# ชื่อโครงงาน: Smart LPWAN Farm

รายวิชา 242-402 Computer Engineering Project Preparation

ภาคการศึกษา 2/2561

**รายชื่อผู้จัดทำ**  
นายเจษฎากร เกิดหนู รหัสนักศึกษา 5835512119

อาจารย์ที่ปรึกษา ผศ.ดร.วโรดม วีระพันธ์

อาจารย์ที่ปรึกษาร่วม อ.ธรรมรัฏฐ์ สมิตะลัมพะ

*ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยสงขลานครินทร์*

**ชื่อโครงงาน** **Smart LPWAN Farm**

**ผู้จัดทำ**  นายเจษฎากร เกิดหนู รหัสนักศึกษา 5835512119

**ภาควิชา** วิศวกรรมคอมพิวเตอร์

**ปีการศึกษา** 2561

**อาจารย์ที่ปรึกษาโครงงาน**

............................................

( )

**คณะกรรมการสอบ**

.............................................. ............................................ ............................................

( ) ( ) ( )

โครงงานนี้เป็นส่วนหนึ่งของรายวิชา Computer Engineering *Project* P*reparation* ตามหลักสูตรปริญญา วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์

………………………………

( )

ผู้จัดการหลักสูตร  
 ภาควิชาวิศวกรรมคอมพิวเตอร์

**หนังสือรับรองความเป็นเอกลักษณ์**

ผู้จัดทำที่ได้ลงนามท้ายนี้ ขอรับรองว่ารายงานฉบับนี้เป็นรายงานที่มีความเป็นเอกลักษณ์ โดยที่ผู้จัดทำไม่ได้มีการคัดลอกมาจากที่ใดเลย เนื้อหาทั้งหมดถูกรวบรวมจากการพัฒนาในขั้นตอนต่าง ๆ ของการจัดทำโครงงาน หากมีส่วนใดที่จำเป็นต้องนำเอาข้อความจากผลงานของผู้อื่น หรือบุคคลอื่นใดที่ไม่ใช่ตัวข้าพเจ้า ข้าพเจ้าได้ทำอ้างอิงถึงเอกสารเหล่านั้นไว้อย่างเหมาะสม และขอรับรองว่ารายงานฉบับนี้ไม่เคยเสนอต่อสถาบันใดมาก่อน

ผู้จัดทำ ..............................................

( )

**ชื่อโครงงาน** **Smart LPWAN Farm**

**ผู้จัดทำ**  นายเจษฎากร เกิดหนู รหัสนักศึกษา 5835512119

**ภาควิชา** วิศวกรรมคอมพิวเตอร์

**ปีการศึกษา** 2561

**บทคัดย่อ**

ปัจจุบันเกษตรกรต้องใช้เวลาในการรดน้ำต้นไม้กับเครื่องสูบน้ำแรงดันต่ำ ส่งผลให้ต้องใช้เวลาในการจัดการกับน้ำที่มีแรงดันไม่ทั่วถึง ส่งผลให้เกิดความชื้นที่ไม่ทั่วถึงทำให้พืชผลทางการเกษตรไม่สามารถเติบโตได้อย่างสมบูรณ์ ในการทดลองนี้ จะสามารถนำมาแก้ปัญหานี้ได้ โดยใช้เทคโนโลยีเครือข่ายไร้สายระยะไกล ที่มีความสามารถในการส่งสัญญาณต่าง ๆ ได้ไกล และความสามารถเพิ่มเติมคือมีความประหยัดพลังงาน ทำให้มีความสะดวกในการติดตั้ง และ มีความสะดวกในการใช้งาน

**คำสำคัญ:** LPWAN, NB-Iot, และ LoRaWAN**Project Title**  **Smart LPWAN Farm**

**Author**  Mr.Jesadakorn Kirtnu 58355512119

**Department** Computer Engineering

**Academic Year** 2561

**Abstract**

At present, farmers have to spend time watering plants with low pressure pumps. Resulting in uneven moisture, causing agricultural crops to not grow completely. In this experiment Will be able to solve this problem. Using Long Rang and Lowpow technology Which has the ability to send various signals far and the additional capability is energy saving Making it convenient to install and convenient to use.

**Keywords:** LPWAN, NB-Iot, and LoRaWAN

**คำนำ**

รายงานฉบับนี้จัดทำขึ้นเพื่อนำเทคโนโลยีของเครือข่ายไรเสายระยะไกลมาประยุกต์ใช้งาน ให้สามารถทำประโยชน์ให้กับเกษตรกร หรือผู้ที่นำไปศึกษาต่อ โดยขั้นตอนการประยุกต์ใช้นั้นผู้จัดทำสามารถทดลองและนำไปใช้กับระบบนี้ได้ แม้ว่าจะมีข้อจำกัดต่าง ๆ ผู้จัดทำหวังเป็นอย่างยิ่งว่า การทดลองของผู้จัดทำ จะสามารถก่อให้เกิดประโยชน์แก่ผู้อื่นได้ ทั้งขอขอบพระคุณ ผศ.ดร.วโรดม วีระพันธ์ ผู้ให้คำปรึกษาแก่ผู้จัดทำ

นายเจษฎากร เกิดหนู

ผู้จัดทำ

22 กุมภาพันธ์ 2562

## สารบัญ

[**ชื่อโครงงาน: Smart LPWAN Farm i**](#_gjdgxs)

[**หนังสือรับรองความเป็นเอกลักษณ์ ii**](#_3znysh7)

[**บทคัดย่อ iii**](#_2et92p0)

[**Abstract iv**](#_tyjcwt)

[**คำนำ v**](#_3dy6vkm)

[**สารบัญ vi**](#_1t3h5sf)

[**บทที่ 1 บทนำ 1**](#_4d34og8)

[**1.1 ความเป็นมา…………….. 1**](#_2s8eyo1)

[**1.2** **วัตถุประสงค์ของโครงงาน 1**](#_3rdcrjn)

[**1.3** **ขอบเขตของโครงงาน 1**](#_1ksv4uv)

[**1.4** **ขั้นตอนในการดำเนินงาน 1**](#_1y810tw)

[**1.5** **ประโยชน์ที่คาดว่าจะได้รับ 2**](#_2xcytpi)

[**1.6** **สถานที่ทำโครงงาน 2**](#_1ci93xb)

[**1.7** **เครื่องมือที่ใช้ในการพัฒนา 2**](#_2bn6wsx)

[**บทที่ 2 ความรู้พื้นฐาน 4**](#_3as4poj)

[**2.1** **เอกสารและงานวิจัยที่เกี่ยวข้อง 4**](#_1pxezwc)

[**บทที่ 3 รายละเอียดการทำงาน 8**](#_49x2ik5)

[**3.1** **System Specification 8**](#_147n2zr)

[**3.2** **System Architecture 9**](#_ihv636)

[**3.3 System Design 10**](#_2grqrue)

[**3.4** **System Implementation 11**](#_3fwokq0)

[**3.5 แผนการดำเนินงาน 12**](#_1v1yuxt)

[**บทที่ 4 ผลการดำเนินงานและสรุปผล 13**](#_4f1mdlm)

[**4.1 ผลการดำเนินงาน 13**](#_2u6wntf)

## สารบัญ(ต่อ)

[**4.2 สรุปผลการทดลอง 13**](#_19c6y18)

[**4.3 ปัญหาและอุปสรรค 13**](#_3tbugp1)

[**4.4 ข้อเสนอแนะ 13**](#_28h4qwu)

[**ภาคผนวก 15**](#_37m2jsg)

**สารบัญรูปภาพ**

[ภาพที่ 3-1 บอร์ด STM32 ของ Cattelecom 8](#_3o7alnk)

[ภาพที่ 3-2 NB-Iot AIS 8](#_23ckvvd)

[ภาพที่ 3-3 โครงข่ายของ Lora 9](#_32hioqz)

[ภาพที่ 3-4 เครือข่ายของ LoRa 10](#_41mghml)

[ภาพที่ 3-5 Flow Chart การทำงาน 10](#_vx1227)

**สารบัญคำย่อ**

NB-Iot Narrow band Internet of Things

Lora Low Power Wide Area Networks

LPWAN Low Power Wide Area Networks

# บทที่ 1 บทนำ

## ความเป็นมา

อุปกรณ์ Iot นั้นสามารถใช้งานได้อย่างหลากหลาย โดยผู้จัดทำนำมาใช้งานกับพืชผลทางการเกษตร สำหรับเกษตรกรที่มีที่ดินหลากหลายสามารถนำอุปกรณ์ในโครงงานนี้ไปใช้เพื่ออำนวยความสำดวก เพิ่มพืชผลการผลิต และลดระยะเวลาในการดูแลรักษา อุปกรณ์ Iot จึงเป็นอุปกรณ์ที่ใช้ได้มีประโยชน์กับการจัดการระบบการเกษตร ผู้จัดทำจึงได้พัฒนาให้ระบบขึ่นมาเพื่อช่วยอำนวยความสะดวกให้เกษตรกรไทย

* 1. **วัตถุประสงค์ของโครงงาน**
     1. เพื่อลดระยะเวลาในการดูแลการรดน้ำพืชผลทางการเกษตร
     2. ช่วยอำนวยความสะดวกให้ผู้ใช้ในการจัดการระบบน้ำ
     3. เพื่อวิเคราะห์สถิติเพื่อให้สามารถวิเคราะห์และจัดการระบบน้ำที่แตกต่างกันในแต่ละผู้ใช้
     4. เพื่อลดค่าใช้จ่ายในระยะยาว
     5. เพื่อเพิ่มผลผลิตทางการเกษตรซึ่งเกิดจากการมีน้ำที่เหมาะสม

## ขอบเขตของโครงงาน

* + 1. ขอบเขตของอุปกรณ์
* ใช้ในการเปิด-ปิด ประตูน้ำแต่ละช่อง
* ใช้บันทึกสถิติเก็บในฐานข้อมูล
* สามารถใช้ระบบอัตโนมัติซึ่งตั้งโดยผู้ใช้ได้
* สามารถใช้แบตเตอรี่เพื่อใช้งานได้ทุกที่
  + 1. ข้อจำกัดของอุปกรณ์
* ใช้งานได้เฉพาะบริเวณที่ครอบคลุมสัญญาณ 4G (LTE), 3G, 2G

## ขั้นตอนในการดำเนินงาน

* + 1. ทำการติดตั้งอุปกรณ์ตรวจวัดความชื้น
    2. สร้างระบบเก็บข้อมูลความชื้น
    3. สร้างแอพพลิเคชั่นในการรับและส่งข้อมูล

## ประโยชน์ที่คาดว่าจะได้รับ

* + 1. ช่วยอำนวยความสะดวกสะบายให้กับผู้ใช้
    2. ลดระยะเวลาในการดูแลระบบน้ำ
    3. เพิ่มความแม่นยำให้กับความชื้นในดินส่งผลให้พืชได้รับน้ำอย่างเต็มที่
    4. สามารถประหยัดค่าใช้จ่ายในกรณีที่ค่าความชื้นในดินยังมากระบบจะไม่สูบน้ำ

## สถานที่ทำโครงงาน

* + 1. ห้องปฏิบัติการฮาร์ดแวร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต
    2. ห้องชมรมฮาร์แวร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต
    3. ห้องปฏิบัติการซอฟต์แวร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

## เครื่องมือที่ใช้ในการพัฒนา

* + 1. Hardware
* ASUS A550JX Intel Core i7-4720HQ (2.60 - 3.60 GHz) NVIDIA GeForce GTX 950M (4GB GDDR3) 4 GB DDR3L
* Arduino Uno R3
* Devio NB-Shield I (Quectel BC95)
* Solar cell Solar panels 5.5v 110ma 0.6W
* Relay Module 2 CH 5V 10A, 250V
* Stepup and Down USB or microUSB to output adjustable voltage of 1.5 -24V
* Step up & Down (Buck&Boost) Super XL6009 5-32V to 1.25-32V 4A
* Battery 12V 5Ah
* USB TTL
* Soil Moisture Sensor Module v1
  + 1. ภาษาที่ใช้
* Java
* Java Script
* HTML
* C, C++
* Python
* PHP
* ATCommand
* Unix Command
  + 1. ระบบฐานข้อมูลที่ใช้
* MongoDB
* InfluxDB
* MySQL
  + 1. ระบบเบื้องหลังการทำงาน
* Aismagellan
* Clound Amazon AWS
* Ubuntu Server 16.04 LTS
* NodeJS
* Windows Server 2012

# บทที่ 2 ความรู้พื้นฐาน

## เอกสารและงานวิจัยที่เกี่ยวข้อง

* + 1. **LoRaWAN (Long Range Low Power Wireless Platform)**

ลอราเป็นชื่อที่เรียกย่อมาจาก Long Range Low Power Wireless Platform โดยนำสองตัวอักษร ด้านหน้าของสองคำแรกมาใช้ ลักษณะเฉพาะของลอราคือ การมอดูเลตด้วยเทคนิคเชิร์ปสเปรดสเปกตรัม (Chirp Spread Spectrum Modulation) โดยใช้สัญญาณเชิร์ปความชันคงที่ (Constant Ramp Chirp Signal) ในการเพิ่มประสิทธิภาพการรับสัญญาณให้มีค่าความไว (Sensitivity) ที่ดีขึ้นกว่ากระบวนการมอดูเลตชนิด อื่นๆ โดยความแตกต่างของความถี่ระหว่างตัวรับและตัวส่งของสัญญาณเชิร์ปความชันคงที่มีลักษณะคล้ายกับ ความแตกต่างของเวลา ซึ่งง่ายต่อการจัดการ และส่งผลให้วงจรรับและวงจรส่งสามารถใช้อุปกรณ์กำเนิด ความถี่ที่มีราคาไม่สูงได้ ไม่จำเป็นต้องใช้อุปกรณ์กำเนิดความถี่ที่มีความแม่นยำสูง (Augustin, Yi, Clausen, & Townsley, 2016) ซึ่งค่าความถี่ที่แตกต่างกันของตัวรับและตัวส่งอาจมีความแตกต่างมากถึง 20% ได้ โดยค่าความไวของการรับจากการมอดูเลตชนิดนี้สามารถทำให้รับได้ที่ระดับสัญญาณต่ำกว่า -140 dBm ซึ่งถือว่าต่ำมากเมื่อเทียบกับการมอดูเลตชนิดอื่นๆ ที่ใช้อยู่ในซิกบีและไวไฟ ที่อยู่ในระดับ -100 dBm ถึง -110 dBm เท่านั้น อีกหนึ่งประสิทธิภาพที่โดดเด่นของลอราคือ ความสามารถในการดีมอดูเลตหลายสัญญาณ ที่ถูกส่งมาพร้อมกันที่ความถี่เดียวกันได้ โดยสัญญาณที่ถูกส่งมาพร้อมกันจะต้องมีอัตราเชิร์ปที่แตกต่างกัน โดยใช้ค่าสเปรดแฟกเตอร์ที่แตกต่างกัน ผลของการดีมอดูเลตหลายสัญญาณพร้อมกันที่ความถี่เดียว ทำให้ ลอราสามารถรองรับจำนวนอุปกรณ์ไอโอทีได้จำนวนมาก จากที่กล่าวมา ลอราเป็นกระบวนการในชั้นกายภาพ และมีการจัดเฟรมข้อมูลด้วยรูปแบบเฉพาะในชั้นเส้นทางเชื่อมโยงข้อมูล การนำลอรามาใช้งานไอโอทีจำเป็น ต้องส่งต่อข้อมูลจากอุปกรณ์ลอราเข้าสู่อินเทอร์เน็ตผ่านลอราเกตเวย์ (LoRa Gateway) ไปยังลอราแวน (LoRaWAN) ซึ่งมีโพรโทคอลในการส่งผ่านข้อมูลเข้าสู่อินเทอร์เน็ตได้ [1]

* + 1. **NB-IoT (Narrow Band Internet of Things, NB-IoT)**

เอ็นบีไอโอที (Narrow Band Internet of Things, NB-IoT) ถูกนำเสนอโดย 3GPP ผู้กำกับดูแล มาตรฐานด้านการสื่อสารบนโครงข่ายโทรศัพท์เคลื่อนที่ โดยถูกออกแบบให้ใช้กำลังงานต่ำ ความเร็วในการ สื่อสารและความถี่ในการส่งข้อมูลต่ำ อุปกรณ์เอ็นบีไอโอทีทำงานบนย่านความถี่เดียวกันกับที่ GSM, 3G หรือ LTE (Wang et. al, 2016) ซึ่งเป็นย่านความถี่ Licensed Band ที่ต้องได้รับการอนุญาตใช้งานจากหน่วยงาน ที่กำกับดูแลทอพอลอจี การเชื่อมต่อใช้ทอพอลอจีสตาร์ ส่งและรับข้อมูลจากสถานีฐานของเครือข่ายโทรศัพท์ เคลื่อนที่ที่ให้บริการ เอ็นบีไอโอทีใช้แถบความถี่อย่างน้อย 180 kHz ซึ่งสามารถทำได้สามลักษณะคือ ใช้อยู่ บนคลื่นความถี่หนึ่งช่องของ GSM ใช้อยู่บนแถบความถี่คุมของ LTE หรือใช้อยู่บนคลื่นความถี่เดียวกันกับ LTE โดยให้ใช้บนแถบความถี่หนึ่งบล็อก มีความเร็วในการสื่อสาร 250 kbps และมีความไวการรับสัญญาณได้ ในระดับมากกว่า -150 dBm จึงมีระยะทางการสื่อสารที่ไกลมาก โดยมีความไวของการรับสัญญาณดีกว่า GSM และ LTE ที่ใช้อยู่เดิมประมาณ 20 dB ด้วย การที่ผู้ให้บริการโครงข่ายโทรศัพท์เคลื่อนที่เป็นผู้ดำเนินการ สื่อสารข้อมูลกับอุปกรณ์เอ็นบีไอโอทีจึงไม่จำเป็นต้องมีอุปกรณ์อินเทอร์เน็ตเกตเวย์ ข้อมูลจะถูกส่งจาก อุปกรณ์เอ็นบีไอโอทีผ่านโครงข่ายโทรศัพท์เคลื่อนที่ไปยังแอปพลิเคชันเซิร์ฟเวอร์ได้โดยตรง นอกจากนี้ในชั้นกายภาพยังต้องพิจารณาถึงแถบความถี่ที่จะใช้งานในการส่งสัญญาณแบบไร้สายด้วย โดยแบ่งแถบความถี่ออกเป็นสองประเภท คือ 1) Unlicensed Band และ 2) Licensed Band ซึ่งถูกกำหนด การใช้งานในประเทศไทยโดยคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม แห่งชาติ หรือ กสทช. แถบความถี่ย่าน Unlicensed Band ในประเทศไทยมีการกำหนดให้สามารถใช้งานได้ โดยมีค่ากำลังส่งสูงสุดไม่เกินค่าที่กำหนดไว้ในตารางที่ 1 นอกจากนี้ที่ประชุม กสทช. มีมติเห็นชอบให้ใช้ คลื่นความถี่ย่าน 920-925 MHz เพื่อรองรับเทคโนโลยีไอโอที ตาม (ร่าง) ประกาศ กสทช. เรื่อง มาตรฐาน ทางเทคนิคของเครื่องโทรคมนาคมและอุปกรณ์สำหรับเครื่องวิทยุคมนาคมที่ไม่ใช่ประเภท Radio Frequency Identification: RFID ซึ่งใช้คลื่นความถี่ย่าน 920-925 MHz [2]

* + 1. **LoRa (CMWX1ZZABZ) Specification**

**Interfaces** : I2C, UART, USB, SPI   
**Main ICs** : STM32L, SX1276   
**Reference Clocks** : Integrated 32MHz clock (TCXO with frequency error=±2 ppm) and 32.768KHz clock (frequency error=±20 ppm)   
**Supported Frequencies** : 868 MHz, 915 MHz   
**Module Size** : 12.5 mm x 11.6 mm x 1.76 mm (Max)   
**Weight** : 0.48g (Typ)   
**Package** : Metal Shield can   
**RoHS** : This module is compliant with the RoHS directive [3]

* + 1. **NB-Iot (Quectel BC95) Specification**

**Power Supply :** Supply voltage: 3.1V ~ 4.2V   
 Typical supply voltage: 3.6V   
**Power Saving Mode :** Maximum power consumption in PSM: 5uA   
**Transmitting Power :** 23dBm±2dB   
**Temperature Range :** Operation temperature range: -30°C ~ +75°C1)   
 Extended temperature range: -40°C ~ +85°C2)   
**USIM Interface :** Only support USIM card: 3.0V UART Interfaces   
**Main port:**  Used for AT command communication and data transmission, and the baud rate is 9600bps   
 Main port can also be used for firmware upgrading, and the baud rate is 115200bps   
**Debug port** : Debug port is used for debugging   
 Only support 921600bps baud rate Internet Protocol   
**Features** **:** Support IPV4/IPV6\*/UDP/CoAP   
**SMS**\* **:** Text and PDU mode   
 Point to point MO and MT   
**Data Transmission Feature :** Single tone with 15kHz subcarrier: 24kbps (DL), 15.625kbps (UL) **AT Commands :** Compliant with 3GPP TS 27.005, 27.007 and Quectel enhanced AT commands   
**Physical Characteristics :** Size: (19.9±0.15)mm × (23.6±0.15)mm × (2.2±0.2mm) Weight: 1.8g±0.2g   
**Firmware Upgrade :** Firmware upgrade via main port or DFOTA\*   
**Antenna Interface** **:** Connected to antenna pad with 50 Ohm impedance control   
**RoHS** **:** All hardware components are fully compliant with EU RoHS directive [4]

* + 1. **Soil Moisture Sensor V.1**

This is a summary of the soil moisture sensor can be used to detect moisture, when the soil is dry, the module outputs a high level, whereas output low. Using this sensor make an automatic watering system, so that your garden plants without people to manage.

* Operating voltage: 3.3V~5V.
* Adjustable sensitivity (shown in blue digital potentiometer adjustment)
* Dual output mode,analog output more accurate.
* A fixed bolt hole for easy installation.
* With power indicator (red) and digital switching output indicator (green).
* Having LM393 comparator chip, stable.
* Panel PCB Dimension: 3cm x 1.5cm.
* Soil Probe Dimension: 6cm x 2cm.
* Cable Length: 21cm.
* VCC: 3.3V-5V.
* GND: GND.
* DO: digital output interface (0 and 1).
* AO: analog output interface. [5]
  + 1. **Solar cell Solar panels**

Solar cell Solar panels โซลาเซลล์ 5.5v 110ma 0.6W ขนาด 84.5x55.5mm

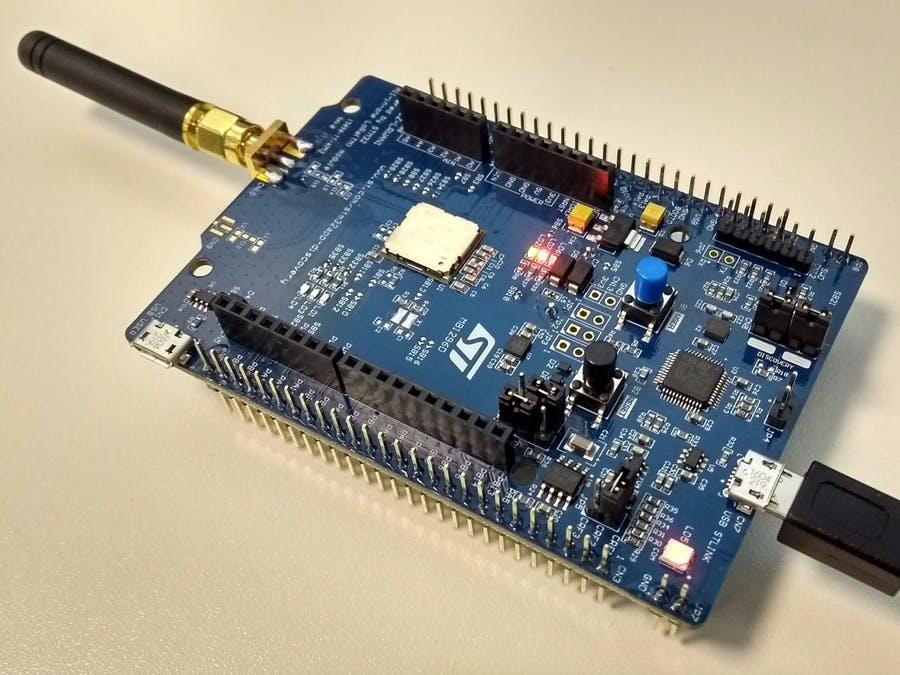
* Gross weight: 0.02KG
* Current: 110MA
* Voltage: 5.5V
* Power: 0.605W
* Material: Epoxy
* Weight: 0.02KG
* Product specifications 84.5\*55.5MM [6]

# บทที่ 3 รายละเอียดการทำงาน

การดำเนินการในการติดตั้งระบบ ผู้จัดทำได้ดำเนินการตามขั้นตอนดังนี้

## System Specification

* + 1. **ซอฟต์แวร์ที่จำเป็น**
* Arduino Software (IDE) เพื่อใช้สำหรับการโปรแกรม ในที่นี้ใช้ได้ทั้งตัวของ QC95 และ STM32
* Node.js ใช้เพื่อสร้าง Socket ในการติดต่อระหว่าง Client กับ Server ของ LoRa
  + 1. **ฮาร์แวรที่จำเป็น**
* LoRa ในการทดลองนี้ใช้ของ Catelecom ในการทดลอง ดังรูป 3-1
* NB-Iot ใช้ของ AIS การทดลอง ดังรูป 3-2



*ภาพที่ 3-1 บอร์ด STM32 ของ Cattelecom*



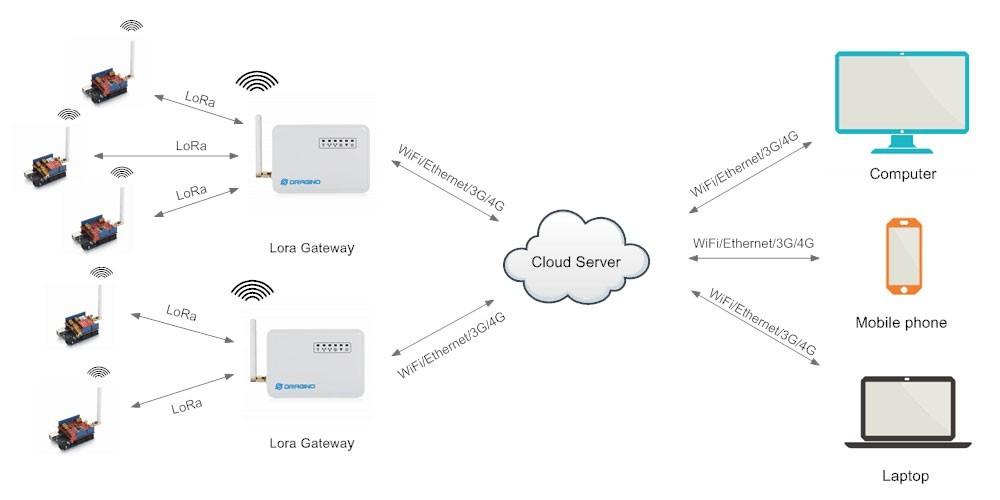
*ภาพที่ 3-2 NB-Iot AIS*

* + 1. **อุปกรณ์ต่อพ่วง**
* Soil Moisture Sensor Module v1
* Step up & Down (Buck&Boost) Super XL6009 5-32V to 1.25-32V 4A

## System Architecture

* + 1. **การเชื่อมต่อของ LoRa**

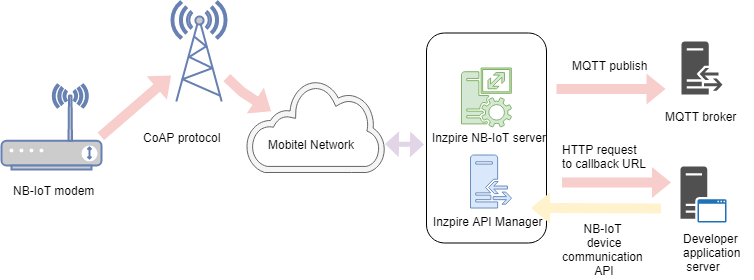
การเชื่อมต่อ สามารถอธิบายได้จากรูป 3-3 จากด้านซ้าย เราเรียกตัวที่นำไปใช้งานว่า End-Device ซึ่งเป็นตัวที่เราจะโปรแกรมลงไปใช้งาน ถัดมาเรียกว่า Gateway ซึ่งเปรียบเสมือนตัว Access Point ในระบบ Wireless ถัดมา Clound Server คือการเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต และภาพด้านขวาสุด คือ Applicaion Server ทำหน้าที่ในการร้องขอข้อมูล หรือสั่งการต่าง ๆ



*ภาพที่ 3-3 โครงข่ายของ Lora*

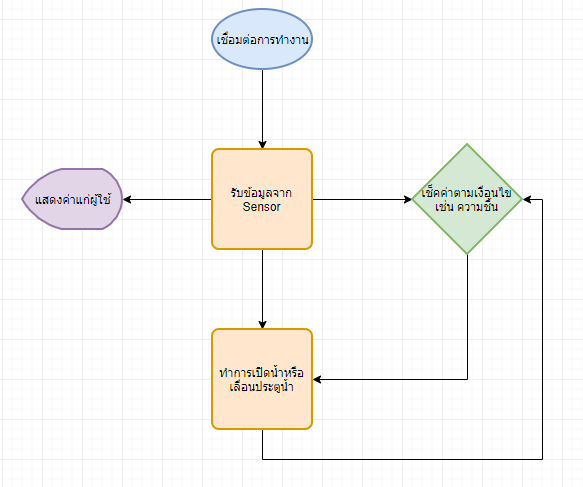
* + 1. **การเชื่อมต่อของ NB-Iot**

การเชื่อมต่อของ NB-Iot นั้นจากรูป 3-4 จากทางด้านซ้ายคือตัว NB-Iot ถัดมา เป็นเสาสัญญาณของซึ่งรับ-ส่งโดยใช้โปรโตคอล UDP/CoAP ถัดมาเป็นตัว Server เพื่อใช้รับค่าข้อมูลต่าง ๆ สุดท้ายทางด้านขวาสุด คือตัว Application Server ที่ใช้สั่งการหรือรับข้อมูล



*ภาพที่ 3-4 เครือข่ายของ LoRa*

## 3.3 System Design



*ภาพที่ 3-5 Flow Chart การทำงาน*

## System Implementation

* + 1. **อัพโหลดโปรแกรมให้ LoRa**

การอัพโหลดโปรแกรมเราสามารถทำได้โดยใช้ Arduino IDE ขั้นต่อไปหลังจากอัพโหลดแล้วนั้น เมื่อทำการทดสอบการเชื่อมต่อแล้วนั้นสามารถใช้งานได้โดย Join เข้าไปในระบบของ Cat ซึ่งมี Gateway ครอบคลุมจังหวัดภูเก็ต ผู้จัดทำได้ทำการทดลองใช้ Light sensor เพื่อทดลองรับและส่งค่าของข้อมูลต่าง ๆ ขึ้นไปในระบบพบว่าสามารถใช้งานได้

* + 1. **อัพโหลดโปรแกรมขึ้น NB-Iot**

ในการอัพโหลดโปรแกรมขึ้นไปนั้นสามารถดูข้อมูลได้ที่ [www.aismagellen.com](http://www.aismagellen.com) ได้ซึ่งเราสามารถนำข้อมูลที่ได้มาใช้ ซึ่งการเชื่อมต่อจะใช้โปรโตคอล UDP/CoAP เราสามารถนำข้อมูลออกมาได้ในรูปแบบของ JSON ซึ่งจะนำมาประมวลผลต่อไปใน React-Native

* + 1. **เขียนโครงสร้างแอพพลิเคชั่น**

ใช้ React-Native เพื่อสร้าง User Interface ในการเชื่อมต่อกับ Arduino Nb-Iot โดยสามารถใช้งานได้ทั้งระบบ Android และ ระบบ IOS โดย UI พื้นฐานจะมีการกดปิดน้ำ – เปิดน้ำ เพื่อให้ Solenoid Valve เปิด – ปิด ตามคำสั่งที่ส่งมาจากแอพพลิเคชั่น

* + 1. **สร้างเซิฟเวอร์เพื่อเป็นตัวกลางการติดต่อ**

เพื่อให้การเชื่อมต่อมีความเสถียรผู้จัดทำจึงใช้ Clound VPS ในการเป็นตัวกลางการติดต่อ โดยใช้ของ Amazon AWS ซึ่งมีทั้ง Linux, Windows โดยผู้จัดทำใช้ระบบปฏิบัติการ Linux เนื่องจากใช้ทรัพยากรณ์น้อย และ สนับสนุนการใช้งานด้านการพัฒนาโปรแกรมต่าง ๆ ได้ง่ายมากกว่าของ Windows

## 3.5 แผนการดำเนินงาน

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **การดำเนินงาน / ระยะเวลา** | **ปี พ.ศ.2562** | | | | | | | | | | | | | | | | | | | |
| **มกราคม** | | | | **กุมภาพันธ์** | | | | **มีนาคม** | | | | **เมษายน** | | | | **พฤษภาคม** | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| จัดซื้ออุปกรณ์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ทดลองเชื่อมต่อ Lora, NB-Iot |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| จัดซื้ออุปกรณ์เพิ่มเติม |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| เรียนรู้การเขียน React-Native |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| เริ่มต้นทดสอบ Application |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ตรวจสอบ Bug และ แก้ไข |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# บทที่ 4 ผลการดำเนินงานและสรุปผล

## 4.1 ผลการดำเนินงาน

สามารถใช้การเชื่อมต่อกับเซิฟเวอร์ตัวกลางได้และส่งข้อมูลระหว่างกัน ซึ่งในขั้นต่อไปผู้จัดทำจะทำการ เพิ่มข้อมูลในการรับและส่งรหะหว่างกันโดยเพิ่มส่วนของ Sensor วัดอุณหภูมิ, Sensor วัดแสง, ส่วนของตัวบอกระดับแบตเตอรี่เพื่อให้การใช้งานมีความละเอียดมากขึ้น

## 4.2 สรุปผลการทดลอง

การทดลองนี้เมื่อเราสามารถเชื่อมต่อได้แล้วสามารถนำข้อมูลไปใช้ได้อย่างหลากหลาย

**4.2.1 การรับและส่งผ่านข้อมูลบน CloundVPS Amazon AWS** สามารถเชื่อมต่อไปยัง loraiot.cattelecom.com ซึ่งเป็นตัวรับข้อมูลทางฝั่งของ Server

**4.2.2 NB-Iot** ใช้งานในการสั่ง Relay เพื่อเปิด-ปิดการทำงานของโมเดลจำลองเพื่อผ่านไฟได้

## 4.3 ปัญหาและอุปสรรค

แบตเตอร์รี่ที่เลี้ยงให้กับบอร์ดอาจใช้พลังงานมากกว่าที่ผู้ผลิตได้ระบุไว้ ดังนั้นควรหาแหล่งพลังงานอื่นทดแทน เช่น Solar Cell พลังงานลม

การทำงานในความชื้น อาจทำให้บอร์ดและ Sensor เสียหายได้

## 4.4 ข้อเสนอแนะ

ควรหาตัวบอร์ดหรือชิบควบคุมที่มีราคาถูก เนื่องจากยังเป็นเทคโนโลยีใหม่ อาจมีราคาแพง แต่ปัจจุบันเริ่มมีผู้ผลิตเพิ่มมากขึ้น ผู้นำไปใช้สามารถใช้ชิบตัวอื่นนอกจากที่ได้นำเสนอในโครงงานนี้

การตั้งค่าในการติดต่อกับ Nb-Iot บางครั้งข้อมูลที่ส่งระหว่างกันอาจะเกิดการสูญหายเนื่องจากโปรโตคอลที่ช้ เป็นโปรโตคอล UDP ซึ่งการใช้งานควรเพิ่มเงื่อนไขในการเช็คว่าข้อมูลรับและส่งสมบูรณ์หรือไม่

**บรรณานุกรม**

|  |  |
| --- | --- |
| [1] | ร. กอเจริญ, “เอ็นบีไอโอที,” ใน *Wireless Technologies for Internet of Things*, 2017, p. 14. |
| [2] | ร. กอเจริญ, “เอ็นบีไอโอที,” *Wireless Technologies for Internet of Things,* pp. 14, 15, 2017. |
| [3] | Murata Manufacturing Co., Ltd., “www.murata.com,” Murata, 16 10 2018. [ออนไลน์]. Available: https://wireless.murata.com/RFM/data/type\_abz.pdf. [%1 ที่เข้าถึง26 02 2019]. |
| [4] | Quectel Wireless Solutions Co., Ltd, “www.quectel.com,” quectel.com, 15 06 2017. [ออนไลน์]. Available: https://www.quectel.com/UploadImage/Downlad/Quectel\_BC95\_Hardware\_Design\_V1.3.pdf. [%1 ที่เข้าถึง02 26 2019]. |
| [5] | บ. โ. จำกัด, “http://www.arduino.codemobiles.com/,” 2 August 2018. [ออนไลน์]. Available: http://www.arduino.codemobiles.com/product/174/%E0%B9%80%E0%B8%8B%E0%B9%87%E0%B8%99%E0%B9%80%E0%B8%8B%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%8A%E0%B8%B7%E0%B9%89%E0%B8%99%E0%B9%83%E0%B8%99%E0%B8%94%. |
| [6] | ArduinoAll. All rights reserved, 21 March 2019. [ออนไลน์]. Available: https://www.arduinoall.com/product/2450/solar-cell-solar-panels-%E0%B9%82%E0%B8%8B%E0%B8%A5%E0%B8%B2%E0%B9%80%E0%B8%8B%E0%B8%A5%E0%B8%A5%E0%B9%8C-5-5v-110ma-0-6w-%E0%B8%82%E0%B8%99%E0%B8%B2%E0%B8%94-84x55mm. |

**ภาคผนวก**

|  |
| --- |
| **โค้ดสำหรับการเชื่อมต่อ LoRa** |
| #include "LoRaWAN.h"  #define DeviceAddr "BBBBAAAA" // LSB (AAAABBBB) // Device Address  #define NetworkSSKey "28AED22B7E1516A609CFABF715884F3C" // Network Session Key  #define AppSSKey "1628AE2B7E15D2A6ABF7CF4F3C158809" // Application Session Key  int cnt = 0;  void setup( void )  {  Serial.begin(9600);  LoRaWAN.begin(AS923);  if (!LoRaWAN.busy() && !LoRaWAN.joined()) {  Serial.println("start join ABP");  int result = LoRaWAN.joinABP(DeviceAddr, NetworkSSKey, AppSSKey);  if (result) {  Serial.println("Join success");  }  else {  Serial.println("Join failed");  }  }  }  void loop( void )  {  LoRaWAN.beginPacket();  LoRaWAN.write(0xef);  LoRaWAN.write(0xbe);  LoRaWAN.write(0xad);  LoRaWAN.write(0xde);  LoRaWAN.write(cnt++);  int result = LoRaWAN.endPacket();  if (result) {  Serial.print("DR: ");  Serial.print(LoRaWAN.getDataRate());  Serial.print(", TxPower: ");  Serial.print(LoRaWAN.getTxPower());  Serial.print("dbm, UpLinkCounter: ");  Serial.print(LoRaWAN.getUpLinkCounter());  Serial.print(", DownLinkCounter: ");  Serial.println(LoRaWAN.getDownLinkCounter());  Serial.print("Payload:: ");  Serial.println(cnt);  Serial.print("============================");  Serial.println("");  }  else {  Serial.println("Send package failed");  }  delay(10000);  } |

|  |
| --- |
| **โค้ดสำหรับการเชื่อมต่อ NB-Iot (เชื่อมต่อกับ aismagellan.io)** |
| int sensorPin = A0; // select the input pin for the potentiometer  int sensorValue = 0; // variable to store the value coming from the sensor  void setup() {  // declare the ledPin as an OUTPUT:  Serial.begin(9600);  }  void loop() {  // read the value from the sensor:  sensorValue = analogRead(sensorPin);  delay(1000);  Serial.print("sensor = " );  Serial.println(sensorValue);  } |

|  |
| --- |
| **โค้ดสำหรับการเชื่อมต่อ ระหว่าง NB-IoT กับ Server (ฝั่ง Server)** |
| var dgram = require("dgram"); var server = dgram.createSocket("udp4");  server.on("error", function (err) {  console.log("server error:\n" + err.stack);  server.close(); });  server.on("message", function (msg, rinfo) {  console.log("server got: " + msg + " from " + rinfo.address + ":" + rinfo.port);  var ack = new Buffer("Hello ack");  server.send(ack, 0, ack.length, rinfo.port, rinfo.address, function(err, bytes) {  console.log("sent ACK.");  });  });  server.on("listening", function () {  var address = server.address();  console.log("server listening " + address.address + ":" + address.port); });  server.bind({  address: '0.0.0.0',  port: 7000,  exclusive: true }); |

|  |
| --- |
| **โค้ดสำหรับการเชื่อมต่อ ระหว่าง NB-iot กับ Server (ฝั่ง Client)** |
| #include "AIS\_NB\_BC95.h"  String serverIP = "IP address"; <--- IP ของ Google Cloud Platform String serverPort = "7000";  String udpData = "HelloWorld";  AIS\_NB\_BC95 AISnb;  const long interval = 5000; //millisecond unsigned long previousMillis = 0;  void setup() {   AISnb.debug = true;    Serial.begin(9600);    AISnb.setupDevice(serverPort);  String ip1 = AISnb.getDeviceIP();   delay(1000);    pingRESP pingR = AISnb.pingIP(serverIP);  previousMillis = millis();  } void loop() {   unsigned long currentMillis = millis();  if (currentMillis - previousMillis >= interval)  {  UDPSend udp = AISnb.sendUDPmsgStr(serverIP, serverPort, udpData);  previousMillis = currentMillis;  }  UDPReceive resp = AISnb.waitResponse(); } |

|  |
| --- |
| **โค้ดสำหรับการเชื่อมต่อ ระหว่าง Line Messaging กับ Server (ฝั่ง Client)** |
| const express = require('express')  const bodyParser = require('body-parser')  const axios = require('axios')  const app = express()  const port = process.env.PORT || 5006  const LINE\_SECRETE\_TOKEN = require('../01\_backend\_config').LINE\_SECRETE\_TOKEN  const NBserver = require('./nbserver')  require('dotenv').config()  async function line() {      console.log(`LineServerStartAtPort ${port}`)      app.use(bodyParser.urlencoded({ extended: false }))      app.use(bodyParser.json())      app.post('/webhook', (req, res) => {          let reply\_token = req.body.events[0].replyToken          let msg = req.body.events[0].message.text          reply(reply\_token, msg)          res.sendStatus(200)      })      app.listen(port)      async function reply(reply\_token, msg) {          let headers = {              'Content-Type': 'application/json',              'Authorization': LINE\_SECRETE\_TOKEN //ใส่ Token ที่ได้จาก Line devoloper          }          let resMessage = async (msg) => {              if (await msg == 'เปิดแจ้งเตือน' || await msg == "1") {                  return await onBot(true)              }              else if (await msg == 'ปิดแจ้งเตือน' || await msg == "2") {                  return await offBot(false)              }              else if (await msg == 'สวัสดี'){                  return await 'สวัสดีมีอะไรให้เราช่วย'              }              else if (await msg == 'ดูอุณหภูมิ' || await msg == "3"){                  return await getLastData()              }              else if (await msg == 'ดูความชื้น' || await msg == "4"){                  return await "ยังไม่เปิดใช้งาน"              }              else if (await msg == 'ปิดน้ำ' || await msg == "6") {                  await NBserver.sendSw(false)                  return await "ปิดน้ำแล้ว";              }              else if (await msg == 'เปิดน้ำ' || await msg == "7") {                  await NBserver.sendSw(true)                  return await "เปิดน้ำแล้ว";              }              else if (await msg == 'ดูคำสั่ง' || await msg == 'help' || await msg == '?') {                  return await `1. ปิดการแจ้งเตือน \n2. เปิดการแ้งเตือน \n3. ดูอุณหภูมิ \n4. ดูความชื้น \n5. ดูรูป \n6. ปิดน้ำ \n7. เปิดน้ำ`              }              else {                  return await 'โปรดพิมพ์ว่า "?" หรือ "ดูคำสั่ง" เพื่อดูคำสั่งทั้งหมด'              }          }          onBot = async (command) => {              await console.log(command)          }          offBot = async (command) => {              await console.log(command)          }          getLastData = async () => {              const get = await axios.get('http://localhost:5004/api/getData')              console.log(await get.data[0].data)              return await get.data[0].data          }          let body = await JSON.stringify({              replyToken: reply\_token,              messages: [{                  type: 'text',                  text: await resMessage(msg)              }]          })          axios({              method: 'POST',              headers: headers,              data: body,              url: 'https://api.line.me/v2/bot/message/reply'          })      }  }  module.exports = { line } |