



Android Application ควบคุมกังหันน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ

โดย

นางสาวเชวณันท์ จรรย์วรรณ รหัส B5601943

นางสาวบุษกร สมพมิตร รหัส B5612550

นางสาวสุธีรา แก้วพรรณา รหัส B5613939

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 527499 โครงการวิศวกรรมโทรคมนาคม
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ.2557

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ประจำภาคการศึกษาที่ 1 ปีการศึกษา 2559

Android Application ควบคุมกังหันน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ

คณะกรรมการสอบโครงการ


(ผู้ช่วยศาสตราจารย์ ดร.วิภาวี หัตถกรรม)
กรรมการ/อาจารย์ที่ปรึกษาโครงการ


(รองศาสตราจารย์ ดร.พีระพงษ์ อุฑารสกุล)
กรรมการ


(ผู้ช่วยศาสตราจารย์ ดร.ปิยาภรณ์ มีสวัสดิ์)
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับรายงานโครงการฉบับนี้ เป็นส่วนหนึ่งของ
การศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม วิชา 527499 โครงการวิศวกรรม
โทรคมนาคมประจำปีการศึกษา 2559

โครงการ	Android Application ควบคุมกังหันน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ		
จัดทำโดย	นางสาวเชาวนันท์ จรรย์วรรณ	รหัส B5601943	
	นางสาวบุษกร สมพมิตร	รหัส B5612550	
	นางสาวสุธีรา แก้วพรรณา	รหัส B5613939	
อาจารย์ที่ปรึกษา	ผศ. ดร. วิภาวี หัตถกรรม		
สาขาวิชา	สาขาวิชาวิศวกรรมโทรคมนาคม		
ภาคการศึกษา	ภาคการศึกษา 1/2559		

บทคัดย่อ

ในปัจจุบันการเพาะเลี้ยงสัตว์น้ำในประเทศไทยได้รับความนิยมเป็นอย่างมากและประชากรส่วนใหญ่ประกอบอาชีพประมงและเพาะเลี้ยงสัตว์น้ำในกระชังเพื่อเลี้ยงชีพ จึงถือว่ามีความสำคัญต่อเศรษฐกิจไทยเป็นอย่างมาก ในปัจจุบันเทคโนโลยีนั้นได้มีการพัฒนาอย่างต่อเนื่องและรวดเร็วอย่างก้าวกระโดด มีนวัตกรรมต่างๆเกิดขึ้นมากมาย อย่างเช่น Internet of Things ซึ่งเป็นการเชื่อมต่ออุปกรณ์ end devices ต่างๆที่ใช้วัดคุมเข้าในเครือข่าย Internet ซึ่งส่วนมากแล้วจะมี Smartphone เข้ามาช่วยในการรับข้อมูลและควบคุม end device ต่างๆเหล่านี้

คณะผู้จัดทำโครงการจึงได้คิดที่จะพัฒนาเทคโนโลยีสารสนเทศเพื่อการเพาะเลี้ยงสัตว์น้ำ โดยพัฒนา Application บนระบบปฏิบัติการ Android บน Smartphone โดยสามารถควบคุมกังหันน้ำผ่าน Cloud Platform เพื่อเพิ่มออกซิเจนในน้ำและแสดงผลการวัดคุณสมบัติของน้ำใน Application บน Smartphone

กิตติกรรมประกาศ

การศึกษาและจัดทำโครงงานวิศวกรรมโทรคมนาคมประเภทประดิษฐ์คิดค้นและการจำลองชิ้นงานระบบเซนเซอร์ไร้สายตรวจสอบสภาพแวดล้อมพื้นที่เพาะเลี้ยงสัตว์น้ำผ่าน Network Platform for Internet of Everything เพื่อศึกษาการทำงานของเซนเซอร์วัดความขุ่นในน้ำ เซนเซอร์วัดอุณหภูมิในน้ำ และเซนเซอร์วัดค่าความเป็นกรด – ด่าง (pH) ในน้ำ โครงงานเล่มนี้สำเร็จลงได้ด้วยความรู้ความกรุณาช่วยเหลือแนะนำอย่างดีจากอาจารย์ที่ปรึกษาโครงงาน ผศ.ดร.วิภาวี หัตถกรรม ที่ให้คำแนะนำปรึกษาที่เป็นประโยชน์เกี่ยวกับปัญหาที่เกิดขึ้นและให้ข้อเสนอแนะต่างๆ ในการทำโครงงานตลอดจนการแก้ไขปัญหาข้อผิดพลาดต่างๆ มาโดยตลอดจนทำให้โครงงานเล่มนี้สำเร็จสมบูรณ์ ผู้จัดทำโครงงานขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์ทุกท่านที่ได้ให้ความรู้ในสาขาวิชาต่างๆ ที่ได้รับมาเป็นประโยชน์ในการประยุกต์ใช้เพื่อจัดทำโครงงานในครั้งนี้จนสำเร็จ

ขอขอบคุณ รศ.ดร.สุรินทร์ บุญอนันตสิน สาร สำนักวิชาเทคโนโลยีการเกษตรที่อนุเคราะห์ข้อมูล และความรู้เกี่ยวกับการเพาะพันธุ์สัตว์น้ำ ตลอดจนให้ความสนใจในการนำเทคโนโลยีสารสนเทศไปประยุกต์ใช้ในการเพาะเลี้ยงสัตว์น้ำในฟาร์ม มทส.

ขอขอบคุณ อาจารย์สุนัย พลายมี ตำแหน่ง หัวหน้าวิจัยเพาะพันธุ์สัตว์น้ำ ที่อนุเคราะห์ข้อมูล และความรู้เกี่ยวกับการเพาะพันธุ์สัตว์น้ำ ตลอดจนให้ความสนใจในการนำเทคโนโลยีสารสนเทศไปประยุกต์ใช้ในการเพาะเลี้ยงสัตว์น้ำในฟาร์ม มทส.

ขอขอบคุณ นายจรัส สัตยมุข นักศึกษาปริญญาโท สาขาวิชาวิศวกรรมโทรคมนาคม ที่ให้คำปรึกษาแนะนำและการแก้ไขปัญหาในโครงงานครั้งนี้

ผู้จัดทำคาดหวังว่าโครงงานวิศวกรรมโทรคมนาคม เรื่อง Android Application ควบคุมกังหันน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ นี้จะเป็นประโยชน์แก่ผู้ที่สนใจทั่วไปได้ศึกษาและพัฒนาต่อไปได้ในอนาคต

คณะผู้จัดทำ

นางสาวเชาวนันท จรรย์วรรณ

นางสาวบุษกร สมพมิตร

นางสาวสุธีรา แก้วพรรณา

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูป	ง
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ตารางการดำเนินงาน	3
1.6 ผลที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	
2.1 บทนำ	5
2.2 คุณภาพน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ	5
2.2.1 อุณหภูมิ(Temperature) และช่วงแสง(Photoperiod)	5
2.2.2 ความขุ่นและความโปร่งใส(Turbidity and Transparency)	6
2.2.3 ค่าความเป็นกรด-ด่าง (pH)	7
2.2.4 ปริมาณออกซิเจนในน้ำ(Dissolved Oxygen)	7
2.3 Sensor วัดอุณหภูมิในน้ำ	8
2.4 Sensor วัดค่า pH ในน้ำ	10
2.4.1 การใช้งานของ Analog pH Meter Kit	10
2.5 Sensor วัดค่าความขุ่นในน้ำ	11
2.6 Relay Module	12
2.6.1 หลักการทำงานของ Relay Module	12
2.7 Amplifier Module	13
2.8 Wemos d1 WiFi ESP8266	14
2.8.1 การติดตั้ง Driver Arduino บน Windows	15

สารบัญ(ต่อ)

เรื่อง	หน้า
2.9 Internet of Things	20
2.10 NETPIE (Network Platform for Internet of Everything)	21
2.10.1 สัมผัส NETPIE และสร้าง Application	22
2.10.2 การติดตั้ง Microgear Library ของ Arduino	28
2.10.3 การสร้าง Datasource บน Freeboard	31
2.10.4 การเขียนโปรแกรมลงบอร์ด WeMos d1 wifi ESP8266	33
2.11 APPLICATION KKMC IoT ควบคุมอุปกรณ์ IoT	37
2.11.1 วิธีใช้งาน KKMC IoT	37
2.11.2 เซตอัพ APPLICATION เพื่อให้ใช้งานได้	40
2.11.3 การเพิ่ม Widget เพื่อแสดง ข้อมูลหรือสั่งให้เปิดปิด อุปกรณ์	42
บทที่ 3 การแสดงผลข้อมูลผ่าน Application	
3.1 กล่าวนำ	45
3.2 NETPIE	45
3.2.1 การใช้งาน NETPIE	46
3.2.2 Application, Gear , Key	47
3.2.3 การสร้าง Datasource บน Freeboard	50
3.3 Application KKMC IoT	55
3.3.1 การตั้งค่า Application เพื่อให้ใช้งานได้	55
3.3.2 การเพิ่ม Widget เพื่อแสดงข้อมูลและสั่งเปิด/ปิด อุปกรณ์	57
3.4 การเชื่อมต่อการสื่อสารให้อุปกรณ์ หรือ THING	60
3.5 การแสดงข้อมูล บน NETPIE Freeboard	61
3.6 การแสดงข้อมูล ผ่าน Android Application KKMC IoT	62
บทที่ 4 การทดสอบอุปกรณ์และแสดงผลข้อมูล	
4.1 การติดตั้งอุปกรณ์	63
4.2 วิเคราะห์ผลการทดลอง	69
4.3 สรุปผลการทดลอง	72

สารบัญ(ต่อ)

เรื่อง	หน้า
บทที่ 5 บทสรุปของโครงการ	
5.1 กล่าวสรุป	73
5.2 ปัญหาและการแก้ไข	74
5.3 ข้อเสนอแนะ	74
5.4 แนวทางในการพัฒนาต่อไป	74
ประวัติผู้เขียน	
เอกสารอ้างอิง	
ภาคผนวก	
ภาคผนวก ก Code การเชื่อมต่อกับระบบการให้บริการ Cloud Platform ของ NETPIE	77
ภาคผนวก ข ภาพรวม โครงการ	85



สารบัญรูป

รายการ	หน้า
รูปที่ 2.1 DS18B20 เซ็นเซอร์วัดอุณหภูมิแบบกันน้ำได้	8
รูปที่ 2.2 การต่อ Sensor วัดอุณหภูมิในน้ำ เข้ากับบอร์ด Arduino	9
รูปที่ 2.3 Analog pH Meter Kit	10
รูปที่ 2.4 Analog Turbidity Sensor	11
รูปที่ 2.5 Relay Module 4 Channel	12
รูปที่ 2.6 สัญลักษณ์ในวงจรไฟฟ้าของรีเลย์	12
รูปที่ 2.7 ADS1115 I2C ADC 4 Channel 16-Bit with Programmable Gain Amplifier Module	13
รูปที่ 2.8 WeMos d1 wifi esp8266	14
รูปที่ 2.9 การทำงานของพอร์ตต่างๆในบอร์ด WeMos d1 wifi esp8266	14
รูปที่ 2.10 คาวาน์โฮลด์ Arduino	15
รูปที่ 2.11 หน้าต่างหลังเปิดใช้งาน Arduino IDE 1.6.12	15
รูปที่ 2.12 เลือก Preferences	16
รูปที่ 2.13 แสดงหน้าต่างของ Preferences	16
รูปที่ 2.14 ใส่ URL ลงใน Addition Board Manager URLs	17
รูปที่ 2.15 เลือก Boards Manager	18
รูปที่ 2.16 ESP8266 เลือกเวอร์ชัน 2.2.0	18
รูปที่ 2.17 Download Boards ESP8266	19
รูปที่ 2.18 เลือกใช้งานงาน ESP8266 กับ Arduino IDE	20
รูปที่ 2.19 เข้าไปยังเว็บไซต์ netpic.io จากนั้นคลิกที่ SIGN UP FREE	22
รูปที่ 2.20 กรอกข้อมูลสำหรับการสมัครให้เรียบร้อย แล้วคลิก SIGN UP	22
รูปที่ 2.21 ใส่ข้อมูลเรียบร้อยและกด SIGN UP	23
รูปที่ 2.22 เข้าหน้า LOG IN เพื่อทำการเข้าระบบ	23
รูปที่ 2.23 LOG IN เข้ามาในระบบ	24
รูปที่ 2.24 หน้าหลักใช้งาน NETPIE	24
รูปที่ 2.25 เริ่มสร้าง Application โดยกดตรงปุ่ม [+]	25
รูปที่ 2.26 ใส่ชื่อ Application	25

สารบัญรูป

รายการ	หน้า
รูปที่ 2.27 Application ชื่อที่ตั้งเข้าไป	26
รูปที่ 2.28 การตั้งชื่อและเลือกชนิด Application Key	26
รูปที่ 2.29 Key สำหรับไว้ให้อุปกรณ์เอาไปใช้ในการสื่อสาร	27
รูปที่ 2.30 แสดง KEY และ SECRET เพื่อใช้ในการเชื่อมต่อกับ NETPIE	27
รูปที่ 2.31 เข้าไปแล้วกดที่ Download ZIP	28
รูปที่ 2.32 ทำการแตกไฟล์เอาไฟล์เคอร์ Microgear -ESP8266- Arduino -Master	28
รูปที่ 2.33 ทำการแก้ไขกันเพื่อให้สามารถใช้งานได้กับอุปกรณ์	29
รูปที่ 2.34 กรอกข้อมูล Key ที่สร้างไว้	29
รูปที่ 2.35 ชุดคำสั่งที่ควบคุมรีเลย์ให้เปิด,ปิด	30
รูปที่ 2.36 NETPIE Freeboard	31
รูปที่ 2.37 เพิ่ม Datasource	31
รูปที่ 2.38 เลือก NETPIE Microgear	32
รูปที่ 2.39 กำหนดค่าให้ Datasource	32
รูปที่ 2.40 Datasource ที่สร้างขึ้น	33
รูปที่ 2.41 โค้ดในโปรแกรม ArduinoIDE	34
รูปที่ 2.42 Upload โค้ดลง บอร์ด WeMos D1 wifi esp8266	34
รูปที่ 2.43 โค้ด NETPIE ESP8266 basic sample	35
รูปที่ 2.44 Verify ตรวจสอบความถูกต้องของโค้ด	36
รูปที่ 2.45 Upload โค้ดลง บอร์ด WeMos D1 wifi ESP8266	37
รูปที่ 2.46 แอปพลิเคชัน KKMC IoT	38
รูปที่ 2.47 ติดตั้งแอปพลิเคชัน KKMC IoT	38
รูปที่ 2.48 ยอมรับแอปพลิเคชัน KKMC IoT	39
รูปที่ 2.49 โหลดและติดตั้ง Application KKMC IoT เสร็จสมบูรณ์	39
รูปที่ 2.50 หน้าแรกของ Application KKMC IoT	40
รูปที่ 2.51 ตั้งค่า Application KKMC IoT	40
รูปที่ 2.52 แสดงการตั้งค่า Application KKMC IoT	40
รูปที่ 2.53 APP ID ในเว็บ NETPIE	41

สารบัญรูป

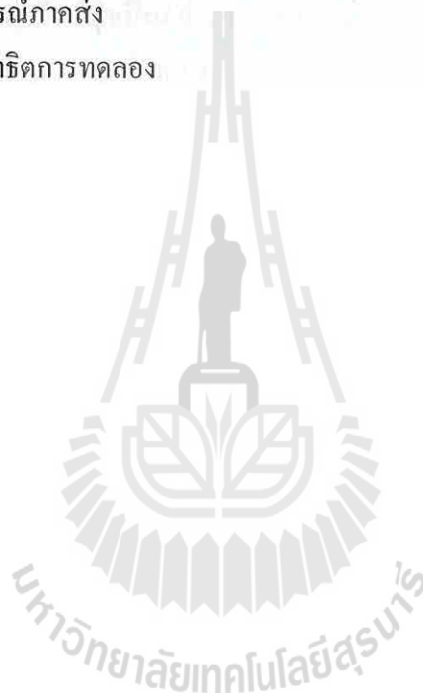
รายการ	หน้า
รูปที่ 2.54 REST KEY ในเว็บ NETPIE	41
รูปที่ 2.55 กรอกข้อมูล APP ID และ REST KEY เสร็จสมบูรณ์	42
รูปที่ 2.56 แสดงปุ่มการเพิ่ม Widget	42
รูปที่ 2.57 แสดงหน้าต่าง Add new Topic	43
รูปที่ 2.58 แสดงรูปแบบ Widget type	43
รูปที่ 2.58 กำหนด topic ในการรับข้อมูลอุณหภูมิ	44
รูปที่ 2.60 หน้าหลักของ Application แสดงค่าพารามิเตอร์	44
รูปที่ 3.1 สมัครสมาชิกเว็บ NETPIE	46
รูปที่ 3.2 กรอกข้อมูลสำหรับการสมัครให้เรียบร้อย แล้วคลิก SIGN UP	46
รูปที่ 3.3 LOGIN เข้ามาในระบบ	47
รูปที่ 3.4 หน้าหลักใช้งาน NETPIE	47
รูปที่ 3.5 สร้าง Application ID	48
รูปที่ 3.6 เริ่มสร้าง Application โดยกดตรงปุ่ม [+]	48
รูปที่ 3.7 Key สำหรับไว้ให้อุปกรณ์เอาไปใช้ในการสื่อสาร	49
รูปที่ 3.8 APP ID ในเว็บ NETPIE	49
รูปที่ 3.9 แสดง Key Secret และ REST API auth เพื่อใช้ในการเชื่อมต่อกับ NETPIE และ Application	49
รูปที่ 3.10 แสดง Key Secret และ REST API auth เพื่อใช้ในการเชื่อมต่อกับ NETPIE Freeboard	50
รูปที่ 3.11 หน้าหลักของ NETPIE เพื่อเข้าสู่ Freeboard Management	50
รูปที่ 3.12 สร้างชื่อ NETPIE Freeboard	50
รูปที่ 3.13 เพิ่ม Datasource	51
รูปที่ 3.14 Type บน Datasource	51
รูปที่ 3.15 กำหนดค่าให้ Datasource	52
รูปที่ 3.16 แสดง Datasource ที่สร้างขึ้น	52
รูปที่ 3.17 การเลือก Type แบบ Indicator Light แสดงสถานะของกังหันน้ำ	53
รูปที่ 3.18 ใส่ข้อมูล Widget ของการแสดงสถานะการเปิด/ปิด กังหันน้ำ	53

สารบัญรูป

รายการ	หน้า
รูปที่ 3.19 Widget Type แบบ Gauge	53
รูปที่ 3.20 เงื่อนไขการแสดงค่าเซนเซอร์อุณหภูมิในน้ำ บน NETPIE freeboard	54
รูปที่ 3.21 เงื่อนไขการแสดงค่าเซนเซอร์ความเป็นกรด-ด่าง(pH)ในน้ำ บน NETPIE freeboard	54
รูปที่ 3.22 เงื่อนไขการแสดงค่าเซนเซอร์ความชื้นของน้ำ บน NETPIE freeboard	55
รูปที่ 3.23 หน้าแรกของ Application KKMC IoT	55
รูปที่ 3.24 แสดงการตั้งค่า Application KKMC IoT	56
รูปที่ 3.25 กรอกข้อมูล App ID และ Rest Key เสร็จสมบูรณ์	56
รูปที่ 3.26 แสดงการเพิ่ม Widget	57
รูปที่ 3.27 หน้าต่าง Add new Topic	57
รูปที่ 3.28 รูปแบบ Widget Type 3 แบบ	58
รูปที่ 3.29 รูปแบบ Widget Type แบบ On/off Switch	58
รูปที่ 3.30 กำหนด Topic เพื่อรับค่าข้อมูลอุณหภูมิในน้ำ	59
รูปที่ 3.31 กำหนด Topic เพื่อรับค่าข้อมูลความชื้นในน้ำ	59
รูปที่ 3.32 หน้าหลักของ Application แสดงค่าพารามิเตอร์ในน้ำและควบคุมกังหันน้ำ	60
รูปที่ 4.1 ชุดอุปกรณ์	63
รูปที่ 4.2 ภาพอุปกรณ์ภายในของภาคส่ง	64
รูปที่ 4.3 ภาพอุปกรณ์ทั้งหมด	64
รูปที่ 4.4 หน้าแรกของ Application KKMC IoT	65
รูปที่ 4.5 การตั้งค่า Application KKMC IoT	65
รูปที่ 4.6 แสดงหน้าต่าง Add new Topic	66
รูปที่ 4.7 รูปแบบ Widget type	66
รูปที่ 4.8 เชื่อมต่อ Application KMMC IoT กับ NETPIE	67
รูปที่ 4.9 ค่าพารามิเตอร์ของเซนเซอร์และแสดงสถานะเปิดการทำงานของกังหันน้ำ	68
รูปที่ 4.10 ค่าพารามิเตอร์ของเซนเซอร์และแสดงสถานะเปิดการทำงานของกังหันน้ำ	68
รูปที่ 4.11 ค่าพารามิเตอร์ของเซนเซอร์และแสดงสถานะปิดการทำงานของกังหันน้ำ	69
รูปที่ 4.12 ค่าพารามิเตอร์ของเซนเซอร์และแสดงสถานะปิดการทำงานของกังหันน้ำ	69

สารบัญรูป

รายการ	หน้า
รูปที่ 4.12 สถานะการเชื่อมต่ออุปกรณ์เมื่อส่งขึ้น NETPIE สำเร็จ	72
ภาคผนวก	
รูปที่ 1 (ข) สำรวจพื้นที่ในการทำโครงการ	85
รูปที่ 2 (ข) สำรวจพื้นที่ในการทำโครงการ	85
รูปที่ 3 (ข) สำรวจพื้นที่ในการทำโครงการ	86
รูปที่ 4 (ข) อุปกรณ์ภาคส่ง	86
รูปที่ 5 (ข) ชุดสาริตการทดลอง	87



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันการเพาะเลี้ยงสัตว์น้ำในประเทศไทยได้รับความนิยมเป็นอย่างมากและประชากรส่วนใหญ่ประกอบอาชีพประมงและเพาะเลี้ยงสัตว์น้ำในกระชังเพื่อเลี้ยงชีพ ซึ่งถือว่ามีความสำคัญต่อเศรษฐกิจไทยเป็นอย่างมาก และด้วยสภาพแวดล้อมต่างๆในกระชังเพาะเลี้ยงสัตว์น้ำมีผลต่อการเจริญเติบโตของสัตว์น้ำเป็นอย่างมาก โดยทั่วไปแล้วมักประสบกับปัญหาสัตว์น้ำตายระหว่างเลี้ยงโดยไม่ทราบสาเหตุ ซึ่งส่วนใหญ่มาจากสภาพน้ำที่เน่าเสียทำให้สัตว์น้ำตาย และปัญหาน้ำเน่าเสียเกิดจากค่าความเป็นกรด-ด่างจะเปลี่ยนแปลงตลอดเวลา เนื่องจากการสะสมของเสียที่สัตว์น้ำถ่ายออกมา รวมทั้งเศษอาหารตกค้างในบ่อและชาก ฟีชีน้ำ การสังเคราะห์แสงของพืชน้ำ ก็มีผลทำให้ค่าความเป็นกรด-ด่างเปลี่ยนแปลงในช่วงบ่ายเวลาประมาณ 15.00-16.00น. การสังเคราะห์แสงจะถึงจุดสูงสุด ทำให้ค่าความเป็นกรด-ด่างในน้ำสูงมาก (สมชาย ธรรมขันหา, 2559) แต่ในช่วงกลางคืนพืชน้ำต่างๆจะหยุดสังเคราะห์แสงเนื่องจากไม่มีแสงแถมมีแต่การหายใจเพียงอย่างเดียว ค่าความเป็นกรด-ด่างจะลดลงมาก ค่าความเป็นกรด-ด่างในกระชัง ควรรักษาให้อยู่ในระดับ 7-9 และระดับที่เหมาะสมที่สุดคือ 7.5-8.5 (สมชาย ธรรมขันหา, 2559) ซึ่งเหมาะสมต่อการเจริญเติบโตของสัตว์น้ำมากที่สุดสัตว์น้ำสามารถจะกินอาหารได้ดีด้วย ดังนั้นควรรักษาระดับค่าความเป็นกรด-ด่างให้อยู่ในช่วง 7.5-8.5

นอกจากค่าความเป็นกรด-ด่างแล้ว ค่าความขุ่นในกระชังเพาะเลี้ยงสัตว์น้ำยังส่งผลกระทบต่อ การเจริญเติบโตของสัตว์น้ำ ซึ่งค่าที่เหมาะสมที่สุดในการเพาะเลี้ยงสัตว์น้ำไม่ควรเกิน 50 NTU (สมชาย ธรรมขันหา, 2559) รวมไปถึงอุณหภูมิในกระชังเพาะเลี้ยงสัตว์น้ำมีการเปลี่ยนแปลงตลอดเวลาซึ่งอุณหภูมิของน้ำที่เหมาะสมที่สุดในช่วง 25-30 องศาเซลเซียส(ไมตรี ดวงสวัสดิ์, 2558) ถ้าอุณหภูมิสูงหรือต่ำเกินไป อาจทำให้สัตว์น้ำอ่อนแอหรือตายได้ ทำให้ชาวประมงต้องเฝ้าระวังดูแลสัตว์น้ำในกระชังและสภาพแวดล้อมของน้ำในกระชังที่เพาะเลี้ยงสัตว์น้ำอย่างใกล้ชิดเพื่อไม่ให้สัตว์ตายและปัญหาดังกล่าวทำให้ผู้ที่ทำอาชีพประมงต้องใช้แรงงานคนในการดูแลจำนวนมาก และต้องคอยตรวจสอบวัดค่าต่างๆในน้ำที่เพาะเลี้ยงสัตว์เพื่อให้เหมาะสมต่อการเจริญเติบโตของสัตว์น้ำอยู่เสมอ

คณะผู้จัดทำโครงการจึงได้คิดที่จะพัฒนาเทคโนโลยีสารสนเทศและ Internet of Things เพื่อการเพาะเลี้ยงสัตว์น้ำโดยประยุกต์ใช้ Application บนระบบปฏิบัติการ Android บน Smartphone ซึ่งสามารถควบคุมกังหันน้ำผ่าน Cloud Platform เพื่อเพิ่มออกซิเจนในน้ำและแสดงผลการวัดคุณสมบัติของน้ำใน Application บน Smartphone ได้

1.2 วัตถุประสงค์

- 1.เพื่อศึกษาการใช้งานและการทำงานของบอร์ด ESP8266
- 2.เพื่อศึกษาการส่งข้อมูลและพัฒนการใช้งานผ่านระบบปฏิบัติการ Android
- 3.เพื่อศึกษาการใช้งานระบบการให้บริการ Cloud Platform ของ NETPIE (Network Platform for Internet of Everything)

1.3 ขอบเขตงาน

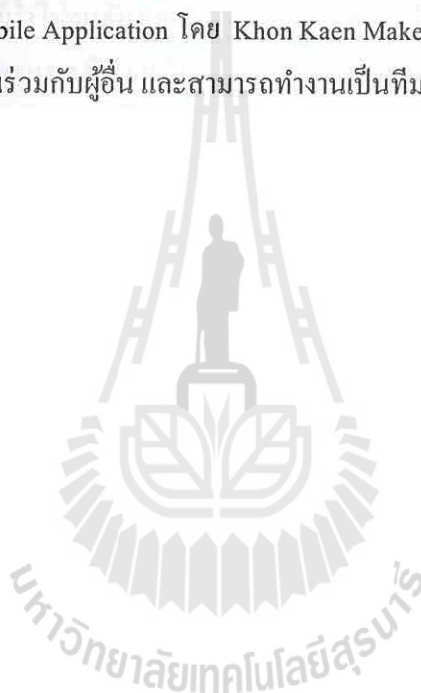
1. ศึกษาการทำงานของบอร์ด ESP8266
2. ศึกษาการส่งข้อมูลผ่าน Application เพื่อควบคุมกังหันน้ำ และแสดงค่าข้อมูลจากเซนเซอร์
3. สามารถนำเอาข้อมูลในระบบ Cloud Platform มาแสดงค่าเซนเซอร์และควบคุมกังหันน้ำผ่าน Application และผ่าน NETPIE Freeboard ได้อย่างมีประสิทธิภาพ

1.4 ขั้นตอนการดำเนินงาน

- 1.ศึกษา ค้นคว้าหาข้อมูล
- 2.เขียนโครงงานและเสนอโครงงานกับอาจารย์ที่ปรึกษา
- 3.หาซื้ออุปกรณ์ที่จำเป็นต้องใช้ในโครงงานนี้
- 4.ออกแบบวงจรและเขียน Code และเก็บค่าเซนเซอร์
- 5.ศึกษาการใช้งานระบบการให้บริการ Cloud Platform ของ NETPIE (Network Platform for Internet of Everything)
- 6.ศึกษาการใช้งานและการทำงานของ บอร์ด ESP8266 Arduino UNO R3
- 7.ศึกษาการใช้งานและการทำงานของ บอร์ด ESP8266 D1 Arduino WiFi
- 8.ศึกษาข้อมูลเกี่ยวกับปัจจัยต่างๆ ที่มีผลต่อการเพาะเลี้ยงสัตว์น้ำ
- 9.สร้างอุปกรณ์และนำไปทดสอบ
- 10.สรุปผลการทดลองและเขียนรายงาน
- 11.นำเสนอโครงงาน

1.6 ผลที่คาดว่าจะได้รับ

- 1.สามารถนำความรู้ทางทฤษฎีที่ได้เรียนมาประยุกต์ใช้ในทางปฏิบัติได้
- 2.สามารถใช้งานระบบการให้บริการ Cloud Platform ของ NETPIE (Network Platform for Internet of Everything) เพื่อนำไปประยุกต์ใช้และต่อยอดเทคโนโลยีอื่นๆต่อไป
- 3.ระบบที่ออกแบบ วัดค่าพารามิเตอร์ต่างๆในน้ำ และควบคุมกังหันน้ำ สามารถช่วยให้ผู้ใช้ทราบข้อมูล ควบคุมกังหันน้ำผ่านทาง Application และผ่านทาง NETPIE Freeboard
- 4.ได้ศึกษารูปแบบการส่งข้อมูลและควบคุมกังหันน้ำผ่าน Application KKMC IoT (ออกแบบและพัฒนา Mobile Application โดย Khon Kaen Maker Club) เชื่อมต่อผ่าน NETPIE
- 5.รู้จักการทำงานร่วมกับผู้อื่น และสามารถทำงานเป็นทีมได้



บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 บทนำ

ในการเพาะเลี้ยงสัตว์น้ำสิ่งที่จะต้องคำนึงถึงคือคุณภาพของน้ำที่ใช้ในการเพาะเลี้ยงสัตว์น้ำ สิ่งส่งผลกระทบต่อทำให้คุณภาพของน้ำมีสภาพที่ไม่เหมาะสมในการเพาะเลี้ยงสัตว์น้ำโดยส่วนใหญ่แล้วจะเป็นพวกอาหารของสัตว์น้ำที่ให้ในแต่ละวัน ซึ่งการจัดเก็บอาหารหรือวัตถุดิบมีความสำคัญมาก ถึงแม้ว่าอาหารสัตว์น้ำที่ผลิตขึ้นมาจะมีคุณภาพดี หากการจัดเก็บรักษาทั้งวัตถุดิบและอาหารสัตว์น้ำไม่ถูกต้องเหมาะสมก็ย่อมมีผลทำให้อาหารสัตว์น้ำสูญเสียคุณภาพและเป็นแหล่งให้เชื้อราและแบคทีเรียเจริญเติบโตได้ ซึ่งแน่นอนว่าย่อมมีผลกระทบไปถึงการเจริญเติบโตและคุณภาพของสัตว์น้ำที่เกษตรกรเลี้ยง และความปลอดภัยต่อผู้บริโภค

โดยเกษตรกรต้องมีการเฝ้าระวังและตรวจคุณภาพของน้ำที่ใช้เลี้ยงสัตว์น้ำอย่างสม่ำเสมอ ในกรณีที่พบว่าคุณภาพของน้ำมีสภาพที่ไม่เหมาะสมในการเพาะเลี้ยงสัตว์น้ำ และให้สังเกตดูว่าถ้าสัตว์น้ำมีอาการผิดปกติให้ดำเนินการแก้ไขทันที เช่น พิจารณาปรับสภาพแวดล้อมในบ่อ และลดปริมาณอาหารก่อน พิจารณาการใช้ยา (กรณีบ่อดิน และกระชัง)

ในการเพาะเลี้ยงสัตว์น้ำ ผู้เลี้ยงส่วนใหญ่จะนิยมใช้ยาและสารเคมีในการรักษาสัตว์น้ำที่มีอาการผิดปกติ ซึ่งเป็นการแก้ไขปัญหาก็ไม่ถูกต้อง และอาจก่อให้เกิดปัญหาอื่นๆตามมา เช่น ปัญหาการดื้อยาซึ่งจะทำให้ต้องใช้ยาในปริมาณมากขึ้นหรืออาจไม่สามารถใช้ยาชนิดเดิมรักษาให้หายได้อีก ซึ่งปัญหาการเกิดโรค มีสาเหตุสำคัญคือ สภาพแวดล้อมในบ่อ คุณภาพน้ำ ทำให้สัตว์น้ำอ่อนแอและเกิดการติดเชื้อได้ง่าย โดยส่วนมากแล้วคุณภาพของน้ำที่เหมาะสมต่อการเพาะเลี้ยงสัตว์จะขึ้นกับค่าอุณหภูมิของน้ำ ค่าความเป็นกรด-ด่างและค่าความขุ่น ค่าพวกนี้จะมีการเปลี่ยนแปลงอยู่ตลอดเวลาขึ้นกับสภาพแวดล้อม ภูมิอากาศในแต่ละวัน

2.2 คุณภาพน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ

2.2.1 อุณหภูมิ(Temperature) และช่วงแสง(Photoperiod)

อุณหภูมิของน้ำเป็นปัจจัยที่สำคัญทั้งทางตรงและทางอ้อมต่อการดำรงชีพ การเจริญเติบโต การเจริญพันธุ์ให้ดีขึ้น (Stacey, 1983) ในปลาทองพบว่าถ้าอุณหภูมิน้ำต่ำกว่า 14 องศาเซลเซียส แม้มีไข่แต่จะไม่วาง จะวางไข่เมื่ออุณหภูมิเพิ่มขึ้นที่ 20 องศาเซลเซียส (Yamamoto และ Yamazaki, 1966) ปกติอุณหภูมิของน้ำตามธรรมชาติจะผันแปรกับอุณหภูมิอากาศ และเส้น Latitude กล่าวคือ อุณหภูมิจะสูงเมื่ออยู่ใกล้เส้นศูนย์สูตร และจะต่ำเมื่อใกล้ขั้วโลก สัตว์น้ำเป็นพวกสัตว์เลือดเย็น (Poikilotherma) จะปรับอุณหภูมิร่างกายตามอุณหภูมิน้ำจะทนได้ช่วงอุณหภูมิ

ที่ต่ำหากอุณหภูมิสูงและช่วงแสงมากกิจกรรมต่าง ๆ ในการดำรงชีพจะสูงตามและลดลงเมื่ออุณหภูมิต่ำ ขบวนการเมตาโบลิซึมจะเพิ่มขึ้น 2-3 เท่า เมื่ออุณหภูมิน้ำเพิ่มขึ้น 10 องศาเซลเซียส ขบวนการที่สำคัญได้แก่ การหายใจ การว่ายน้ำ การกินอาหาร การย่อยอาหารและขับถ่าย ปกติอุณหภูมิตัวปลาจะต่ำกว่าอุณหภูมิของน้ำ 0.5-1 องศาเซลเซียส เหงือกเป็นอวัยวะสำคัญที่รักษาอุณหภูมิ ปลาขนาดเล็กชนิดเดียวกันสามารถปรับและรักษาอุณหภูมิได้ดีกว่าปลาขนาดใหญ่ เนื่องจากสัดส่วนปริมาตรของเหงือกกับร่างกายมีมากกว่า เพราะฉะนั้นการเปลี่ยนแปลงอุณหภูมิอย่างรวดเร็วจะเกิดอันตรายต่อ สัตว์น้ำ ทำให้ร่างกายอ่อนแอหรือช็อกตายได้ เช่น การขนส่งสัตว์น้ำ ในขณะที่อุณหภูมิสูง อุณหภูมิยังมีอิทธิพลต่อการควบคุมการถ่ายน้ำแร่ธาตุในร่างกาย ความหนาแน่นของน้ำ การละลายของออกซิเจนในแหล่งน้ำลดลง เมื่ออุณหภูมิสูงขึ้น การแบ่งชั้นของน้ำตามอุณหภูมิในแหล่งน้ำลึกมากกว่า 2.5 เมตร (Thermal Stratification) การเปลี่ยนแปลงอุณหภูมิยังมีผลต่อการเจริญเติบโตของ พืชน้ำ แพลงก์ตอน ซึ่งเกี่ยวข้องกับกำลังผลิตเบื้องต้นของแหล่งน้ำ เช่น ไคอะตอม ของแหล่งน้ำที่อุณหภูมิ 15-25 องศาเซลเซียส สาหร่ายสีเขียวชอบที่อุณหภูมิ 25-35 องศาเซลเซียส สาหร่ายน้ำเงินแกมเขียว ที่อุณหภูมิมากกว่า 35 องศาเซลเซียส ซึ่งไม่เป็นประโยชน์บางครั้งทำให้เกิดน้ำเสียเป็นพิษแก่สัตว์น้ำ อุณหภูมิสูงเกินไป ทำให้สาหร่ายประเภทต่างๆ เช่น ยากำจัดศัตรูพืชและแมลง มีพิษมากยิ่งขึ้น

(<http://aquaclub.net/forum/index.php?topic=19106.0;wap2>)

2.2.2 ความขุ่นและความโปร่งใส(Turbidity and Transparency)

ความขุ่นหรือความโปร่งใสของน้ำเป็นตัวที่กำหนดกำลังผลิตของแหล่งน้ำได้ เนื่องจากความขุ่นของน้ำเกิดจากอนุภาคของสารแขวนลอยกับอนุภาคดิน สารอินทรีย์แร่ธาตุต่าง ๆ สิ่งที่มีชีวิตขนาดเล็ก เช่น แพลงก์ตอน แบคทีเรีย เป็นตัวที่ควบคุมให้ปริมาณแสงอาทิตย์ส่องผ่านลงสู่

แหล่งน้ำนอกจากนี้ความขุ่นของน้ำจะมีผลต่อการหายใจของสัตว์น้ำ เนื่องจากอนุภาคสารแขวนลอย มีโอกาสอุดตามช่องเหงือกทำให้ประสิทธิภาพการหายใจลดลง ผลต่ออัตราการเจริญเติบโต การสร้างเซลล์สืบพันธุ์และอัตราการฟักไข่ลดลง ความขุ่นทำให้น้ำผิวบนดูดซับความร้อนได้ดีทำให้อุณหภูมิสูงขึ้น

ค่าของความขุ่นนิยมแสดงในรูปของหน่วย (Unit) ซึ่งหมายถึงระดับความลึกของน้ำที่สามารถมองเห็นแสงสว่างจากแสงเทียนมาตรฐาน น้ำที่ใสจะมีค่าความขุ่นไม่เกิน 25 หน่วย ส่วนน้ำขุ่นปานกลางจะมีค่าความขุ่นระหว่าง 25-100 หน่วย และน้ำขุ่นมากจะมีค่าความขุ่นเกิน 100 หน่วยขึ้นไป ความขุ่นที่เป็นอันตรายต่อสัตว์น้ำจนถึงแก่ชีวิตจะต้องมีมากกว่า 20,000 หน่วยขึ้นไป

โดยทั่วไปแล้วปริมาณสารแขวนลอย นิยมวัดเป็นน้ำหนักในรูปของมิลลิกรัมต่อลิตร (mg/L) แหล่งน้ำที่ให้ผลผลิตทางการประมงที่ดี ควรจะมีค่าปริมาณสารแขวนลอยอยู่ในช่วงระหว่าง 25-80 มิลลิกรัมต่อลิตร แต่ถ้าอยู่ในช่วงระหว่าง 80-400 มิลลิกรัมต่อลิตรจะให้ผลผลิตลดลง นอกจากนี้ปริมาณสารแขวนลอยเป็นอีกดัชนีหนึ่งที่ใช้วัดค่าความขุ่นของน้ำ
(<http://aqua.c1ub.net/forum/index.php?topic=19106.0;wap2>)

2.2.3 ค่าความเป็นกรด-ด่าง(pH)

ในแหล่งน้ำค่าเป็นกรด-ด่าง ที่เหมาะสมควรมีค่าระหว่าง 6.5-8 น้ำเป็นกรดจะทำให้ น้ำเชื้ออ่อนแอ สัตว์น้ำไม่สามารถเจริญพันธุ์ได้ตามปกติในสภาพที่ไม่เหมาะสม มีฤทธิ์เป็นกรด เมื่อมีค่าน้อยกว่า 7

pH ของแหล่งน้ำทั่วไปมีค่า 5-9 ขึ้นอยู่กับสภาพแวดล้อมและภูมิประเทศ และจะผันแปรตาม pH ของดิน อิทธิพลของจุลินทรีย์และแพลงค์ตอนพืชก็สามารถทำให้ pH เปลี่ยนแปลง pH มีความสำคัญต่อการใช้อาหารของสัตว์น้ำ พืชน้ำ ใช้ได้ดีหรือไม่ขึ้นอยู่กับ pH ในน้ำ ถ้า pH ต่ำกว่า 4.5 หรือสูงกว่า 11 ทำให้สัตว์น้ำไม่สามารถดำรงชีวิตหรือเจริญเติบโตได้ pH ระดับสูงทำให้แอมโมเนียมีค่ามากขึ้น ปกติแล้วในรอบวันค่า pH จะสูงขึ้นเรื่อยจนถึงช่วงบ่าย และลดลงในเวลากลางคืน การปรับค่า pH ของน้ำจากความเป็นกรดให้เป็นด่าง โดยมากใช้ปูนขาวที่นิยมมี 3 วิธี CaCO_3 , Ca(OH)_2 และ CaO ปริมาณมากน้อยขึ้นอยู่กับความรุนแรงของค่า pH และคุณภาพดิน
(<http://aqua.c1ub.net/forum/index.php?topic=19106.0;wap2>)

2.2.4 ปริมาณออกซิเจนในน้ำ(Dissolved Oxygen)

ออกซิเจนเป็นอีกปัจจัยหนึ่งที่มีความสำคัญมากในการดำรงชีพ เนื่องจากต้องใช้ ออกซิเจนในขบวนการหายใจเพื่อการเจริญเติบโต ความสามารถละลายน้ำของออกซิเจนขึ้นกับ ความดันอากาศ อุณหภูมิ ปริมาณเกลือแร่ เช่นที่ อุณหภูมิ 0 องศาเซลเซียส การละลายของ ออกซิเจนอยู่ที่ 114 มก./ลิตร และที่ 6-9 มก./ลิตร ที่ 35 องศาเซลเซียสที่ความดัน 1 บรรยากาศ เมื่ออุณหภูมิสูงทำให้เกิดกิจกรรมใช้ออกซิเจนมาก ทำให้ปริมาณออกซิเจนลดลงอาจเกิดการขาด ออกซิเจนได้ ในทางตรงข้ามการเกิดออกซิเจนละลายเกินจุดอิ่มตัว เนื่องจากการผลิตออกซิเจน มากเกินไป จากขบวนการสังเคราะห์แสงของพืชสีเขียว ก็เป็นอันตรายต่อสัตว์น้ำเช่นกัน
(<http://aqua.c1ub.net/forum/index.php?topic=19106.0;wap2>)

2.3 Sensor วัดอุณหภูมิในน้ำ



รูปที่ 2.1 DS18B20 เซ็นเซอร์วัดอุณหภูมิแบบกันน้ำได้

(ข้อมูลจาก <http://www.arduinoberry.com/product/21/ds18b20>)

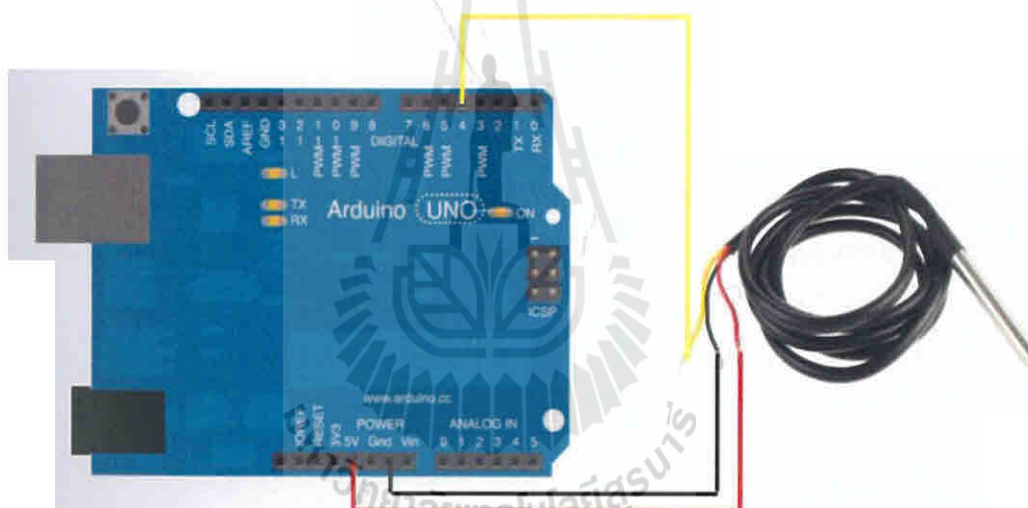
DS18B20 เซ็นเซอร์วัดอุณหภูมิแบบกันน้ำได้เหมาะสำหรับการวัดอุณหภูมิในของเหลว
ต่อกับตัวต้านทานดังรูปเพื่อเชื่อมต่อกับ Arduino

คุณสมบัติของ Sensor วัดอุณหภูมิในน้ำ

- ใช้แรงดันไฟเลี้ยง Vdd (หรือ Vcc) ได้ในช่วง 3.0V ถึง 5.5V
- มี 3 ขา (สำหรับตัวถัง TO-92) คือ Gnd (Pin 1), DQ (Pin 2), VDD (Pin 3)
- ใช้งานได้สองแบบ: normal mode (ใช้ทั้ง 3 ขา) และ parasite power mode (ใช้เพียง 2 ขา คือ DQ และ GND ในขณะที่ขา Vdd จะต่อกับขา Gnd)
- สามารถนำไอซีมาพ่วงต่อกันในบัสเดียว (เส้นสัญญาณ DQ) ได้หลายอุปกรณ์
- ในการใช้งาน จะต้องต่อ pull-up $4.7k\Omega$ (หรือน้อยกว่าได้เล็กน้อย) ที่ขา DQ กับแรงดันไฟเลี้ยง
- วัดอุณหภูมิได้ในช่วง -55°C ถึง $+125^{\circ}\text{C}$
- มีความแม่นยำ $\pm 0.5^{\circ}\text{C}$ สำหรับอุณหภูมิในช่วง -10°C ถึง $+85^{\circ}\text{C}$
- มีความละเอียดของค่าที่อ่านได้ 12 บิต (Resolution)
- ใช้เวลาในการแปลงข้อมูลสำหรับ ADC ไม่เกิน 750 msec (มิลลิวินาที) สำหรับข้อมูล 12 บิต
- ไอซีแต่ละตัวมีหมายเลขเฉพาะตัว ขนาด 64 บิต (64-bit serial code)
- สำหรับตระกูล DS18B20 มีค่าไบต์สำหรับ 8-bit family code ตรงกับ 28h (0x28) เป็นไบต์แรกของหมายเลขอุปกรณ์

ภายในไอซี DS18B20 มีหน่วยความจำแบบ SRAM ขนาดความจุ 9 ไบต์ (Byte 0 ถึง Byte 9) และเรียกว่า Scratchpad ส่วนหนึ่งของหน่วยความจำนี้ จะใช้สำหรับเก็บค่าอุณหภูมิที่ได้จากการอ่านและแปลงเป็นข้อมูลดิจิทัลในแต่ละครั้ง (ใช้ 2 ไบต์ และเก็บไว้ใน Byte 0 และ Byte 1) และยังมีการคำนวณค่า CRC (checksum) ขนาดหนึ่งไบต์ด้วย (เก็บไว้ใน Byte 8)

ในการเขียนโปรแกรมสำหรับ Arduino เพื่อเชื่อมต่อกับไอซี DS18B20 สามารถใช้ library "OneWire.h" ของ Arduino ซึ่งทำให้สะดวกในการเขียนโปรแกรม เช่น เพื่อการค้นหาหลายเลข (64-bit code) ของอุปกรณ์ที่ต่ออยู่กับบัส 1-Wire การสั่งให้ไอซีตามหมายเลขอุปกรณ์ อ่านและแปลงค่าอุณหภูมิ จากนั้นเว้นระยะเวลาอย่างน้อย 750msec แล้วจึงอ่านค่าจากหน่วยความจำ Scratchpad ของไอซี



รูปที่ 2.2 การต่อ Sensor วัดอุณหภูมิในน้ำ เข้ากับบอร์ด Arduino

(ข้อมูลจาก <https://www.arduitronics.com/product/478/waterproof-temperature-sensor-ds18b20-3-meter>)

2.4 Sensor วัดค่า pH ในน้ำ



รูปที่ 2.3 Analog pH Metre Kit

(ข้อมูลจาก <http://www.thaieasyelec.com/products/sensors/ph-d-o-orp/analog-ph-meter-kit-detail.html>)

ชุดตรวจวัดค่า pH (ความเป็นกรด-ด่าง) ในน้ำ ในชุดประกอบด้วยหัวโพรบเซนเซอร์ ตรวจวัดและวงจรอินเทอร์เฟสสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์ เอาต์พุตเป็นอนาลอก ใช้ไฟเลี้ยง 5 โวลต์ ใช้จุ่มเพื่อตรวจวัดเป็นระยะเวลานานๆ ไม่เหมาะกับแช่ต่อเนื่อง อุณหภูมิเหมาะสมที่ 25 องศาเซลเซียส สามารถตรวจสอบค่า pH ได้ตั้งแต่ 0 ถึง 14 โดยช่วงที่เหมาะสมกับการใช้งานอยู่ในช่วง pH ประมาณ 6 ถึง 8 โดยอุณหภูมิและค่า pH ที่สูงหรือต่ำเกินไป และความขุ่นของน้ำมีผลต่ออายุการใช้งานของเซนเซอร์ หากใช้น้ำขุ่น เป็นกรดมาก หรือเป็นด่างมาก อุณหภูมิสูง อายุการใช้งานของเซนเซอร์จะสั้นลง

2.4.1 การใช้งานของ Analog pH Meter Kit

การใช้อินเตอร์ด Arduino วัดค่า pH ของน้ำด้วย pH sensor kits โดย บอร์ด Arduino จะต่อเข้ากับตัว pH sensor module ผ่านทาง SoftwareSerial ของบอร์ดส่วนการต่ออุปกรณ์เข้ากับ pH sensor module ให้ต่ออย่างระมัดระวังเพราะถ้าสลับสายอาจทำให้บอร์ดพังได้ เพราะตัว pH sensor module ออกแบบไว้ให้ใช้กับไฟ 3.3v แต่ตัว MCU ของบอร์ดสามารถรับไฟได้ถึง 5V ดังนั้นจึงสามารถใช้กับบอร์ด Arduino ได้

คุณสมบัติของ Analog pH Metre Kit

- Module Power : 5.00V
- Module Size : 43 x 32mm
- Measuring Range : 0 - 14pH

- Measuring Temperature: 0 °C – 60 °C
- Accuracy : $\pm 0.1\text{pH}$ (25)
- Response Time : $\leq 1\text{min}$
- pH Sensor with BNC Connector
- pH2.0 Interface (3 foot patch)
- Gain Adjustment Potentiometer
- Power Indicator LED

2.5 Sensor วัดค่าความขุ่นในน้ำ



รูปที่ 2.4 Analog Turbidity Sensor

(ข้อมูลจาก <https://www.arduitronics.com/product/140/4-channel-relay>)

เซ็นเซอร์ตรวจวัดความขุ่นในน้ำ โดยการวัดระดับของความขุ่นสามารถใช้ในการตรวจจับอนุภาคแขวนลอยในน้ำโดยการวัดการส่งผ่านแสงและกระจายอัตราซึ่งการเปลี่ยนแปลงที่มีปริมาณของสารแขวนลอยทั้งหมดในน้ำ

2.5.1 หลักการทำงาน Analog Turbidity Sensor

เซ็นเซอร์วัดระดับความขุ่นของน้ำโดยตรวจสอบจากการนำและหักเหของแสง ทำงานที่แรงดัน 5 โวลต์ ให้เอาต์พุตได้สองแบบ คือ อนุลอกแรงดันตั้งแต่ 0.4 ถึง 5 โวลต์ และ ดิจิตอล High / Low โดยปรับค่าที่ตัวต้านทานปรับค่าได้

คุณสมบัติของ Sensor วัดค่าความขุ่นในน้ำ

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time: <500ms
- Insulation Resistance: 100M (Min)
- Output Method: Analog

- Analog output: 0-4.5V
- Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5 องศา ~90 องศา
- Storage Temperature: -10องศา~90องศา
- Weight: 30g
- Adapter Dimensions: 38mm*28mm*10mm/1.5inches *1.1inches*0.4inche

2.6 Relay Module

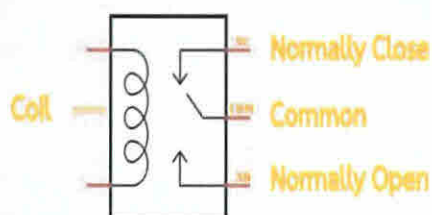


รูปที่ 2.5 Relay Module 2 Channels แบบ Active High

(ข้อมูลจาก <https://www.arduitronics.com/product/641/2-channel-relay-opto-isolated>)

2.6.1 หลักการทำงานของ Relay Module

รีเลย์ (Relay) เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่ง ซึ่งทำหน้าที่ตัดต่อวงจรแบบเดียวกับสวิตช์ โดยควบคุมการทำงานด้วยไฟฟ้า Relay มีหลายประเภท ตั้งแต่ Relay ขนาดเล็กที่ใช้ในงานอิเล็กทรอนิกส์ทั่วไป จนถึง Relay ขนาดใหญ่ที่ใช้ในงานไฟฟ้าแรงสูง โดยมีรูปร่างหน้าตาแตกต่างกันออกไป แต่มีหลักการทำงานที่คล้ายคลึงกัน สำหรับการนำ Relay ไปใช้งาน จะใช้ในการตัดต่อวงจร ทั้งนี้ Relay ยังสามารถเลือกใช้งานได้หลากหลายรูปแบบ



รูปที่ 2.6 สัญลักษณ์ในวงจรไฟฟ้าของรีเลย์

ภายใน Relay จะประกอบไปด้วยขดลวดและหน้าสัมผัส

หน้าสัมผัส NC (Normally Close) เป็นหน้าสัมผัสปกติปิด โดยในสภาวะปกติหน้าสัมผัสนี้จะต่อเข้ากับขา COM (Common) และจะลดยหรือไม่สัมผัสกันเมื่อมีกระแสไฟฟ้าไหลผ่านขดลวด

หน้าสัมผัส NO (Normally Open) เป็นหน้าสัมผัสปกติเปิด โดยในสภาวะปกติจะลดยอยู่ไม่ถูกต่อกับขา COM (Common) แต่จะเชื่อมต่อกันเมื่อมีกระแสไฟฟ้าไหลผ่านขดลวด ขา COM (Common) เป็นขาที่ถูกใช้งานร่วมกันระหว่าง NC และ NO ขึ้นอยู่กับว่า ขณะนั้นมีกระแสไฟฟ้าไหลผ่านขดลวดหรือไม่ หน้าสัมผัสใน Relay 1 ตัวอาจมีมากกว่า 1 ชุด

การทำงานของ Relay ในส่วนของขดลวด เมื่อมีกระแสไฟฟ้าไหลผ่าน จะทำให้ขดลวดเกิดการเหนี่ยวนำและทำหน้าที่เสมือนแม่เหล็กไฟฟ้า ส่งผลให้ขา COM ที่เชื่อมต่ออยู่กับหน้าสัมผัส NC (ในสภาวะที่ยังไม่เกิดการเหนี่ยวนำ) ย้ายกลับเชื่อมต่อกับหน้าสัมผัส NO แทน และปล่อยให้ขา NC ลอย เมื่อมองที่ขา NC กับ COM และ NO กับ COM แล้วจะเห็นว่ามีการทำงานติด-ดับลักษณะคล้ายการทำงานของสวิตช์เราสามารถอาศัยคุณสมบัตินี้ไปประยุกต์ใช้งานได้

2.7 Amplifier Module



รูปที่ 2.7 ADS1115 I2C ADC 4 Channel 16-Bit with Programmable Gain Amplifier Module (ข้อมูลจาก <https://www.arduitronics.com/product/851/ads1115-i2c-adc-4-channel-16-bit-with-programmable-gain-amplifier-module>)

- 1x ADS1115 I2C ADC 4 Channel 16-Bit with Programmable Gain Amplifier Module

- โมดูลแปลงสัญญาณแอนาลอกเป็นดิจิทัล จำนวน 4 อินพุต AN0, AN1, AN2, AN3 หรือทำเป็น 2 differential inputs

- ความละเอียดขนาด 16 บิต

- สามารถโปรแกรมอัตราขยายได้ (Programmable Gain Amplifier:PGA)
- เชื่อมต่อกับไมโครคอนโทรลเลอร์แบบ I2C Address: 0x48 - 0x4B
- แรงดันไฟเลี้ยง: 2.0V หรือ 5.5 Vdc (แนะนำ 3.3 หรือ 5.0V)

2.8 Wemos D1 WiFi esp8266



รูปที่ 2.8 Wemos D1 WiFi esp8266

(ข้อมูลจาก <http://www.arduinothai.com/product/802/wemos-d1-wifi-uno-development-board-esp8266>)

Wemos D1 WiFi esp8266 เป็นบอร์ดที่ถูกออกแบบมาให้มีลักษณะเหมือนกับ Arduino UNO(จะทำให้ใช้ shield บางตัวที่เข้ากับ UNO ได้)สามารถใช้กับ Arduino IDE เขียนโปรแกรมได้

- Based on ESP-8266EX
- Compatible, IDE to program
- 11 x I / O pins
- 1 x ADC pin (input range 0-3.3V)
- Support OTA wireless uploading
- Onboard 5V 1A Switching Power Supply (maximum input voltage of 24V)

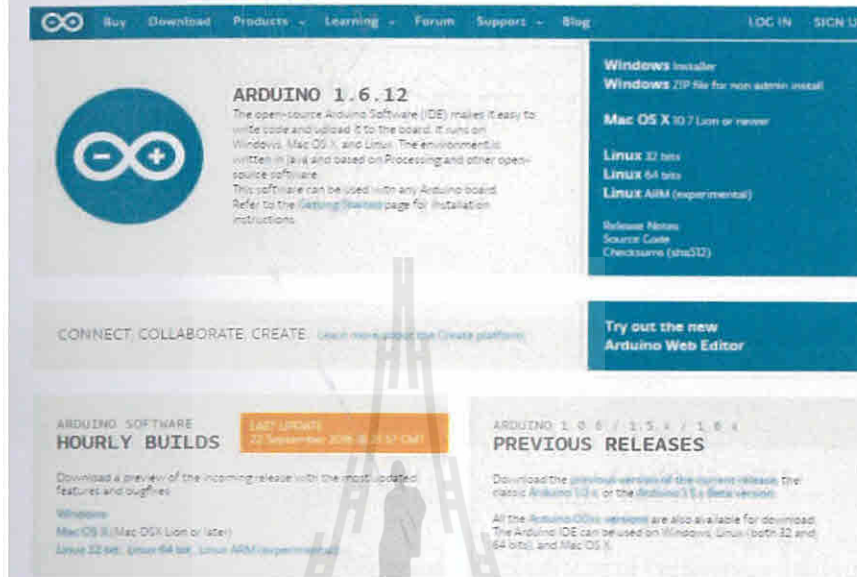
Pin	Function	ESP8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	Analog input, max 3.3V input	A0
D0	IO	GPIO16
D1	IO, SCL	GPIO5
D2	IO, SDA	GPIO4
D3	IO, 10k Pull-up	GPIO0
D4	IO, 10k Pull-up, (BUILTIN_LED)	GPIO2
D5	IO, SCK	GPIO14
D6	IO, MISO	GPIO12
D7	IO, MOSI	GPIO13
D8	IO, 10k Pull-down, SS	GPIO15
G	Ground	GND
5V	5V	
3V3	3.3V	3.3V
RST	Reset	RST

รูปที่ 2.9 การทำงานของพอร์ตต่างๆในบอร์ด Wemos D1 WiFi esp8266

2.8.1 การติดตั้ง Driver Arduino บน Windows

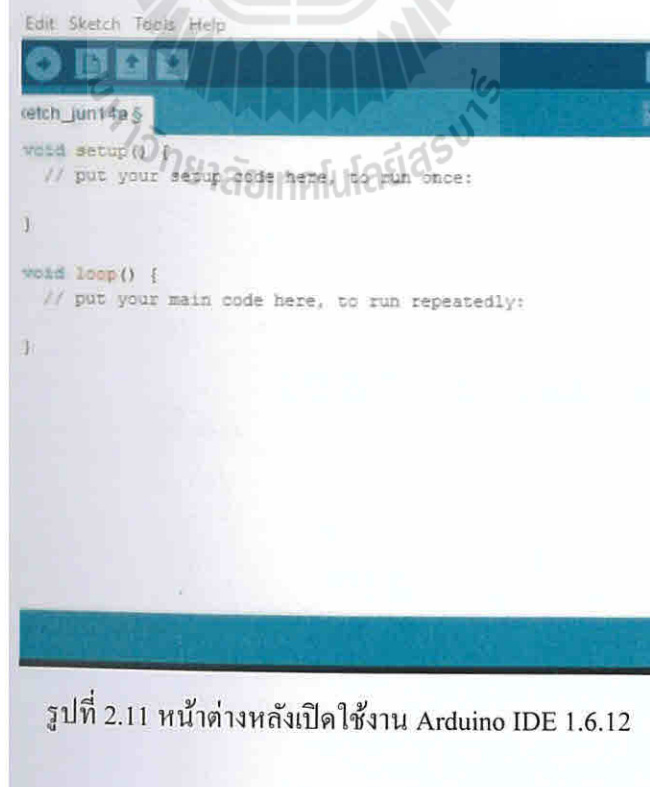
ขั้นตอนที่ 1 ดาวน์โหลดไฟล์โปรแกรมได้จากเว็บไซต์

<http://www.arduino.cc/en/Main/Software> เลือกระบบปฏิบัติการที่ต้องการจะติดตั้ง



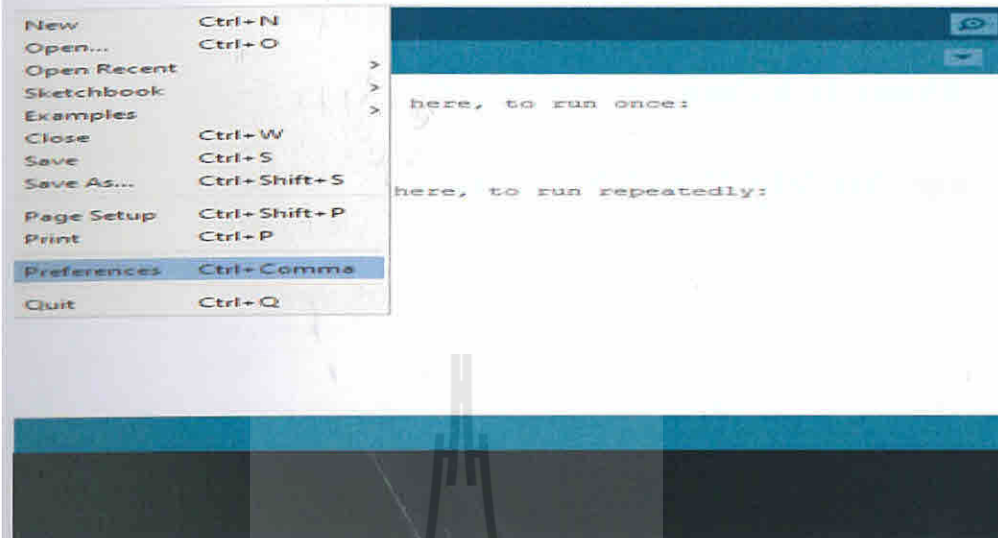
รูปที่ 2.10 ดาวน์โหลด Arduino IDE

ขั้นตอนที่ 2 เมื่อเปิดโปรแกรมแล้วจะพบกับหน้าต่าง ดังรูป



รูปที่ 2.11 หน้าต่างหลังเปิดใช้งาน Arduino IDE 1.6.12

ขั้นตอนที่ 3 ติดตั้ง Broad ESP8266 ลงบน Arduino IDE

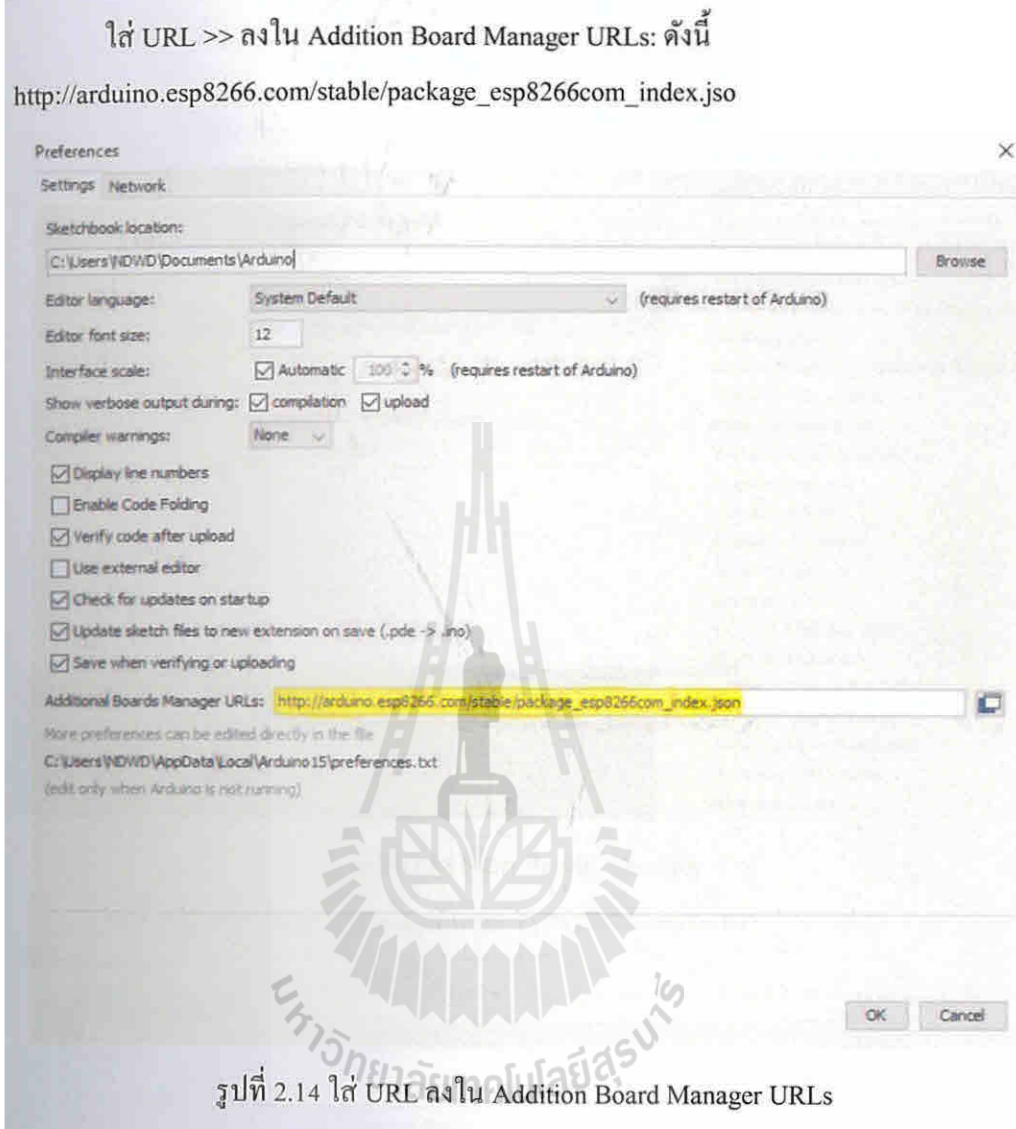


รูปที่ 2.12 เลือก Preferences

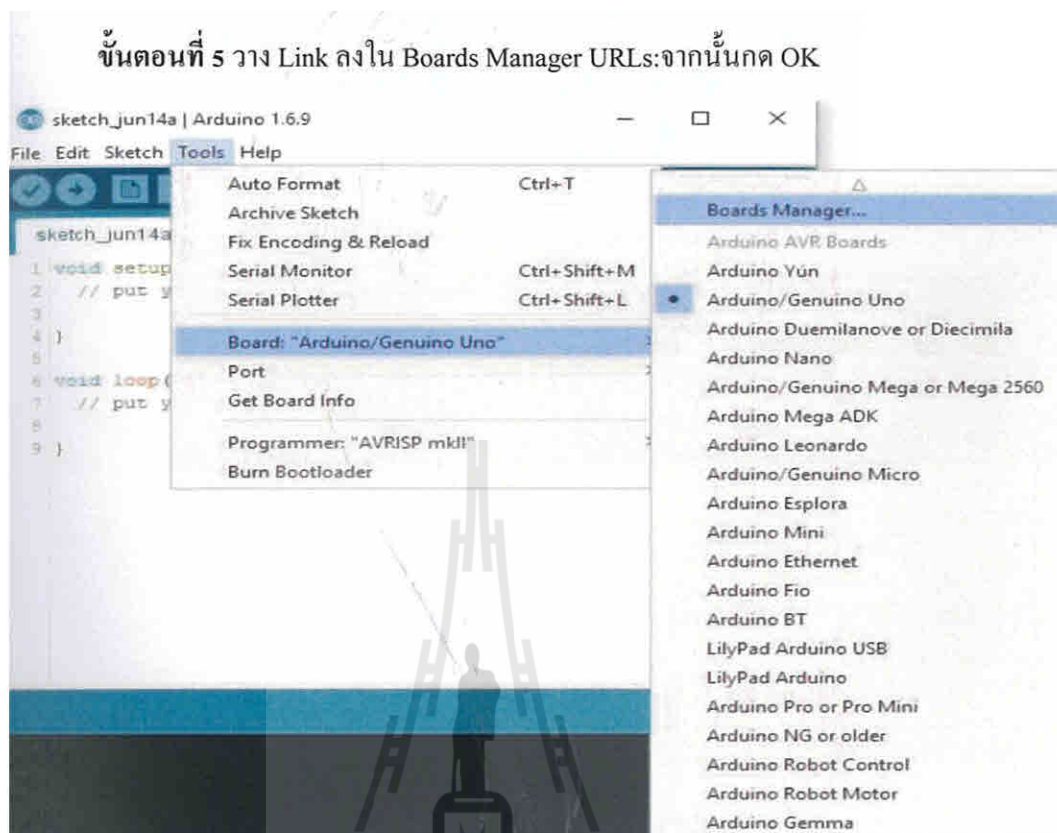
ขั้นตอนที่ 4 เข้าเมนู File >> Preferences จะขึ้นหน้าต่าง Preferences ให้สังเกตในช่อง Additional Board



รูปที่ 2.13 แสดงหน้าต่างของ Preferences



ขั้นตอนที่ 5 วาง Link ลงใน Boards Manager URLs:จากนั้นกด OK



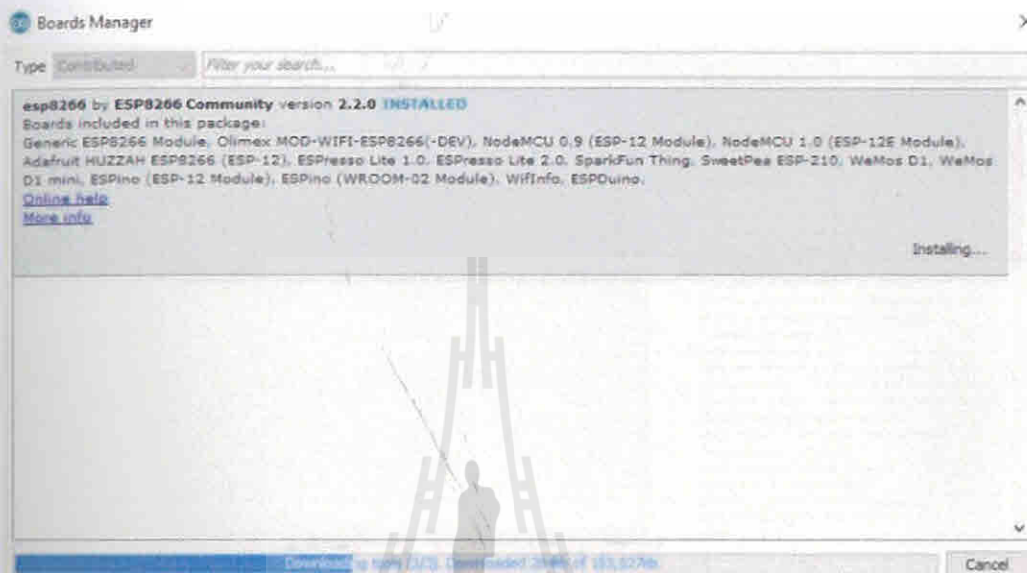
รูปที่ 2.15 เลือก Boards Manager

จะขึ้นหน้าต่าง Boards Manager เลือก Type เป็น Contributed



รูปที่ 2.16 ESP8266 เลือกเวอร์ชัน 2.2.0

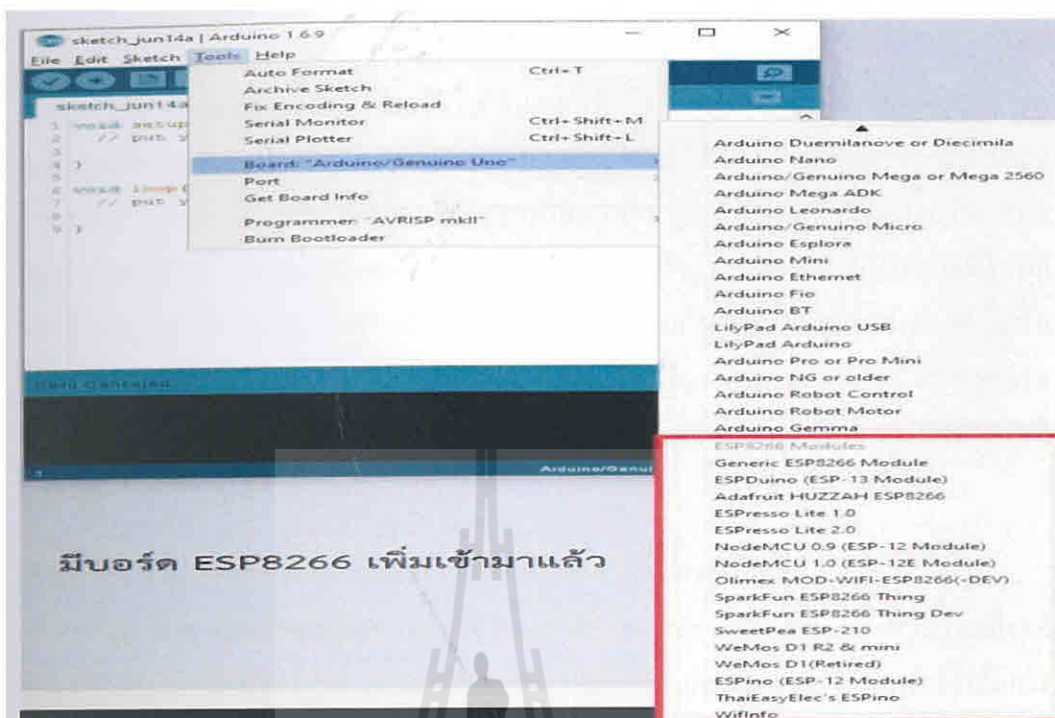
หน้าต่าง Boards Manager จะแสดง Boards ของ ESP8266 เลือกเวอร์ชัน 2.2.0 หรือใหม่กว่าในอนาคต Boards และกด Install รอตัวโปรแกรมจะโหลด Boards ESP8266 ให้ และติดตั้งให้เอง



รูปที่ 2.17 Download Boards ESP8266

ขั้นตอนที่ 6 เมื่อติดตั้งบอร์ด ESP8266 เสร็จเรียบร้อยแล้ว ให้ปิดโปรแกรม Arduino IDE ก่อน แล้วจึงเปิดขึ้นมาใหม่

- เมื่อเปิดโปรแกรม Arduino IDE ขึ้นมาใหม่แล้ว ให้ลองเลือกไปที่ Menu Tools >> Board ซึ่งจะพบว่า มี Menu สำหรับเลือกใช้งาน ESP8266 กับ Arduino IDE ขึ้นมาให้เลือกใช้งานแล้ว



รูปที่ 2.18 เลือกใช้งานงาน ESP8266 กับ Arduino IDE

2.9 Internet of Things

Internet of Things หรือ IoT คือ สภาพแวดล้อมอันประกอบด้วยสรรพสิ่งที่สามารถสื่อสารและเชื่อมต่อกันได้ผ่านโพรโทคอลการสื่อสารทั้งแบบใช้สายและไร้สาย โดยสรรพสิ่งต่างๆ มีวิธีการระบุตัวตนได้ รับรู้บริบทของสภาพแวดล้อมได้ และมีปฏิสัมพันธ์โต้ตอบและทำงานร่วมกันได้ ความสามารถในการสื่อสารของสรรพสิ่งนี้จะนำไปสู่นวัตกรรมและบริการใหม่อีกมากมาย ตัวอย่างเช่น เซ็นเซอร์ภายในบ้านตรวจจับการเคลื่อนไหวของผู้อยู่อาศัย และส่งสัญญาณไปสั่งเปิด/ปิดสวิตช์ไฟตามห้องต่างๆ ที่มีคนหรือไม่มีคนอยู่ อุปกรณ์วัดสัญญาณชีพของผู้ป่วย ผู้สูงอายุและส่งข้อมูลไปยังบุคลากรทางการแพทย์ หรือส่งข้อความเรียกหน่วยกู้ชีพหรือรถฉุกเฉิน เป็นต้น

นอกจากนี้ IoT จะเปลี่ยนรูปแบบและกระบวนการผลิตในภาคอุตสาหกรรมไปสู่ยุคใหม่หรือที่เรียกว่า Industry 4.0 ที่จะอาศัยการเชื่อมต่อสื่อสารและทำงานร่วมกันระหว่างเครื่องจักรมนุษย์ และข้อมูล เพื่อเพิ่มอำนาจในการตัดสินใจที่รวดเร็วและความถูกต้องแม่นยำสูง โดยที่ข้อมูลทั้งหลายที่เก็บจากเซ็นเซอร์ที่ใช้ตรวจวัดตัวอุปกรณ์และสภาพแวดล้อมจะถูกนำมาวิเคราะห์ ให้ได้ผลลัพธ์เพื่อนำไปปรับปรุงกระบวนการผลิตได้อย่างทันที นอกจากการข้ามขีดจำกัดเรื่องเวลาแล้ว ระบบควบคุมหรือระบบวิเคราะห์ข้อมูล อาจไม่ได้ขึ้นอยู่กับเครื่องจักรแต่สามารถควบคุมสั่งการได้โดยไร้ขีดจำกัดเรื่องสถานที่

Internet of Things (IoT) เทคโนโลยีอินเทอร์เน็ตที่เชื่อมต่ออุปกรณ์และ เครื่องมือต่างๆ เช่น โทรศัพท์มือถือ รถยนต์ ตู้เย็น โทรทัศน์ และอื่นๆ เข้าไว้ด้วยกัน โดยเครื่องมือต่างๆ จะสามารถ เชื่อมโยงและสื่อสารกันได้โดยผ่านระบบอินเทอร์เน็ต ในอนาคต ผู้บริโภคทั่วไปจะเริ่มคุ้นเคยกับ เทคโนโลยีที่ทำให้ สามารถควบคุมสิ่งของต่างๆ ทั้งจากในบ้าน และสำนักงานหรือจากที่ไหนก็ได้ เช่น การควบคุมอุณหภูมิภายในบ้าน การเปิดปิดไฟ ไปจนถึงการสั่งให้เครื่องทำกาแฟ เริ่มต้มน้ กาแฟ แต่อย่างไรก็ตาม ยังมีเทคโนโลยีอื่นๆ ที่จำเป็นจะต้องถูกพัฒนาขึ้นก่อนที่ IoT จะเป็นความ จริ่งขึ้นมา เช่น ระบบตรวจจับต่างๆ (Sensors) รูปแบบการ เชื่อมต่อระหว่างอุปกรณ์ และระบบที่ ผังตัวอยู่ในคอมพิวเตอร์

2.10 NETPIE (Network Platform for Internet of Everything)

NETPIE เป็นแพลตฟอร์มบริการสำหรับการพัฒนา IoT Solution ซึ่งสามารถทำให้ สามารถเชื่อมต่อสิ่งต่างๆ ได้อย่างง่ายดาย และ ไร้ความกังวลในเรื่องต่างๆ เช่น การบริหารจัดการ การเชื่อมต่อ การยืนยันตัวตนผู้ใช้และสิ่งของ การจำกัดการเข้าถึงบริการ และเรื่องเกี่ยวกับ System Admin ทั่วไป ดังนั้นสามารถทุ่มเทเวลาที่เหลือเพื่อสร้างสรรค์ผลิตภัณฑ์ IoT ได้อย่าง เต็มที่

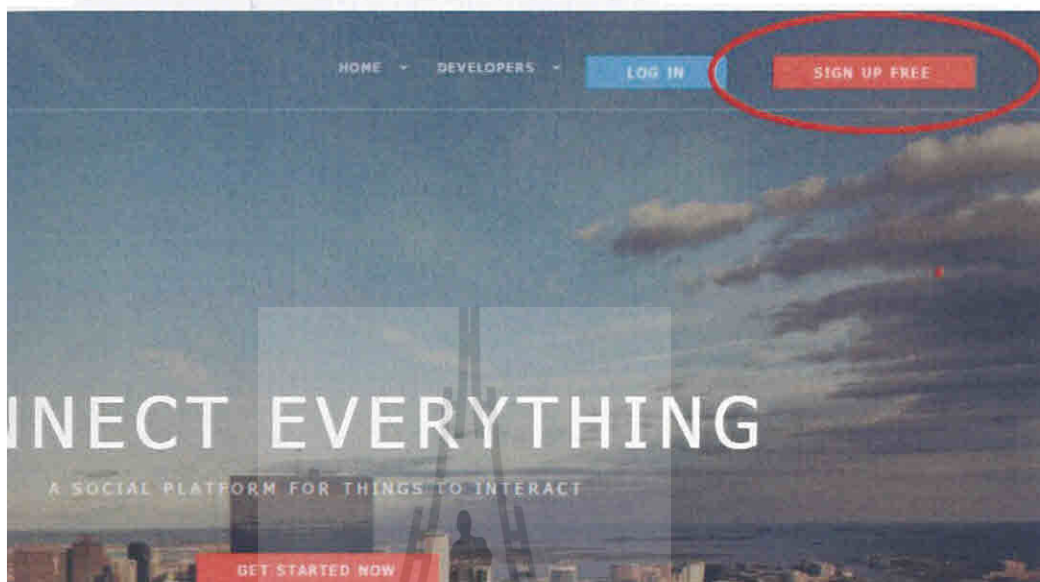
NETPIE ไม่ใช่แค่ MQTT¹ broker (Message Queue Telemetry Transport broker) แต่เรา สร้างการสื่อสารแบบ Publish-Subscribe ในรูปแบบเฉพาะที่ทำให้อุปกรณ์ IoT หรือ “Things” สามารถติดต่อสื่อสารได้ตอบกันได้อย่างเสรี การสื่อสารแบบนี้สามารถรองรับได้ทั้งโปรโตคอล MQTT และ HTTP REST การสร้างให้ “Things” สามารถสื่อสารกันได้ด้วยโปรโตคอล MQTT คุณเพียงแค่ว่า library ของ NETPIE ที่เรียกกันว่า Microgears ในการเชื่อมต่อ ซึ่ง Microgears เหล่านี้เป็น open-source ที่สามารถดาวน์โหลดได้ฟรีที่ GitHub²

¹ ที่มา : <https://github.com/cmmakerclub/netpie-manager>

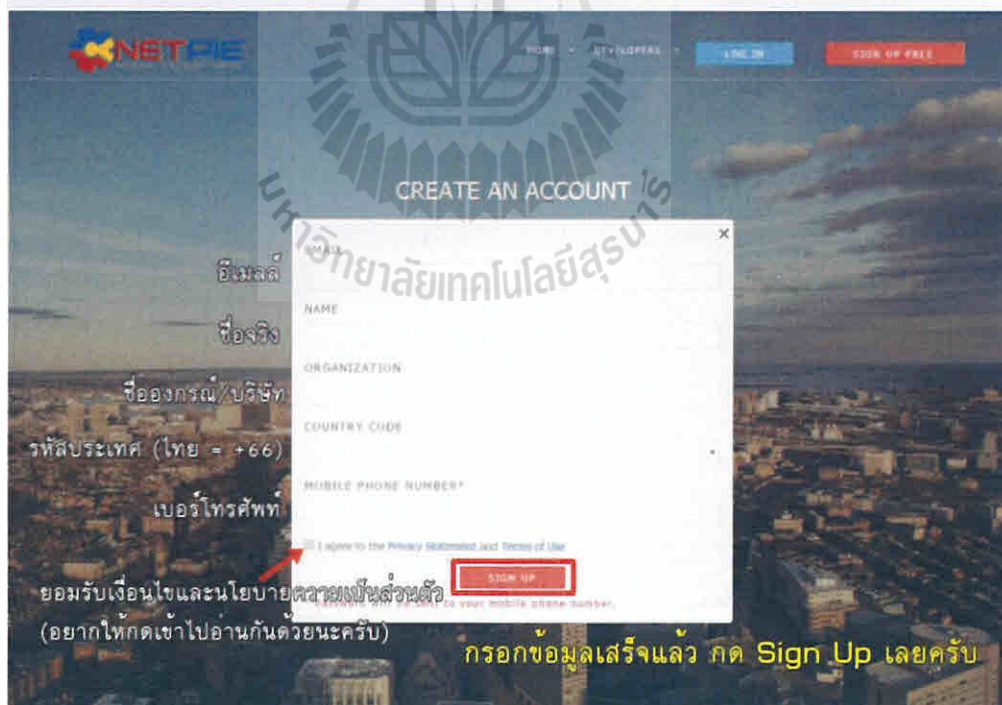
² สามารถดาวน์โหลดได้ฟรีที่ : <https://gist.github.com/>

2.10.1 สมัคร NETPIE และสร้าง Application

สมัครสมาชิกเว็บ NETPIE

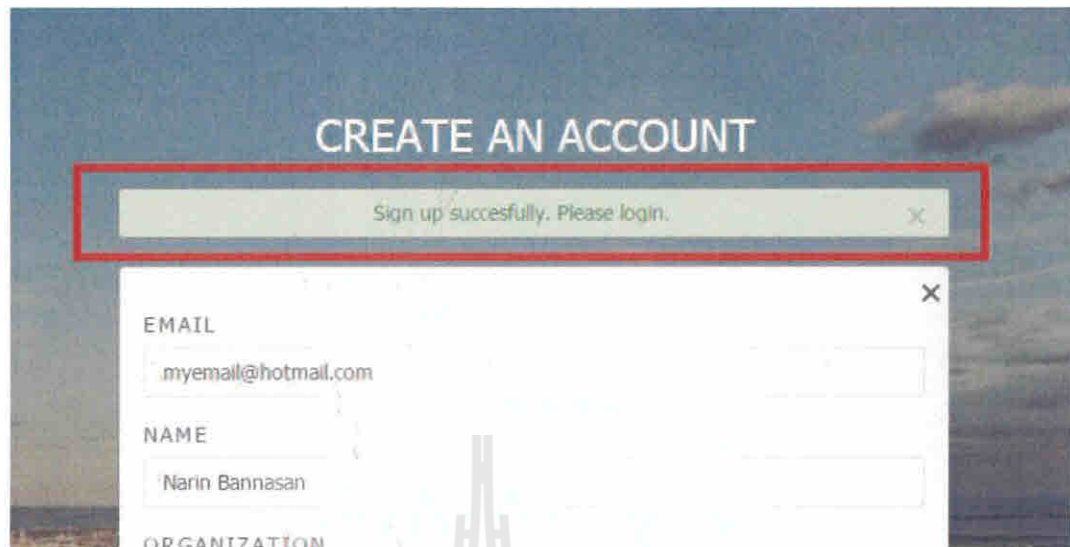


รูปที่ 2.19 เข้าไปยังเว็บไซต์ NETPIE จากนั้นคลิกที่ SIGN UP FREE



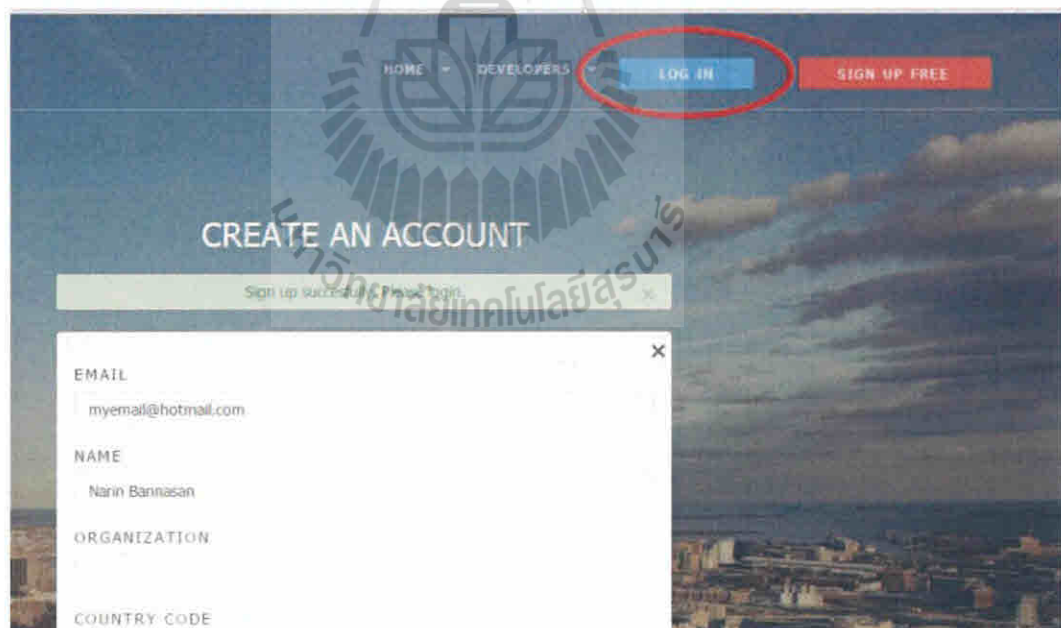
กรอกข้อมูลเสร็จแล้ว กด Sign Up เลยครับ

รูปที่ 2.20 หน้าลงทะเบียนใช้งาน NETPIE



รูปที่ 2.21 ใส่ข้อมูลเรียบร้อยและกด SIGN UP

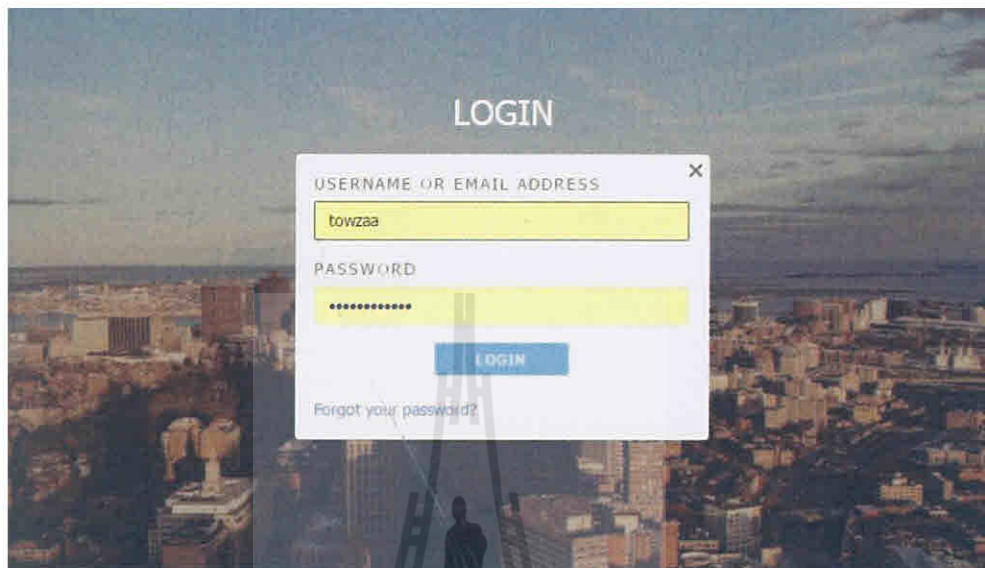
รอรับข้อความยืนยันในโทรศัพท์จาก NETPIE โดยจะส่งข้อความเข้ามายังเบอร์โทรศัพท์ที่ได้ลงทะเบียนไว้



รูปที่ 2.22 เข้าหน้า LOG IN เพื่อทำการเข้าระบบ

ให้ทำการกรอก ID ที่เราสมัคร และ Password ที่ได้จากข้อความในโทรศัพท์ลงไปแล้ว

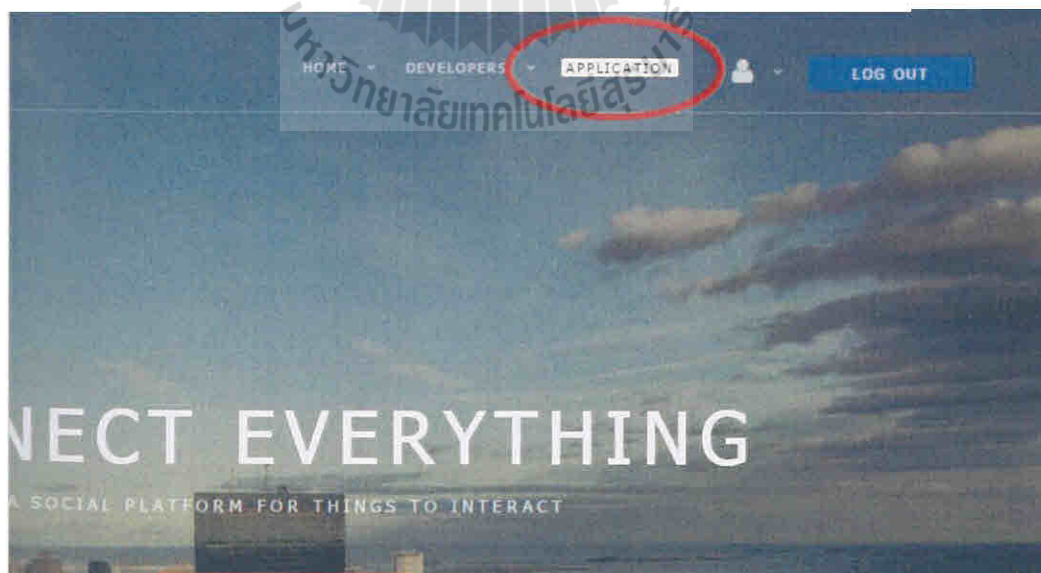
กด LOG IN



รูปที่ 2.23 LOG IN เข้ามาในระบบ

Application, Gear , Key

เมื่อทำการ Reset Password เสร็จแล้ว จะเข้าสู่หน้าหลักก็สามารถใช้งาน NETPIE ได้อย่างสมบูรณ์

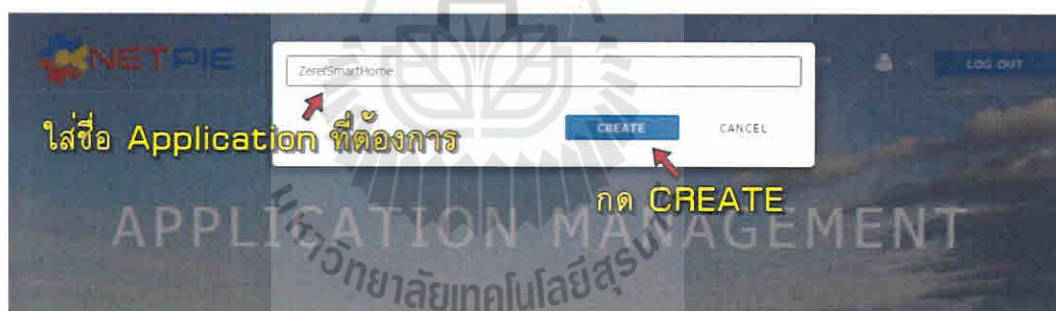


รูปที่ 2.24 หน้าหลักใช้งาน NETPIE

หน้า APPLICATION MANAGEMENT (การจัดการ APPLICATION) ก็จะแสดงข้อมูลจำนวน Application ที่เราสร้างไว้ และจำนวนอุปกรณ์ (Gear หรือ Things) ในระบบของเรา

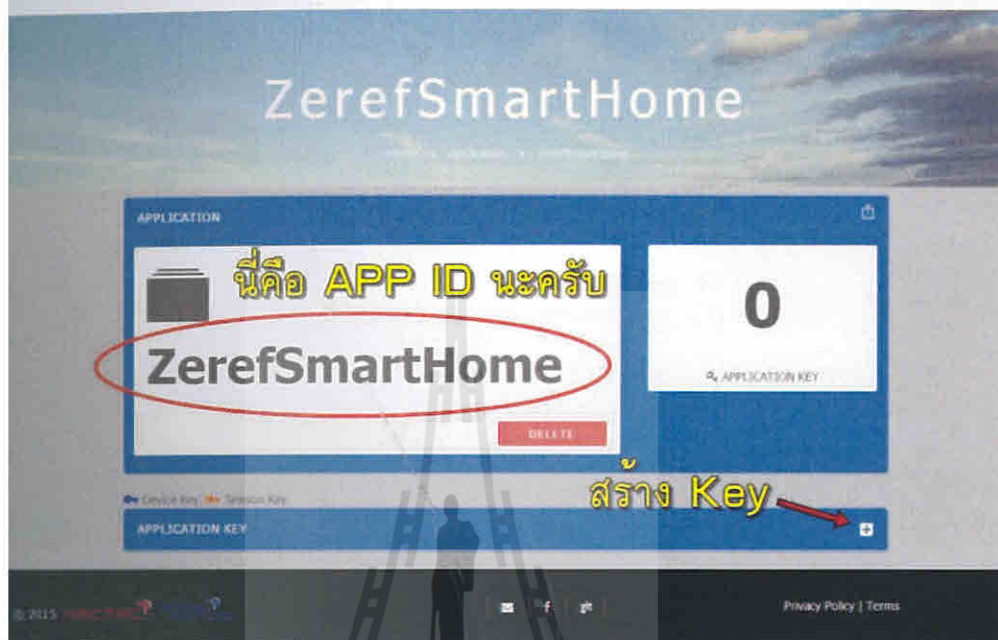


รูปที่ 2.25 เริ่มสร้าง Application โดยกดตรงปุ่ม [+]



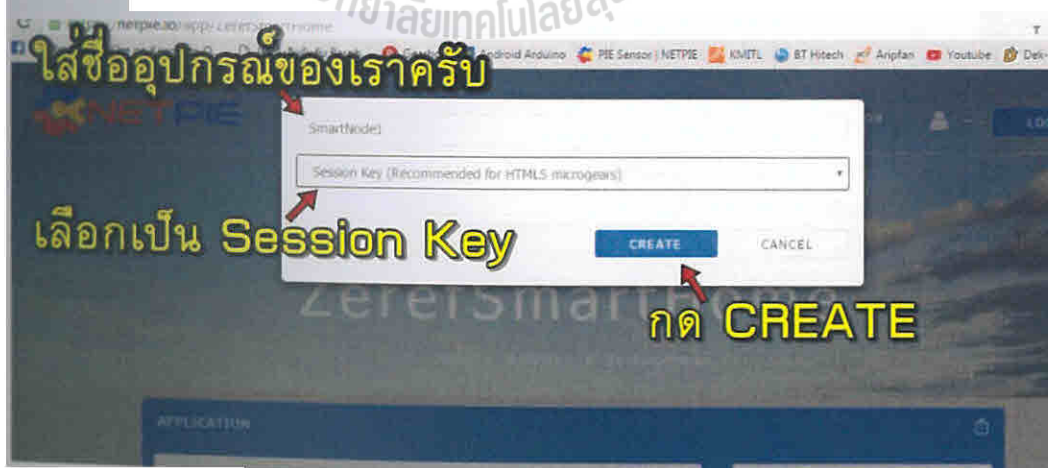
รูปที่ 2.26 ใส่ชื่อ Application

จากนั้นก็จะได้ Application ของเราโดยที่ชื่อที่เราตั้งเข้าไป จะเป็น APP ID ของ Application (จะได้นำไปใส่ในโค้ด) แล้วก็จะมาสร้าง Key กัน กดที่ปุ่ม [+] ตามในรูป

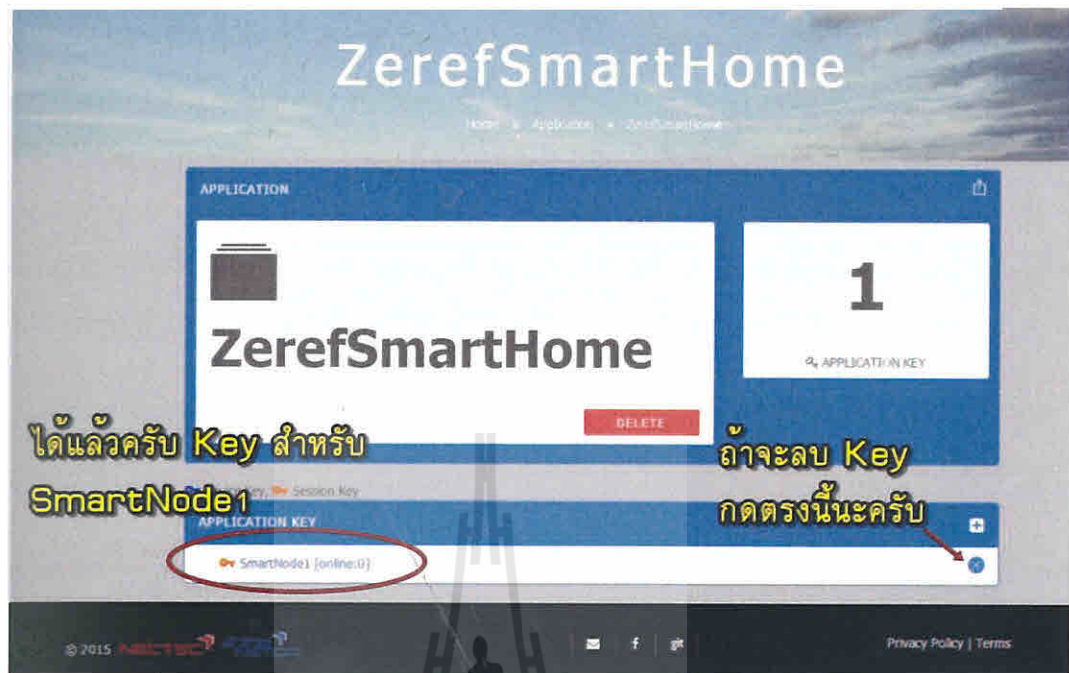


รูปที่ 2.27 Application ชื่อที่ตั้งเข้าไป

จากนั้นก็ปรากฏหน้าต่างสำหรับการสร้าง Key ขึ้นมา ก็ให้ทำการตั้งชื่อให้อุปกรณ์ของเราเช่น ตั้งชื่อเป็น SmartNode1 แล้วก็ทำการเลือก Type ให้เป็น Session Key จากนั้นก็กด CREATE

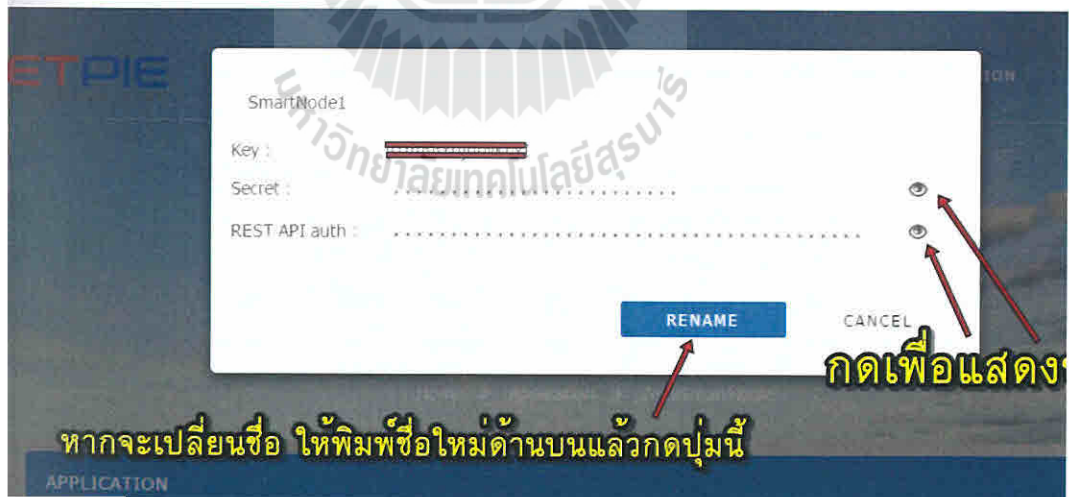


รูปที่ 2.28 การตั้งชื่อและเลือกชนิด Application Key



รูปที่ 2.29 Key สำหรับไว้ให้อุปกรณ์ของเราเอาไปใช้ในการสื่อสาร

ให้กดที่ Key ที่สร้างไว้ ก็จะปรากฏข้อมูล Key ขึ้นมาหากจะเปลี่ยนชื่อของอุปกรณ์ ก็พิมพ์ชื่อใหม่ด้านบน แล้วก็กดปุ่ม RENAME

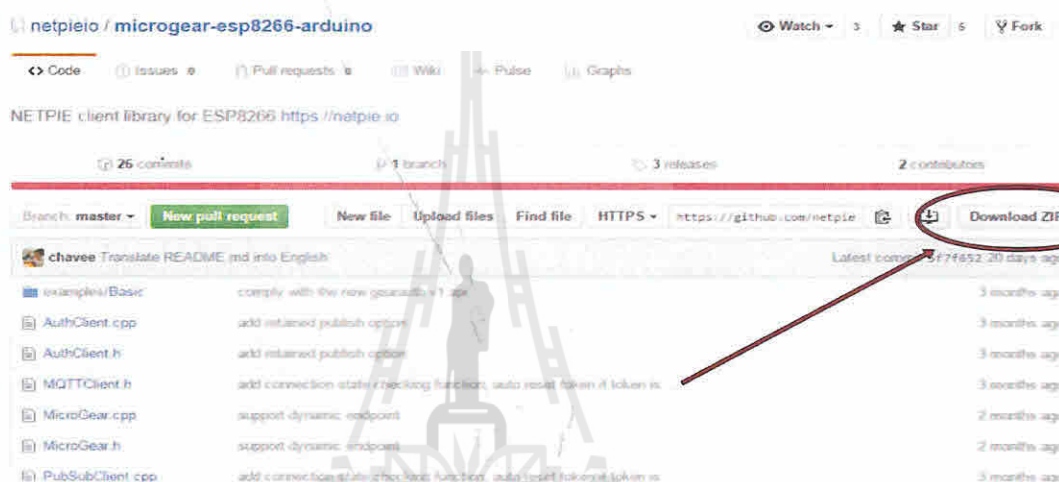


รูปที่ 2.30 แสดง KEY และ SECRET เพื่อใช้ในการเชื่อมต่อกับ NETPIE

2.10.2 การติดตั้ง Microgear Library ของ Arduino

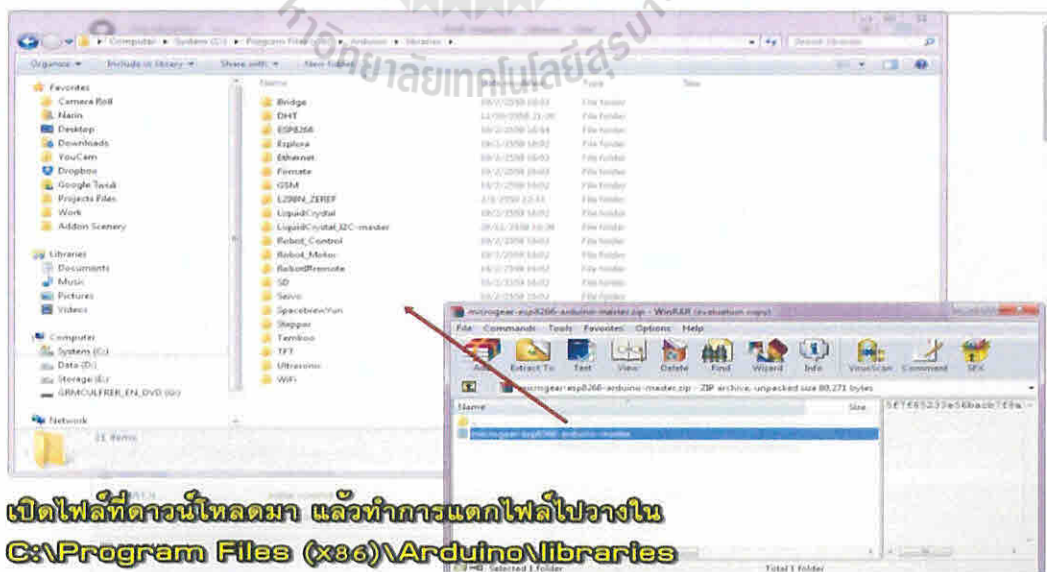
Microgear Library (สำหรับ Arduino)

ในโค้ดของ NETPIE นั้นจะมีการ Include Library ซึ่งจะเป็นการเพิ่มคำสั่งพิเศษเฉพาะของ NETPIE เข้าไปเพื่อให้ Arduino สามารถใช้คำสั่งพิเศษเฉพาะของ NETPIE ได้ ดังนั้นเราจึงต้องทำการติดตั้ง Library ของ Microgear ก่อน ดาวน์โหลด Microgear Library (สำหรับ Arduino) สามารถดาวน์โหลดได้ฟรี: <https://github.com/netpieio/microgear-esp8266-arduino>



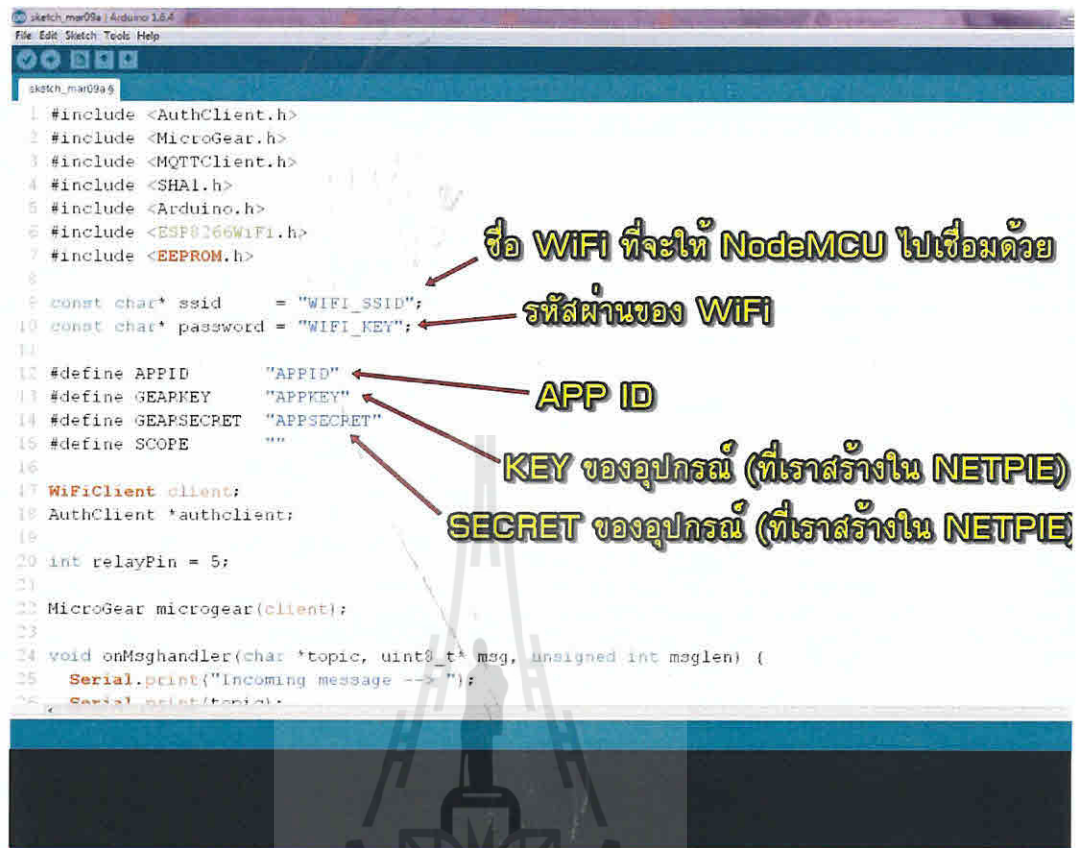
รูปที่ 2.31 เข้าไปแล้วกดที่ Download ZIP

เมื่อดาวน์โหลดเสร็จแล้วก็ทำการเปิดไฟล์ แล้วทำการแตกไฟล์เอาโฟลเดอร์ Microgear - ESP8266- Arduino -Master ไปวางไว้ในโฟลเดอร์ Libraries ของ Arduino

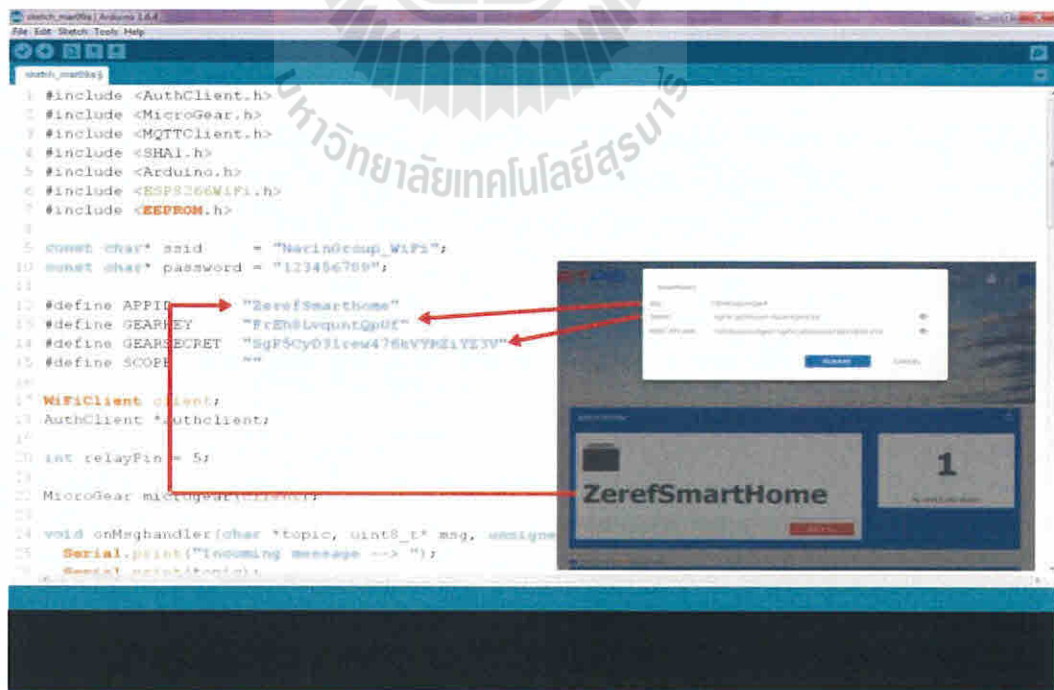


เปิดไฟล์ที่ดาวน์โหลดมา แล้วทำการแตกไฟล์ไปวางใน
C:\Program Files (x86)\Arduino\libraries

รูปที่ 2.32 ทำการแตกไฟล์เอาโฟลเดอร์ Microgear -ESP8266- Arduino -Master

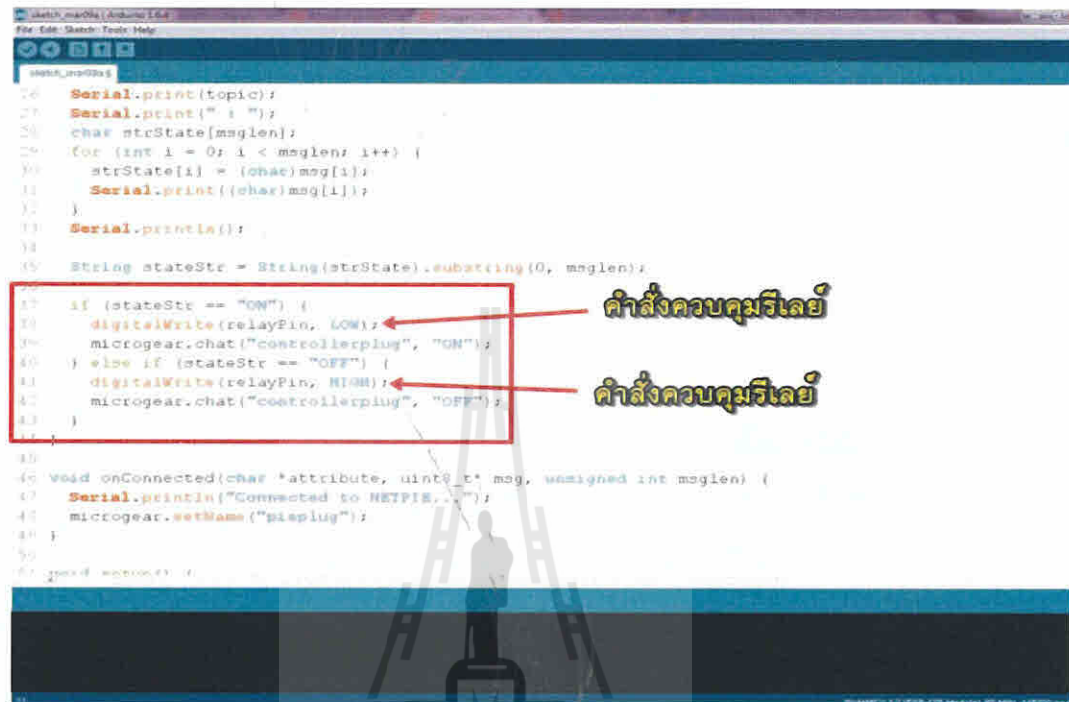


รูปที่ 2.33 ทำการแก้ไขกันเพื่อให้สามารถใช้งานได้กับอุปกรณ์



รูปที่ 2.34 กรอกข้อมูล Key ที่สร้างไว้

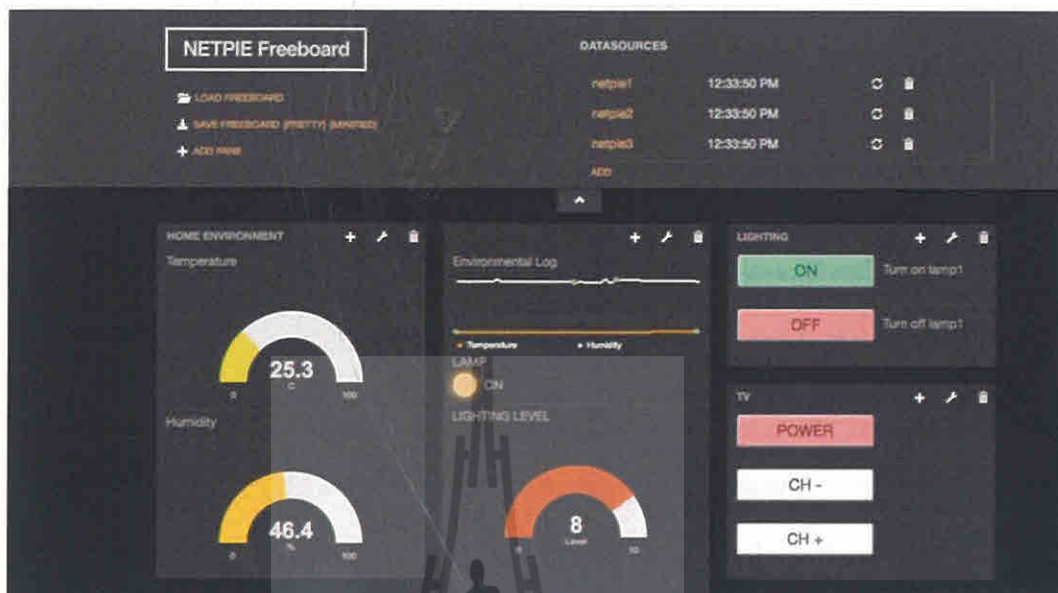
เราก็จะไปแก้ไขชุดคำสั่ง ให้เลื่อนลงไป ประมาณบรรทัดที่ 37 ก็จะพบกับชุดคำสั่งที่ควบคุมรีเลย์ให้เปิด,ปิด



รูปที่ 2.35 ชุดคำสั่งที่ควบคุมรีเลย์ให้เปิด,ปิด

ทำการกด Compile เพื่อตรวจสอบว่าโค้ดของเรามีปัญหาที่จุดไหนหรือเปล่าก็ให้ทำการตรวจสอบว่าเราพิมพ์ถูกไหม มีปีกกาตัวไหนหายหรือเปล่า หากไม่มีปัญหาอะไรก็ กด Upload (กด Ctrl+U) ได้เลย (อย่าลืม Save ไฟล์) สำหรับการเตรียมโค้ดฝัง Arduino ของ NodeMCU V2

2.10.3 การสร้าง Datasource บน Freeboard



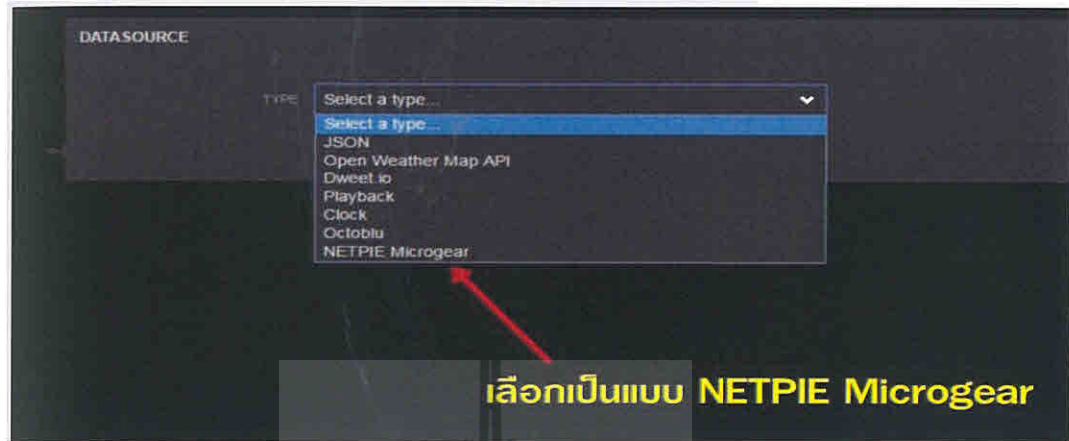
รูปที่ 2.36 Dashboard ของ NETPIE Freeboard

ขั้นตอนที่ 1 ให้ทำการเปิด NETPIE Freeboard ขึ้นมาก่อน แล้วดูตรงส่วนบนขวา จะเห็น ส่วนของ Datasource อยู่ให้ทำการกดที่ปุ่ม ADD



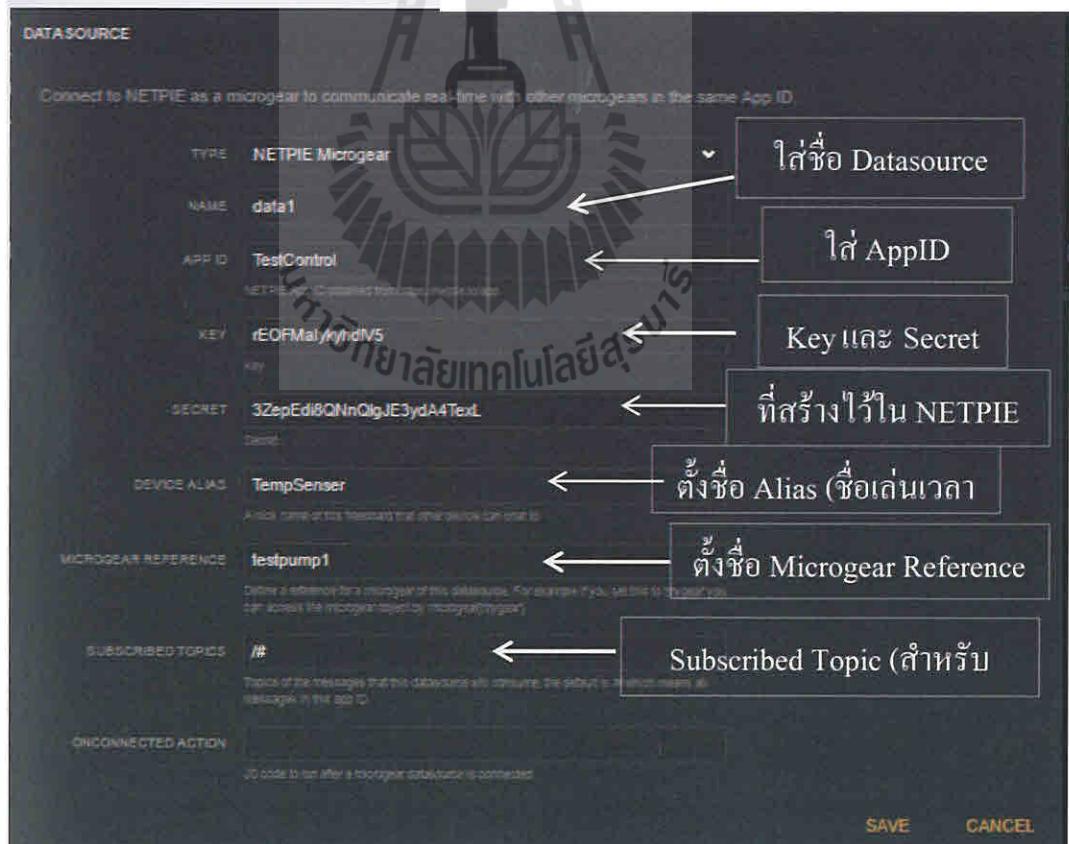
รูปที่ 2.37 การเพิ่ม Datasource

ขั้นตอนที่ 2 ทำการเลือก Type เป็นแบบ NETPIE Microgear



รูปที่ 2.38 เลือกเลือกประเภท NETPIE Microgear

ขั้นตอนที่ 3 การกำหนดค่าให้ Datasource ที่เราจะสร้าง ก็จะต้องใส่ข้อมูลต่างๆดังนี้ เมื่อกรอกข้อมูลเรียบร้อยแล้ว ก็ให้กด Save



รูปที่ 2.39 กำหนดค่าให้ Datasource

ขั้นตอนที่ 4 Datasource ที่เราสร้างขึ้นปรากฏในรายชื่อ Datasource



รูปที่ 2.40 Datasource ที่สร้างขึ้น

2.10.4 การเขียนโปรแกรมลงบอร์ด WeMos D1 wifi ESP8266

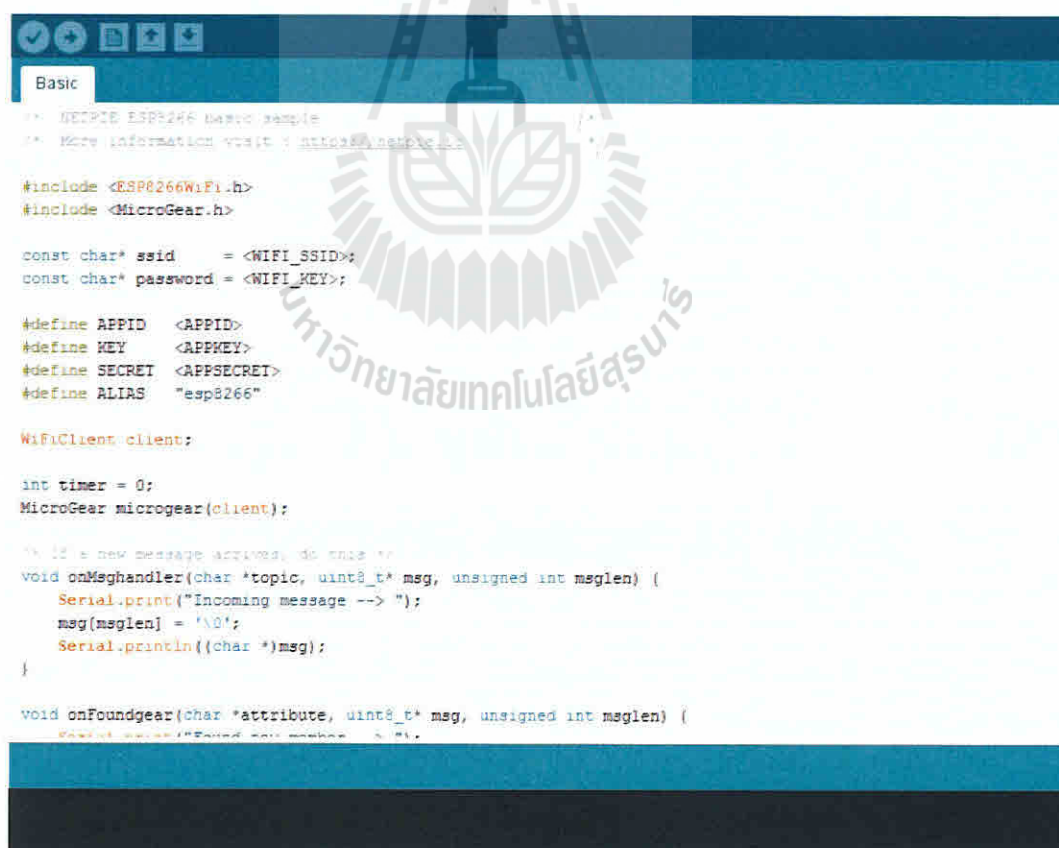
ในการเขียนโปรแกรมลงบอร์ด WeMos D1 wifi ESP8266 ด้วยโปรแกรม ArduinoIDE ซึ่งเขียนโค้ดด้วยภาษาซี โปรแกรมของ Arduino แบ่งได้ เป็นสองส่วนคือ void setup() และ void loop() โดยฟังก์ชัน setup() เมื่อโปรแกรมทำงานจะทำคำสั่งของฟังก์ชันนี้เพียงครั้งเดียวใช้ในการกำหนดค่าเริ่มต้นของการทำงาน ส่วนฟังก์ชัน loop() เป็นส่วนทำงาน โปรแกรมจะทำคำสั่งในฟังก์ชันนี้ต่อเนื่องตลอดเวลา โดยปกติใช้ กำหนดโหมดการทำงานของขาต่างๆ กำหนดการสื่อสารแบบอนุกรม ส่วนของ loop() เป็นโค้ดโปรแกรมที่ทำงาน เช่น อ่านค่าอินพุต ประมวลผลส่งงาน เอาต์พุต โดยส่วนกำหนดค่าเริ่มต้น เช่น ตัวแปรจะต้องเขียนที่ส่วนหัวของโปรแกรม ก่อนถึงตัวฟังก์ชันนอกจากนั้นยังต้อง คำนึงถึงตัวพิมพ์เล็ก-ใหญ่ ของตัวแปรและชื่อฟังก์ชันให้ถูกต้อง

2.10.5 การเขียนโปรแกรมลงบอร์ด WeMos D1 wifi ESP8266 พร้อมแสดงค่า

บน NETPIE

NETPIE เป็น Platform as a Service บริการการเชื่อมต่อข้อมูลและแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ต่างๆ ได้เหมาะสำหรับ Internet of Things ซึ่งรูปแบบการใช้งานจะเหมือนกับ MQTT พ่วงความสามารถด้าน Authentication และ Token เพิ่มความปลอดภัยในการสื่อสาร เราสามารถเชื่อมต่อกับ NETPIE ได้หลายช่องทาง ทั้งอุปกรณ์ IoT อย่าง Arduino, Raspberry Pi, ESP8266 Dev Kit, HTML5 หรือแม้กระทั่ง Node.js Application ดังนั้นหากใช้ ESP8266 Dev Kit ของค่ายใดอยู่ก็สามารถใช้งานได้เลย ดู Library ของอุปกรณ์ต่างๆ ได้ที่หน้าโครงการใน GitHub NETPIE ให้บริการโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)

ขั้นตอนที่ 1 Arduino IDE ใช้ NodeMCU Dev Kit กับ เพิ่ม Library MicroGear ESP8266 Arduino จากนั้น เปิดตัวอย่างโค้ดขึ้นมา



```

Basic
//> NETPIE ESP8266 Basic sample
//> More information visit : https://netpie.netpie.cc/

#include <ESP8266WiFi.h>
#include <MicroGear.h>

const char* ssid      = <WIFI_SSID>;
const char* password  = <WIFI_KEY>;

#define APPID    <APPID>
#define KEY      <APPKEY>
#define SECRET   <APPSECRET>
#define ALIAS    "esp8266"

WiFiClient client;

int timer = 0;
MicroGear microgear(client);

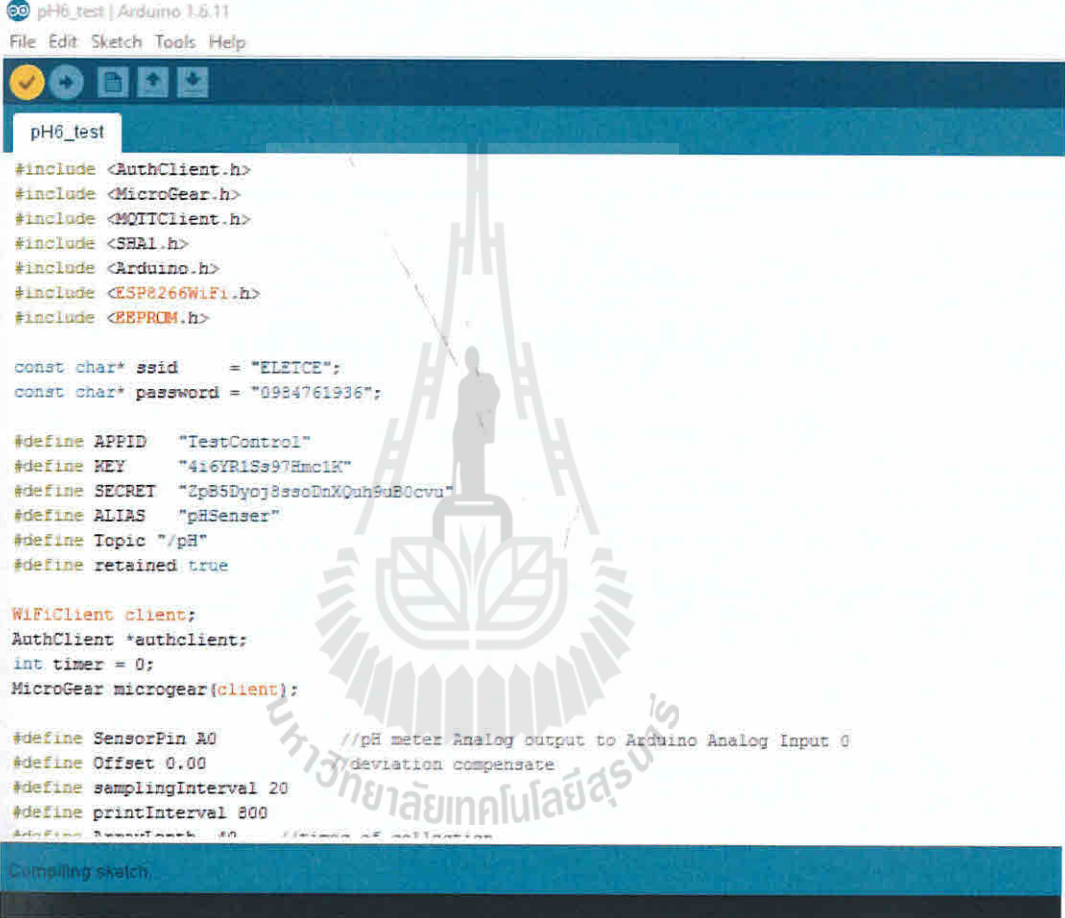
/* If a new message arrives, do this */
void onMsgHandler(char *topic, uint8_t* msg, unsigned int msglen) {
  Serial.print("Incoming message --> ");
  msg[msglen] = '\0';
  Serial.println((char *)msg);
}

void onFoundgear(char *attribute, uint8_t* msg, unsigned int msglen) {
  Serial.print("Found gear: ");
  Serial.println((char *)msg);
}

```

รูปที่ 2.43 โค้ด NETPIE ESP8266 basic sample

ขั้นตอนที่ 2 ทำการแก้ไข Ssid และ Password สำหรับเชื่อมต่อ wifi จากนั้นแก้ไข APPID, GEARKEY, GEARSECRET ให้ถูกต้องตาม Application ที่เราสร้างไว้ ตั้ง Upload โค้ด แล้วลองทดสอบดู ตัวอย่างนี้จะส่ง publish ค่าอุณหภูมิ (ค่าคงที่ไม่ได้อ่านจาก sensor) ไปยัง Topic ชื่อ /Temp กลับไปที่หน้า Key Management จะพบว่าการสร้าง MicroGear และมีการเชื่อมต่อเข้ามาแล้วกด Verify เพื่อตรวจสอบความถูกต้องของโค้ด



```

pH_test | Arduino 1.8.11
File Edit Sketch Tools Help

pH_test

#include <AuthClient.h>
#include <MicroGear.h>
#include <MQTTClient.h>
#include <SRSL.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <EEPROM.h>

const char* ssid = "ELEICE";
const char* password = "0984761936";

#define APPID "TestControl"
#define KEY "416YR1Se97Hmo1K"
#define SECRET "ZpB5Dypj8ssoDnXQun9uB0cvu"
#define ALIAS "pHSensor"
#define Topic "/pH"
#define retained true

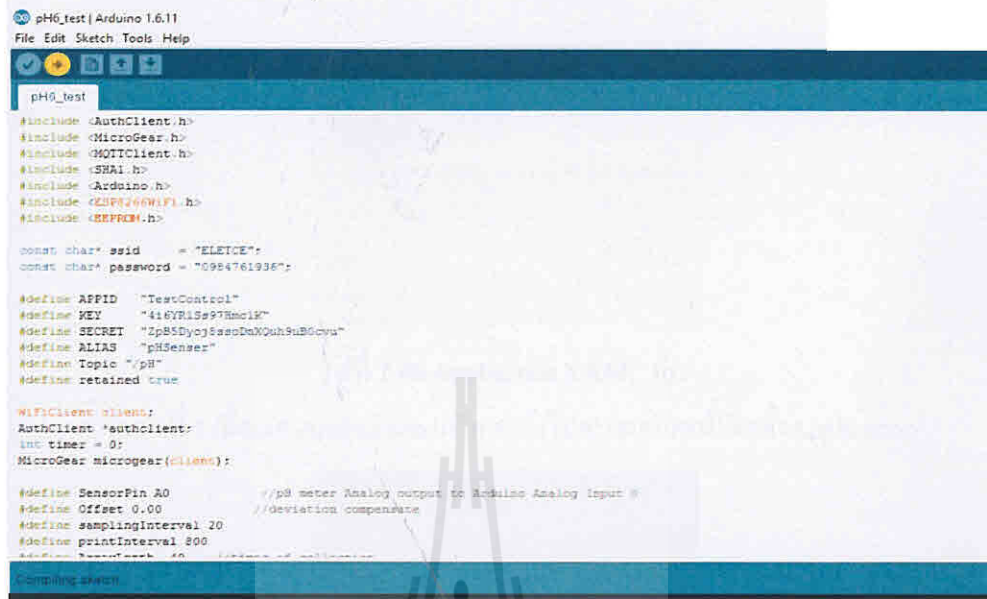
WiFiClient client;
AuthClient *authclient;
int timer = 0;
MicroGear microgear(client);

#define SensorPin A0 //pH meter Analog output to Arduino Analog Input 0
#define Offset 0.00 //deviation compensate
#define samplingInterval 20
#define printInterval 800
#define BuzzerPin 40 //pin of buzzer

Compiling sketch
  
```

รูปที่ 2.44 Verify ตรวจสอบความถูกต้องของโค้ด

ขั้นตอนที่ 3 ทำการ Upload โค้ดลง บอร์ด WeMos D1 wifi ESP8266



```

pH6_test | Arduino 1.6.11
File Edit Sketch Tools Help

pH6_test

#include <AuthClient.h>
#include <MicroGear.h>
#include <MQTTClient.h>
#include <SHA1.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <EEPROM.h>

const char* ssid = "ELETICE";
const char* password = "0984761936";

#define APPID "TestControl"
#define KEY "4t6VR1Se97HmclK"
#define SECRET "Zp8SDyog5ss0DnX0uh9nB6Gova"
#define ALIAS "pHSensor"
#define Topic "pH"
#define retained true

WiFiClient client;
AuthClient *authclient;
int timer = 0;
MicroGear microgear(client);

#define SensorPin A0 //pH meter Analog output to Arduino Analog Input *
#define Offset 0.00 //deviation compensate
#define samplingInterval 20
#define printInterval 800
#define Timeout 40

Compiling sketch
  
```

รูปที่ 2.45 Upload โค้ดลง บอร์ด WeMos D1 wifi ESP8266

2.11 APPLICATION KKMC IoT ควบคุมอุปกรณ์ IoT

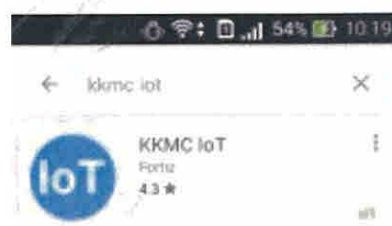
Application KKMC IoT ทำหน้าที่รับค่า มาแสดงตามที่เรากำหนด เราสามารถเพิ่มการใช้งานของ Application KKMC IoT ได้ด้วยการใส่เพียง APP ID และ REST KEY เท่านั้น แอปพลิเคชันก็พร้อมใช้งานทันที

2.11.1 วิธีใช้งาน KKMC IoT

ขั้นตอนที่ 1 ให้ติดตั้งแอปพลิเคชัน KKMC IoT ลงมือถือก่อน โดยสามารถ โหลดมาติดตั้งกันได้ผ่านทาง Google Play

โดยค้นหาด้วยคำว่า KKMC IoT หรือ สามารถดาวน์โหลดฟรีได้จาก

<https://play.google.com/store/apps/details?id=com.numberx.kkmc timer>



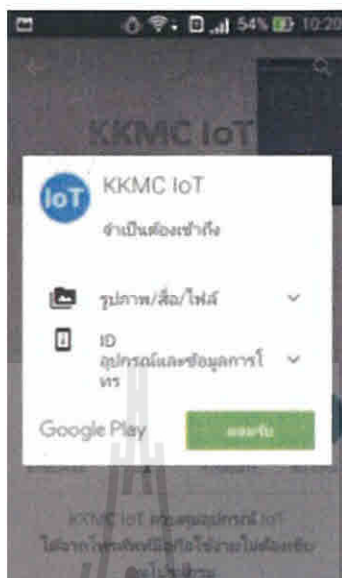
รูปที่ 2.46 Application KKMC IoT

ขั้นตอนที่ 2 เมื่อเจอ Application แล้วกดเข้าไปจะเจอกับหน้าติดตั้งApplication



รูปที่ 2.47 Application KKMC IoT

ขั้นตอนที่ 3 ให้กดติดตั้งได้เลย โดยตัวแอปพลิเคชันจะมีการขอการเข้าถึง ให้กด ขอมรับ



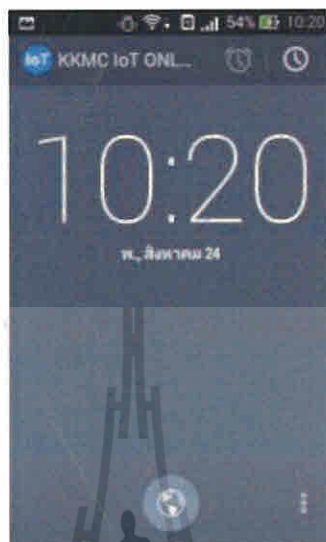
รูปที่ 2.48 ขอมรับ Application KKMC IoT

ขั้นตอนที่ 4 เมื่อโหลดและติดตั้งเสร็จสมบูรณ์ จะเจอกับหน้าตาแบบนี้ให้ กดเปิดได้เลย



รูปที่ 2.49 โหลดและติดตั้ง Application KKMC IoT เสร็จสมบูรณ์

ขั้นตอนที่ 5 เมื่อโหลดและติดตั้ง Application KKMC IoT เสร็จสมบูรณ์ เปิดเข้ามาจะเจอหน้าแรกของ Application



รูปที่ 2.50 หน้าแรกของ Application KKMC IoT

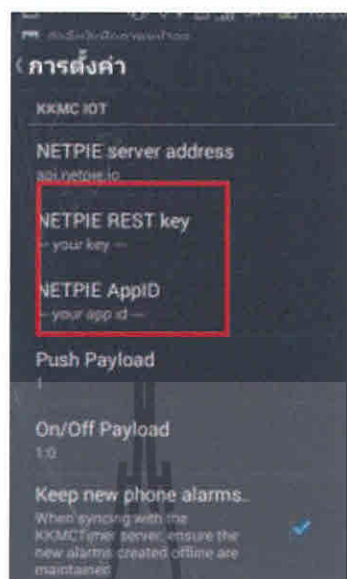
2.11.2 เซตอัพ APPLICATION เพื่อให้ใช้งานได้

ขั้นตอนที่ 1 ให้สังเกตที่ด้านซ้ายล่างจะเจอกับ จุด 3 จุด เมื่อกดเข้าไปจะพบกับเมนูตั้งค่า



รูปที่ 2.51 ตั้งค่า Application KKMC IoT

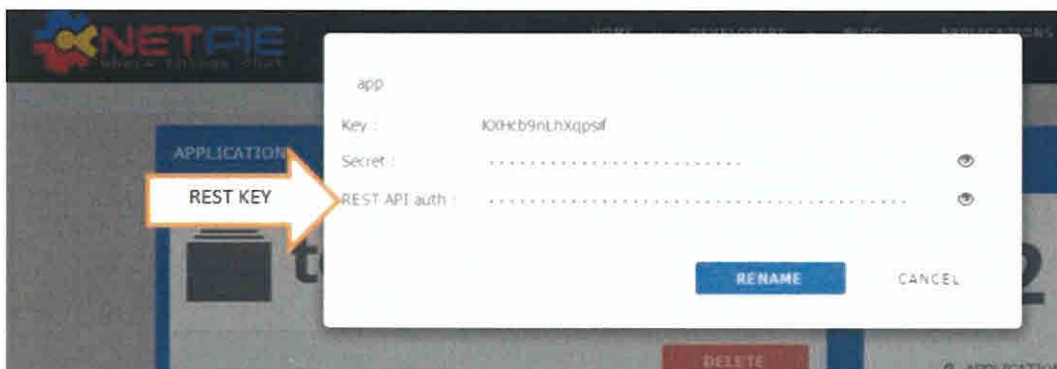
ขั้นตอนที่ 2 ใส่ APP ID และ REST KEY โดยต้องไปสร้าง Application ในเว็บ NETPIE



รูปที่ 2.52 แสดงการตั้งค่า Application KKMC IoT

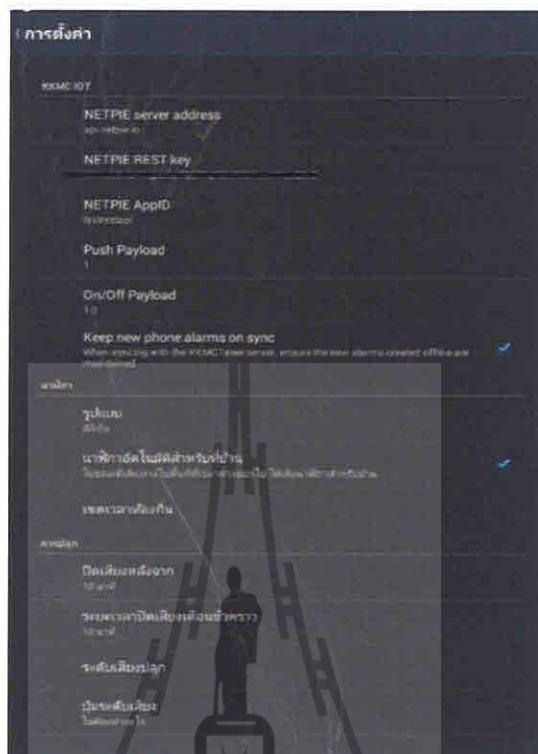


รูปที่ 2.53 APP ID ในเว็บ NETPIE



รูปที่ 2.54 REST KEY ในเว็บ NETPIE

ขั้นตอนที่ 3 นำค่า APP ID และ REST KEY กรอกใส่การตั้งค่าให้ถูกต้อง



รูปที่ 2.55 กรอกข้อมูล APP ID และ REST KEY เสร็จสมบูรณ์

2.11.3 การเพิ่ม Widget เพื่อแสดง ข้อมูลหรือสั่งให้ปิดเปิด อุปกรณ์

ขั้นตอนที่ 1 กดรูป + ด้านบนเพื่อเรียกหน้าต่าง Add



รูปที่ 2.56 แสดงปุ่มการเพิ่ม Widget

ขั้นตอนที่ 2 จากนั้นจะเป็นการเรียกหน้าต่าง Add new Topic ขึ้นมา

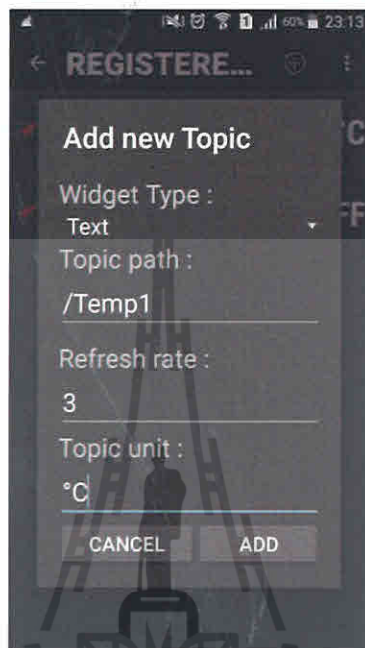
รูปที่ 2.57 แสดงหน้าต่าง Add new Topic

ขั้นตอนที่ 3 Widget type สามารถเลือกได้ว่าจะต้องการใช้งานอะไร มีให้เลือกใช้ 3 รูปแบบ คือ

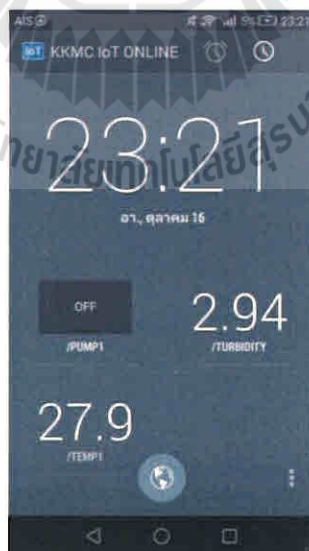
- 1.Text คือการการแสดงตัวหนังสือ
2. Push Button คือ ปุ่ม กดติดปล่อยดับ
- 3.On/Off Switch คือ สวิตช์ เปิดปิด

รูปที่ 2.58 แสดงรูปแบบ Widget type

ขั้นตอนที่ 4 เลือก Widget Type เป็น Ttext และ กำหนด Topic ในการรับข้อมูลอุณหภูมิ คือ /Tem หลังจากกด Add ก็จะเริ่มรับค่าเข้ามาจากตัวส่ง



รูปที่ 2.59 กำหนด Topic ในการรับข้อมูลอุณหภูมิ



รูปที่ 2.60 หน้าหลักของ Application แสดงค่าพารามิเตอร์

บทที่ 3

การแสดงผลข้อมูลผ่าน Application

3.1 กล่าวนำ

ในบทที่ผ่านมาจะกล่าวถึงวิธีการได้มาของข้อมูลและการแสดงค่าพารามิเตอร์ต่างๆของข้อมูล ซึ่งทำให้ผู้ใช้งานสามารถนำข้อมูลมาไปแก้ไขปัญหาหรือสามารถประยุกต์ใช้กับการใช้งานอื่นๆตามที่ต้องการ ด้วยความสามารถของ Cloud Platform บน NETPIE (Network Platform For Internet Of Everything) และ Android Application KKMCIoT บน Smartphone ที่ทำหน้าที่เป็นตัวแสดงผลและควบคุมการทำงาน ทำให้สามารถเข้าถึงข้อมูลได้ทุกสถานที่โดยผ่านอินเทอร์เน็ตได้ และยังสามารถตรวจสอบค่าอุณหภูมิ ค่าความเป็นกรด-ด่าง(pH) ค่าความชื้นในน้ำว่าค่าสุดท้ายเป็นเท่าใด โดยข้อมูลจะเรียกจาก Cloud Platform บน NETPIE (Network Platform For Internet Of Everything) นำมาแสดงผลใน Application ทำให้สะดวกต่อการดูภาพรวมของข้อมูล ซึ่งบทนี้จะกล่าวถึงเทคนิคการแสดงผลข้อมูลผ่าน Application

3.2 NETPIE

NETPIE เป็นแพลตฟอร์มบริการสำหรับการพัฒนา IoT Solution ซึ่งสามารถทำให้ผู้ใช้งานสามารถเชื่อมต่อสิ่งต่างๆได้อย่างง่ายดาย และไร้ความกังวลในเรื่องต่างๆ เช่น การบริหารจัดการ การเชื่อมต่อ การยืนยันตัวตนผู้ใช้และสิ่งของ การจำกัดการเข้าถึง

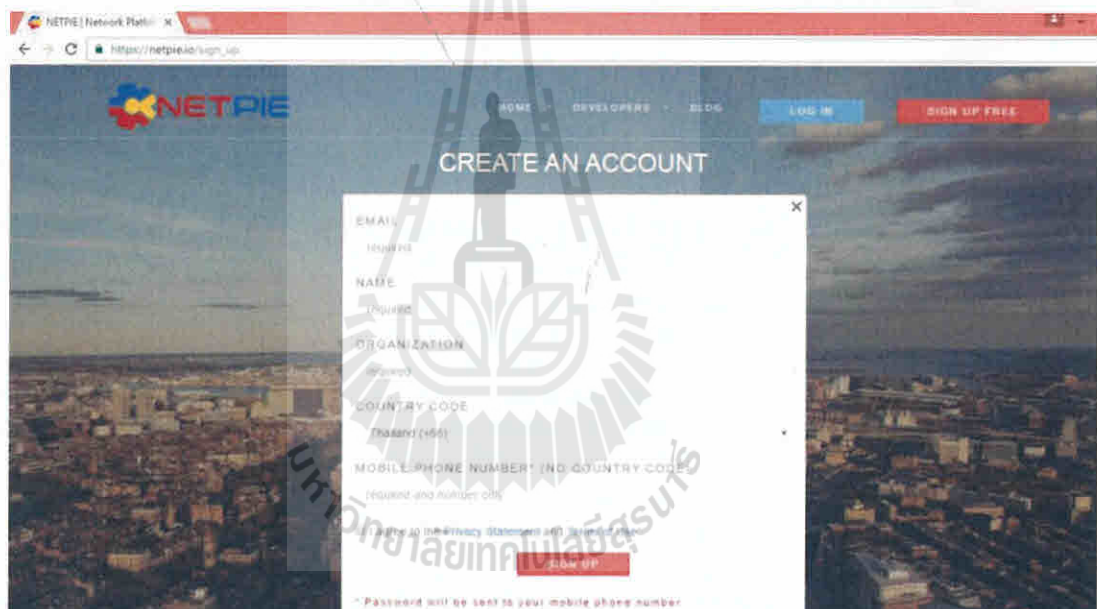
NETPIE ไม่ใช่แค่ MQTT broker แต่เราสร้างการสื่อสารแบบ publish-subscribe ในรูปแบบเฉพาะที่ทำให้อุปกรณ์ IoT หรือ “Things” สามารถติดต่อสื่อสารโต้ตอบกันได้อย่างเสรี การสื่อสารแบบนี้สามารถรองรับได้ทั้งโปรโตคอล MQTT และ HTTP REST การสร้างให้ “Things” สามารถสื่อสารกันได้ด้วยโปรโตคอล MQTT เพียงแค่ใช้ library ของ NETPIE ที่เรียกว่า Microgears

3.2.1 การใช้งาน NETPIE

ขั้นตอนที่ 1 สมัครสมาชิกเว็บ NETPIE¹



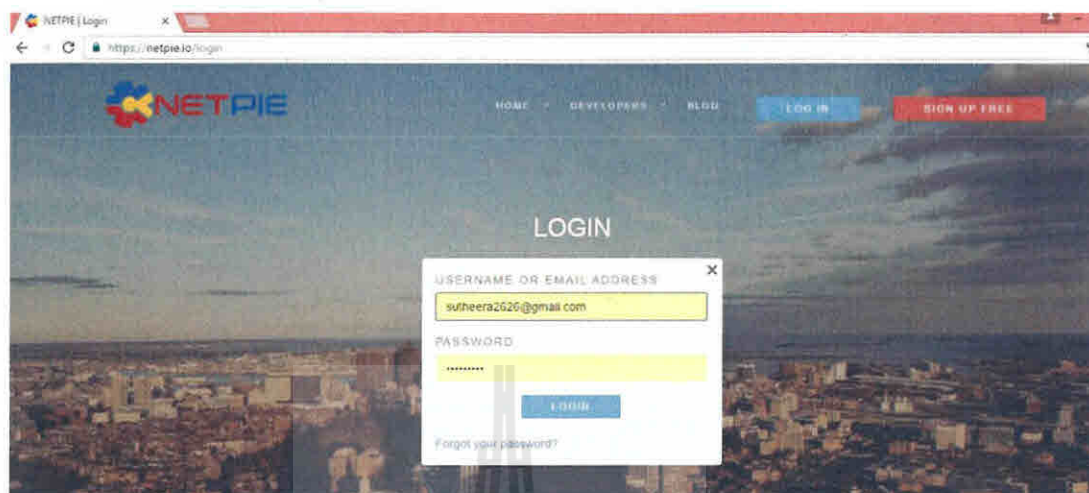
รูปที่ 3.1 สมัครสมาชิกเว็บ NETPIE



รูปที่ 3.2 กรอกข้อมูลสำหรับการสมัครให้เรียบร้อย แล้วคลิก SIGN UP

¹ สมัครสมาชิกได้จากเว็บ <https://netpie.io/>

ขั้นตอนที่ 2 เมื่อสมัครสมาชิกเสร็จ LOGIN เข้ามาในระบบ



รูปที่ 3.3 LOGIN เข้ามาในระบบ

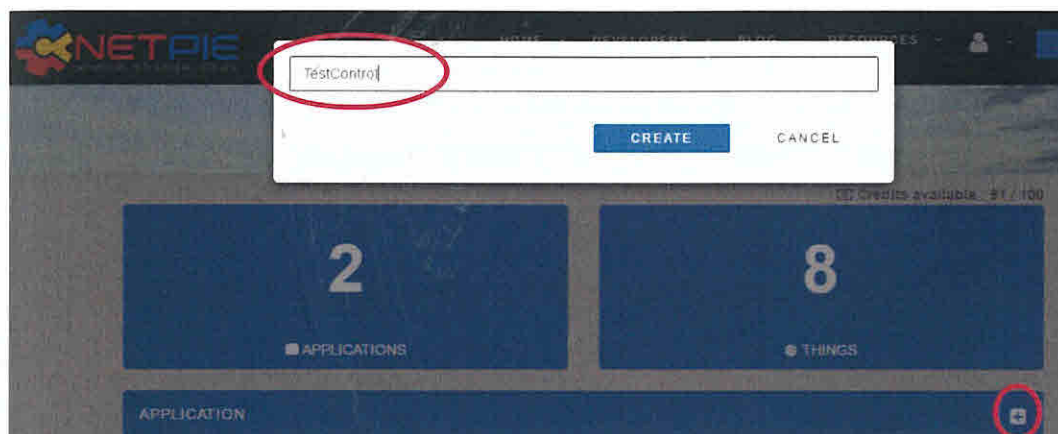
3.2.2 Application, Gear , Key

ขั้นตอนที่ 1 เมื่อทำการ Reset Password เสร็จแล้ว จะเข้าสู่หน้าหลักก็สามารถใช้งาน NETPIE ได้อย่างสมบูรณ์และสร้าง Application ในระบบ NETPIE



รูปที่ 3.4 หน้าหลักใช้งาน NETPIE

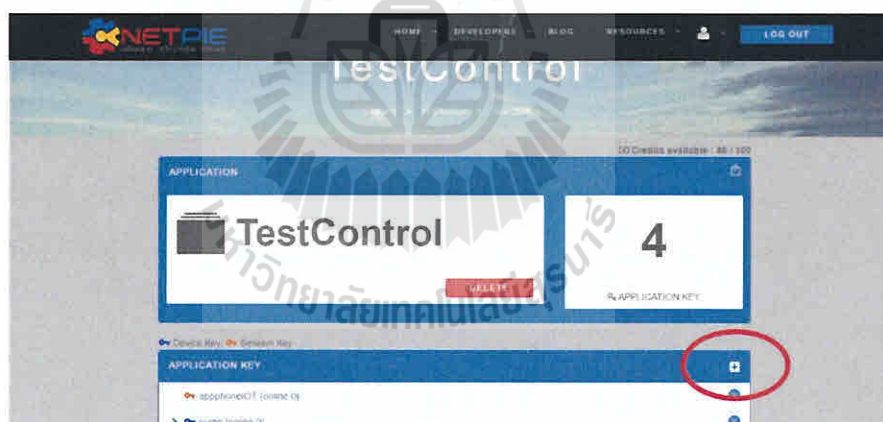
ขั้นตอนที่ 2 สร้าง Application ID (คลิกที่เครื่องหมาย + เพื่อสร้าง ID) ซึ่ง ID นี้เปรียบเสมือนชื่อของ project ที่พัฒนาอยู่ ต้องเป็นชื่อที่ไม่ซ้ำกับบัญชี NETPIE ในที่นี้สร้าง Application ID ชื่อว่า “TestControl”



รูปที่ 3.5 สร้าง Application ID

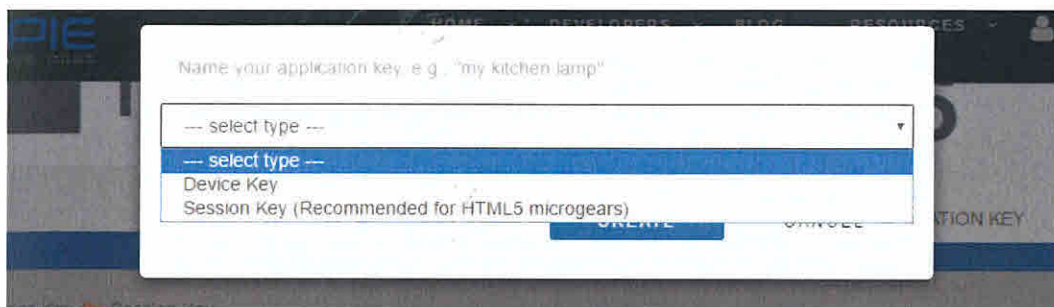
ขั้นตอนที่ 3 การจัดการแอปพลิเคชัน หน้า Application Management ก็จะแสดงข้อมูลจำนวน Application ที่สร้างไว้ และจำนวนอุปกรณ์ (Gear หรือ Things) ในระบบ

คลิกที่เครื่องหมาย + เพื่อสร้าง Application key ตั้งชื่อ Application key ซึ่งอาจจะเป็นคำบรรยายที่สื่อถึง Thing สามารถมี ช่องว่าง หรืออักขระพิเศษได้ และชื่อสามารถซ้ำได้ เช่น "a light sensor in my kitchen" หรือ "a lamp in my bedroom"

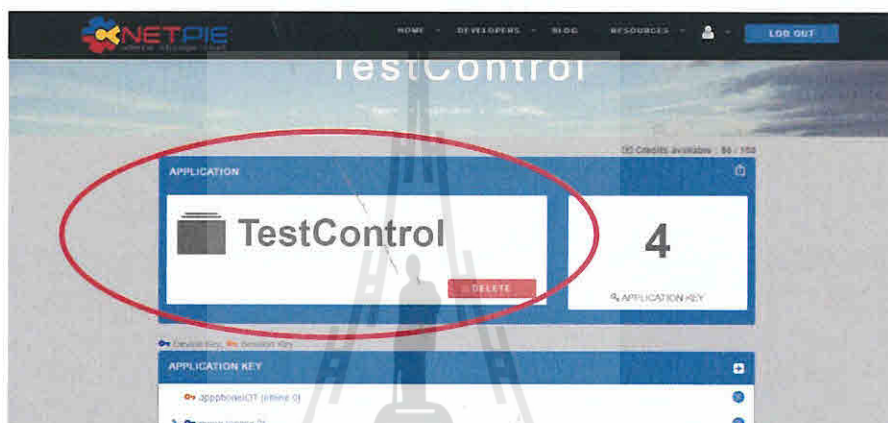


รูปที่ 3.6 เริ่มสร้าง Application โดยกดตรงปุ่ม [+]

ขั้นตอนที่ 4 ทำการเลือกชนิดของ key ในชนิดแรก คือ Device Key นั้นสามารถใช้ได้กับอุปกรณ์หรือ Application ทั่วๆไป ส่วนชนิดที่สองที่เรียกว่า Session Key เหมาะกับการเลือกให้กับ Application ที่ใช้ HTML5 Microgear เช่น Browser ที่ run HTML5 บนหน้าเว็บสำหรับ monitor อุปกรณ์เซนเซอร์ จะเหมาะสมกว่า สำหรับ key ทั้งสองประเภทนั้น อุปกรณ์หรือ Things สองชิ้นขึ้นไปสามารถร่วมใช้ Application Key เดียวกันได้

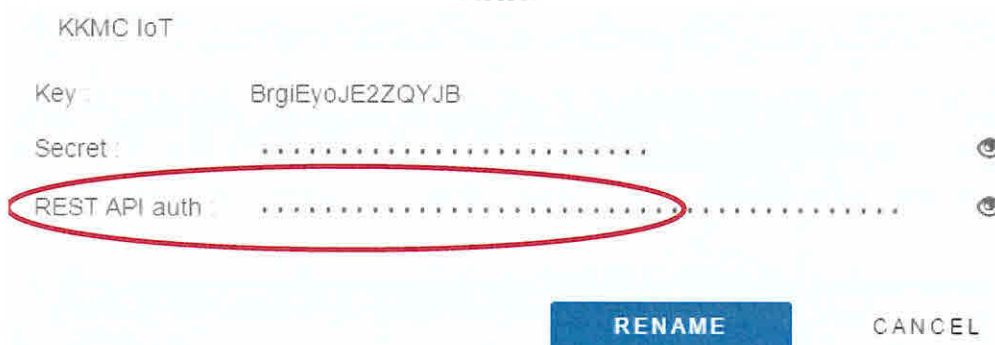


รูปที่ 3.7 Key สำหรับไว้ให้อุปกรณ์เอาไปใช้ในการสื่อสาร



รูปที่ 3.8 APP ID ในเว็บ NETPIE

ขั้นตอนที่ 5 ทำการเลือกชนิดของ Key เป็น Device Key และตั้งชื่อว่า “KKMC IoT” เพื่อให้สามารถใช้ได้กับ Application KKMC IoT บน Smartphone ซึ่งสามารถเชื่อมต่อกับ Application KKMC IoT ได้โดยนำข้อมูล REST API auth ไปตั้งค่าลง Application



รูปที่ 3.9 แสดง Key Secret และ REST API auth เพื่อใช้ในการเชื่อมต่อกับ NETPIE และ Application

ขั้นตอนที่ 6 ทำการเลือกชนิดของ Key เป็น Session Key และตั้งชื่อว่า “Freeboard NETPIE” เพื่อให้สามารถแสดงค่าพารามิเตอร์ต่างๆ รวมทั้งแสดงสถานะการเปิด/ปิดของกังหันน้ำให้ใช้ได้กับ NETPIE Freeboard

Freeboard NETPIE

Key : rEOFMalykyhdIV5

Secret :

REST API auth :

RENAME CANCEL

รูปที่ 3.10 แสดง Key Secret และ REST API auth เพื่อใช้ในการเชื่อมต่อกับ NETPIE Freeboard

3.2.3 การสร้าง Datasource บน Freeboard

ขั้นตอนที่ 1 เข้าสู่หน้าหลักของ NETPIE และเลือก Freeboard เข้าสู่ Freeboard Management เพื่อสร้าง Datasource



รูปที่ 3.11 หน้าหลักของ NETPIE เพื่อเข้าสู่ Freeboard Management

ขั้นตอนที่ 2 คลิกที่เครื่องหมาย + เพื่อสร้าง Freeboard ตั้งชื่อ Freeboard ว่า “PumpControl”



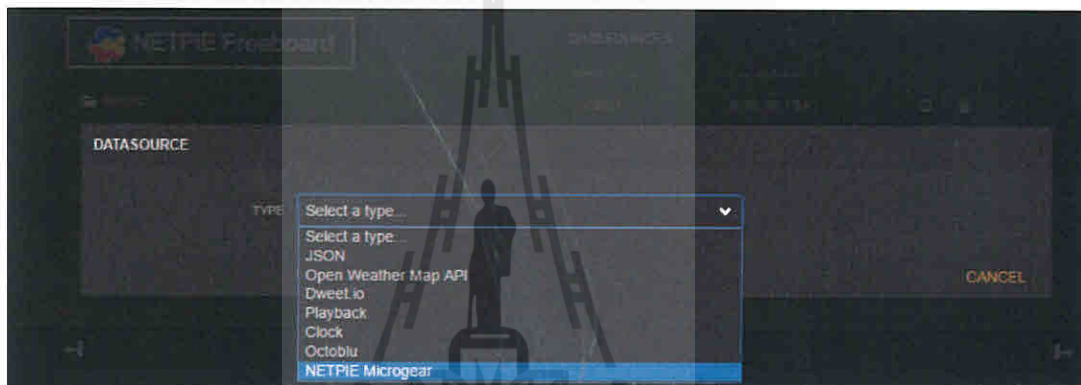
รูปที่ 3.12 สร้างชื่อ NETPIE Freeboard

ขั้นตอนที่ 3 ทำการเปิด NETPIE Freeboard ฝั่งส่วนบนขวา จะมีส่วนของ Datasource ให้ทำการกดที่ปุ่ม ADD



รูปที่ 3.13 เพิ่ม Datasource

ขั้นตอนที่ 3 ทำการเลือก Type เป็นแบบ NETPIE Microgears



รูปที่ 3.14 Type บน Datasource

ขั้นตอนที่ 4 การกำหนดค่าให้ Datasource ใส่ข้อมูล Name, App ID, Key, Secret, Device Alias, Microgear Reference เมื่อกรอกข้อมูลเรียบร้อยแล้ว ให้กด Save

DATASOURCE

Connect to NETPIE as a microgear to communicate real-time with other microgears in the same App ID

TYPE: NETPIE Microgear

NAME: data1

APP ID: TestControl

KEY: rEOFMalykyhdIV5

SECRET: 3ZepEd8QNnOlgJE3ydA4TexL

DEVICE ALIAS: TempSenser

MICROGEAR REFERENCE: testpump1

SUBSCRIBED TOPICS: /#

ONCONNECTED ACTION:

SAVE CANCEL

รูปที่ 3.15 กำหนดค่าให้ Datasource

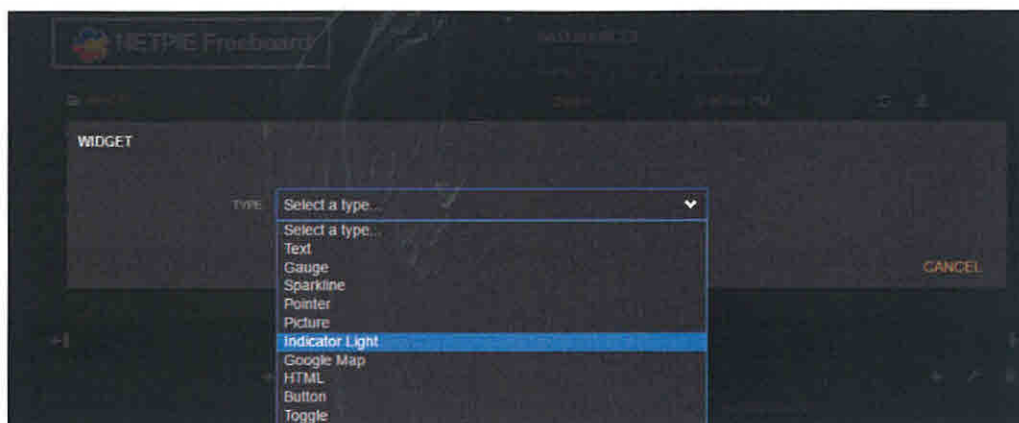
ขั้นตอนที่ 5 Datasource ที่เราสร้างขึ้นปรากฏในรายชื่อ Datasource

NETPIE Freeboard

DATASOURCE 1			
NAME	data1	Created	6:46:46 PM
STATUS	Ready	Updated	6:46:28 PM
ADD			

รูปที่ 3.16 แสดง Datasource ที่สร้างขึ้น

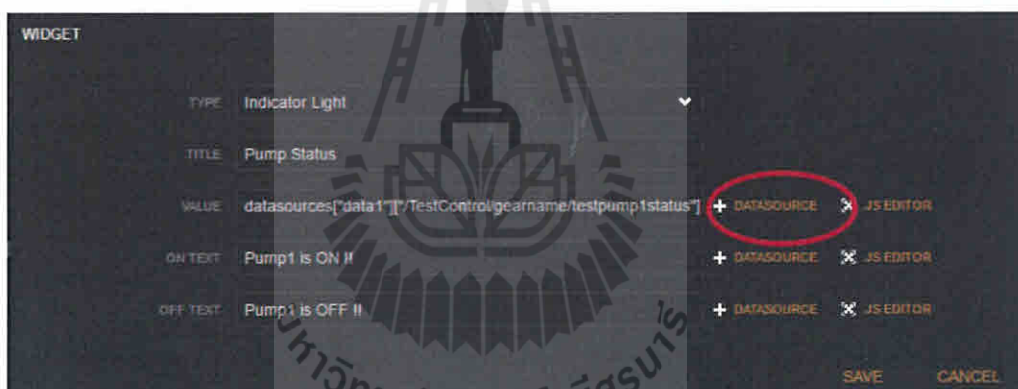
ขั้นตอนที่ 6 การเลือกสถานะการเปิด/ปิดของกังหันน้ำ โดยเลือก Widget Type แบบ Indicator Light



รูปที่ 3.17 การเลือก Type แบบ Indicator Light แสดงสถานะของกังหันน้ำ

ขั้นตอนที่ 7 ใส่ข้อมูล Datasource ที่ช่อง Value โดยจะใส่ค่าดังนี้

`datasources["data1"]["/TestControl/gearname/testpump1status"] == "ON"` หมายถึง เมื่อมีการส่งข้อความคำสั่ง testpump1status ก็จะแสดงขึ้นไฟสถานะติด ที่ NETPIE freeboard



รูปที่ 3.18 ใส่ข้อมูล Widget ของการแสดงสถานะการเปิด/ปิด กังหันน้ำ

ขั้นตอนที่ 8 เลือก Widget Type แบบ Gauge เพื่อแสดงค่าพารามิเตอร์ในน้ำ แบบ

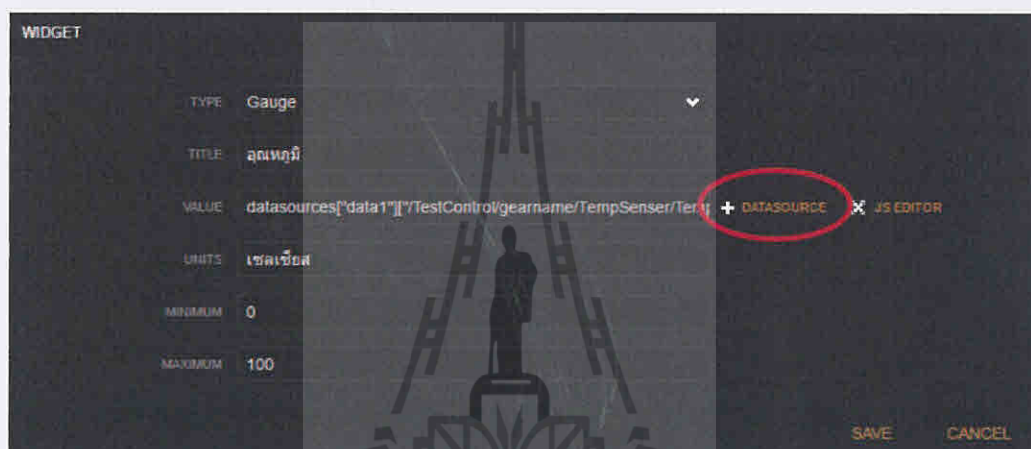
Real-Time



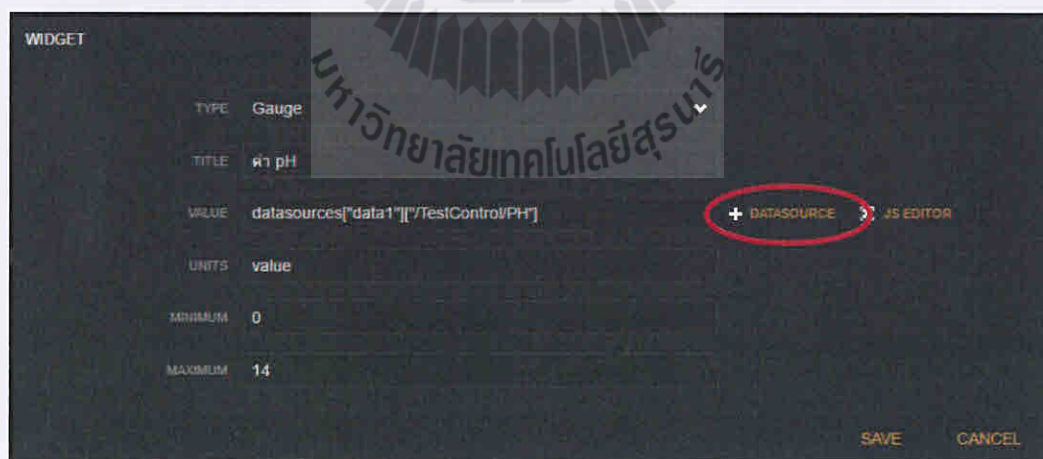
รูปที่ 3.19 Widget Type แบบ Gauge

ขั้นตอนที่ 8 ใส่ข้อมูล Datasource ที่ช่อง Value โดยจะใส่ค่าดังนี้

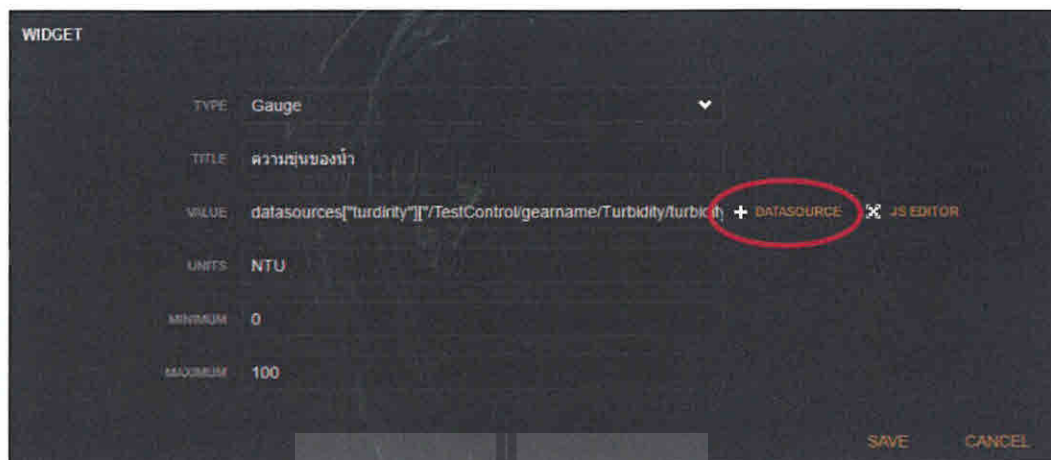
1. `datasources["data1"]["TestControl/gearname/TempSenser/Temperature"]` หมายถึงการส่งค่าอุณหภูมิในน้ำ เพื่อมาแสดงค่าแบบ Real-Time ที่ NETPIE freeboard
2. `datasources["data1"]["TestControl/pH"]` หมายถึงการส่งค่าความเป็นกรด-ด่าง(pH)ในน้ำ เพื่อมาแสดงค่าแบบ Real-Time ที่ NETPIE freeboard
3. `datasources["data1"]["TestControl/gearname/Turbidity/Turbidity"]` หมายถึงการส่งค่าความขุ่นในน้ำ เพื่อมาแสดงค่าแบบ Real-Time ที่ NETPIE freeboard



รูปที่ 3.20 เงื่อนไขการแสดงค่าเซนเซอร์อุณหภูมิในน้ำ บน NETPIE freeboard



รูปที่ 3.21 เงื่อนไขการแสดงค่าเซนเซอร์ความเป็นกรด-ด่าง(pH)ในน้ำ บน NETPIE freeboard



รูปที่ 3.22 ขั้นตอนการตั้งค่าแสดงค่าเซนเซอร์ความขุ่นของน้ำ บน NETPIE freeboard

3.3 Application KKMC IoT

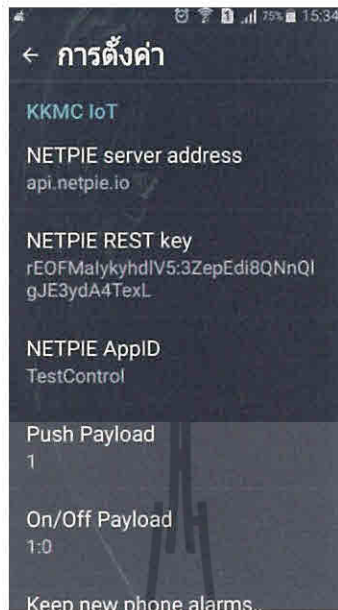
3.3.1 การตั้งค่า Application เพื่อให้ใช้งานได้

ขั้นตอนที่ 1 กดเลือกเมนูตั้งค่าที่ปุ่มด้านล่าง



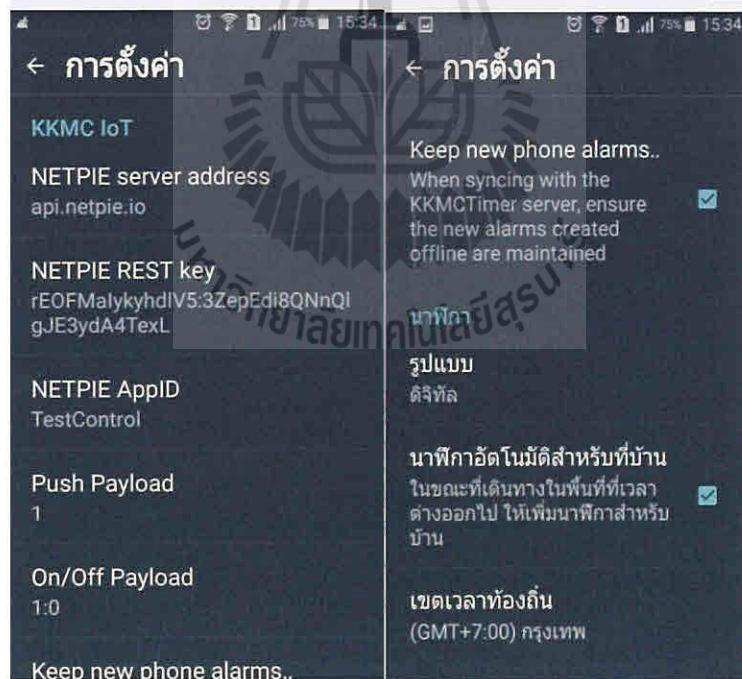
รูปที่ 3.23 หน้าแรกของ Application KKMC IoT

ขั้นตอนที่ 2 ใส่ App ID และ Rest Key โดยต้องไปสร้าง Application ในเว็บ NETPIE ก่อน



รูปที่ 3.24 แสดงการตั้งค่า Application KKMCIoT

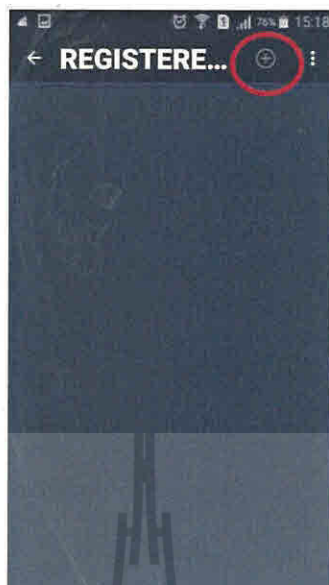
ขั้นตอนที่ 3 นำค่า App ID และ Rest Key กรอกใส่การตั้งค่าให้ถูกต้อง



รูปที่ 3.25 กรอกข้อมูล App ID และ Rest Key เสร็จสมบูรณ์

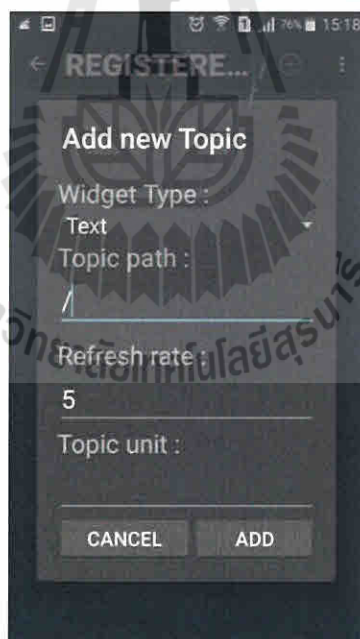
3.3.2 การเพิ่ม Widget เพื่อแสดงข้อมูลและสั่งเปิด/ปิด อุปกรณ์

ขั้นตอนที่ 1 กดรูป + ด้านบนเพื่อเรียกหน้าต่าง Add



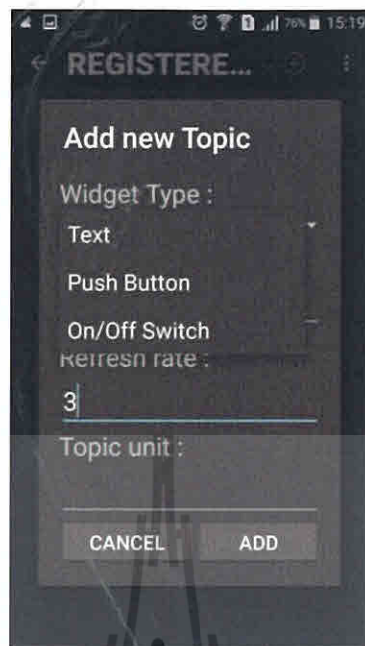
รูปที่ 3.26 แสดงการเพิ่ม Widget

ขั้นตอนที่ 2 หลังจากกดจะเป็นการเรียกหน้าต่าง Add new Topic ขึ้นมา

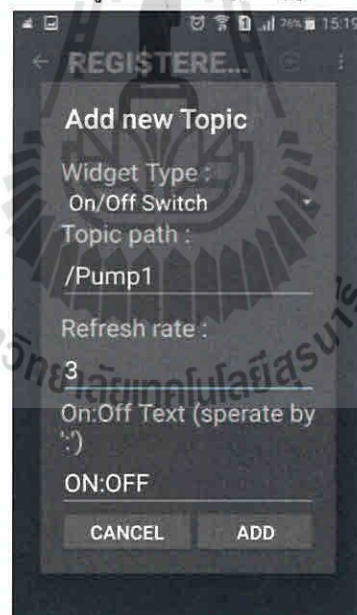


รูปที่ 3.27 หน้าต่าง Add new Topic

ขั้นตอนที่ 3 Widget Type ให้เลือกใช้รูปแบบ On/off Switch คือ สวิตช์ เปิดปิด และกำหนด Topic เป็น /Pump1 เป็นการควบคุมการเปิด/ปิด ของกังหันน้ำ



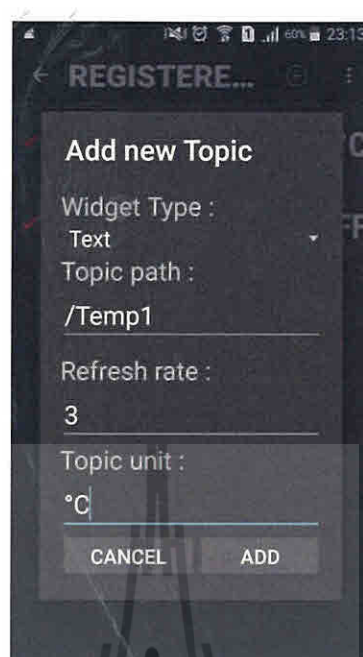
รูปที่ 3.28 รูปแบบ Widget Type 3 แบบ



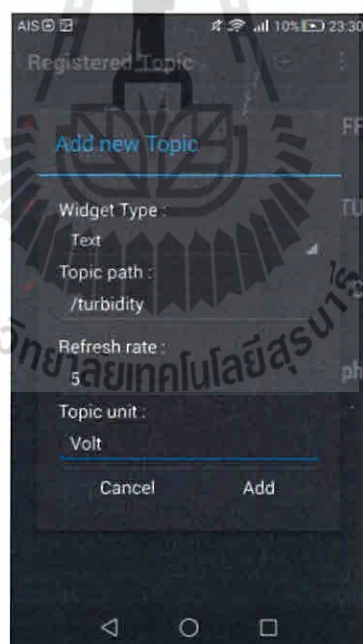
รูปที่ 3.29 รูปแบบ Widget Type แบบ On/off Switch

ขั้นตอนที่ 4 เลือก Widget Type เป็น Text และกำหนด Topic ดังนี้

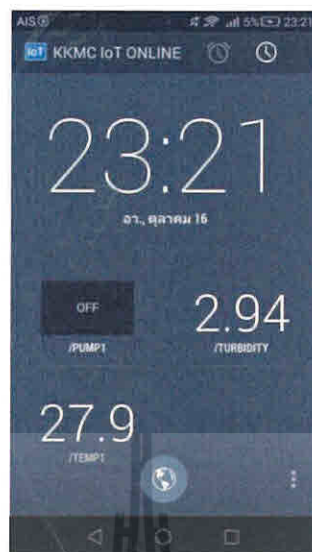
1. /Temp1 เพื่อรับข้อมูลเซนเซอร์อุณหภูมิในน้ำ หลังจากกด Add ก็จะเริ่มรับค่าเข้ามาจากตัวส่ง
2. /turbidity เพื่อรับข้อมูลเซนเซอร์ความขุ่นในน้ำ หลังจากกด Add ก็จะเริ่มรับค่าเข้ามาจากตัวส่ง



รูปที่ 3.30 กำหนด Topic เพื่อรับค่าข้อมูลอุณหภูมิในน้ำ



รูปที่ 3.31 กำหนด Topic เพื่อรับค่าข้อมูลความขุ่นในน้ำ



รูปที่ 3.32 หน้าหลักของ Application แสดงค่าพารามิเตอร์ในน้ำและควบคุมกังหันน้ำ

3.4 การเชื่อมต่อการสื่อสารให้อุปกรณ์ หรือ THING

เนื่องจากข้อมูลอยู่ภายในระบบการให้บริการ Cloud Platform ของ NETPIE (Network Platform for Internet of Everything) และต้องการเชื่อมต่อข้อมูลเพื่อนำมาแสดงผล โดย NETPIE จะใช้ Library ที่ชื่อ Microgears² ซึ่ง Microgear สำหรับ Arduino จะมีการเชื่อมต่อฐานข้อมูลได้ดังนี้

```
#include <AuthClient.h>
#include <MicroGear.h>
#include <MQTTClient.h>
#include <SHA1.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <EEPROM.h>
#include <MicroGear.h>
```

```
const char* ssid = "sutheera";
const char* password = "kaofang1111";
```

² สามารถดาวน์โหลดโค้ดได้จาก <https://github.com/netpieio/microgear-esp8266-arduino>

```
#define APPID "TestControl"
#define KEY "nxTVRwi8XgziqfS"
#define SECRET "FBWAAfq9oumgq34zK7IGx7d38"
```

```
WiFiClient client;
```

```
AuthClient *authclient;
```

ตรงส่วนของ KEY, SECRET และ APPID สามารถแก้ไขโดยใช้ข้อมูลที่ได้อัปเดตไว้จากเว็บ NETPIE และการดึงข้อมูลมาใช้จะต้องมีการตั้ง Topic ให้กับอุปกรณ์ทุกตัวเพื่อให้อุปกรณ์เชื่อมต่อส่งข้อมูลถึงกันได้ มีการเชื่อมต่อโดยใช้คำสั่งดังนี้

```
#define Topic1 "/Temp1"
#define Topic2 "/turbidity"
#define Topic3 "/pH"
microgear.subscribe("/Pump1");
```

3.5 การแสดงข้อมูล บน NETPIE Freeboard

เมื่อเชื่อมต่อข้อมูลกับฐานข้อมูลของระบบการให้บริการ Cloud Platform ของ NETPIE (Network Platform for Internet of Everything) ได้แล้ว เราจะนำข้อมูลไปแสดงผลโดยใช้บริการของ NETPIE Freeboard ซึ่งจะเป็นการแสดงผลแบบ Real-Time และยังมีการสื่อสารแบบ publish-subscribe ในรูปแบบเฉพาะที่ทำให้อุปกรณ์ IoT หรือ “Things” สามารถติดต่อสื่อสารโต้ตอบกันได้อย่างเสรี การสื่อสารแบบนี้สามารถรองรับได้ทั้งโปรโตคอล MQTT และ HTTP REST

ระบบจะมีส่วนของ Device Alias ใช้เป็นเหมือนชื่อเรียกเวลาเราต้องการ chat ไปหา Datasource ผ่านคำสั่งต่างๆ โดยมีคำสั่งดังนี้

```
void onMsgghandler(char *topic, uint8_t* msg, unsigned int msglen) {
    Serial.print("Incoming message --> ");
    msg[msglen] = '\0';
    Serial.println((char *)msg);
    //-----ช่วงประมวลผลคำสั่งเปิด/ปิด-----//
    String stopic(topic);
    String smsg((char*)msg);
    if(stopic == "/TestControl/Pump1"){
        if(smsg == "1" ){
```

```

digitalWrite(D6, HIGH);
microgear.chat("testpumpstatus", "ON"); // ขึ้นไฟสถานะติด ที่ NETPIE Freeboard
}else{
digitalWrite(D6, LOW);
microgear.chat("testpumpstatus", "OFF"); // ไฟสถานะดับ ที่ NETPIE Freeboard
}
}
}
}

```

การส่งค่าและแสดงข้อมูลของค่าอุณหภูมิ ค่าความขุ่นค่า ความเป็นกรด-ด่าง(pH) ในน้ำสามารถใช้คำสั่งได้ดังนี้

```

microgear.chat("TempSenser/Temperature",temp); // ทำการ chat ตัวแปรต่างๆ ออกไป
microgear.chat("Turbidity/Turbidity",turbidity);
microgear.chat("pHSenser/pH Meter",pHValue);

```

3.6 การแสดงข้อมูล ผ่าน Android Application KKMC IoT

Application KKMC IoT ถูกออกแบบและพัฒนาโดย Khon Kaen Maker Club ซึ่งเราเลือก Application นี้มาใช้งาน เพราะเป็น Application ควบคุมอุปกรณ์ IoT ผ่าน NETPIE ซึ่งสามารถกดสร้าง Widget การควบคุมอุปกรณ์ได้เพียงเข้าไปตั้งค่า Topic สำหรับการเชื่อมต่อ จากนั้นก็สามารถใช้งานได้ ภายใน Application มีทั้งการแสดงผล และควบคุมอุปกรณ์ ก่อนที่จะเชื่อมกับ Application ได้ สามารถใช้คำสั่งดังนี้

```

microgear.publish(Topic1,tempread,retained);
microgear.publish(Topic2,turbidity,retained);
microgear.publish(Topic3, pHValue,retained);

```

บทที่ 4

การทดสอบอุปกรณ์และแสดงผลข้อมูล

4.1 การติดตั้งอุปกรณ์

การทดสอบอุปกรณ์ผู้จัดทำได้นำไปทดสอบกับชุดสาริทธิการจำลอง เพื่อวัดค่าอุณหภูมิ ค่าความเป็นกรด-ด่าง(pH) ค่าความขุ่นของน้ำ และควบคุมกังหันน้ำ โดยให้แสดงค่าและควบคุมการเปิด-ปิดมอเตอร์ของกังหันบน Application KKMC IoT อุปกรณ์ประกอบไปด้วย เซนเซอร์วัดอุณหภูมิแบบกันน้ำได้ เซนเซอร์วัดค่าความเป็นกรด-ด่างในน้ำ(pH) และ เซนเซอร์วัดค่าความขุ่นในน้ำ Relay Module ควบคุมการเปิด/ปิดมอเตอร์ของกังหัน ใช้ออร์ด Wemos D1 WiFi ESP8266 อ่านค่าจากเซนเซอร์และ Relay Module แล้วส่งค่าไปยัง Cloud Platformบน NETPIEเชื่อมต่อกับ Android Application KKMC IoT เพื่อให้เห็นผลบน Smartphone



รูปที่ 4.1 ชุดอุปกรณ์



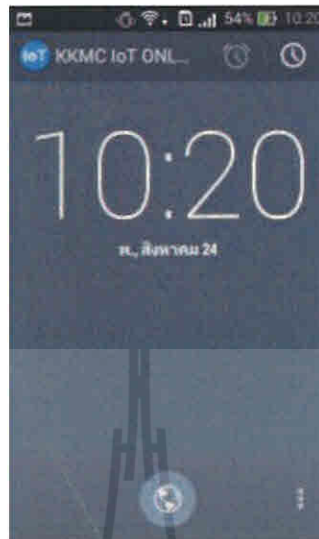
รูปที่ 4.2 ภาพอุปกรณ์ภายในของภาคส่ง

ขั้นตอนที่ 1 ติดตั้งและต่ออุปกรณ์ทั้งหมด



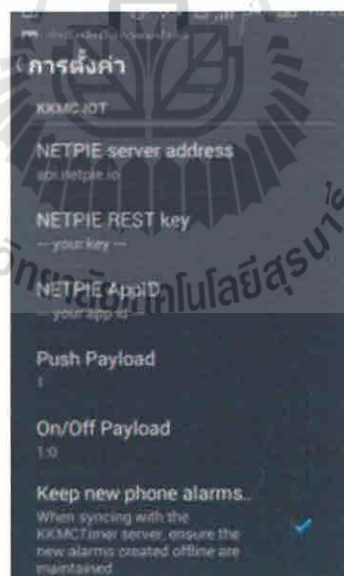
รูปที่ 4.3 ภาพอุปกรณ์ทั้งหมด

ขั้นตอนที่ 2 เข้าไปที่ Application KKMC IoT บน Smartphone



รูปที่ 4.4 หน้าแรกของ Application KKMC IoT

ขั้นตอนที่ 3 ใส่ APP ID และ REST KEY โดยต้องไปสร้าง Application ในเว็บ NETPIE



รูปที่ 4.5 การตั้งค่า Application KKMC IoT

ขั้นตอนที่ 4 จากนั้นจะเป็นการเรียกหน้าต่าง Add new Topic ขึ้นมา

รูปที่ 4.6 แสดงหน้าต่าง Add new Topic

ขั้นตอนที่ 5 ทำการใส่ Widget type สามารถเลือกได้ว่าต้องการใช้งานอะไร มีให้เลือกใช้ 3 รูปแบบ คือ

- 1.Text คือการการแสดงตัวหนังสือ
2. Push Button คือ ปุ่ม กดติดปล่อยดับ
- 3.On/Off Switch คือ สวิตช์ เปิดปิด

รูปที่ 4.7 รูปแบบ Widget type

ขั้นตอนที่ 6 ทำการเชื่อมต่อ Application KMMC IoT กับ NETPIE เมื่อทำการเชื่อมต่อและ
ส่งข้อมูลสำเร็จจะขึ้นคำว่า “Publish to topic success”



รูปที่ 4.8 เชื่อมต่อ Application KMMC IoT กับ NETPIE

ขั้นตอนที่ 7 ทดสอบแสดงค่าพารามิเตอร์ของเซนเซอร์ และสั่งเปิดมอเตอร์กักหน้ใน
Application KMMC IoT บน Smartphone



รูปที่ 4.9 ค่าพารามิเตอร์ของเซนเซอร์และแสดงสถานะเปิดการทำงานของกังหันน้ำ
ขั้นตอนที่ 8 ทดสอบแสดงค่าพารามิเตอร์ของเซนเซอร์ และสั่งเปิดมอเตอร์กังหันน้ำบน

NETPIE



รูปที่ 4.10 ค่าพารามิเตอร์ของเซนเซอร์และแสดงสถานะเปิดการทำงานของกังหันน้ำ

ขั้นตอนที่ 9 ทดสอบแสดงค่าพารามิเตอร์ของเซนเซอร์ และสั่งปิดมอเตอร์กังหันน้ำใน

Application KKMC IoT บน Smartphone



รูปที่ 4.11 ค่าพารามิเตอร์ของเซนเซอร์และแสดงสถานะปิดการทำงานของกังหันน้ำ
ขั้นตอนที่ 10 ทดสอบแสดงค่าพารามิเตอร์ของเซนเซอร์ และสั่งปิดมอเตอร์บน NETPIE



รูปที่ 4.12 ค่าพารามิเตอร์ของเซนเซอร์และแสดงสถานะปิดการทำงานของกังหันน้ำ

4.2 วิเคราะห์ผลการทดลอง

จากผลการทดลองทำให้ทราบค่าพารามิเตอร์ทั้ง 3 ตัว ของเซนเซอร์วัดค่าอุณหภูมิ เซนเซอร์วัดค่าความขุ่นและเซนเซอร์วัดค่าความเป็นกรด-ด่างในน้ำ(pH) แสดงผลผ่าน Android Application IoT โดยผู้จัดทำได้ติดตั้งอุปกรณ์ อ่านค่าพารามิเตอร์ของเซนเซอร์ทั้ง 3 ตัว และได้ทดลองการควบคุมกังหันน้ำผ่าน Android Application KKMC IoT ซึ่งค่าพารามิเตอร์ที่แสดงใน

Application บน Smartphone จะมีความหน่วงเวลาประมาณ 5 วินาทีเมื่อเทียบกับค่าพารามิเตอร์ของ เซนเซอร์ที่แสดงบน NETPIE Freeboard

การควบคุมกังหันน้ำผ่าน Android Application KKMC IoT สามารถเชื่อมต่อ Code เพื่อ ควบคุมได้ดังนี้

```
void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen) {

    Serial.print("Incoming message --> ");

    msg[msglen] = '\0';

    Serial.println((char *)msg);

    //-----ช่วงประมวลผลคำสั่งเปิด/ปิด-----//

    String stopic(topic);

    String smsg((char*)msg);

    if(stopic == "/TestControl/Pump1"){

        if(smsg == "1" ){

            digitalWrite(D6,HIGH);

            microgear.chat("testpumpstatus", "ON"); // ขึ้นไฟสถานะติด ที่ NETPIE Freeboard

        }else{

            digitalWrite(D6,LOW);

            microgear.chat("testpumpstatus", "OFF"); // ไฟสถานะดับ ที่ NETPIE Freeboard

        }

    }
```

จาก Code ข้างต้น ในการกำหนดการเปิด/ปิด ถ้าส่งค่า 1 กังหันน้ำจะเริ่มทำงาน และมีการ chat ถึงกันระหว่างชุดอุปกรณ์กับ Smartphone โดยจะเชื่อมกันและถูกประมวลผลการทำงานจาก Cloud Platform ของ NETPIE ด้วยคำสั่ง microgear.chat("testpumpstatus", "ON"); หมายความว่ามีการส่งข้อความ "ON" ไปหาผู้รับชื่อ "testpumpstatus" ส่งผลให้มีสถานะขึ้นไฟติดที่ NETPIE Freeboard ในการกำหนดการเปิด/ปิด ถ้าส่งค่าที่นอกเหนือจาก 1 กังหันน้ำจะหยุดทำงาน และไฟสถานะจะดับ ที่ NETPIE Freeboard โดยการควบคุมเปิด/ปิดค่าจะหน่วงเวลาประมาณ 5 วินาที

การส่งค่าขึ้น NETPIE Freeboard จะเป็นการส่งค่าพารามิเตอร์ต่างๆแบบ Real Time จะมี การตั้ง Topic เฉพาะของแต่ละอุปกรณ์ เพื่อนำมาเรียกใช้งานทั้งใน NETPIE Freeboard และ Application KKMC IoT บน Smartphone มีดังนี้

```
#define retained true
```

```
#define Topic1 "/Temp1"
```

```
#define Topic2 "/turbidity"
```

```
#define Topic3 "/pH"
```

```
microgear.subscribe("/Pump1");
```

การส่งค่าและแสดงข้อมูลของค่าอุณหภูมิ ค่าความขุ่นค่า ความเป็นกรด-ด่าง(pH) ในน้ำ สามารถใช้คำสั่งได้ดังนี้

```
microgear.chat("TempSenser/Temperature",temp); // ทำการ chat ตัวแปรต่างๆ ออกไป
```

```
microgear.chat("Turbidity/Turbidity",turbidity);
```

```
microgear.chat("pHSenser/pH Meter",pHValue);
```

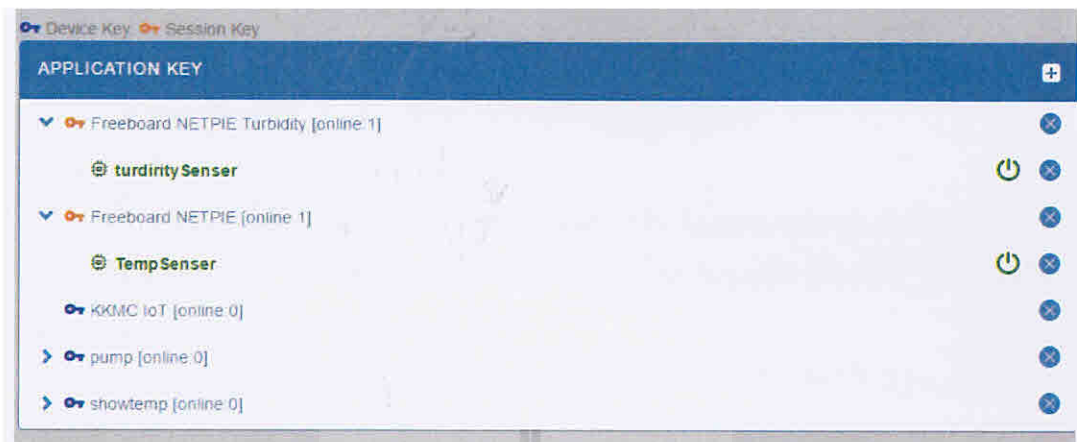
การใช้คำสั่ง Microgear.publish จะเป็นการส่งข้อความแบบไม่เจาะจงผู้รับไปยัง Topic (หัวข้อสนทนา)ที่กำหนด

```
microgear.publish(Topic1,tempread,retained);
```

```
microgear.publish(Topic2,turbidity,retained);
```

```
microgear.publish(Topic3, pHValue,retained);
```

เมื่อทำการส่งข้อมูลขึ้น NETPIE ผู้ใช้สามารถเช็คอุปกรณ์ที่กำลังทำงานอยู่ได้ที่ Application Key โดยอุปกรณ์ที่ทำงาน จะมีสถานะ [online:1] หมายถึง ผู้ใช้กำลังใช้งานอุปกรณ์อยู่ 1 อุปกรณ์



รูปที่ 4.13 สถานะการเชื่อมต่ออุปกรณ์เมื่อส่งขึ้น NETPIE สำเร็จ

4.3 สรุปผลการทดลอง

เมื่อชุดอุปกรณ์การทดลองได้ส่งข้อมูลขึ้น NETPIE แสดงค่าพารามิเตอร์ของเซนเซอร์แต่ละตัวและการควบคุมมอเตอร์ของกังหันน้ำผ่าน Android Application KKMC IoT บน Smartphone เพื่อที่จะให้ผู้เชี่ยวชาญสามารถเข้าไปตรวจสอบค่าและควบคุมการเปิด/ปิดมอเตอร์กังหันน้ำ เพื่อแก้ไขปัญหาได้ทันเวลา ซึ่งจะเห็นได้ว่าจะสามารถใช้ Android Application KKMC IoT ได้ตามแสดงที่ต่างๆที่มีอินเทอร์เน็ต และยังสามารถนำอุปกรณ์ไปประยุกต์ใช้กับอุปกรณ์อื่นๆอีกต่อไป

บทที่ 5

บทสรุปของโครงการ

5.1 กล่าวสรุป

เนื้อหาในบทนี้เป็นการกล่าวถึงบทสรุปของโครงการ Android Application ควบคุมกักกันน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ จะประกอบไปด้วยปัญหาที่พบในขณะดำเนินงาน วิธีการแก้ไข ข้อเสนอแนะและวิธีการพัฒนาโครงการต่อไป

โครงการ Android Application ควบคุมกักกันน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ มีส่วนประกอบหลักดังนี้

1. เซนเซอร์วัดอุณหภูมิในน้ำ
2. เซนเซอร์วัดค่าความขุ่นในน้ำ
3. เซนเซอร์วัดค่าPHในน้ำ
4. Relay Module ทำหน้าที่เป็นตัวควบคุมการเปิด/ปิดของกักกัน
5. Wemos d1 WiFi ESP8266 ทำหน้าที่ส่งข้อมูลไปที่ Cloud Platform ของ NETPIE
6. Amplifier Module ทำหน้าที่รับขา Analog Sensor
7. NETPIE เป็น Cloud Platform ให้บริการสำหรับ IoT
8. Android Application ทำหน้าที่แสดงผลค่าพารามิเตอร์และควบคุมการเปิด/ปิดกักกันน้ำ

จากชุดอุปกรณ์ควบคุมการกักกันน้ำและแสดงค่าพารามิเตอร์ทั้ง3ตัวได้ทำการต่อวงจรและออกแบบระบบการควบคุมการเปิด/ปิดกักกันน้ำ ทำให้สามารถอ่านค่าพารามิเตอร์และเปิด/ปิดกักกันน้ำได้ในจุดบริเวณต่างๆได้สะดวก เพราะในการควบคุมผ่าน Android Application ทำให้สามารถดูค่าคุณภาพของน้ำว่าต่ำมากเกินไปหรือสูงมากเกินไป ทำให้ Relay Module ควบคุมการเปิด/ปิดน้ำ เมื่อส่งข้อมูลและแสดงผลผ่าน NETPIE ทำให้สะดวกในการตรวจสอบข้อมูล

การทดสอบการใช้งานเบื้องต้น ได้จำลองระบบที่พัฒนาขึ้น จากผลการทดลองเบื้องต้นพบว่าชุดอุปกรณ์สามารถทำงานได้ดี และสามารถส่งข้อมูลและควบคุมได้เป็นอย่างดี

5.2 ปัญหาและการแก้ไข

1.การทดลองใช้มอเตอร์ 1.5V ในการควบคุมการหมุนของกังหัน อุปกรณ์หมุนได้ช้า

การแก้ไข เปลี่ยนขนาดมอเตอร์เพื่อให้อุปกรณ์ทดลองใช้งานได้

2.บอร์ด Wemos d1 WiFi ESP8266 มี Analog 1 Analog แต่เซนเซอร์ที่นำมาใช้งานเป็นเซนเซอร์ความชื้นและเซนเซอร์วัดความเป็นกรด-ด่าง(pH) ใช้ขา Analog ทั้ง 2 ตัว

การแก้ไข การเพิ่ม Amplifier Module มาใช้เพิ่มการอ่านค่า Analog

3.การใช้งานของ Android Application KKMC IoT ต้องอยู่ในพื้นที่ที่มีสัญญาณ WiFi หรือ 3G ถึงสามารถใช้งานได้

การแก้ไข ใช้การปล่อยสัญญาณและสอดสอปดแบบพกพา หรืออาจต้องมี Router ในพื้นที่การทำงาน

5.3 ข้อเสนอแนะ

อุปกรณ์เซนเซอร์ที่เชื่อมต่อกับบอร์ด อาศัยแรงดันไฟฟ้าและกระแสจากบอร์ดป้อนเข้ามา ขอมอเตอร์ควบคุมเปิด/ปิด ดังนั้นเพื่อความปลอดภัยต่อบอร์ด Arduino ในการรับสัญญาณที่ได้จึง ต้องมีการคำนวณสร้างวงจรเพิ่มเติมเพื่อเติมเพื่อลดทอนแรงดันของสัญญาณให้เหมาะสม

ในการส่งข้อมูลของบอร์ด WeMos d1 WiFi ESP8266 จำเป็นต้องมีสัญญาณ WiFi หรือ 3G เพื่อเชื่อมต่อบอร์ด WeMos d1 WiFi ESP8266 ถึงสามารถส่งข้อมูลไปถึง Cloud Platform ของ NETPIE ได้

5.4 แนวทางในการพัฒนาต่อไป

- 1.ชุดอุปกรณ์นี้สามารถที่จะควบคุมอุปกรณ์ไฟฟ้าตัวอื่นๆได้อีก เช่น ควบคุมการเปิด/ปิด อุปกรณ์ไฟฟ้าภายในบ้าน
- 2.ชุดอุปกรณ์ชุดนี้สามารถที่จะใช้รับค่าจากเซนเซอร์ตัวอื่นๆได้ เพื่อการตรวจคุณภาพของน้ำที่สมบูรณ์ยิ่งขึ้น

ประวัติผู้เขียน



นางสาวเชวณันท์ จรรย์วรรณ เกิดเมื่อวันที่ 29 พฤษภาคม พ.ศ.2537
ภูมิลำเนาอยู่ที่ ตำบลโนนแหลมทอง อำเภอสหพันธ์ จังหวัดกาฬสินธุ์
สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนสมเด็จพระพิทยาคม
อำเภอสมเด็จ จังหวัดกาฬสินธุ์ เมื่อปีพ.ศ.2555
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



นางสาวบุษกร สมมิตร เมื่อวันที่ 22 พฤษภาคม พ.ศ.2538
ภูมิลำเนาอยู่ที่ ตำบลหนองสองห้อง อำเภอหนองสองห้อง
จังหวัดขอนแก่น
สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนเมืองพลพิทยาคม
อำเภอพล จังหวัดขอนแก่น เมื่อปี พ.ศ. 2555
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



นางสาวสุธีรา แก้วพรรณา เกิดเมื่อวันที่ 26 พฤษภาคม พ.ศ. 2538
ภูมิลำเนาอยู่ที่ ตำบลเชียงคาน อำเภอเชียงคาน จังหวัดเลย
สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนเลยพิทยาคม
อำเภอเมือง จังหวัดเลย เมื่อปี พ.ศ.2555
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

เอกสารอ้างอิง

Amplifier Module. (ออนไลน์). เข้าถึงได้จาก

<https://www.arduitronics.com/product/851/ads1115-i2c-adc-4-channel-16-bit-with-programmable-gain-amplifier-module>

KKMC IoT แอปพลิเคชันควบคุมอุปกรณ์ IoT. (ออนไลน์). เข้าถึงได้จาก

<http://www.kkmakerclub.com/2016/05/26/kkmc-iot/>

เซ็นเซอร์วัดค่า pH ในน้ำ. (ออนไลน์). เข้าถึงได้จาก

<http://www.thaieasyelec.com/products/sensors/ph-d-o-orp/analog-ph-meter-kit-detail.html>

เซ็นเซอร์วัดค่าความขุ่นในน้ำ. (ออนไลน์). เข้าถึงได้จาก

<https://www.arduitronics.com/product/140/4-channel-relay>

เซ็นเซอร์วัดอุณหภูมิแบบกันน้ำได้. (ออนไลน์). เข้าถึงได้จาก

<http://www.arduinoberry.com/product/21/ds18b20>

คุณภาพน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำ. (ออนไลน์). เข้าถึงได้จาก

<http://aqua.club.net/forum/index.php?topic=19106.0;wap2>

ธนพร โถมนกร, ไพฑูรย์ โคตรเจริญ, สันชัย ในพลกรัง, ระบบเซนเซอร์ไร้สายตรวจสอบสภาพแวดล้อม

แปลงอุ้งผ่านอินเทอร์เน็ต, โครงการวิศวกรรมโทรคมนาคม 527499, สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี, 2558.

ระบบการใช้งาน NETPIE. (ออนไลน์). เข้าถึงได้จาก

<https://netpie.io/>

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. (2559). คู่มือการใช้งาน NETPIE (ออนไลน์).

เข้าถึงได้จาก <https://drive.google.com/file/d/0BxilWPgyXTmvSjZVdEFSc3g0Z2c/view>

หลักการทำงานของ Relay Module. (ออนไลน์). เข้าถึงได้จาก

<https://www.arduitronics.com/product/641/2-channel-relay-opto-isolated>

ภาคผนวก

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ก

Code การเชื่อมต่อกับระบบการให้บริการ Cloud Platform ของ NETPIE

```
#include <AuthClient.h>
#include <MicroGear.h>
#include <MQTTClient.h>
#include <SHA1.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <EEPROM.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <Adafruit_ADS1015.h>
```

1. APPID เป็นชื่อที่เรากำหนดไว้ใน NETPIE
2. Key เป็นกุญแจที่ใช้สำหรับอุปกรณ์เชื่อมต่อ NETPIE
3. Secret เป็นรหัสลับหรือรหัสผ่านที่ใช้คู่กับ key สำหรับให้อุปกรณ์เชื่อมต่อ NETPIE โดยทั้ง 3 ตัวนี้จะเป็นข้อมูลเฉพาะ และไม่ซ้ำกันภายในระบบ NETPIE

```
const char* ssid = "sutheera"; // ชื่อ WiFi ในการเชื่อมต่ออุปกรณ์
const char* password = "kaofang1111"; // รหัส WiFi ในการเชื่อมต่ออุปกรณ์
#define APPID "TestControl"
#define KEY "nxTVRwi8XgzqfS"
#define SECRET "FBWAAfq9oumgq34zK7IGx7d38"
#define ALIAS "testpump" // ชื่ออุปกรณ์
```

```
#define retained true
#define Topic1 "/Temp1"
#define Topic2 "/turbidity"
#define Topic3 "/pH"
```

ตั้ง Topic เฉพาะของแต่ละอุปกรณ์ เพื่อนำมาเรียกใช้งานทั้งใน NETPIE Freeboard และ Application KKMC IoT บน Smartphone

```
#define D6 12 // กำหนดขา D6 ต่อเข้าตัว relay IN1
#define temp D5 // กำหนดขา D5 เพื่ออ่านค่าเซนเซอร์อุณหภูมิ
```

```

#define SensorPin A0          //pH meter Analog Input 0
#define Offset 0.00          //deviation compensate(ส่วนเบี่ยงเบนชัดเจน)
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth 40        //times of collection
int pHArray[ArrayLenth]; //เก็บค่าเฉลี่ยของเซนเซอร์
int pHArrayIndex=0;

OneWire oneWire(temp);
DallasTemperature sensors(&oneWire);

WiFiClient client;
AuthClient *authclient;

Adafruit_ADS1115 ads(0x48);

int timer = 0;
MicroGear microgear(client);

void onMsgHandler(char *topic, uint8_t* msg, unsigned int msglen) {
  Serial.print("Incoming message --> ");
  msg[msglen] = '\0';
  Serial.println((char *)msg);
  //-----ช่วงประมวลผลคำสั่งเปิด/ปิด-----//
  String stopic(topic);
  String smsg((char*)msg);
  if(stopic == "/TestControl/Pump1"){
    if(smsg == "1" ){
      digitalWrite(D6, HIGH);
      microgear.chat("testpumpstatus", "ON"); // ขึ้นไฟสถานะติด ที่ NETPIE Freeboard
    }
  }
}

```

ในการกำหนดการเปิด/ปิด ถ้าส่งค่า 1 กังหันน้ำจะ
เริ่มทำงาน และขึ้นไฟสถานะติดที่ NETPIE
Freeboard

```

    }else{
        digitalWrite(D6, LOW);
        microgear.chat("testpumpstatus", "OFF"); // ไฟสถานะดับ ที่ NETPIE Freeboard
    }
}
}
}

```

ในการกำหนดการเปิด/ปิด ถ้าส่งค่าที่
นอกเหนือจาก 1 กังหันน้ำจะหยุดทำงาน และ
ไฟสถานะจะดับ ที่ NETPIE Freeboard

```

void onFoundgear(char *attribute, uint8_t* msg, unsigned int msglen) {
    Serial.print("Found new member --> ");
    for (int i=0; i<msglen; i++)
        Serial.print((char)msg[i]);
    Serial.println();
}

void onLostgear(char *attribute, uint8_t* msg, unsigned int msglen) {
    Serial.print("Lost member --> ");
    for (int i=0; i<msglen; i++)
        Serial.print((char)msg[i]);
    Serial.println();
}

void onConnected(char *attribute, uint8_t* msg, unsigned int msglen) {
    Serial.println("Connected to NETPIE...");
    microgear.setName("mygear");
    microgear.subscribe("/Pump1");
    microgear.setAlias(ALIAS);
}

```

```

void setup() {
    microgear.on(MESSAGE,onMsghandler);
    microgear.on(PRESENT,onFoundgear);
    microgear.on(ABSENT,onLostgear);
    microgear.on(CONNECTED,onConnected);

    Serial.begin(9600);
    Serial.println("Starting...");
    pinMode(D6, OUTPUT);

    if (WiFi.begin(ssid, password)) {
        while (WiFi.status() != WL_CONNECTED) {
            delay(500);
            Serial.print(".");
        }

        Serial.println("WiFi connected");
        Serial.println("IP address: ");
        Serial.println(WiFi.localIP());
        Serial.println("Dallas Temperature IC Control Library Demo");
        Serial.println("pH meter experiment!");
        sensors.begin();
        ads.begin();
        microgear.init(KEY,SECRET,ALIAS);
        microgear.connect(APPID);
    }

    void loop(void)

```

```

{
    if (microgear.connected())
    {
        microgear.loop();

        timer=0;
    }

    float tempread = sensors.getTempCByIndex(0);
    float turbidity;
    int16_t adc1,adc2,adc3;

    char temp[10]; // สร้างตัวแปรไว้ใช้ส่งข้อมูลทาง microgear.chat (ต้องใช้รูปแบบ char)
    // -----ความขุ่นของน้ำ-----
    adc1 = ads.readADC_SingleEnded(1);
    turbidity = (adc1 * 0.1875)/1000;
    Serial.println(turbidity);
    delay(500);
    // -----ค่าอุณหภูมิในน้ำ-----
    Serial.print(" Requesting temperatures...");
    sensors.requestTemperatures();
    Serial.println("DONE");
    Serial.print("Temperature for Device 1 is: ");
    Serial.print(sensors.getTempCByIndex(0));
    sprintf(temp,"%d.%d",(int)tempread);
    // -----ส่งค่าต่างๆ ขึ้น netpic-----
    microgear.chat("TempSenser/Temperature",temp); // ทำการ chat ตัวแปรต่างๆ ออกไป
    microgear.chat("Turbidity/Turbidity",turbidity);

```

```

// -----ส่งค่าต่างๆ ไปที่ AppKKMCiot-----
microgear.publish(Topic1,tempread,retained);
microgear.publish(Topic2,turbidity,retained);

static unsigned long samplingTime = millis();
static unsigned long printTime = millis();
static float pHValue,voltage;
if(millis()-samplingTime > samplingInterval)
{
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime=millis();
}
if(millis() - printTime > printInterval) //Every 800 milliseconds, print a numerical, convert the state of
the LED indicator
{
    Serial.print("Voltage:");
    Serial.print(voltage,2);
    Serial.print(" pH value: ");
    Serial.println(pHValue,2);
    printTime=millis();
}
// -----ส่งค่าขึ้น netpie-----
microgear.chat("pHSenser/pH Meter",pHValue);

//-----ส่งค่าpH ไปที่ AppKKMCiot-----

microgear.publish(Topic3, pHValue,retained);

```

```

    }

}

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

```

```

    }else {
        if(arr[i]>max){
            amount+=max;    //arr>max
            max=arr[i];
        }else {
            amount+=arr[i]; //min<=arr<=max
        }
    } //if
} //for

avg = (double)amount/(number-2);

} //if

return avg;

}

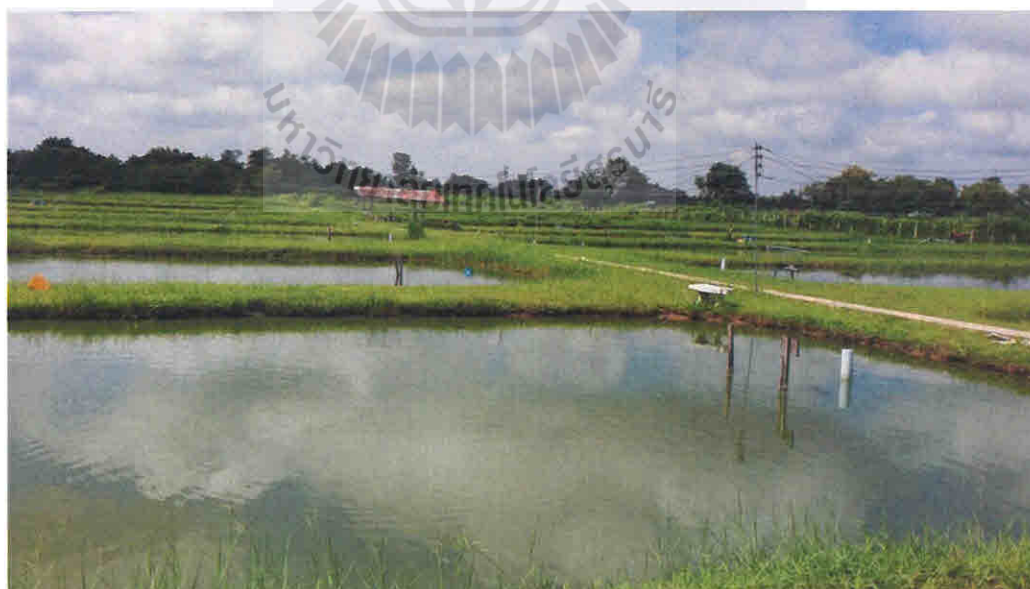
```

ภาคผนวก ข

ภาพรวมโครงการ



รูปที่ 1 (ข) สำรวจพื้นที่ในการทำโครงการ



รูปที่ 2 (ข) สำรวจพื้นที่ในการทำโครงการ



รูปที่ 3 (ข) สำรวจพื้นที่ในการทำโครงการ



รูปที่ 4 (ข) อุปกรณ์ภาคส่ง



รูปที่ 5 (ข) ชุดสาธิตการทดลอง

