความชื้น สองไบต์ต่อมาสำหรับอุณหภูมิและไบต์สุดท้ายเป็น checksum หรือ parity bits



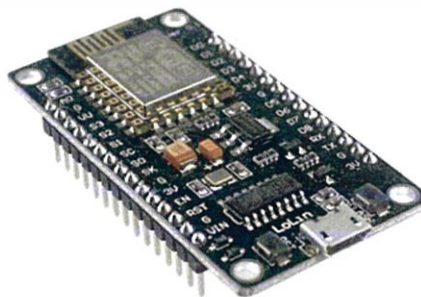
รูปที่ 2.13 ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH

ตารางที่ 2.2 คุณสมบัติของ Sensor วัดอุณหภูมิและความชื้น DHT22

Accuracy humidity	±2% RH (Max +-5%RH)
Accuracy temperature	±0.5 °C
Resolution or sensitivity humidity	0.1% RH
Resolution or sensitivity temperature	0.1 °C
Repeatability humidity	±1% RH
Repeatability temperature	±0.2 °C
Humidity hysteresis	±0.3% RH
Long-term Stability	±0.5% RH/year
Sensing period Average	2s
Temperature range	-40-80 °C
Humidity from	0-100% RH
Input Voltage	3.3-6 V
Standby current	40-50 µA
Measuring current	1-1.5 mA

2.4 NodeMCU Board

NodeMCU เป็นบอร์ดไมโครคอนโทรลเลอร์ที่เป็นบอร์ดอินพุตเอาต์พุตมาพร้อมกับโมดูล WIFI (ESP8266) ซึ่งเป็นหัวใจสำคัญในการใช้เชื่อมต่อกับอินเทอร์เน็ตนั่นเอง ตัวโมดูล ESP8266 นั้นมีอยู่ด้วยกันหลายรุ่นมาก ตั้งแต่เวอร์ชันแรกที่เป็น ESP-01 ไปจนถึงปัจจุบันมีถึง ESP-12 และที่ฝังอยู่ใน NodeMCU version 1 นั้นเป็น ESP-12 แต่ใน version 2 และ 3 นั้นจะใช้เป็น ESP-12E แทน ซึ่งการใช้งานโดยรวมก็ไม่แตกต่างกันมากนัก NodeMCU นั้นมีลักษณะคล้ายกับ Arduino ตรงที่มีพอร์ตอินพุตเอาต์พุต built in มาในตัว สามารถเขียนโปรแกรมคอนโทรลอุปกรณ์ I/O ได้โดยไม่ต้องผ่านอุปกรณ์อื่นๆ และสามารถเขียน Arduino IDE ใช้งานร่วมกับ NodeMCU ได้ จึงสามารถใช้ภาษา C ในการเขียนโปรแกรมได้ ทำให้เราสามารถใช้งานมันได้หลากหลายมากยิ่งขึ้น NodeMCU สามารถทำอะไรได้หลายอย่าง โดยเฉพาะเรื่องที่เกี่ยวข้องกับ IoT ไม่ว่าจะเป็นการทำ Web Server ขนาดเล็ก การควบคุมการเปิดปิดไฟผ่าน WiFi และอื่นๆ อีกมากมาย [4]



รูปที่ 2.14 NodeMCU Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 HTML และ Dreamweaver CS6

HTML เป็นภาษาคอมพิวเตอร์ในรูปแบบหนึ่งที่ใช้สำหรับสร้างหน้าเว็บ (Web Page) เพื่อเก็บข้อมูลข่าวสารที่ต้องการในรูปของเอกสารไฮเปอร์เท็กซ์ (Hypertext) ที่มีคุณสมบัติสามารถเชื่อมโยงหน้าเว็บหนึ่งไปยังหน้าเว็บอื่นๆ ได้ โดยโครงสร้างของ HTML จะมีตัวกำกับ หรือแท็ก (Tag) สำหรับใช้ในการควบคุมการแสดงผลของข้อความ รูปภาพ ตาราง และวัตถุอื่นๆ ผ่านเว็บเบราว์เซอร์ (Web Browser) [2]

โปรแกรม Dreamweaver CS6 เป็นโปรแกรมที่ใช้สำหรับสร้างและพัฒนาเว็บเพจให้มีประสิทธิภาพสูงและสะดวกรวดเร็ว ปัจจุบัน Dreamweaver ได้พัฒนาเป็น CS แล้ว ซึ่งเป็นโปรแกรมสร้างเว็บเพจแบบเสมือนจริงของค่าย Adobe ซึ่งช่วยให้ผู้ที่ต้องการสร้างเว็บเพจไม่ต้องเขียนภาษา HTML หรือโค้ดโปรแกรม หรือที่ศัพท์เทคนิคเรียกว่า "WYSIWYG" โปรแกรม Dreamweaver นี้มีฟังก์ชันที่ทำให้ผู้ใช้สามารถจัดวางข้อความ รูปภาพ ตาราง ฟอรัม วิดีโอรวมถึงองค์ประกอบอื่นๆ ภายในเว็บเพจได้อย่างสวยงามตามที่ผู้ต้องการ โดยไม่ต้องใช้ภาษาสคริปต์ที่ยุ่งยากซับซ้อนเหมือนเมื่อก่อน Dreamweaver มีทั้งในระบบปฏิบัติการแมคอินทอช (Macintosh) และไมโครซอฟท์ วินโดวส์ อีกทั้งยังสามารถทำงานบนระบบปฏิบัติการแบบยูนิกซ์ผ่านโปรแกรมจำลองอย่าง WINE ได้อีกด้วย ซึ่งเวอร์ชันล่าสุดของโปรแกรมตัวนี้คือ Adobe Dreamweaver CS6 [5]

2.6 งานวิจัยที่เกี่ยวข้อง

ในปี ค.ศ. 2014 ผศ.ดร.สุรเชษฐ์ กานต์ประชา และคณะได้ศึกษาหลักการของการตรวจจับตำแหน่งของรถไฟฟ้าโดยอาศัยระบบ GPS (Global Positioning System) ซึ่งในปัจจุบันมีอยู่ในอุปกรณ์สื่อสารโดยทั่วไป ยกตัวอย่างเช่น สมาร์ทโฟนและสมการทางคณิตศาสตร์มาใช้ในการคำนวณและประมาณเวลาในการมาถึงของรถไฟฟ้า ณ บ้ายจอดรถ โดยมีการออกแบบในส่วนของส่วนติดต่อผู้ใช้ผ่านโปรแกรมประยุกต์สำหรับผู้ใช้สมาร์ทโฟน เพื่อให้ผู้ใช้บริการรถไฟฟ้าสามารถประมาณเวลาและตำแหน่งของรถไฟฟ้าที่ต้องการจะใช้บริการโดยอาศัยสมาร์ทโฟน และมีการออกแบบในส่วนของบ้าย Seven Segment เพื่อใช้ในการบอกเวลาการมาถึงของรถไฟฟ้า ณ บ้ายจอดรถ ซึ่งจากการทดสอบการทำงาน พบว่า ระบบที่พัฒนาขึ้นสามารถระบุเวลาที่มาถึงของรถไฟฟ้าได้อย่างถูกต้อง นอกจากนี้ ในโครงการวิจัยขึ้นนี้ได้มีการนำเอาตรรกศาสตร์คลุมเครือมาประยุกต์ใช้ในการลดปริมาณการส่งข้อมูลระหว่างสมาร์ทโฟนกับเครื่องบริการเว็บ (WebServer) ซึ่งจำเป็นจะต้องมีการติดต่อสื่อสารผ่านระบบโทรศัพท์มือถือ 3G ซึ่งจากการออกแบบตรรกศาสตร์คลุมเครือ พบว่า สามารถลดปริมาณการส่งข้อมูลผ่านระบบ 3G ได้ โดยที่ความถูกต้องของข้อมูลยังไม่เปลี่ยนแปลง ซึ่งนั่นก็หมายความว่า ผู้ใช้สามารถลดค่าใช้จ่ายในการส่งข้อมูลผ่านระบบ 3G สำหรับการใช้บริการโปรแกรมประยุกต์ที่ถูกพัฒนาขึ้นได้นั่นเอง [6]

ในปี ค.ศ. 2016 คุณสุวิทย์ ภูมิฤทธิกุล และคุณปานวิทย์ ฐะนะนุติ ได้ศึกษาและพัฒนาอุปกรณ์ต่างๆ ให้สามารถทำงานบนแนวคิดของ Internet of Things โดยมีการนำมาประยุกต์ใช้งานในด้านต่างๆ มากขึ้น เช่น การประยุกต์ใช้งานใน Smart City การประยุกต์ใช้งานในอุตสาหกรรมและเกษตรกรรม รวมไปถึงการเฝ้าระวังและการเตือนภัยต่อสุขภาพของมนุษย์ ในบทความวิชาการนี้ได้อธิบายนิยามของ IoT ซึ่งถูกนิยามโดย IEEE และกล่าวถึงงานวิจัยต่างๆ ในมุมมองที่เกี่ยวข้องกับการประยุกต์การใช้งาน IoT เพื่อการเฝ้าระวังและเตือนภัยต่อสุขภาพของมนุษย์ สุดท้ายบทความนำเสนอแนวทางของการนำหลักการการประมวลผลข้อมูลขนาดใหญ่ (Big Data) โดยการใช้โปรแกรม Hadoop

เพื่อวิเคราะห์ข้อมูลหรือสารสนเทศที่ถูกจัดเก็บไว้บนระบบ Cloud ซึ่งจะเป็นประโยชน์กับการเฝ้าระวังและการเตือนภัยต่อสุขภาพของมนุษย์แบบ Cloud Computing ในอนาคต [7]



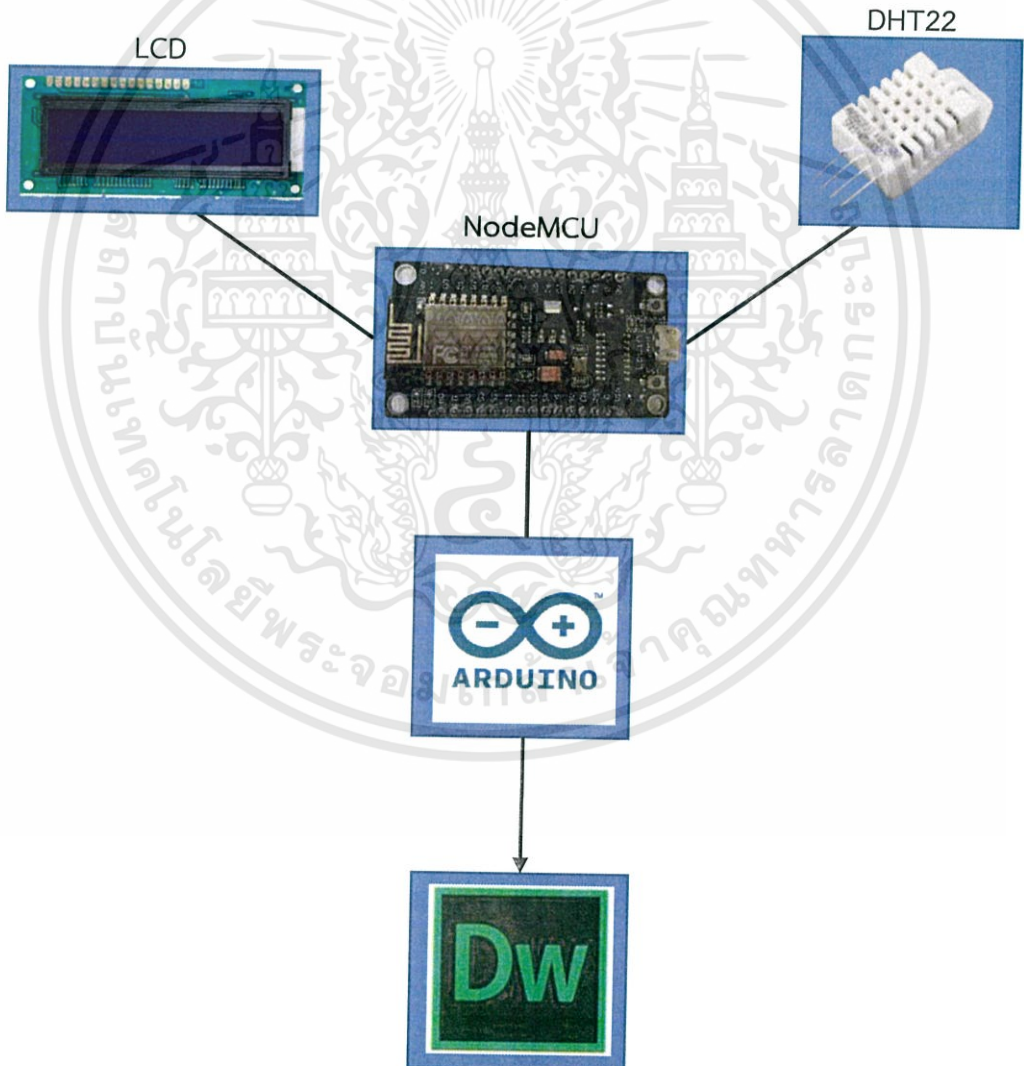
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงานวิจัย

ในบทนี้จะกล่าวถึงการออกแบบการติดต่อสื่อสารระหว่างตัวเซนเซอร์วัดอุณหภูมิและความชื้น จอ LCD และเว็บไซต์ โดยมีการส่งข้อมูลผ่านระบบเครือข่าย IoT ซึ่งใช้ NodeMCU ที่ เป็นหัวใจสำคัญในการควบคุมการทำงานของเซนเซอร์ให้สามารถการเชื่อมต่อเข้ากับอินเทอร์เน็ตผ่าน WIFI เพื่อรับ-ส่งข้อมูล แล้วแสดงค่าอุณหภูมิและความชื้นบนจอ LCD และบนเว็บไซต์ อีกทั้งยังสามารถควบคุมการทำงานของอุปกรณ์ผ่านเว็บไซต์ได้อีกด้วย ซึ่งมีรายละเอียดดังต่อไปนี้

3.1 ขั้นตอนการออกแบบวงจร



รูปที่ 3.1 แผนภาพไดอะแกรมของการออกแบบวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 วัสดุอุปกรณ์ที่ใช้ในการทดลอง

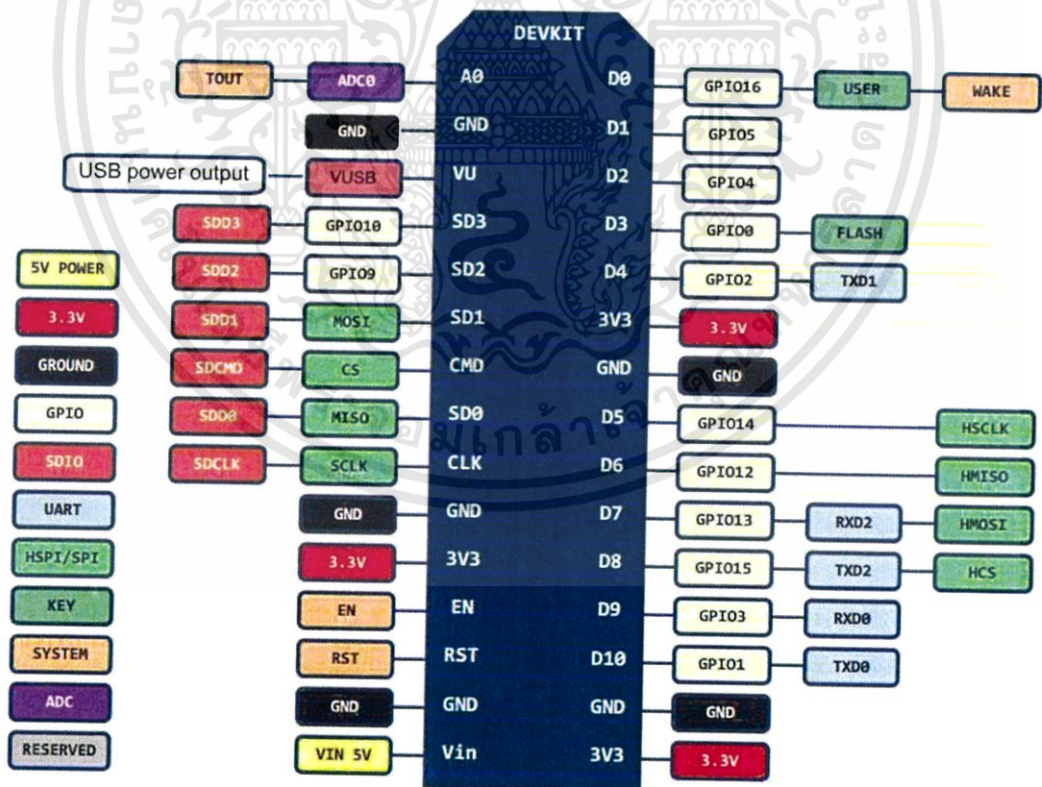
1) NodeMCU

ในส่วนนี้จะใช้ NodeMCU ในการควบคุมการทำงานของเซนเซอร์ และจอ LCD โดยการเขียนโปรแกรมคำสั่ง Arduino IDE สามารถแสดงได้ดังรูปที่ 3.2 [4]



รูปที่ 3.2 NodeMCU

โดยมีแผนผังของขาเพื่อแสดงการทำงานดังรูปที่ 3.3



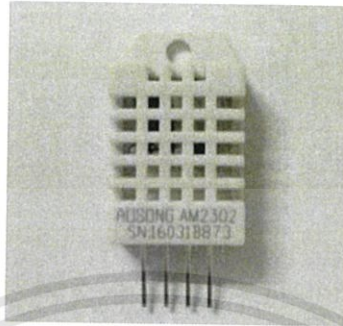
D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

รูปที่ 3.3 ขาการทำงานของ NodeMCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) DHT22 Temperature & Relative Humidity Sensor

ในส่วนเซนเซอร์จะใช้เป็น DHT22 ที่เป็นอุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสามารถแสดงได้ดังรูปที่ 3.4 [8]



รูปที่ 3.4 DHT22 Temperature & Relative Humidity Sensor

การทำงานของ DHT22 Temperature & Relative Humidity Sensor

- ใช้แรงดันไฟเลี้ยงได้ในช่วง 3.3V ถึง 5.5V DC (ตั้งนั้นจึงใช้ได้กับ 3.3V และ 5V)
- วัดอุณหภูมิได้ในช่วง $-40\text{ }^{\circ}\text{C}$ ถึง $80\text{ }^{\circ}\text{C}$ ($\pm 0.5\text{ }^{\circ}\text{C}$ accuracy)
- วัดความชื้นสัมพัทธ์ได้ในช่วง 0-100 RH% (2-5% accuracy)
- อัตราการวัดสูงสุด 0.5 Hz
- คอนเนกเตอร์แบบ 4 ขา (0.1'' / 2.54mm spacing)

Pin 1 = VCC

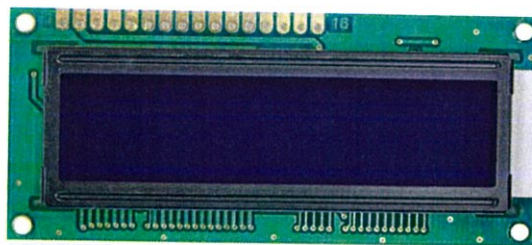
Pin 2 = SDA (Serial data, bidirectional)

Pin 3 = N.C. (Not Connect)

Pin 4 = GND

3) LCD (I²C)

ในส่วนของจอ Liquid Crystal Display (LCD) ซึ่งเป็นจอแสดงผลแบบดิจิตอลที่มีขนาด 16x2 หรือ 16 ตัวอักษร 2 บรรทัด แต่เนื่องจากขาของ NodeMCU มีไม่พอที่จะต่อกับจอ LCD ได้ จึงใช้ I²C เข้ามาช่วยในส่วนเชื่อมต่อกับจอแสดงผล LCD แสดงได้ดังรูปที่ 3.5

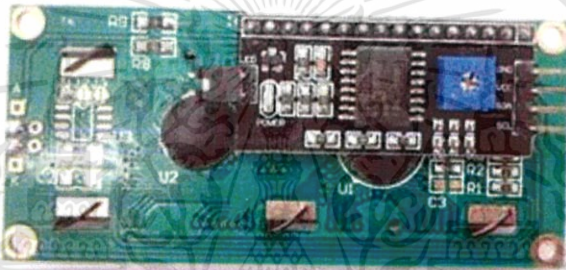


รูปที่ 3.5 จอแสดงผล LCD

จอ LCD ที่มีการเชื่อมต่อแบบ I²C หรือเรียกอีกอย่างว่าการเชื่อมต่อแบบอนุกรมจะเป็นจอ LCD ธรรมดาทั่วไปที่มาพร้อมกับบอร์ด I²C Bus ที่ทำให้การใช้งานได้สะดวกยิ่งขึ้น ในรูปแบบ I²C จะใช้ขาในการเชื่อมต่อกับไมโครคอนโทรลเลอร์เพียง 4 ขา ซึ่งทำให้ใช้งานได้ง่ายและสะดวกมากยิ่งขึ้น สามารถแสดงได้จากตารางด้านล่างที่ 3.1 และรูปที่ 3.6 แสดงการเชื่อมต่อ I²C ของหน้าจอ LCD

ตารางที่ 3.1 การควบคุมการแสดงผลของจอ LCD (I²C)

Pin	No	Symbol description
1	GND	Ground
2	VCC	5VDC
3	SDA	Serial data
4	SCL	Serial clock



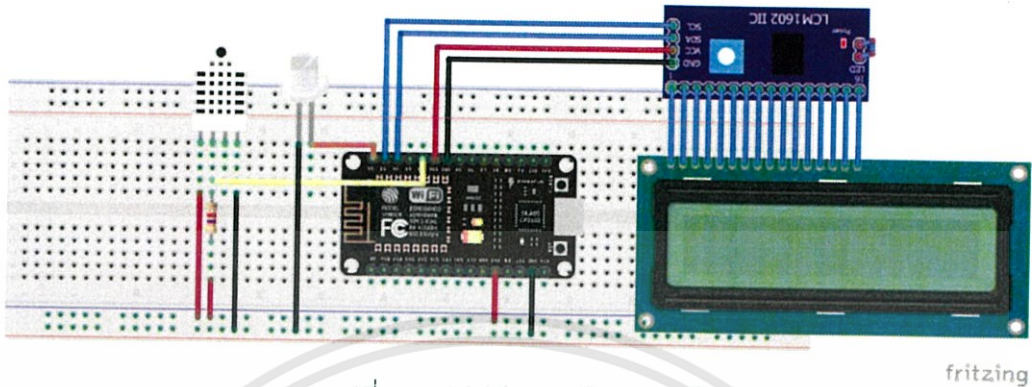
รูปที่ 3.6 การเชื่อมต่อ I²C

จากตารางที่ 3.1 อธิบายได้ว่า การควบคุมหรือการสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) ในตัวอยู่แล้ว ผู้ใช้สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของ LCD (I²C) เช่นเดียวกันกับจอ LCD แบบธรรมดา คือรหัสคำสั่งที่ใช้ในการควบคุมนั้นเหมือนกัน แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล ซึ่งจอ LCD 16x2 ที่มีการส่งข้อมูลรูปแบบ I²C จะใช้ขาเพียง 4 ขาเท่านั้นในการเชื่อมต่อ ได้แก่

- 1) GND เป็น Ground ใช้ต่อระหว่าง Ground ของระบบไมโครคอนโทรลเลอร์กับ LCD
- 2) VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับ LCD มีขนาด +5VDC
- 3) SDA (Serial Data) เป็นขาที่ใช้ในการรับส่งข้อมูล
- 4) SCL (Serial Clock) เป็นขาสัญญาณนาฬิกาในการรับส่งข้อมูล

3.2 วิธีการทดลอง

1) ต่อวงจรดังรูปที่ 3.7



รูปที่ 3.7 การต่อวงจรกับอุปกรณ์

2) ติดตั้ง Board NodeMCU ให้กับ Arduino IDE โดยมีขั้นตอนการติดตั้งดังนี้

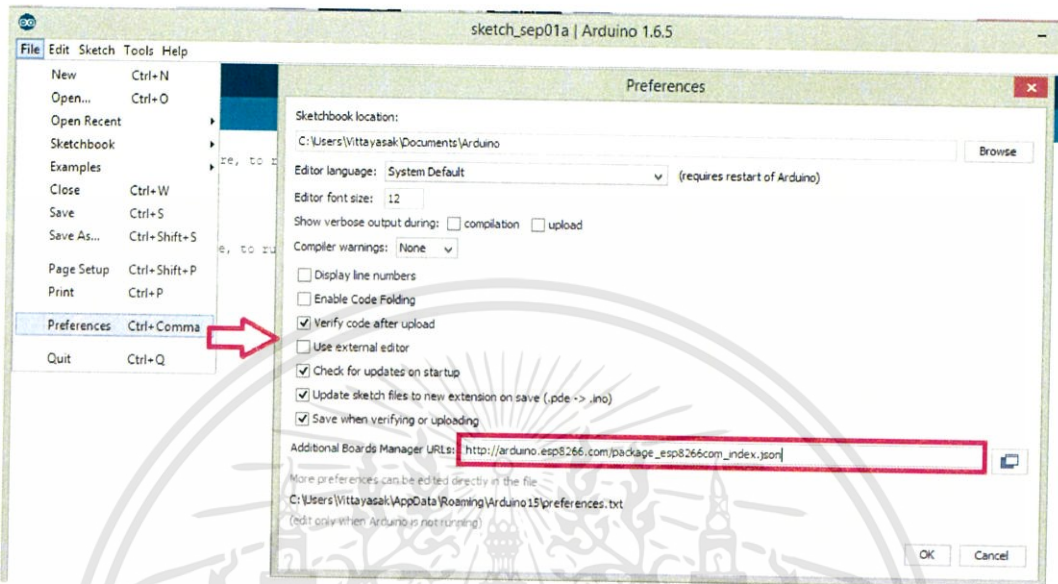
- ดาวน์โหลด Arduino IDE จากเว็บไซต์ <https://www.arduino.cc/en/Main/Software> แล้วติดตั้งโปรแกรมดังรูปที่ 3.8



รูปที่ 3.8 หน้าต่างดาวน์โหลด Arduino IDE

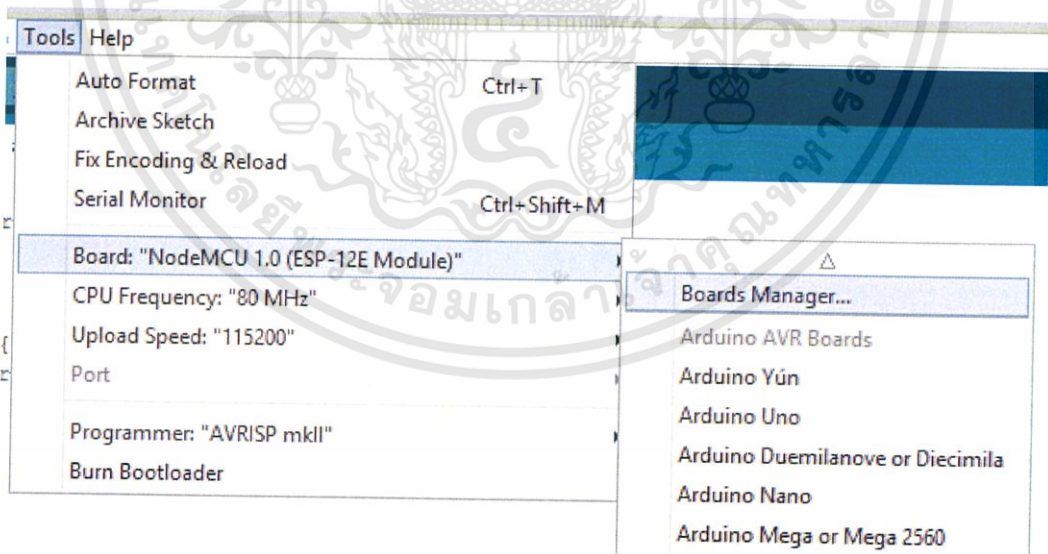
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพิ่ม Board ของ ESP8266 โดยการเปิดโปรแกรม เลือก File-> Preferences แล้วใส่ URL นี้ http://arduino.esp8266.com/package_esp8266com_index.json ลงไปในช่องกรอบบัสแสดงดังรูปที่ 3.9



รูปที่ 3.9 การเพิ่ม Board ของ ESP8266

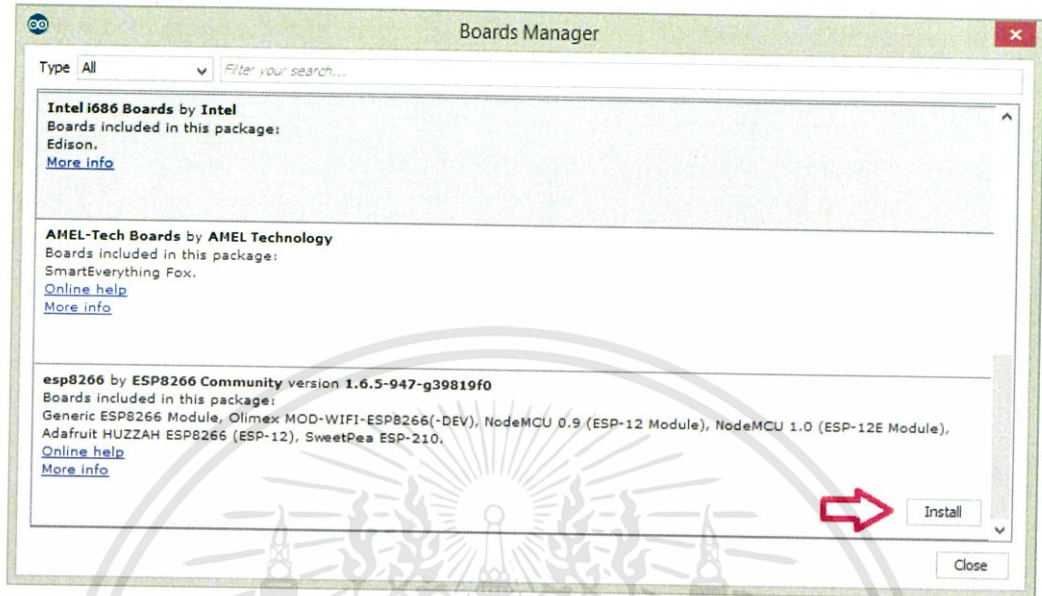
- กด OK แล้วกลับมาที่หน้าหลัก เลือก Tool -> Boards Manager ดังรูปที่ 3.10



รูปที่ 3.10 หน้าต่างฟังก์ชันเครื่องมือ

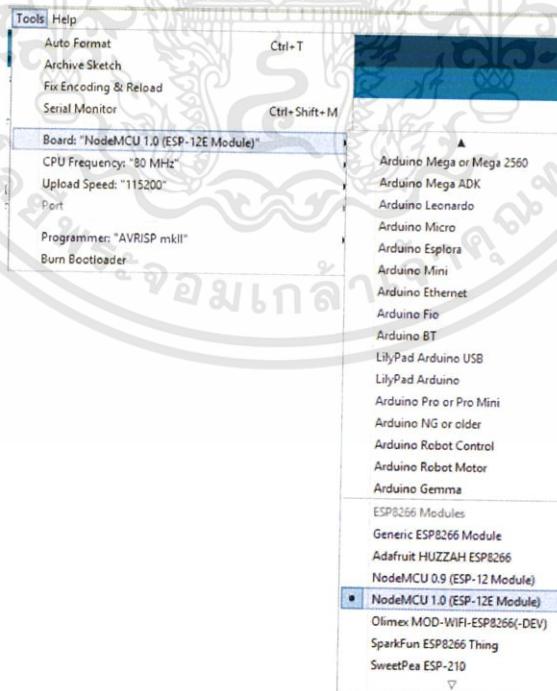
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หากคำว่า esp8266 by ESP8266 Community คลิกที่ปุ่ม Install ดังรูปที่ 3.11 จากนั้นรอโหลดจนเสร็จ



รูปที่ 3.11 หน้าต่าง Boards Manager

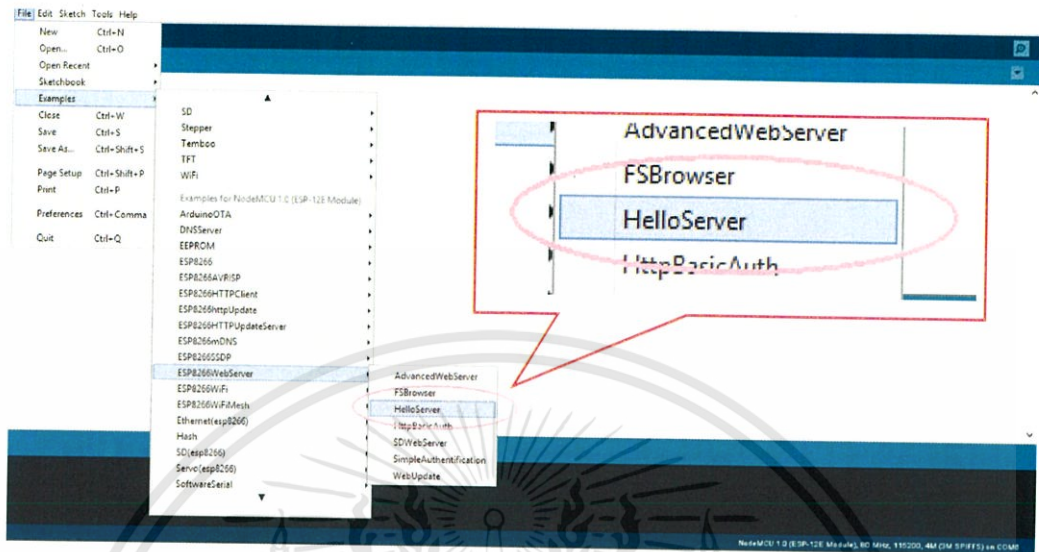
- เมื่อเสร็จแล้วกลับไปหน้าหลัก เลือกบอร์ด NodeMCU 1.0 ESP8266-12E ดังรูปที่ 3.12 เพื่อจบการติดตั้ง



รูปที่ 3.12 การเลือกบอร์ด NodeMCU

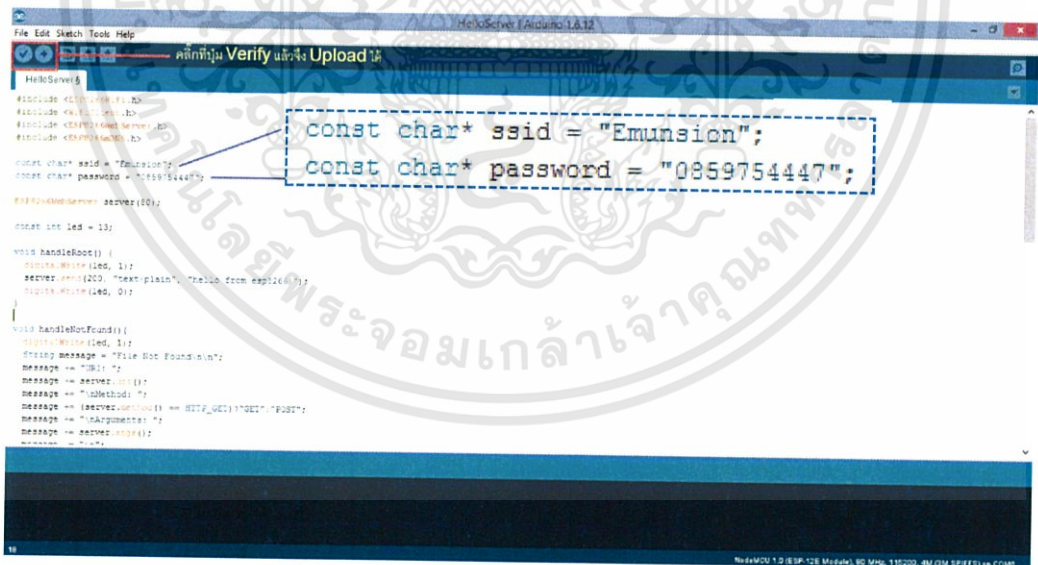
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การทดสอบการเชื่อมต่อของ NodeMCU กับ WIFI โดยเลือกที่ File --> Examples --> ESP8266WebServer --> เลือก HelloServer ดังรูปที่ 3.13



รูปที่ 3.13 การทดสอบการเชื่อมต่อของ NodeMCU กับ WIFI

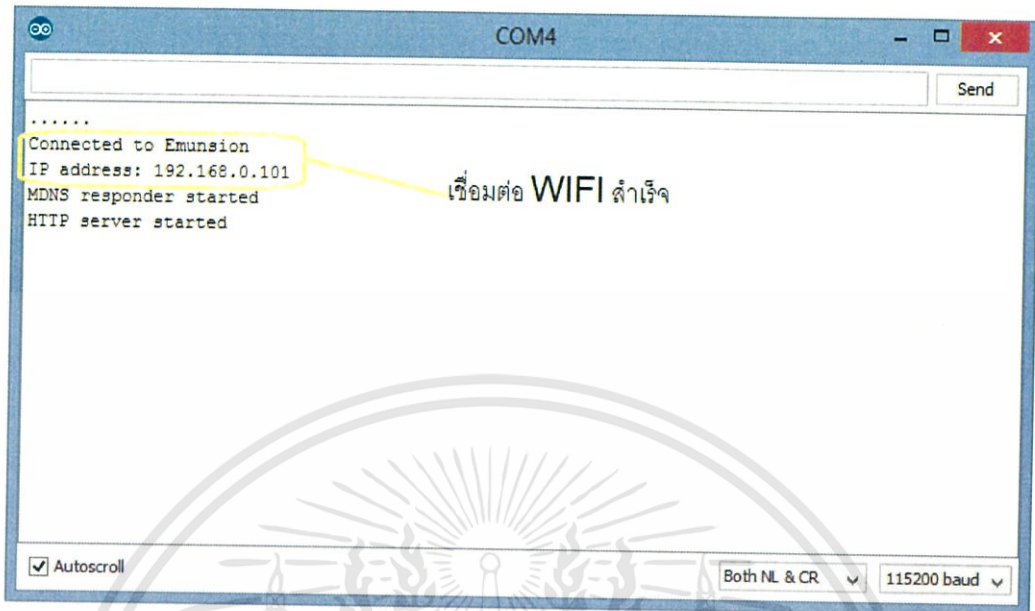
- จากนั้นให้ใส่ชื่อและรหัส WIFI ตามรูปด้านล่าง แล้วคลิกที่ปุ่ม Verify และทำการ Upload ได้ดังรูปที่ 3.14



รูปที่ 3.14 หน้าต่างของโค้ดตัวอย่างในการทดสอบการเชื่อมต่อของ NodeMCU กับ WIFI

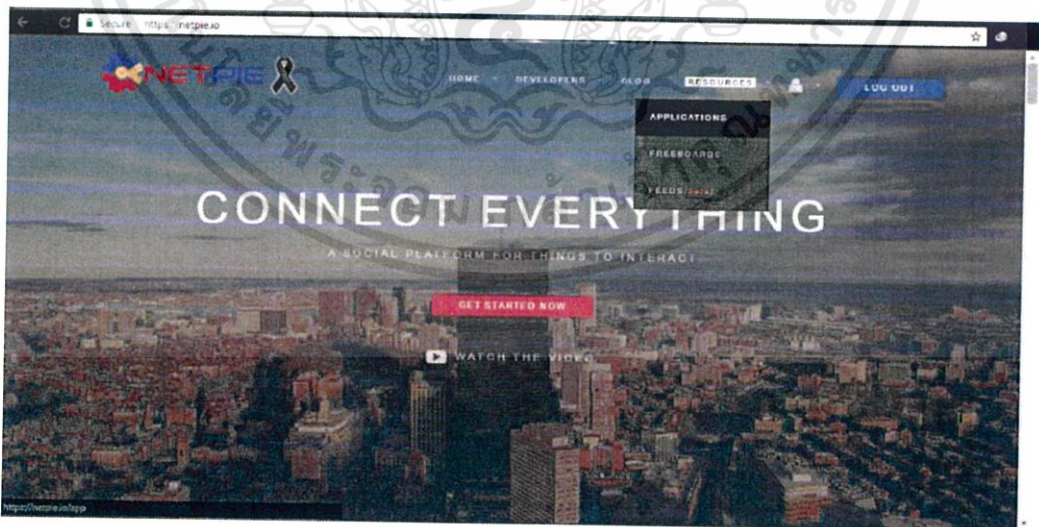
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลิกที่ Serial Monitor เพื่อดูข้อความที่แสดงผล ดังรูปที่ 3.15



รูปที่ 3.15 หน้าต่าง Serial Monitor

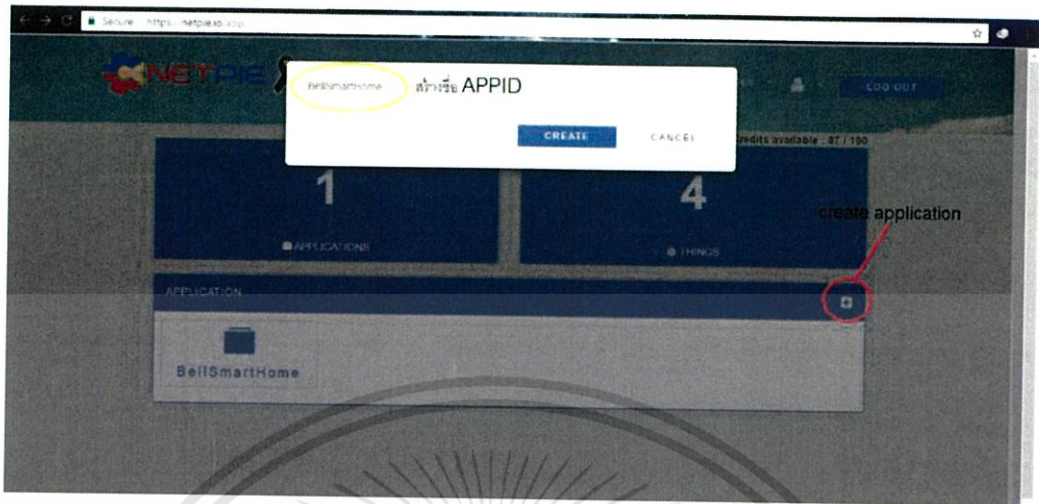
- 4) สร้าง IoT Cloud Platform ซึ่งเป็นส่วน MQTT Brokers ในการ Publish/Subscribe ของอุปกรณ์ ซึ่งในงานวิจัยนี้จะใช้ IoT Platform ของ netpie และ thingspeak ตามขั้นตอนดังนี้
- ไปที่เว็บไซต์ <https://netpie.io/> login --> RESOURCES --> เลือก APPLICATIONS ตามรูปที่ 3.16



รูปที่ 3.16 หน้าต่างเว็บไซต์ NETPIE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลือก create application แล้วใส่ชื่อ APPID ดังรูปที่ 3.17



รูปที่ 3.17 การ create application

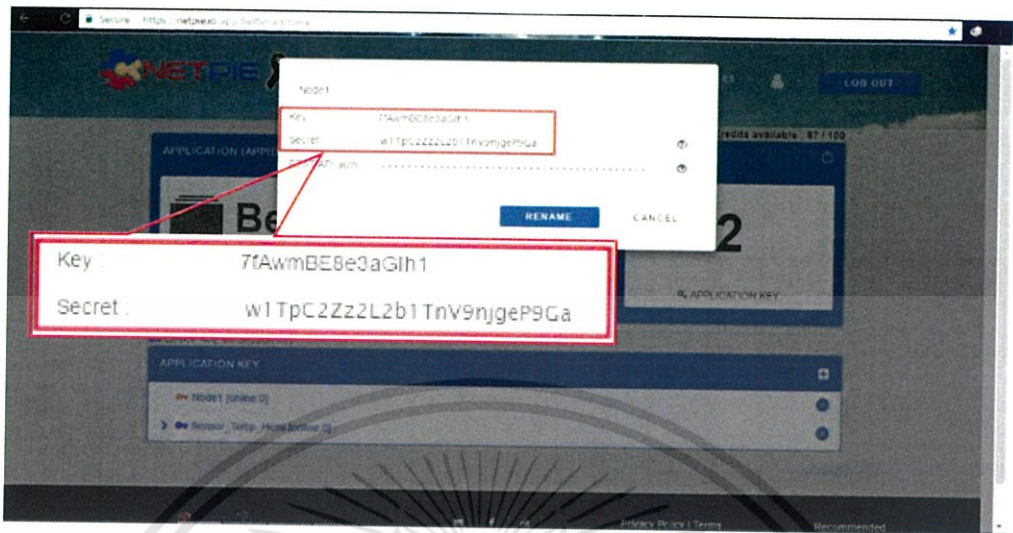
- ใส่ชื่อ application key แล้วเลือก Session key ดังรูปที่ 3.18



รูปที่ 3.18 การเลือกรูปแบบของการ create application

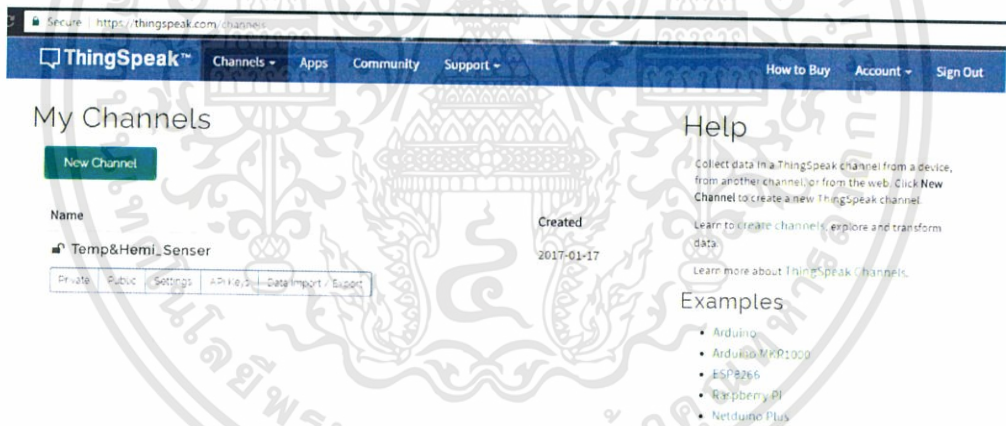
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บันทึกรหัส Key และ Secret ไว้ใน Arduino IDE ดังรูปที่ 3.19



รูปที่ 3.19 หน้าต่างของรหัส Key และ Secret

- ไปที่เว็บไซต์ <https://thingspeak.com/> login --> เลือก New Channel ดังรูปที่ 3.20



รูปที่ 3.20 หน้าต่างของเว็บไซต์ ThingSpeak

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการตั้งค่าของหน้า channel เสร็จแล้วให้กด save channel ดังรูปที่ 3.21

รูปที่ 3.21 การสร้าง Channel

- เมื่อสร้างช่อง Channel เสร็จ ให้คลิกไปยังช่องที่เราสร้างขึ้น แล้วกดที่แถบ API Key เพื่อบันทึกรหัส Key ไว้ใช้ใน Arduino IDE ดังรูปที่ 3.22

Temp&Hemi_Sensor

Channel ID: 215146
 Author: cmunion
 Access: Public

Private View Public View Channel Settings API Keys Data Import Export

Write API Key

Key YFCXDY0GD5YZ60JP

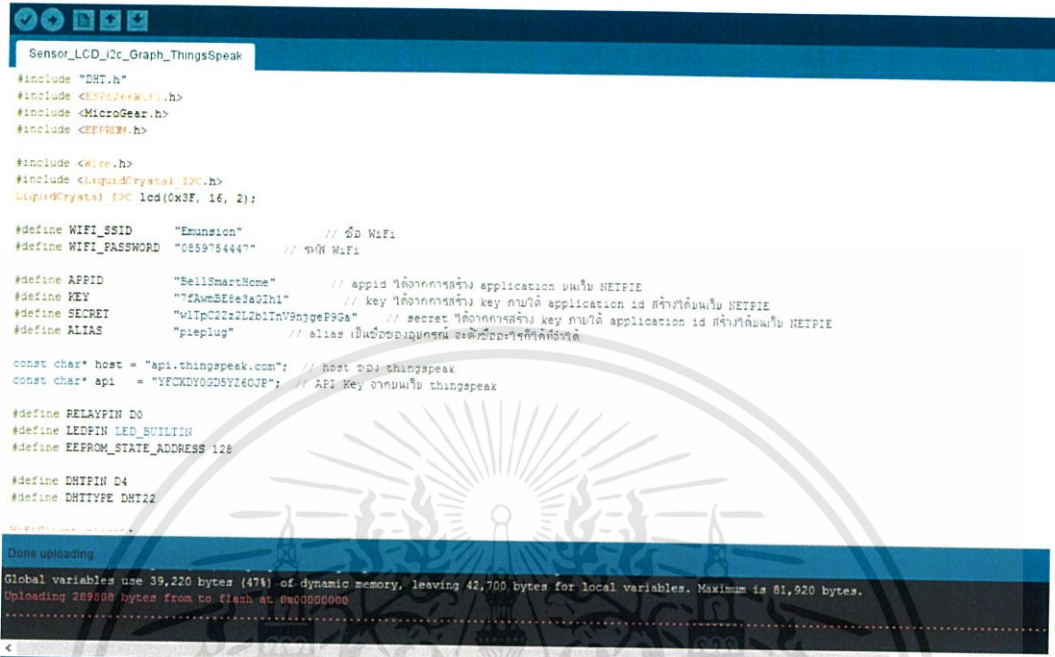
Generate New Write API Key

Read API Keys

Key AVMKV36E4BCJ0I81

รูปที่ 3.22 หน้าต่างของรหัส API Key

5) เปิดโปรแกรม Arduino IDE ได้การทำงาน จากนั้นคลิกที่ปุ่ม Verify แล้วจึง Upload ดังรูปที่ 3.23



```

Sensor_LCD_I2c_Graph_ThingsSpeak
#include "DHT.h"
#include <ESP8266WiFi.h>
#include <MicroGear.h>
#include <EEPROM.h>

#include <avr.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);

#define WIFI_SSID "Emunston" // ชื่อ WiFi
#define WIFI_PASSWORD "0859754447" // รหัส WiFi

#define APID "BellSmartHome" // apid สำหรับการสร้าง application บนเว็บ NETPIE
#define KEY "7FAwBSEe3e3e2h1" // key สำหรับการสร้าง key กายใช้ application id สร้างได้บน NETPIE
#define SECRET "w1IpC2Zz2L2e1TnV8n3geP9Ga" // secret สำหรับการสร้าง key กายใช้ application id สร้างได้บน NETPIE
#define ALIAS "pieplug" // alias เป็นชื่อของอุปกรณ์ จะด้วยชื่ออะไรก็ได้

const char* host = "api.thingspeak.com"; // host ของ thingspeak
const char* api = "YFCXDY0GDSYZ60JP"; // API Key จากเว็บ thingspeak

#define RELAYPIN D0
#define LEDPIN LED_BUILTIN
#define EEPROM_STATE_ADDRESS 128

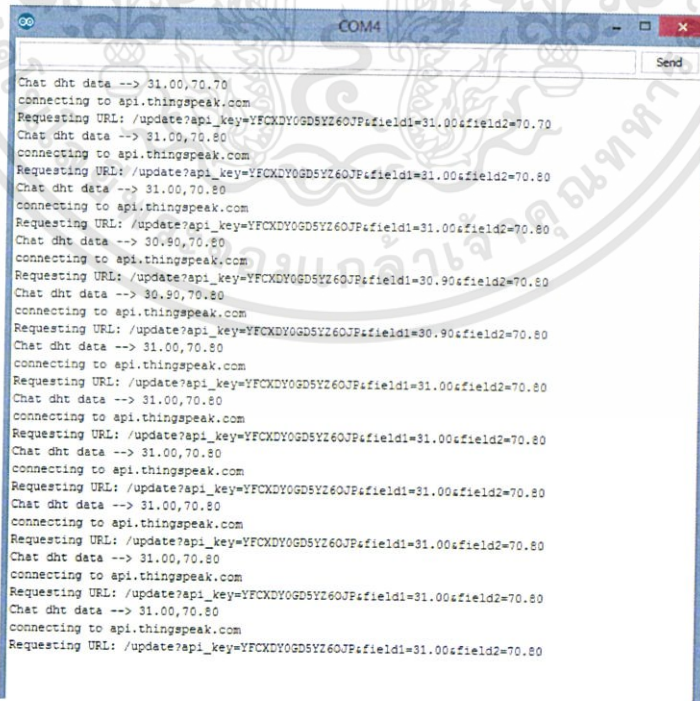
#define DHTPIN D4
#define DHTTYPE DHT22

Done uploading.
Global variables use 39,220 bytes (47%) of dynamic memory, leaving 42,700 bytes for local variables. Maximum is 81,920 bytes.
Uploading 287908 bytes from flash at 0x00000000

```

รูปที่ 3.23 หน้าต่างการ Upload

- หลังจากที่ได้ Upload เสร็จให้คลิกที่ Serial Monitor เพื่อดูข้อความที่แสดงผล ดังรูปที่ 3.24



```

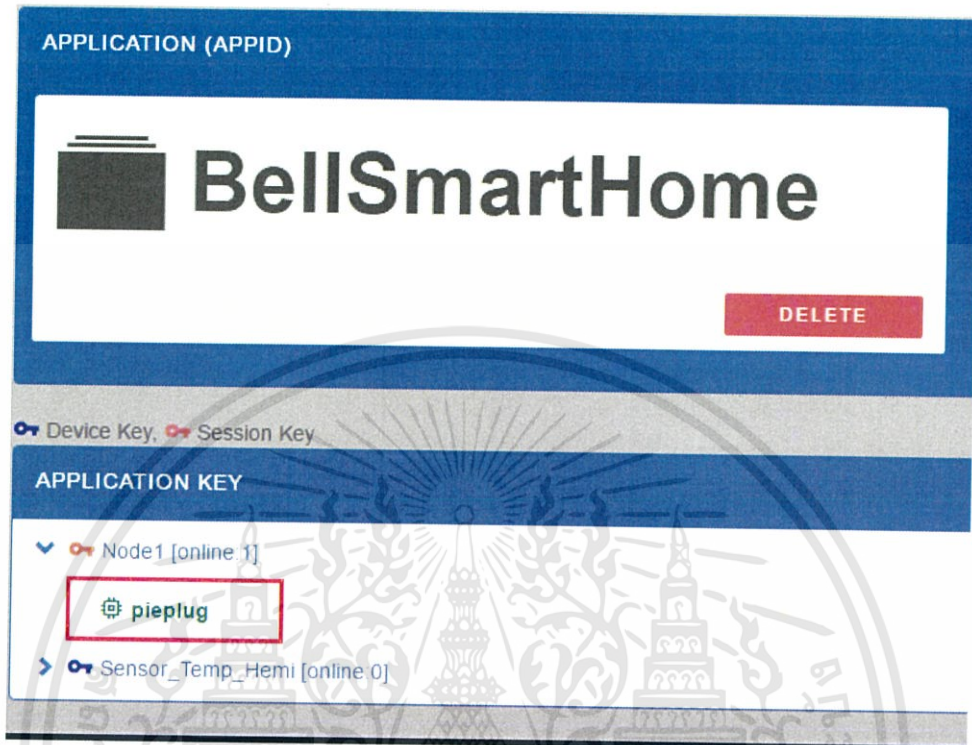
COM4
Chat dht data --> 31.00,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=31.00&field2=70.80
Chat dht data --> 31.00,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=31.00&field2=70.80
Chat dht data --> 31.00,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=31.00&field2=70.80
Chat dht data --> 30.90,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=30.90&field2=70.80
Chat dht data --> 30.90,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=30.90&field2=70.80
Chat dht data --> 31.00,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=31.00&field2=70.80
Chat dht data --> 31.00,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=31.00&field2=70.80
Chat dht data --> 31.00,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=31.00&field2=70.80
Chat dht data --> 31.00,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=31.00&field2=70.80
Chat dht data --> 31.00,70.80
connecting to api.thingspeak.com
Requesting URL: /update?api_key=YFCXDY0GDSYZ60JP&field1=31.00&field2=70.80

```

รูปที่ 3.24 หน้าต่างการทำงานของโค้ด Arduino IDE

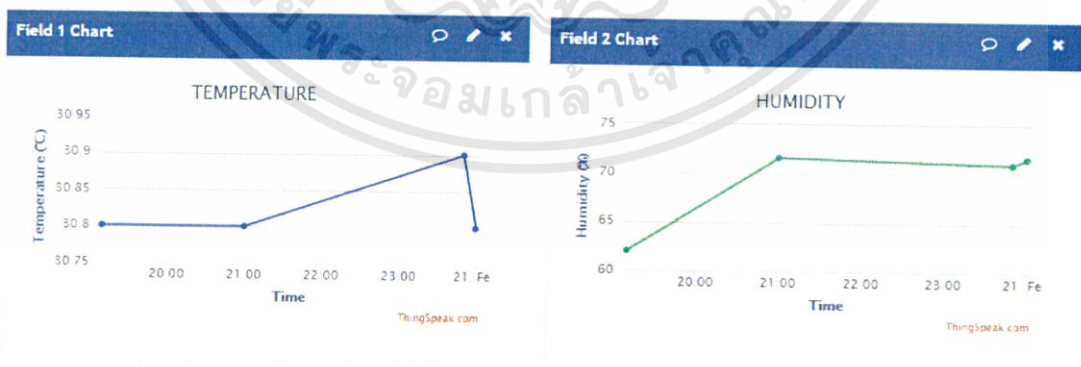
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) เข้าไปที่เว็บ NETPIE จะสังเกตเห็นว่ามีอุปกรณ์เชื่อมต่อชื่อว่า pieplug นั่นคือ NodeMCU ของเรากำลังเชื่อมต่อและส่งข้อมูลให้กับ NETPIE ดังรูปที่ 3.25 (เชื่อมต่อกับเว็บไซต์ที่จะสร้างต่อไป)



รูปที่ 3.25 หน้าต่างแสดงการเชื่อมต่อระหว่าง NodeMCU กับ NETPIE

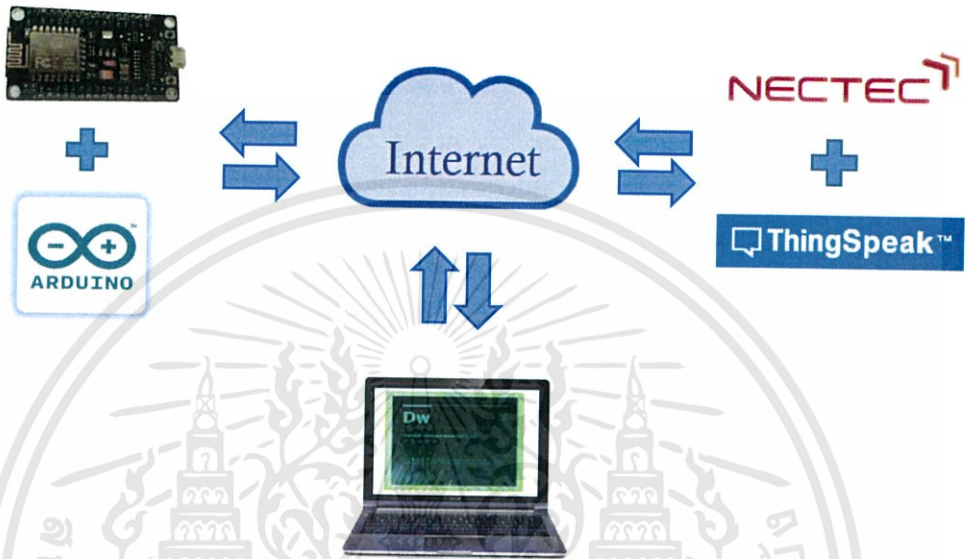
7) เข้าไปที่เว็บ thingspeak จะสังเกตเห็นว่าเซนเซอร์ของเรากำลังส่งค่าอุณหภูมิและความชื้นแล้วบันทึกข้อมูลในรูปแบบของกราฟในเว็บ thingspeak รูปที่ 3.26 (นำกราฟใส่ในเว็บไซต์ที่จะสร้างต่อไป)



รูปที่ 3.26 หน้าต่างการบันทึกอุณหภูมิและความชื้น

3.3 การประยุกต์ใช้โปรแกรม

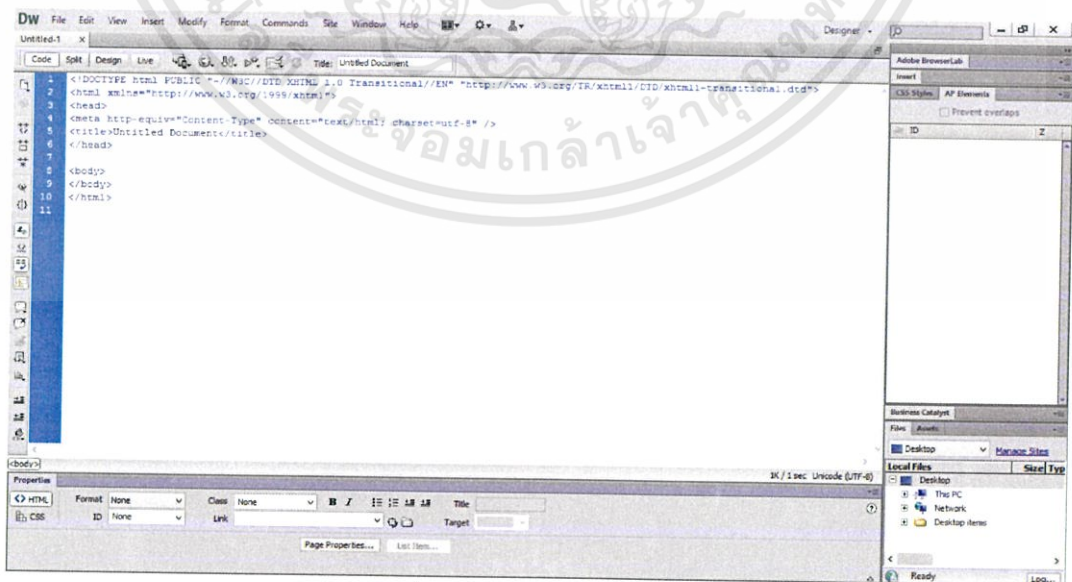
ในหัวข้อนี้จะกล่าวถึงการเขียน HTML ด้วยภาษา C โดยใช้โปรแกรม Dreamweaver CS6 เพื่อใช้เป็นตัวรับ-ส่งและแสดงข้อมูลทั้งหมดในหน้าเว็บไซต์ โดยมี NETPIE และ thingspeak เป็น Broker ซึ่งเป็นตัวกลางในการติดต่อสื่อสารและเชื่อมต่อของอุปกรณ์ต่างๆ ดังรูปที่ 3.27



รูปที่ 3.27 แผนภาพไดอะแกรมแสดงการเชื่อมต่อของอุปกรณ์ในระบบ IoT Sensor

3.3.1 ขั้นตอนการสร้างเว็บไซต์

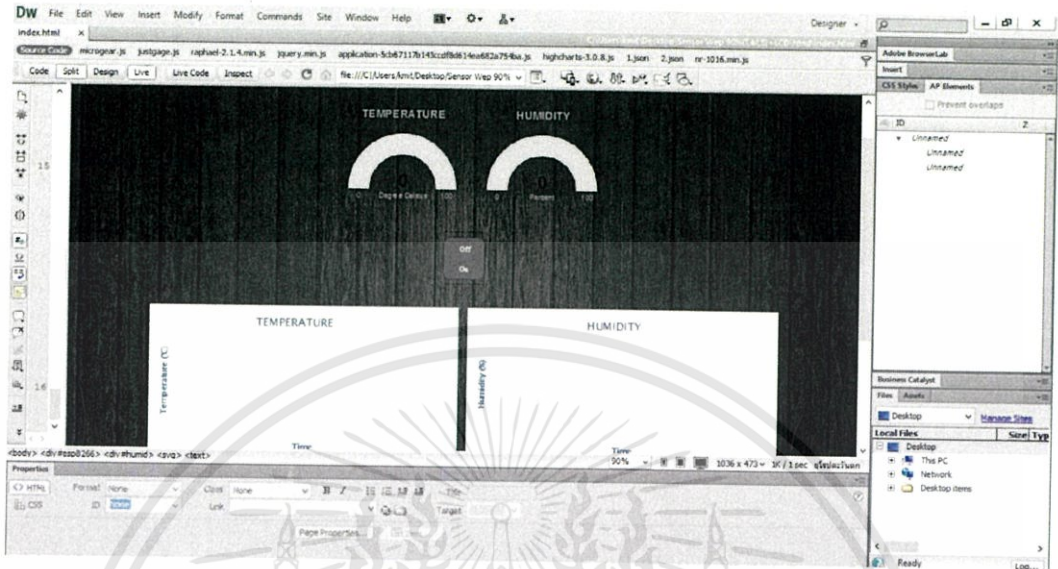
- เปิดโปรแกรม Dreamweaver CS6 สร้างหน้า HTML ดังรูปที่ 3.28



รูปที่ 3.28 หน้าต่างการทำงานโปรแกรม Dreamweaver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เขียนโค้ดสร้างหน้าเว็บไซต์ โดยการสร้างหน้าเว็บไซต์นี้จะประกอบด้วยส่วนของ
เกจอุณหภูมิกับเกจความชื้น สวิตช์เปิดปิดไฟ และกราฟอุณหภูมิและความชื้น ดังรูปที่ 3.29



รูปที่ 3.29 หน้าต่างการแสดงผลของโปรแกรม Dreamweaver

3.3.2 ขั้นตอนการอัปโหลด Web Server

1) สมัคร hosting จากเว็บไซต์ <https://hostinger.in.th/> แล้วสร้างชื่อโดเมน ดัง

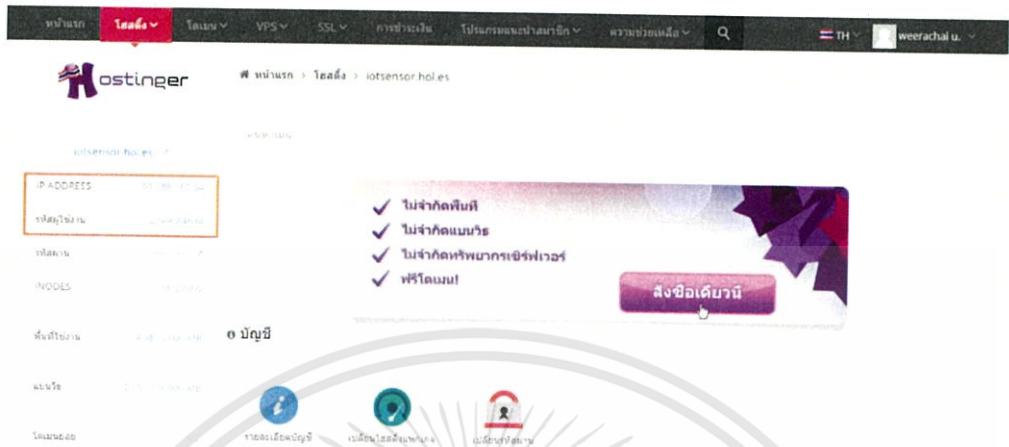
รูปที่ 3.30



รูปที่ 3.30 หน้าต่างของเว็บไซต์ hostinger

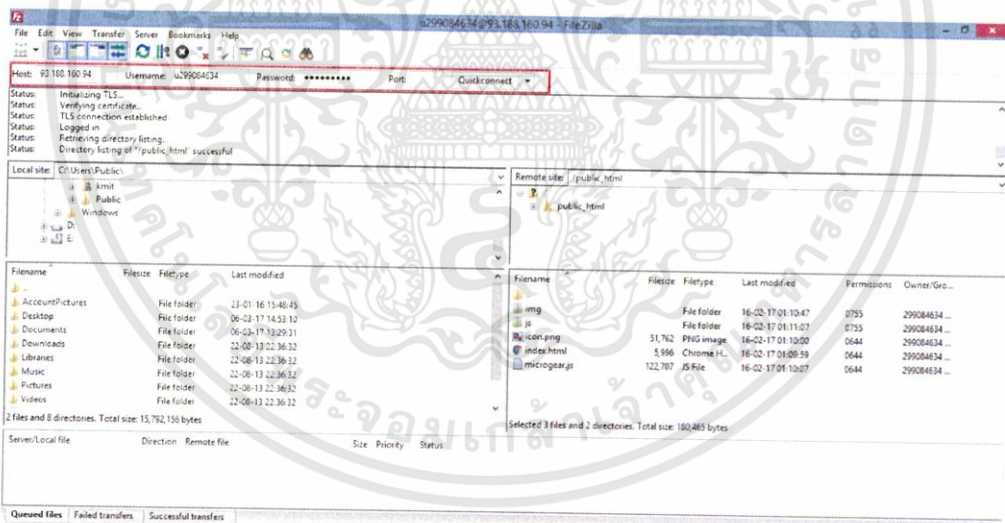
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) บันทึก IP ADDRESS และ PASSWORD ไว้ให้เชื่อมต่อกับโปรแกรม FileZilla Client ดังรูปที่ 3.31



รูปที่ 3.31 หน้าต่างของรหัส IP ADDRESS และ PASSWORD

3) เปิดโปรแกรม FileZilla Client และกรอก host, user และ password จากนั้น copy โฟลเดอร์เว็บไซต์ที่สร้างขึ้นมาสู่โฟลเดอร์ Public_html ดังรูปที่ 3.32

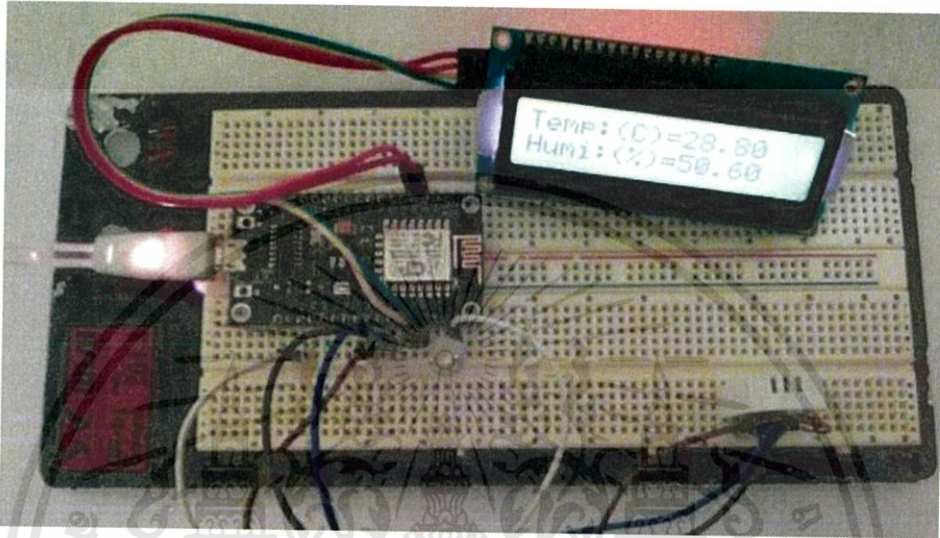


รูปที่ 3.32 หน้าต่างของโปรแกรม FileZilla Client

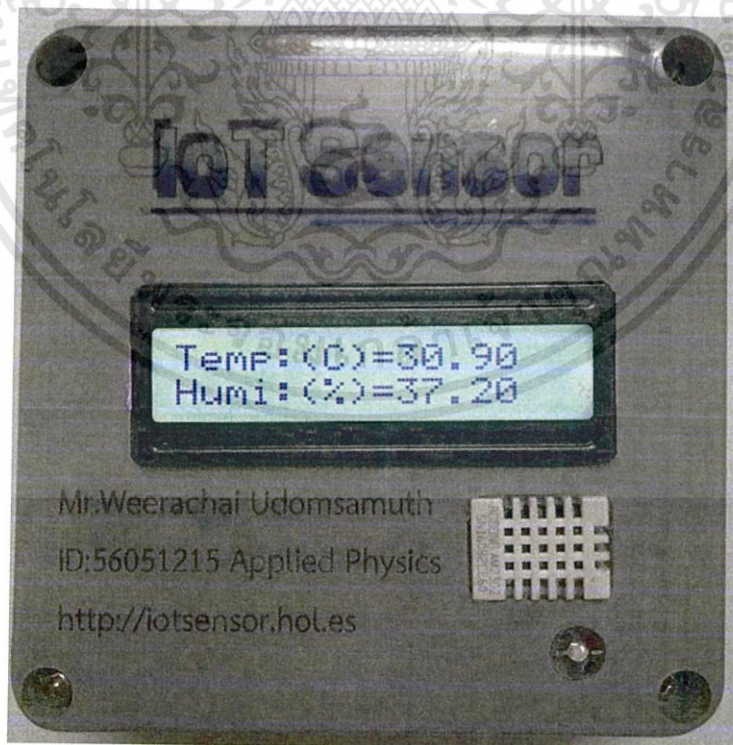
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การทดสอบการตอบสนองระหว่าง NodeMCU กับเว็บไซต์ และการบันทึกข้อมูล

เมื่อทำการต่อวงจรและสร้างเว็บไซต์ดังรูปที่ 3.33, 3.34 และ 3.35 เสร็จตามขั้นตอน จากนั้นผู้วิจัยจึงทดสอบการทำงานโดยการเปรียบเทียบค่าอุณหภูมิและความชื้นจากจอ LCD กับบนเว็บไซต์ ดังรูปที่ 3.36 ซึ่งเป็นการส่งข้อมูลจาก NodeMCU ไปยังเว็บไซต์ และควบคุมสวิตช์เปิดปิดบนเว็บไซต์ ดังรูปที่ 3.37 และ 3.38 ซึ่งเป็นการส่งข้อมูลจากเว็บไซต์ไปยัง NodeMCU เพื่อทำการเปิดปิดไฟ LED

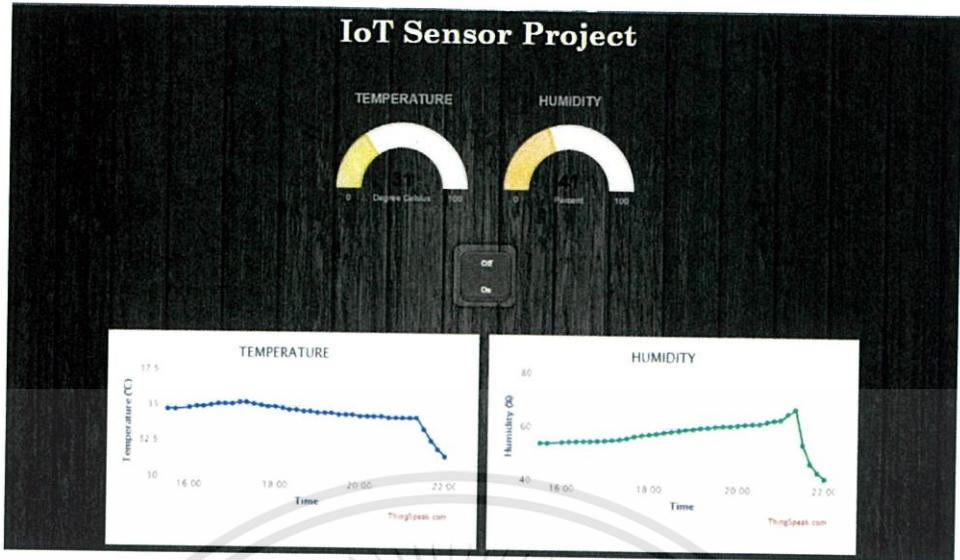


รูปที่ 3.33 วงจรการทำงาน



รูปที่ 3.34 อุปกรณ์ IoT Sensor

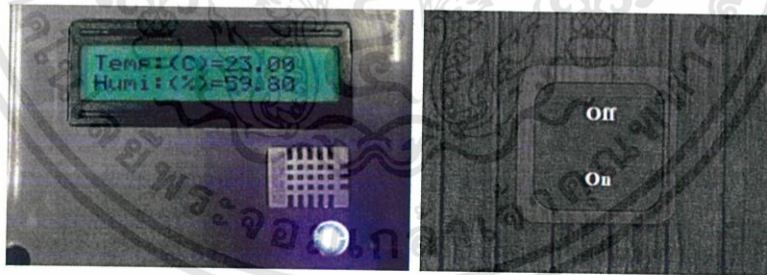
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.35 เว็บไซต์ที่สร้างขึ้น



รูปที่ 3.36 เปรียบเทียบค่าอุณหภูมิและความชื้นจากจอ LCD กับบนเว็บไซต์



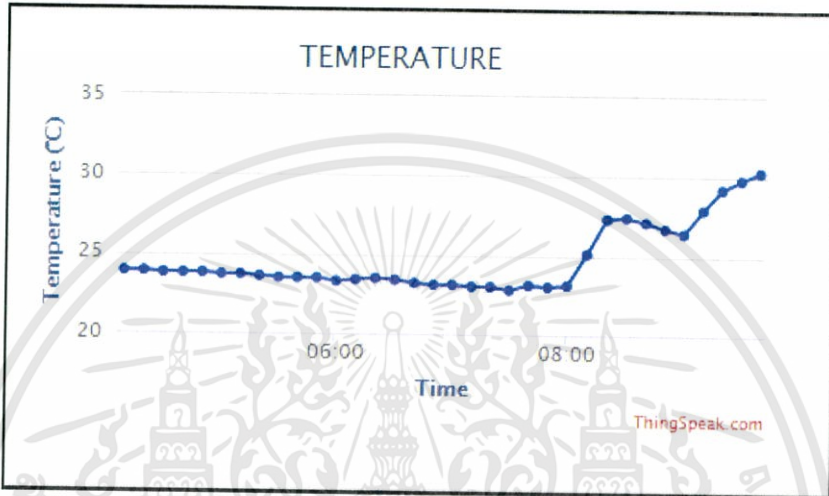
รูปที่ 3.37 เมื่อสั่งการให้สวิตช์เปิด (On) จะทำให้ไฟ LED ติด



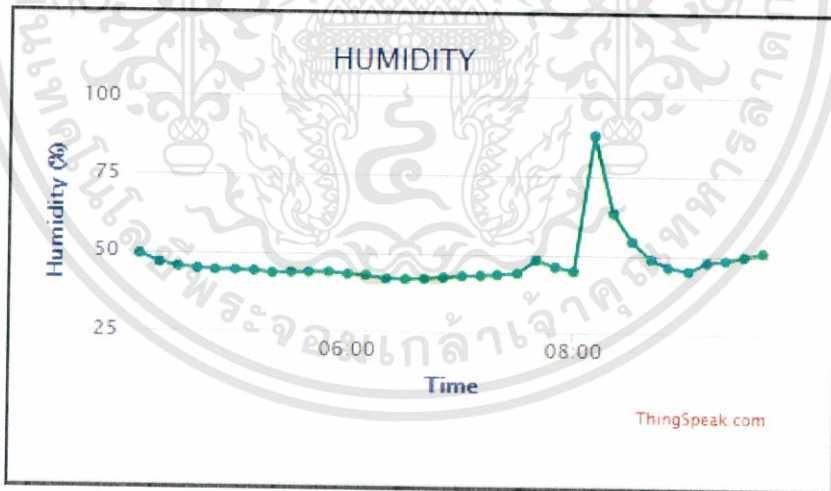
รูปที่ 3.38 เมื่อสั่งการให้สวิตช์ปิด (Off) จะทำให้ไฟ LED ดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบันทึกข้อมูลของอุณหภูมิและความชื้นจะถูกบันทึกในรูปแบบของกราฟความสัมพันธ์ระหว่างอุณหภูมิกับเวลาดังรูปที่ 3.39 และกราฟความสัมพันธ์ระหว่างความชื้นกับเวลาดังรูปที่ 3.40 โดยการบันทึกข้อมูลของตัวเซนเซอร์จะถูกส่งข้อมูลไปยังเว็บ thingspeak เพื่อบันทึกข้อมูลของอุณหภูมิและความชื้นซึ่งมีอัตราการบันทึกอยู่ที่ 1 ข้อมูล ต่อ 10 นาที จะสังเกตได้จากกราฟข้อมูลในรูปที่ 4.6 และ 4.7 ซึ่งสามารถบันทึกข้อมูลได้ประมาณ 35 – 40 ค่าต่อกราฟ คือสามารถดูข้อมูลได้ย้อนหลังประมาณ 5 ชั่วโมงและข้อมูลเก่าจะถูกแทนที่ด้วยข้อมูลใหม่ทันที



รูปที่ 3.39 กราฟความสัมพันธ์ระหว่าง Temperature (°C) กับ Time (min)



รูปที่ 3.40 กราฟความสัมพันธ์ระหว่าง Humidity (%) กับ Time (min)

บทที่ 4

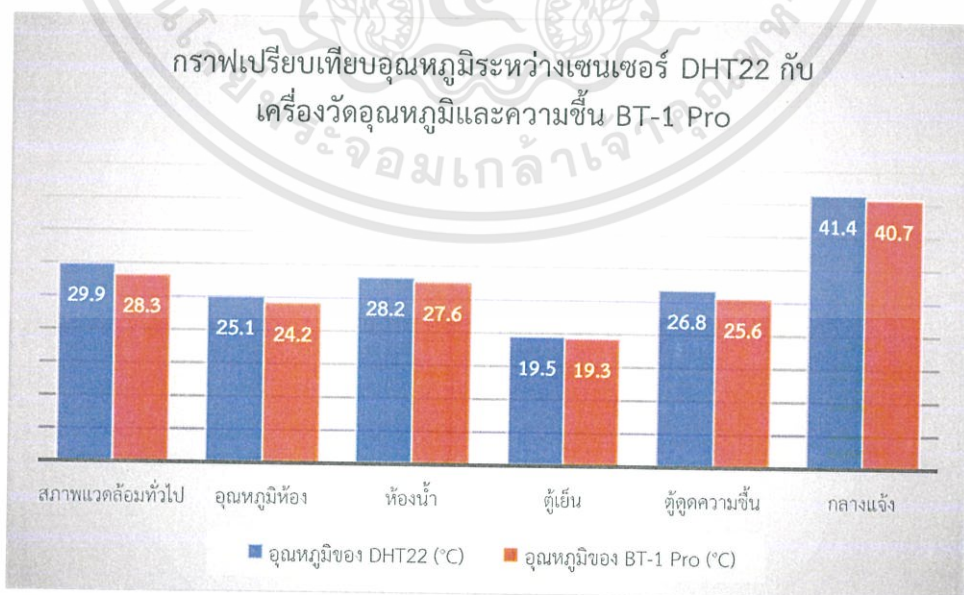
ผลการวิจัยและการอภิปรายผล

ในบทนี้จะกล่าวถึงการศึกษาสมบัติการทำงานของเซนเซอร์ DHT22 ภายใต้สภาวะแวดล้อมที่แตกต่างกัน ผู้วิจัยได้เลือกทดสอบวัดค่าอุณหภูมิและความชื้น แล้วนำมาเปรียบเทียบกับเครื่องวัดอุณหภูมิและความชื้น (Big LCD-display Thermometer and Hygrometer, BT-1 Pro) เพื่อตรวจสอบความแม่นยำของเซนเซอร์

4.1 คุณสมบัติของเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro กับเซนเซอร์วัดอุณหภูมิและความชื้น DHT22

ตารางที่ 4.1 การเปรียบเทียบค่าของอุณหภูมิระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro

สภาพแวดล้อมต่างๆ	อุณหภูมิของ DHT22 (°C)	อุณหภูมิของ BT-1 Pro (°C)	% คลาดเคลื่อน
สภาพแวดล้อมทั่วไป	29.9	28.3	5.65
ที่อุณหภูมิห้อง	25.1	24.2	3.72
ห้องน้ำ	28.2	27.6	2.17
ตู้เย็น	19.5	19.3	1.04
ตู้ดูดความชื้น	26.8	25.6	4.69
กลางแจ้ง	41.4	40.7	1.72

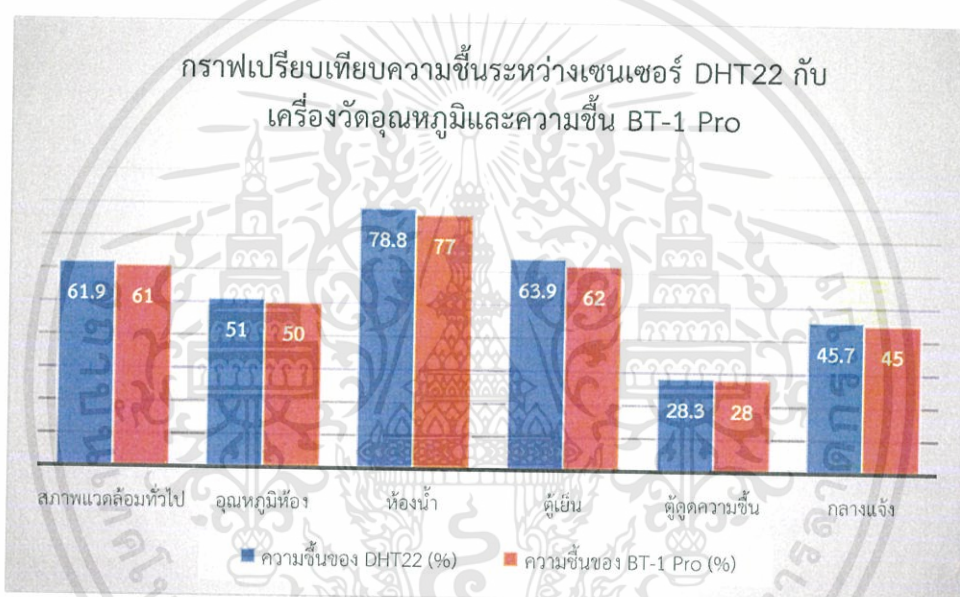


รูปที่ 4.1 กราฟเปรียบเทียบค่าของอุณหภูมิระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การเปรียบเทียบค่าของความชื้นระหว่างเซนเซอร์ DHT22 กับ เครื่องวัดอุณหภูมิและความชื้น BT-1 Pro

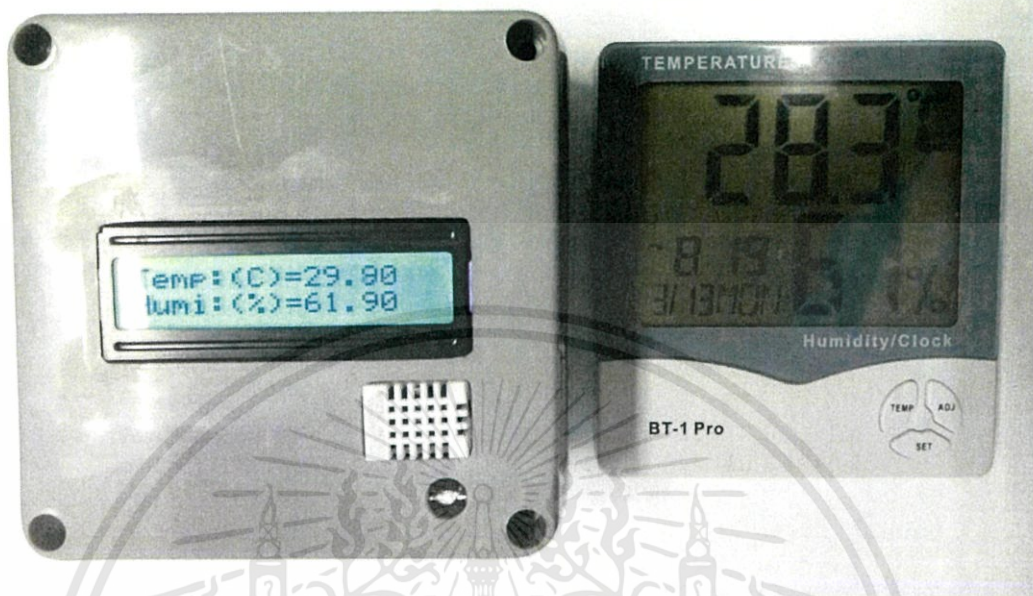
สภาพแวดล้อมต่างๆ	ความชื้นของ DHT22 (%)	ความชื้นของ BT-1 Pro (%)	% คลาดเคลื่อน
สภาพแวดล้อมทั่วไป	61.9	61	1.47
ที่อุณหภูมิห้อง	51.0	50	2.00
ห้องน้ำ	78.8	77	2.34
ตู้เย็น	63.9	62	3.06
ตู้ดูดความชื้น	28.3	28	1.07
กลางแจ้ง	45.7	45	1.56



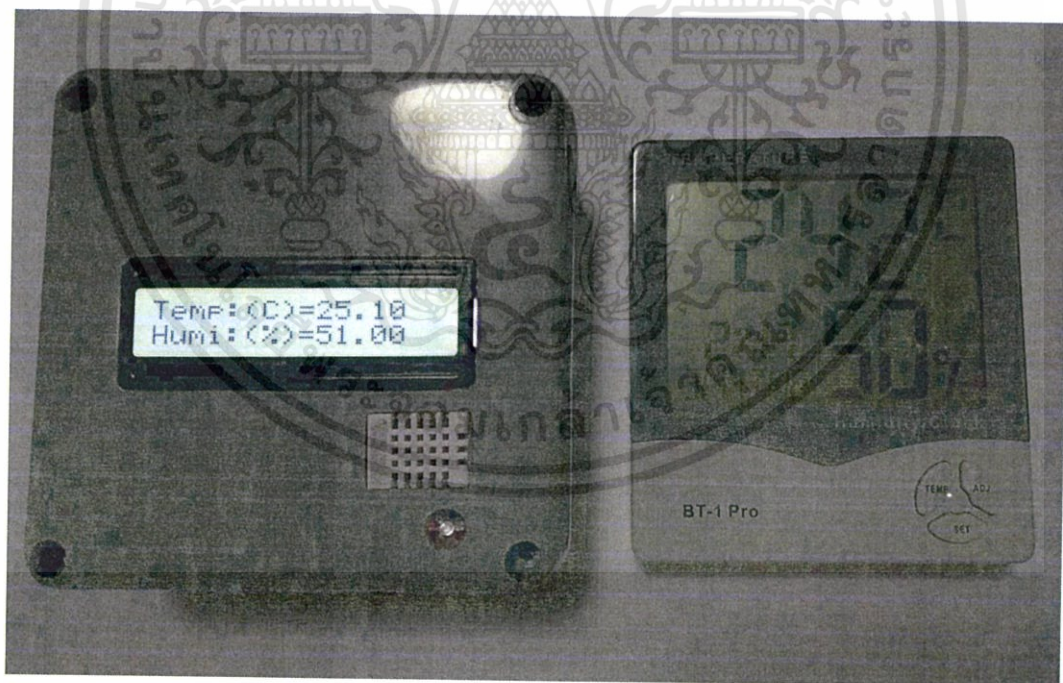
รูปที่ 4.2 กราฟเปรียบเทียบค่าของความชื้นระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro

เมื่อเปรียบเทียบข้อมูลในตารางที่ 4.1 การเปรียบเทียบอุณหภูมิระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro และตารางที่ 4.2 การเปรียบเทียบความชื้นระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro แล้วนำมาเขียนกราฟเปรียบเทียบค่าของอุณหภูมิระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro ดังรูปที่ 4.1 และกราฟเปรียบเทียบค่าของความชื้นระหว่างเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro ดังรูปที่ 4.2 พบว่าเซนเซอร์ DHT22 มีประสิทธิภาพตรงตามตารางแสดงคุณสมบัติ และข้อมูลที่วัดได้มีค่าใกล้เคียงกับค่าที่วัดจากเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro ที่นำมาใช้เปรียบเทียบเป็นมาตรฐาน อีกทั้งเซนเซอร์ยังสามารถวัดอุณหภูมิและความชื้นในสภาพแวดล้อมที่แตกต่างกันได้อีกด้วย

การทดสอบสมบัติการทำงานของเซนเซอร์ DHT22 กับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro ภายใต้สภาวะแวดล้อมต่างๆ ในสถานที่จริง สามารถแสดงดังรูปที่ 4.3 ถึง 4.8

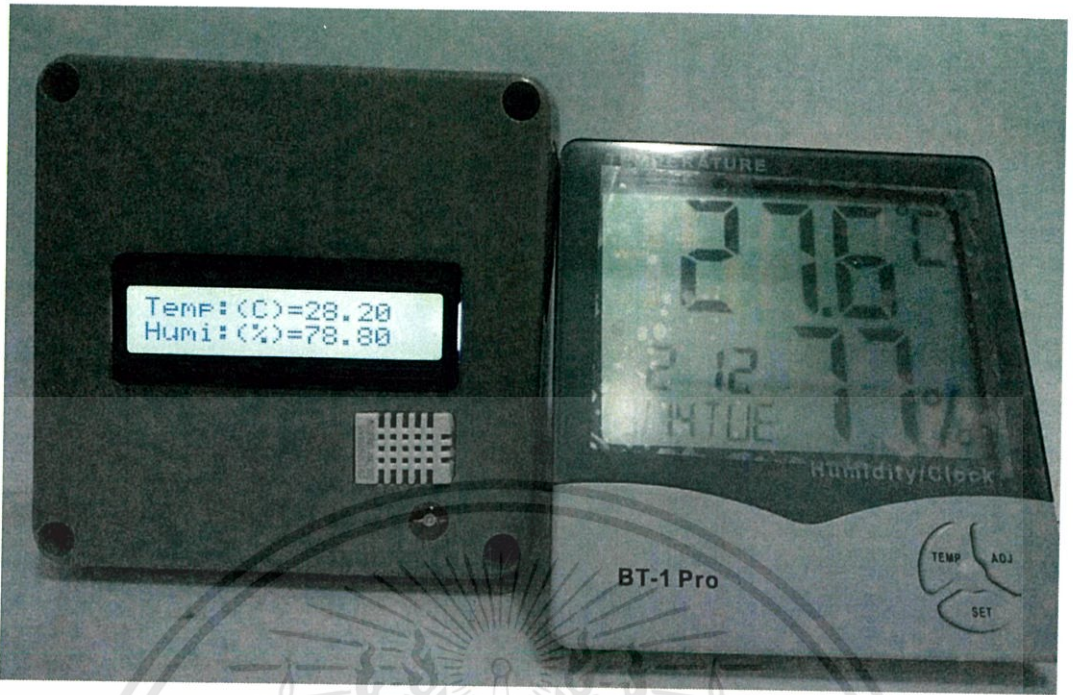


รูปที่ 4.3 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่สภาพแวดล้อมทั่วไป

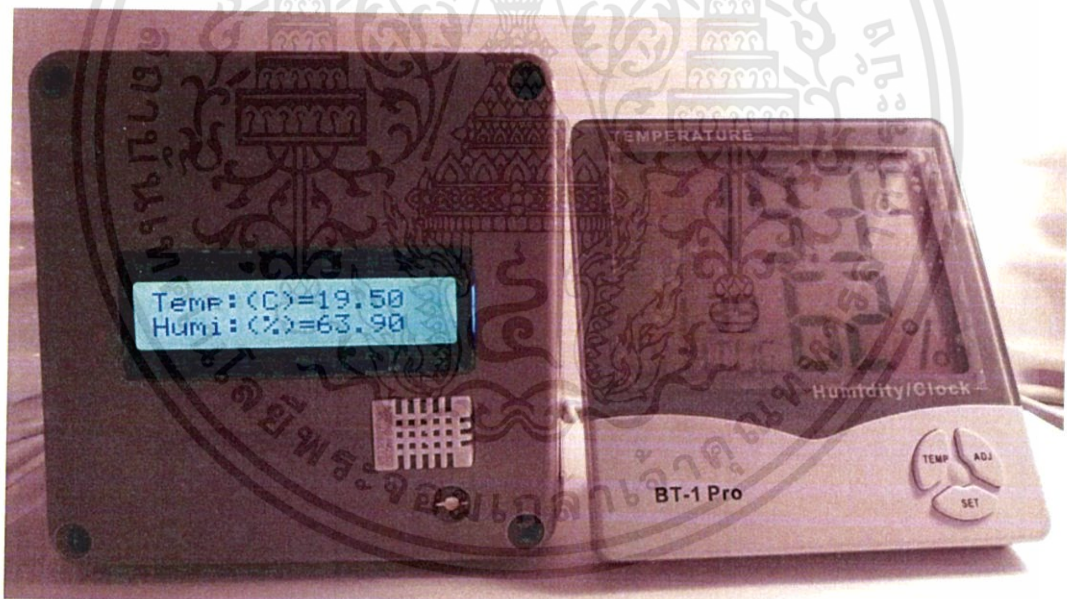


รูปที่ 4.4 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่อุณหภูมิห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่ห้องน้ำ

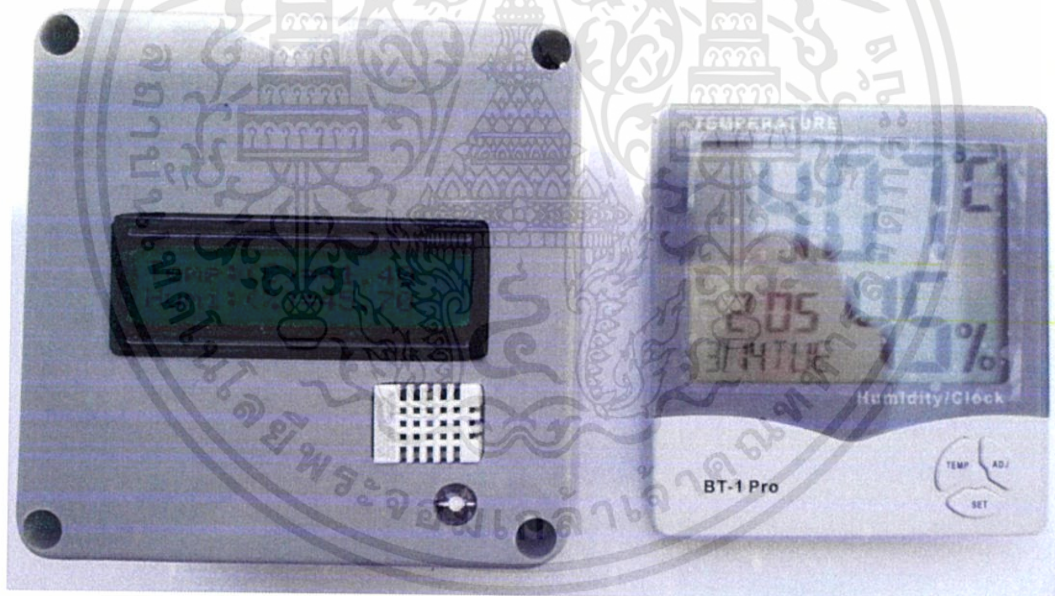


รูปที่ 4.6 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่ตู้เย็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่ตู้ดูดความชื้น



รูปที่ 4.8 ค่าของอุณหภูมิและความชื้นเมื่อวัดที่กลางแจ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในงานวิจัยนี้เป็นการทดลองเกี่ยวกับการตอบสนองระหว่าง NodeMCU กับเว็บไซต์ โดยใช้เซนเซอร์ DHT22 ในการส่งค่าของอุณหภูมิและความชื้นไปยังเว็บไซต์ <http://iotsensor.hol.es> และศึกษาการตอบสนองกลับของเว็บไซต์โดยการสั่งเปิดปิดไฟ LED จากการทดลองพบว่า เว็บไซต์มีการรับ-ส่งข้อมูลอุณหภูมิและความชื้นซึ่งจะแสดงผลผ่านเว็บไซต์เป็นค่าตัวเลขและกราฟของอุณหภูมิและความชื้น ณ เวลานั้นๆ อีกทั้งยังมีการบันทึกข้อมูลเพื่อที่จะสามารถดูข้อมูลย้อนหลังได้ โดยการบันทึกข้อมูลจะอยู่ที่ประมาณ 35-40 ค่าต่อกราฟและสามารถบันทึกข้อมูลย้อนหลังได้ 5 ชั่วโมง เนื่องจากกราฟมีพื้นที่จำกัด ดังนั้นหากต้องการดูข้อมูลย้อนหลังมากกว่า 5 ชั่วโมง จำเป็นต้องเก็บค่าของข้อมูลที่จากเดิม 1 ข้อมูล ต่อ 10 นาที เป็น 1 ข้อมูล ต่อ 20 นาทีหรือมากกว่านั้น ส่งผลให้ข้อมูลที่ได้มีความละเอียดไม่มากนัก และในการทดลองการทำงานของเซนเซอร์ DHT22 เมื่อเปรียบเทียบกับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro พบว่าค่าอุณหภูมิและความชื้นที่ได้จากเซนเซอร์ DHT22 จะมีค่ามากกว่าค่าอุณหภูมิและความชื้นที่ได้จากเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro อยู่เล็กน้อย เนื่องจาก IoT Sensor ได้ติดตั้งเซนเซอร์ไว้ด้านนอกของอุปกรณ์เพื่อประสิทธิภาพการทำงานสูงสุด เมื่อมีการสัมผัสกับสภาพแวดล้อมนั้นๆ ทำให้เซนเซอร์ DHT22 มีการตอบสนองต่อสิ่งเร้าเร็วกว่าเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro และเมื่อคำนวณหาเปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ยของอุณหภูมิและความชื้นจะมีค่าเท่ากับ 3.16 % และ 1.92 % ตามลำดับ ดังนั้นเซนเซอร์ DHT22 มีประสิทธิภาพในการทำงานเทียบเท่ากับเครื่องวัดอุณหภูมิและความชื้น BT-1 Pro

5.2 ข้อเสนอแนะ

- 1) ในการเลือกใช้หลอด LED ควรเลือกใช้หลอดที่ใช้กระแสต่ำๆ เพื่อการประหยัดพลังงานของแบตเตอรี่
- 2) เนื่องจากอุปกรณ์ IoT Sensor มีระยะเวลาการทำงานประมาณ 12 ชั่วโมง ซึ่งถ้าต้องการจะเพิ่มระยะเวลาการทำงานให้มากขึ้นจึงจำเป็นต้องเพิ่มแบตเตอรี่เข้าไปอีกก้อน
- 3) ไม่มีระบบ Config เนื่องจากการเชื่อมต่อ WIFI จะต้องเข้าไปแก้ไขโค้ดในส่วนของ SSID และ Password ซึ่งมีความยากลำบากมาก จึงควรปรับปรุงในส่วนของโค้ด Arduino IDE ให้มีระบบ Config WIFI
- 4) ถ้าไม่มีอินเทอร์เน็ตอุปกรณ์ IoT Sensor จะไม่สามารถทำงานได้เลย

เอกสารอ้างอิง

- [1] กรมชลประทาน กระทรวงเกษตรและสหกรณ์. 2559. “โครงการศึกษาด้านสารสนเทศเรื่อง Internet of Things (IoT).” ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร.
- [2] พิษิต วิจิตรบุญรักษ์. “HTML ภาษาเขียนเว็บ.” มหาวิทยาลัยกรุงเทพ.
- [3] ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ. คู่มือการใช้งาน NETPIE. เข้าถึงได้จาก : <https://netpie.io/tutorials>.
- [4] ศรีเดช ปัญจพันธ์. NodeMCU Board. เข้าถึงได้จาก : <http://bigtui.exteen.com/20151210/1-board-nodemcu-v-2-esp8266-wi-fi-module>.
- [5] สุขุม แป้นศรี. “พื้นฐานการใช้โปรแกรม Adobe dreamweaver CS6.” วิทยาลัยอาชีวศึกษานครศรีธรรมราช.
- [6] สุรเชษฐ์ กานต์ประชา. “โครงการวิจัย เรื่อง ระบบติดตามและประมาณเวลาการเดินทางไฟฟ้าด้วยสมาร์ตโฟนผ่านเครือข่าย 3G” คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร.
- [7] สุวิทย์ ภูมิฤทธิกุล และปานวิทย์ ฐะนุติ. 2559. “Internet of Thing เพื่อการเฝ้าระวังและเตือนภัยต่อสุขภาพของมนุษย์ และการวิเคราะห์ข้อมูลที่ได้โดยใช้โปรแกรม Hadoop.” วิทยานิพนธ์วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีปทุมวัน และเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- [8] ห้องปฏิบัติการระบบสมองกลฝังตัว ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ (KMUTNB). DHT22 / AM2302 Temperature & Relative Humidity Sensor. เข้าถึงได้จาก : http://cpre.kmutnb.ac.th/esl/learning/index.php?article=dht22_am2302.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโค้ด Arduino IDE

```

#include "DHT.h"

#include <ESP8266WiFi.h>

#include <MicroGear.h>

#include <EEPROM.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);

#define WIFI_SSID      "Emursion"
#define WIFI_PASSWORD  "0859754447"

#define APPID          "BellSmartHome"
#define KEY             "7fAwmBE8e3aGh1"
#define SECRET         "w1TpC2Zz2L2b1TnV9njgeP9Ga"
#define ALIAS          "pieplug"

const char* host = "api.thingspeak.com";
const char* api  = "YFCXDY0GD5YZ6OJP";

#define RELAYPIN D0
#define LEDPIN LED_BUILTIN
#define EEPROM_STATE_ADDRESS 128
#define DHTPIN D4
#define DHTTYPE DHT22

WiFiClient client;
AuthClient *authclient;
MicroGear microgear(client);

```

```

DHT dht(DHTPIN, DHTTYPE);

int lastPublish = 0;

float humid = 0;

float temp = 0;

String data;

char state = 0;

char stateOutdated = 0;

void sendState() {
  if (state==0)
    microgear.chat("webapp","0");
  else
    microgear.chat("webapp","1");
  Serial.println("send state..");
  stateOutdated = 0;
}

void updateIO()
{
  if (state >= 1) {
    digitalWrite(RELAYPIN, LOW);
    digitalWrite(LEDPIN, HIGH);

    EEPROM.write(EEPROM_STATE_ADDRESS, 1);
    EEPROM.commit();
  }
  else {
    state = 0;
    digitalWrite(RELAYPIN, HIGH);
    digitalWrite(LEDPIN, LOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EEPROM.write(EEPROM_STATE_ADDRESS, 0);
EEPROM.commit();
}
}

```

```

void onMsgHandler(char *topic, uint8_t* msg, unsigned int msglen) {
    char m = *(char *)msg;

```

```

    Serial.print("Incoming message --> ");
    msg[msglen] = '\0';
    Serial.println((char *)msg);

    if (m == '0' || m == '1') {
        state = m=='0'?0:1;
        EEPROM.write(EEPROM_STATE_ADDRESS, state);
        EEPROM.commit();
        updateIO();
    }
    if (m == '0' || m == '1' || m == '?') stateOutdated = 1;
}

```

```

void onConnected(char *attribute, uint8_t* msg, unsigned int msglen) {
    Serial.println("Connected to NETPIE...");
    microgear.setAlias(ALIAS);
    stateOutdated = 1;
}

```

```

void initWiFi(){
    if (WiFi.begin(WIFI_SSID, WIFI_PASSWORD)) {
        while (WiFi.status() != WL_CONNECTED) {
            delay(10);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Serial.print(".");
    }
}

Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

```

```

void setup() {
    lcd.begin();
    Serial.begin(115200);
    Serial.println("Starting...");
    pinMode(RELAYPIN, OUTPUT);
    pinMode(LED_BUILTIN, OUTPUT);
    dht.begin();
    EEPROM.begin(512);
    state = EEPROM.read(EEPROM_STATE_ADDRESS) == 1 ? 1 : 0;
    updateIO();

    initWiFi();
    microgear.on(MESSAGE, onMsgHandler);
    microgear.on(CONNECTED, onConnected);
    microgear.init(KEY, SECRET);
    microgear.connect(APPID);
}

void loop() {

```

```

    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.println("Temp:(C)=");
    lcd.setCursor(9,0);

```

```

lcd.print(temp);
lcd.setCursor(0,1);
lcd.println("Humi:(%)=");
lcd.setCursor(9,1);
lcd.print(humid);

if(WiFi.status()!=WL_CONNECTED){
  WiFi.disconnect();
  initWiFi();
} else {
  if (microgear.connected()) {
    microgear.loop();
    if (millis() - lastPublish > 1000) {
      float h = dht.readHumidity();
      float t = dht.readTemperature();
      if (!isnan(h)) humid = h;
      if (!isnan(t)) temp = t;

      data = String(temp)+String(",")+String(humid);
      Serial.print("Chat dht data --> ");
      Serial.println(data);
      lastPublish = millis();
      microgear.chat("webapp",data);

      String data = "{\"humid\": " + (String)humid + ",\"temp\": " + (String)temp + "}";
      Serial.print("connecting to ");

      Serial.println(host);

      WiFiClient client;
      const int httpPort = 80;
      if (!client.connect(host, httpPort)) {
        Serial.println("connection failed");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return;
}
url += api;
url += "&field1=";
url += t;
url += "&field2=";
url += h;
Serial.print("Requesting URL: ");
Serial.println(url);
// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
delay(10);
// Read all the lines of the reply from server and print them to Serial
while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
}
}
if (stateOutdated) sendState();
}else{
    Serial.println("connection lost, reconnect...");
    microgear.connect(APPID);
    delay(10);
}
}
}
}

```

ส่วนของโค้ด HTML

```
<script src="https://cdn.netpie.io/microgear.js"></script>
```

```
<script src="js/justgage.js"></script>
```

```
<script src="js/raphael-2.1.4.min.js"></script>
```

```
<style>
```

```
  body {
    text-align: center;
```

```
}
```

```
#temp, #humid {
  width:200px; height:190px;
  display: inline-block;
```

```
}
#name {
  font-size: 20px;
  color: blue;
```

```
}
#esp8266 {
  display: inline;
```

```
}
```

```
#nectec {
  padding: 10px;
  width:127px;
  height:30px;
```

```
}
```

```
/* SWITCH */
```

```
.cube-switch {
```

```
  border-radius:10px;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

border:1px solid rgba(0,0,0,0.4);
box-shadow: 0 0 8px rgba(0,0,0,0.6), inset 0 100px 50px rgba(255,255,255,0.1);
/* Prevents clics on the back */
cursor:default;
display: block;
height: 75px;
position: relative;
overflow:hidden;
/* Prevents clics on the back */
pointer-events:none;
width: 75px;
white-space: nowrap;
background:#fffff;
margin: 15px;
}

/* The switch */
.cube-switch .switch {
border:1px solid rgba(0,0,0,0.6);
border-radius:0.7em;
box-shadow:
inset 0 1px 0 rgba(255,255,255,0.3),
inset 0 -7px 0 rgba(0,0,0,0.2),
inset 0 50px 10px rgba(0,0,0,0.2),
0 1px 0 rgba(255,255,255,0.2);
display:block;
width: 60px;
height: 60px;
margin-left:-30px;
margin-top:-30px;
position:absolute;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

top: 50%;
left: 50%;
width: 60px;

background:#666;
transition: all 0.2s ease-out;

```

```

/* Allows click */
cursor:pointer;
pointer-events:auto;
}

```

```

/* SWITCH Active State */
.cube-switch.active {
  /*background:#222;
  box-shadow:
    0 0 5px rgba(0,0,0,0.5),
    inset 0 50px 50px rgba(55,55,55,0.1);*/
}

```

```

.cube-switch.active .switch {
  background:#333;
  box-shadow:
    inset 0 6px 0 rgba(255,255,255,0.1),
    inset 0 7px 0 rgba(0,0,0,0.2),
    inset 0 -50px 10px rgba(0,0,0,0.1),
    0 1px 0 rgba(205,205,205,0.1);
}

```

```

.cube-switch.active:after,
.cube-switch.active:before {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

background:#333;
box-shadow:
  0 1px 0 rgba(255,255,255,0.1),
  inset 1px 2px 1px rgba(0,0,0,0.5),
  inset 3px 6px 2px rgba(200,200,200,0.1),
  inset -1px -2px 1px rgba(0,0,0,0.3);
}

```

```

.cube-switch.active .switch:after,
.cube-switch.active .switch:before {
  background:#222;
  border:none;
  margin-top:0;
  height:1px;
}

```

```

.cube-switch .switch-state {
  display: block;
  position: absolute;
  left: 40%;
  color: #ffffff;

  font-size: .5em;
  text-align: center;
}

```

```

/* SWITCH On State */

```

```

.cube-switch .switch-state.on {
  bottom: 15%;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* SWITCH Off State */
.cube-switch .switch-state.off {
  top: 15%;
}

</style>

<script>
var temp,humid;
window.onload = function(){
  temp = new JustGage({
    id: "temp",
    value: 0,
    min: 0,
    max: 100,
    title: "TEMPERATURE",
    label: "Degree Celsius",
    showMinMax: false
  });

  humid = new JustGage({
    id: "humid",
    value: 0,
    min: 0,
    max: 100,
    title: "HUMIDITY",
    label: "Percent "
  });
};
</script>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<script>

const APPID = "BellSmartHome";
const KEY = "7fAwmBE8e3aGlh1";
const SECRET = "w1TpC2Zz2L2b1TnV9njgeP9Ga";
const ALIAS = "webapp";

var microgear = Microgear.create({
    key: KEY,
    secret: SECRET,
    alias : ALIAS
});

var x=0,timestamp=0;

function switchPress(){
    if(document.getElementById("cube-switch").className == "cube-switch
active"){
        microgear.chat("pieplug","0");
    }else if(document.getElementById("cube-switch").className == "cube-
switch"){
        microgear.chat("pieplug","1");
    }
}

microgear.on('message',function(topic,msg) {
    var split_msg = msg.split(",");

    if(split_msg.length==2){
        temp.refresh(split_msg[0]);
        humid.refresh(split_msg[1]);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(msg=="1"){
    document.getElementById("cube-switch").className = "cube-
switch active";
}else if(msg=="0"){
    document.getElementById("cube-switch").className = "cube-
switch";
}
});

```

```

microgear.on('connected', function() {
microgear.setAlias(ALIAS);
document.getElementById("data").innerHTML = '<p></p>';
microgear.chat("pieplug","?");
});
microgear.connect(APPID);
</script>

```

```
<div id="data"></div>
```

```

<h3><font face = "Century Schoolbook"><h1 span style="color:
#ffffff;">IoT Sensor Project</h1></span></font></h3>

```

```
<div id="esp8266">
```

```
<div id="temp"></div>
```

```
<div id="humid"></div>
```

```
<center>
```

```
<div href="" class="cube-switch" id="cube-switch" onclick="switchPress()">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<span class="switch">
    <h3><span class="switch-state off">Off</span>
    <span class="switch-state on">On</span></h3>
</span>
</div>
</center>
</div>
</br><iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/215146/charts/1?bgcolor=%23FFFFFF&color=%2
30066FF&days=1&dynamic=true&results=1500&timescale=10&title=TEMPERATURE&ty
pe=line&xaxis=Time&yaxis=Temperature+%28%C2%B0C%29"></iframe>&nbsp;&nbsp; 
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/215146/charts/2?bgcolor=%23ffffff&color=%2300
CC00&days=1&dynamic=true&results=1500&timescale=10&title=HUMIDITY&type=line
&xaxis=Time&yaxis=Humidity+%28%25%29"></iframe></br>
<h3 style="text-align: center;">
    <font face = "Century Schoolbook">
        <h3><span style="color: #ffffff;">Mr.Weerachai Udomsamuth</br>
        <span style="color: #ffffff;">Associate Professor. Wichit Sirichote</br></h3>
    </font>
</h3>
<center></center>
<font face = "Century Schoolbook"><h4><span style="color: #ffffff;">Department of
Applied Physics, Faculty of Science, King Mongkut's Institute of Technology
Ladkrabang. (KMITL)</br></h4></font>
</div>
<style type="text/css">
html {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

background-image: url("img/blackwood1.jpg");
background-repeat: no-repeat;
background-position: center center;
background-attachment: fixed;
-o-background-size: 100% 100%, auto;
-moz-background-size: 100% 100%, auto;
-webkit-background-size: 100% 100%, auto;
background-size: 100% 100%, auto;
}
</style>

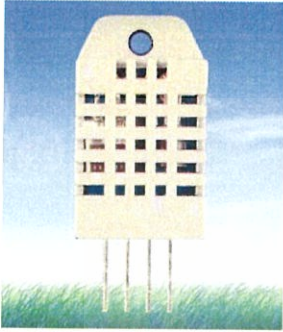
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Digital relative humidity & temperature sensor RHT03

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:

RHT03 output calibrated digital signal. It applies exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements are 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.

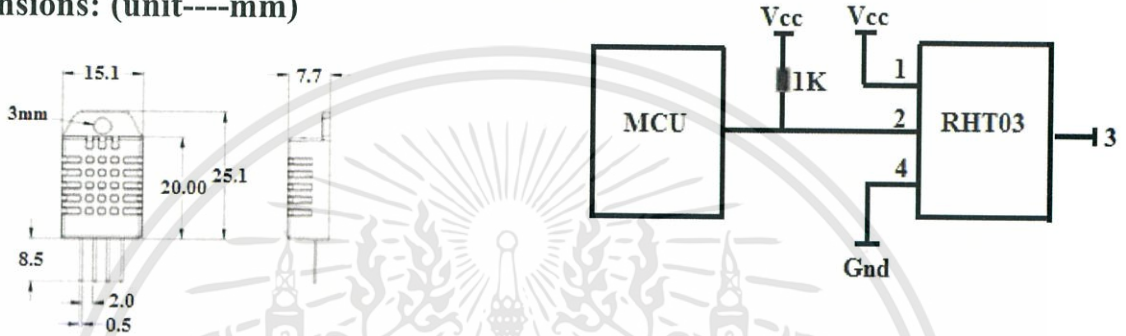
Small size & low consumption & long transmission distance(100m) enable RHT03 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	RHT03	
Power supply	3.3-6V DC	
Output signal	digital signal via MaxDetect 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

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	NULL
4	GND

5. Electrical connection diagram:

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-6V 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

MaxDetect 1-wire bus is used for communication between MCU and RHT03. (MaxDetect 1-wire bus is specially designed by MaxDetect Technology Co., Ltd. , it's different from Maxim/Dallas 1-wire bus, so it's incompatible with Dallas 1-wire bus.)

Illustration of MaxDetect 1-wire bus:

Data is comprised of integral and decimal part, the following is the formula for data.

DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum

If the data transmission is right, check-sum should be:

Check sum=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data

Example: MCU has received 40 bits data from RHT03 as

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

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

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

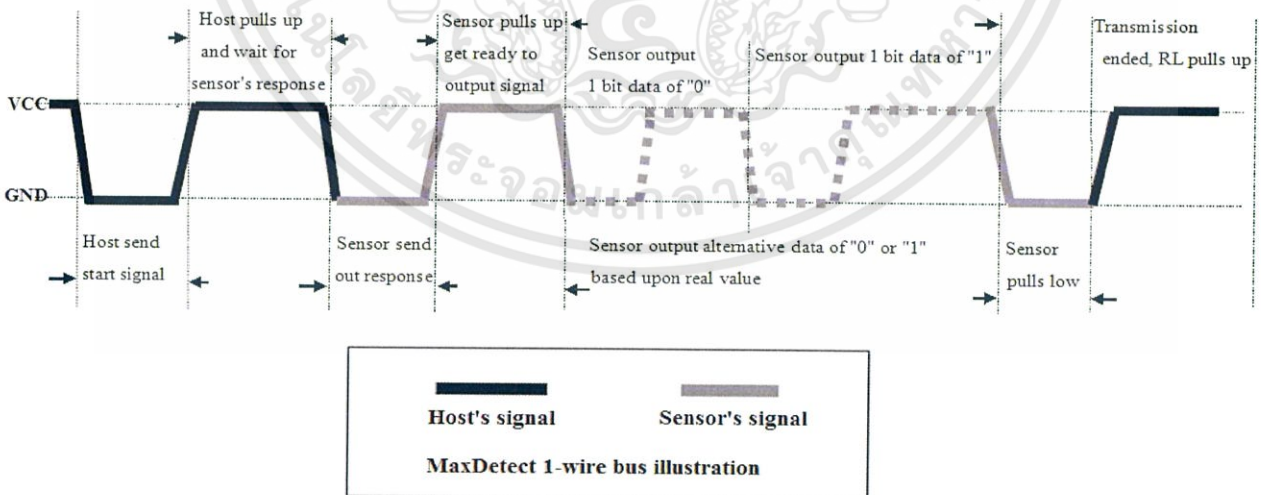
T=(0000 0001 0101 1111)/10=35.1□

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

Example: 1000 0000 0110 0101, T= minus 10.1□
 16 bits T data

When MCU send start signal, RHT03 change from standby-status to running-status. When MCU finishes sending the start signal, RHT03 will send response signal of 40-bit data that reflect the relative humidity and temperature to MCU. Without start signal from MCU, RHT03 will not give response signal to MCU. One start signal for one response data from RHT03 that reflect the relative humidity and temperature. RHT03 will change to standby status when data collecting finished if it don't receive start signal from MCU again.

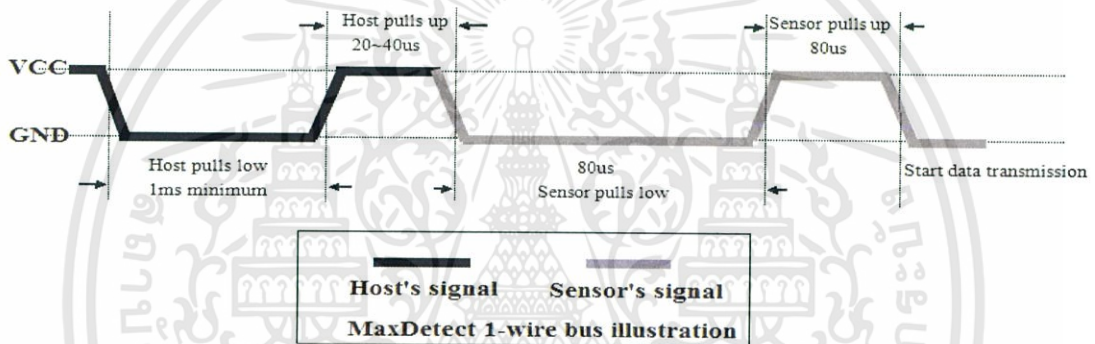
See below figure for overall communication process, the interval of whole process must beyond 2 seconds.



1) Step 1: MCU send out start signal to RHT03 and RHT03 send response signal to MCU

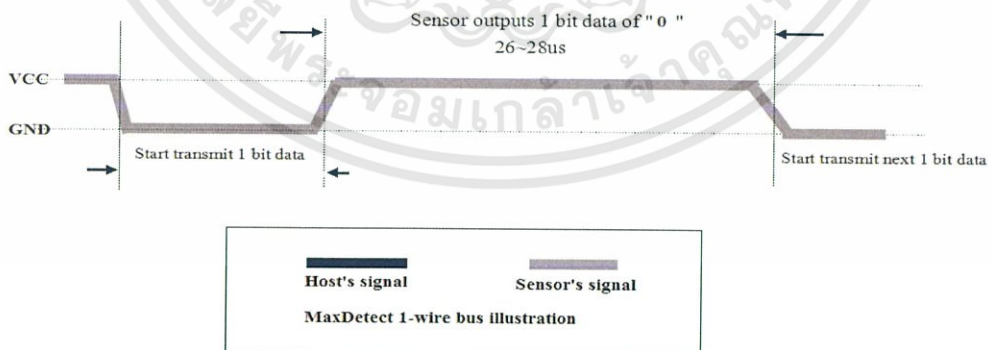
Data-bus's free status is high voltage level. When communication between MCU and RHT03 begins, MCU will pull low data-bus and this process must beyond at least 1~10ms to ensure RHT03 could detect MCU's signal, then MCU will pull up and wait 20-40us for RHT03's response.

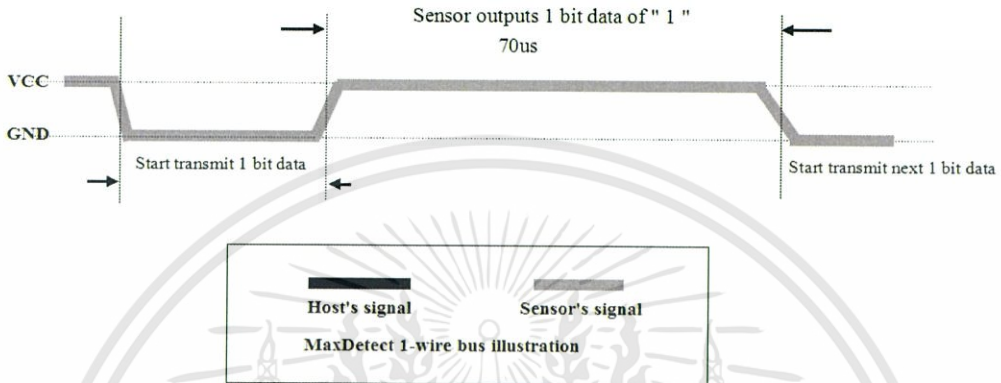
When RHT03 detect the start signal, RHT03 will pull low the bus 80us as response signal, then RHT03 pulls up 80us for preparation to send data. See below figure:



2). Step 2: RHT03 send data to MCU

When RHT03 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 RHT03 is always high-voltage-level, it means RHT03 is not working properly, please check the electrical connection status.

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 RHT03 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 RHT03's sensitive-elements and debase RHT03's sensitivity.

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

Step one: Keep the RHT03 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the RHT03 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.

RHT03 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

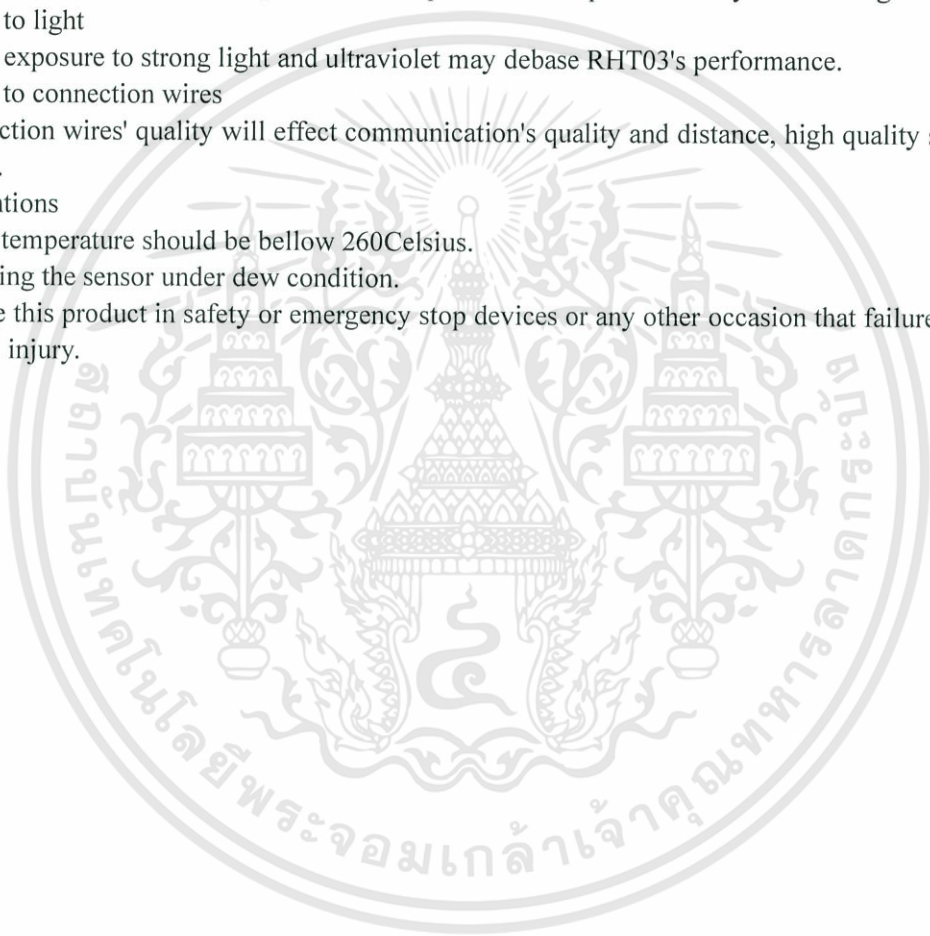
Long time exposure to strong light and ultraviolet may debase RHT03's performance.

(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 RHT03 may cause personal injury.





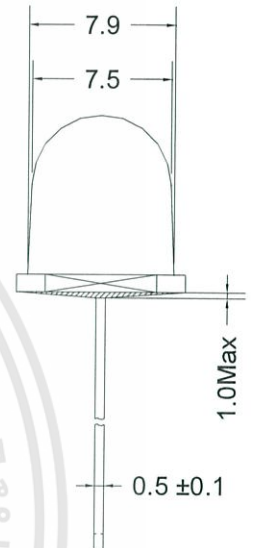
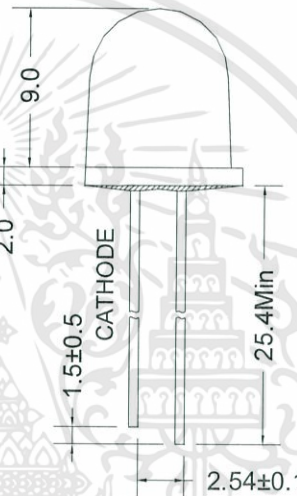
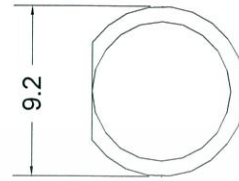
ATTENTION
OBSERVE PRECAUTIONS
FOR HANDLING
ELECTROSTATIC
DISCHARGE
SENSITIVE
DEVICES

➤ **Features/特征:**

- Single color/单色
- High bright output/高亮度输出
- Low power consumption/低功耗
- High reliability and long life/
可靠性高、寿命长

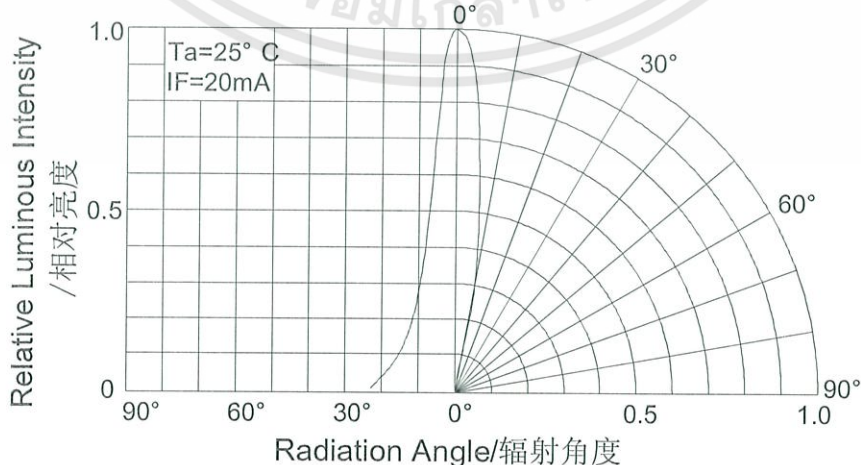
➤ **Descriptions/描述:**

- Dice material/芯片材质: InGaN
- Emitting Color/发光颜色:
Super Bright Blue/ 高亮度蓝色
- Device Outline/产品外形:
φ 8mm Round Type/ 8mm 圆形
- Lens Type 胶体颜色:
Water Clear/ 无色透明



1. All dimensions are millimeters/单位: mm.
2. Tolerance is +/-0.20mm unless otherwise noted/
没有标注的公差均为±0.20mm.

➤ **Directivity/指向特性:**





➤ **Absolute maximum ratings/极限参数 (Ta = 25°C)**

Parameter 参数	Symbol 符号	Test Condition 测试条件	Values 数值		Unit 单位
			Min.	Max.	
Reverse Voltage 反向电压	V _R	I _R = 30 μ A	5	--	V
Forward Current 正向工作电流	I _F	----	----	25	mA
Power Dissipation 损耗功率	P _d	----	----	90	mW
Pulse Current 正向峰值电流	I _{peak}	Duty=0.1mS, 1kHz	----	100	mA
Operating Temperature 工作温度范围	T _{opr}	----	-20	+85	°C
Storage Temperature 储存温度范围	T _{str}	----	-25	+100	°C

➤ **Electrical and optical characteristics/光电参数 (Ta = 25°C)**

Parameter 参数	Symbol 符号	Test Condition 测试条件	Values 数值			Unit 单位
			Min.	Typ.	Max.	
Forward Voltage 正向电压	V _F	I _F =20mA	----	3.2	3.6	V
Reverse Current 反向电流	I _R	V _R =5V	----	----	30	μ A
Dominate Wavelength 主波长	λ _d	I _F =20mA	----	470	----	nm
Peak Wavelength 峰值波长	λ _p	I _F =20mA	----	468	----	nm
Spectral Line half-width 半波长宽度	Δ λ	I _F =20mA	----	20	----	nm
Luminous Intensity 发光强度	I _v	I _F =20mA	----	5000	----	mcd
Viewing Angle 指向角度	2 θ 1/2	I _F =20mA	----	15	----	deg.



➤ **Typical electrical/optical characteristic curves/光电特性曲线:**

Fig.1 正向电流 Vs. 正向电压

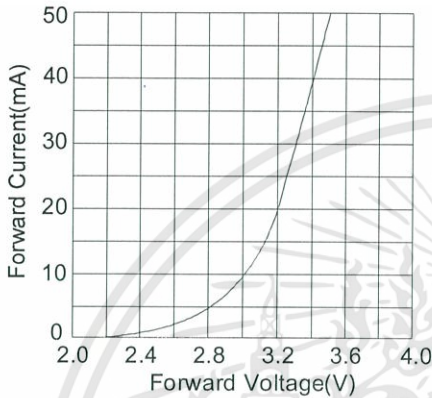


Fig.2 相对亮度 Vs. 正向电流

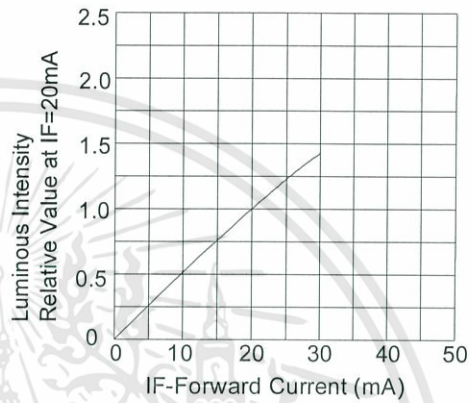


Fig.3 正向电流 Vs. 环境温度

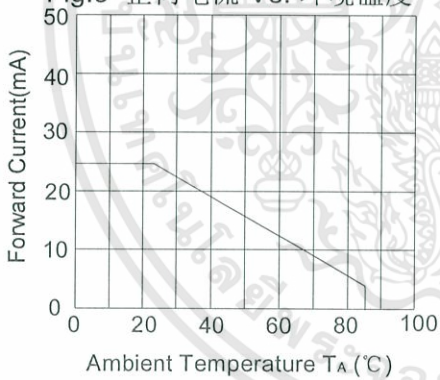


Fig.4 相对亮度 Vs. 环境温度

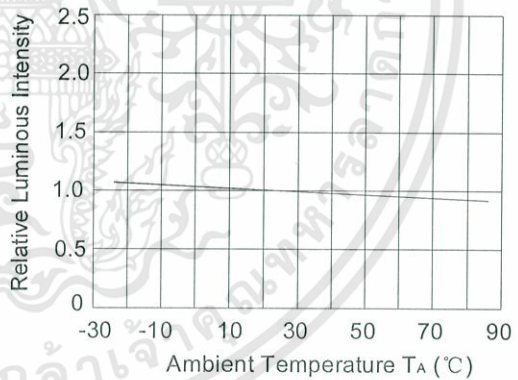
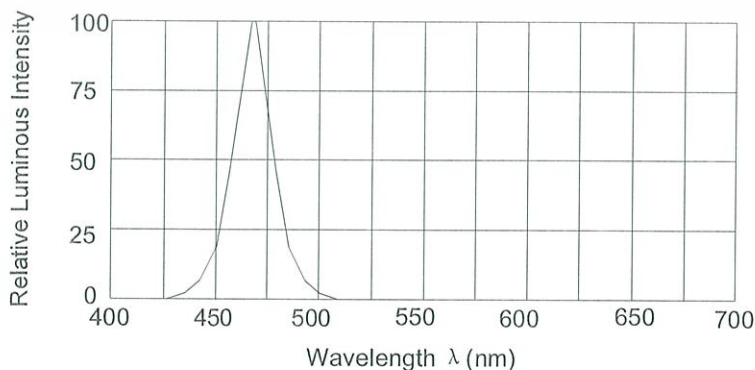


Fig.5 相对亮度 Vs. 波长





➤ **Lead Forming/成形**

1. Any lead forming or bending must be done before soldering.

支架成形必须在焊接前完成。

2. When forming leads, there must be a minimum of 2mm clearance between the base of the LED lens and the lead bend.

必需离胶体 2 毫米才能折弯支架。

3. Avoid bending the leads at the same point more than once.

避免在管脚同一位置两次或多次弯。

4. During assembly onto PCB, the lead pitch of the LED must match the pitch of the mounting holes on the PCB during component placement.

支架成形需保证引脚和间距与线路板上一致。

➤ **Soldering Condition/焊接条件**

Careful attention should be paid during soldering. When soldering, leave more than 2mm from solder joint to case, and soldering beyond the base of the tie bar is recommended.

焊接时请特别注意，焊接点要离LED封装体底部2mm以上。

Avoiding applying any stress to the lead frame while the LEDs are at high temperature particularly when soldering.

在 LED 处于高温，特别是在焊接时，请避免对支架施压。



Recommended soldering conditions/推荐焊接条件:

Hand Soldering 手工焊接		DIP Soldering 浸焊	
Temp.at tip of iron 电烙铁温度	300℃Max.(30WMax.) 最高温度300℃（功率不超过30瓦）	Preheat temp. 预热温度	100℃ Max. (60 sec Max.) 最高温度100℃（不超过60秒）
Soldering time 焊接时间	3 sec Max. 时间不超过3秒	Bath temp. 浸焊温度	260℃ Max 最高260℃.
Distance 焊接位置	2mm Min.(From solder joint to case) 大于2毫米（从焊点到胶体）	Bath time. 浸焊时间	3 sec Max. 不超过3秒
		Distance 浸焊位置	2mm Min 大于2毫米.

➤ **Cleaning/清洗**

1. Do not clean LEDs with water, Alcohol are recommended solvents for cleaning. When using other solvents, it should be confirmed beforehand whether the solvents will dissolve the resin or not.

不要用水清洗，推荐使用酒精清洗，当使用其他溶剂清洗时应事先确认该溶剂是否溶解树脂。

2. LEDs may be damaged by ultrasonic-washed. Before cleaning, a pre-test should be done to confirm whether any damage to the LEDs will occur.

LED有可能在超声波清洗过程中被破坏。为了保证安全，在清洗前，请先确认。

➤ **Storage/保存**

1. Environmental temperature: -40℃---100℃, Recommended: -20℃---50℃

环境温度：-40℃---100℃，推荐使用-20℃---50℃；

2. Environmental humidity: 30%---70%, Recommended: 40%---60%

环境湿度：30%---70%，推荐使用40%---60%；



➤ **Static Electricity/静电**

1. Static Electricity or power surge will damage the LED.

It is recommended that a wrist band or an anti-electrostatic glove be used when handling the LEDs.

静电或激增电压将损害 LED。

建议使用发光二极管时佩戴手腕带或防静电手套。

2. All production machinery and test instruments must be electrically grounded.

所有的生产机器和检测仪器，必须接地。

3. Maintain a humidity level of 50% or higher in production areas.

生产现场湿度必须保持在50%或者更高。

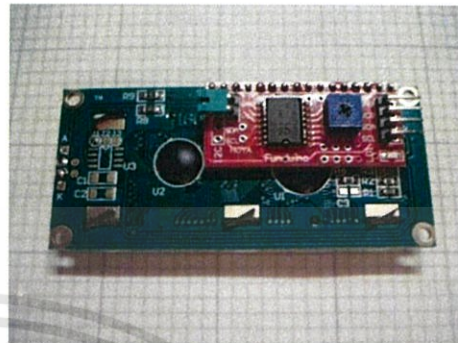
4. Use anti-static packaging for transport and storage.

使用防静电包装，运输和储存。



Datasheet

I2C 1602 Serial LCD Module



Product features:

The I2C 1602 LCD module is a 2 line by 16 character display interfaced to an I2C daughter board. The I2C interface only requires 2 data connections, +5 VDC and GND to operate

For in depth information on I2C interface and history, visit: <http://www.wikipedia/wiki/i2c>

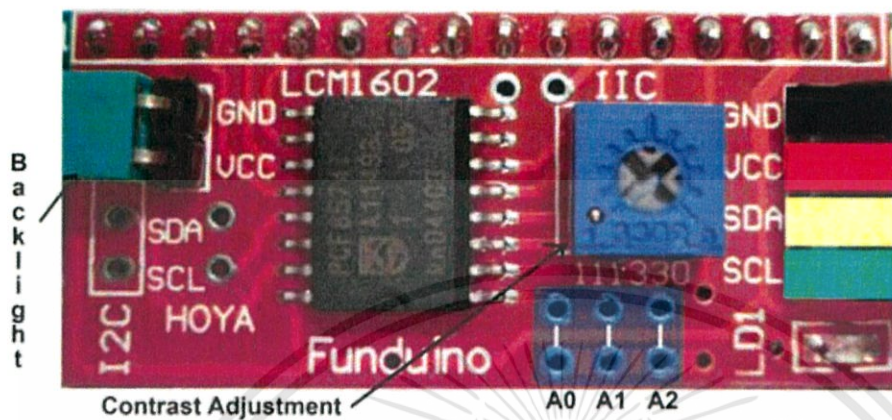
Specifications:

I2C Address Range	2 lines by 16 character 0x20 to 0x27 (Default=0x27, addressable)
Operating Voltage	5 Vdc
Backlight	White
Contrast	Adjustable by potentiometer on I2c interface
Size	80mm x 36mm x 20 mm
Viewable area	66mm x 16mm

Power:

The device is powered by a single 5Vdc connection.

Pinout Diagram:



Pin/Control Descriptions:

Pin #	Name	Type	Description
1	GND	Power	Supply & Logic ground
2	VCC	Power	Digital I/O 0 or RX (serial receive)
3	SDA	I/O	Serial Data line
4	SCL	CLK	Serial Clock line
A0	A0	Jumper	Optional address selection A0 - see below
A1	A1	Jumper	Optional address selection A1 - see below
A2	A2	Jumper	Optional address selection A2 - see below
Backlight		Jumper	Jumpered - enable backlight, Open - disable backlight
Contrast		Pot	Adjust for best viewing

Addressing:

A0	A1	A2	Address
Open	Open	Open	0x27
Jumper	Open	Open	0x26
Open	Jumper	Open	0x25
Jumper	Jumper	Open	0x24
Open	Open	Jumper	0x23
Jumper	Open	Jumper	0x22
Open	Jumper	Jumper	0x21
Jumper	Jumper	Jumper	0x20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Software:

Download the required LCD Arduino™ library for this device from:

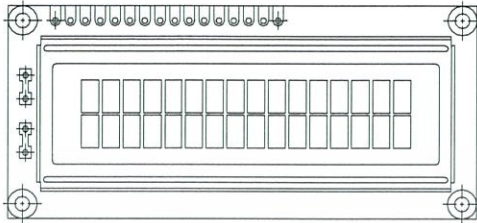
<http://www.circuitattic.com/downloads/category/3-sample-code.html?download=9%3Aanother-i2c-library-easier-to-use>

Replace current liquid crystal library found in the Arduino library directory with the above
(Note: If you use the examples included with the library, be sure to change address to 0x27)

Simple example using library above.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a
//chars and 2 line display
void setup()
{
    lcd.init(); // initialize the lcd
    lcd.backlight();
    lcd.clear();
    delay(100);
    for(int i = 0; i< 3; i++)
    {
        lcd.backlight();
        delay(250);
        lcd.noBacklight();
        delay(250);
    }
    lcd.backlight();
}
void loop()
{
    int x=0;
    lcd.clear();
    lcd.setCursor(2,0); //Start at character 0 on line 0
    lcd.print("Hello World");
    lcd.setCursor(0,1); //Start at character 0 on line 1
    lcd.print(" opencircuit.nl");
    delay(3000); //Wait 3 seconds
    lcd.clear();
    lcd.setCursor(0,0); //Start at character 0 on line 0
    lcd.print("Cursor Blink");
    lcd.blink();
    delay(2000);
    lcd.setCursor(0,0);
    lcd.print("Cursor noBlink");
    lcd.noBlink();
    delay(2000);
}
```

16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA

ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING

ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	–	7.0	V
Input Voltage	VI	- 0.3	–	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS

ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT	
			MIN.	TYP.	MAX.		
Input Voltage	VDD	VDD = + 5V	4.7	5.0	5.3	V	
		VDD = + 3V	2.7	3.0	5.3	V	
Supply Current	IDD	VDD = 5V	–	1.2	3.0	mA	
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	- 20 °C	–	–	–	V	
		0°C	4.2	4.8	5.1		
		25°C	3.8	4.2	4.6		
		50°C	3.6	4.0	4.4		
LED Forward Voltage	VF	25°C	–	4.2	4.6	V	
LED Forward Current	IF	25°C	Array	–	130	260	mA
			Edge	–	20	40	
EL Power Supply Current	IEL	Vel = 110VAC:400Hz	–	–	5.0	mA	

DISPLAY CHARACTER ADDRESS CODE:

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F

LCD-016M002B



Vishay

16 x 2 Character LCD

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+ 3V or + 5V
3	Vo	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/W	H/L Read/Write Signal
6	E	H → L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/Vee	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (OV)

DIMENSIONS in millimeters

