

การพัฒนาโมดูล ZIGBEE สำหรับมิเตอร์อัจฉริยะ
DEVELOPMENT OF ZIGBEE MODULE FOR SMART METER



ปริญญาบัตรฉบับนี้เป็นส่วนหนึ่งของงานศึกษาและงานวิจัยของปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี กรุงเทพมหานคร

ปีการศึกษา 2557

การพัฒนาโมดูล ZIGBEE สำหรับมิเตอร์อัจฉริยะ
DEVELOPMENT OF ZIGBEE MODULE FOR SMART METER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

DEVELOPMENT OF ZIGBEE MODULE FOR SMART METER



THIS PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE BACHELOR DEGREE IN ELECTRICAL ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2014

ปีการศึกษา 2557

การพัฒนาโมดูล ZIGBEE สำหรับมิเตอร์อัจฉริยะ
DEVELOPMENT OF ZIGBEE MODULE FOR SMART METER



อาจารย์ที่ปรึกษา

รศ.ดร.สมยศ เกียรติวนิชวิไล

ผศ.ดร.ศิริเดช บุญแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรืออ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2557

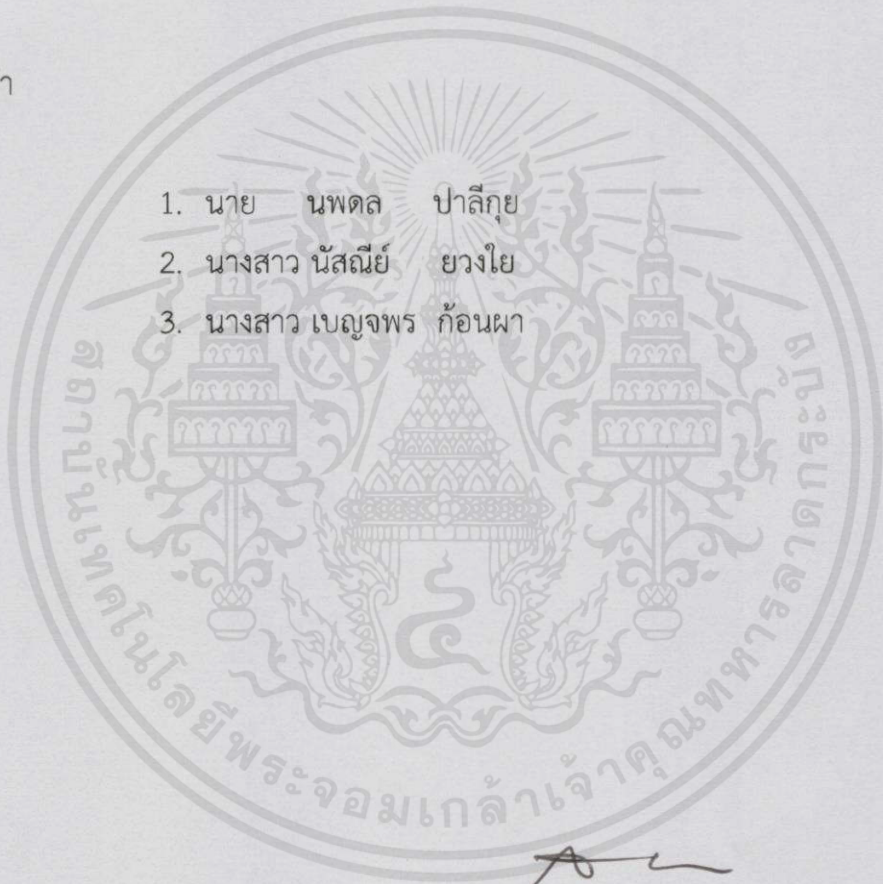
ภาควิชาวิศวกรรมไฟฟ้า


คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เรื่อง การพัฒนาโมดูล ZIGBEE สำหรับมิเตอร์อัจฉริยะ

ผู้จัดทำ

1. นาย นพดล ปาลีภัย
2. นางสาว นัสณีย์ ยวงโย
3. นางสาว เบญจพร ก้อนผา




..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.สมยศ เกียรติวนิชวิไล)

 (11๕)
..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.ศิริเดช บุญแสง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโมดูล Zigbee สำหรับมิเตอร์อัจฉริยะ

นาย นพดล ปาสีกุญ

นางสาว นัสณีย์ ยวงโย

นางสาว เบญจพร ก้อนผา

รศ.ดร. สมยศ เกียรติวนิชวิไล อาจารย์ที่ปรึกษา

ผศ.ดร. ศิริเดช บุญแสง อาจารย์ที่ปรึกษา

ปีการศึกษา 2557

บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อนำระบบเครือข่ายไร้สายมาประยุกต์ใช้กับมิเตอร์อัจฉริยะโดยได้ทำการออกแบบและพัฒนาโมดูล Zigbee เพื่อความสะดวกในการส่งข้อมูลจากมิเตอร์อัจฉริยะมาแสดงผลยังผู้ใช้ไฟฟ้า อีกทั้งยังเป็นการลดค่าใช้จ่ายในการเดินสายทำให้สามารถนำข้อมูลมาวิเคราะห์ตรวจสอบได้อย่างถูกต้องและมีประสิทธิภาพ สำหรับปริญญาานิพนธ์ฉบับนี้ได้แบ่งส่วนระบบการทำงานเป็น 3 ส่วนด้วยกัน คือ ส่วนโหนดอุปกรณ์ปลายทาง (End Device Node) ส่วนโหนดเราเตอร์ (Router Node) และส่วนโหนดโคออดิเนเตอร์ (Coordinator Node) เริ่มจากการจำลองการส่งข้อมูลของมิเตอร์อัจฉริยะโดยให้โมดูล Zigbee ในภาคส่งนี้เป็นโหนดอุปกรณ์ปลายทาง (End Device Node) ส่งข้อมูลผ่านโมดูล Zigbee ตัวกลางที่เป็นโหนดเราเตอร์ (Router Node) เพื่อใช้ในการสร้างเครือข่ายส่งข้อมูลต่อไปให้ โมดูล Zigbee ในภาครับซึ่งทำหน้าที่เป็นโหนดโคออดิเนเตอร์ (Coordinator Node) ไปแสดงผลยังหน้าจอกอมพิวเตอร์ของผู้ใช้งาน โดยในการส่งข้อมูลของ Zigbee นั้นจะใช้โหมดคำสั่งแบบ API ซึ่งข้อมูลและคำสั่งจะอยู่ในรูป Frame-Based ทั้งหมดจึงทำให้การส่งข้อมูลภายในเครือข่ายมีความถูกต้องแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Development of Zigbee Modle for Smart Meter

Noppadon Paleekui

Nasanee Youngyai

Benjaporn Konpha

Assoc. Prof. Dr. Somyot Kiatwanitwilai Advisor

Asst. Prof. Dr. Silidech Boonsang Advisor

Year 2014

ABSTRACT

This project proposes to apply a wireless design of smart meter and developing Zigbee Module for transmitting data. This is convenient to transmit data from smart meter to the user interface. Moreover, It can reduce cost in wiring that are suitable and also provide more accuracy in data analyzing and high efficiency. This project is consisted of 3 operations which is End Device Node, Router Node and Coordinator Node. By initiating data simulations on smart meter and assign transmitting Zigbee module as an End Device Node then transfer the data to the Central Zigbee Module which is assigned as Router Node to form a transferring network and pass the data to the Receiving Zigbee which is assigned as Coordinator Node and visualize the data into the user's computer. All Zigbee's data transferring is being transfer in API mode which all data and command are in Frame-Based format hence more accuracy in data transferring across the network.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์สำเร็จลุล่วงไปได้ด้วยดีต้องขอขอบพระคุณ ผศ.ดร.สมยศ เกียรติวนิชวิไล และผศ.ดร.ศิริเดช บุญแสง อาจารย์ที่ปรึกษาปริญญาานิพนธ์ ที่ช่วยผลักดันและให้โอกาสโครงการนี้ได้ทำสำเร็จเป็นจริงขึ้นมา โดยคอยช่วยเหลือ ชี้แนะแนวทางการทำงาน และให้คำปรึกษาข้อแนะนำที่เป็นประโยชน์ต่อโครงการมาโดยตลอด ตลอดจนตรวจและแก้ไขปริญญาานิพนธ์จนสำเร็จเรียบร้อยเป็นอย่างดี

ขอขอบคุณรุ่นพี่ และเพื่อนๆที่ให้คำแนะนำและชี้แนะแนวทางในการทำงาน และที่ขาดไม่ได้คือเพื่อนๆ พี่ๆภาควิชาวิศวกรรมไฟฟ้า วิศวกรรมอิเล็กทรอนิกส์ วิศวกรรมโทรคมนาคมและชุมชนโรบอทที่ช่วยชี้แนะแนวทาง ให้คำแนะนำและร่วมแก้ไขปัญหาเสมอ ไม่ว่าปัญหานั้นจะเล็กหรือใหญ่ก็ตามทำให้โปรเจกต์สามารถเดินหน้าไปด้วยดีมาโดยตลอด

นอกจากนั้นผู้จัดทำต้องขอขอบคุณ พี่ตัวเล็ก เจ้าหน้าที่ห้องปฏิบัติการที่ให้เบิกใช้เครื่องมือ ตลอดจน พี่วรรณ เจ้าหน้าที่ห้องธุรการที่ให้คำแนะนำเกี่ยวกับการเบิกใบเสร็จของโครงการ

ท้ายนี้คณะผู้จัดทำ ขอกราบขอบพระคุณ บิดามารดา และครอบครัวซึ่งให้การสนับสนุนในด้านการเงินและให้กำลังใจอีกทั้งเพื่อนๆ พี่ๆ น้องๆ ที่ให้ความช่วยเหลือในด้านการจัดหาอุปกรณ์ต่างๆ ในการทำโครงการครั้งนี้จนสำเร็จได้ด้วยดี จึงขอขอบคุณไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	1
1.3 ขอบเขตและข้อกำหนดของโครงการ	2
1.4 วิธีการที่ใช้ในโครงการ	2
1.5 แผนการดำเนินโครงการ	3
1.6 ประโยชน์ที่ได้รับจากการวิจัย	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 ZIGBEE	5
2.1.1 เทคโนโลยี Zigbee	5
2.1.2 ชนิดของอุปกรณ์ Zigbee	7
2.1.3 การประยุกต์ใช้งาน	8
2.1.4 โครงสร้างของโพรโทคอล Zigbee	8
2.1.5 ขั้นตอนการทำงานของโพรโทคอล Zigbee	10
2.1.6 Topology ในการใช้งาน Zigbee Network	11
2.1.7 การส่งข้อมูลของ Zigbee	13
2.1.8 การติดต่อสื่อสารของ Zigbee ผ่านทางพอร์ตอนุกรม	15
2.2 ลักษณะทั่วไปของโมดูล XBEE	16
2.3 การสื่อสารข้อมูล	16
2.3.2 ข้อกำหนด หรือโพรโทคอล	19
2.3.3 มาตรฐาน	20
2.4 การสื่อสารข้อมูลแบบอนุกรม	20
2.4.1 UART	22
2.4.2 แนวคิดวิธีเชื่อมต่อระหว่างอุปกรณ์	23
2.4.3 TTL (Transistor-Transistor Logic)	23

เอกสารนี้เป็นเอกสารที่ส่งมอบให้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.4.4 USB to TTL Converter Module	24
2.5 มิเตอร์อัจฉริยะ (SMART METER)	25
2.5.1 ส่วนประกอบของมิเตอร์อัจฉริยะ	26
2.5.2 ข้อมูลจำเพาะของโปรโตคอลมิเตอร์ไฟฟ้าที่ใช้งาน	26
2.6 ARDUINO	28
2.6.1 ข้อมูลเบื้องต้นของ Arduino	28
2.6.2 โปรแกรมภาษาของ Arduino	28
2.6.3 Arduino Uno R3	29
2.7 แหล่งจ่ายไฟฟ้า (POWER SUPPLY)	30
2.7.1 หม้อแปลงไฟฟ้า (Transformer)	30
2.7.2 วงจรเรียงกระแส (Rectifier)	31
2.7.3 วงจรกรองกระแส (Filter)	31
2.7.4 วงจรรักษาระดับแรงดัน (Voltage Regulators)	32
2.8 ไอซีชิฟเฟอร์เบอร์ 74LS245	33
2.8.1 หลักการทำงานของไอซี 74LS245	34
2.8.2 การทำงานของขาควบคุมของไอซี 74LS245	34
บทที่ 3 การออกแบบระบบการทำงาน	35
3.1 การออกแบบโครงสร้างการทำงานของระบบ	35
3.2 หลักการออกแบบวงจร	36
3.2.1 การออกแบบวงจรเชื่อมต่อโมดูลZigbeeทำหน้าที่เป็นโหนดอุปกรณ์ปลายทาง	36
3.2.2 การออกแบบวงจรเชื่อมต่อโมดูลZigbeeทำหน้าที่เป็นโหนดเรอเตอ์	38
3.2.3 การออกแบบวงจรเชื่อมต่อโมดูลZigbeeทำหน้าที่เป็นโหนดโคออดิเนเตอร์	40
3.3 หลักการออกแบบโปรแกรม	41
3.3.1 หลักการออกแบบโปรแกรมฝั่งส่ง	41
3.3.2 หลักการออกแบบโปรแกรมฝั่งรับ	46
บทที่ 4 การทดลองและผลการทดลอง	50
4.1 การทดลองเกี่ยวกับปัจจัยที่มีผลต่อการส่งข้อมูล	50
4.1.1 การทดลองระยะทางการส่งข้อมูลระหว่างโมดูลZigbee 2ตัว	50
4.1.2 การทดลองระยะทางการส่งข้อมูลเป็นเครือข่ายโดยใช้โหมดAPI	53
4.2 การทดลองการจัดการเครือข่ายการติดต่อสื่อสารของโมดูลZIGBEEโดยใช้โหมดAPI	55
4.2.1 การทดลองเชื่อมต่อเครือข่ายของโมดูลZigbee แบบ Cluster Tree (Tree)	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดกแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสาร ทุกครั้งที่มีการนำ ไปใช้

สารบัญ(ต่อ)

	หน้า
4.2.2 การทดสอบเชื่อมต่อเครือข่ายของโมดูลZigbee แบบ Star	57
4.2.3 การทดสอบเชื่อมต่อเครือข่ายของโมดูลZigbee แบบ Mesh	57
4.2.4 การทดสอบการใช้โปรแกรมแสดงข้อมูลของมิเตอร์ไฟฟ้าบนหน้าจocomพิวเตอร์	58
บทที่ 5 สรุปผลการทดลอง	60
5.1 สรุปผลการทดลอง	60
5.1.1 ส่วนโหนดอุปกรณ์ปลายทาง	60
5.1.2 ส่วนโหนดเรดเดอร์	60
5.2 วิเคราะห์ผลการทดลอง	60
5.3 ปัญหาที่พบในระหว่างดำเนินโครงการ	60
5.4 แนวทางการแก้ไขปัญหาและพัฒนา	61
เอกสารอ้างอิง	62
ภาคผนวก	64
ภาคผนวก ก บทความทางวิชาการ	65
ภาคผนวก ข โปรแกรมการทำงาน	70
ภาคผนวก ค เอกสารกำกับอุปกรณ์อิเล็กทรอนิกส์	91
ภาคผนวก ง ข้อมูลโปรโตคอลของมิเตอร์ไฟฟ้า	116
ประวัติผู้เขียน	129

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ปริมาณการใช้Throughputของมาตรฐานการสื่อสารไร้สายแบบต่างๆ	6
2.2 โมเดลการส่งถ่ายข้อมูลแบบStar(ซ้าย)และPeer-to-Peer(ขวา)	7
2.3 ตัวอย่างการสร้างเครือข่ายของอุปกรณ์ Zigbee	9
2.4 โครงสร้าง Zigbeeโดยแบ่งเป็น layer	9
2.5 ชั้นโพรโตคอลการรับส่งข้อมูล	11
2.6 Star (Broadcast) Network	12
2.7 Cluster Tree (Tree) Network	12
2.8 Mesh Network	13
2.9 รูปแบบของคำสั่ง AT (AT Command)	14
2.10 รูปแบบของคำสั่ง API (API Command)	15
2.11 การต่อโมโครคอนโทรลเลอร์กับZigbee	16
2.12 องค์ประกอบ5ประการของระบบสื่อสารข้อมูล	18
2.13 แผนผังแสดงทิศทางการไหลของข้อมูลแบบ Simplex	18
2.14 แผนผังแสดงทิศทางการไหลของข้อมูลแบบ Half-Duplex ซึ่งข้อมูลสามารถไหลได้ทั้งทิศทางจากซ้ายไปขวาและขวาไปซ้าย ไม่พร้อมกันในแต่ละช่วงเวลา	19
2.15 แผนผังแสดงทิศทางการไหลของข้อมูล Full-Duplex ซึ่งข้อมูลสามารถไหลได้ทั้งทิศทางจากซ้ายไปขวาและขวาไปซ้าย ได้พร้อมกันในแต่ละช่วงเวลา	19
2.16 การสื่อสารข้อมูลแบบขนานและอนุกรม	21
2.17 การสื่อสารข้อมูลแบบ Synchronous	22
2.18 การรับส่งข้อมูลแบบ Asynchronous	23
2.19 การเชื่อมต่อระหว่างบอร์ด SUN7 เพื่อส่ง Data หากันด้วย UART	23
2.20 ระดับแรงดัน TTL	24
2.21 USB to TTL Converter Module	24
2.22 ขาของ USB to TTL Converter Module	25
2.23 ชั้นของโพรโตคอล	27
2.24 โครงสร้างข้อความRS-485	27
2.25 โครงสร้างโพรโตคอล MEATH	27
2.26 Arduino Uno R3	29
2.27 Block Diagramแหล่งจ่ายไฟฟ้า	30
2.28 Center-tapped Full-wave Rectifier	31
2.29 Full wave Bridge Rectifier	31

สารบัญรูป(ต่อ)

รูปที่	หน้า
2.30 บล็อกไดอะแกรมของแหล่งจ่ายไฟตรงคงค่าแรงดัน	32
2.31 วงจรRegulator	33
2.32 โครงสร้างของ IC 74LS245	34
3.1 โครงสร้างการทำงานของระบบ	35
3.2 วงจรไฟเลี้ยงของโมดูลZigbeeทำหน้าที่เป็นโหนดอุปกรณ์ปลายทาง	37
3.3 วงจรการทำงานของโมดูลZigbeeทำหน้าที่เป็นโหนดอุปกรณ์ปลายทาง	37
3.4 วงจรไฟเลี้ยงโมดูลZigbeeทำหน้าที่เป็นโหนดเรเตอร์	38
3.5 วงจรการทำงานของโมดูลZigbeeทำหน้าที่เป็นโหนดเรเตอร์	39
3.6 วงจรไฟเลี้ยงของโมดูลZigbeeทำหน้าที่เป็นโหนดโคออดิเนเตอร์	40
3.7 วงจรการทำงานของโมดูลZigbeeทำหน้าที่เป็นโหนดโคออดิเนเตอร์	41
3.8 วงจรการทำงานของโมดูลZigbeeทำหน้าที่เป็นโหนดโคออดิเนเตอร์	41
3.9 ตัวอย่างข้อมูลฝั่งส่งที่สร้างขึ้นมาในรูปแบบโปรโตคอลMEATH	42
3.10 ตัวอย่างPacketข้อมูลในโหมดAPI ของฝั่งส่ง(มิเตอร์ไฟฟ้าตัวที่1)	42
3.11 ตัวอย่างPacketข้อมูลในโหมดAPI ของฝั่งส่ง(มิเตอร์ไฟฟ้าตัวที่2)	44
3.12 Flowchart แสดงการทำงานของโปรแกรมจำลองการสื่อสารข้อมูลของมิเตอร์ไฟฟ้า	45
3.13 Packetข้อมูลที่ได้รับจากมิเตอร์ไฟฟ้าตัวที่1ในโหมดAPI ของฝั่งรับ	46
3.14 Packetข้อมูลที่ได้รับจากมิเตอร์ไฟฟ้าตัวที่2ในโหมดAPI ของฝั่งรับ	46
3.15 Flowchart แสดงการทำงานของโปรแกรมฝั่งรับ (ศูนย์กลางการควบคุมและแสดงข้อมูล)	49
4.1 กราฟความสัมพันธ์ระหว่างระยะทางกับระดับความเข้มของสัญญาณเฉลี่ย	51
4.2 กราฟระดับความเข้มของสัญญาณและเปอร์เซ็นต์ความถูกต้องในการรับ-ส่งข้อมูลที่ระยะ30เมตร	52
4.3 กราฟระดับความเข้มของสัญญาณและเปอร์เซ็นต์ความถูกต้องในการรับ-ส่งข้อมูลที่ระยะ73เมตร	53
4.4 โมดูลZigbeeที่ทำหน้าที่เป็นโหนดโคออดิเนเตอร์	55
4.5 โมดูลZigbeeที่ทำหน้าที่เป็นโหนดเรเตอร์	56
4.6 โมดูลZigbeeที่ทำหน้าที่เป็นโหนดอุปกรณ์ปลายทาง	56
4.7 การเชื่อมต่อโมดูลZigbeeในส่วนควบคุมและแสดงผล(โหนดโคออดิเนเตอร์)	58
4.8 โปรแกรมแสดงข้อมูลการใช้ไฟฟ้าของmeter1(โหนดอุปกรณ์ปลายทาง)	59
4.9 โปรแกรมแสดงข้อมูลการใช้ไฟฟ้าของmeter2 (โหนดเรเตอร์)	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทางสน อื่นทั้งที่มีให้คิดเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 การเปรียบเทียบคุณสมบัติของเทคโนโลยีไร้สาย	5
2.2 กำหนดอินพุตของ IC 74LS245	34
3.1 รายละเอียดPacket ข้อมูลในโหมดAPI ของอุปกรณ์ปลายทาง	43
4.1 ผลการวัดประสิทธิภาพการรับส่งข้อมูลของโมดูลZigbeeจากระยะทาง โดยไม่ผ่านสิ่งกีดขวาง	51
4.2 ผลการวัดประสิทธิภาพการรับส่งข้อมูลของโมดูลZigbeeจากระยะทางแบบเป็นเครือข่าย	54



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันได้มีการขยายการติดตั้งมิเตอร์อัจฉริยะ (Smart Meter) เพิ่มมากขึ้น โดยมีเตอร์อัจฉริยะเป็นเทคโนโลยีที่ทำให้ผู้ใช้ไฟฟ้ามีความสะดวกสบายเนื่องจากผู้ใช้ไฟฟ้าสามารถติดต่อสื่อสารโดยตรงกับระบบควบคุมของผู้ผลิตไฟฟ้าได้บ้อยเท่าที่ต้องการ ทำให้ผู้ใช้ไฟฟ้าสามารถอ่านหรือคำนวณค่าใช้จ่ายในการใช้ไฟฟ้าได้ตลอดเวลา ซึ่งส่งผลให้ผู้ใช้ไฟฟ้าสามารถบริหารจัดการการใช้ไฟฟ้าให้เหมาะสมเพื่อลดค่าใช้จ่ายในการใช้ไฟฟ้าลงได้ สำหรับช่องทางการสื่อสารระหว่างมิเตอร์อัจฉริยะกับผู้ใช้งานสามารถเลือกได้จากหลากหลายรูปแบบโดยจะนิยมใช้เทคโนโลยีในการติดต่อสื่อสารข้อมูลแบบไร้สายเพื่อความสะดวกรวดเร็ว และเป็นการลดค่าใช้จ่ายในการเดินสายไฟอีกด้วย

เทคโนโลยีการสื่อสารในปัจจุบันได้เข้ามามีบทบาทสำคัญอย่างยิ่งในชีวิตประจำวัน และได้มีการพัฒนาอย่างต่อเนื่องเพื่อให้มีประสิทธิภาพมากยิ่งขึ้น โดยมีการพัฒนาการส่งข้อมูลระหว่าง 2 แห่ง จากการส่งข้อมูลเชื่อมต่อผ่านสาย พัฒนามาเป็นการส่งข้อมูลแบบไร้สายแทน จึงทำให้ผู้ใช้งานมีความสะดวกสบาย และยังช่วยลดค่าใช้จ่ายในการเดินสายอีกด้วย ด้วยเหตุนี้จึงมีการนำเทคโนโลยีการสื่อสารไร้สายมาใช้เป็นตัวกลางในการส่งผ่านข้อมูล ซึ่งในอนาคตมีแนวโน้มที่จะมีการผลิตและการใช้งานอย่างแพร่หลาย ทั้งนี้เทคโนโลยีการสื่อสารข้อมูลแบบไร้สาย ที่เรารู้จักและคุ้นเคยในชีวิตประจำวันได้แก่ ระบบโทรศัพท์เคลื่อนที่ไร้สาย, ระบบการสื่อสารผ่านคลื่น RF, จีพีเอสไวเลส (GPRS Wireless), Wi-Fi, บลูทูธ (Bluetooth), Zigbee, ไมโครเวฟ (Microwave) และอินฟราเรด (Infrared) เป็นต้น ซึ่งการสื่อสารข้อมูลแบบไร้สายในแต่ละรูปแบบก็จะมีคุณลักษณะโดดเด่นเฉพาะตัวขึ้นที่อยู่กับการใช้งาน เมื่อเราจะเลือกชนิดของการสื่อสารข้อมูลแบบไร้สายมาใช้ เราต้องพิจารณาถึงคุณลักษณะและองค์ประกอบต่างๆให้เหมาะสมกับการใช้งาน โดย Zigbee ถือเป็นโมดูลไร้สายประเภทหนึ่งที่มีประสิทธิภาพสูง มีราคาถูก ใช้พลังงานน้อย จึงทำให้มีอายุการใช้งานแบตเตอรี่ที่ยาวนาน ดังนั้นจึงมีแนวคิดที่จะนำเสนอการพัฒนาโมดูล Zigbee สำหรับมิเตอร์อัจฉริยะ โดยจะนำ Zigbee มาพัฒนาในการรับส่งข้อมูลระหว่างมิเตอร์อัจฉริยะกับผู้ใช้งาน

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อศึกษาการใช้งาน Zigbee

1.2.2 เพื่อวิเคราะห์และออกแบบพัฒนาโมดูล Zigbee ในการสร้างเครือข่ายไร้สาย

1.2.3 เพื่อจำลองการส่งข้อมูลของมิเตอร์อัจฉริยะโดยใช้ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแบบสิ่งนี้ออกไปและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.4 เพื่อออกแบบโปรแกรมแสดงข้อมูลที่ได้รับบนหน้าจคอมพิวเตอร์สำหรับ
ผู้ใช้งาน

1.3 ขอบเขตและข้อกำหนดของโครงการ

1.3.1 สามารถวิเคราะห์และออกแบบวงจรที่ใช้ในการรับส่งข้อมูลโดยผ่านอุปกรณ์
Zigbee ได้

1.3.3 สามารถทำการเชื่อมต่อเพื่อให้เกิดการรับส่งข้อมูลระหว่างอุปกรณ์ Zigbee
และสร้างเครือข่ายของ Zigbee ได้

1.3.2 สามารถจำลองการสื่อสารข้อมูลของมิเตอร์อัจฉริยะให้กับ Zigbee ได้

1.3.4 สามารถสร้างโปรแกรมแสดงข้อมูลให้ผู้ใช้งานทราบได้

1.4 วิธีการที่ใช้ในโครงการ

1.4.1 ศึกษาข้อมูลเทคโนโลยีการสื่อสารไร้สายแบบ Zigbee จากหนังสือ บทความ
ปริญญาานิพนธ์ และงานวิจัยต่างๆ

1.4.2 ออกแบบวงจรโมดูล Zigbee ที่สามารถนำไปใช้งานกับมิเตอร์ไฟฟ้าอัจฉริยะได้

1.4.3 จัดเตรียมอุปกรณ์และเครื่องมือที่เกี่ยวข้องกับงานวิจัย ตลอดจนศึกษา
วิธีการใช้งานอุปกรณ์และเครื่องมือต่างๆอย่างถูกวิธี และประกอบสร้างโมดูล Zigbee เพื่อใช้ในการ
ทดสอบประสิทธิภาพการส่งข้อมูลของ Zigbee รวมทั้งแก้ไขข้อบกพร่องของชิ้นงาน

1.4.4 ทดสอบประสิทธิภาพการรับส่งข้อมูล ผ่านโปรแกรม X-CTU

1.4.5 เขียนโปรแกรมเพื่อจำลองการส่งข้อมูลของมิเตอร์อัจฉริยะและสร้างโปรแกรม
เพื่อใช้แสดงข้อมูลสำหรับผู้ใช้งาน

1.4.6 สร้างเครือข่าย Zigbee ในโหมด API โดยให้โมดูล Zigbee ตัวกลางเป็น Router
โมดูล Zigbee ที่ต่ออยู่กับมิเตอร์อัจฉริยะเป็น End Device และโมดูลที่ต่ออยู่กับผู้ใช้งานเป็น
Coordinator

1.4.7 ทดสอบการรับส่งข้อมูลของเครือข่าย Zigbee ตรวจสอบและแก้ไขข้อผิดพลาด

1.4.8 สรุปผลการทดลอง

1.4.9 จัดทำเอกสารรูปเล่มปริญญาานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 แผนการดำเนินงาน

ลำดับ	ขั้นตอนการดำเนินงาน	ปี 2557					ปี 2558					
		ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	
1	ศึกษา ค้นคว้าและรวบรวมข้อมูล	←→										
2	ออกแบบวงจรโมดูล Zigbee		←→									
3	จัดหาซื้ออุปกรณ์และประกอบสร้างโมดูล Zigbee			←→								
4	ทดสอบประสิทธิภาพการรับส่งข้อมูล			←→								
5	จัดทำเอกสารและนำเสนอโครงการครั้งที่1	←→										
6	เขียนโปรแกรมจำลองการส่งข้อมูลของมิเตอร์อัจฉริยะ						←→					
7	สร้างโปรแกรมเพื่อใช้แสดงข้อมูลในส่วนผู้ใช้งาน						←→					
8	สร้างเครือข่ายZigbeeในโหมดAPI						←→					
8	ตรวจสอบ แก้ไขการรับส่งข้อมูลของเครือข่ายโมดูลZigbeeในโหมดAPI						←→					
9	จัดทำเอกสารและนำเสนอโครงการครั้งที่2						←→					

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ประโยชน์ที่ได้รับจากการวิจัย

1.6.1 โครงการนี้เป็นการบูรณาการระหว่างวิทยาการในด้านต่างๆทางวิศวกรรมไฟฟ้าเช่น ระบบสื่อสารโทรคมนาคม อิเล็กทรอนิกส์ และระบบไฟฟ้ากำลัง ซึ่งก่อให้เกิดประโยชน์แก่ผู้ค้นคว้าวิจัยในด้านที่กล่าวมา

1.6.2 สามารถนำเอาเทคโนโลยีการสื่อสารข้อมูลแบบไร้สาย Zigbee มาพัฒนาต่อยอดให้เกิดประโยชน์ในด้านการสื่อสาร และการสร้างเครือข่ายโดยใช้ Zigbee ให้เกิดประสิทธิภาพสูงสุด

1.6.3 สามารถพัฒนาทักษะในการเขียนโปรแกรมภาษาซี

1.6.4 สามารถนำเอาเทคโนโลยีการสื่อสารไร้สาย Zigbee มาประยุกต์ใช้กับไมโครอัจฉริยะได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 Zigbee

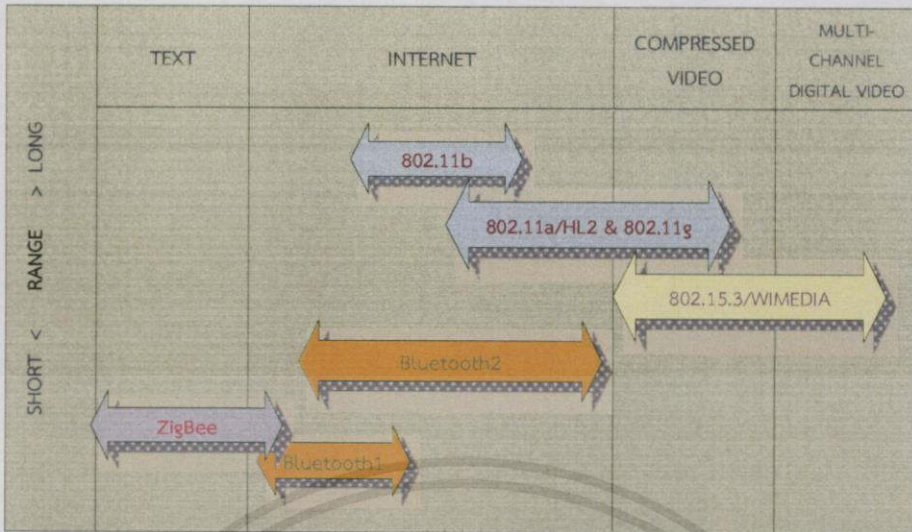
2.1.1 เทคโนโลยี Zigbee

Zigbee เป็นเทคโนโลยีสื่อสารที่ออกแบบขึ้นสำหรับการสื่อสารในเครือข่ายแบบไร้สาย โดยที่ Zigbee ได้มาจากการเลียนแบบพฤติกรรมของการสื่อสารของผึ้ง โดยผึ้งจะบินซิกแซกและจะให้ข้อมูลข่าวสารระหว่างผึ้งด้วยกันที่เกี่ยวกับ ตำแหน่ง ระยะทาง และทิศทางของอาหารที่พวกมันกำลังหาอยู่ Zigbee ถูกสร้างขึ้นในการทำระบบเครือข่ายไร้สายส่วนบุคคล (Wireless Personal Area Network) โดยเริ่มจากการกำหนดมาตรฐานการรับ-ส่งข้อมูลแบบ IEEE 802.15.4 ที่เน้นการสื่อสารแบบประหยัดพลังงานจึงทำให้แบตเตอรี่มีอายุการใช้งานที่ยาวนาน ความเร็วการรับ-ส่งข้อมูลต่ำ และมีราคาถูกซึ่งการสื่อสารไร้สายแต่ละแบบก็จะมีคุณลักษณะเฉพาะตัวในการนำไปใช้งาน เมื่อเราต้องการเลือกชนิดของการสื่อสารไร้สายมาใช้ เราต้องพิจารณาลักษณะการใช้งานและองค์ประกอบต่างๆ เพื่อการเลือกใช้งานได้อย่างเหมาะสม เทคโนโลยีการสื่อสารข้อมูลแบบไร้สายในปัจจุบันมีการใช้งานกันอย่างแพร่หลาย ที่เรารู้จักและคุ้นเคยในชีวิตประจำวันได้แก่ โทรศัพท์เคลื่อนที่ไร้สาย, GSM/CDMA, Wi-Fi, Bluetooth, Zigbee, ไมโครเวฟ(Microwave), อินฟราเรด(Infrared) เป็นต้น

ตารางที่ 2.1 การเปรียบเทียบคุณสมบัติของเทคโนโลยีไร้สาย

	Zigbee	Bluetooth	Wi-Fi
Standard	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.11b
Application Focus	Monitoring & Control	Cable Replacement	Web, Video, E-mail
Transmission Rang(meters)	1-100	1-10	1-100
Battery Life(Days)	100-1,000	1-7	0.5-5.0
Network Size(amount of nodes)	>64,000	7	32
Stack Size(kb)	4-32	250	1,000
Throughput(kb/s)	20-250	720	11,000

เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ปริมาณการใช้Throughputของมาตรฐานการสื่อสารไร้สายแบบต่างๆ

มาตรฐาน IEEE 802.15.4

มาตรฐาน IEEE 802.15.4 กำหนดขึ้นสำหรับการรับส่งข้อมูลเบื้องต้นในวงจรเครื่องรับส่งวิทยุ(Physical Layer) และการควบคุมการรับส่ง (Link Layer) ดังต่อไปนี้ การสื่อสารใช้คลื่นวิทยุ ความถี่ 2.4 กิกะเฮิรตซ์ (GHz) แบ่งเป็น 16 ช่องสัญญาณๆละ 5 เมกะเฮิรตซ์ (MHz) สำหรับความถี่ 900 เมกะเฮิรตซ์ (MHz) แบ่งเป็น 10 ช่องสัญญาณๆละ 2 เมกะเฮิรตซ์ (MHz) ใช้การผสมสัญญาณ (modulation) แบบ Offset Quadrature Phase Shift Keying (Offset-QPSK) และใช้การแก้ปัญหาสัญญาณรบกวนแบบ Direct Sequence Spread Spectrum(DSSS) ที่อัตราการสเปรดดิ้ง (spreading) 2 ล้าน chip/sec ควบคุมการรับส่งข้อมูลโดยใช้โปรโตคอลแบบ Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) และเพื่อให้การสื่อสารเครือข่ายเช่นเซอร์ไร้สายเป็นมาตรฐานเดียวกัน จึงกำหนดมาตรฐานเพิ่มสำหรับการเชื่อมต่อเป็นเครือข่าย (Network Layer) และนำไปใช้งาน (Application Layer) ร่วมกับมาตรฐาน IEEE 802.15.4 เป็นมาตรฐานใหม่ที่กำหนดโดยองค์กร Zigbee Alliance

คุณสมบัติทั่วไปของZigbee

Zigbee เป็นมาตรฐานของอุปกรณ์ไร้สายถูกกำหนดโดยกลุ่ม Zigbee Alliance ซึ่งเริ่มก่อตั้งขึ้นเมื่อปี 2002 เพื่อใช้ควบคุมอุปกรณ์ต่างๆรอบตัวที่ใช้ในชีวิตประจำวันในแบบไร้สายซึ่งไม่ต้องการความเร็วสูงมากนัก เช่น สวิตช์เปิดปิดแสงสว่าง ระบบควบคุมอุณหภูมิห้อง เป็นต้น รวมไปถึงใช้ในการรับค่าจาก Sensor ต่างๆ โดย Zigbee ถูกออกแบบโดยมุ่งให้มีคุณลักษณะดังนี้

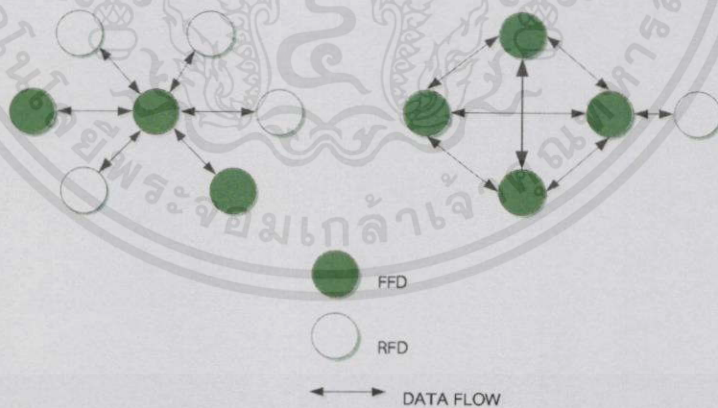
1. เป็นเครือข่ายไร้สายในระยะใกล้ มีการเชื่อมต่ออย่างซับซ้อนเพื่อรองรับการเชื่อมต่อสำหรับเครือข่ายขนาดใหญ่ ได้ด้วยรูปแบบมาตรฐาน 3 รูปแบบได้แก่ Multiple Topology คือ สตาร์ (star) เพียร์ทูเพียร์ (peer-to-peer) และ เมช (mesh)

2. ราคาไม่แพง
3. ติดตั้งง่าย สามารถประยุกต์ใช้งานได้หลากหลาย
4. สามารถรับส่งข้อมูลได้โดยเชื่อมมันในความถูกต้องได้
5. ใช้พลังงานในการทำงานต่ำ
6. มาตรฐาน Zigbee ตาม IEEE 802.15.4 กำหนดให้มีโหมดของการใช้พลังงาน 2 สถานะ คือ สถานะที่มีการรับและส่งของข้อมูลซึ่งเรียกว่า Active และอีกสถานะคือสถานะที่ไม่มีการทำงานชั่วขณะหนึ่งหรือเรียกว่า Sleep
7. สื่อสารใช้ความถี่วิทยุ ย่านความถี่ 2.4GHz มี 16 ช่องสัญญาณ อัตราการรับส่งข้อมูล 250Kbps

2.1.2 ชนิดของอุปกรณ์ Zigbee

แบบ Physical Device มี 2 ประเภท คือ

1. Full Function Device: FFD เป็นเราเตอร์ (Router) ที่เป็นสื่อกลางในการส่งข้อมูลจากอุปกรณ์อื่นๆ ใช้พลังงานจากสายส่งกำลังไฟฟ้า (Power line) ทำงานได้ในทุกๆ โครงสร้างเครือข่าย (Topology) และสามารถทำเป็นจุดเชื่อมต่อกันได้
2. Reduced Function Device: RFD เหมาะแก่การเชื่อมต่อภายในเครือข่ายใช้พลังงานจากแบตเตอรี่ ไม่สามารถถ่ายทอดข้อมูลจากอุปกรณ์อื่นๆ ได้ ทำได้ง่ายในเครือข่ายที่เป็นแบบดาว (star) รูปแบบเครือข่ายตามมาตรฐาน IEEE 802.15.4 แบ่งออกเป็น 2 แบบ ได้แก่ แบบดาวและแบบตาข่าย (Peer-Peer) ร่างแห ดังรูปที่ 2.2



รูปที่ 2.2 โมเดลการส่งถ่ายข้อมูลแบบ Star (ข่าย) และ Peer-to-Peer (ตาข่าย)

แบบ Logical Device มี 3 ประเภท คือ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 1. Zigbee Coordinators เป็นจุดที่ประสานเชื่อมต่อกัน ทำหน้าที่เริ่มจัดการเชื่อม
ไม่ว่ากรณีใดๆ วิศวกรจัดการโหนดในโครงข่ายและเก็บข่าวสารของโหนดในโครงข่ายเอกสารทุกครั้งที่มีการนำไปใช้

2. Zigbee Routers ทำหน้าที่จัดการเส้นทางของข้อความที่ส่งผ่านภายในโครงข่าย โหนดใดๆ
3. Zigbee End Devices เป็นโหนดที่อยู่ในส่วนของผู้ใช้งาน โดยสามารถเป็นได้ทั้ง RFD และ FFD

2.1.3 การประยุกต์ใช้งาน

แบบแยกตามประเภทของข้อมูลข่าวสารที่มีอยู่ 3 แบบคือ

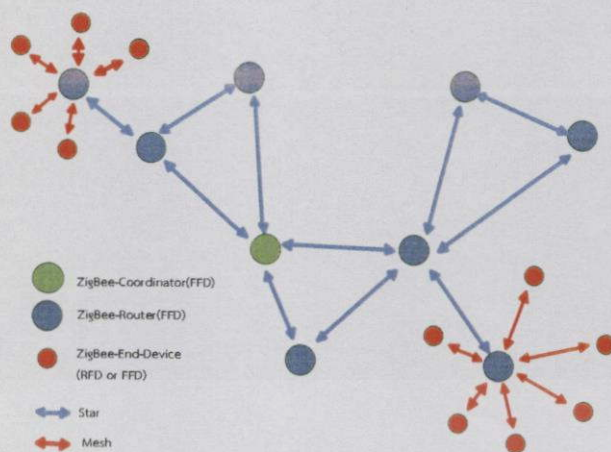
1. ข้อมูลแบบ Periodic ข้อมูลเป็นช่วงเวลา โปรแกรมสามารถควบคุมอัตราการส่ง และตัวตรวจจับสัญญาณกระตุ้น เซ็คข้อมูลและทำให้ข้อมูลไม่เคลื่อนไหว ใช้สำหรับเซนเซอร์ และ มิเตอร์
2. ข้อมูลแบบ Intermittent เป็นลักษณะที่มีการส่งผ่านข้อมูลเมื่อมีการใช้งานเช่น สวิตซ์ไฟ
3. ข้อมูลแบบ Repetitive low latency ใช้ในงานที่ต้องการ Latency น้อยๆ โดยการสื่อสารจะใช้วิธีจัดสรรช่วงเวลา และสามารถใช้กลไกแบบ GTS เพื่อรับประกันคุณภาพของการบริการ นำไปใช้ในงาน เช่น เมาส์ไร้สาย

2.1.4 โครงสร้างของโพรโทคอล Zigbee

โพรโทคอล Zigbee ถูกออกแบบมาในส่วนของ Network & Security Layer, Application Framework Layer และ Application & Profile Layer ส่วนของ PHY Layer และ MAC Layer ใช้ มาตรฐาน IEEE 802.15.4 ซึ่งเหมาะกับโพรโทคอล Zigbee เพราะมีอัตราส่งต่ำและ ใช้พลังงานน้อย โดยโพรโทคอล Zigbee ได้แบ่งประเภทของอุปกรณ์เป็น 3 ประเภทเป็น Logical Device คือ Zigbee Coordinator, Zigbee Router และ Zigbee End Device

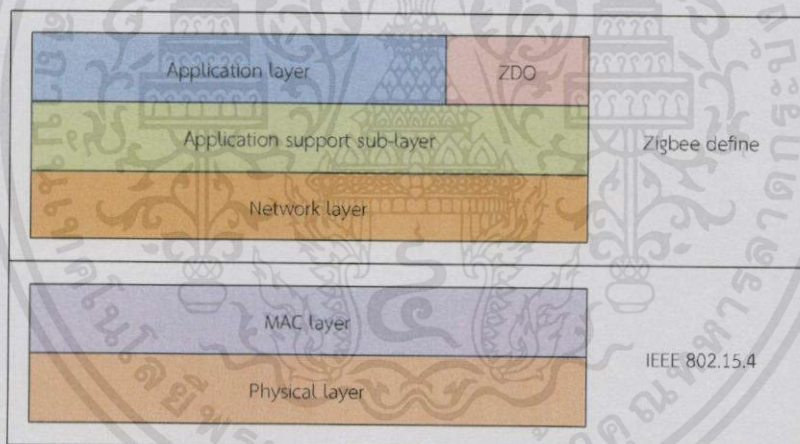
1. Zigbee Coordinator เป็นจุดเชื่อมต่อกัน ทำหน้าที่เก็บข้อมูลในเครือข่าย
2. Zigbee Router จัดการเส้นทางของข้อความที่ส่งผ่านภายในโครงข่ายของ โหนดใดๆ
3. Zigbee End Device เป็นโหนดที่อยู่ในส่วนของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ตัวอย่างการสร้างเครือข่ายของอุปกรณ์ Zigbee

ZigBee ถูกออกแบบมาเฉพาะในส่วนของ Application layer, Application support Layer และ Network layer เท่านั้น แต่ใช้ MAC layer และ Physical layer ตามมาตรฐาน IEEE 802.15.4



รูปที่ 2.4 โครงสร้าง Zigbee โดยแบ่งเป็น layer

โครงสร้างของ ZigBee จะแบ่งเป็น layer ต่างๆ ดังนี้

1. Application layer เป็นชั้นที่มีส่วนของ Endpoint อยู่ เรียกว่า Application framework โดยมี ZigBee Device Object (ZDO) ทำหน้าที่ในการจัดการในการเข้าถึงและใช้งาน Application layer

2. Application support sub-layer ทำหน้าที่ในการสร้างเฟรมของ Application layer และทำหน้าที่ในการรับ-ส่งข้อมูล รวมถึงการจัดการด้านต่างๆ ที่เกี่ยวข้องกับ Application layer

ไม่ว่ากรณีใดๆ ทางสำนักพิมพ์ฯ ขอสงวนสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Network layer ทำหน้าที่ในการ routing ข้อมูลต่างๆ จากต้นทางไปยังปลายทางที่อาจอยู่ภายในเครือข่ายเดียวกันหรือต่างเครือข่ายกัน

ZigBee Alliance ได้คิดหาวิธีแก้ปัญหาที่จะสามารถควบคุมสัญญาณให้ได้โดย

1. Physical Layer ถูกออกแบบมาให้มีความเหมาะสมกับราคาที่ย่อมเยา การเข้าถึงโดยตรงตามลำดับอนุญาติให้ใช้วงจรอะนาล็อกที่ไม่ซับซ้อนและมีความคงทนมากในการปรับปรุง

2. Media Access Control (MAC) layer ถูกออกแบบมาโดยยอมให้สามารถใช้งานได้กับโพรโทคอลหลายๆแบบ แต่ต้องไม่ซับซ้อน การจัดการพลังงานต้องไม่ใช่ขั้นตอนมาก MAC ต้องยอมให้ลดหน้าที่การทำงานของอุปกรณ์ (Reduced Functionality Device: RFD) บางตัวที่ไม่จำเป็นต้องใช้แสง (flash) หรือต้องใช้ RAM หรือ ROM ทำงานมากลง MAC ถูกออกแบบมาเพื่อให้รองรับได้กับจำนวนอุปกรณ์มากๆ โดยที่ ต้องไม่เกิดการหยุดชะงักด้วย (parked)

3. Network Layer จะได้รับการออกแบบเพื่อยอมให้ช่วงของเครือข่ายแผ่กว้าง โดยที่ ต้องไม่ใช้อุปกรณ์ส่งสัญญาณที่ใช้พลังงานสูง ชั้น Network Layer ต้องสามารถรองรับโหนดจำนวนมากได้โดยใช้เวลาแฝงต่ำด้วย

2.1.5 ชั้นตอนการทำงานของโพรโทคอล Zigbee

โมเดลนี้ได้ถูกแบ่งย่อยออกเป็น 7 ชั้นอันได้แก่ Application, Presentation, Session, Transportation, Network, Data Link และ Physical ตามลำดับจากบนลงล่าง อธิบายได้ดังนี้

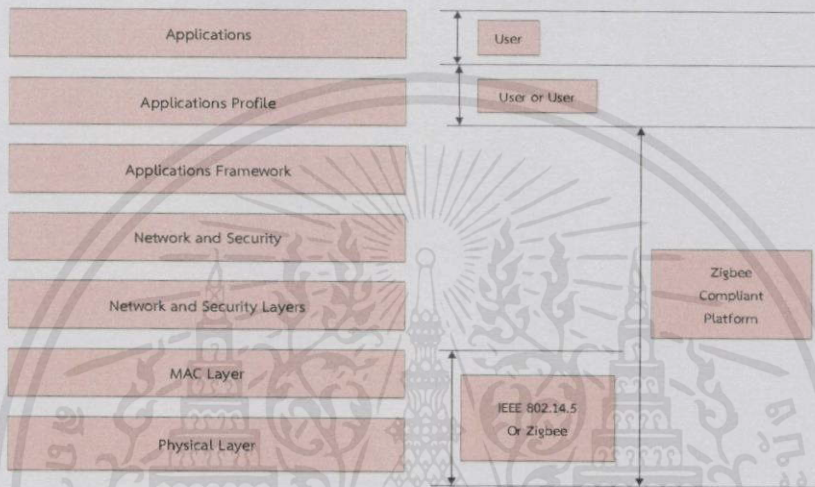
1. ชั้นประยุกต์ (Application Layer) ชั้นที่เจ็ดเป็นชั้นที่อยู่ใกล้ผู้ใช้มากที่สุดและเป็นชั้นที่ทำงานส่งและรับข้อมูลโดยตรงกับผู้ใช้ ตัวอย่างเช่น ซอร์ฟแวร์โปรแกรมต่างๆที่อาศัยอยู่บน layer นี้

2. ชั้นแสดงผล (Presentation Layer) ชั้นที่หกทำหน้าที่รับผิดชอบเรื่องรูปแบบของการแสดงผลเพื่อโปรแกรมต่างๆที่ใช้งานระบบเครือข่ายทำให้ทราบว่าข้อมูลที่ได้เป็นประเภทใด เช่น รูปภาพ เอกสารไฟล์ วิดีโอ

3. ชั้นจัดแบ่งข้อมูล (Session Layer) ชั้นที่ห้าทำหน้าที่ในการจัดการกับ session ของโปรแกรม ชั้นนี้เองที่ทำให้ในหนึ่งโปรแกรมยกตัวอย่างเช่น โปรแกรมค้นดูเว็บ (Web browser) สามารถทำงานติดต่ออินเทอร์เน็ตได้พร้อมๆกันหลายหน้าต่าง

4. ชั้นแลกเปลี่ยนข้อมูล (Transport Layer) ชั้นที่สี่ทำหน้าที่ดูแลจัดการเรื่องของความผิดพลาดที่เกิดขึ้นจากการสื่อสาร ซึ่งการตรวจสอบความผิดพลาดนั้นจะพิจารณาจากข้อมูลส่วนที่เรียกว่า checksum และอาจมีการแก้ไขข้อผิดพลาดนั้นๆ โดยพิจารณาจากฝั่งต้นทางกับฝั่งปลายทาง

5. ชั้นเครือข่าย (Network Layer) ชั้นที่สามจะจัดการการติดต่อสื่อสารข้ามเน็ตเวิร์ค ซึ่งเป็นการทำงานติดต่อข้ามเน็ตเวิร์คแทนชั้นอื่นๆที่อยู่ข้างบน
6. ชั้นเชื่อมต่อข้อมูล (Data Link Layer) ชั้นที่สองนี้ทำหน้าที่จัดเตรียมข้อมูลที่ส่งผ่านไปยังสื่อตัวกลาง
7. ชั้นกายภาพ (Physical Layer) เป็นชั้นแรกของสื่อที่ใช้ในการติดต่อสื่อสาร ซึ่งอาจจะเป็นทั้งแบบที่ใช้สายหรือไม่ใช้สาย



รูปที่ 2.5 ชั้นโพรโตคอลการรับส่งข้อมูล

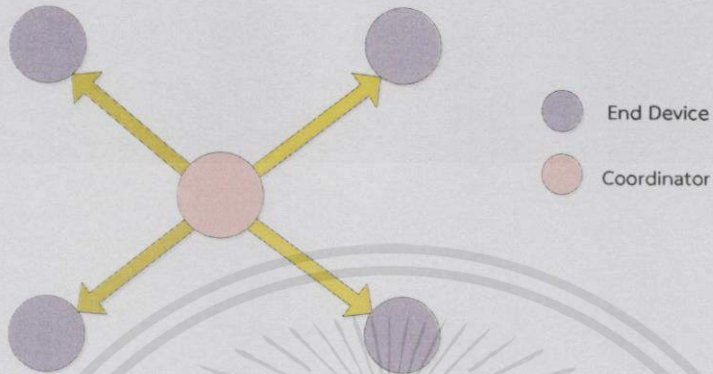
จากรูปที่ 2.5 ชั้นโพรโตคอลการรับส่งข้อมูลความถี่การรับ-ส่งข้อมูลตามมาตรฐานการเลือกใช้งานแต่ละย่านความถี่นั้นควรเป็นไปตามลักษณะและความเหมาะสมต่อการใช้งาน โดยงานวิจัยนี้ได้เลือกใช้มาตรฐาน IEEE 802.15.4 หรือเรียกว่า Zigbee ความถี่ 2.4 GHz มี 16 ช่องสัญญาณ อัตราการรับส่งข้อมูล 250 Kbps ซึ่งเป็นย่านการสื่อสารระยะใกล้ที่เน้นการใช้พลังงานต่ำและทนต่อสภาพสัญญาณการรบกวนสูง ซึ่งเป็นมาตรฐานที่ออกแบบมาสำหรับการติดต่อกับเซนเซอร์ชนิดต่างๆส่งข้อมูลไปยังเครื่องเซนเซอร์ที่ต้องการและสามารถเพิ่มจำนวนอุปกรณ์เซนเซอร์ให้มากขึ้นได้

2.1.6 Topologyในการใช้งาน Zigbee Network

ในการสร้างโครงข่ายไร้สายของ ZigBee นั้น จะต้องประกอบด้วย โหนดจำนวนอย่างน้อยที่สุด 2 ชนิด คือ Coordinator node และ node ลูกข่าย ชนิดใดชนิดหนึ่ง (Router/End device) จึงจะสามารถสื่อสารและทำงานในรูปแบบของ PAN (Personal area network) ได้ โดย ZigBee สามารถแบ่งรูปแบบ เครือข่ายได้เป็น 3 รูปแบบ ดังนี้

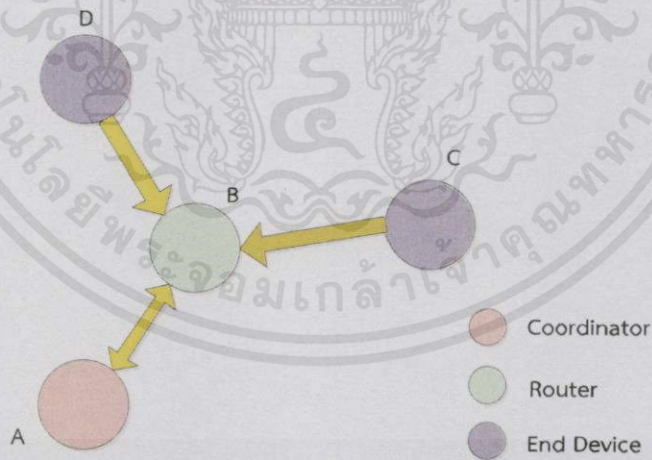
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ห้ามทำซ้ำโดยไม่ขออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Star (Broadcast) การเชื่อมต่อแบบ Star หรือแบบ Broadcast เป็นการรับส่งข้อมูลแบบ ไม่เฉพาะเจาะจง จุดหมายปลายทางหรือ XBee ทุกตัวที่อยู่ในระบบเครือข่ายเดียวกัน สามารถ รับข้อมูลทุกข้อมูลได้ทุกตัว



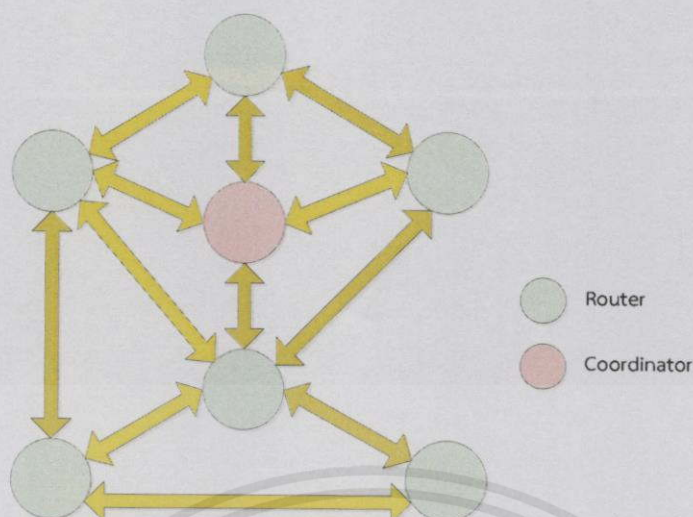
รูปที่ 2.6 Star (Broadcast) Network

2. Cluster Tree (Tree) เป็นการ รับส่งข้อมูล แบบ ส่งผ่าน เช่น A ต้องการติดต่อกับ C แต่ C อยู่ไกลจาก A จน A ไม่สามารถ ติดต่อกับ C ได้ แต่พอดีมี B ที่อยู่ระหว่าง A กับ C ดังนั้น Cluster Tree จะใช้ B เป็นเหมือน ตัวกลาง เชื่อมการติดต่อ (Repeater) ระหว่าง A กับ C



รูปที่ 2.7 Cluster Tree (Tree) Network

3. Mesh การเชื่อมต่อเครือข่ายแบบ Mesh เป็นโครงข่ายที่มีประสิทธิภาพสูง เนื่องจาก ข้อมูลสามารถส่งไปถึงเป้าหมายได้หลายทาง ทำให้ ระบบนี้สามารถรับส่งข้อมูลไปยังจุดหมายปลายทางได้ แม้จะเกิดความเสียหายของระบบในบางส่วนก็ตาม (ขึ้นอยู่กับ การออกแบบระบบของผู้ใช้ด้วย) ระบบนี้จึงเป็นระบบที่ได้รับความนิยมเป็นอย่างมาก



รูปที่ 2.8 Mesh Network

2.1.7 การส่งข้อมูลของ Zigbee

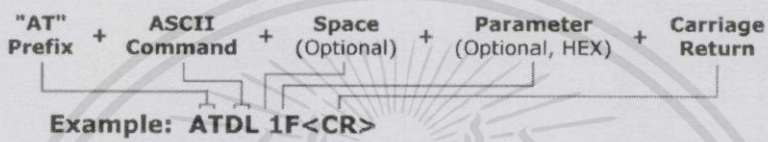
การส่งข้อมูลแบบ RF ของแต่ละแพ็คเกจในส่วนของ header คือ Zigbee จะสามารถกำหนดค่าแอดเดรสอ้างอิงได้ 2 แบบจะประกอบไปด้วย Source Address และ Destination Address โดยที่ IEEE 802.15.4 จะมีโครงสร้าง 2 แบบ นั่นคือแอดเดรสอ้างอิงแบบ short 16-bit address และแอดเดรสอ้างอิงแบบ long 64-bit address ปกติแล้วจะถูกกำหนดค่ามาจากโรงงานเป็นแอดเดรสอ้างอิง 64 บิต ซึ่งจะสามารถอ่านค่าได้จากพารามิเตอร์ SH (Serial Number High) และ SL (Serial Number Low) การใช้งานแอดเดรสอ้างอิง 64 บิต สามารถทำได้โดยกำหนดพารามิเตอร์ MY ให้มีค่าน้อยกว่า 0xFFFF หรือ 0xFFFE ส่วนการกำหนดแอดเดรสอ้างอิง 16 บิตนั้นทำได้โดย กำหนดพารามิเตอร์ MY ให้มีค่าน้อยกว่า 0xFFFE โดยจะเรียกเป็นโหมดการส่งข้อมูล 2 ประเภทคือ

1. โหมดยูนิแคสต์ (Unicast Mode) คือ การรับส่งข้อมูลโดยอาศัยหลักการตอบรับ (Acknowledgement) คือ หากทางด้านส่งข้อมูลไปแต่ไม่ได้รับการตอบรับจากตัวรับก็จะทำการส่งข้อมูลใหม่
2. โหมดบรอดแคสต์ (Broadcast Mode) คือ การส่งข้อมูลไปยังปลายทางให้ได้รับข้อมูลทุกตัว Xbee ยังมีการตั้งค่าการทำงานแบ่งเป็น 4 โหมดหลักๆ ดังนี้
 - 1) โหมดไอดีล (Idle Mode) เป็นโหมดที่ไม่ได้รับส่งข้อมูลตัว Xbee เตรียมที่จะทำงานในโหมดอื่นๆ ต่อไปทันทีหากมีเงื่อนไขบางอย่าง
 - 2) โหมดรับ-ส่งข้อมูล (Transmit/Receive Mode) คือ ช่วงที่ Xbee มีการรับหรือส่งข้อมูล โดยแบ่งลักษณะการทำงานย่อยออกเป็นแบบโดยตรง (Direct) กับโดยอ้อม (Indirect), การกำหนดแอดเดรสต้นทางและปลายทาง, Clear Channel Assessment และการตอบรับ

3) โหมดสลีป (Sleep Mode) คือช่วงที่Xbee อยู่ในสถานะที่พลังงานต่ำที่สุดเมื่อไม่มีการใช้งาน

4) โหมดคำสั่ง คือ ส่วนการปรับพารามิเตอร์ของXbee ซึ่งจะมีการกำหนด 2 แบบ คือ คำสั่งแบบAT กับคำสั่งแบบAPI

ซึ่งสองโหมดนั้นจะมีรูปแบบการส่งข้อมูลและคำสั่งที่ใช้ติดต่อกับZigbee Module ต่างกันออกไปใน AT Command นั้น Zigbee Module จะทำหน้าที่เป็น Serial Line Replacement นั่นคือมีรูปแบบการใช้งานเหมือนกับการเชื่อมต่อด้วยสาย Serial ธรรมดา นั่นเองซึ่งคำสั่งแบบ AT มีรูปแบบรูปที่ 2.9



รูปที่ 2.9 รูปแบบของคำสั่ง AT (AT Command)

จากรูปที่ 2.9 เป็นตัวอย่างการใช้งานAT Command ซึ่งเป็นการเซตค่าของ Address ของ Zigbee Module ปลายทางที่ต้องการติดต่อกับ ในส่วน Low Address เป็น 1F นั่นเอง

ตัวอย่างของAT Command ที่สามารถเลือกใช้ได้ เช่น

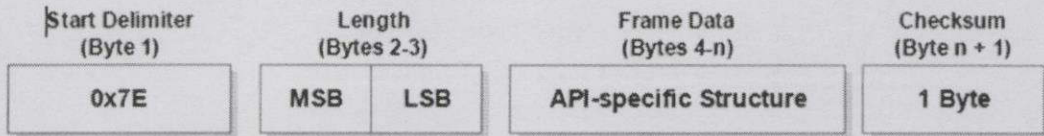
-ATRR X โดย X มีค่าได้ตั้งแต่ 0 ถึง 6 ซึ่งคำสั่งนี้จะเป็นการกำหนดจำนวนครั้งของ Xbee Retry ที่ต้องการให้ทำการลองส่งใหม่เมื่อเกิดการส่งผิดพลาดขึ้นที่ชั้น MAC Layer โดย Xbee Retry 1 ครั้ง นั้นจะสั่งให้ทำ มีการส่งข้อมูลใหม่ในชั้น MAC Layer 3 ครั้ง

-ATND เป็นการสั่งให้ทำการค้นหา Zigbee Module ซึ่งอยู่ภายในระยะที่สามารถรับส่งข้อมูลกันได้ โดยจะแสดงรายละเอียดเช่น Address แบบ 16Bits เลข Serial Number ของ Zigbee Module นั้น ระดับของสัญญาณที่ติดต่อกับ Node นั้นได้ เป็นต้น

-ATAP X โดย X มีค่าได้ตั้งแต่ 0 ถึง 2 คำสั่งนี้จะเป็นการสั่งให้ Zigbee ทำการเปิด/ปิดการใช้ API Mode

สำหรับ API Mode (Application Programming Interface Mode) ทั้งข้อมูลและคำสั่งจะอยู่ในรูปแบบ Frame-Based ทั้งหมด โดย API Frame จะมีรูปแบบรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MSB = Most Significant Byte, LSB = Least Significant Byte

รูปที่ 2.10 รูปแบบของคำสั่ง API (API Command)

Byte 1 เป็น Start Delimiter บอกการเริ่มต้น Frame มีค่าเป็น 0x7E เสมอ

Byte 2 ถึง 3 เป็นจำนวน Byte ของส่วน Frame Data

Byte 4 ถึง n เป็นส่วนคำสั่งและพารามิเตอร์หรือเป็นข้อมูล

Byte n+1 ใช้เป็น checksum

ตัวอย่างของ API Frame

คำสั่ง ATND สำหรับค้นหาโทนซึ่งอยู่ภายในรัศมี เมื่ออยู่ในรูปแบบ API Command จะมีรูปแบบ

7E 00 04 08 01 4E 44 64

โดยมีความหมายคือ

-7E เป็น Start Delimiter

-00 04 เป็น Length

-08 เป็นการระบุว่า เป็น AT Command

-01 เป็นการระบุว่า เป็นคำสั่งชนิดที่จะมีผลการทำงานตอบกลับมา

-4E 44 เป็นคำสั่ง ND ในรูปแบบ Hexadecimal

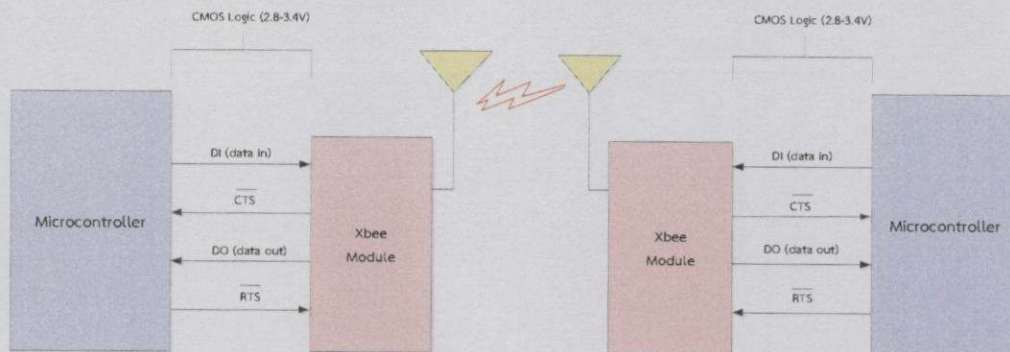
-64 เป็น Checksum

2.1.8 การติดต่อสื่อสารของ Zigbee ผ่านทางพอร์ตอนุกรม

Zigbee อินเทอร์เน็ตเฟสกับอุปกรณ์ผู้ใช้ (Host) ผ่านทางพอร์ตอนุกรมอะซิงโครนัส ระดับตรรกะ ส่งผ่านโมดูลทางพอร์ตอนุกรม โมดูลสามารถติดต่อสื่อสารกับตรรกะต่างๆเข้ากับ UART การไหลของข้อมูล อุปกรณ์มีอินเทอร์เน็ตเฟส UART สามารถติดต่อโดยตรงกับพินของ RF โมดูล ดังรูปที่ 2.11

ข้อมูลส่งเข้า UART โมดูลผ่านขา Data In (ขา 3) ด้วยสัญญาณอนุกรมอะซิงโครนัส สัญญาณจะว่างเมื่อไม่มีข้อมูลถูกส่ง แต่ละข้อมูลไบต์ประกอบด้วยบิตเริ่มต้น(บิตต่ำ) 8 ข้อมูลบิต และบิตหยุด(บิตสูง)ตามรูปแบบงานโมดูล UART ดังนั้นเวลาและพาริตีที่ตรวจสอบต้องการข้อมูลติดต่อสื่อสาร การสื่อสารอนุกรมขึ้นอยู่กับ 2 UARTS ถูกจัดโครงสร้างกับการจัดตั้งอัตราบอर्ड(baud rate), พาริตี(parity), บิตเริ่มต้น(start bit), บิตหยุด(stop bit) และบิตข้อมูล(data bit)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และสงวนสิทธิ์ในเนื้อหา มีอยู่ภายใต้เงื่อนไขการใช้งานที่ปรากฏในเอกสารฉบับนี้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 การต่อไมโครคอนโทรลเลอร์กับZigbee

2.2 ลักษณะทั่วไปของโมดูลXbee

Xbee เป็นอุปกรณ์ที่มีไมโครคอนโทรลเลอร์ และ RF IC อยู่ภายใน ทำหน้าที่เป็นอุปกรณ์รับส่งสัญญาณ (Transceiver) แบบฮาร์ฟดูเพล็กซ์ ย่านความถี่ 2.4 GHz มีการจัดการโดยใช้พลังงานต่ำ ใช้งานง่าย มีส่วนติดต่อที่ใช้รับส่งข้อมูลกับxbee เป็นการสื่อสารแบบอนุกรม ซึ่งไมโครคอนโทรลเลอร์สามารถนำขาที่ใช้ติดต่อสื่อสารแบบอนุกรมของ xbee ต่อเข้ากับพอร์ตสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ได้

Firmware ที่ใช้กับ xbee จะใช้โหลดผ่านโปรแกรมทั้งนี้ xbee แต่ละรุ่นจะสามารถตั้งค่าฟังก์ชัน (Setting Function) การใช้งานได้มากมาย ทำให้Firmware ที่จะต้องโหลดเข้าไปนั้น มีมากมายหลายแบบ ต้องเลือกให้เหมาะสมกับการใช้งาน

Xbeeสามารถใช้งานตามมาตรฐานZigbeeได้โดยไม่ต้องเขียนโปรแกรมสร้างเครือข่ายZigbee เพราะทางผู้ผลิตได้จัดทำเฟิร์มแวร์ที่จะโหลดเข้าไปในตัว xbee ให้สามารถตั้งค่าพารามิเตอร์ ผ่านซอฟต์แวร์โปรแกรมX-CTU หรือโปรแกรมที่เขียนขึ้นเอง โดยเมื่อตั้งค่าxbeeให้ทำงานเป็นอุปกรณ์ในเครือข่ายซิกบีแล้วเรียก xbeeแต่ละตัวว่าโหนด

Xbee มี 2 รุ่น คือรุ่น Series1 และรุ่น Series 2 (ZB) และมีขนาดกำลังให้เลือกอีก 2 แบบคือ แบบธรรมดา (1mW – 2mW) และแบบ PRO (50mW- 60mW) ซึ่งจะมีผลเรื่องระยะทางการรับส่งข้อมูล โดยแต่ละรุ่นนั้นสามารถสร้างเครือข่ายได้ 2 แบบคือ Star และ Cluster Tree แต่จะมีเพียงรุ่น Series2เท่านั้นที่จะทำเครือข่ายแบบMesh ได้

2.3 การสื่อสารข้อมูล

การสื่อสารข้อมูล (Data Communication) เป็นจุดเริ่มต้นของการดำเนินกิจกรรมต่างๆ เช่น กระบวนการตัดสินใจอุตสาหกรรม การประกอบธุรกิจ หรือแม้กระทั่ง การพัฒนาซอฟต์แวร์คอมพิวเตอร์ (Computer Software) ซึ่งจำเป็นจะต้องมีการเข้าถึงข้อมูลอยู่ในลักษณะของเครือข่ายคอมพิวเตอร์ (Computer Network) เทคโนโลยีที่พัฒนาก้าวหน้าขึ้นอย่างรวดเร็ว ได้มีส่วนในการปรับปรุงการทำงานของเครือข่ายคอมพิวเตอร์ให้มีการเชื่อมต่อสื่อสาร (Communication

Link) ความเร็วสูงที่สามารถรองรับสัญญาณปริมาณมากได้ซึ่งนำไปสู่การขยายขอบเขตความสามารถในการสื่อสารอย่างเช่น การบริการสื่อผสม (Multimedia) ของระบบโทรศัพท์ไร้สาย (Mobile Phone) การประชุมร่วมทางไกล (Teleconference) การถ่ายทอดสดภาพและเสียง ผ่านทางเครือข่ายอินเทอร์เน็ต (Internet) หรือแม้กระทั่งการควบคุมการทำงานของเครื่องจักรระยะไกล เป็นต้น จุดประสงค์หลักของการสื่อสารข้อมูลคือการแลกเปลี่ยนข่าวสารระหว่างกันได้อย่างถูกต้องแม่นยำในเวลาที่ต้องการ

2.3.1 ลักษณะพื้นฐานของการสื่อสารข้อมูล

เมื่อกล่าวถึงการสื่อสาร (Communication) มักจะหมายถึงการแลกเปลี่ยนข่าวสาร ซึ่งอาจจะเกิดขึ้นภายในระยะทางใกล้ๆหรือไกลออกไปก็ได้ (ในกรณีหลังมักจะใช้คำอุปสรรค Tele- นำหน้าซึ่งมีความหมายว่า “ไกล”) เมื่อนำคำว่าข้อมูล (Data) ซึ่งหมายถึงการนำเสนอข่าวสาร ในรูปแบบที่มีการตกลงกันระหว่างแต่ละฝ่ายมาผนวกเป็นการสื่อสารข้อมูล (Data Communication) จึงหมายถึงการแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ (Device) 2 ตัว ผ่านทางตัวกลางส่งผ่าน (Transmission Media) การบ่งชี้ประสิทธิภาพในการสื่อสารข้อมูล ซึ่งประกอบขึ้นจากระบบสื่อสารดังกล่าว สามารถทำได้โดยพิจารณาถึงลักษณะพื้นฐานของการสื่อสาร ดังต่อไปนี้

1. การนำส่ง (Delivery) ระบบต้องจัดส่งข้อมูลไปยังปลายทางได้อย่างถูกต้อง นั่นคือผู้รับ (Users) หรืออุปกรณ์-ตัวรับ (Receiving Devices) ที่ระบุเท่านั้น จึงจะสามารถรับข้อมูลได้
2. ความเที่ยงตรง (Accuracy) ข้อมูลที่ส่งไปต้องเที่ยงตรง การเปลี่ยนแปลงใดก็ตามที่เกิดขึ้นในขั้นตอนการส่งจะต้องได้รับการแก้ไขปรับเปลี่ยนให้ถูกต้องเมื่อถึงปลายทาง
3. ความตรงต่อเวลา (Timeliness) การจัดส่งข้อมูลจะต้องเสร็จสมบูรณ์ภายในระยะเวลาที่กำหนดโดยเฉพาะการสื่อสารแบบเวลาจริง (Real Time) ระบบต้องส่งข้อมูลทันทีที่ข้อมูลได้ถูกสร้างขึ้นในลำดับที่ถูกต้องโดยไม่เกิดการหน่วงล่าช้า (Delay) น้อยที่สุด

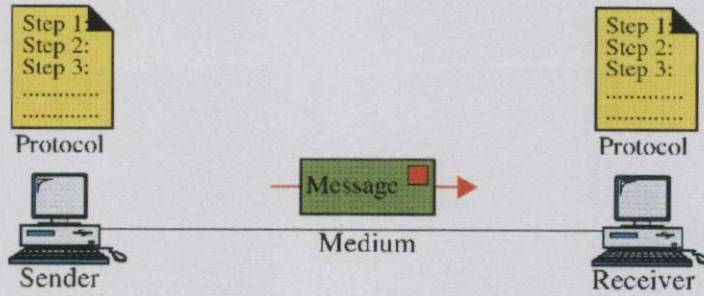
องค์ประกอบในการสื่อสาร

จากรูปที่ 2.12 ระบบการสื่อสารข้อมูลมีองค์ประกอบ 5 ประการ กล่าวคือ

1. Message คือ ชุดของข้อมูลข่าวสาร (Information) ที่ต้องการสื่อสาร
2. Sender ได้แก่ อุปกรณ์ซึ่งทำหน้าที่ส่งข่าวสาร เช่น คอมพิวเตอร์ กล้องวิดีโอ ฯลฯ
3. Receiver ได้แก่ อุปกรณ์ซึ่งทำหน้าที่รับข่าวสาร เช่น เครื่องรับโทรทัศน์ ฯลฯ
4. Medium คือตัวกลางที่ข่าวสารใช้ในการเดินทางระหว่างอุปกรณ์รับ-ส่ง เช่น

คลื่นวิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

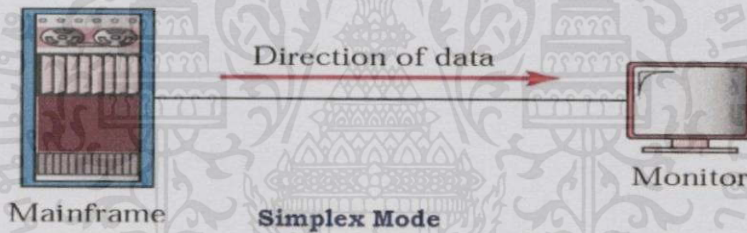


รูปที่ 2.12 องค์ประกอบ 5 ประการของระบบสื่อสารข้อมูล

ทิศทางการไหลของข้อมูล

การสื่อสารข้อมูลระหว่างอุปกรณ์ รับ-ส่ง 2 ชุดสามารถทำได้ 3 รูปแบบ (Mode) กล่าวคือ

1. Simplex คือ การสื่อสารที่เกิดขึ้นได้เพียงทิศทางเดียวตลอดเวลาการเชื่อมต่อจะมีอุปกรณ์ชุดเดียวเท่านั้นทำหน้าที่ส่งข้อมูล (Transmitter) และอุปกรณ์อีกชุดหนึ่งทำหน้าที่รับข้อมูล (Receiver) เช่น การสื่อสาร Mainframe กับ Monitor ดังรูปที่ 2.13

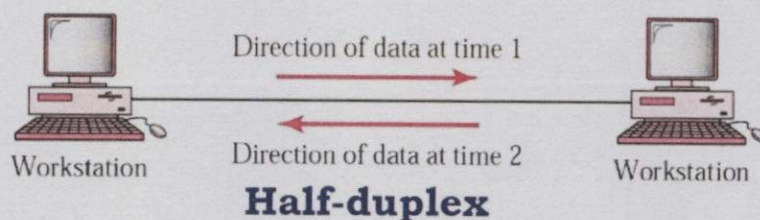


รูปที่ 2.13 แผนผังแสดงทิศทางการไหลของข้อมูลแบบ Simplex

2. Half-Duplex คือ การสื่อสารที่อุปกรณ์ทั้งสองชุด สามารถส่งและรับข้อมูลได้ แต่อุปกรณ์แต่ละชุดไม่สามารถทำหน้าที่สองอย่างพร้อมๆกันได้ กล่าวคือ ณ เวลาใดๆจะมีเพียงอุปกรณ์ชุดเดียวเท่านั้นทำหน้าที่ส่งข้อมูล ในขณะที่อุปกรณ์อีกชุดหนึ่งรับข้อมูล ซึ่งอุปกรณ์คู่นี้สามารถสลับบทบาทหน้าที่ได้ตลอดการเชื่อมต่อ เช่น การสื่อสารวิทยุรับ-ส่ง (CB: Citizen Band) การสื่อสารบนพอร์ต (Port) อนุกรมของระบบอุตสาหกรรม เป็นต้น

ในกรณีทั่วไประหว่างการสื่อสารแบบ Half Duplex ฝ่ายที่ส่งข้อมูลในขณะนั้นต้องระบุข่าวสาร 3 ส่วน ได้แก่ ชื่อหรือหมายเลขผู้ส่ง ข้อความที่ต้องการส่งและรหัสที่บ่งชี้ว่าการส่งข้อมูลได้เสร็จสิ้นลงพร้อมจะรับข้อมูล (เช่น over ทราบแล้วเปลี่ยน เป็นต้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

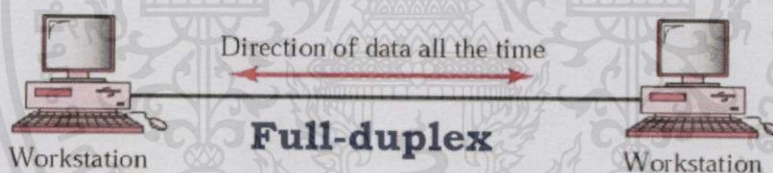


รูปที่ 2.14 แผนผังแสดงทิศทางการไหลของข้อมูลแบบ Half-Duplex ซึ่งข้อมูลสามารถไหลได้ทั้งทิศทางจากซ้ายไปขวาและขวาไปซ้าย ไม่พร้อมกันในแต่ละช่วงเวลา

3. Full-Duplex คือการสื่อสารที่อุปกรณ์ทั้งสองชุดสามารถส่งและรับข้อมูลในขณะเดียวกันได้ โดยมีข้อกำหนดว่า สัญญาณการสื่อสารต้องสามารถรองรับทิศทางของข้อมูลได้ทั้งสองทาง ซึ่งอาจทำได้ 2 รูปแบบได้แก่

3.1 การเชื่อมต่อ ประกอบด้วย ช่องสัญญาณ 2 ช่อง แยกออกจากกันเป็นอิสระ โดยที่แต่ละช่องทำหน้าที่ถ่ายโอนข้อมูลในทิศทางตรงข้ามกัน

3.2 การเชื่อมต่อมีเพียงช่องสัญญาณเดียวแต่สามารถรองรับการถ่ายโอนข้อมูลได้ทั้งสองทิศทาง ตัวอย่างการสื่อสารด้วยระบบ Full-Duplex ได้แก่ โทรศัพท์ทั้งแบบเคลื่อนที่และธรรมดาซึ่งคู่สายที่เกี่ยวข้องสามารถพูด-คุยพร้อมๆกันได้ และMSN Messenger เป็นต้น



รูปที่ 2.15 แผนผังแสดงทิศทางการไหลของข้อมูล Full-Duplex ซึ่งข้อมูลสามารถไหลได้ทั้งทิศทางจากซ้ายไปขวาและขวาไปซ้าย ได้พร้อมกันในแต่ละช่วงเวลา

2.3.2 ข้อกำหนด หรือโปรโตคอล

ข้อกำหนด (Protocols) คือกลุ่มของกฎหรือข้อตกลงในการสื่อสารข้อมูล ซึ่งได้นิยามไว้ เป็นที่เข้าใจตรงกันระหว่างหน่วย (Entity) ที่เกี่ยวข้องในการรับและส่งข่าวสาร การนิยาม Protocol ในการสื่อสารประกอบด้วยองค์ประกอบ 3 ประการได้แก่

1. Syntax ได้แก่ โครงสร้างหรือรูปแบบของข้อมูลโดยมากมักหมายถึง ลำดับ (Order) ของข้อมูล เช่น Protocol อย่างง่าย อาจนิยามให้ 8 บิตแรกหมายถึงตำแหน่งผู้ส่ง 8 บิตถัดมาเป็นตำแหน่งของผู้รับ และข้อมูลที่ตามมาก็คือเนื้อหาของข่าวสาร เป็นต้น

เอกสารนี้เป็นเอกสารที่ 2. Semantics ได้แก่ การตีความหมายของแต่ละส่วนของบิตข้อมูล และประโยชน์ด้านการคำนวณการที่จะต้องปฏิบัติอันเนื่องมาจากการตีความหมายนั้น เช่น หมายเลขตำแหน่งอาจการนำไปใช้หมายถึง ตำแหน่งของ Node ถัดไปในเส้นทางหรืออาจหมายถึงตำแหน่งปลายทางก็ได้

3. Timing ได้แก่ ความเร็วของการรับและส่งที่ทำการตกลงซึ่งจำเป็นต้องสอดคล้องกันทั้งสองฝั่งของอุปกรณ์ เพื่อป้องกันการสูญหายของข้อมูล

2.3.3 มาตรฐาน

มาตรฐาน (Standard) คือ Protocol ที่เป็นที่ยอมรับสำหรับการสื่อสารรูปหนึ่งๆ ตัวอย่างเช่น การสื่อสารในรูปแบบ Internet ยอมรับ Protocol 2 ลำดับชั้น ได้แก่

1. Transmission Control Protocol (TCP) สำหรับการทำงานลำดับสูง เช่น การจัดกลุ่มข้อมูลและการตรวจจับข้อผิดพลาด

2. Internetworking Protocol (IP) สำหรับการบริหารจัดการส่ง/ลำเลียงข้อมูล Standard สามารถแบ่งออกได้เป็น 2 ชนิด ได้แก่

2.1) De facto (by fact) ได้แก่ มาตรฐานที่มีการบังคับอย่างเป็นทางการและมีการควบคุมโดยองค์กรที่รับผิดชอบเช่น CCITT เป็นต้น

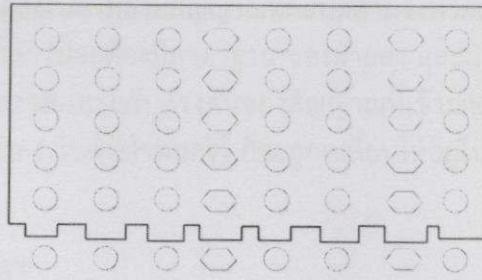
2.2) De jure (by regulation or by law) ได้แก่ มาตรฐานที่มีการบังคับอย่างเป็นทางการและมีการควบคุมโดยองค์กรที่รับผิดชอบเช่น CCITT เป็นต้น

2.4 การสื่อสารข้อมูลแบบอนุกรม

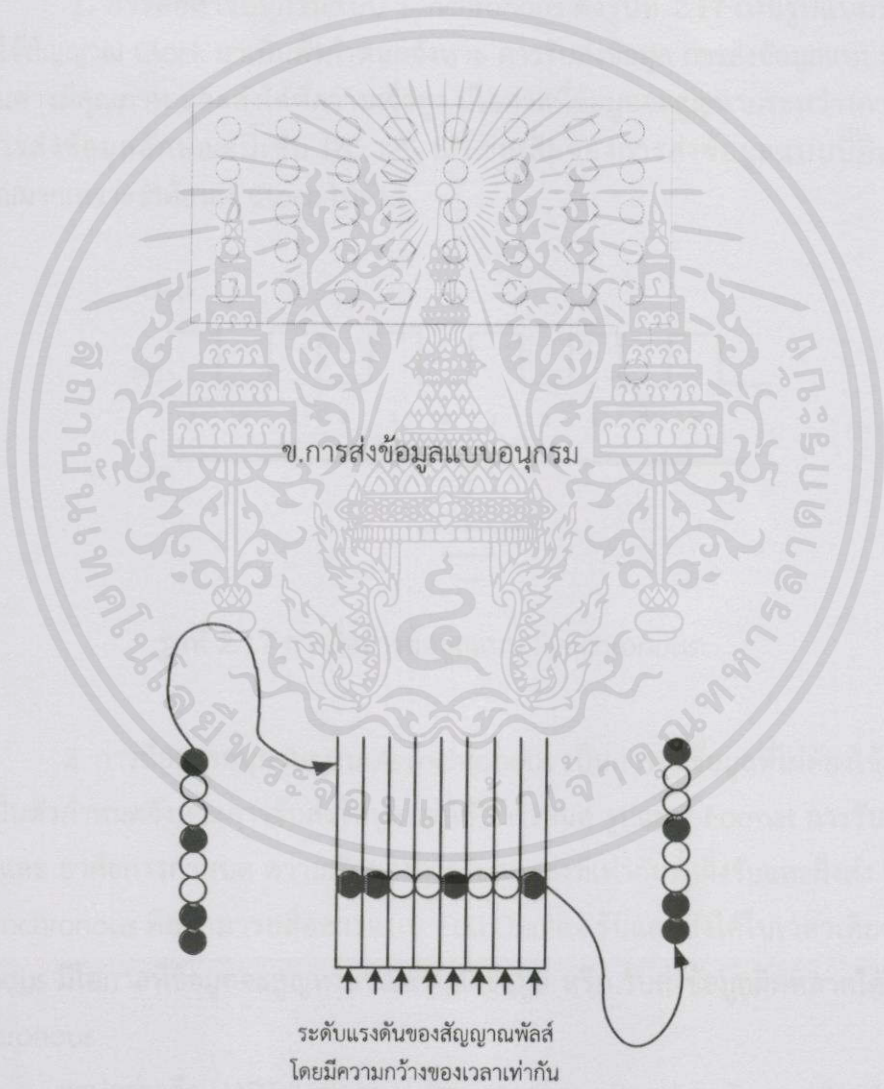
สำหรับไมโครคอนโทรลเลอร์โดยทั่วไปแล้ว เป็นข้อมูลที่มีความยาวขนาด 1 ไบต์หรือ 8 บิตซึ่งโดยปกติถ้าเราจะให้ส่งข้อมูลพร้อมๆกันไป 8 บิตจะเป็นวิธีการส่งข้อมูลแบบขนาน แสดงได้ดังรูป 2.16ก. จะเป็นการส่งข้อมูลขนาด 8 บิตพร้อมกันไปยังอุปกรณ์ภายนอก และจะต้องมีจำนวนของสายสัญญาณจำนวน 8 เส้น เพื่อให้พอดีกับจำนวนของบิตที่ต้องการจะส่ง การส่งข้อมูลแบบขนานจึงทำให้มีการส่งข้อมูลที่มีความรวดเร็ว แต่ถ้าหากมีการสื่อสารข้อมูลในระยะไกล ก็จะต้องใช้จำนวนของสาย และระยะทางของสายมากขึ้นจึงทำให้มีการสิ้นเปลืองค่าใช้จ่ายสูง

ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงถูกนำมาใช้ ในการสื่อสาร โดยจะใช้สายเพียงเส้นเดียวในการส่งข้อมูล หรือรับข้อมูล (คำว่าเส้นเดียวหมายความว่าสายส่ง (Tx) 1 เส้น สายรับ (Rx) 1 เส้น และสายกราวด์ร่วม (Ground) 1 เส้น) นำมาใช้สื่อสารข้อมูลกับอุปกรณ์ภายนอกในระยะทางที่ไกล ดังในรูป 2.16ข. ถ้าหากต้องการส่งข้อมูลขนาด 8 บิต ก็จะทำให้การส่งข้อมูลออกไปทีละบิตเป็นลำดับไป จนกว่าจะครบจำนวนทั้ง 8 บิต ดังในรูป 2.16ค. จะแสดงการเปลี่ยนข้อมูลแบบขนานให้เป็นแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก. การส่งข้อมูลแบบขนาน



ข. การส่งข้อมูลแบบอนุกรม

ระดับแรงดันของสัญญาณพัลส์
โดยมีความกว้างของเวลาเท่ากัน

ค. การเปลี่ยนแปลงสัญญาณจากขนานเป็นอนุกรมและจากอนุกรมเป็นขนาน

รูปที่ 2.16 การสื่อสารข้อมูลแบบขนานและอนุกรม

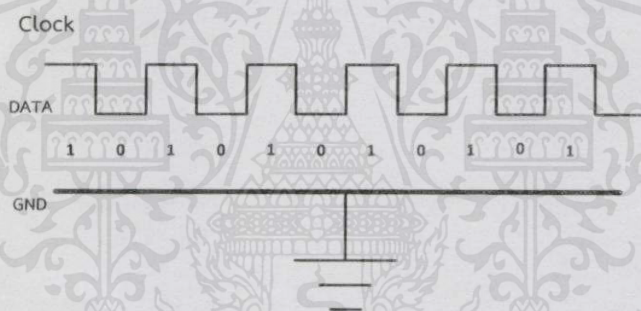
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลจะถูกส่งไปตามสายสัญญาณที่ละบิตตามจังหวะเวลาที่กำหนด เป็นความกว้างของพัลส์ โดยจังหวะเวลาที่กล่าวนี้จะต้องมีมาตรฐาน ของฝ่ายส่ง และฝ่ายรับด้วย ในการรับสัญญาณที่ส่งมาที่ละบิต จะทำการตรวจสอบระดับแรงดันของสัญญาณที่เข้ามาเพื่อแปลงเป็นลอจิก "1" หรือ "0" เมื่อรับข้อมูลเข้ามาครบใน 1 ไบต์ที่กำหนดไว้ ก็จะถูกเปลี่ยนให้อยู่ในรูปแบบของข้อมูลแบบขนานเหมือนเดิม

2.4.1 UART

ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter หมายถึง อุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัส (Asynchronous) ซึ่งเป็นส่วนหนึ่งในการสื่อสารอนุกรม แบบ Asynchronous แท้จริงแล้วการสื่อสารแบบอนุกรมจะแบ่งเป็น 2 แบบ คือ

1. การสื่อสารอนุกรมแบบ Synchronous ดังรูปที่ 2.17 เป็นรูปแบบที่ใช้วิธีส่งข้อมูล โดยใช้สัญญาณ Clock มาเป็นตัวกำหนดจังหวะ การรับส่งข้อมูล การส่งข้อมูลแบบนี้เป็นการรับส่งที่ค่อนข้างมีคุณภาพ และส่งได้ด้วยความเร็วสูง มีโอกาสที่ข้อมูลจะสูญหายระหว่างการส่งน้อย ตัวอย่างการส่งข้อมูลลักษณะนี้เช่น I2C, I2S, SPI ข้อเสียของการส่งข้อมูลแบบนี้คือ ต้องใช้สายสัญญาณมากเพราะว่าต้องส่ง Clock ไปด้วย



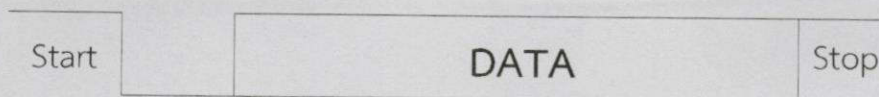
รูปที่ 2.17 การสื่อสารข้อมูลแบบ Synchronous

2. การสื่อสารอนุกรมแบบ Asynchronous เป็นการส่งข้อมูลที่ไม่ต้องใช้สัญญาณ Clock มาเป็นตัวกำหนดจังหวะการรับส่งข้อมูลแต่ ใช้วิธีกำหนด รูปแบบ Format การรับส่งข้อมูลขึ้นมาแทน และ อาศัยการกำหนด ความเร็วของการรับและส่งที่เท่ากันทั้งฝั่งรับและฝั่งส่ง ข้อดีของการใช้ Asynchronous คือสามารถสื่อสารแบบ Full Duplex รับและส่งได้ในเวลาเดียวกัน แต่ Asynchronous มีโอกาสที่ข้อมูลจะสูญหายขณะรับส่งข้อมูล หรือ รับส่งข้อมูลผิดพลาดได้มากกว่าแบบ Synchronous

สรุปกล่าวคือ UART (Universal Asynchronous Receiver Transmitter) หมายถึง

รูปแบบการส่งข้อมูล ที่ถูกกำหนดขึ้นมาเพื่อใช้รับส่งข้อมูลแบบ Asynchronous โดยมีรูปแบบดังรูปที่ 2.18

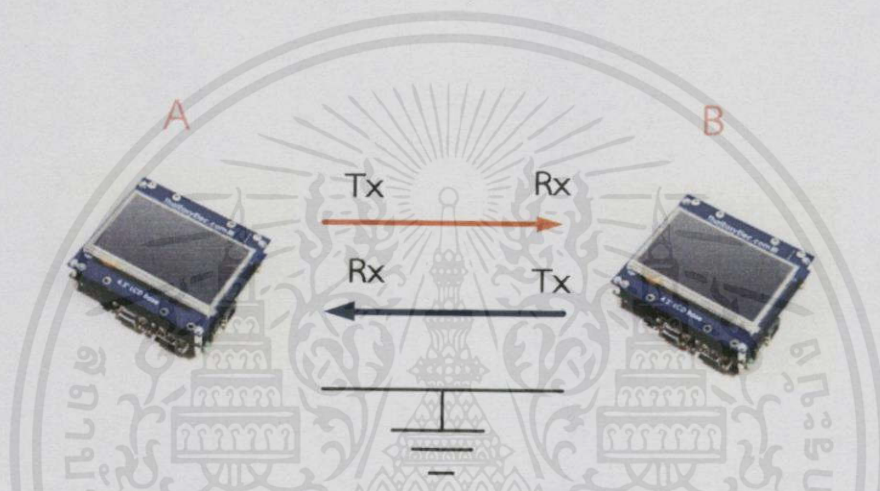
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 การรับส่งข้อมูลแบบ Asynchronous

เริ่มต้นจาก Start Bit เป็น Logic 0 จากนั้นจะตามด้วย Data ที่เราส่ง แล้วจะถูกปิดด้วย STOP Bit เป็น Logic 1

2.4.2 แนวคิดวิธีเชื่อมต่อระหว่างอุปกรณ์



รูปที่ 2.19 การเชื่อมต่อระหว่างบอร์ด SUN7 เพื่อส่ง Data หากันด้วย UART

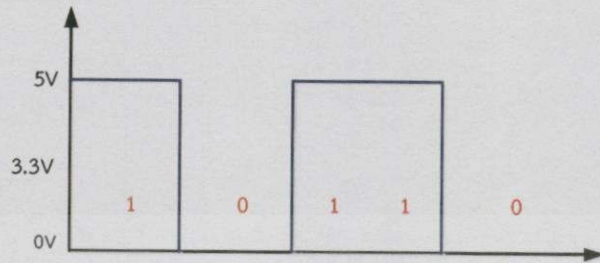
จากรูปแสดงถึงการเชื่อมต่อ ระหว่างบอร์ด SUN7 เพื่อส่ง Data หากันระหว่างบอร์ดด้วย UART

1. A ส่งข้อมูล ออกไปทางขา Tx ไปยัง B ซึ่งเป็นฝั่งรับ เพราะฉะนั้น ต้องต่อสายสัญญาณจากขา Tx ของ A ไปยังขา Rx ของ B
2. B ส่งข้อมูล ออกไปทางขา Tx ไปยัง A ซึ่งเป็นฝั่งรับ เพราะฉะนั้น ต้องต่อสายสัญญาณจากขา Tx ของ B ไปยังขา Rx ของ A
3. ต้องต่อ GND ของทั้ง A และ B ร่วมกันเพื่อให้ระดับแรงดันของทั้ง 2 บอร์ดมีจุดอ้างอิงเดียวกัน

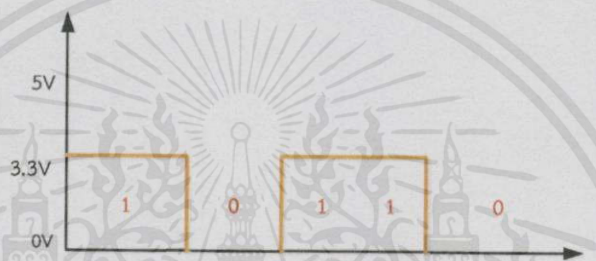
2.4.3 TTL (Transistor-Transistor Logic)

TTL เป็นระดับแรงดันที่ถูกกำหนดขึ้นในยุคแรกๆเพื่อใช้ระหว่าง Transistor กับ Transistor ภายในวงจรรวม(IC) ดังนั้น TTL จะใช้ระดับแรงดัน อยู่ที่ 0 - 5 V แต่ในปัจจุบันมีอุปกรณ์หลายเบอร์ที่ทำงานในช่วง 0 - 3.3 V (เรียกแรงดันระดับนี้ว่า LVTTTL) ซึ่งผู้ใช้ควรตรวจสอบจาก

Datasheet ของอุปกรณ์ที่ใช้เสียก่อนว่าเป็นระดับแรงดันแบบใด เพราะหากใช้ผิดประเภทจะทำให้ อุปกรณ์เสียหาย



TTL 0 – 5V



TTL 0 – 3.3 V

รูปที่ 2.20 ระดับแรงดัน TTL

2.4.4 USB to TTL Converter Module

ในหัวข้อนี้จะกล่าวถึงรายละเอียดของ USB to TTL Converter Module เพื่อใช้ในการเชื่อมต่อระหว่างอุปกรณ์กับคอมพิวเตอร์แบบ UART Serial Interface

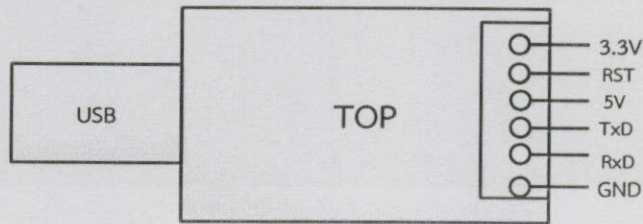
โดยทั่วไปแล้ว การจะเชื่อมต่ออุปกรณ์หรือโมดูลที่มีแรงดันระดับ TTL (3.3V หรือ 5V) เข้ากับเครื่องคอมพิวเตอร์ จะต้องทำการแปลงสัญญาณระดับ TTL ไปเป็นสัญญาณระดับ RS232 เพื่อให้ใช้งานกับคอมพิวเตอร์ได้ ซึ่งจะต้องสร้างวงจร MAX232



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.21 USB to TTL Converter Module

USB to TTL Converter Module เป็นตัวแปลงสัญญาณการเชื่อมต่อแบบ Serial จากอุปกรณ์หรือโมดูลที่แรงดันระดับ TTL (3.3 V หรือ 5 V) ไปเป็นแรงดัน USB มาตรฐานเพื่อเชื่อมต่อเข้าเครื่องคอมพิวเตอร์โดยไม่ต้องใช้วงจรอื่นเพิ่มเติม



รูปที่ 2.22 ขาของ USB to TTL Converter Module

Pin Definition

1. 3.3 Volt
2. RESET (only for programming)
3. 5 Volt
4. Transmit (Sent) Data
5. Receive Data
6. Ground

2.5 มิเตอร์อัจฉริยะ (Smart Meter)

มิเตอร์อัจฉริยะ(Smart Meter) คือมิเตอร์วัดพลังงานไฟฟ้าชนิดใหม่ที่ถูกพัฒนาขึ้นมาเพื่อแทนที่มิเตอร์วัดพลังงานแบบเดิมที่มีอยู่(แบบจานหมุน) โดยการอ่านค่าพลังงานไฟฟ้าที่ถูกใช้ไปด้วยวงจรรีเลย์ทรานซิสเตอร์และส่งข้อมูลในรูปแบบดิจิทัลไปที่ระบบควบคุม(Control System) มิเตอร์อัจฉริยะสามารถติดต่อสื่อสารโดยตรงกับระบบควบคุมได้บ่อยเท่าที่ต้องการ โดยไม่จำเป็นต้องส่งพนักงานมาเพื่ออ่านค่าหน่วยไฟฟ้าเดือนละครั้งแบบปัจจุบัน ดังนั้นผู้ขายไฟฟ้าและผู้ใช้ไฟฟ้าจะสามารถอ่าน หรือคำนวณค่าใช้จ่ายของการใช้พลังงานได้เสมอ ช่องทางการสื่อสารอาจเลือกได้จากหลายรูปแบบ ตัวอย่างเช่น รูปแบบการสื่อสารผ่านทางเครือข่ายโทรศัพท์เคลื่อนที่ หรือรูปแบบการสื่อสารผ่านคลื่น RF เป็นต้น นอกจากนี้การบำรุงรักษาซ่อมแซมตัวมิเตอร์สามารถทำได้ง่ายขึ้นเพราะความซับซ้อนทางด้านชิ้นส่วนกลไกมีน้อยลงกว่าแบบเดิมมาก เมื่อเกิดข้อผิดพลาดขึ้นระบบมิเตอร์อัจฉริยะสามารถแจ้งเหตุความผิดปกติทางไฟฟ้า เช่น ไฟตก ไฟดับได้ ดังนั้นจึงตรวจสอบหาจุดที่เป็นต้นเหตุที่ทำให้ระบบทำงานผิดพลาดได้ไวกว่าเดิมอีกด้วย นอกจากนี้ประโยชน์ที่กล่าวมาแล้ว มิเตอร์อัจฉริยะยังสามารถคำนวณค่าไฟฟ้าที่มีวิธีการคิดที่ซับซ้อนได้ เช่น ตามช่วงเวลา หรือตามความต้องการการใช้ไฟฟ้า ยกตัวอย่างเช่น เมื่อความต้องการไฟฟ้าสูง ราคา ก็จะสูงตามไปด้วย ซึ่งก็จะช่วยให้ผู้ที่ไม่มีความจำเป็นต้องใช้ไฟฟ้าในขณะนั้น ชะลอการใช้ไฟฟ้าไปใช้ในเวลาที่มีความต้องการไฟฟ้า

น้อยกว่า โดยรวมแล้วการคิดค่าไฟฟ้าแบบนี้ จะช่วยลดความต้องการไฟฟ้าสูงสุดลงได้ มาตรการจูงใจยังสนับสนุนการคิดค่าไฟฟ้าแบบจ่ายล่วงหน้าซึ่งอาจจะนำมาใช้ในอนาคต นอกจากนี้การตัดต่อการให้บริการไฟฟ้าแก่ผู้ใช้ไฟฟ้าสามารถทำได้จากระยะไกลได้

2.5.1 ส่วนประกอบของมิเตอร์อัจฉริยะ

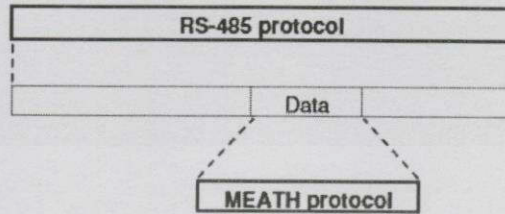
1. ชุดตรวจจับสัญญาณแรงดันไฟฟ้า มีหน้าที่ตรวจจับสัญญาณแรงดันไฟฟ้าเพื่อนำไปประมวลผลในหน่วยประมวลผลกลาง
2. ชุดตรวจจับสัญญาณกระแสไฟฟ้า มีหน้าที่ตรวจจับสัญญาณกระแสไฟฟ้าเพื่อนำไปประมวลผลในหน่วยประมวลผลกลาง
3. หน่วยประมวลผลกลาง มีหน้าที่ควบคุมการทำงานทั้งหมดของมิเตอร์อัจฉริยะ เช่น การติดต่อสื่อสารกับอุปกรณ์ต่างๆ การคำนวณค่าพลังงานไฟฟ้า เป็นต้น
4. อุปกรณ์สื่อสารไร้สาย มีหน้าที่เป็นตัวกลางการติดต่อสื่อสารกับคอมพิวเตอร์ในปัจจุบันมีอยู่มากมายหลายรูปแบบดังนี้
 - บลูทูธ อุปกรณ์สื่อสารแบบสองทิศทาง ขนาดเล็ก ราคาถูก ใช้งานง่าย และใช้พลังงานต่ำ โดยสื่อสารคลื่นวิทยุระยะสั้น ระยะทางการติดต่อประมาณ 10 เมตร
 - ริงส์อินฟราเรด อุปกรณ์สื่อสารแบบสองทิศทางขนาดเล็ก ใช้พลังงานต่ำ ระยะทางการติดต่อสื่อสารประมาณ 1 เมตร โดยห้ามมีสิ่งกีดขวาง
 - Zigbee อุปกรณ์สื่อสารไร้สายทั้งแบบทางเดียวและสองทิศทาง อัตราการรับส่งข้อมูลต่ำ สามารถสร้างโครงข่ายตรวจจับไร้สายได้
 - ระบบเครือข่ายไร้สาย อุปกรณ์สื่อสารแบบสองทิศทาง โดยมาตรฐาน IEEE 802.11 สามารถเข้าถึงระบบ Ethernet ได้ผ่านอุปกรณ์ขยายสัญญาณ (Access Point)
 - การสื่อสารผ่านสายไฟฟ้า (Power Line Communication) การสื่อสารผ่านสายไฟฟ้า เชื่อมต่อผ่านอแดปเตอร์ ติดตั้งง่าย มีอัตราการรับส่งข้อมูลสูง แต่อาจได้รับผลกระทบจากสัญญาณรบกวนขึ้นที่สายไฟฟ้า
5. แหล่งจ่ายไฟกระแสตรง มีหน้าที่จ่ายไฟเลี้ยงให้กับมิเตอร์ไฟฟ้าอัจฉริยะในการทำงาน
6. หน้าจอแสดงผล มีหน้าที่ในการแสดงข้อมูลต่างๆ ตามที่หน่วยประมวลผลกลางสั่ง

2.5.2 ข้อมูลจำเพาะของโปรโตคอลมิเตอร์ไฟฟ้าที่ใช้งาน

มิเตอร์อิเล็กทรอนิกส์ SX1 รุ่น A31N ของ Mitsubishi Electric Automation (Thailand) Co., Ltd.

เอกสารนี้เป็นเอกสารที่ขอยืมไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

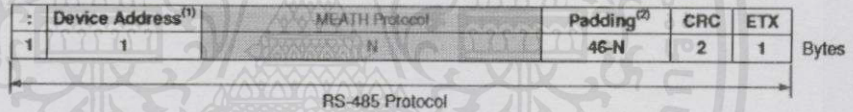
การสื่อสารข้อความประกอบด้วย 2 โพรโตคอล โพรโตคอลแรกคือโพรโตคอล RS-485 ที่ครอบคลุมข้อความทั้งหมดและโพรโตคอลที่สองคือโพรโตคอล MEATH ซึ่งเป็นโพรโตคอลที่อยู่ภายใน ชั้นของโพรโตคอลแสดงได้ดังรูปที่ 2.23



รูปที่ 2.23 ชั้นของโพรโตคอล

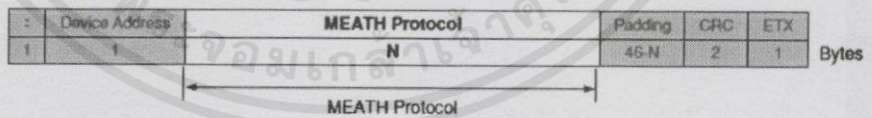
โครงสร้างของโพรโตคอล RS-485

โครงสร้างข้อความ RS-485 สำหรับข้อความทุกชนิด(ส่งข้อความ,รับข้อความ,รับเชื่อมต่อข้อความ) มีลักษณะเดียวกัน แสดงได้ดังรูปที่ 2.24



รูปที่ 2.24 โครงสร้างข้อความ RS-485

โครงสร้างของโพรโตคอล MEATH



รูปที่ 2.25 โครงสร้างโพรโตคอล MEATH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Arduino

2.6.1 ข้อมูลเบื้องต้นของ Arduino

Arduino เป็นภาษาอิตาลี ซึ่งใช้เป็นชื่อของโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูล AVR แบบ Open Source ที่ได้รับการปรับปรุงมาจากโครงการพัฒนา Open Source ของ AVR อีกโครงการหนึ่งที่มีชื่อว่า "Wiring" แต่เนื่องจากโครงการของ "Wiring" เลือกใช้ AVR เบอร์ ATmega128 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีจำนวนของหน่วยความจำและ I/O ค่อนข้างมาก และที่สำคัญ ATmega128 เป็นชิปที่มีตัวถังแบบ SMD จึงทำให้เป็นอุปสรรคสำหรับผู้เริ่มต้นในการสร้างบอร์ดและต่อวงจรขึ้นมาใช้งานกันเอง และบอร์ดจะมีขนาดค่อนข้างใหญ่ จึงไม่ได้รับความนิยมเท่าที่ควร แต่หลังจากที่ทางทีมงาน Arduino นำ Source Code ของ "Wiring" มาปรับปรุงใหม่โดยให้สามารถใช้งานกับไมโครคอนโทรลเลอร์ AVR ขนาดเล็ก อย่าง Mega8 และ Mega168 ได้ จึงทำให้ระบบวงจรของบอร์ดมีขนาดเล็กกว่า "Wiring" มากและยังใช้อุปกรณ์น้อยชิ้น ทำให้ง่ายต่อการต่อวงจรใช้งานกันเอง และยังประหยัดต้นทุนในการสร้างบอร์ดไปได้มาก ด้วยเหตุนี้เองทำให้ "Arduino" ได้รับความนิยมจากผู้ใช้งานทั่วโลกเป็นอย่างมากในระยะเวลาอันรวดเร็ว

Arduino มีจุดเด่นในเรื่องของความง่ายในการเรียนรู้และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่างๆ ขึ้นมาสนับสนุนการใช้งานด้วยรูปแบบที่ง่าย ไม่ซับซ้อน ซึ่งถึงแม้ว่า Arduino เองจะมีรูปแบบการใช้งานคล้ายๆ กันกับไมโครคอนโทรลเลอร์อย่าง Basic Stamp ของ Parallax, BX-24 ของ Netmedias และ Handy Board ของ MIT แต่ก็มีความโดดเด่นกว่าของรายการอื่นๆ หลายอย่าง เป็นต้นว่า

- ราคาไม่แพง เนื่องจากมี Source Code และวงจรแจกให้ฟรี สามารถต่อวงจรขึ้นมาใช้งานได้เอง
- โปรแกรมที่ใช้พัฒนาของ Arduino รองรับการทำงานทั้ง Windows, Linux และ Macintosh OSX
- มีรูปแบบคำสั่งที่ง่ายต่อการใช้งาน แต่สามารถนำไปใช้งานได้จริงที่มีความซับซ้อนมากๆ ได้และยังสามารถสร้างคำสั่งและ Library ใหม่ๆ ขึ้นมาใช้งานได้เมื่อมีความชำนาญมากขึ้นแล้ว
- มีการเปิดเผยวงจรและ Source Code ทั้งหมดทำให้สามารถนำไปพัฒนาต่อยอดเพิ่มเติมได้ตามความต้องการทั้ง Hardware และ Software

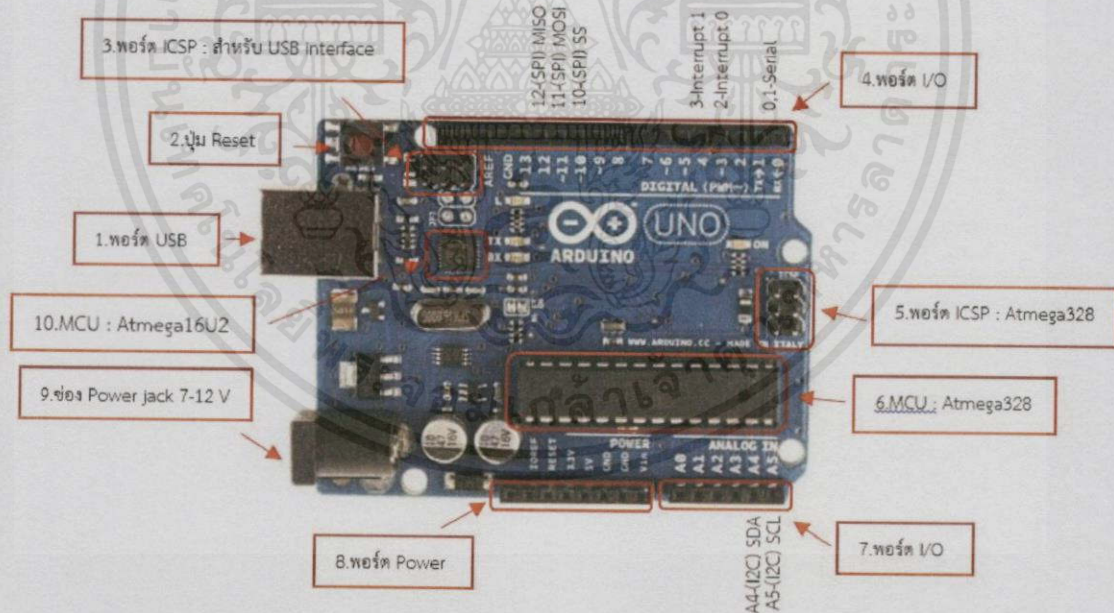
2.6.2 โปรแกรมภาษาของ Arduino

โปรแกรมภาษาของ Arduino จะใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่งที่มีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับ ภาษาซีมาตรฐาน (ANSI-C) อื่นๆ เพียงแต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรมและให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐานของ ANSI-C โดยตรง ซึ่งในความเป็นจริงแล้วในการเขียนโปรแกรมของ Arduino เราสามารถใช้คำสั่งต่างๆ ที่เป็นคำสั่งตามมาตรฐานของ ANSI-C เข้ามาใช้ในการเขียนโปรแกรมได้ทันที โดยรูปแบบการเขียนโปรแกรมและการใช้งานคำสั่งต่างๆ นั้นสามารถอ้างอิงจากหนังสือ ตำรา ของภาษาซีมาตรฐาน ANSI-C ได้โดยตรง

การใช้งานงานArduino ในความเป็นจริงแล้วArduinoนั้นไม่ใช่ C-Compilerโดยตรง แต่Arduinoจะมีลักษณะการทำงานเช่นเดียวกับ Text Editor ของภาษา C++ ตัวหนึ่งโดยจะทำงานร่วมกับ Utility บางส่วนที่Arduinoสร้างขึ้นมารองรับ โดยArduinoจะใช้รูปแบบการทำงานของText Editor เป็นฉากหน้าในการติดต่อสื่อสารกับผู้ใช้เท่านั้น ส่วนเบื้องหลังจริงๆนั้นArduinoจะไปเรียกใช้ตัวแปลภาษาซีและ Utility อื่นๆที่ใช้เป็นเครื่องมือในการพัฒนาโปรแกรมของไมโครคอนโทรลเลอร์ตระกูลAVRอีกทีหนึ่ง โดยArduinoจะเลือกใช้ C-Compiler ของ “GNU AVR-GCC Toolchain” ร่วมกับ Library Function ของ “avr-libc” Utility ที่ใช้ในการ Upload Code ให้กับAVR นั้นก็จะใช้ของ “AVRDude” ดังนั้นผู้ที่เขียนภาษาซีของAVRเป็นอยู่แล้วและต้องการประยุกต์ใช้งานArduino ให้ได้ประสิทธิภาพการทำงานมากยิ่งขึ้นไปอีก ก็สามารถศึกษาข้อกำหนดและหน้าที่ในการใช้งาน Library และคำสั่งอื่นๆที่บรรจุไว้ในLibraryต่างๆทั้งจากของ “GNU AVR-GCC Toolchain” และ “avr-libc” เพิ่มเติมอีกเพื่อใช้เป็นแนวทางในการปรับปรุงและประยุกต์ใช้งานArduinoในรูปแบบที่สลับซับซ้อนมากขึ้นไปได้อีก

2.6.3 Arduino Uno R3

เป็นบอร์ด Arduino ที่ได้รับความนิยมมากที่สุด เนื่องจากราคาไม่แพง ส่วนใหญ่โปรเจคและ Library ต่างๆที่พัฒนาขึ้นมา Support จะอ้างอิงกับบอร์ดนี้เป็นหลัก และข้อดีอีกอย่างคือ กรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ง่าย



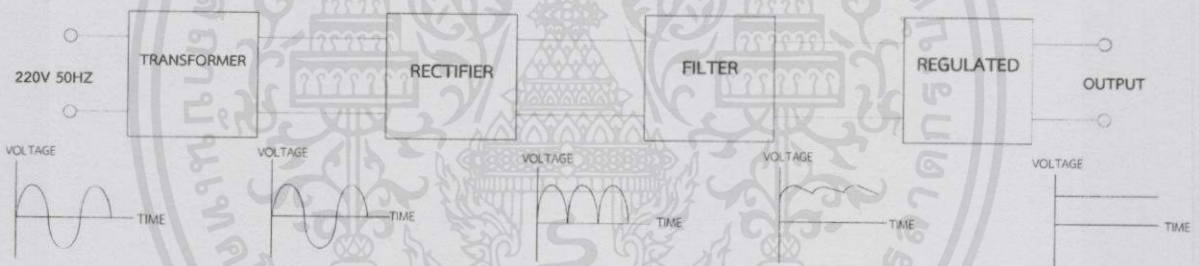
รูปที่ 2.26 Arduino Uno R3

1. USBPort: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. Reset Button: เป็นปุ่มReset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
3. ICSP Port ของ Atmega16U2: เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
4. I/OPort: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆเพิ่มเติมด้วย

- เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
5. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
 6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
 7. I/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็นช่องรับสัญญาณอนาล็อกตั้งแต่ขา A0-A5
 8. PowerPort: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอกประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, V_{in}
 9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
 10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

2.7 แหล่งจ่ายไฟฟ้า (Power Supply)

ในวงจรอิเล็กทรอนิกส์ทุกวงจรเกือบ 100% จะต้องมีแหล่งจ่ายไฟเลี้ยงให้วงจร แต่แหล่งกำเนิดแรงดันไฟฟ้าที่หาได้ง่ายที่สุดก็คือ ขนาดแรงดันไฟ 220 V 50Hz ที่มีอยู่ตามอาคาร บ้านเรือน หรือในโรงงานอุตสาหกรรม ดังนั้นจึงต้องมีการออกแบบแหล่งจ่ายไฟที่มีแหล่งกำเนิดมาจากไฟฟ้า 220V 50Hz ให้ตรงกับความต้องการในการใช้งานจริง



รูปที่ 2.27 Block Diagram แหล่งจ่ายไฟฟ้า

จาก Block Diagram เริ่มต้นจาก Power Line (AC 220 โวลต์) ป้อนให้กับหม้อแปลง (Transformer) เพื่อลดระดับแรงดันให้อยู่ในระดับที่เหมาะสม จากนั้นเข้าวงจรเรียงกระแส (Rectifier) ซึ่งทำหน้าที่เปลี่ยนกระแสสลับให้เป็นกระแสตรง ไฟกระแสตรงที่ได้จากส่วนนี้จะยังมีลักษณะเป็นลูกคลื่นอยู่จึงต้องนำไปกรองให้เรียบด้วยวงจรกรองกระแส (Filter) และส่วนสุดท้ายเป็นวงจรควบคุมแรงดัน (Voltage Regulator) ซึ่งทำหน้าที่ควบคุมรักษาระดับแรงดัน Output ของวงจรให้มีระดับคงที่ตามต้องการเสมอ ไม่เปลี่ยนแปลงตามสภาวะของ Load ต่อไปเราจะได้ศึกษาถึงส่วนต่างของ Power Supply

2.7.1 หม้อแปลงไฟฟ้า (Transformer)

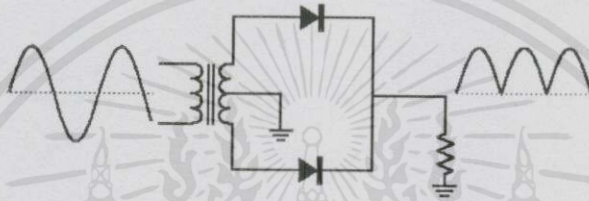
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ในการเลือกใช้หม้อแปลงไฟฟ้ามีหลักการคือ จะต้องเลือกหม้อแปลงที่สามารถจ่ายแรงดันและกระแสได้เพียงพอกับการใช้งานจริง แต่ก็ไม่ควรที่จะเลือกหม้อแปลงที่มีขนาดใหญ่ (จ่าย

แรงดันและกระแสได้สูง) เพราะจะทำให้สิ้นเปลืองค่าใช้จ่าย ดังนั้นในการเลือกหม้อแปลงไฟฟ้าให้มีขนาดพอเหมาะกับการที่ต้องการก็จะพิจารณาขนาดของแรงดันไฟตรง (VO) และกระแสไฟตรงสูงสุด (IO) ที่สามารถจ่ายให้โหลดได้ แล้วนำมากำหนดขนาดของแรงดันขดทุติยภูมิ (Secondary) และกระแสใช้งานที่ขดทุติยภูมิของหม้อแปลง

2.7.2 วงจรเรียงกระแส (Rectifier)

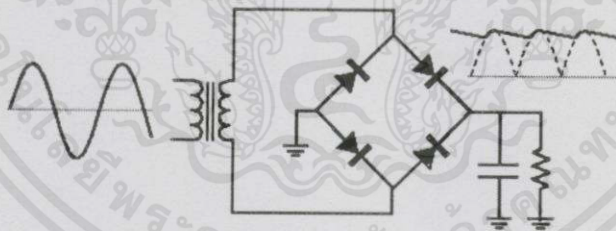
วงจร Rectifier จะทำหน้าที่เป็นตัวเปลี่ยนกระแสไฟสลับที่มาจากหม้อแปลงให้เป็นไฟฟ้ากระแสตรง โดยวงจร Rectifier ที่นิยมใช้ในทางปฏิบัติจะมีอยู่ 2 ชนิด

1. Center-tapped Full-wave Rectifier



รูปที่ 2.28 Center-tapped Full-wave Rectifier

2. Full wave Bridge Rectifier สามารถให้แรงดันไฟตรงออกมาในลักษณะ Full wave ได้โดยไม่ต้องใช้หม้อแปลงที่มี Tap กลาง ดังวงจรข้างล่าง



รูปที่ 2.29 Full wave Bridge Rectifier

2.7.3 วงจรกรองกระแส (Filter)

วงจร Filter ใน Power Supply ส่วนใหญ่จะประกอบด้วยตัวเก็บประจุ โดยหน้าที่ของ Capacitor Filter นี้คือเก็บประจุเอาไว้แล้วจ่ายออกในช่วงที่ไม่มีแรงดันออกมาซึ่งจะทำให้แรงดัน output ที่ได้มีความราบเรียบยิ่งขึ้น ซึ่งวงจรกรองกระแสนี้จะต่ออยู่ระหว่าง Rectifier และ Voltage Regulator โดยค่าความเรียบของรูปคลื่นนี้จะบอกเป็น Ripple Factor โดยที่

$$\text{Ripple Factor}(r) = \frac{V_r}{V_{dc}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
หรือ

$$\% \text{Ripple Factor}(r) = \frac{V_r}{V_{dc}} \times 100$$

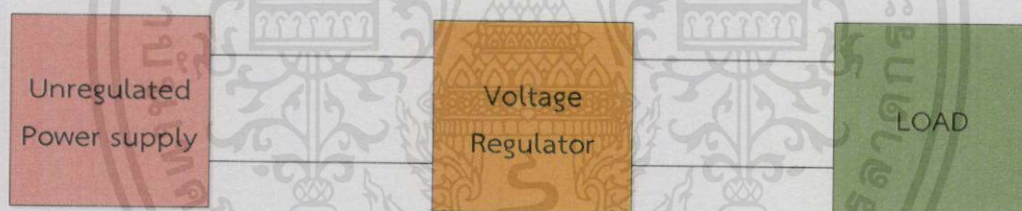
เมื่อ V_r คือค่าแรงดันที่กระเพื่อมอยู่บนแรงดันไฟตรง DC

2.7.4 วงจรรักษาระดับแรงดัน (Voltage Regulators)

วงจรรักษาระดับแรงดัน (Voltage Regulators) มีหน้าที่รักษาระดับแรงดันไฟตรงที่จ่ายให้โหลดมีค่าคงที่ ไม่เปลี่ยนแปลงตามสภาพโหลด โดยทั่วไป สามารถแบ่งชนิดของ Regulator ออกเป็น 2 ชนิด คือ Series Regulator และ Switching Regulator โดย Series Regulator จะควบคุมแรงดันเอาต์พุตให้คงที่ได้โดยการใช้ตัวเปรียบเทียบแรงดันกับแรงดันเอาต์พุตที่ย้อนกลับมา ส่วน Switching Regulator นั้นจะทำงานในลักษณะ On-off ก็将通过วงจร LC Filter เพื่อกรองกระแสให้เรียบเป็นไฟตรง ส่วนการควบคุมแรงดันเอาต์พุตให้คงที่ ทำได้โดยการปรับ ค่าเวลาในการ On (ton)

วงจรรักษาระดับแรงดันจะต่อระหว่างแหล่งจ่ายไฟตรงไม่คงค่า (Unregulated Power Supply) กับโหลด มีหน้าที่จ่ายไฟตรงให้กับโหลดและสามารถรักษาแรงดันให้คงตัวขณะที่

1. โหลดเปลี่ยน นั่นคือกระแสขาออกเปลี่ยนแปลง
2. แรงดันของแหล่งจ่ายไฟตรงไม่คงค่า เปลี่ยนแปลงทั้งนี้รวมถึงระลอกคลื่นด้วย (ripple)
3. อุณหภูมิของวงจรเปลี่ยนแปลง

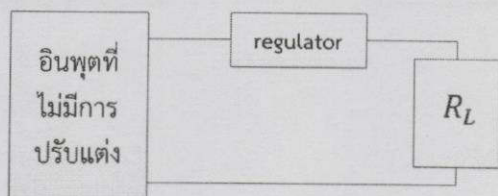


รูปที่ 2.30 บล็อกไดอะแกรมของแหล่งจ่ายไฟตรงคงค่าแรงดัน

จากบล็อกไดอะแกรมจะเห็นได้ว่าสัญญาณที่ออกมาจากเอาต์พุตของวงจรกรองแรงดันคือบล็อกแรกในรูปที่ 2.30 ซึ่งยังไม่เรียบเท่าที่ควร ยังมีการเปลี่ยนแปลงขึ้นๆลงๆในลักษณะคล้ายสัญญาณฟันเลื่อยเพราะว่าตัวเก็บประจุจะเก็บและคายประจุอยู่ตลอดเวลา ถ้าจะนำวงจรนี้ไปใช้งานเลยก็ได้แต่ต้องใช้กับงานที่ไม่ต้องการความละเอียดของแหล่งจ่ายไฟมากนัก

แต่ถ้าเป็นงานที่ต้องการความละเอียดของแหล่งจ่ายไฟวงจรนี้จะใช้งานไม่ได้ หนทางที่จะนำมาแก้ไขคือ ใช้วงจรเรกกูเลเตอร์ซึ่งจะทำหน้าที่ปรับแต่งสัญญาณที่ไม่ราบเรียบอยู่นั้น ให้เรียบจนเกือบเป็นไฟ DC 100% การสร้างวงจรเรกกูเลเตอร์จะมีอยู่ 2 ลักษณะคือ การสร้างแบบอนุกรมกับโหลด และแบบขนานกับโหลดดังรูปที่ 2.31 โดยใช้ ซีเนอร์ไดโอด ทรานซิสเตอร์ และไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก.แบบอนุกรมกับโหลด



ข.แบบขนานกับโหลด

รูปที่ 2.31 วงจรRegulator

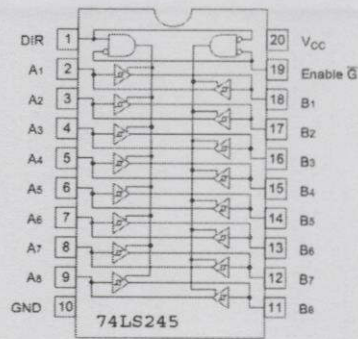
วงจรแหล่งจ่ายไฟสมัยใหม่ที่นิยมกันมากที่สุดคือ วงจรเรกกูเลเตอร์โดยใช้ไอซี เพราะให้ออกแบบวงจรง่าย ราคาถูก ขนาดเล็กและให้คุณภาพสูง ในการออกแบบวงจรเรกกูเลเตอร์โดยใช้ไอซีในลักษณะต่างๆดังนี้คือ วงจรเรกกูเลเตอร์โดยใช้ IC หลายขา วงจรเรกกูเลเตอร์โดยใช้ IC 3 ขา แบบแรงดันเอาต์พุตคงที่บวกและลบ และวงจรเรกกูเลเตอร์โดยใช้ไอซี 3 ขาแบบปรับค่าแรงดันเอาต์พุตค่าได้

วงจรเรกกูเลเตอร์โดยใช้ไอซีจะได้สัญญาณแรงดันอินพุตมาจากวงจรเรกติไฟเออร์ที่ผ่านการกรองแรงดันมาแล้วโดยวงจรเรกกูเลเตอร์จะทำหน้าที่ปรับแต่งแรงดันให้เรียบขึ้นและรักษาระดับแรงดันให้คงที่ตลอดการใช้งาน

2.8 ไอซีบัฟเฟอร์เบอร์ 74LS245

IC เบอร์ 74LS245 จะทำหน้าที่เป็นบัฟเฟอร์ (Buffer) ข้อมูลขนาด 8 บิต ในกรณีที่ต้องการเชื่อมต่อพอร์ตกับอุปกรณ์ภายนอก ในกรณีที่ไอซีไมโครคอนโทรลเลอร์จะไม่สามารถที่จ่ายกระแสมากเกินไปกว่า ที่จะขับอุปกรณ์เอาต์พุตโดยตรงได้ ดังนั้นจึงต้องผ่านไอซีที่ทำหน้าที่บัฟเฟอร์เสียก่อน จึงเปรียบเสมือนเป็นอุปกรณ์ป้องกันหรือตัวกันชนให้กับไอซีไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.32 โครงสร้างของ IC 74LS245

ตารางที่ 2.2 กำหนดอินพุตของ IC 74LS245

INPUTS		OPERATION
Enable G	DIR	
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H หมายถึง ลอจิก 1

L หมายถึง ลอจิก 0

X หมายถึง ไม่สนใจว่าเป็นสภาวะ ลอจิก 0 หรือ 1

2.8.1 หลักการทำงานของไอซี 74LS245

ไอซีเบอร์ 74LS245 ชื่อ Octal Bus Transceivers with 3 state Outputs ทำหน้าที่อยู่ในส่วนงานระบบบัส สามารถส่งผ่านข้อมูลแบบขนานขนาด 8 บิตสองทิศทาง จากรูปมองจากด้านบนของไอซีขาสองข้างส่งผ่านข้อมูลถึงกันเป็นคู่เช่น ขา 2 คู่กับขา 18, ขา 3 คู่กับขา 17 และขา 4 คู่กับขา 16 เป็นต้นโดยจะมีทั้งหมด 8 คู่

2.8.2 การทำงานของขาควบคุมของไอซี 74LS245

ขา 1 ใช้กำหนดทิศทางในการส่งผ่านข้อมูล โดยเมื่อเป็นลอจิก “1” จะเป็นการส่งข้อมูลจากด้าน A ไปยังด้าน Bขา 19 ทำงานควบคุมการยอมให้ส่งผ่านข้อมูลหรือไม่ โดยเมื่อเป็นลอจิก “0” จะกำหนดให้ไอซีตัวนั้นอยู่ในสภาวะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

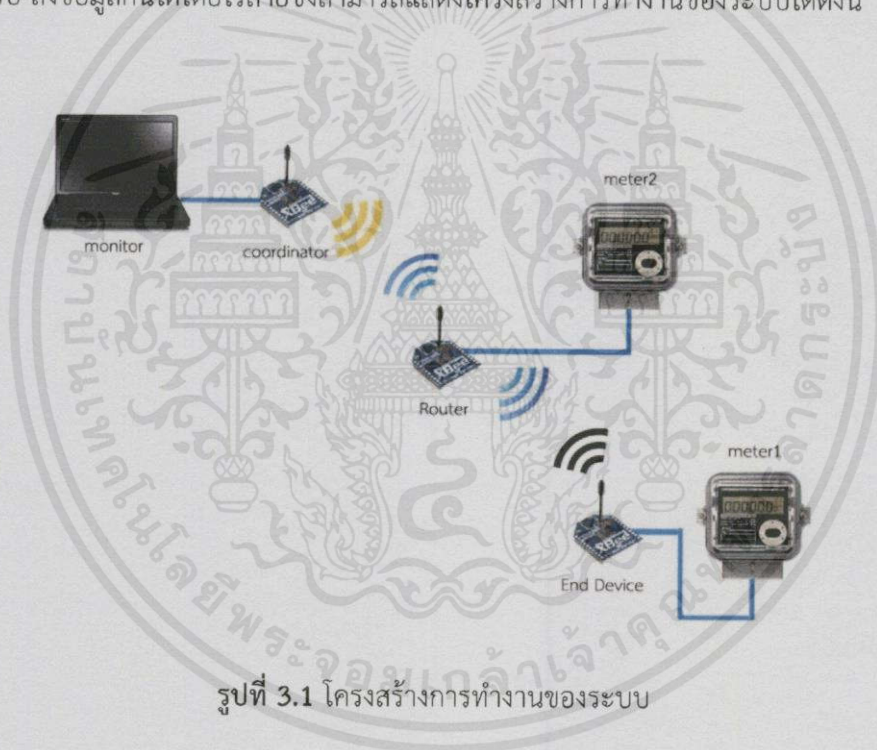
บทที่3

การออกแบบระบบการทำงาน

3.1 การออกแบบโครงสร้างการทำงานของระบบ

การทำงานโดยภาพรวมจะสร้างเครือข่ายการติดต่อสื่อสารโดยใช้ Zigbee เป็นตัวกลางในการติดต่อสื่อสาร โดยมีจุดประสงค์เพื่อการติดต่อสื่อสารแบบไร้สายระหว่างศูนย์กลางการควบคุมและแสดงข้อมูลการใช้ไฟฟ้ากับมิเตอร์อัจฉริยะ ในบทนี้จะอธิบายการออกแบบและการสร้างอุปกรณ์ในแต่ละส่วนอีกทั้งแสดงแนวคิดในการออกแบบ

เพื่อให้สะดวกต่อการใช้งานและยังลดปริมาณและความยาวของสาย จึงมีการออกแบบอุปกรณ์ในการรับ-ส่งระหว่างมิเตอร์อัจฉริยะกับศูนย์กลางควบคุมและแสดงข้อมูล ให้สามารถรับ-ส่งข้อมูลกันได้โดยไร้สายซึ่งสามารถแสดงโครงสร้างการทำงานของระบบได้ดังนี้



รูปที่ 3.1 โครงสร้างการทำงานของระบบ

จากรูปที่ 3.1 รูปโครงสร้างการทำงานของระบบ โดยแบ่งส่วนระบบของโครงงานได้เป็น3ส่วนด้วยกัน คือ ส่วนโหนดอุปกรณ์ปลายทาง (End Device Node) ส่วนโหนดเราเตอร์ (Router Node) และส่วนโหนดโคออดิเนเตอร์ (Coordinator Node)

ในส่วนโมดูล Zigbee ทำหน้าที่เป็นโหนดโคออดิเนเตอร์นี้จะเป็นส่วนของศูนย์กลางการควบคุมและแสดงข้อมูล ซึ่งเป็นตัวกำหนดทิศทางในการติดต่อขอรับข้อมูลการใช้ไฟฟ้าจากมิเตอร์ไฟฟ้าที่เชื่อมต่ออยู่กับโหนดต่างๆภายในเครือข่าย เพื่อนำข้อมูลการใช้ไฟฟ้าไปแสดงผลที่หน้าจอดีไซน์คอมพิวเตอร์โดยผ่านโปรแกรมแสดงข้อมูลการใช้ไฟฟ้า

ในส่วนของโหนดเราเตอร์นั้นจะเป็นตัวจัดการโหนดอุปกรณ์ปลายทางในโครงข่าย โดยจะรับคำสั่งในการติดต่อกับโหนดอุปกรณ์ปลายทางมาจากส่วนของศูนย์กลางการควบคุมและแสดงข้อมูล (โหนดโคออดิเนเตอร์) และเป็นตัวกลางในการส่งข้อมูลการใช้ไฟฟ้าจากมิเตอร์ไฟฟ้า (โหนดอุปกรณ์ปลายทาง) ที่อยู่ไกลจากศูนย์กลางการควบคุมและแสดงข้อมูล ไปให้กับศูนย์กลางการควบคุมและแสดงข้อมูล (โหนดโคออดิเนเตอร์) เพื่อทำการแสดงผลบนหน้าจอกอมพิวเตอร์ผ่านโปรแกรมแสดงข้อมูลการใช้ไฟฟ้า

ในส่วนของโหนดอุปกรณ์ปลายทางนั้น เราจะทำการจำลองสร้างชุดข้อมูลของมิเตอร์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวกำหนด เมื่อมีการติดต่อขอรับข้อมูลมาจากส่วนของศูนย์กลางการควบคุมและแสดงข้อมูล (โหนดโคออดิเนเตอร์) โหนดอุปกรณ์ปลายทางจะทำการส่งข้อมูลให้กับโหนดเราเตอร์ที่เป็นตัวกลางเพื่อจะนำข้อมูลส่งต่อไปให้กับศูนย์กลางการควบคุมและแสดงข้อมูลเพื่อแสดงผลบนหน้าจอกอมพิวเตอร์ต่อไป

3.2 หลักการออกแบบวงจร

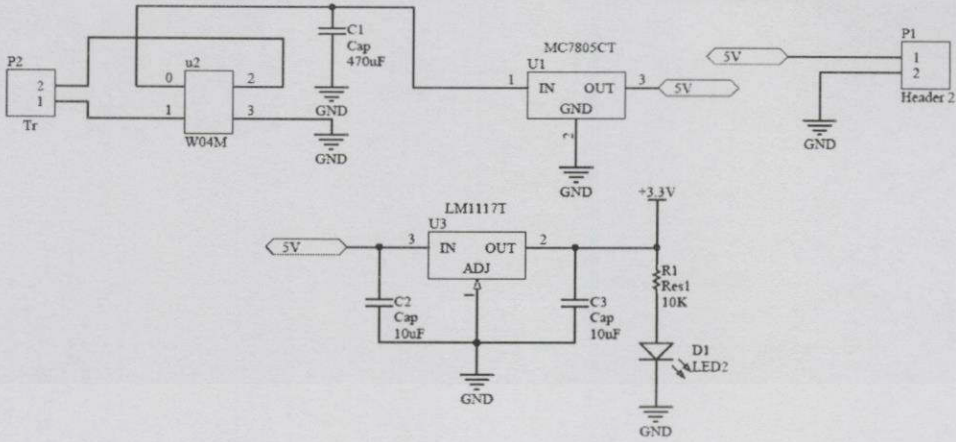
ในโครงงานนี้ได้ทำการออกแบบลายวงจรทั้งหมด3ส่วนด้วยกัน โดยส่วนแรกเป็นวงจรเชื่อมต่อโมดูลZigbeeทำหน้าที่เป็นโหนดอุปกรณ์ปลายทาง ส่วนที่สองเป็นวงจรเชื่อมต่อโมดูลZigbeeทำหน้าที่เป็นโหนดเราเตอร์และส่วนที่สามเป็นวงจรเชื่อมต่อโมดูลZigbeeทำหน้าที่เป็นโหนดโคออดิเนเตอร์

3.2.1 การออกแบบวงจรเชื่อมต่อโมดูลZigbeeทำหน้าที่เป็นโหนดอุปกรณ์

ปลายทาง

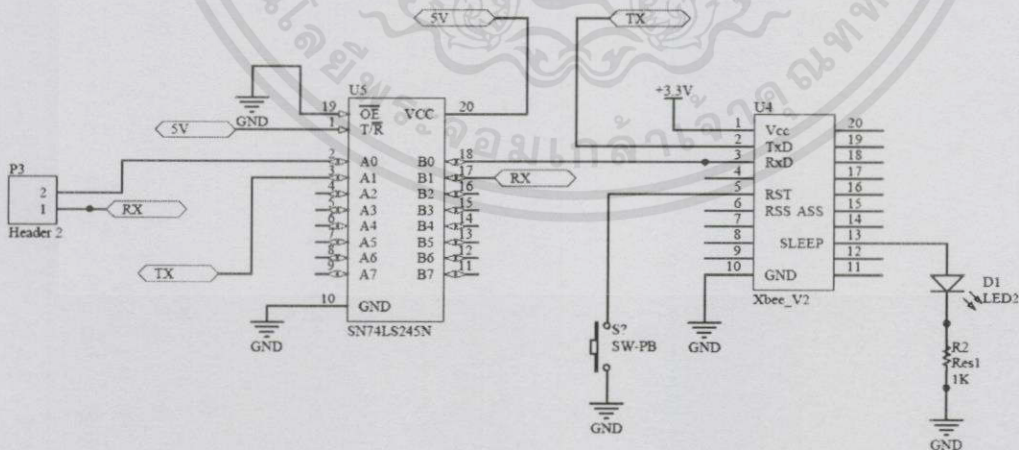
ในส่วนของวงจรเชื่อมต่อโมดูลZigbeeทำหน้าที่เป็นโหนดอุปกรณ์ปลายทางนี้จะต้องใช้โมดูลZigbeeเชื่อมต่อกับไมโครคอนโทรลเลอร์ที่ทำการจำลองการส่งชุดข้อมูลของมิเตอร์ไฟฟ้า ซึ่งโครงงานนี้ได้ใช้Arduino Uno R3 เป็นไมโครคอนโทรลเลอร์ที่ใช้จำลองการส่งชุดข้อมูลของมิเตอร์ไฟฟ้าให้กับโมดูลZigbeeโดยผ่านขาTx (Transmitter) และขาRx (Receiver) รายละเอียดส่วนประกอบต่างๆของวงจรมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจรไฟเลี้ยงของโมดูลZigbeeทำหน้าที่เป็นโหนดอุปกรณ์ปลายทาง

จากรูปที่ 3.2 วงจรไฟเลี้ยงของโมดูลZigbeenั้นเราจะใช้หม้อแปลงขนาด 220/6 โวลต์ โดยจะรับแรงดันในระดับ220โวลต์และในฝั่งที่ออกมาจากหม้อแปลง 6 โวลต์จะนำไปเชื่อมต่อกับส่วนแหล่งจ่ายแรงดัน(Supply Voltage)ของวงจร จากนั้นจะผ่าน Bridge Rectifier เพื่อแปลงสัญญาณไฟฟ้ากระแสสลับ(AC)ให้เป็นสัญญาณไฟฟ้ากระแสตรง(DC) และนำไปผ่านตัวเก็บประจุกรองแรงดัน(Capacitor Filter) เพื่อเป็นการลดระลอกคลื่น(ripple) ทำให้แรงดันไฟตรงเรียบขึ้น หลังจากนั้นจะผ่านไอซีลดระดับแรงดัน (Voltage Regulator IC) เบอร์ 7805 เพื่อลดระดับแรงดันให้เหลือ 5 โวลต์ เพื่อนำไปใช้เป็นไฟเลี้ยงให้กับ Arduino Uno R3 และไอซีลดระดับแรงดันเบอร์ 1117T เพื่อลดระดับแรงดันให้เหลือ3.3 โวลต์ และนำไปใช้เป็นไฟเลี้ยงให้กับโมดูลZigbee



รูปที่ 3.3 วงจรการทำงานของโมดูลZigbeeทำหน้าที่เป็นโหนดอุปกรณ์ปลายทาง

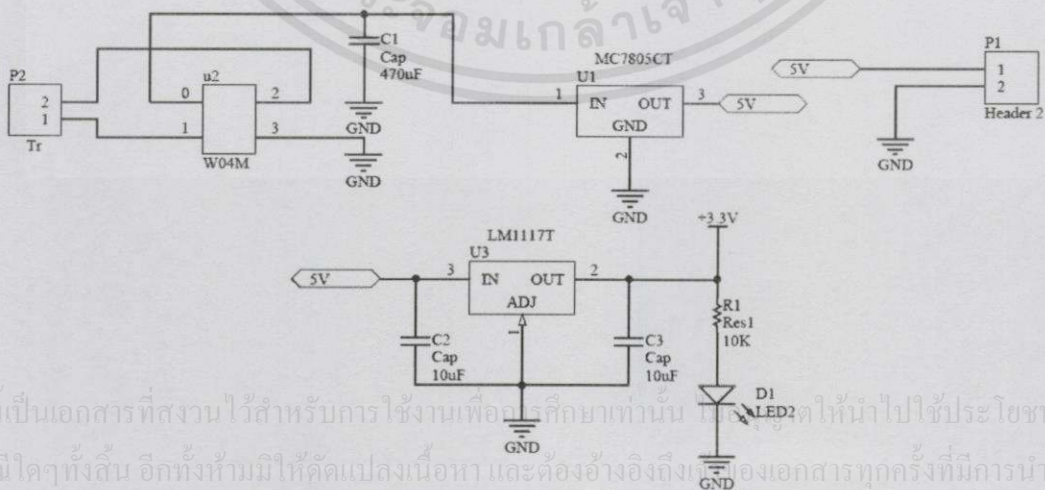
เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับนักเรียนที่เข้าศึกษาในชั้นเรียนของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.3 ส่วนของวงจรแปลงระดับสัญญาณลอจิก จะใช้ไอซี 74LS245 เพื่อแก้ไขปัญหาข้อมูลที่ส่งแบบ UART ระหว่างอุปกรณ์ที่ใช้ระดับสัญญาณลอจิกต่างกันเกิดความผิดพลาด เนื่องจาก Arduino Uno R3 นั้นใช้ระดับสัญญาณลอจิก 5 โวลต์แต่โมดูล Zigbee จะใช้ระดับสัญญาณลอจิก 3.3 โวลต์ โดยสัญญาณที่ออกจากขาที่กำหนดให้เป็นขา Tx ของ Arduino Uno R3 จะเชื่อมต่อกับขาที่ 2 ของไอซี 74LS245 สัญญาณที่ออกจากขาที่กำหนดให้เป็นขา Rx ของ Arduino Uno R3 จะเชื่อมต่อกับขาที่ 17 ของไอซี 74LS245 ส่วนสัญญาณที่ออกจากขาที่ 18 ของไอซี 74LS245 จะเชื่อมต่อกับขาที่ 3 ซึ่งเป็นขา Rx ของโมดูล Zigbee ขาที่ 3 ของไอซี 74LS245 จะเชื่อมต่อกับขาที่ 2 ซึ่งเป็นขา Tx ของโมดูล Zigbee

ในส่วนของโมดูล Zigbee นั้น ไฟเลี้ยงขนาด 3.3 โวลต์ จะถูกต่อเข้ากับขาที่ 1 (Vcc) ของโมดูล Zigbee นำสวิตช์รีเซตต่อเข้ากับขาที่ 5 (RST) ไว้เพื่อรีเซ็ตโมดูล Zigbee ในกรณีที่มีการทำงานผิดพลาด ส่วนหลอด LED ที่ใช้เพื่อแสดงสถานะ ON/SLEEP ของโมดูล Zigbee ในขาที่ 13 จะเห็นการทำงานเมื่อ LED ที่ขาที่ 13 ติด (ON) หมายความว่าโมดูล Zigbee อยู่ในสถานะพร้อมที่จะติดต่อสื่อสาร หรือ เมื่อโมดูล Zigbee เชื่อมต่ออยู่แล้วแต่หลอด LED ดับ หมายความว่าโมดูล Zigbee อยู่ในสถานะ Sleep Mode

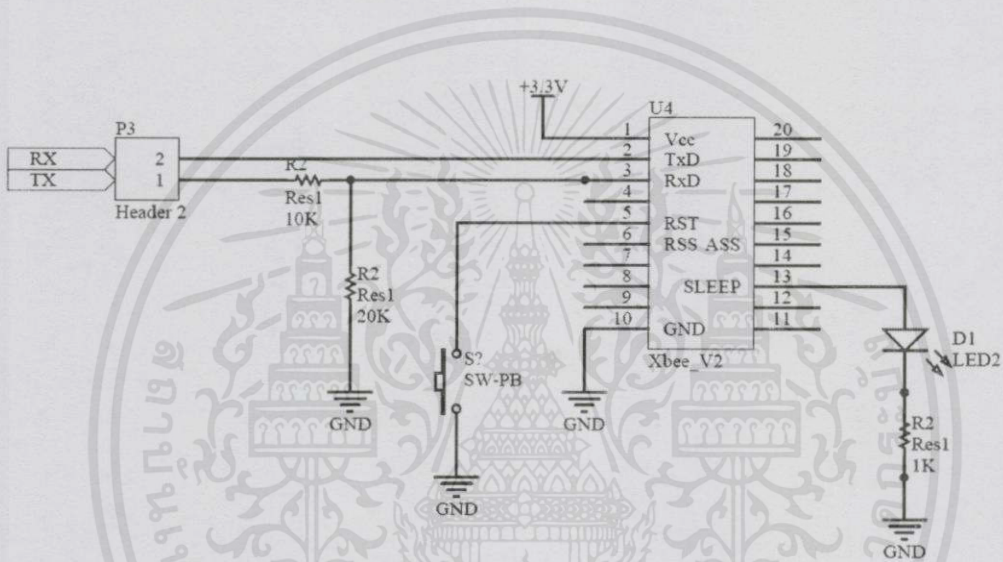
3.2.2 การออกแบบวงจรเชื่อมต่อโมดูล Zigbee ทำหน้าที่เป็นโหนดเราเตอร์

ในส่วนของวงจรเชื่อมต่อโมดูล Zigbee ทำหน้าที่เป็นโหนดเราเตอร์นี้จะทำการเชื่อมต่อกับอุปกรณ์จำลองมิเตอร์ไฟฟ้าซึ่งในที่นี้เราจะใช้ Arduino Uno R3 เป็นไมโครคอนโทรลเลอร์ที่ใช้จำลองการส่งข้อมูลของมิเตอร์ไฟฟ้า อีกทั้งโหนดเราเตอร์จะทำหน้าที่เป็นตัวสร้างโครงข่ายในการติดต่อกับโหนดอุปกรณ์ปลายทาง (มิเตอร์ไฟฟ้า) ตัวอื่นๆที่อยู่ไกลจากโหนดโคออดิเนเตอร์ เพื่อนำข้อมูลจากโหนดอุปกรณ์ปลายทาง (มิเตอร์ไฟฟ้า) ในโครงข่าย ส่งไปแสดงผลที่หน้าจอกอมพิวเตอร์ โดยรายละเอียดส่วนประกอบต่างๆของวงจรแสดงได้ดังนี้



รูปที่ 3.4 วงจรไฟเลี้ยงโมดูล Zigbee ทำหน้าที่เป็นโหนดเราเตอร์

จากรูปที่ 3.4 วงจรไฟเลี้ยงนั้น เราจะใช้หม้อแปลงขนาด 220/6 โวลต์ โดยจะรับแรงดันในระดับ220โวลต์และในฝั่งที่ออกมาจากหม้อแปลง 6 โวลต์จะนำไปเชื่อมต่อกับส่วนแหล่งจ่ายแรงดัน(Supply Voltage)ของวงจร จากนั้นจะผ่าน Bridge Rectifier เพื่อแปลงสัญญาณไฟฟ้ากระแสสลับ(AC)ให้เป็นสัญญาณไฟฟ้ากระแสตรง(DC) และนำไปผ่านตัวเก็บประจุกรองแรงดัน (Capacitor Filter) เพื่อเป็นการลดระลอกคลื่น(ripple) ทำให้แรงดันไฟตรงเรียบขึ้น หลังจากนั้นจะผ่านไอซีลดระดับแรงดัน (Voltage Regulator IC) เบอร์ 7805 เพื่อลดระดับแรงดันให้เหลือ 5 โวลต์ และนำไปใช้เป็นไฟเลี้ยงให้ Arduino Uno R3 และไอซีลดระดับแรงดันเบอร์ 1117T เพื่อลดระดับแรงดันให้เหลือ3.3 โวลต์ และนำไปใช้เป็นไฟเลี้ยงให้กับโมดูลZigbee



รูปที่ 3.5 วงจรการทำงานของโมดูลZigbeeทำหน้าที่เป็นโหนดเราเตอร์

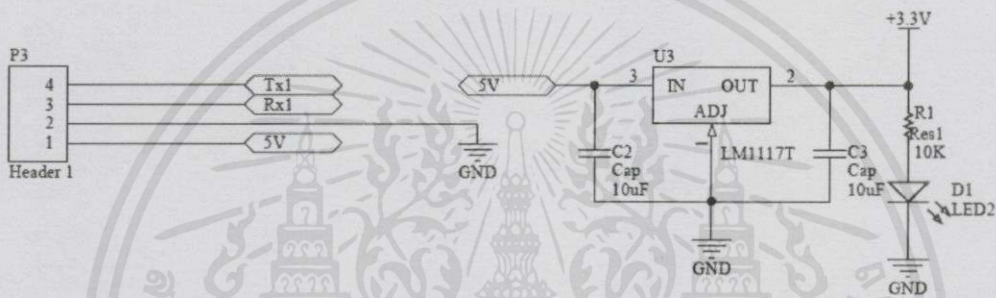
จากรูปที่ 3.5 เพื่อแก้ไขปัญหาข้อมูลที่ส่งแบบ UART ระหว่างอุปกรณ์ที่ใช้ระดับสัญญาณลอจิกต่างกันเกิดความผิดพลาด ทำได้โดยต่อตัวต้านทานขนาด10KΩระหว่างขาRxของโมดูล ZigbeeกับขาTxของ Arduino Uno R3 และต่อตัวต้านทานขนาด20KΩลงกราวด์ระหว่างขาRxของโมดูลZigbeeกับขาTxของ Arduino Uno R3

ในส่วนของโมดูลZigbe่นั้น ไฟเลี้ยงขนาด3.3โวลต์ จะถูกต่อเข้ากับขาที่1 (Vcc) ของโมดูลZigbee นำสวิตช์รีเซตต่อเข้าขาที่ 5 (RST) ไว้เพื่อรีเซตโมดูลZigbee ในกรณีที่มีการทำงานผิดพลาด ส่วนหลอด LED ที่ใช้เพื่อแสดงสถานะON/SLEEP ของโมดูลZigbeeในขาที่13 จะเห็นการทำงานเมื่อ LED ที่ขาที่13ติด (ON) หมายความว่าโมดูลZigbeeอยู่ในสถานะพร้อมที่จะติดต่อสื่อสาร หรือ เมื่อโมดูลZigbeeเชื่อมต่ออยู่แล้วแต่หลอด LED ดับหมายความว่าโมดูลZigbeeอยู่ในสถานะ Sleep Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การออกแบบวงจรเชื่อมต่อโมดูล Zigbee ทำหน้าที่เป็นโหนดโคออดิเนเตอร์

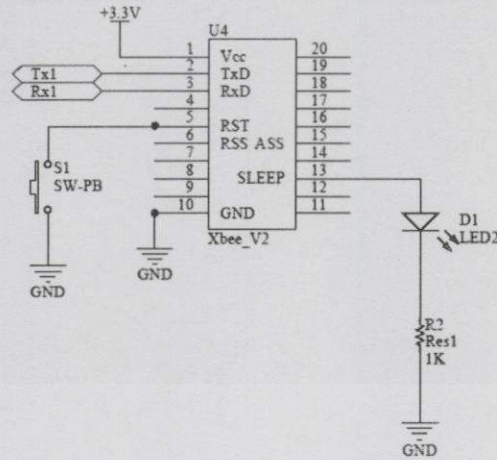
ในส่วนของวงจรเชื่อมต่อโมดูล Zigbee ทำหน้าที่เป็นโหนดโคออดิเนเตอร์ จะเป็นส่วนที่เชื่อมกับศูนย์กลางการควบคุมและแสดงข้อมูลโดยจะทำการติดต่อขอรับข้อมูลการใช้ไฟฟ้าจาก มิเตอร์ไฟฟ้าที่เชื่อมต่ออยู่กับโหนดต่างๆ ในโครงข่ายมาแสดงผลบนหน้าจอคอมพิวเตอร์ผ่านโปรแกรม แสดงข้อมูลการใช้ไฟฟ้า โมดูล Zigbee ที่ทำหน้าที่เป็นโหนดโคออดิเนเตอร์นี้จะเชื่อมต่อกับ คอมพิวเตอร์โดยผ่าน USB to TTL Converter Module ซึ่งเป็นตัวแปลงสัญญาณการเชื่อมต่อแบบ อนุกรม จากโมดูล Zigbee ที่แรงดันระดับ TTL (3.3V) ไปเป็นแรงดัน USB มาตรฐาน (5V) เพื่อเชื่อมต่อ เข้าเครื่องคอมพิวเตอร์โดยตรง โดยรายละเอียดส่วนประกอบต่างๆ ของวงจรแสดงได้ดังนี้



รูปที่ 3.6 วงจรไฟเลี้ยงของโมดูล Zigbee ทำหน้าที่เป็นโหนดโคออดิเนเตอร์

จากรูปที่ 3.6 ส่วนวงจรไฟเลี้ยงของโมดูล Zigbee ทำหน้าที่เป็นโหนดโคออดิเนเตอร์ นี้จะใช้ไฟเลี้ยงจากคอมพิวเตอร์โดยผ่าน USB to TTL Converter Module ซึ่งมีระดับแรงดันเป็น แรงดันไฟฟ้ากระแสตรง (DC) 5 โวลต์ จะถูกแปลงให้เหลือ 3.3 โวลต์โดยผ่านไอซีระดับแรงดัน (Voltage Regulator IC) เบอร์ 1117T เพื่อนำไปใช้เป็นไฟเลี้ยงให้กับโมดูล Zigbee อีกทั้ง USB to TTL Converter Module ยังทำให้สามารถส่งข้อมูล UART ระหว่างคอมพิวเตอร์กับโมดูล Zigbee ได้ โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรการทำงานของโมดูลZigbeeทำหน้าที่เป็นโหนดโคออดิเนเตอร์

จากรูปที่ 3.7 วงจรการทำงานของโมดูลZigbeeenั้นจะใช้ไฟเลี้ยงขนาด3.3โวลต์ต่อเข้ากับขาที่1 (Vcc) ของโมดูลZigbee นำสวิตช์รีเซตต่อเข้ากับขาที่ 5 (RST) ไว้เพื่อรีเซตโมดูลZigbee ในกรณีที่มีการทำงานผิดพลาด ส่วนหลอด LED ที่ใช้เพื่อแสดงสถานะON/SLEEP ของโมดูลZigbeeในขาที่13 จะเห็นการทำงานเมื่อ LED ที่ขาที่13ติด (ON) หมายความว่าโมดูลZigbeeอยู่ในสถานะพร้อมที่จะติดต่อสื่อสาร หรือ เมื่อโมดูลZigbeeเชื่อมต่ออยู่แล้วแต่หลอด LED ดับหมายความว่าโมดูลZigbeeอยู่ในสถานะ Sleep Mode

3.3 หลักการออกแบบโปรแกรม

3.3.1 หลักการออกแบบโปรแกรมฝั่งส่ง

ในส่วนของโปรแกรมฝั่งส่งจะมีการจำลองการส่งข้อมูลของมิเตอร์ไฟฟ้า โดยใช้ไมโครคอนโทรลเลอร์ในการสร้างข้อมูล โดยข้อมูลที่สร้างขึ้นมานั้นจะมีรูปแบบเป็นโปรโตคอลMEATH ซึ่งเป็นรูปแบบโปรโตคอลของมิเตอร์ไฟฟ้าแสดงได้ดังนี้

1. โปรโตคอลMEATH

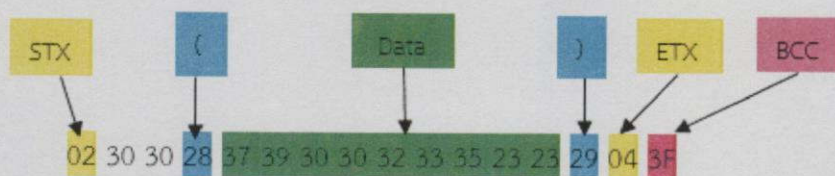
Data message

STX	0	0	(Data)	ETX	BCC
-----	---	---	---	------	---	-----	-----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่สิ่งนี้ให้ผู้อื่นต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 Unit : -

รูปที่ 3.8 วงจรการทำงานของโมดูลZigbeeทำหน้าที่เป็นโหนดโคออดิเนเตอร์

ตัวอย่างข้อมูลฝั่งส่งที่สร้างขึ้นในรูปแบบโปรโตคอลMEATHแสดงได้ดังรูปที่ 3.9

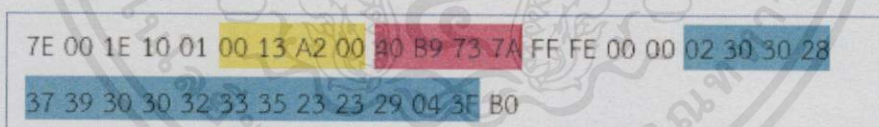


รูปที่ 3.9 ตัวอย่างข้อมูลฝั่งส่งที่สร้างขึ้นในรูปแบบโปรโตคอลMEATH

เมื่อมีการสร้างข้อมูลในฝั่งส่งขึ้นมาแล้วข้อมูลนี้จะถูกดำเนินการกำหนดรูปแบบ Packet ให้สอดคล้องกับโปรโตคอลมาตรฐานของ Zigbee ที่เป็นโหมด API

2. โปรโตคอลมาตรฐานของ Zigbee (โหมด API)

โปรโตคอลมาตรฐานของ Zigbee ที่เป็นโหมด API ในส่วนของฝั่งส่งที่ถูกกำหนดให้เป็นโหนดอุปกรณ์ปลายทาง(มิเตอร์ไฟฟ้าตัวที่1) ได้กำหนดรูปแบบ Packet ไว้ดังตัวอย่าง รูปที่ 3.10 ซึ่งจะเห็นได้ว่าค่า Destination Address หรือค่า SH SL คือ Address ของโมดูลZigbeeที่มีเตอร์ไฟฟ้าตัวที่1ต้องการจะส่งข้อมูลให้ นั่นก็คือโมดูลZigbeeที่เชื่อมต่อกับโหนดโคออดิเนเตอร์หรือศูนย์กลางการควบคุมและแสดงข้อมูลนั่นเอง



รูปที่ 3.10 ตัวอย่างPacketข้อมูลในโหมดAPI ของฝั่งส่ง(มิเตอร์ไฟฟ้าตัวที่1)

รายละเอียดโปรโตคอลของZigbee (โหมดAPI) แสดงได้ดังนี้

- 1) Start Delimiter เป็นส่วนเริ่มต้นของ API จะใช้ 0X7E เป็นตัวค้นเพื่อบอกให้รู้ว่านี่คือจุดเริ่มต้น ของ API มีขนาด 1 Byte
- 2) Length คือ จำนวน Byte ของ Frame Data มีขนาด 2 Byte หาได้จากจำนวน Byte ตั้งแต่ Byte ถัดไปของ Length จนถึง Byte สุดท้ายก่อนถึง Checksum

- 3) Destination Address เป็นค่า SH SL ของโมดูลZigbeeตัวรับ (โหนดโคออดิเนเตอร์) ซึ่งเป็นค่าที่ตั้งมาจากโรงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษายเท่านั้น มิอาจนำเอาไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทางสนธิ หากมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) Frame Data คือ คำสั่งและDataที่ต้องการส่งหรือที่รับมา Frame Data จะมีรายละเอียด แตกต่างกันไป ตาม Frame Type โดยเราจะใช้ ZigBee Transmit Request Type

5) Checksum เป็นส่วนที่มีไว้ตรวจสอบความถูกต้องของ Data ที่ได้รับมาว่าถูกต้องหรือไม่ มีขนาด 1 Byte โดยมีหลักการคำนวณคือ นำค่าแต่ละ Byte ของ Data มาบวกกัน จากนั้นนำผลรวมที่ได้ไป ลบออกจาก 0xFF จากนั้นนำ 1 Byte แรกมาเป็นคำตอบ

ตารางที่ 3.1 รายละเอียดPacket ข้อมูลในโหมดAPI ของอุปกรณ์ปลายทาง

ค่าที่แสดงใน Packet (Byte)	คำอธิบาย
7E	Start Delimiter
00 1E	Length
10 01 00 13 A2 00 40 B9 73 7A FF FE 00 00 02 30 30 28 37 39 30 30 32 33 35 23 23 29 04 3F	Frame Data ของ ZigBee Transmit Request
10	Frame Type เป็น Byte ที่บอกว่า เป็น ZigBee Transmit Request
01	Frame ID เป็นตัวกำหนดว่า การส่ง Data ครั้งนี้ต้องการการตอบกลับจากตัวรับเพื่อยืนยันว่าการส่งสำเร็จหรือไม่ โดย 00 คือ ไม่ต้องการการตอบกลับ และ 01 คือต้องการการตอบกลับ
00 13 A2 00 40 B9 73 7A	64 bit Destination Address หรือค่า SH SL ของตัวรับ (เป็นค่าที่ตั้งมาจากโรงงาน)
FF FE	16-bit Destination Network Address
00	Broadcast Radius ค่าจำนวน Hop สูงสุดซึ่ง 00 คือ set เป็นค่าสูงสุด
00	Options ตั้งค่า option ที่ใช้ในการส่งมีค่าดังนี้ 0x00 - ไม่ใช่ Option 0x01 - Disable ACK 0x20 - Enable APS encryption (if EE=1) 0x40 - Use the extended transmission timeout for this destination
02 30 30 28 37 39 30 30 32 33 35 23 23 29 04 3F	Data ที่ส่งออกไป
B0	ค่า Checksum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานที่กรมกษัตริย์เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโปรโตคอลมาตรฐานของ Zigbee ที่เป็นโหมด API ในส่วนของฝั่งส่งที่ถูกกำหนดให้เป็นมิเตอร์ไฟฟ้าที่เชื่อมต่อกับโหนดเรดเดอร์(มิเตอร์ไฟฟ้าตัวที่2) ได้กำหนดรูปแบบ Packet ไว้ดังตัวอย่าง รูปที่ 3.11 โดยรายละเอียดของPacketข้อมูลอธิบายได้เช่นเดียวกับรายละเอียดPacket ข้อมูลของมิเตอร์ไฟฟ้าตัวที่1

```
7E 00 1E 10 01 00 13 A2 00 40 B9 73 7A FF FE 00 00 02 30 30 28
37 39 30 30 32 33 36 23 23 29 04 3C B2
```

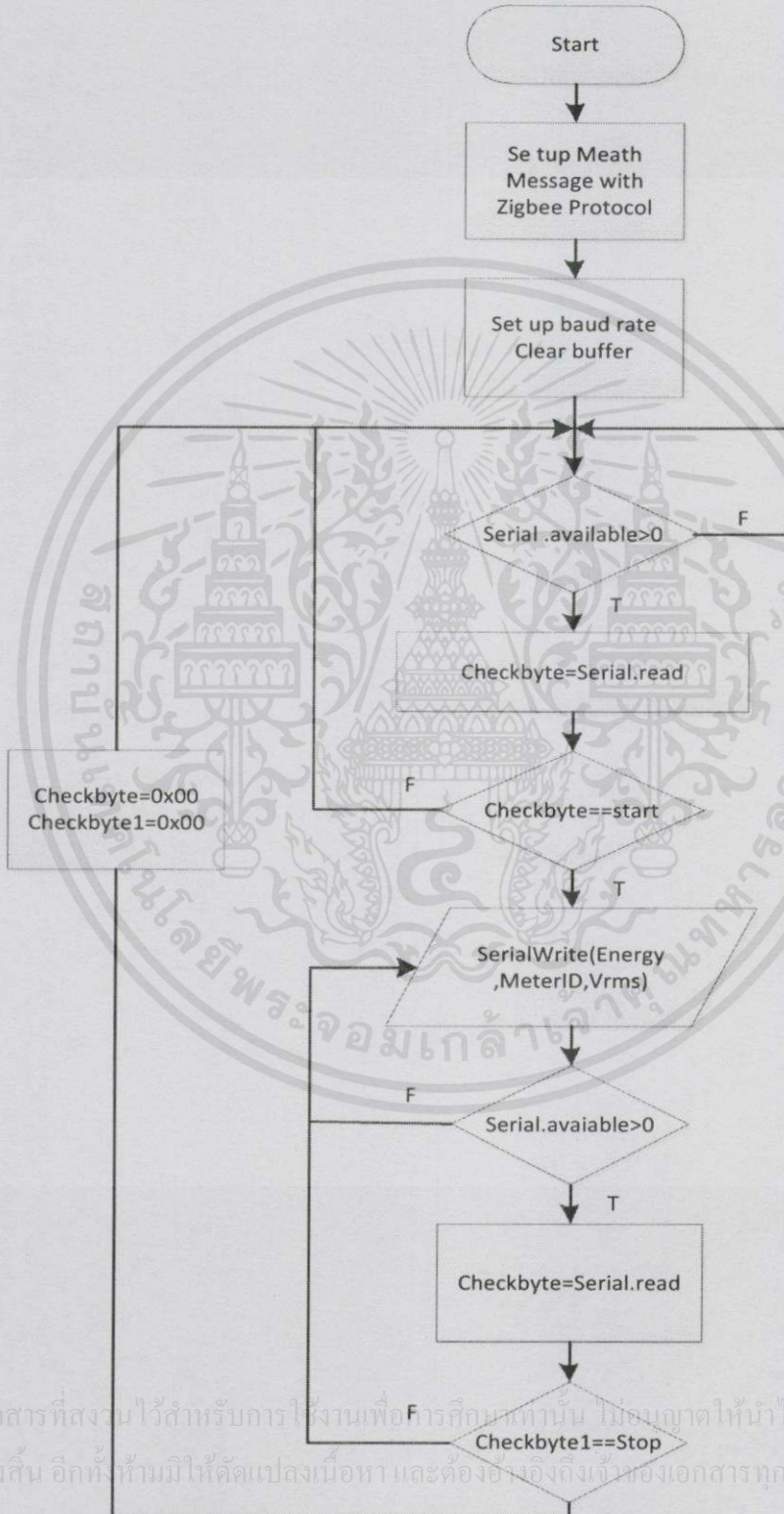
รูปที่ 3.11 ตัวอย่างPacketข้อมูลในโหมดAPI ของฝั่งส่ง(มิเตอร์ไฟฟ้าตัวที่2)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การออกแบบโปรแกรมจำลองการสื่อสารข้อมูลของมิเตอร์ไฟฟ้า (ฝั่งส่ง)

ในการออกแบบโปรแกรมจำลองการสื่อสารข้อมูลของมิเตอร์ไฟฟ้านั้น มิเตอร์ไฟฟ้า จะได้รับการติดต่อจากฝั่งรับให้ส่งข้อมูลการใช้ไฟฟ้าไปให้กับฝั่งรับในรูปแบบ Packet ของโหมด API



รูปที่ 3.12 Flowchart แสดงการทำงานของโปรแกรมจำลองการสื่อสารข้อมูลของมิเตอร์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 หลักการออกแบบโปรแกรมฝั่งรับ

ในส่วนโปรแกรมฝั่งรับนั้น จะทำการออกแบบโปรแกรมให้สามารถกำหนดทิศทางในการขอรับข้อมูลการใช้ไฟฟ้าจากมิเตอร์ไฟฟ้าที่อยู่ในโครงข่ายได้ โดยให้ฝั่งรับ(โหนดโคออดิเนเตอร์) ขอรับข้อมูลการใช้ไฟฟ้าจากมิเตอร์ไฟฟ้าด้วยการกำหนดค่า Destination Address หรือค่า SH SL ของมิเตอร์ไฟฟ้าตัวที่ต้องการข้อมูล

โปรโตคอลมาตรฐานของ Zigbee ที่เป็นโหมด API ในส่วนของฝั่งรับ เมื่อได้รับข้อมูลมาจากมิเตอร์ไฟฟ้าตัวที่1 ได้กำหนดรูปแบบ Packet ไว้ดังตัวอย่าง รูปที่ 3.13 ซึ่งอธิบายได้เช่นเดียวกันกับรายละเอียด Packet ข้อมูลของฝั่งส่ง

```
7E 00 1C 90 00 13 A2 00 40 CA A8 91 6A 0A 41 02 30 30 28 37 39
30 30 32 33 35 23 23 29 04 3F 1C
```

รูปที่ 3.13 Packet ข้อมูลที่ได้รับจากมิเตอร์ไฟฟ้าตัวที่1ในโหมดAPI ของฝั่งรับ

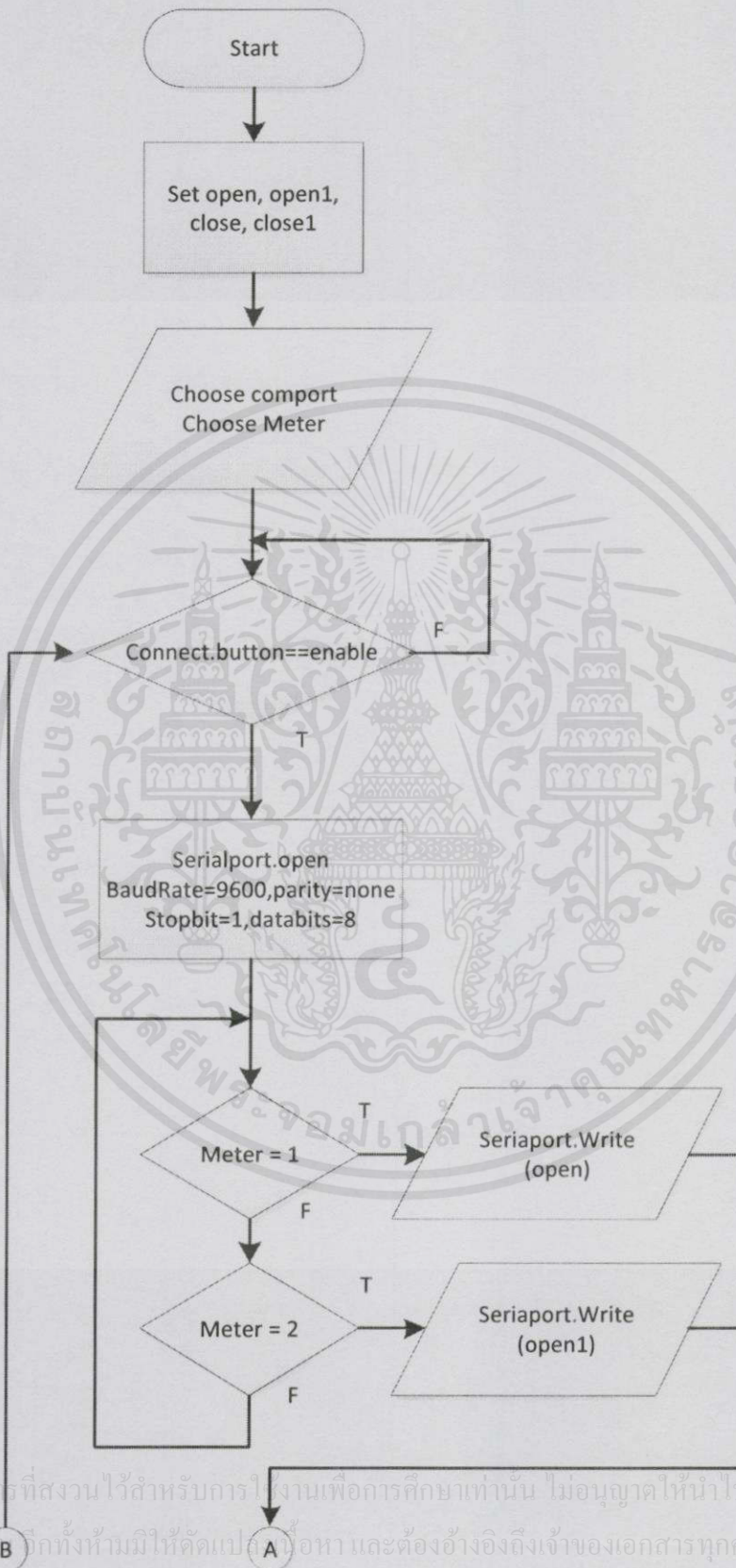
โปรโตคอลมาตรฐานของ Zigbee ที่เป็นโหมด API ในส่วนของฝั่งรับ เมื่อได้รับข้อมูลมาจากมิเตอร์ไฟฟ้าตัวที่2 ได้กำหนดรูปแบบ Packet ไว้ดังตัวอย่าง รูปที่ 3.14 ซึ่งอธิบายได้เช่นเดียวกันกับรายละเอียด Packet ข้อมูลของฝั่งส่ง

```
7E 00 1C 90 00 13 A2 00 40 CA A8 91 6A 0A 41 02 30 30 28 37 39
30 30 32 33 35 23 23 29 04 3F 1C
```

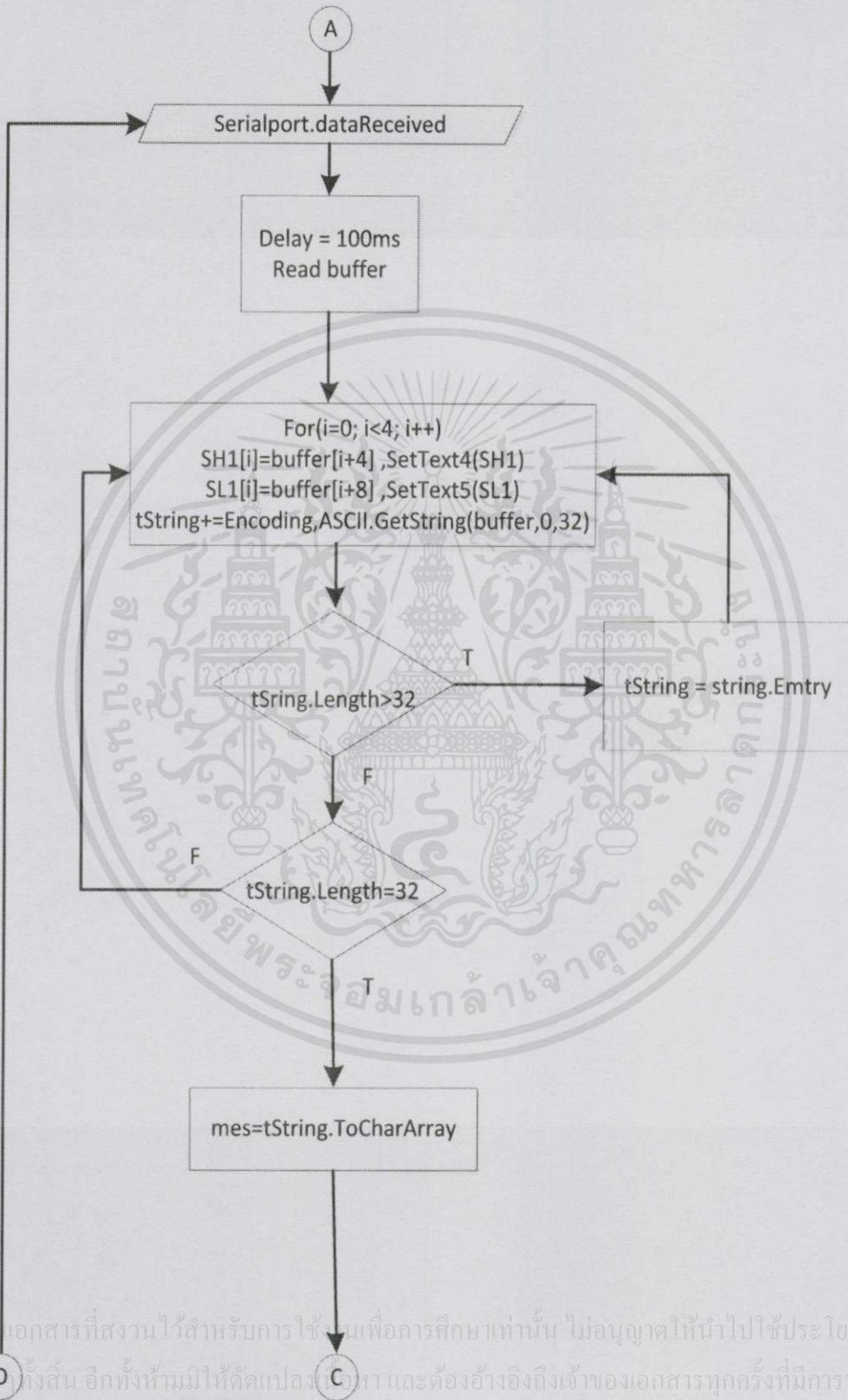
รูปที่ 3.14 Packet ข้อมูลที่ได้รับจากมิเตอร์ไฟฟ้าตัวที่2ในโหมดAPI ของฝั่งรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

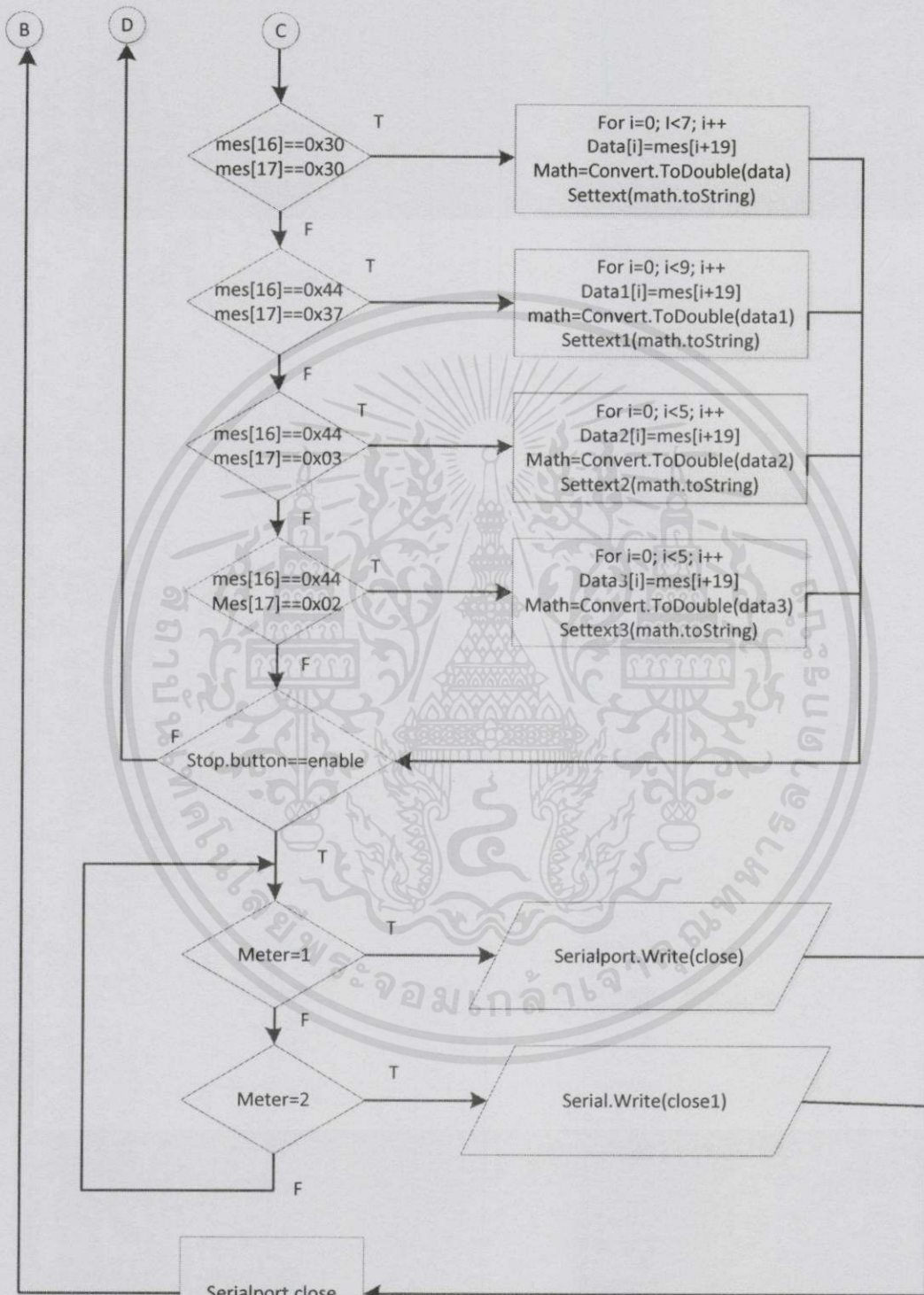
1. การออกแบบโปรแกรมฝั่งรับ (ศูนย์กลางการควบคุมและแสดงข้อมูล)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งนี้ (B) ก็กัทั้งห้ามมิให้คัดแปล (A) เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณี (D) ใดทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลง (C) ใดๆ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบที่ 3.15 Flowchart แสดงการทำงานของโปรแกรมฝั่งรับ (ศูนย์กลางการควบคุมและแสดงข้อมูล) ใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองเกี่ยวกับปัจจัยที่มีผลต่อการส่งข้อมูล

เมื่อสามารถรับ-ส่งข้อมูลไร้สายได้แล้ว ในการใช้งานจริงจะมีปัจจัยสำคัญที่ทำให้การส่งข้อมูลเกิดความผิดพลาด นั่นคือระยะในการส่งหรือสิ่งกีดขวางที่เกิดขึ้นในระหว่างการส่ง ซึ่งตามความเป็นจริงมีเตอร์ไฟฟ้าที่ถูกติดตั้งตามสถานที่ต่างๆ เช่น อพาร์ทเมนต์ ห้างสรรพสินค้า ฟาร์มเกษตร หรือสถานที่ใดๆ นั้น ตัวรับและตัวส่งไม่ได้ส่งข้อมูลเป็นแบบเส้นตรง แต่อาจจะอยู่ในสถานที่ที่ทับหรือมีสิ่งกีดขวางอย่างเช่น เครื่องจักร วัตถุ ที่สัญญาณไม่สามารถทะลุผ่านได้

4.1.1 การทดลองระยะทางการส่งข้อมูลระหว่างโมดูล Zigbee 2 ตัว

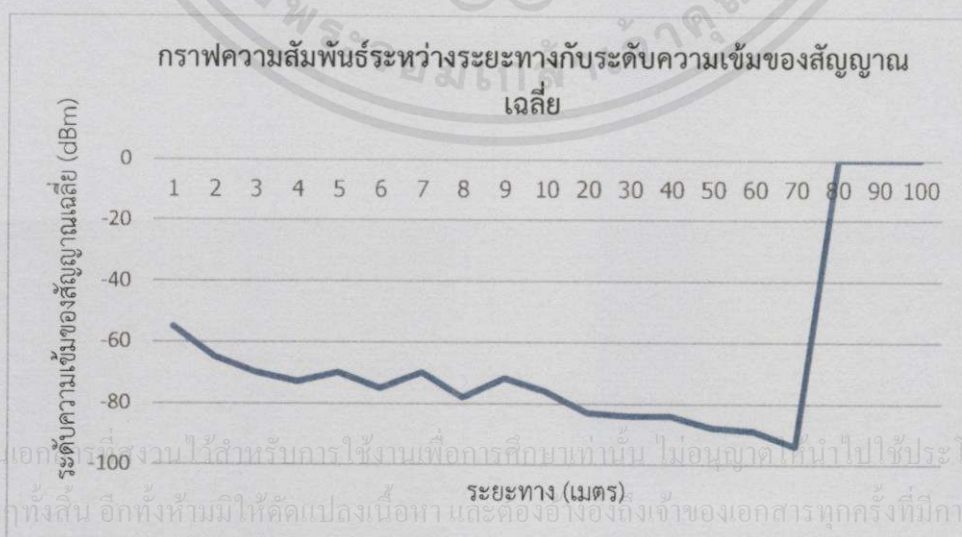
ในการทดลองนี้จะทำการทดลองวัดระยะทางที่สามารถส่งข้อมูลได้ โดยวางโมดูล Zigbee 2 ตัวให้มีค่าระยะห่างที่ต่าง ๆ กันโดยไม่ผ่านสิ่งกีดขวาง ตามคุณสมบัติของ Xbee Series 2 สามารถส่งข้อมูลในร่มได้สูงสุดที่ระยะ 40 เมตร และสามารถส่งข้อมูลกลางแจ้งได้สูงสุดที่ระยะ 120 เมตร

สำหรับการทดลองนี้จะทำการทดสอบประสิทธิภาพในการรับ-ส่งข้อมูลโดยใช้โปรแกรมในส่วนการทดสอบระยะทางของ Zigbee ในโปรแกรม X-CTU โดยเมื่อทำการรับ-ส่งข้อมูลระหว่าง Zigbee ทั้ง 2 ตัวที่ระยะทางต่างๆ กันจะได้ผลการทดลองดังตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
"ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้"

ตารางที่ 4.1 ผลการวัดประสิทธิภาพการรับส่งข้อมูลของโมดูลZigbeeจากระยะทาง โดยไม่ผ่านสิ่งกีดขวาง

ระยะทาง (เมตร)	ระดับความเข้มของสัญญาณเฉลี่ย (dBm)	ความสามารถในการรับส่งข้อมูล
1	-55	สามารถรับส่งได้ปกติ
2	-65	สามารถรับส่งได้ปกติ
3	-70	สามารถรับส่งได้ปกติ
4	-73	สามารถรับส่งได้ปกติ
5	-70	สามารถรับส่งได้ปกติ
6	-75	สามารถรับส่งได้ปกติ
7	-70	สามารถรับส่งได้ปกติ
8	-78	สามารถรับส่งได้ปกติ
9	-72	สามารถรับส่งได้ปกติ
10	-76	สามารถรับส่งได้ปกติ
20	-83	สามารถรับส่งได้ปกติ
30	-84	สามารถรับส่งได้ปกติ
40	-84	สามารถรับส่งได้ปกติ
50	-88	ข้อมูลที่ส่งมาเริ่มช้า
60	-89	ข้อมูลส่งมาได้ช้าและขาดการส่งข้อมูลเป็นระยะ
70	-94	ข้อมูลส่งมาได้น้อยครั้งและขาดการติดต่อเป็นเวลานาน
80	0	ไม่สามารถรับส่งได้
90	0	ไม่สามารถรับส่งได้
100	0	ไม่สามารถรับส่งได้



รูปที่ 4.1 กราฟความสัมพันธ์ระหว่างระยะทางกับระดับความเข้มของสัญญาณเฉลี่ย

จากการทดลองพบว่าในการรับ-ส่งข้อมูลระหว่างZigbeeตามระยะทางที่ไกลขึ้นจะทำให้ค่าระดับความเข้มของสัญญาณเฉลี่ย(dBm)มีค่าติดลบเพิ่มมากขึ้น ข้อมูลส่งได้ช้า ขาดการส่งข้อมูลเป็นระยะหรือขาดการติดต่อเป็นเวลานาน จนกระทั่งZigbeeไม่สามารถรับ-ส่งข้อมูลถึงกันได้จึงมีค่าระดับความเข้มของสัญญาณเฉลี่ย(dBm)เป็น0

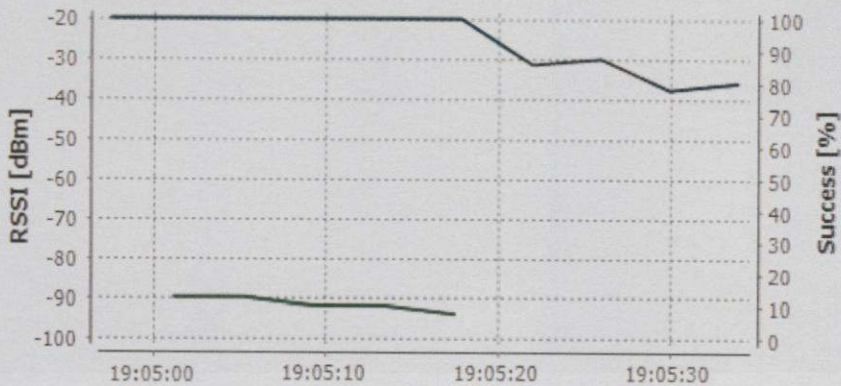
เมื่อนำผลการทดสอบมาวิเคราะห์ก็พบว่าจากระยะที่สามารถรับ-ส่งข้อมูลได้ปกติ เช่นที่ระยะทาง 30 เมตร จะมีกราฟระดับความเข้มของสัญญาณและเปอร์เซ็นต์ความถูกต้องในการรับ-ส่งข้อมูลดังรูปที่ 4.2 โดยจากกราฟเส้นสีน้ำเงินจะเป็นกราฟที่บ่งบอกถึงเปอร์เซ็นต์ความถูกต้องในการรับ-ส่งข้อมูล จะเห็นได้ว่าความถูกต้องในการรับส่งข้อมูลจะเป็น100เปอร์เซ็นต์ตลอดระยะเวลาการรับ-ส่งข้อมูล ส่วนระดับความเข้มของสัญญาณจะบ่งบอกโดยกราฟเส้นสีเขียวซึ่งมีค่าประมาณ -84 dBm



รูปที่ 4.2 กราฟระดับความเข้มของสัญญาณและเปอร์เซ็นต์ความถูกต้องในการรับ-ส่งข้อมูลที่ระยะ30 เมตร

ระยะที่เริ่มมีความผิดพลาดในการรับ-ส่งข้อมูลและรับข้อมูลได้ช้าลง=55 เมตร ระยะสูงสุดที่สามารถรับ-ส่งข้อมูลได้จริง=73 เมตร จะมีกราฟระดับความเข้มของสัญญาณและเปอร์เซ็นต์ความถูกต้องในการรับ-ส่งข้อมูลดังรูปที่ 4.3 โดยจากกราฟเส้นสีน้ำเงินจะเป็นกราฟที่บ่งบอกถึงเปอร์เซ็นต์ความถูกต้องในการรับ-ส่งข้อมูล จะเห็นได้ว่าความถูกต้องในการรับส่งจะลดลงจาก100เปอร์เซ็นต์เหลือ80เปอร์เซ็นต์ในระหว่างระยะเวลาการรับ-ส่งข้อมูล ส่วนระดับความเข้มของสัญญาณจะบ่งบอกโดยกราฟเส้นสีเขียวซึ่งมีค่าประมาณ -94 dBm ในการทดสอบนี้ผลที่ได้ขึ้นอยู่กับสภาพแวดล้อมของแต่ละพื้นที่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 กราฟระดับความเข้มของสัญญาณและเปอร์เซ็นต์ความถูกต้องในการรับ-ส่งข้อมูลที่ระยะ 73 เมตร

4.1.2 การทดลองระยะทางการส่งข้อมูลเป็นเครือข่ายโดยใช้โหมดAPI

ในการทดลองนี้เราจะใช้สถานที่ในการทดลองเสมือนกับการนำไปใช้งานจริงโดยให้ ส่วนของผู้ใช้งานหรือศูนย์กลางการควบคุมและแสดงข้อมูลอยู่ในห้องปิดทึบ ส่วนโหนดเราเตอร์และ ส่วนของการจำลองการส่งข้อมูลของมิเตอร์ไฟฟ้า (โหนดอุปกรณ์ปลายทาง) จะถูกติดตั้งไว้ด้านนอกห้องปิดทึบตามลำดับ โดยให้โหนดเราเตอร์และส่วนของการจำลองการส่งข้อมูลของมิเตอร์ไฟฟ้า (โหนดอุปกรณ์ปลายทาง) มีระยะห่างต่างกันไป

ในขั้นแรกจะทำการทดสอบการรับ-ส่งข้อมูลของโมดูล Zigbee 2 ตัวก่อนโดยที่โมดูล Zigbee ส่วนผู้ใช้งานจะอยู่ในห้องปิดทึบและโมดูล Zigbee อีกตัวหนึ่งจะอยู่ด้านนอกห้อง พบว่าโมดูล Zigbee สามารถทำการรับส่งข้อมูลกันได้จนถึงระยะ 11.6 เมตร ดังนั้นจึงนำโหนดเราเตอร์ไปติดตั้งไว้ที่ระยะห่างจากส่วนของผู้ใช้งาน 11.6 เมตรแทน จากนั้นจึงทำการทดสอบโดยการเปลี่ยนระยะทางระหว่างโหนดเราเตอร์กับโหนดอุปกรณ์ปลายทางไปเรื่อยๆ ได้ผลดังตารางที่ 4.2

ตารางที่ 4.2 ผลการวัดประสิทธิภาพการรับส่งข้อมูลของโมดูล Zigbee จากระยะทางแบบเป็น
เครือข่าย

ระยะทางระหว่าง ผู้ใช้งานกับโหนดเรา เตอร์ (เมตร)	ระยะทางระหว่าง โหนดเราเตอร์กับ โหนดอุปกรณ์ ปลายทาง (เมตร)	ระยะทางรวม (เมตร)	ความสามารถในการรับส่ง ข้อมูล
11.6	1	12.6	สามารถรับส่งได้ปกติ
11.6	2	13.6	สามารถรับส่งได้ปกติ
11.6	3	14.6	สามารถรับส่งได้ปกติ
11.6	4	15.6	สามารถรับส่งได้ปกติ
11.6	5	16.6	สามารถรับส่งได้ปกติ
11.6	6	17.6	สามารถรับส่งได้ปกติ
11.6	7	18.6	สามารถรับส่งได้ปกติ
11.6	8	19.6	สามารถรับส่งได้ปกติ
11.6	9	20.6	สามารถรับส่งได้ปกติ
11.6	10	21.6	สามารถรับส่งได้ปกติ
11.6	20	31.6	สามารถรับส่งได้ปกติ
11.6	30	41.6	สามารถรับส่งได้ปกติ
11.6	40	51.6	สามารถรับส่งได้ปกติ
11.6	50	61.6	ข้อมูลส่งมาได้ช้าและขาด การส่งข้อมูลเป็นระยะ
11.6	60	71.6	ข้อมูลส่งมาได้ช้าและขาด การส่งข้อมูลเป็นระยะ
11.6	70	81.6	ไม่สามารถรับส่งได้
11.6	80	91.6	ไม่สามารถรับส่งได้
11.6	90	101.6	ไม่สามารถรับส่งได้
11.6	100	111.6	ไม่สามารถรับส่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.2 ผลการวัดประสิทธิภาพการรับส่งข้อมูลของโมดูลZigbeeจากระยะทางแบบเป็นเครือข่ายพบว่า

ระยะทางรวมที่การรับ-ส่งข้อมูลเริ่มมีความผิดพลาดและรับข้อมูลได้ช้าลง =61.6 เมตร

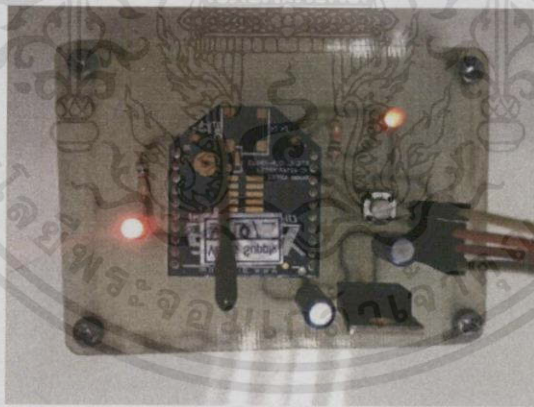
ระยะทางรวมสูงสุดที่สามารถรับ-ส่งข้อมูลได้จริง =81.6 เมตร

ในการทดสอบนี้ผลที่ได้ขึ้นอยู่กับสภาพแวดล้อมของแต่ละพื้นที่ด้วย

4.2 การทดลองการจัดการเครือข่ายการติดต่อสื่อสารของโมดูลZigbeeโดยใช้โหมดAPI

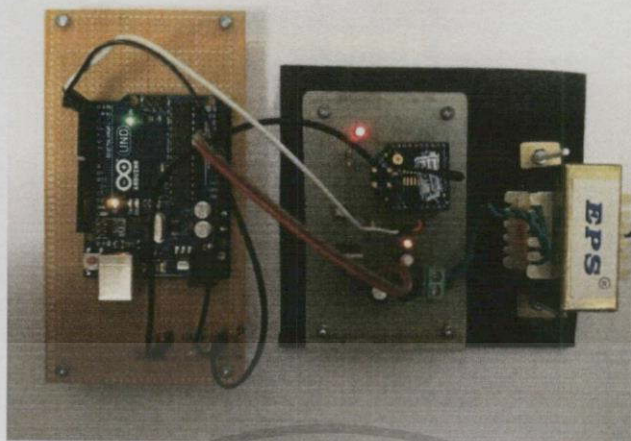
4.2.1 การทดลองเชื่อมต่อเครือข่ายของโมดูลZigbee แบบ Cluster Tree (Tree)

ในการทดลองเชื่อมต่อเครือข่ายของโมดูลZigbee แบบ Cluster Tree (Tree) ได้ทำการทดสอบการส่งข้อมูลไร้สายผ่านโมดูลZigbeeจากโหนดอุปกรณ์ปลายทาง(มิเตอร์1) ผ่านโมดูลZigbeeที่ทำหน้าที่เป็นโหนดเราเตอร์(มิเตอร์2) เพื่อเป็นการขยายระยะทางและสร้างเครือข่ายให้กับระบบการทำงาน ไปยังโมดูลZigbeeส่วนควบคุมและแสดงผล(โหนดโคออดิเนเตอร์) เพื่อแสดงผลผ่านโปรแกรมแสดงข้อมูลการใช้ไฟฟ้าในคอมพิวเตอร์ในโหมดAPI โดยอุปกรณ์ที่ใช้ในการทำการทดลองมีดังนี้

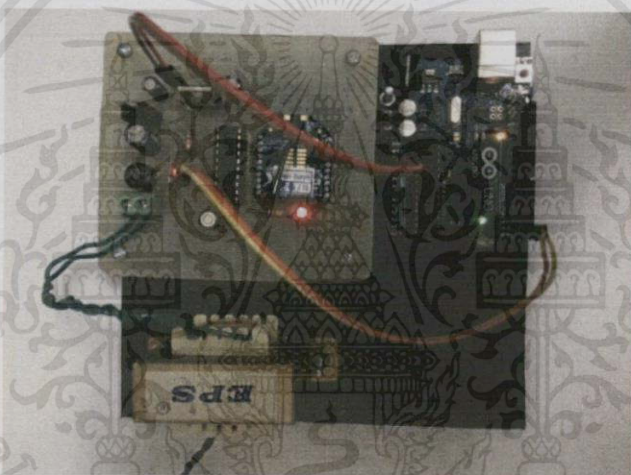


รูปที่ 4.4 โมดูลZigbeeที่ทำหน้าที่เป็นโหนดโคออดิเนเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 โมดูล Zigbee ที่ทำหน้าที่เป็น โหนดเราเตอร์



รูปที่ 4.6 โมดูล Zigbee ที่ทำหน้าที่เป็น โหนดอุปกรณ์ปลายทาง

ในการส่งข้อมูลในรูปแบบของเครือข่ายจากมิเตอร์ไฟฟ้าไปยังผู้ใช้งานโดยใช้โหมด API นั้น โมดูล Zigbee ในส่วนจำลองการส่งข้อมูลของมิเตอร์ไฟฟ้าจะมี Function set เป็น Router/End Device API ซึ่งจะทำหน้าที่ส่งข้อมูลให้กับโหนดเราเตอร์ โดยโมดูล Zigbee ที่โหนดเราเตอร์จะมี Function set เป็น Router/End Device API ซึ่งจะทำหน้าที่ส่งข้อมูลต่อไปให้ยังส่วนของศูนย์ควบคุมและแสดงผล (โหนดโคออดิเนเตอร์) โมดูลในส่วนนี้จะมี Function set เป็น Coordinator API

ในการทดลองจะนำโหนดอุปกรณ์ปลายทาง (มิเตอร์ 1) และโหนดเราเตอร์ (มิเตอร์ 2) วางห่างจากโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) 40 เมตร จากการทดลองพบว่า โหนดอุปกรณ์ปลายทาง (มิเตอร์ 1) และโหนดเราเตอร์ (มิเตอร์ 2) สามารถส่งข้อมูลไปให้กับโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) ได้ทันทีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการเพิ่มระยะทางของโหนดอุปกรณ์ปลายทาง (มิเตอร์1) ออกไปเป็น 80 เมตร โดยที่ โหนดเราเตอร์ (มิเตอร์2) วางห่างจากโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) 40 เมตร เท่าเดิม พบว่าโหนดอุปกรณ์ปลายทาง (มิเตอร์1) สามารถส่งข้อมูลผ่านทางโหนดเราเตอร์ (มิเตอร์2) ไปยังโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) ได้

จึงทำการทดลองต่อโดยการส่งข้อมูลจากโหนดอุปกรณ์ปลายทาง (มิเตอร์1) ผ่าน โหนดเราเตอร์ (มิเตอร์2) ส่งข้อมูลต่อไปยังโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) ซึ่งพบว่าสามารถส่งข้อมูลได้

จากนั้นทำการวางโหนดเราเตอร์ (มิเตอร์2) ให้ไกลออกไปเป็น 80 เมตร พบว่าไม่สามารถส่งข้อมูลมายังโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) ได้

4.2.2 การทดสอบเชื่อมต่อเครือข่ายของโมดูลZigbee แบบ Star

ทำการตั้งค่าโมดูลZigbee ให้มี Function set เป็น Coordinator และเชื่อมต่อเข้ากับคอมพิวเตอร์ และทำการตั้งค่าโมดูลZigbeeที่เชื่อมต่อกับมิเตอร์1และมิเตอร์2 ให้มี Function set เป็น End Device

ในการทดลองจะนำโหนดอุปกรณ์ปลายทาง (มิเตอร์1) และโหนดอุปกรณ์ปลายทาง (มิเตอร์2) วางห่างจากโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) 40 เมตร จากการทดลองพบว่า โหนดอุปกรณ์ปลายทาง (มิเตอร์1) และโหนดอุปกรณ์ปลายทาง (มิเตอร์2) สามารถส่งข้อมูลไปให้กับโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) ได้

เมื่อทำการเพิ่มระยะทางของโหนดอุปกรณ์ปลายทาง (มิเตอร์1) ออกไปเป็น 80 เมตร โดยที่ โหนดอุปกรณ์ปลายทาง (มิเตอร์2) วางห่างจากโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) 40 เมตร เท่าเดิม พบว่าโหนดอุปกรณ์ปลายทาง (มิเตอร์1) ไม่สามารถส่งข้อมูลไปหาโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) ได้เนื่องจากมีระยะทางที่ไกล

4.2.3 การทดสอบเชื่อมต่อเครือข่ายของโมดูลZigbee แบบ Mesh

ทำการตั้งค่าโมดูลZigbee ให้มี Function set เป็น Coordinator และเชื่อมต่อเข้ากับคอมพิวเตอร์ และทำการตั้งค่าโมดูลZigbeeที่เชื่อมต่อกับมิเตอร์1และมิเตอร์2 ให้มี Function set เป็น Router

ในการทดลองจะนำโหนดเราเตอร์ (มิเตอร์1) และโหนดเราเตอร์ (มิเตอร์2) วางห่างจากโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) 40 เมตร จากการทดลองพบว่า โหนดเราเตอร์ (มิเตอร์1) และโหนดเราเตอร์ (มิเตอร์2) สามารถส่งข้อมูลไปให้กับโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) ได้

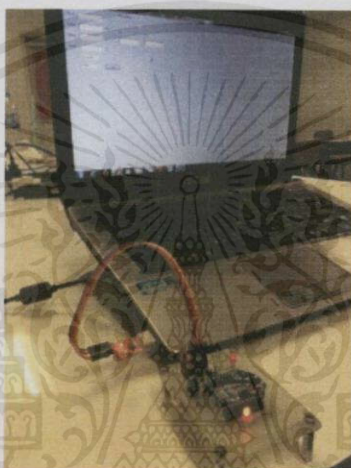
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เมื่อทำการเพิ่มระยะทางของโหนดเราเตอร์ (มิเตอร์1) ออกไปเป็น 80 เมตร โดยที่ โหนดเราเตอร์ (มิเตอร์2) วางห่างจากโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน) 40 เมตร เท่าเดิม พบว่า

โหนดเราเตอร์ (มิเตอร์1) สามารถส่งข้อมูลผ่านทางโหนดเราเตอร์(มิเตอร์2) ไปยังโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน)ได้

จากนั้นทำการวางโหนดเราเตอร์ (มิเตอร์2)ให้ไกลออกไปเป็น 80 เมตร พบว่าไม่สามารถส่งข้อมูลมายังโหนดโคออดิเนเตอร์ (ส่วนของผู้ใช้งาน)ได้ เนื่องจากโหนดเราเตอร์ (มิเตอร์1) และโหนดเราเตอร์ (มิเตอร์2) มีระยะทางที่ไกลจากโหนดโคออดิเนเตอร์มาก

4.2.4 การทดสอบการใช้โปรแกรมแสดงข้อมูลของมิเตอร์ไฟฟ้าบนหน้าจอคอมพิวเตอร์



รูปที่ 4.7 การเชื่อมต่อโมดูลZigbeeในส่วนควบคุมและแสดงผล(โหนดโคออดิเนเตอร์)

1) ในขั้นตอนแรกของระบบการทำงานในส่วนของศูนย์กลางการควบคุมและแสดงผลข้อมูล(โหนดโคออดิเนเตอร์)นั้นจะต้องมีการเชื่อมต่อโมดูลZigbeeเข้ากับคอมพิวเตอร์ดัง รูปที่ 4.7

2) ทำการเลือกพอร์ตที่เชื่อมต่อกับโมดูลZigbeeให้ถูกต้อง

3) ในส่วนของ Choose meter เมื่อทำการเลือกmeter1ซึ่งก็คือมิเตอร์ไฟฟ้าที่เป็นโหนดอุปกรณ์ปลายทางและกดปุ่มconnect หน้าต่างของโปรแกรมจะแสดงข้อมูลการใช้ไฟฟ้าของmeter1ซึ่งแสดงได้ดังรูปที่ 4.8 โดยหน้าต่างโปรแกรมของmeter1จะแสดงข้อมูลดังนี้

Meter ID, Energy (KWh), Voltage (V), Current (A) และDestination Addressหรือค่าSH SL ของmeter1(โหนดอุปกรณ์ปลายทาง)

4) หากไม่ต้องการให้มิเตอร์ไฟฟ้าส่งข้อมูลมาอีกแล้วให้กดปุ่มstop

เมื่อต้องการทราบข้อมูลของ meter2 (มิเตอร์ไฟฟ้าที่เชื่อมต่อกับโหนดเราเตอร์) ก็

จะมีการใช้งานของโปรแกรมเช่นเดียวกับmeter1 โดยหน้าต่างของโปรแกรมจะแสดงข้อมูลการใช้ไฟฟ้าของmeter2 ซึ่งแสดงได้ดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.8 โปรแกรมแสดงข้อมูลการใช้ไฟฟ้าของmeter1(โหนดอุปกรณ์ปลายทาง)

รูปที่ 4.9 โปรแกรมแสดงข้อมูลการใช้ไฟฟ้าของmeter2 (โหนดเรอเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

โครงการงานเรื่องการพัฒนาโมดูล Zigbee สำหรับมิเตอร์อัจฉริยะ จากการทดลองในแต่ละส่วนของโครงการสามารถสรุปผลได้ดังนี้

5.1.1 ส่วนโหนดอุปกรณ์ปลายทาง

สามารถจำลองโปรโตคอลในการสื่อสารของมิเตอร์อัจฉริยะและสามารถควบคุมโหนดอุปกรณ์ปลายทางให้ส่งข้อมูลมาให้กับส่วนควบคุมและแสดงผลได้โดยใช้โหมดAPIเป็นตัวกำหนดทิศทางในการติดต่อสื่อสาร

5.1.2 ส่วนโหนดเราเตอร์

เป็นตัวกลางในการส่งผ่านข้อมูลจากโหนดอุปกรณ์ปลายทางไปยังส่วนควบคุมและแสดงผล (โหนดโคออดิเนเตอร์) ได้ โดยเป็นการเพิ่มระยะทางและสร้างเครือข่ายการติดต่อสื่อสารกันระหว่างโหนดอุปกรณ์ปลายทางกับส่วนควบคุมและแสดงผลที่อยู่ไกลกันมากๆ

5.1.3 ส่วนควบคุมและแสดงผล (โหนดโคออดิเนเตอร์)

สามารถควบคุมการส่งข้อมูลของโหนดอุปกรณ์ปลายทางได้โดยใช้โหมดAPIเป็นตัวกำหนดทิศทางในการติดต่อสื่อสาร สามารถรับข้อมูล Packet ที่ส่งมาจากอุปกรณ์ปลายทางมาทำการกรองข้อมูลก่อนที่จะแสดงผลผ่านโปรแกรมแสดงการใช้ไฟฟ้าบนหน้าจอกอมพิวเตอร์

จากการทดลองพบว่า สามารถนำโมดูล Zigbee มาสร้างเครือข่ายได้ 3 รูปแบบ คือ การเชื่อมต่อเครือข่ายแบบ Star, Cluster tree (tree) และ Mesh ซึ่งจากการทดลองพบว่า การสร้างเครือข่ายแบบ Mesh เป็นการสร้างเครือข่ายที่มีประสิทธิภาพสูงเนื่องจากสามารถส่งข้อมูลไปถึงเป้าหมายได้หลายทิศทาง

5.2 วิจารณ์ผลการทดลอง

ในการทดลองนี้พบว่าระยะทางและสิ่งกีดขวางในการติดต่อสื่อสารข้อมูลระหว่างโมดูล Zigbee อาจจะทำให้การติดต่อสื่อสารข้อมูลระหว่างกันเกิดความล่าช้าหรือเกิดความผิดพลาดขึ้นได้

5.3 ปัญหาที่พบในระหว่างดำเนินโครงการ

เอกสารนี้เป็นเอกสารที่ 5.3.1 เนื่องจากการทำงานนี้ต้องใช้ความรู้ในหลายๆด้านทั้งในส่วนของการคำนวณฮาร์ดแวร์และซอฟต์แวร์ รวมถึงเทคโนโลยีอื่นๆที่ใช้ในโครงการนี้ ผู้ทำโครงการมีความรู้ไม่เพียงพอจึงทำให้เสียเวลามากในการศึกษาและค้นคว้าข้อมูล

5.3.2 ปัญหาความไม่คงทน และการชำรุดได้ง่ายของอุปกรณ์อิเล็กทรอนิกส์เมื่อทำการทดลองมากๆ จึงทำให้เสียเวลาซ่อมแซมและจัดหาอุปกรณ์ใหม่

5.3.3 อุปกรณ์ที่ใช้ในการทำโครงการบางอย่างมีราคาแพงมาก เช่น Xbee

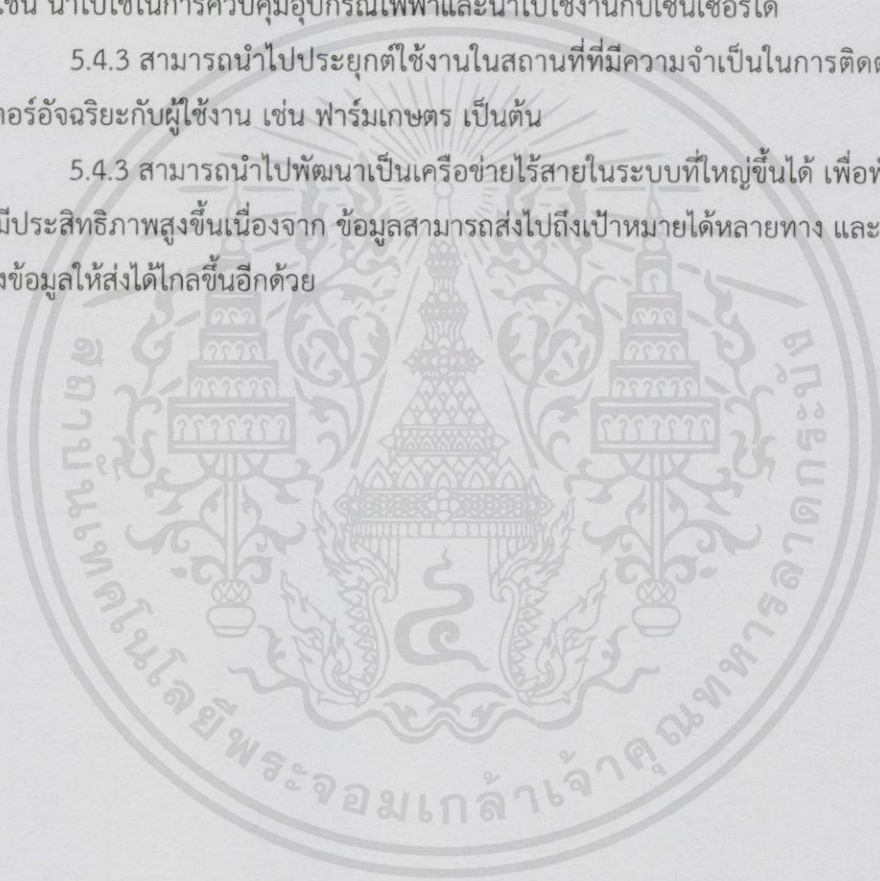
5.4 แนวทางการแก้ไขปัญหาและพัฒนา

5.4.1 โครงการนี้สามารถนำโมดูล Zigbee ไปปรับปรุงให้สามารถรองรับการใช้งานในการรับข้อมูลกับมิเตอร์อัจฉริยะที่ใช้งานจริงได้

5.4.2 นอกจากนี้สามารถนำโมดูล Zigbee ไปประยุกต์ใช้งานกับ การใช้งานอื่นๆ ได้มากขึ้น เช่น นำไปใช้ในการควบคุมอุปกรณ์ไฟฟ้าและนำไปใช้งานกับเซนเซอร์ได้

5.4.3 สามารถนำไปประยุกต์ใช้งานในสถานที่ที่มีความจำเป็นในการติดต่อสื่อสารระหว่างมิเตอร์อัจฉริยะกับผู้ใช้งาน เช่น ฟาร์มเกษตร เป็นต้น

5.4.3 สามารถนำไปพัฒนาเป็นเครือข่ายไร้สายในระบบที่ใหญ่ขึ้นได้ เพื่อพัฒนาเป็นโครงข่ายที่มีประสิทธิภาพสูงขึ้นเนื่องจาก ข้อมูลสามารถส่งไปถึงเป้าหมายได้หลายทาง และแก้ปัญหาการส่งข้อมูลให้ส่งได้ไกลขึ้นอีกด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] เอกชัย มะการ, เรียนรู้ เข้าใจ ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino, พิมพ์ครั้งที่1., กรุงเทพฯ, สนพ.บริษัท อีทีที จำกัด, 2552.
- [2] กฤตภัค เกตุเรือง, กษิตรีเดช ชื่นชูผล และ กาญจนา ลีมวัฒนาชัย, “ระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน ZIGBEE,” ปรินญาณินพนวิศวกรรมศาสตรบัณฑิต, ภาควิชาวิศวกรรมโทรคมนาคม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2551.
- [3] ปณัสนัย คุณาคติภพ, ประวุฒิ คุ่มโชคไพศาล และ พงศธร อิศรไกรศีล, “เครือข่ายเซนเซอร์ไร้สายในโรงงานอุตสาหกรรมโดยใช้ Zigbee,” ปรินญาณินพนวิศวกรรมศาสตรบัณฑิต, ภาควิชาวิศวกรรมสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2551.
- [4] เขมชาติ กริชไพฑูรย์, ญรรูฐดิ ผังวิไล และ ทัธชา ไชยแสงคำ, “การศึกษาและออกแบบโปรโตคอลซิกบีสำหรับระบบโทรมาตร,” ปรินญาณินพนวิศวกรรมศาสตรบัณฑิต, ภาควิชาวิศวกรรมอัตโนมัติ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2555.
- [5] ธีระวัฒน์ ประกอบผล, การพัฒนาแอปพลิเคชันด้วย Visual C# 2010, พิมพ์ครั้งที่ 2., กรุงเทพฯ, สำนักพิมพ์ซิมพลีฟาย, 2556.
- [6] อรดี มุสิกานนท์, “Zigbee ระบบสื่อสารทางเลือกในระบบAMI,” ไฟฟ้าสาร, ปีที่18, ฉบับที่ 3 ก.ค., 2554, หน้า45-48.
- [7] COMPUTER AND COMMUNICATION “การสื่อสารข้อมูล(DataCommunication)” เข้าถึงได้จาก
<http://personal.sut.ac.th/paramate/files/compcom/compcomm01.pdf>
- [8] THAI EASY ELEC “XBee Topology” เข้าถึงได้จาก
<http://www.thaieasyelec.com/article-wiki/embedded-electronics-application/xbee-basic-configuration-in-network-application.html>
- [9] THAI EASY ELEC “Arduino คืออะไร” เข้าถึงได้จาก
<http://www.thaieasyelec.com/article-wiki/basic-electronics/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1-arduino-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-arduino.html>

- [10] ปฏิบัติการวิศวกรรมไฟฟ้าเบื้องต้น คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ “DC Power Supplies” เข้าถึงได้จาก
<http://www.ee.eng.cmu.ac.th/OnlineCourses/252283/Lab5-Theory1.html>
- [11] เอกสารประกอบการสอน วิชา 222210 การออกแบบวงจรอิเล็กทรอนิกส์ “Power Supply Design” เข้าถึงได้จาก
<http://www.te.kmutnb.ac.th/~msn/222210lecture08.pdf>
- [12] ภาคไอซีบัฟเฟอร์ (Buffer) และไดเวอร์ Driver “IC 74LS245” เข้าถึงได้จาก
http://www.technican.ac.th/nan_ntc/adisak51/start04.html
- [13] สถาบันวิจัยพลังงาน จุฬาลงกรณ์มหาวิทยาลัย “ความรู้พื้นฐานเกี่ยวกับมาตรอัจฉริยะ (Smart Meter) และแนวทางการทำข้อกำหนดเฉพาะมาตรอัจฉริยะ” เข้าถึงได้จาก
<http://www.eri.chula.ac.th/eri-main/wp-content/uploads/2014/03/Paper-3-of-3-Smart-Meter.pdf>
- [14] พูนศักดิ์ พรเพิ่มพูน, “ระบบไร้สายโดยใช้ Zigbee เพื่อควบคุมและติดตามสถานะเครื่องจักร และเซ็นเซอร์ในโรงงานผ่านเครือข่ายอินเทอร์เน็ต,” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต, คณะสถิติประยุกต์, สถาบันบัณฑิตพัฒนบริหารศาสตร์, 2556.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโมดูล ZIGBEE สำหรับมิเตอร์อัจฉริยะ

DEVELOPMENT OF ZIGBEE MODULE FOR SMART METER

นพพล ปาลีภูษ นัสณีษ์ชวงษ์ และเบญจพร ก้อนผา

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขที่ 1 ซอยฉลองกรุง 1 แขวงลาดกระบัง เขตลาดกระบัง กรุงเทพมหานคร 10520 โทร/โทรสาร. 0-2739-2478

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อนำระบบเครือข่ายไร้สายมาประยุกต์ใช้กับมิเตอร์อัจฉริยะ โดยได้ทำการออกแบบและพัฒนาโมดูล Zigbee เพื่อความสะดวกในการส่งข้อมูลจากมิเตอร์อัจฉริยะมาแสดงผลยังผู้ใช้ไฟฟ้า โดยแบ่งส่วนระบบการทำงานเป็นส่วนด้วยกัน คือ ส่วนโหนดอุปกรณ์ปลายทาง (End Device Node) ส่วนโหนดเราเตอร์ (Router Node) และส่วนโหนดโคออดิเนเตอร์ (Coordinator Node) เริ่มจากการจำลองการส่งข้อมูลของมิเตอร์อัจฉริยะ โดยให้โมดูล Zigbee ในภาคส่งนี้เป็นโหนดอุปกรณ์ปลายทาง (End Device Node) ส่งข้อมูลผ่านโมดูล Zigbee ตัวกลางที่เป็นโหนดเราเตอร์ (Router Node) เพื่อใช้ในการสร้างเครือข่ายส่งข้อมูลต่อไปให้ โมดูล Zigbee ในภาครับซึ่งทำหน้าที่เป็นโหนดโคออดิเนเตอร์ (Coordinator Node) ไปแสดงผลยังหน้าจอมคอมพิวเตอร์ของผู้ใช้งาน โดยในการส่งข้อมูลของ Zigbee นั้นจะใช้โหมดคำสั่งแบบ API ซึ่งข้อมูลและคำสั่งจะอยู่ในรูป Frame-Based ทั้งหมดจึงทำให้การส่งข้อมูลภายในเครือข่ายมีความถูกต้องแม่นยำ

คำสำคัญ: โมดูล Zigbee, ระบบเครือข่ายไร้สาย, มิเตอร์อัจฉริยะ

ABSTRACT

A paper propose to apply a wireless design of smart meter by designing and developing Zigbee Module for transmitting data. This project is consisted of 3 operations which is End Device Node, Router Node and Coordinator Node. By initiating data simulations on smart meter and assign transmitting Zigbee module as an End Device Node then transfer the data to the Central Zigbee Module which is assigned as Router Node to form a transferring network and pass the data to the Receiving Zigbee which is assigned as Coordinator Node and visualize the data into the user's computer. All Zigbee's data transferring is being transfer in API mode which all data and command are in Frame-Based format hence more accuracy in data transferring across the network.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น

Keywords: Zigbee Module, Wireless network, Smart Meter

1. บทนำ

ในปัจจุบันประเทศไทยได้มีการขยายการติดตั้งมิเตอร์อัจฉริยะ (Smart Meter) เพิ่มขึ้น โดยมิเตอร์อัจฉริยะเป็นเทคโนโลยีที่ทำให้ผู้ใช้ไฟฟ้ามีความสะดวกสบายเนื่องจากผู้ใช้ไฟฟ้าสามารถติดต่อสื่อสารโดยตรงกับระบบควบคุมของผู้ผลิตไฟฟ้าได้บอค่าที่ค้างชำระ ซึ่งส่งผลให้ผู้ใช้ไฟฟ้าสามารถบริหารจัดการการใช้ไฟฟ้าให้เหมาะสมเพื่อลดค่าใช้จ่ายในการใช้ไฟฟ้าลงได้ สำหรับช่องทางการสื่อสารระหว่างมิเตอร์อัจฉริยะกับผู้ใช้งานสามารถเลือกได้จากหลากหลายรูปแบบโดยจะนิยมใช้เทคโนโลยีในการติดต่อสื่อสารข้อมูลแบบไร้สายเพื่อความสะดวกรวดเร็ว และเป็นกรลดค่าใช้จ่ายในการเดินสายไฟอีกด้วย โดย Zigbee ถือเป็นโมดูลไร้สายประเภทหนึ่งที่มีประสิทธิภาพสูง มีราคาถูก ใช้พลังงานน้อย จึงทำให้มีอายุการใช้งานแบตเตอรี่ที่ยาวนาน ดังนั้นจึงมีแนวคิดที่จะนำเสนอการพัฒนาโมดูล Zigbee สำหรับมิเตอร์อัจฉริยะ โดยจะนำ Zigbee มาพัฒนาในการรับส่งข้อมูลระหว่างมิเตอร์อัจฉริยะกับผู้ใช้งาน

2. เทคโนโลยี Zigbee

Zigbee เป็นมาตรฐานของอุปกรณ์ไร้สายถูกกำหนดโดยกลุ่ม Zigbee Alliance ซึ่งเริ่มก่อตั้งขึ้นเมื่อปี 2002 เพื่อใช้ควบคุมอุปกรณ์ต่างๆ รอบตัวที่ใช้ในชีวิตประจำวัน ในแบบไร้สายซึ่งไม่ต้องการความเร็วสูงมากนัก เช่น สวิตช์เปิดปิดแสงสว่าง ระบบควบคุมอุณหภูมิห้องเป็นต้น รวมไปถึงใช้ในการรับค่าจาก Sensor ต่างๆ โดย Zigbee ถูกออกแบบโดยมุ่งให้มีคุณลักษณะดังนี้

1. สื่อสารใช้ความถี่วิทยุ ย่านความถี่ 2.4GHz มี 16 ช่องสัญญาณ อัตราการรับส่งข้อมูล 250Kbps
2. เป็นเครือข่ายไร้สายในระยะใกล้ มีการเชื่อมต่ออย่างซับซ้อนเพื่อรองรับการเชื่อมต่อสำหรับเครือข่ายขนาดใหญ่ ได้ด้วยรูปแบบมาตรฐาน 3 รูปแบบได้แก่ Multiple Topology คือ สตาร์ (star) เพียร์ทูเพียร์ (peer-to-peer) และ เมช (mesh)
3. มาตรฐาน Zigbee ตาม IEEE 802.15.4 กำหนดให้มีโหมดของการใช้พลังงาน 2 สถานะ คือ สถานะที่มีการรับและส่งของข้อมูลซึ่งเรียกว่า

ปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากนำไปใช้

Active และอีกสถานะคือสถานะที่ไม่มีการทำงานชั่วขณะหนึ่งหรือเรียกว่า Sleep

4. สามารถรับส่งข้อมูลได้โดยเชื่อมั่นในความถูกต้องได้
5. ดัดตั้งง่าย สามารถประยุกต์ใช้งานได้หลากหลาย
6. ใช้พลังงานในการทำงานต่ำ
7. ราคาไม่แพง

2. การส่งข้อมูลของZigbee

การส่งข้อมูลแบบ RF ของแต่ละแพ็คเกจในส่วนของ header คือZigbeeจะสามารถกำหนดค่าแอดเดรสอ้างอิงได้ 2 แบบจะประกอบไปด้วย Source Address และDestination Address โดยที่ IEEE 802.15.4 จะมีโครงสร้าง 2 แบบ นั่นคือแอดเดรสอ้างอิงแบบ short 16-bit address และแอดเดรสอ้างอิงแบบ long 64-bit address ปกติแล้วจะถูกกำหนดค่ามาจากโรงงานเป็นแอดเดรสอ้างอิง 64 บิต ซึ่งจะสามารถอ่านค่าได้จากพารามิเตอร์ SH (Serial Number Low) และ SL (Serial Number High) การใช้งานแอดเดรสอ้างอิง 64 บิต

โหมดคำสั่งเป็นการปรับส่วนของพารามิเตอร์ของXbee ซึ่งจะมีการกำหนด 2 แบบคือ คำสั่งแบบAT กับคำสั่งแบบAPI

คำสั่งแบบ API จะมีรูปแบบการใช้งานที่ค่อนข้างจะยุ่งยากมากกว่าคำสั่งแบบAT แต่สามารถใช้งานได้ยืดหยุ่นกว่า เช่น การใช้งานคำสั่ง Remote และ การควบคุม I/O โดยโหมดคำสั่งแบบAPIใน 1 ชุดคำสั่งจะแบ่งออกเป็น 4 กลุ่มด้วยกันคือ

- 1) Start Delimiter เป็นส่วนเริ่มต้นของ API จะใช้ 0X7E เป็นตัวกันเพื่อบอกให้รู้ว่านี่คือจุดเริ่มต้นของ API มีขนาด 1 Byte
- 2) Length คือ จำนวน Byte ของ Frame Data มีขนาด 2 Byte
- 3) Frame Data คือ คำสั่ง และ Data ที่ต้องการส่ง หรือ ที่รับมา Frame Data จะมีรายละเอียดแตกต่างกันไป ตาม Frame Type ที่ใช้ซึ่งจะอธิบายในหัวข้อต่อไปอีกที
- 4) Checksum เป็นตัวที่เอาไว้ตรวจสอบเช็คความถูกต้องของ Data ที่ได้รับมาว่าถูกต้องหรือไม่ มีขนาด 1 Byte

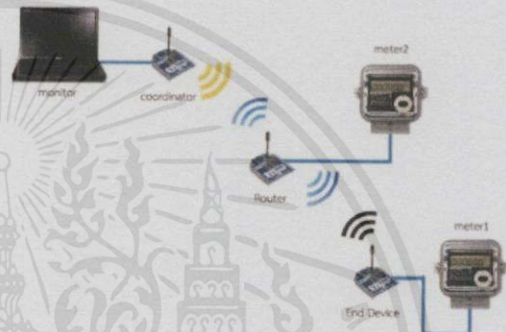
3. มิเตอร์อัจฉริยะ (Smart Meter)

มิเตอร์อัจฉริยะ (Smart Meter) คือมิเตอร์วัดพลังงานไฟฟ้าชนิดใหม่ที่ถูกพัฒนามาเพื่อแทนที่มิเตอร์วัดพลังงานแบบเดิมที่มีอยู่ (แบบจานหมุน) โดยสามารถอ่านค่าพลังงานไฟฟ้าที่ถูกใช้ไปด้วยวงจรอิเล็กทรอนิกส์และส่งข้อมูลในรูปแบบดิจิทัลไปที่ระบบควบคุม (Control System) มิเตอร์อัจฉริยะสามารถติดต่อสื่อสารโดยตรงกับระบบควบคุมได้บ้อยเท่าที่ต้องการ โดยไม่จำเป็นต้องส่งพนักงานมาเพื่ออ่านค่าหน่วยไฟฟ้าเดือนละครั้งแบบปัจจุบัน ช่องทางการสื่อสารอาจเลือกได้จากหลายรูปแบบ ตัวอย่างเช่น รูปแบบการสื่อสารผ่านทางเครือข่ายโทรศัพท์เคลื่อนที่ หรือรูปแบบการสื่อสารผ่านคลื่น RF เป็นต้น

4. การออกแบบระบบการทำงาน

การทำงานโดยภาพรวมจะสร้างเครือข่ายการติดต่อสื่อสารโดยใช้Zigbeeเป็นศูนย์กลางในการติดต่อสื่อสาร โดยมีจุดประสงค์เพื่อการติดต่อสื่อสารแบบไร้สายระหว่างศูนย์กลางการควบคุมและแสดงข้อมูลการใช้ไฟฟ้ากับมิเตอร์อัจฉริยะ ในตอนนี้จะอธิบายการออกแบบและการสร้างอุปกรณ์ในแต่ละส่วนอีกทั้งแสดงแนวคิดในการออกแบบ

เพื่อให้สะดวกต่อการใช้งานและยังคงปริมาณและความยาวของสาย จึงมีการออกแบบอุปกรณ์ในการรับ-ส่งระหว่างมิเตอร์อัจฉริยะกับศูนย์กลางการควบคุมและแสดงข้อมูล ให้สามารถรับ-ส่งข้อมูลกันได้โดยไร้สายซึ่งสามารถแสดงโครงสร้างการทำงานของระบบได้ดังนี้



รูปที่ 1 โครงสร้างการทำงานของระบบ

จากรูปที่ 1 รูปการโครงสร้างทำงานของระบบที่ใช้การสร้างเครือข่ายแบบ Cluster Tree (Tree) Network ซึ่งเป็นรูปแบบการสร้างเครือข่ายในการติดต่อสื่อสารแบบส่งผ่าน โดยแบ่งส่วนระบบของโครงงานได้เป็น 3 ส่วนด้วยกัน คือ ส่วนโหนดอุปกรณ์ปลายทาง (End Device Node) ส่วนโหนดเรดเดอร์ (Router Node) และส่วนโหนดโคออดิเนเตอร์ (Coordinator Node)

ในส่วน โหมด Zigbeeทำหน้าที่เป็น โหนดโคออดิเนเตอร์นี้จะเป็นส่วนของศูนย์กลางการควบคุมและแสดงข้อมูล ซึ่งเป็นตัวกำหนดทิศทางในการติดต่อรับข้อมูลการใช้ไฟฟ้าจากมิเตอร์ไฟฟ้าที่เชื่อมต่ออยู่กับโหนดต่างๆในเครือข่าย เพื่อนำข้อมูลการใช้ไฟฟ้าไปแสดงผลที่หน้าจอคอมพิวเตอร์ โดยผ่านโปรแกรมแสดงข้อมูลการใช้ไฟฟ้า

ในส่วนของโหนดเรดเดอร์นั้นจะเป็นตัวจัดการโหนดอุปกรณ์ปลายทางในโครงข่ายโดยจะรับคำสั่งในการติดต่อกับโหนดอุปกรณ์ปลายทางมาจากส่วนของศูนย์กลางการควบคุมและแสดงข้อมูล (โหนดโคออดิเนเตอร์) และเป็นตัวกลางในการส่งข้อมูลการใช้ไฟฟ้าจากมิเตอร์ไฟฟ้า (โหนดอุปกรณ์ปลายทาง) ที่อยู่ไกลจากศูนย์กลางการควบคุมและแสดงข้อมูล ไปให้กับศูนย์กลางการควบคุมและแสดงข้อมูล (โหนดโคออดิเนเตอร์) เพื่อทำการแสดงผลบนหน้าจอคอมพิวเตอร์ผ่านโปรแกรมแสดงข้อมูลการใช้ไฟฟ้า

ในส่วนของโหนดอุปกรณ์ปลายทางนั้น เราจะทำการจำลองสร้างชุดข้อมูลของมิเตอร์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวกำหนด เมื่อมีการติดต่อขอรับข้อมูลมาจากส่วนของศูนย์กลางการควบคุมและแสดงข้อมูล (โหนดโคออดิเนเตอร์) โหนดอุปกรณ์ปลายทาง จะทำการส่งข้อมูลให้กับโหนดเรดาร์ที่เป็นตัวกลางที่จะนำข้อมูลส่งต่อไปให้กับศูนย์กลางการควบคุมและแสดงข้อมูลเพื่อแสดงผลบนหน้าจอคอมพิวเตอร์ต่อไป

5. หลักการออกแบบโปรแกรม

5.1 หลักการออกแบบโปรแกรมฝั่งส่ง

ในส่วนของโปรแกรมฝั่งส่งจะมีการจำลองการส่งข้อมูลของมิเตอร์ไฟฟ้า โค้ดข้อมูลที่สร้างขึ้นมานั้นจะมีรูปแบบเป็นโปรโตคอล MEATH ซึ่งแสดงตัวอย่างได้ดังรูปที่ 2



รูปที่ 2 ตัวอย่างข้อมูลฝั่งส่งที่สร้างขึ้นมาในรูปแบบโปรโตคอล MEATH

เมื่อมีการสร้างข้อมูลในฝั่งส่งขึ้นมาแล้วข้อมูลนี้จะถูกดำเนินการกำหนดรูปแบบ Packet ให้สอดคล้องกับโปรโตคอลมาตรฐานของ Zigbee ที่เป็นโหมด API ซึ่งแสดงตัวอย่างได้ดังรูปที่ 3

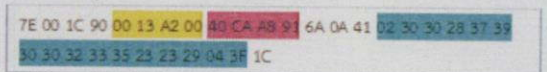


รูปที่ 3 ตัวอย่างPacketข้อมูลในโหมดAPI ของฝั่งส่งตัวอย่างข้อมูลฝั่งส่งที่สร้างขึ้นมาในรูปแบบโปรโตคอล

5.1 หลักการออกแบบโปรแกรมฝั่งรับ

ในส่วนโปรแกรมฝั่งรับนั้น จะทำการออกแบบโปรแกรมให้สามารถกำหนดทิศทางในการขอรับข้อมูลการใช้ไฟฟ้าจากมิเตอร์ไฟฟ้าที่อยู่ในโครงข่ายได้ โค้ดให้ฝั่งรับ(โหนดโคออดิเนเตอร์) ขอรับข้อมูลการใช้ไฟฟ้าจากมิเตอร์ไฟฟ้าด้วยการกำหนดค่า Destination Address หรือค่า SH SLของมิเตอร์ไฟฟ้าตัวที่ต้องการข้อมูล

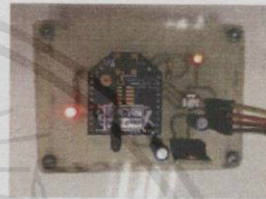
โปรโตคอลมาตรฐานของ Zigbee ที่เป็นโหมด API ในส่วนของฝั่งรับ เมื่อได้รับข้อมูลจากมิเตอร์ไฟฟ้า ได้กำหนดรูปแบบ Packet ไว้ดังตัวอย่าง รูปที่ 4



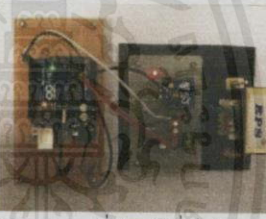
รูปที่ 4 Packetข้อมูลที่รับจากมิเตอร์ไฟฟ้า ในโหมดAPI ของฝั่งรับ

6. การทดลองการจัดการเครือข่ายการติดต่อสื่อสารของโมดูล Zigbeeโดยใช้โหมดAPI

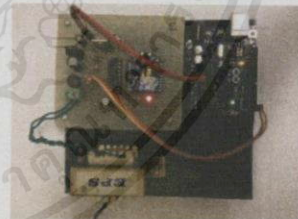
ในการทดลองนี้ได้ทำการทดสอบการส่งข้อมูลไร้สายผ่านโมดูลZigbeeจากอุปกรณ์จำลองการส่งข้อมูลของมิเตอร์ไฟฟ้า(โหนดอุปกรณ์ปลายทาง) ผ่านโมดูลZigbeeที่ทำหน้าที่เป็นโหนดเรดาร์ เพื่อเป็นการขยายระยะทางและสร้างเครือข่ายให้กับระบบการทำงาน ไปยังโมดูลZigbeeส่วนควบคุมและแสดงผล(โหนดโคออดิเนเตอร์) เพื่อแสดงผลผ่านโปรแกรมแสดงข้อมูลการใช้ไฟฟ้าในคอมพิวเตอร์ในโหมด API โค้ดอุปกรณ์ที่ใช้ในการทำการทดลองมีดังนี้



รูปที่ 5 โมดูลZigbeeที่ทำหน้าที่เป็นโหนดโคออดิเนเตอร์



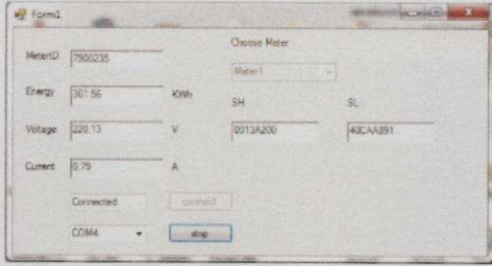
รูปที่ 6 โมดูลZigbeeที่ทำหน้าที่เป็นโหนดเรดาร์



รูปที่ 7 โมดูลZigbeeที่ทำหน้าที่เป็นโหนดอุปกรณ์ปลายทาง

ในส่วนของโปรแกรมแสดงข้อมูลการใช้ไฟฟ้าของมิเตอร์ไฟฟ้าเป็นโปรแกรมที่ใช้ควบคุมและแสดงข้อมูลของมิเตอร์ไฟฟ้าที่อยู่ภายในโครงข่าย โค้ดสามารถเลือกแสดงค่าของมิเตอร์ไฟฟ้าตัวที่ต้องการผ่านการเรียกข้อมูลโดยใช้ Destination Address หรือค่า SH SL ในการติดต่อขอรับข้อมูลของมิเตอร์ไฟฟ้า ตัวอย่างการแสดงผลของmeter1 นั้นจะสามารถแสดงผลข้อมูลการใช้ไฟฟ้าผ่านหน้าต่างของโปรแกรม ได้ดังรูปที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 โปรแกรมแสดงข้อมูลการใช้ไฟฟ้าของมิเตอร์

6. สรุปผลการทดลอง

โครงการเรื่องการพัฒนาโมดูล Zigbee สำหรับมิเตอร์อัจฉริยะ จากการทดลองในแต่ละส่วนของโครงการสามารถสรุปผลได้ดังนี้

สามารถจำลองโปรโตคอลในการสื่อสารของมิเตอร์อัจฉริยะ และสามารถควบคุมโหนดอุปกรณ์ปลายทางให้ส่งข้อมูลมาให้กับส่วนควบคุมและแสดงผลได้โดยใช้โหนดAPIเป็นตัวกำหนดทิศทางในการติดต่อสื่อสาร

จากการทดลองพบว่า สามารถนำโมดูล Zigbee มาสร้างเครือข่ายได้รูปแบบ คือ การเชื่อมต่อเครือข่ายแบบ Star, Cluster tree (tree) และ Mesh ซึ่งจากการทดลองพบว่า การสร้างเครือข่ายแบบ Mesh เป็นการสร้างเครือข่ายที่มีประสิทธิภาพสูงเนื่องจากสามารถส่งข้อมูลไปถึงเป้าหมายได้หลายทิศทาง

7. วิเคราะห์ผลการทดลอง

ในการทดลองนี้พบว่าระยะทางและสิ่งกีดขวางในการติดต่อสื่อสารข้อมูลระหว่างโมดูล Zigbee จะทำให้การติดต่อสื่อสารข้อมูลระหว่างกันเกิดความล่าช้าหรือเกิดความผิดพลาดขึ้นได้

8. ทิศตติกรรมประกาศ

ปริญญานิพนธ์สำเร็จแล้วไปได้ด้วยดีต้องขอขอบพระคุณ ศส.ดร.สมยศ เกียรติวนิชวิไล และศส.ดร.ศิริเดช บุญแสง อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ช่วยผลักดันและให้โอกาสโครงการนี้ให้สำเร็จเป็นจริงขึ้นมา โดยคอยช่วยเหลือ ชี้แนะแนวทางการทำงาน และให้คำปรึกษา ข้อเสนอแนะที่เป็นประโยชน์ต่อโครงการมาโดยตลอด ตลอดจนตรวจและแก้ไขปริญญานิพนธ์จนสำเร็จเรียบร้อยเป็นอย่างดี

ขอขอบคุณรุ่นพี่ และเพื่อนๆ ที่ให้คำแนะนำและชี้แนะแนวทางในการทำงาน และที่ขาดไม่ได้คือเพื่อนๆ ภาควิชาวิศวกรรมไฟฟ้า วิศวกรรมอิเล็กทรอนิกส์ วิศวกรรมโทรคมนาคมและชุมนุมโรบอทที่ช่วยชี้แนะแนวทาง ให้คำแนะนำและร่วมแก้ไขปัญหาเสมอ ไม่ว่าจะปัญหานั้นจะเล็กน้อยหรือใหญ่ก็ตามทำให้โปรเจกต์สามารถเดินหน้าไปด้วยดีมาโดยตลอด

ทำอนึ่งคณะผู้จัดทำ ขอกราบขอบพระคุณ บิดามารดา และครอบครัวซึ่งให้การสนับสนุนในด้านการเงินและให้กำลังใจอีกทั้งเพื่อนๆ พี่ๆ น้องๆ ที่ให้ความช่วยเหลือในด้านการจัดหาอุปกรณ์ต่างๆ ในการทำโครงการครั้งนี้จนสำเร็จได้ด้วยดี จึงขอขอบคุณไว้ ณ ที่นี้ด้วย

9. เอกสารอ้างอิง

- [1] กฤตภัค เกตุเวียง, กษิต์เดช ชื่นชูผล และ กาญจนา ลีมันวัฒนาชัย, “ระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน ZIGBEE,” ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต, ภาควิชาวิศวกรรมโทรคมนาคม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2551
- [2] ปณิษฐ์ คุณาศักดิ์ภพ, ประวุฒิ คุ่มโชคไพศาล และ พงศธร อิศรไกรสีล, “เครือข่ายเซนเซอร์ไร้สายในโรงงานอุตสาหกรรมโดยใช้ Zigbee,” ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต, ภาควิชาวิศวกรรมสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2551
- [3] เขมชาติ กริชไพฑูรย์, ณัฐวุฒิ ผังวิไล และ ทศชา ไซแสงคำ, “การศึกษาและออกแบบโปรโตคอลซิกบีสำหรับระบบโทรมาตร,” ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต, ภาควิชาวิศวกรรมอัตโนมัติ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2555.

10. ประวัติผู้เขียน



นายนพดล ปาลีถูก
เกิดเมื่อวันที่ 12 สิงหาคม 2535
สถานที่เกิดจังหวัดเชียงใหม่
Email: nopadon.paleekui@gmail.com



นางสาวนัสสินี ชวงไช
เกิดเมื่อวันที่ 11 มีนาคม 2536
สถานที่เกิดจังหวัดกรุงเทพมหานคร
Email: ns.youngyai@gmail.com



นางสาวเบญจพร ก้อนผา
เกิดเมื่อวันที่ 11 กุมภาพันธ์ 2536
สถานที่เกิดจังหวัดกรุงเทพมหานคร
Email: benjaporn.konpha@gmail.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแสดงข้อมูลบนหน้าจอคอมพิวเตอร์

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.Threading;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        //Holds data received until we get a terminator.
        string tString = string.Empty;
        ///End of transmission byte in this case EOT (ASCII 4).
        ///7E 00 0F 10 01 00 13 A2 00 40 CA A8 91 FF FE 00 00 30 C9
        byte[] close = { 0x7E, 0x00, 0x0F, 0x10, 0x01, 0x00, 0x13, 0xA2, 0x00, 0x40, 0xCA,
            0xA8, 0x91, 0xFF, 0xFE, 0x00, 0x00, 0x60, 0x99 };
        byte[] open = { 0x7E, 0x00, 0x0F, 0x10, 0x01, 0x00, 0x13, 0xA2, 0x00, 0x40, 0xCA,
            0xA8, 0x91, 0xFF, 0xFE, 0x00, 0x00, 0x06, 0xF3 };
        byte[] close1 = { 0x7E, 0x00, 0x0F, 0x10, 0x01, 0x00, 0x13, 0xA2, 0x00, 0x40, 0xB9,
            0x70, 0x45, 0xFF, 0xFE, 0x00, 0x00, 0x60, 0x2E };
        byte[] open1 = { 0x7E, 0x00, 0x0F, 0x10, 0x01, 0x00, 0x13, 0xA2, 0x00, 0x40, 0xB9,
            0x70, 0x45, 0xFF, 0xFE, 0x00, 0x00, 0x06, 0x88 };
        byte[] SH1 = new byte[4];
        byte[] SL1 = new byte[4];

        char[] data = new char[7];
        char[] data1 = new char[9];
        char[] data2 = new char[5];
        char[] data3 = new char[5];
        char[] mes = new char[32];

        public Form1()
    {

```

```

InitializeComponent();
button1.Enabled = true;
button2.Enabled = false;
foreach (String s in System.IO.Ports.SerialPort.GetPortNames())
{
comboBox1.Items.Add(s);
}
comboBox1.SelectedIndex = 0;
comboBox2.Items.Add("Meter1");
comboBox2.Items.Add("Meter2");
}
private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
Thread.Sleep(100);
byte[] buffer = new byte[32];
int bytesRead;
if (serialPort1.IsOpen)
{
bytesRead = serialPort1.Read(buffer, 0, 32);
for (int i = 0; i < 4; i++)
{
SH1[i] = buffer[i + 4];
SL1[i] = buffer[i + 8];
}
SetText4(BitConverter.ToString(SH1).Replace("-", string.Empty));
SetText5(BitConverter.ToString(SL1).Replace("-", string.Empty));
}
tString += Encoding.ASCII.GetString(buffer, 0,32);
if (tString.Length > 32) { tString = string.Empty;
}
if (tString.Length == 32)
{
mes = tString.ToCharArray( );
if (mes[16] == 0x30 && mes[17] == 0x30)
{
ไม่ว่ากรณีใดๆทั้งสิ้น อีกหนึ่งหน้าเพื่อให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
for (int i = 0; i < 7; i++)

```

```

{
data[i] = mes[i + 19];
}
double math=Convert.ToDouble(new string(data));
SetText(math.ToString());
}
else if (mes[16] == 0x44 && mes[17] == 0x37)
{
for (int i = 0; i < 9; i++)
{
data1[i] = mes[i + 19];
}
double math = Convert.ToDouble(new string(data1));
math *= 0.01;
SetText1(math.ToString());
}
else if (mes[16] == 0x44 && mes[17] == 0x30)
{
for (int i = 0; i < 5; i++)
{
data2[i] = mes[i + 19];
}
double math = Convert.ToDouble(new string(data2));
math *= 0.01;
SetText2(math.ToString());
}
else if (mes[16] == 0x44 && mes[17] == 0x32)
{
for (int i = 0; i < 5; i++)
{
data3[i] = mes[i + 19];
}
double math = Convert.ToDouble(new string(data3));
math *= 0.01;
SetText3(math.ToString());
}

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่สามารถนำออกจากรั้วมหาวิทยาลัยเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าในรูปแบบใดทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tString = string.Empty;
}
}

delegate void SetTextCallback(string text);
private void SetText(string text)
{
//ตรวจสอบว่า textBox1 จำเป็นต้องกลับไปทำงานที่ Main Thread หรือไม่
if (this.richTextBox1.InvokeRequired)
{
//สร้าง Delegate แล้วส่งชื่อ Method ที่จะทำงานไปด้วย
SetTextCallback d = new SetTextCallback(SetText);
this.Invoke(d, new object[] { text });
}
else
{
this.richTextBox1.Text = text;
}
}
private void SetText1(string text)
{
//ตรวจสอบว่า textBox1 จำเป็นต้องกลับไปทำงานที่ Main Thread หรือไม่
if (this.richTextBox2.InvokeRequired)
{
//สร้าง Delegate แล้วส่งชื่อ Method ที่จะทำงานไปด้วย
SetTextCallback d = new SetTextCallback(SetText1);
this.Invoke(d, new object[] { text });
}
else
{
this.richTextBox2.Text = text;
}
}
private void SetText2(string text)
{
//ตรวจสอบว่า textBox1 จำเป็นต้องกลับไปทำงานที่ Main Thread หรือไม่
if (this.richTextBox3.InvokeRequired)
{
//สร้าง Delegate แล้วส่งชื่อ Method ที่จะทำงานไปด้วย

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์และบุคลากรที่ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าในรูปแบบใดก็ตาม หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายวิชาการ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SetTextCallback d = new SetTextCallback(SetText2);
this.Invoke(d, new object[] { text });
}
else
{
this.richTextBox3.Text = text;
}
}
private void SetText3(string text)
{
//ตรวจสอบว่า textBox1 จำเป็นต้องกลับไปทำงานที่ Main Thread หรือไม่
if (this.richTextBox4.InvokeRequired)
{
//สร้าง Delegate แล้วส่งชื่อ Method ที่จะทำงานไปด้วย
SetTextCallback d = new SetTextCallback(SetText3);
this.Invoke(d, new object[] { text });
}
else
{
this.richTextBox4.Text = text;
}
}
private void SetText4(string text)
{
//ตรวจสอบว่า textBox1 จำเป็นต้องกลับไปทำงานที่ Main Thread หรือไม่
if (this.richTextBox5.InvokeRequired)
{
//สร้าง Delegate แล้วส่งชื่อ Method ที่จะทำงานไปด้วย
SetTextCallback d = new SetTextCallback(SetText4);
this.Invoke(d, new object[] { text });
}
else
{
this.richTextBox5.Text = text;
}
}
private void SetText5(string text)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 "ไม่ว่ากรณีใดๆทั้งสิ้น" อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีควรนำไปใช้

```

//ตรวจสอบว่า textBox1 จำเป็นต้องกลับไปทำงานที่ Main Thread หรือไม่
if (this.richTextBox6.InvokeRequired)
{
//สร้าง Delegate แล้วส่งชื่อ Method ที่จะทำงานไปด้วย
SetTextCallback d = new SetTextCallback(SetText5);
this.Invoke(d, new object[] { text });
}
else
{
this.richTextBox6.Text = text;
}
}

private void button2_Click_1(object sender, EventArgs e)
{
if (serialPort1.IsOpen)
{
if (comboBox2.Text == "Meter1")
{
serialPort1.Write(close, 0, close.Length);
}
if (comboBox2.Text == "Meter2")
{
serialPort1.Write(close1, 0, close1.Length);
}
Thread.Sleep(100);
try
{
serialPort1.Close( );
textBox1.Text = "Close";
button2.Enabled = false;
button1.Enabled = true;
comboBox2.Enabled = true;
SetText("");
SetText1("");
SetText2("");
SetText3("");
SetText4("");
SetText5("");
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะในรูปแบบใดก็ตาม อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจำลองการส่งข้อมูลของMeter 1 End device

```
byte checkbyte=0x00;
```

```
byte checkbyte1=0x00;
```

```
char
```

```
IDdatta[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,  
0xFE,0x00,0x00,0x02,0x30,0x30,0x28,0x37,0x39,0x30,0x30,0x32,0x33,0x35,0x23,0x23,0x  
29,0x04,0x3F,0xB0};
```

```
char
```

```
apiENG1[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF  
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x32,0x39,0x33,0x34,0x39,0  
x29,0x04,0x43,0x76};
```

```
char
```

```
apiENG2[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF  
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x30,0x31,0x35,0x36,0  
x29,0x04,0x47,0x7E};
```

```
char
```

```
apiENG3[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF  
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x30,0x36,0x37,0x37,0  
x29,0x04,0x43,0x7A};
```

```
char
```

```
apiENG4[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF  
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x31,0x30,0x31,0x32,0  
x29,0x04,0x47,0x86};
```

```
char
```

```
apiENG5[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF  
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x31,0x34,0x32,0x33,0  
x29,0x04,0x41,0x86};
```

```
char
```

```
apivms1[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF  
F,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x31,0x38,0x32,0x32,0x23,0x23,0x23,0x23,0  
x29,0x04,0x4A,0xB6};
```

```
char
```

```
apivms2[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF  
F,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x32,0x30,0x31,0x33,0x23,0x23,0x23,0x23,0  
x29,0x04,0x43,0xC4};
```

```
char
```

```
apivms3[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF
```

```
F,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x32,0x30,0x33,0x39,0x23,0x23,0x23,0x23,0x29,0x04,0x4B,0xB4};
```

```
char
```

```
apivms4[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x31,0x39,0x37,0x38,0x23,0x23,0x23,0x23,0x29,0x04,0x44,0xB0};
```

```
char
```

```
apivms5[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x31,0x38,0x34,0x31,0x23,0x23,0x23,0x23,0x29,0x04,0x4F,0xB0};
```

```
char
```

```
apiims1[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x30,0x38,0x33,0x23,0x23,0x23,0x23,0x29,0x04,0x48,0xBA};
```

```
char
```

```
apiims2[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x30,0x37,0x39,0x23,0x23,0x23,0x23,0x29,0x04,0x4D,0xB0};
```

```
char
```

```
apiims3[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x31,0x32,0x30,0x23,0x23,0x23,0x23,0x29,0x04,0x40,0xCA};
```

```
char
```

```
apiims4[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x31,0x31,0x32,0x23,0x23,0x23,0x23,0x29,0x04,0x43,0xC6};
```

```
char
```

```
apiims5[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x30,0x39,0x31,0x23,0x23,0x23,0x23,0x29,0x04,0x4B,0xB8};
```

```
int count1=0;
```

```
int count2=0;
```

```
int count3=0;
```

```
byte _start=0x06;
```

```
byte _stop=0x60;
```

```
void MeterID()
```

```
{
```

```
for(int i=0;i<34;i++)
```

เอกสารนี้เป็นเอกสารที่ใช้งานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่า void MeterID() ก็ทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
Serial.print(IDdatta[i]);
delay(100);
}
void Energy(){
if(count1==0){
for(int i=0;i<34;i++)
{
Serial.print(apiENG1[i]);
}
}
if(count1==1)
{
for(int i=0;i<34;i++)
{
Serial.print(apiENG2[i]);
}
}
if(count1==2)
{
for(int i=0;i<34;i++)
{
Serial.print(apiENG3[i]);
}
}
if(count1==3)
{
for(int i=0;i<34;i++)
{
Serial.print(apiENG4[i]);
}
}
if(count1==4)
{
for(int i=0;i<34;i++)
Serial.print(apiENG5[i]);count1=-1;
}
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม หากมีให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
count1++;
delay(100);
}
void Vrms( )
{
if(count2==0)
{
for(int i=0;i<34;i++)
{
Serial.print(apivms1[i]);
}
}
if(count2==1)
{
for(int i=0;i<34;i++)
{
Serial.print(apivms2[i]);
}
}
if(count2==2)
{
for(int i=0;i<34;i++)
{
Serial.print(apivms3[i]);
}
}
if(count2==3)
{
for(int i=0;i<34;i++)
{
Serial.print(apivms4[i]);
}
}
if(count2==4)
for(int i=0;i<34;i++)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print(apivms5[i]);count2=-1;
}
}
count2++;
delay(100);
}
void lrms( )
{
if(count3==0)
{
for(int i=0;i<34;i++)
{
Serial.print(apiims1[i]);
}
}
if(count3==1)
{
for(int i=0;i<34;i++)
{
Serial.print(apiims2[i]);
}
}
if(count3==2)
{
for(int i=0;i<34;i++)
{
Serial.print(apiims3[i]);
}
}
if(count3==3)
{
for(int i=0;i<34;i++)
{
Serial.print(apiims4[i]);
}
}
if(count3==4)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าจะวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(int i=0;i<34;i++)
{
Serial.print(apiims5[i]);count3=-1;
}
}
count3++;
delay(1000);
}
void setup()
{
Serial.begin(9600); //set baud rate to 9600
Serial.flush();
}
void loop()
{
waitting:
if(Serial.available(>0)
{
checkbyte=Serial.read();
if(checkbyte==_start)
{
while(1)
{
Energy( );
MeterID( );
Vrms( );
Irms( );
while(Serial.available(>0)
{
checkbyte1=Serial.read();
if(checkbyte1==_stop)
{
checkbyte=0x00;
checkbyte1=0x00;
goto waitting;
}
}
}
}
}
}
}

```

เอกสารนี้เป็นทรัพย์สินที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}
}

```

โปรแกรมจำลองการส่งข้อมูลของMeter 2 Router

```

byte checkbyte=0x00;
byte checkbyte1=0x00;
char
IDdatta[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,
0xFE,0x00,0x00,0x02,0x30,0x30,0x28,0x37,0x39,0x30,0x30,0x32,0x33,0x36,0x23,0x23,0x
29,0x04,0x3C,0xB2};
char
apiENG1[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x39,0x33,0x34,0x39,0
x29,0x04,0x42,0x76};
char
apiENG2[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x32,0x32,0x37,0x36,0
x29,0x04,0x44,0x7C};
char
apiENG3[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x30,0x31,0x35,0x36,0
x29,0x04,0x47,0x7E};
char
apiENG4[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x30,0x36,0x37,0x37,0
x29,0x04,0x43,0x7A};
char
apiENG5[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF
F,0xFE,0x00,0x00,0x02,0x44,0x37,0x28,0x30,0x30,0x30,0x30,0x33,0x31,0x30,0x31,0x32,0
x29,0x04,0x47,0x86};
char
apivms1[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF
F,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x32,0x30,0x39,0x31,0x23,0x23,0x23,0x23,0
x29,0x04,0x49,0xB8};
char
apivms2[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xF

```

```
F,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x32,0x35,0x36,0x32,0x23,0x23,0x23,0x23,0x29,0x04,0x40,0xBE};
```

```
char
```

```
apivms3[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x31,0x38,0x32,0x32,0x23,0x23,0x23,0x23,0x29,0x04,0x4A,0xB6};
```

```
char
```

```
apivms4[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x32,0x30,0x31,0x33,0x23,0x23,0x23,0x23,0x29,0x04,0x43,0xC4};
```

```
char
```

```
apivms5[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x30,0x28,0x32,0x31,0x38,0x34,0x31,0x23,0x23,0x23,0x23,0x29,0x04,0x4F,0xB0};
```

```
char
```

```
apiims1[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x31,0x30,0x37,0x23,0x23,0x23,0x23,0x29,0x04,0x45,0xC0};
```

```
char
```

```
apiims2[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x31,0x32,0x31,0x23,0x23,0x23,0x23,0x29,0x04,0x41,0xC8};
```

```
char
```

```
apiims3[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x30,0x38,0x33,0x23,0x23,0x23,0x23,0x29,0x04,0x48,0xBA};
```

```
char
```

```
apiims4[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x30,0x37,0x39,0x23,0x23,0x23,0x23,0x29,0x04,0x4D,0xB0};
```

```
char
```

```
apiims5[34]={0x7E,0x00,0x1E,0x10,0x01,0x00,0x13,0xA2,0x00,0x40,0xB9,0x73,0x7A,0xFF,0xFE,0x00,0x00,0x02,0x44,0x32,0x28,0x30,0x30,0x31,0x32,0x30,0x23,0x23,0x23,0x23,0x29,0x04,0x40,0xCA};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int count1=0;
```

```
int count2=0;
```

```

int count3=0;
byte _start=0x06;
byte _stop=0x60;
void MeterID()
{
for(int i=0;i<34;i++)
{
Serial.print(IDdatta[i]);
}
delay(100);
}
void Energy()
{
if(count1==0)
{
for(int i=0;i<34;i++)
{
Serial.print(apiENG1[i]);
}
}
if(count1==1)
{
for(int i=0;i<34;i++)
{
Serial.print(apiENG2[i]);
}
}
if(count1==2)
{
for(int i=0;i<34;i++)
{
Serial.print(apiENG3[i]);
}
}
if(count1==3)
{
for(int i=0;i<34;i++)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print(apiENG4[i]);
}
}
if(count1==4)
{
for(int i=0;i<34;i++)
{
Serial.print(apiENG5[i]);count1=-1;
}
}
count1++;
delay(100);
}
void Vrms()
{
if(count2==0)
{
for(int i=0;i<34;i++)
{
Serial.print(apivms1[i]);
}
}
if(count2==1)
{
for(int i=0;i<34;i++)
{
Serial.print(apivms2[i]);
}
}
if(count2==2)
{
for(int i=0;i<34;i++)
{
Serial.print(apivms3[i]);
}
}
if(count2==3)
{

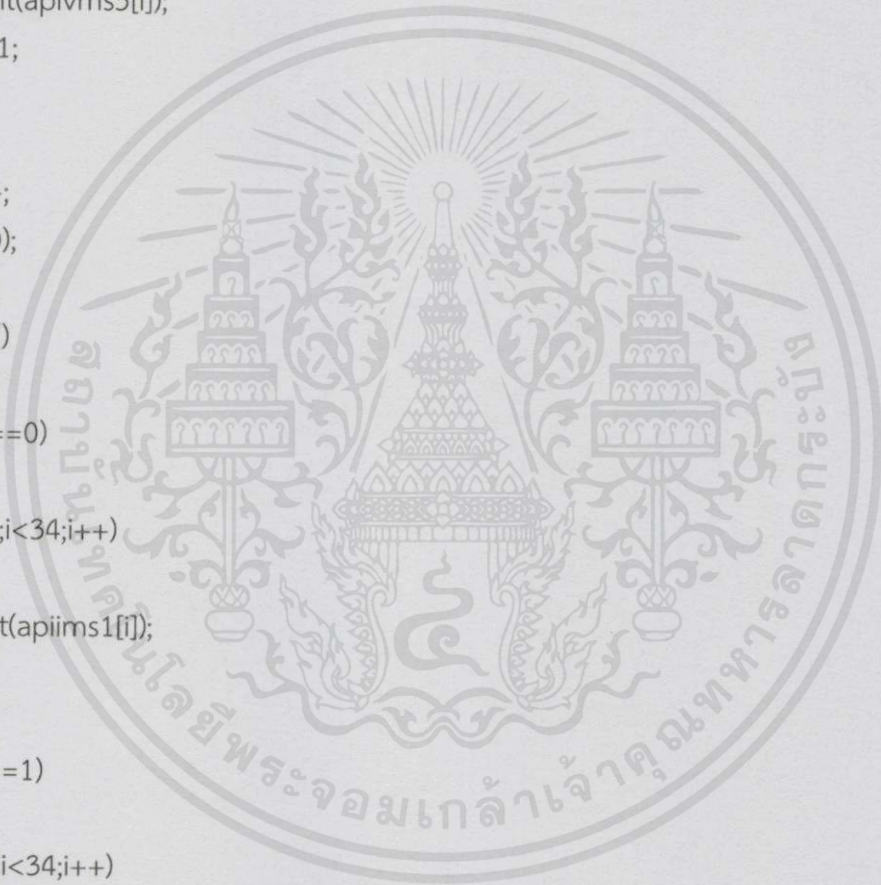
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(int i=0;i<34;i++)
{
Serial.print(apivms4[i]);
}
}
if(count2==4)
{
for(int i=0;i<34;i++)
{
Serial.print(apivms5[i]);
count2=-1;
}
}
count2++;
delay(100);
}
void lrms()
{
if(count3==0)
{
for(int i=0;i<34;i++)
{
Serial.print(apiims1[i]);
}
}
if(count3==1)
{
for(int i=0;i<34;i++)
{
Serial.print(apiims2[i]);
}
}
if(count3==2)
{
for(int i=0;i<34;i++)
Serial.print(apiims3[i]);
}
}

```



เอกสารนี้เป็นเอกสารที่จัดทำไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าวิธีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(count3==3)
{
for(int i=0;i<34;i++)
{
Serial.print(apiims4[i]);
}
}
if(count3==4)
{
for(int i=0;i<34;i++)
{
Serial.print(apiims5[i]);
count3=-1;
}
}
count3++;
delay(1000);
}
void setup()
{
Serial.begin(9600); //set baud rate to 9600
Serial.flush();
}
void loop(){
waitting:
if(Serial.available(>0)
{
checkbyte=Serial.read();
if(checkbyte==_start){
while(1){
Energy();
MeterID();
Vrms();
Irms();
while(Serial.available(>0)
{
checkbyte1=Serial.read();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น "ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่า **while(Serial.available(>0)** ได้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(checkbyte1==_stop)
{
checkbyte=0x00;
checkbyte1=0x00;
goto waitting;
}
}
}
}
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XBee™ Series 2 OEM RF Modules

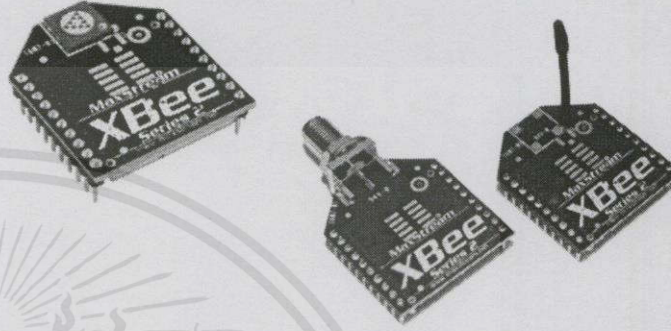
XBee Series 2 OEM RF Modules

ZigBee™ Networks

RF Module Operation

RF Module Configuration

Appendices



Product Manual v1.x.1x - ZigBee Protocol

For OEM RF Module Part Numbers: XB24-BxIT-00x

ZigBee OEM RF Modules by MaxStream, Inc. - a Digi International brand

Firmware Versions:

- 1.0xx - Coordinator, Transparent Operation
- 1.1xx - Coordinator, API Operation
- 1.2xx - Router, End Device, Transparent Operation
- 1.3xx - Router, End Device, API Operation



MaxStream

355 South 520 West, Suite 180
Lindon, UT 84042

Phone: (801) 765-9885

Fax: (801) 765-9895

rf-xperts@maxstream.net

www.MaxStream.net (live chat support)

90000866_A

2007.06.01

© 2007 Digi International, Inc. All rights reserved

No part of the contents of this manual may be transmitted or reproduced in any form or by any means without the written permission of Digi International, Inc.

ZigBee® is a registered trademark of the ZigBee Alliance.

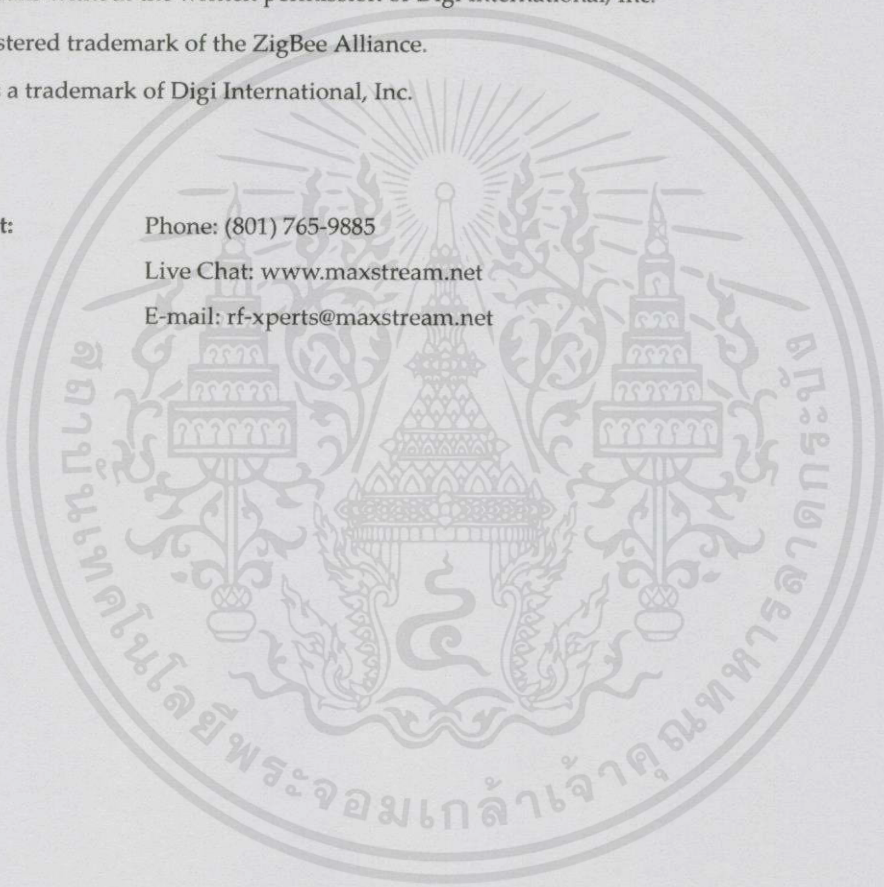
XBee™ Series 2 is a trademark of Digi International, Inc.

Technical Support:

Phone: (801) 765-9885

Live Chat: www.maxstream.net

E-mail: rf-xperts@maxstream.net



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Contents

1. XBee Series 2 OEM RF Modules	4	5. XBee Series 2 Command Reference Tables	29
1.1. Key Features	4	6. API Operation	35
1.1.1. Worldwide Acceptance	4	6.0.1. API Frame Specifications	35
1.2. Specifications	5	6.0.2. API Types	36
1.3. Mechanical Drawings	6	7. Examples	45
1.4. Mounting Considerations	6	7.0.1. Starting an XBee Network	45
1.5. Pin Signals	7	7.0.2. AT Command Programming Examples	46
1.6. Electrical Characteristics	8	8. Manufacturing Support	47
2. RF Module Operation	9	8.1. Interoperability with other EM250 Devices	47
2.1. Serial Communications	9	8.1.1. XBee Data Transmission and Reception	47
2.1.1. UART Data Flow	9	8.1.2. Customizing XBee Default Parameters	47
2.1.2. Serial Buffers	9	8.1.3. XBee Series 2 Custom Bootloader	47
2.1.3. Transparent Operation	11	Appendix A: Definitions	48
2.1.4. API Operation	11	Appendix B: Migrating from the 802.15.4 Protocol	50
2.2. Modes of Operation	12	Appendix C: Agency Certifications	51
2.2.1. Idle Mode	12	Appendix D: Development Guide	52
2.2.2. Transmit Mode	12	Appendix E: Additional Information	60
2.2.3. Receive Mode	13		
2.2.4. Command Mode	13		
2.2.5. Sleep Mode	14		
3. ZigBee Networks	15		
3.1. ZigBee Network Formation	15		
3.1.1. Starting a ZigBee Coordinator	15		
3.1.2. Joining a Router	15		
3.1.3. Joining an End Device	16		
3.2. ZigBee Network Communications	17		
3.2.1. ZigBee Device Addressing	17		
3.2.2. ZigBee Application-layer Addressing	17		
3.2.3. Data Transmission and Routing	18		
4. XBee Series 2 Network Formation	20		
4.1. XBee Series 2 Network Formation	20		
4.1.1. Starting an XBee Series 2 Coordinator	20		
4.1.2. Joining an XBee Series 2 Router to an existing PAN	20		
4.1.3. Joining an XBee Series 2 End Device to an Existing PAN	20		
4.1.4. Network Reset	21		
4.2. XBee Series 2 Addressing	22		
4.2.1. Device Addressing	22		
4.2.2. Application-layer Addressing	23		
4.2.3. XBee Series 2 Endpoint Table	25		
4.3. Advanced Network Features	26		
4.4. I.O. Line Configuration	27		

The XBee Series 2 OEM RF Modules were engineered to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices.

The modules operate within the ISM 2.4 GHz frequency band.



1.1. Key Features

High Performance, Low Cost

- Indoor/Urban: up to 133' (40 m)
- Outdoor line-of-sight: up to 400' (120 m)
- Transmit Power: 2 mW (+3 dBm)
- Receiver Sensitivity: -95 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements
DSSS (Direct Sequence Spread Spectrum)
Each direct sequence channel has over 65,000 unique network addresses available
Point-to-point, point-to-multipoint and peer-to-peer topologies supported
Self-routing, self-healing and fault-tolerant mesh networking

Low Power

XBee Series 2

- TX Current: 40 mA (@3.3 V)
- RX Current: 40 mA (@3.3 V)
- Power-down Current: < 1 μ A @ 25°C

Easy-to-Use

No configuration necessary for out-of box RF communications
AT and API Command Modes for configuring module parameters
Small form factor
Extensive command set
Free X-CTU Software (Testing and configuration software)

Free & Unlimited Technical Support

1.1.1. Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p50] for FCC Requirements.

Systems that contain XBee Series 2 RF Modules inherit MaxStream Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee Series 2 RF Modules are optimized for use in **US, Canada, Australia, Israel and Europe** (contact MaxStream for complete list of agency approvals).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2. Specifications

96

Table 1-01. Specifications of the XBee Series 2 OEM RF Module (PRELIMINARY)

Specification	XBee Series 2
Performance	
Indoor/Urban Range	up to 133 ft. (40 m)
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)
Transmit Power Output (software selectable)	2mW (+3dBm)
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 230400 bps (non-standard baud rates also supported)
Receiver Sensitivity	-95 dBm (1% packet error rate)
Power Requirements	
Supply Voltage	2.8 – 3.4 V
Operating Current (Transmit)	40mA (@ 3.3 V)
Operating Current (Receive)	40mA (@ 3.3 V)
Power-down Current	< 1 uA @ 25°C
General	
Operating Frequency Band	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector
Networking & Security	
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer & Mesh
Number of Channels (software selectable)	16 Direct Sequence Channels
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
Agency Approvals	
United States (FCC Part 15.247)	Pending
Industry Canada (IC)	Pending
Europe (CE)	Pending

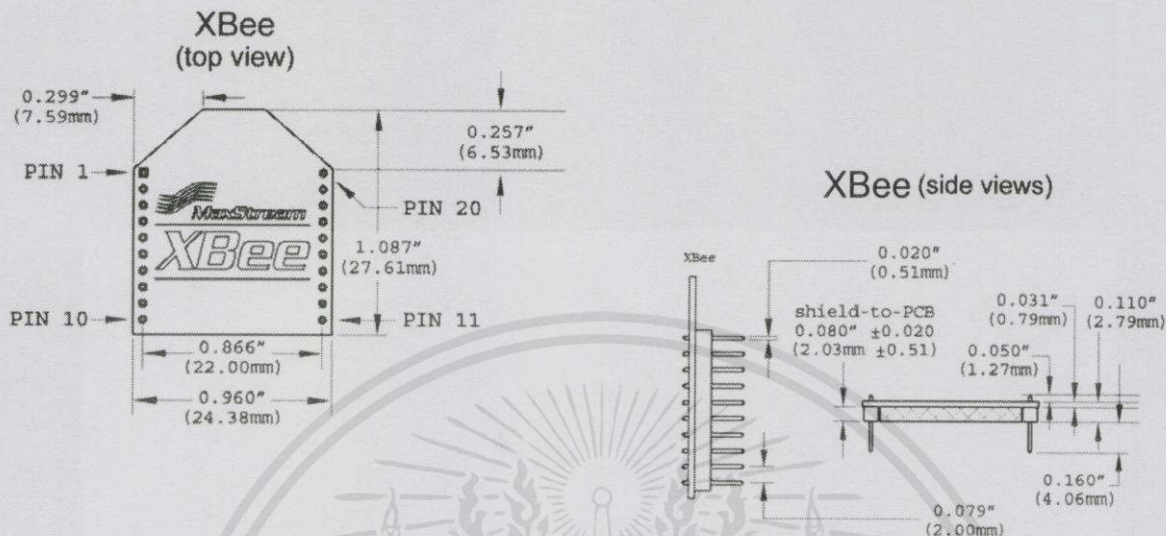
Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Series 2 Antenna" application note located on MaxStream's web site

<http://www.maxstream.net/support/knowledgebase/article.php?kb=153>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3. Mechanical Drawings

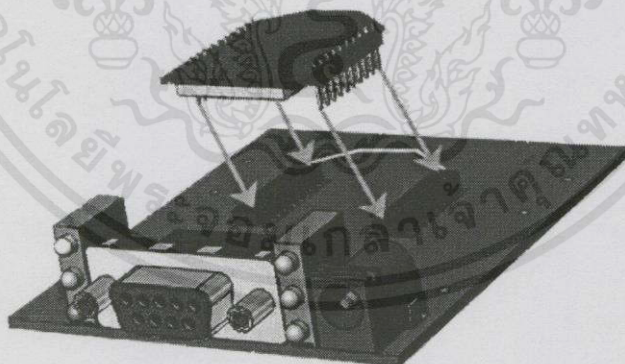
Figure 1-01. Mechanical drawings of the XBee Series 2 OEM RF Modules (antenna options not shown)



1.4. Mounting Considerations

The XBee Series 2 RF Module (through-hole) was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The XBee Series 2 Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

Figure 1-02. XBee Series 2 Module Mounting to an RS-232 Interface Board.



The receptacles used on MaxStream development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, MaxStream currently uses the following receptacles:

- Through-hole single-row receptacles - Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles - Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles - Samtec P/N: SMM-110-02-SM-S

MaxStream also recommends printing an outline of the module on the board to indicate the orientation the module should be mounted.

1.5. Pin Signals

Figure 1-03. XBee Series 2 RF Module Pin Number
(top sides shown - shields on bottom)

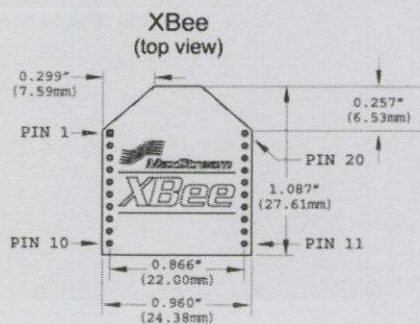


Table 1-02. Pin Assignments for the XBee Series 2 Modules
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DIO8	Either	Digital I/O 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI / DIO10	Output	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM / DIO11	Either	Digital I/O 11
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ/ DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	DIO4	Either	Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	[reserved]	-	Do not connect
15	Associate / DIO5	Either	Associated Indicator, Digital I/O 5
16	RTS / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections to support firmware upgrades: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 30k Ohm resistor attached to RESET
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6. Electrical Characteristics

99

Table 1-03. DC Characteristics of the XBee Series 2 (VCC = 2.8 - 3.4 VDC)

Symbol	Parameter	Condition	Min	Typical	Max	Units
V _{IL}	Input Low Voltage	All Digital Inputs	-	-	0.2 * VCC	V
V _{IH}	Input High Voltage	All Digital Inputs	0.8 * VCC	-	0.18 * VCC	V
V _{OL}	Output Low Voltage	I _{OL} = 2 mA, VCC >= 2.7 V	-	-	0.18 * VCC	V
V _{OH}	Output High Voltage	I _{OH} = -2 mA, VCC >= 2.7 V	0.82 * VCC	-	-	V
I _{IN}	Input Leakage Current	V _{IN} = VCC or GND, all inputs, per pin	-	-	0.5 uA	uA
TX	Transmit Current	VCC = 3.3 V	-	45	-	mA
RX	Receive Current	VCC = 3.3 V	-	50	-	mA
PWR-DWN	Power-down Current	SM parameter = 1	-	< 10	-	uA



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

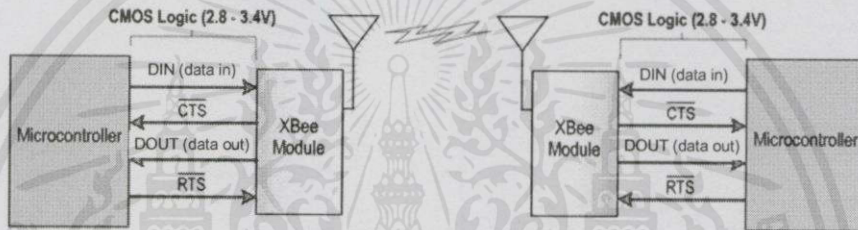
2.1. Serial Communications

The XBee Series 2 OEM RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a MaxStream proprietary RS-232 or USB interface board).

2.1.1. UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

Figure 2-01. System Data Flow Diagram in a UART-interfaced environment (Low-asserted signals distinguished with horizontal line over signal name.)

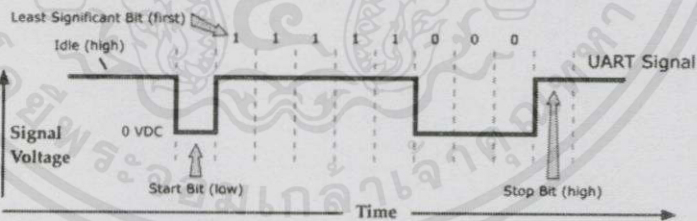


Serial Data

Data enters the module UART through the DIN (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

Figure 2-02. UART data packet 0x1F (decimal number "31") as transmitted through the RF module Example Data Format is 8-N-1 (bits - parity - # of stop bits)



The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

2.1.2. Serial Buffers

The XBee Series 2 modules maintain small buffers to collect received serial and RF data. The serial receive buffer collects incoming serial characters and holds them until they can be processed. The serial transmit buffer collects data that is received via the RF link that will be transmitted out the UART.

Serial Receive Buffer

When serial data enters the RF module through the DIN Pin (3 pin), the data is stored in the serial receive buffer until it can be processed.

Hardware Flow Control (CTS). When the serial receive buffer is 17 bytes away from being full, by default, the module de-asserts $\overline{\text{CTS}}$ (high) to signal to the host device to stop sending data [refer to D7 (DIO7 Configuration) parameter]. $\overline{\text{CTS}}$ is re-asserted after the serial receive buffer has 34 bytes of memory available.

Cases in which the serial receive buffer may become full and possibly overflow:

1. If the module is receiving a continuous stream of RF data, any serial data that arrives on the DIN pin is placed in the serial receive buffer. The data in the serial receive buffer will be transmitted over-the-air when the module is no longer receiving RF data in the network.
2. When data is ready to be transmitted, the module may need to discover a Network Address and/or a Route in order to reach the destination node. Discovery overhead may delay packet transmission.
Refer to the ZigBee Networks --> Mesh Routing sections for more information.

Serial Transmit Buffer

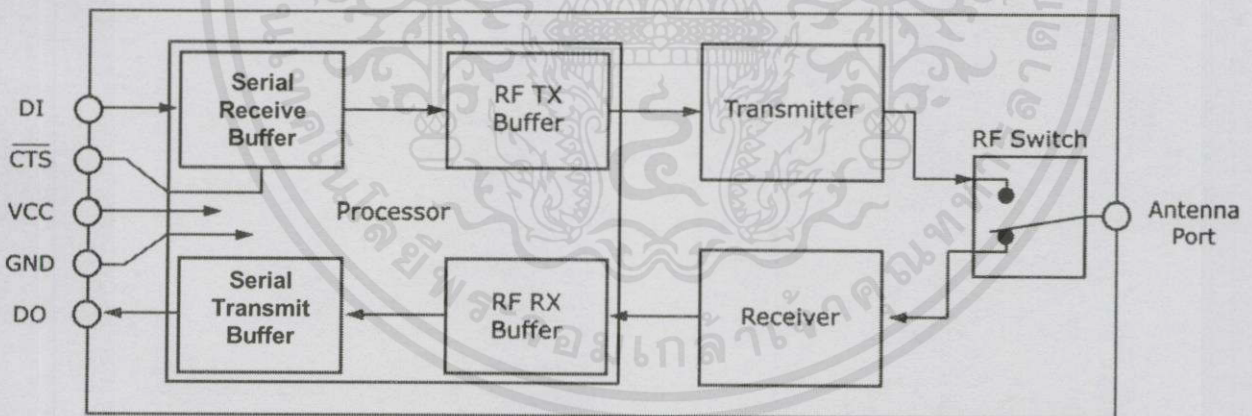
When RF data is received, the data is moved into the serial transmit buffer and is sent out the serial port. If the serial transmit buffer becomes full enough such that all data in a received RF packet won't fit in the serial transmit buffer, the entire RF data packet is dropped.

Hardware Flow Control (RTS). If $\overline{\text{RTS}}$ is enabled for flow control (D6 (DIO6 Configuration) Parameter = 1), data will not be sent out the serial transmit buffer as long as $\overline{\text{RTS}}$ (pin 16) is de-asserted.

Cases in which the serial transmit buffer may become full resulting in dropped RF packets

1. If the RF data rate is set higher than the interface data rate of the module, the module could receive data faster than it can send the data to the host.
2. If the host does not allow the module to transmit data out from the serial transmit buffer because of being held off by hardware flow control.

Figure 2-03. Internal Data Flow Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3. Transparent Operation

102

RF modules that contain the following firmware versions will support Transparent Mode: 1.0xx (coordinator) and 1.2xx (router/end device).

When operating in Transparent Mode, modules are configured using AT Commands and API operation is not supported. The modules act as a serial line replacement - all UART data received through the DIN pin is queued up for RF transmission. Data is sent to a module as defined by the DH (Destination Address High) and DL (Destination Address Low) parameters.

When RF data is received by a module, the data is sent out the DOUT pin.

Serial-to-RF Packetization

Data is buffered in the serial receive buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.
2. Maximum number of characters that will fit (72) in an RF packet is received.
3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the serial receive buffer before the sequence is transmitted.

2.1.4. API Operation

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. RF modules that contain the following firmware versions will support API operation: 1.1xx (coordinator) and 1.3xx (router/end device).

When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DIN pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DOUT pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate, etc.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, and payload information from received data packets.

The API operation option facilitates many operations such as the examples cited below:

- > Transmitting data to multiple destinations without entering Command Mode
- > Receive success/failure status of each transmitted RF packet
- > Identify the source address of each received packet

To implement API operations, refer to the API Operation chapter 6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2. Modes of Operation

103

2.2.1. Idle Mode

When not receiving or transmitting data, the RF module is in Idle Mode. During Idle Mode, the RF module is also checking for valid RF data. The module shifts into the other modes of operation under the following conditions:

- Transmit Mode (Serial data in the serial receive buffer is ready to be packetized)
- Receive Mode (Valid RF data is received through the antenna)
- Sleep Mode (End Devices only)
- Command Mode (Command Mode Sequence is issued)

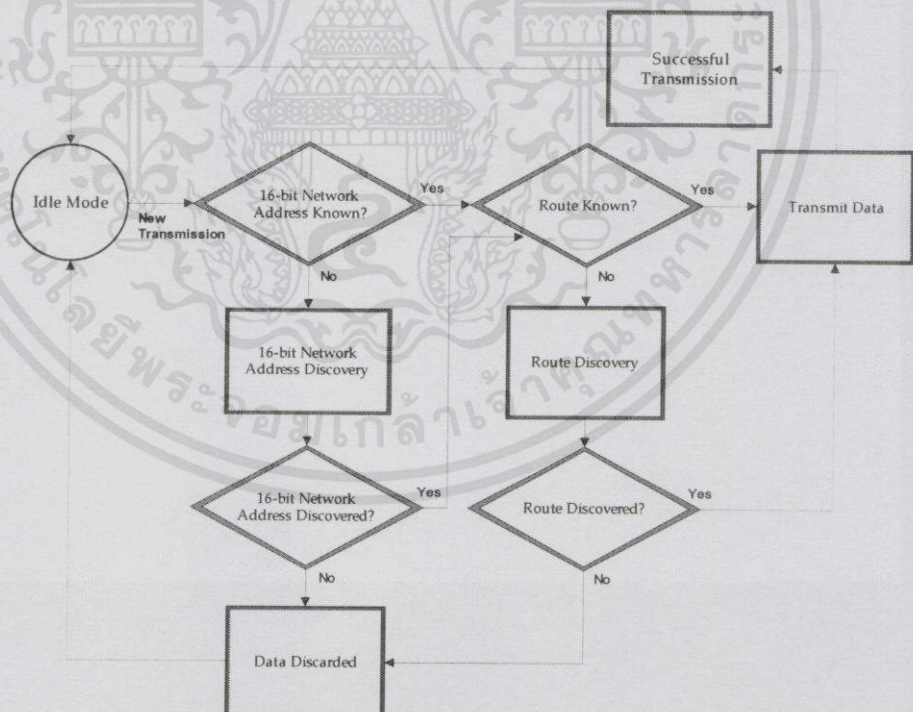
2.2.2. Transmit Mode

When serial data is received and is ready for packetization, the RF module will exit Idle Mode and attempt to transmit the data. The destination address determines which node(s) will receive the data.

Prior to transmitting the data, the module ensures that a 16-bit Network Address and route to the destination node have been established.

If the 16-bit Network Address is not known, Network Address Discovery will take place. If a route is not known, route discovery will take place for the purpose of establishing a route to the destination node. If a module with a matching Network Address is not discovered, the packet is discarded. The data will be transmitted once a route is established. If route discovery fails to establish a route, the packet will be discarded.

Figure 2-04. Transmit Mode Sequence



When data is transmitted from one node to another, a network-level acknowledgement is transmitted back across the established route to the source node. This acknowledgement packet indicates to the source node that the data packet was received by the destination node. If a network acknowledgement is not received, the source node will re-transmit the data. See Data Transmission and Routing in chapter 3 for more information.

2.2.3. Receive Mode

104

If a valid RF packet is received and its address matches the RF module's MY (16-bit Source Address) parameter, the data is transferred to the serial transmit buffer.

2.2.4. Command Mode

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming serial characters are interpreted as commands. Refer to the API Mode section for an alternate means of configuring modules.

AT Command Mode

To Enter AT Command Mode:

Send the 3-character command sequence "+++" and observe guard times before and after the command characters. [Refer to the "Default AT Command Mode Sequence" below.]

Default AT Command Mode Sequence (for transition to Command Mode):

- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters ("+++") within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

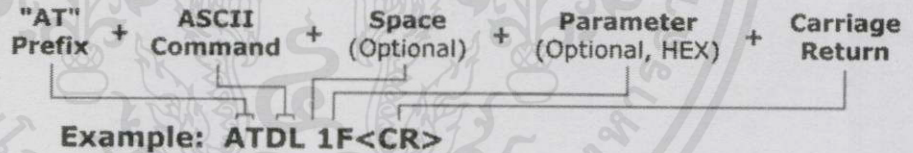
All of the parameter values in the sequence can be modified to reflect user preferences.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. Ensure the 'Baud' setting on the "PC Settings" tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (9600 bps).

To Send AT Commands:

Send AT commands and parameters using the syntax shown below.

Figure 2-05. Syntax for sending AT Commands



To read a parameter value stored in the RF module's register, omit the parameter field.

The preceding example would change the RF module Destination Address (Low) to "0x1F". To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

For modified parameter values to persist in the module's registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset.

System Response. When a command is sent to the module, the module will parse and execute the command. Upon successful execution of a command, the module returns an "OK" message. If execution of a command results in an error, the module returns an "ERROR" message.

To Exit AT Command Mode:

1. Send the ATCN (Exit Command Mode) command (followed by a carriage return).
[OR]
2. If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the RF module automatically returns to Idle Mode.

For an example of programming the RF module using AT Commands and descriptions of each configurable parameter, refer to the "RF Module Configuration" chapter. 105

2.2.5. Sleep Mode

Sleep modes are supported on end devices only. Router and coordinator devices participate in routing data packets and are intended to be mains powered. End devices must be joined to a parent (router or coordinator) before they can participate on a ZigBee network. The parent device does not track when an end device is awake or asleep. Instead, the end device must inform the parent when it is able to receive data. The parent must be able to buffer incoming data packets destined for the end device until the end device can awake and receive the data. When an end device is able to receive data, it sends a poll command to the parent. When the parent router or coordinator receives the poll command, it will transmit any buffered data packets for the end device. Routers and coordinators are capable of buffering one broadcast transmission for sleeping end device children.

The SM, ST, SP, and SN commands are used to configure sleep mode operation.

Data Management

The SP command on the parent determines how long the parent will buffer a packet. It should be set to match the maximum SP value on any end device that may join to it. SP can be set up to 28 seconds (0xAF0).

End Device Sleep Modes

Pin Sleep

Setting SM=1 or SM=2 configures a device as a pin-sleep enabled end device. When operating in this mode, an end device monitors the Sleep_Request pin for a high state. When Sleep_Request goes high, the module enters sleep mode once any pending transmissions have finished. The module remains in a low power state until the Sleep_Request pin goes low.

When the module wakes from pin sleep, it sends a poll request to the parent to see if any data is pending for the end device. Since routers and coordinators can only buffer data up to 30 seconds, end devices should not remain in pin sleep longer than about 28 seconds if incoming data packets must be received. Using pin sleep for more than 28 seconds is recommended only if incoming data packets are not expected.

When the module wakes from a pin sleep mode, the CTS line goes low, and On/Sleep goes high.

Cyclic Sleep

Cyclic sleep allows the end device to sleep for a specified period of time. The period of time is specified by SP. Since routers and coordinators can only buffer data packets for up to 30 seconds, SP on end devices can be set up to 28 seconds (0xAF0). The module will wake after SP time and send a poll request to the parent to check for data. If any serial or RF data is received, the ST time is restarted. Once ST time has expired with no serial or RF activity, the end device will resume cyclic sleep operation.

When the module wakes, CTS goes low allowing the application to send serial data to the module if necessary. The On/Sleep indicator will be set high to alert the application that the end device has awakened. If serial or RF data is received, the ST timer will be reset, otherwise, the end device will resume low power operation.

Off board peripherals may wish to sleep longer than the maximum SP time of the end device. The SN command can be used to not wake off board peripherals for longer than SP time.

For example, if SP=28 seconds, and SN=5, the end device will wake up every 28 seconds and poll the parent for data. If no data is pending, the end device will return to sleep. In this example, if the parent has no data for the end device, On/Sleep will go high after 140 seconds, assuming the parent never has data for the end device. If the parent has data for the end device, On/Sleep will go high and the SN counter will reset.

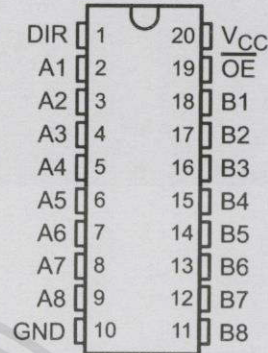
SN54LS245, SN74LS245 OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

SDLS146A – OCTOBER 1976 – REVISED FEBRUARY 2002

- 3-State Outputs Drive Bus Lines Directly
- PNP Inputs Reduce dc Loading on Bus Lines
- Hysteresis at Bus Inputs Improves Noise Margins
- Typical Propagation Delay Times Port to Port, 8 ns

TYPE	I _{OL} (SINK CURRENT)	I _{OH} (SOURCE CURRENT)
SN54LS245	12 mA	-12 mA
SN74LS245	24 mA	-15 mA

SN54LS245 . . . J OR W PACKAGE
SN74LS245 . . . DB, DW, N, OR NS PACKAGE
(TOP VIEW)

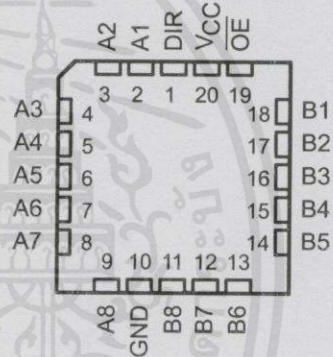


description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control-function implementation minimizes external timing requirements.

The devices allow data transmission from the A bus to the B bus or from the B bus to the A bus, depending on the logic level at the direction-control (DIR) input. The output-enable (OE) input can disable the device so that the buses are effectively isolated.

SN54LS245 . . . FK PACKAGE
(TOP VIEW)



ORDERING INFORMATION

T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP – N	Tube	SN74LS245N	SN74LS245N
	SOIC – DW	Tube	SN74LS245DW	LS245
		Tape and reel	SN74LS245DWR	
	SOP – NS	Tape and reel	SN74LS245NSR	74LS245
SSOP – DB	Tape and reel	SN74LS245DBR	LS245	
-55°C to 125°C	CDIP – J	Tube	SN54LS245J	SN54LS245J
		Tube	SNJ54LS245J	SNJ54LS245J
	CFP – W	Tube	SNJ54LS245W	SNJ54LS245W
	LCCC – FK	Tube	SN54LS245FK	SN54LS245FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

 **TEXAS
INSTRUMENTS**

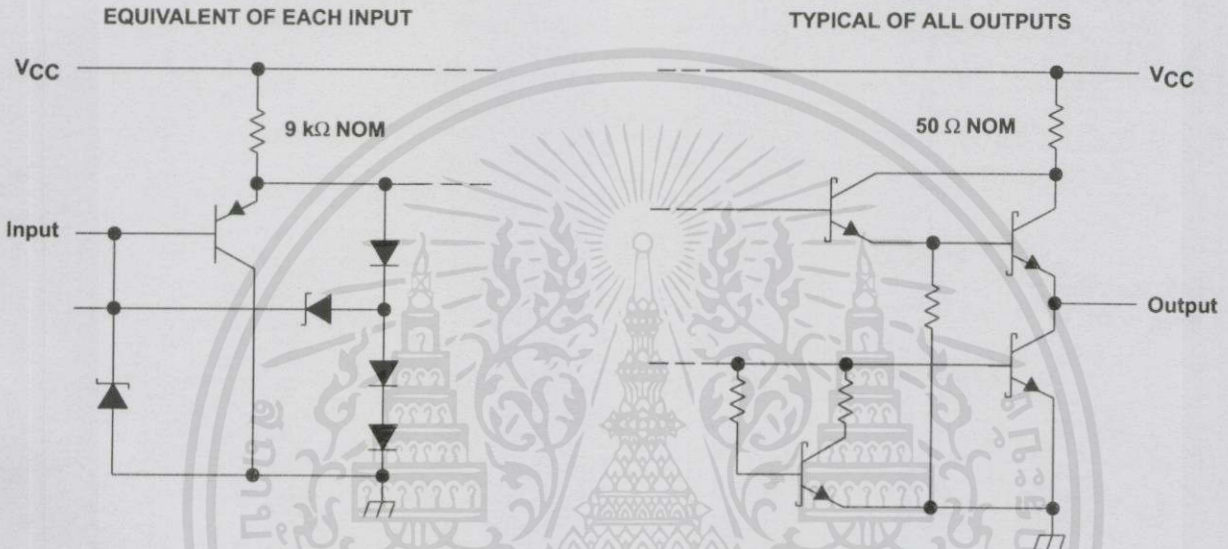
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2002, Texas Instruments Incorporated
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

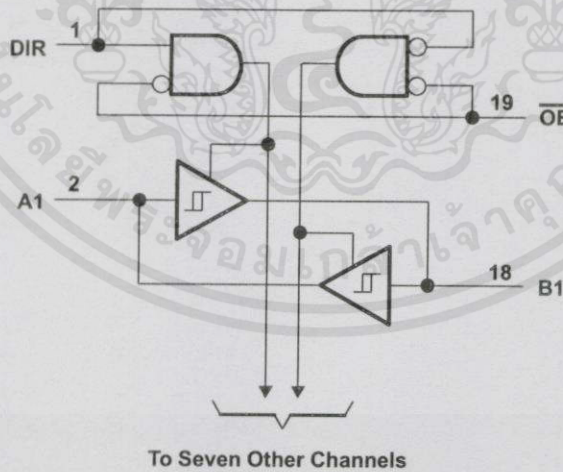
FUNCTION TABLE

INPUTS		OPERATION
\overline{OE}	DIR	
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

schematics of inputs and outputs



logic diagram (positive logic)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54LS245, SN74LS245 OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

SDLS146A – OCTOBER 1976 – REVISED FEBRUARY 2002

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC}	7 V
Input voltage, V_I (see Note 1)	7 V
Package thermal impedance, θ_{JA} (see Note 2):	
DB package	70°C/W
DW package	58°C/W
N package	69°C/W
NS package	60°C/W
Storage temperature range, T_{stg}	-65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. All voltage values are with respect to GND.
2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

		SN54LS245			SN74LS245			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I_{OH}	High-level output current			-12			-15	mA
I_{OL}	Low-level output current			12			24	mA
T_A	Operating free-air temperature	-55		125	0		70	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54LS245, SN74LS245 OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

SDLS146A - OCTOBER 1976 - REVISED FEBRUARY 2002

109

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†	SN54LS245			SN74LS245			UNIT	
			MIN	TYP‡	MAX	MIN	TYP‡	MAX		
V _{IH}	High-level input voltage		2			2			V	
V _{IL}	Low-level input voltage				0.7			0.8	V	
V _{IK}	Input clamp voltage	V _{CC} = MIN, I _I = -18 mA			-1.5			-1.5	V	
Hysteresis (V _{T+} - V _{T-})		A or B V _{CC} = MIN	0.2	0.4		0.2	0.4		V	
V _{OH}	High-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{IL(max)}	I _{OH} = -3 mA	2.4	3.4		2.4	3.4	V	
			I _{OH} = MAX	2			2			
V _{OL}	Low-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{IL(max)}	I _{OL} = 12 mA					0.4	V	
			I _{OL} = 24 mA					0.5		
I _{OZH}	Off-state output current, high-level voltage applied	V _{CC} = MAX, OE at 2 V			20			20	μA	
I _{OZL}	Off-state output current, low-level voltage applied	V _{CC} = MAX, OE at 2 V			-200			-200	μA	
I _I	Input current at maximum input voltage	A or B DIR or OE V _{CC} = MAX	V _I = 5.5 V			0.1			0.1	mA
			V _I = 7 V			0.1			0.1	
I _{IH}	High-level input current	V _{CC} = MAX, V _{IH} = 2.7 V			20			20	μA	
I _{IL}	Low-level input current	V _{CC} = MAX, V _{IL} = 0.4 V			-0.2			-0.2	mA	
I _{OS}	Short-circuit output current§	V _{CC} = MAX	-40		-225	40		-225	mA	
I _{CC}	Supply current	V _{CC} = MAX Outputs open	Total, outputs high		48	70		48	70	mA
			Total, outputs low		62	90		62	90	
			Outputs at high Z		64	95		64	95	

† For conditions shown as MIN or MAX, use the appropriate values specified under recommended operating conditions.

‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.

§ Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Figure 1)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t _{PLH}	Propagation delay time, low- to high-level output	C _L = 45 pF, R _L = 667 Ω			8	12	ns
t _{PHL}	Propagation delay time, high- to low-level output				8	12	
t _{PZL}	Output enable time to low level	C _L = 45 pF, R _L = 667 Ω			27	40	ns
t _{PZH}	Output enable time to high level				25	40	
t _{PLZ}	Output disable time from low level	C _L = 5 pF, R _L = 667 Ω			15	25	ns
t _{PHZ}	Output disable time from high level				15	28	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC78XX/LM78XX/MC78XXA

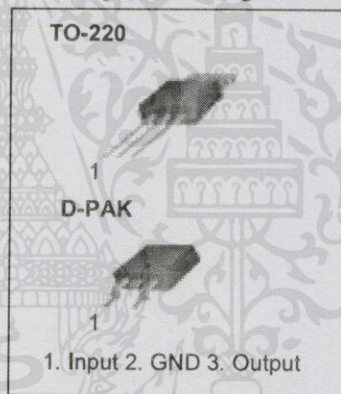
3-Terminal 1A Positive Voltage Regulator

Features

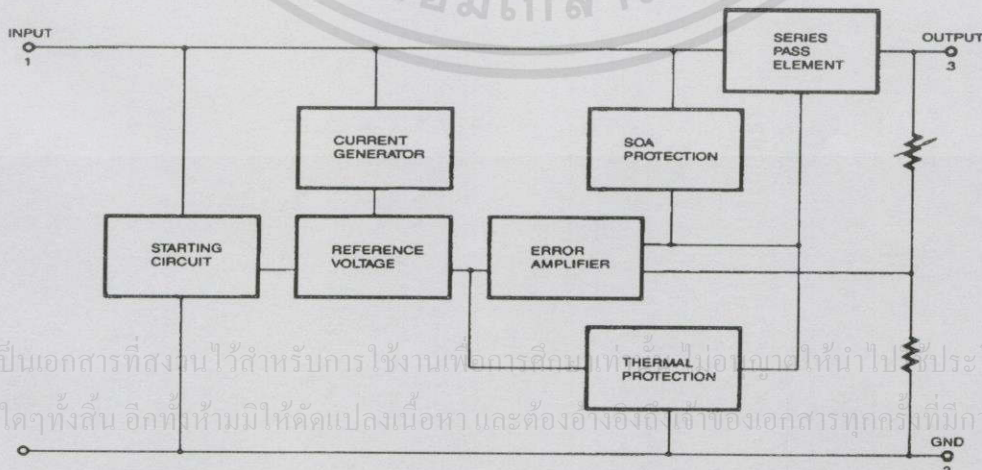
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการสืบค้นเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for $V_O = 5V$ to $18V$) (for $V_O = 24V$)	V_I	35	V
	V_I	40	V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^{\circ}C/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^{\circ}C/W$
Operating Temperature Range	T_{OPR}	0 ~ +125	$^{\circ}C$
Storage Temperature Range	T_{STG}	-65 ~ +150	$^{\circ}C$

Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit, $0^{\circ}C < T_J < 125^{\circ}C$, $I_O = 500mA$, $V_I = 10V$, $C_I = 0.33\mu F$, $C_O = 0.1\mu F$, unless otherwise specified)

Parameter	Symbol	Conditions	MC7805/LM7805			Unit	
			Min.	Typ.	Max.		
Output Voltage	V_O	$T_J = +25^{\circ}C$	4.8	5.0	5.2	V	
		$5.0mA \leq I_O \leq 1.0A$, $P_O \leq 15W$ $V_I = 7V$ to $20V$	4.75	5.0	5.25		
Line Regulation (Note1)	Regline	$T_J = +25^{\circ}C$	$V_O = 7V$ to $25V$	-	4.0	100	mV
			$V_I = 8V$ to $12V$	-	1.6	50	
Load Regulation (Note1)	Regload	$T_J = +25^{\circ}C$	$I_O = 5.0mA$ to $1.5A$	-	9	100	mV
			$I_O = 250mA$ to $750mA$	-	4	50	
Quiescent Current	I_Q	$T_J = +25^{\circ}C$	-	5.0	8.0	mA	
Quiescent Current Change	ΔI_Q	$I_O = 5mA$ to $1.0A$	-	0.03	0.5	mA	
		$V_I = 7V$ to $25V$	-	0.3	1.3		
Output Voltage Drift	$\Delta V_O / \Delta T$	$I_O = 5mA$	-	-0.8	-	mV/ $^{\circ}C$	
Output Noise Voltage	V_N	$f = 10Hz$ to $100KHz$, $T_A = +25^{\circ}C$	-	42	-	$\mu V/V_O$	
Ripple Rejection	RR	$f = 120Hz$ $V_O = 8V$ to $18V$	62	73	-	dB	
Dropout Voltage	V_{Drop}	$I_O = 1A$, $T_J = +25^{\circ}C$	-	2	-	V	
Output Resistance	r_O	$f = 1KHz$	-	15	-	m Ω	
Short Circuit Current	I_{SC}	$V_I = 35V$, $T_A = +25^{\circ}C$	-	230	-	mA	
Peak Current	I_{PK}	$T_J = +25^{\circ}C$	-	2.2	-	A	

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LM1117

800mA Low-Dropout Linear Regulator

General Description

The LM1117 is a series of low dropout voltage regulators with a dropout of 1.2V at 800mA of load current. It has the same pin-out as National Semiconductor's industry standard LM317.

The LM1117 is available in an adjustable version, which can set the output voltage from 1.25V to 13.8V with only two external resistors. In addition, it is also available in five fixed voltages, 1.8V, 2.5V, 2.85V, 3.3V, and 5V.

The LM1117 offers current limiting and thermal shutdown. Its circuit includes a zener trimmed bandgap reference to assure output voltage accuracy to within $\pm 1\%$.

The LM1117 series is available in SOT-223, TO-220, and TO-252 D-PAK packages. A minimum of 10 μ F tantalum capacitor is required at the output to improve the transient response and stability.

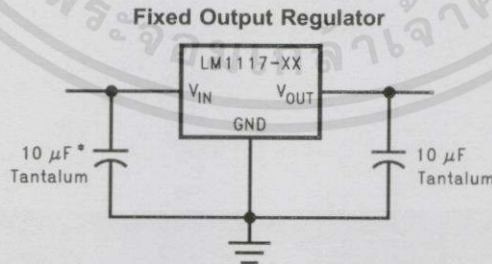
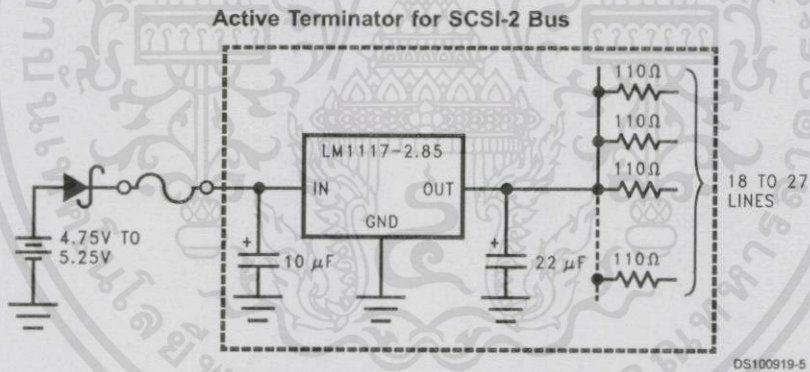
Features

- Available in 1.8V, 2.5V, 2.85V, 3.3V, 5V, and Adjustable Versions
- Space Saving SOT-223 Package
- Current Limiting and Thermal Protection
- Output Current 800mA
- Temperature Range 0°C to 125°C
- Line Regulation 0.2% (Max)
- Load Regulation 0.4% (Max)

Applications

- 2.85V Model for SCSI-2 Active Termination
- Post Regulator for Switching DC/DC Converter
- High Efficiency Linear Regulators
- Battery Charger
- Battery Powered Instrumentation

Typical Application



* Required if the regulator is located far from the power supply filter.
DS100919-28

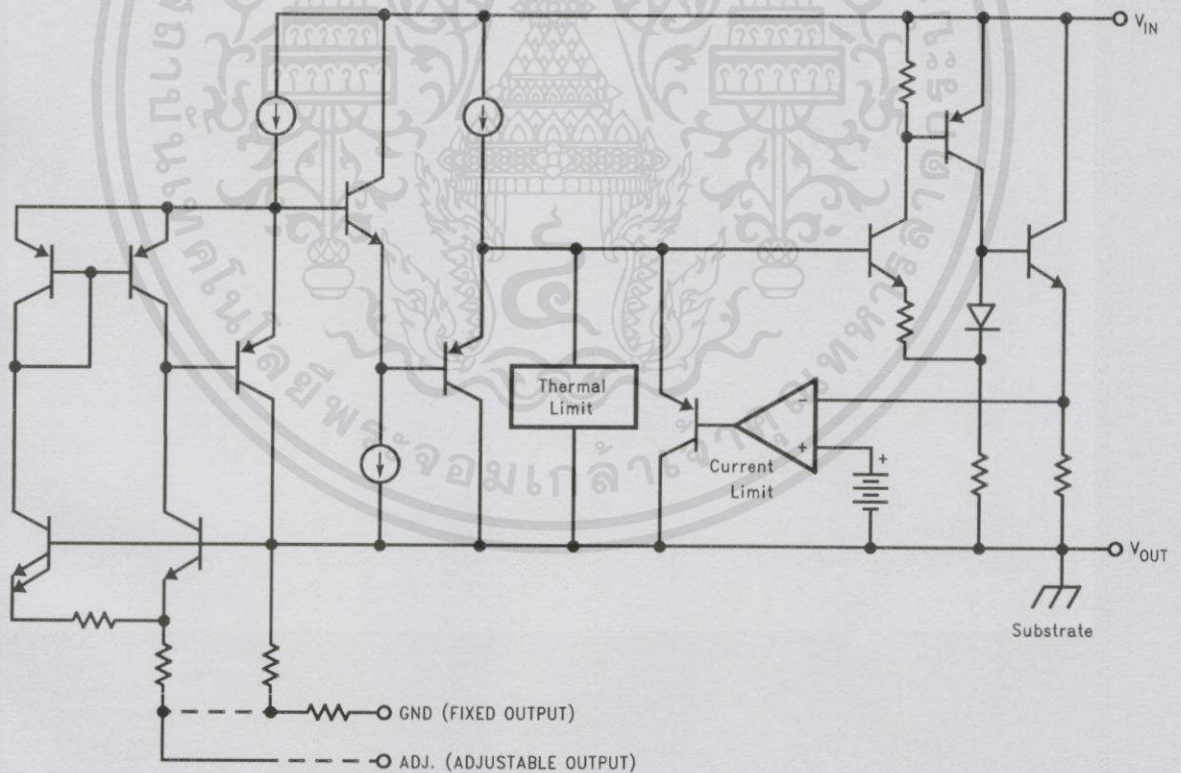
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Package	Temperature Range	Packaging Marking	Transport Media	113
	0°C to +125°C			NSC Drawing
3-lead SOT-223	LM1117MPX-ADJ	N03A	Tape and Reel	MA04A
	LM1117MPX-1.8	N12A	Tape and Reel	
	* LM1117MPX-2.5	N13A	Tape and Reel	
	LM1117MPX-2.85	N04A	Tape and Reel	
	LM1117MPX-3.3	N05A	Tape and Reel	
	LM1117MPX-5.0	N06A	Tape and Reel	
3-lead TO-220	LM1117T-ADJ	LM1117T-ADJ	Rails	T03B
	LM1117T-2.85	LM1117T-2.85	Rails	
	LM1117T-3.3	LM1117T-3.3	Rails	
	LM1117T-5.0	LM1117T-5.0	Rails	
3-lead TO-252	LM1117DTX-ADJ	LM1117DT-ADJ	Tape and Reel	TD03B
	LM1117DTX-1.8	LM1117DT-1.8	Tape and Reel	
	* LM1117DTX-2.5	LM1117DT-2.5	Tape and Reel	
	LM1117DTX-2.85	LM1117DT-2.85	Tape and Reel	
	LM1117DTX-3.3	LM1117DT-3.3	Tape and Reel	
	LM1117DTX-5.0	LM1117DT-5.0	Tape and Reel	

Note: * Contact factory for availability.

Block Diagram



DS100919-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Maximum Input Voltage (V_{IN} to GND)		20V
Power Dissipation (Note 2)	Internally Limited	
Junction Temperature (T_J) (Note 2)		150°C
Storage Temperature Range		-65°C to 150°C
Lead Temperature		

TO-220 (T) Package	260°C, 10 sec
SOT-223 (IMP) Package	260°C, 4 sec
ESD Tolerance (Note 3)	2000V

Operating Ratings (Note 1)

Input Voltage (V_{IN} to GND)		15V
Junction Temperature Range (T_J)(Note 2)		0°C to 125°C

Electrical Characteristics

Typicals and limits appearing in normal type apply for $T_J = 25^\circ\text{C}$. Limits appearing in **Boldface** type apply over the entire junction temperature range for operation, 0°C to 125°C.

Symbol	Parameter	Conditions	Min (Note 5)	Typ (Note 4)	Max (Note 5)	Units
V_{REF}	Reference Voltage	LM1117-ADJ $I_{OUT} = 10\text{mA}$, $V_{IN}-V_{OUT} = 2\text{V}$, $T_J = 25^\circ\text{C}$	1.238	1.250	1.262	V
		$10\text{mA} \leq I_{OUT} \leq 800\text{mA}$, $1.4\text{V} \leq V_{IN}-V_{OUT} \leq 10\text{V}$	1.225	1.250	1.270	V
V_{OUT}	Output Voltage	LM1117-1.8 $I_{OUT} = 10\text{mA}$, $V_{IN} = 3.8\text{V}$, $T_J = 25^\circ\text{C}$	1.782	1.800	1.818	V
		$0 \leq I_{OUT} \leq 800\text{mA}$, $3.2\text{V} \leq V_{IN} \leq 10\text{V}$	1.746	1.800	1.854	V
		LM1117-2.5 $I_{OUT} = 10\text{mA}$, $V_{IN} = 4.5\text{V}$, $T_J = 25^\circ\text{C}$	2.475	2.500	2.525	V
		$0 \leq I_{OUT} \leq 800\text{mA}$, $3.9\text{V} \leq V_{IN} \leq 10\text{V}$	2.450	2.500	2.550	V
		LM1117-2.85 $I_{OUT} = 10\text{mA}$, $V_{IN} = 4.85\text{V}$, $T_J = 25^\circ\text{C}$	2.820	2.850	2.880	V
		$0 \leq I_{OUT} \leq 800\text{mA}$, $4.25\text{V} \leq V_{IN} \leq 10\text{V}$	2.790	2.850	2.910	V
		$0 \leq I_{OUT} \leq 500\text{mA}$, $V_{IN} = 4.10\text{V}$	2.790	2.850	2.910	V
ΔV_{OUT}	Line Regulation (Note 6)	LM1117-3.3 $I_{OUT} = 10\text{mA}$, $V_{IN} = 5\text{V}$, $T_J = 25^\circ\text{C}$	3.267	3.300	3.333	V
		$0 \leq I_{OUT} \leq 800\text{mA}$, $4.75\text{V} \leq V_{IN} \leq 10\text{V}$	3.235	3.300	3.365	V
		LM1117-5.0 $I_{OUT} = 10\text{mA}$, $V_{IN} = 7\text{V}$, $T_J = 25^\circ\text{C}$	4.950	5.000	5.050	V
		$0 \leq I_{OUT} \leq 800\text{mA}$, $6.5\text{V} \leq V_{IN} \leq 12\text{V}$	4.900	5.000	5.100	V
		LM1117-ADJ $I_{OUT} = 10\text{mA}$, $1.5\text{V} \leq V_{IN}-V_{OUT} \leq 13.75\text{V}$		0.035	0.2	%
		LM1117-1.8 $I_{OUT} = 0\text{mA}$, $3.2\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV
		LM1117-2.5 $I_{OUT} = 0\text{mA}$, $3.9\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV
		LM1117-2.85 $I_{OUT} = 0\text{mA}$, $4.25\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV
		LM1117-3.3 $I_{OUT} = 0\text{mA}$, $4.75\text{V} \leq V_{IN} \leq 15\text{V}$		1	6	mV
		LM1117-5.0 $I_{OUT} = 0\text{mA}$, $6.5\text{V} \leq V_{IN} \leq 15\text{V}$		1	10	mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์อื่นนอกเหนือจากนี้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Continued)

Typicals and limits appearing in normal type apply for $T_J = 25^\circ\text{C}$. Limits appearing in **Boldface** type apply over the entire junction temperature range for operation, 0°C to 125°C .

Symbol	Parameter	Conditions	Min (Note 5)	Typ (Note 4)	Max (Note 5)	Units
ΔV_{OUT}	Load Regulation (Note 6)	LM1117-ADJ $V_{\text{IN}} - V_{\text{OUT}} = 3\text{V}$, $10 \leq I_{\text{OUT}} \leq 800\text{mA}$		0.2	0.4	%
		LM1117-1.8 $V_{\text{IN}} = 3.2\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-2.5 $V_{\text{IN}} = 3.9\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-2.85 $V_{\text{IN}} = 4.25\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-3.3 $V_{\text{IN}} = 4.75\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-5.0 $V_{\text{IN}} = 6.5\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	15	mV
$V_{\text{IN}} - V_{\text{OUT}}$	Dropout Voltage (Note 7)	$I_{\text{OUT}} = 100\text{mA}$		1.10	1.20	V
		$I_{\text{OUT}} = 500\text{mA}$		1.15	1.25	V
		$I_{\text{OUT}} = 800\text{mA}$		1.20	1.30	V
I_{LIMIT}	Current Limit	$V_{\text{IN}} - V_{\text{OUT}} = 5\text{V}$, $T_J = 25^\circ\text{C}$	800	1200	1500	mA
	Minimum Load Current (Note 8)	LM1117-ADJ $V_{\text{IN}} = 15\text{V}$		1.7	5	mA
	Quiescent Current	LM1117-1.8 $V_{\text{IN}} \leq 15\text{V}$		5	10	mA
		LM1117-2.5 $V_{\text{IN}} \leq 15\text{V}$		5	10	mA
		LM1117-2.85 $V_{\text{IN}} \leq 10\text{V}$		5	10	mA
		LM1117-3.3 $V_{\text{IN}} \leq 15\text{V}$		5	10	mA
		LM1117-5.0 $V_{\text{IN}} \leq 15\text{V}$		5	10	mA
		Thermal Regulation	$T_A = 25^\circ\text{C}$, 30ms Pulse		0.01	0.1
	Ripple Regulation	$f_{\text{RIPPLE}} = 1\text{ kHz}$, $V_{\text{IN}} - V_{\text{OUT}} = 3\text{V}$ $V_{\text{RIPPLE}} = 1V_{\text{PP}}$	60	75		dB
	Adjust Pin Current			60	120	μA
	Adjust Pin Current Change	$10 \leq I_{\text{OUT}} \leq 800\text{mA}$, $1.4\text{V} \leq V_{\text{IN}} - V_{\text{OUT}} \leq 10\text{V}$		0.2	5	μA
	Temperature Stability			0.5		%
	Long Term Stability	$T_A = 125^\circ\text{C}$, 1000Hrs		0.3		%
	RMS Output Noise	(% of V_{OUT}), $10\text{Hz} \leq f \leq 10\text{kHz}$		0.003		%
	Thermal Resistance Junction-to-Case	3-Lead SOT-223		15.0		$^\circ\text{C/W}$
		3-Lead TO-220		3.0		$^\circ\text{C/W}$
		3-Lead TO-252		10		$^\circ\text{C/W}$
	Thermal Resistance Junction-to-Ambient (No heat sink; No air flow)	3-Lead SOT-223		136		$^\circ\text{C/W}$
		3-Lead TO-220		79		$^\circ\text{C/W}$
		3-Lead TO-252 (Note 9)		92		$^\circ\text{C/W}$

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but specific performance is not guaranteed. For guaranteed specifications and the test conditions, see the Electrical Characteristics.

Note 2: The maximum power dissipation is a function of $T_{J(\text{max})}$, θ_{JA} , and T_A . The maximum allowable power dissipation at any ambient temperature is $P_D = (T_{J(\text{max})} - T_A) / \theta_{\text{JA}}$. All numbers apply for packages soldered directly into a PC board.

Note 3: For testing purposes, ESD was applied using human body model, $1.5\text{k}\Omega$ in series with 100pF .

Note 4: Typical Values represent the most likely parametric norm.

Note 5: All limits are guaranteed by testing or statistical analysis.

Note 6: Load and line regulation are measured at constant junction room temperature.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Single-phase Electronic Meter

SX1-A31N

AMR Protocol Specifications

SPEC. NO. : **MED-A0028A**

Date : February 8, 2012

Changed : Rev.A, November 1st, 2012

Drawn : K. Nuttapol

Checked : *K. Nuttapol*Approved : *T. Sato*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SX1 PROTOCOL SPECIFICATION

The communications messages are consist of two protocol. First is RS-485 protocol, its cover whole of message and the second is inner protocol we named MEATH protocol. Protocol layers are show in figure 1.1-1 and specifications of each are describe in following sections.

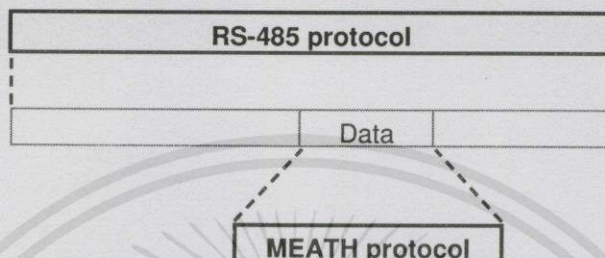


Figure 1.1-1 Protocol stack layer

1. RS-485 Protocol

1.1 Specifications

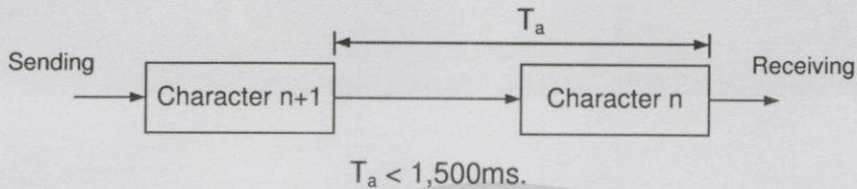
Table 1.1-1 RS-485 communication and RS-485 protocol specifications

Type of transmission	Asynchronous serial bit, Half duplex										
Physical interface	RS – 485 (2wires) line driver.										
Transmission speed	19,200 bps.										
Protocol standard	RS-485 Protocol										
Technique	Single Master / Multi Slaves										
Packet size	51 bytes (Fixed)										
Error detection	<p>CCITT CRC16 (cyclical redundancy check) Polynomial = 0x1021 Non-reflect algorithm (MSB first) Initial value – 0xFFFF Note : Testing value = “123456789” CRC value = 0x29B1 See annex A for more details</p>										
Data format	<p>Binary LSB first</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>ST</td> <td>B0</td> <td>B1</td> <td>B2</td> <td>B3</td> <td>B4</td> <td>B5</td> <td>B6</td> <td>B7</td> <td>SP</td> </tr> </table> <p>ST : Start bit (1bit / logic 0) SP : Stop bit (1bit / logic 1) B0 – B7 : Binary data (8bits)</p>	ST	B0	B1	B2	B3	B4	B5	B6	B7	SP
ST	B0	B1	B2	B3	B4	B5	B6	B7	SP		

1.2 Communication Timing

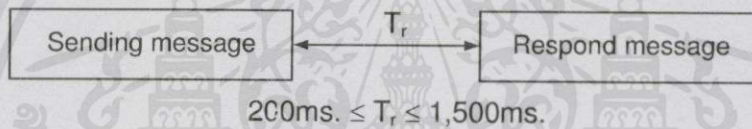
1.2.1 Time between each character

Every byte within a packet must precede the previous byte by take the time less than 1,500 ms. If time is more than, all of data will be flush out from the receive buffer.



1.2.2 Time between each packet (Response time)

The response time from a master request to a slave response or in case of sent data in multiple packet. Any new packet of data will be wait $200 \leq t_r \leq 1,500ms$ before send to receiver.



S/W target for response waiting time = 210 ms.

S/W target for response timeout = 1,500 ms.

S/W target for inter character timeout = 1,400 ms.

1.3 Protocol Structure

RS-485 message structure for every message type (Send message, Receive message, and Receive connect message) are the same, this show in figure 1.3-1.

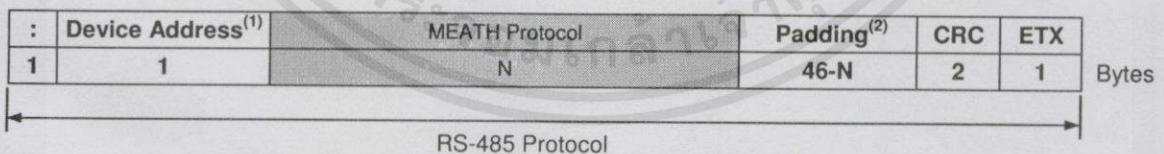


Figure 1.3-1 RS-485 message structure

⁽¹⁾Device address (RS-485 address)

Data size : 1 byte
 Data format : Binary
 Default device address : See annex B

⁽²⁾Padding

Data size : 46-N byte
 Data format : Fill in with # ASCII

Remark Special character ASCII code in communication message

: = 0x3A
= 0x23
ETX = 0x03

1.4 Protocol Analyze

For get data from received message, first analyze RS-485 protocol (outer protocol layer) to get MEATH protocol layer (inner protocol layer), and second analyze the MEATH protocol for get the target data. MEATH protocol will describe on section 2.

1.5 Error Detection

When CRC is incorrect, the meter will reject the command and will not reply to the master.

2. MEATH Protocol Specifications

2.1 Specifications

Table 2.1-1 MEATH protocol specifications

Protocol standard	MEATH Protocol (Custom)										
Error detection	Block Check Character (BCC) See annex A for more detail										
Data format	ASCII LSB first <table border="1" style="margin-left: 40px;"> <tr> <td>ST</td> <td>B0</td> <td>B1</td> <td>B2</td> <td>B3</td> <td>B4</td> <td>B5</td> <td>B6</td> <td>P</td> <td>SP</td> </tr> </table> <p style="margin-left: 40px;"> ST : Start bit (1 bit/ logic 0) SP : Stop bit (1 bit/ logic 1) P : Parity bit (1 bit/ even) B0 – B6: Character data (7 bit) </p>	ST	B0	B1	B2	B3	B4	B5	B6	P	SP
ST	B0	B1	B2	B3	B4	B5	B6	P	SP		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Communication Sequence

As figure 2.2-1 sequence of message is starts from connect message to establish the communication. Then will read target data (such as kWh energy, line voltage and current) and when all target data reading finished program will send disconnect command to terminate the communication.

Sequence of message

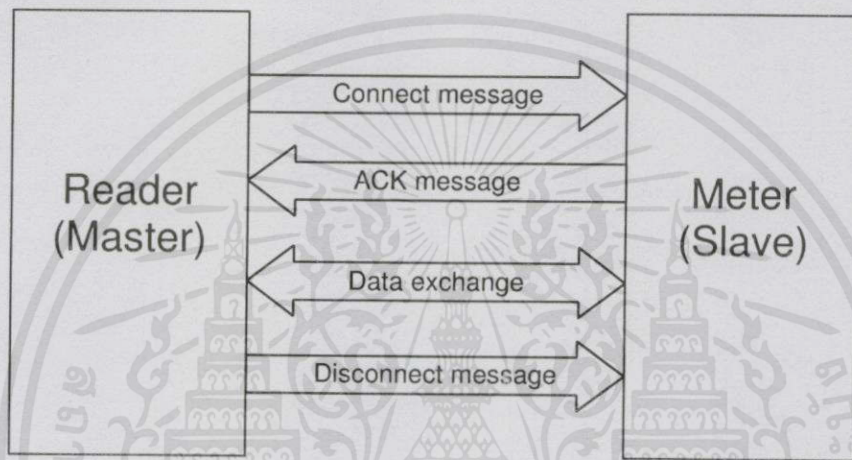


Figure 2.2-1 Message sequence

Sequence of data exchange

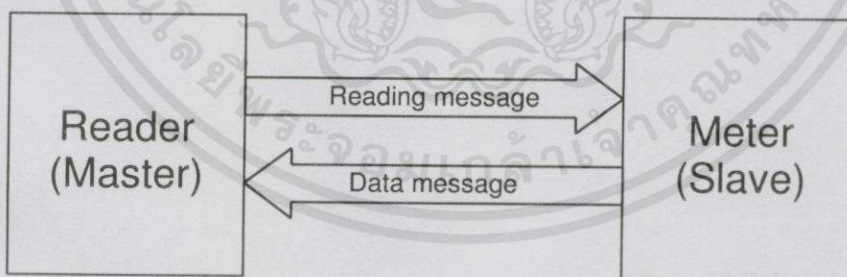


Figure 2.2-2 Data exchange sequence

The sequence of data exchange will start from the reader sends reading command message to the meter. After meter receives the command message it will send the data message to the reader. For reading command meter will send data message back one by one for each command.

2.3 Protocol Structure

MEATH protocol will start after “Device address” (“Device address” is on RS-485 protocol layer) and finish at “BCC”, this show in figure 2.3-1.

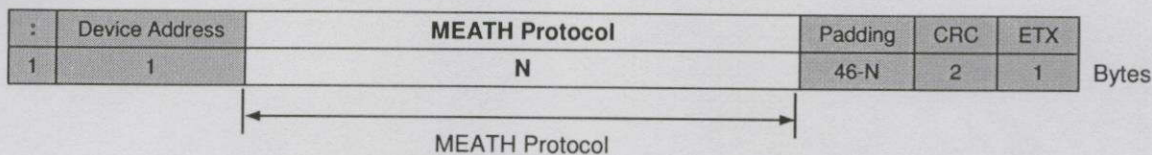


Figure 2.3-1 MEATH protocol structure

2.4 MEATH protocol message

2.4.1 Connect / Disconnect Message

2.4.1.1 Connect Message

Send : Connect message

SOH	P	1	STX	(Data)	ETX	BCC
-----	---	---	-----	---	------	---	-----	-----

Data size : 20 bytes

Data (Binary): D2 53 B4 B8 35 D4 D7 CF D7 C9 D2 C5 53 50 D2 CF CA C5 C3 D4

Receive : ACK message

ACK

Data size : 1 byte

Data (Binary) : 06

2.4.1.2 Disconnect message

Send : Disconnect message

SOH	B	0	ETX	BCC
-----	---	---	-----	-----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 Meter Information

2.4.2.1 Meter ID* reading

Send : Reading message

SOH	R	2	STX	0	0	()	ETX	BCC
-----	---	---	-----	---	---	-----	-----	-----

Receive : Data message

STX	0	0	(Data)	ETX	BCC
-----	---	---	---	------	---	-----	-----

Data size : 7 bytes

Data format : ASCII

Unit : -

*See annex B

2.4.3 Energy Value

2.4.3.1 kWh energy reading

Send : Reading message

SOH	R	2	STX	D	7	()	ETX	BCC
-----	---	---	-----	---	---	-----	-----	-----

Receive : Data message

STX	D	7	(Data)	ETX	BCC
-----	---	---	---	------	---	-----	-----

Data size : 9 bytes

Data format : ASCII

Unit : Wh

2.4.4 PQM Value

2.4.4.1 RMS line voltage reading

Send : Reading message

SOH	R	2	STX	D	0	()	ETX	BCC
-----	---	---	-----	---	---	-----	-----	-----

Receive : Data message

STX	D	0	(Data)	ETX	BCC
-----	---	---	---	------	---	-----	-----

Data size : 5 bytes

Data format : ASCII

เอกเป็นเอกสารที่ 10 mV ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4.2 RMS current reading

Send : Reading message

SOH	R	2	STX	D	2	()	ETX	BCC
-----	---	---	-----	---	---	-----	-----	-----

Receive : Data message

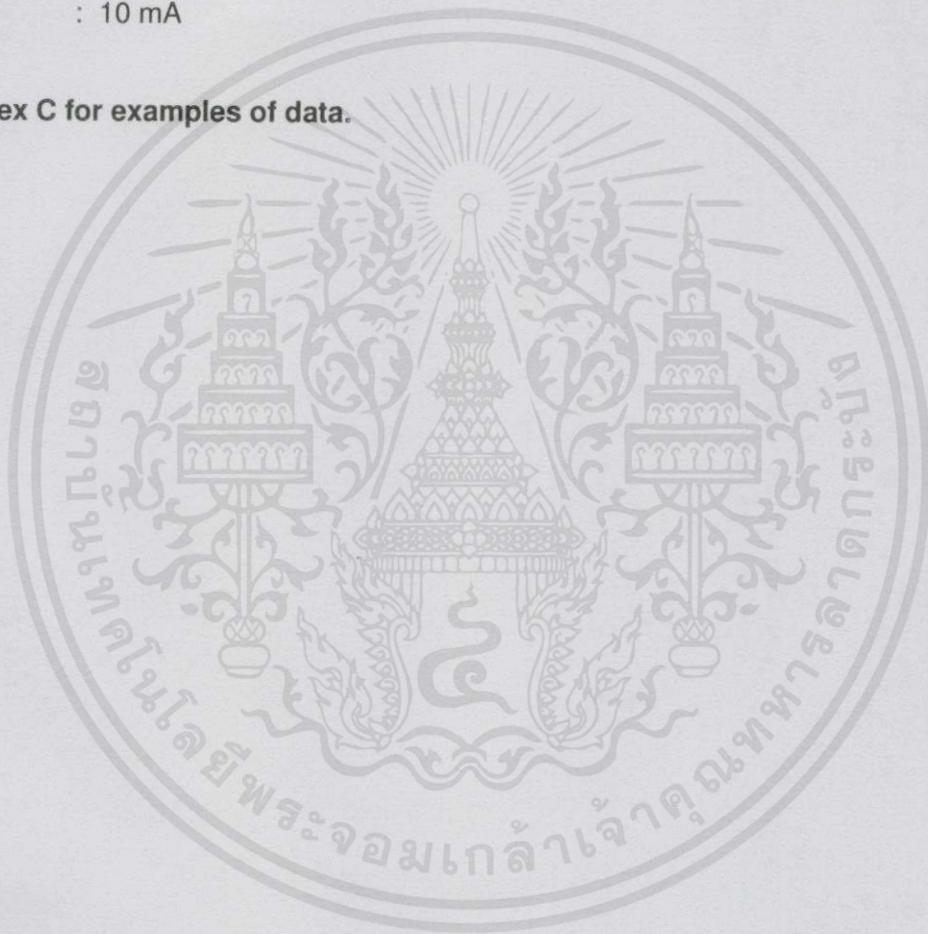
STX	D	2	(Data)	ETX	BCC
-----	---	---	---	------	---	-----	-----

Data size : 5 bytes

Data format : ASCII

Unit : 10 mA

See annex C for examples of data.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Annex A BCC and CRC calculation

1. Block check character (BCC) calculation

1) Rule for generating the longitudinal parity block check character

1.1) Block check character

1.1.1 The block check character shall be composed of 7 bits plus an even parity bit

1.1.2 Each of the first 7 bits of the block check character shall be the modulo 2 binary sum of every element in the same bit 0 to bit 6 column of the successive character of the transmitted block.

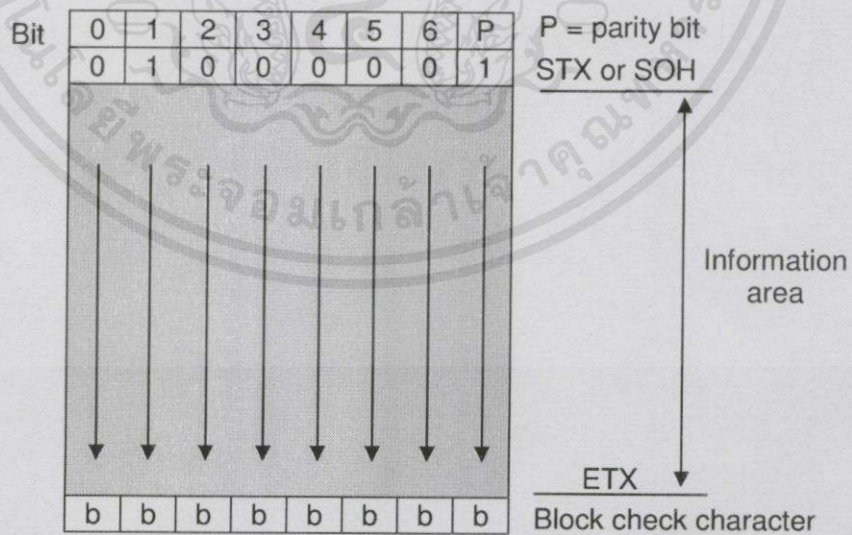
1.1.3 The longitudinal parity of each column of the block, including the block check character, shall be even.

1.2) Summation

1.2.1 The summation to obtain the block check character shall be started by the first appearance of either SOH (Start of Heading) or STX (Start of Text).

1.2.2 The starting character shall not be included in the summation.

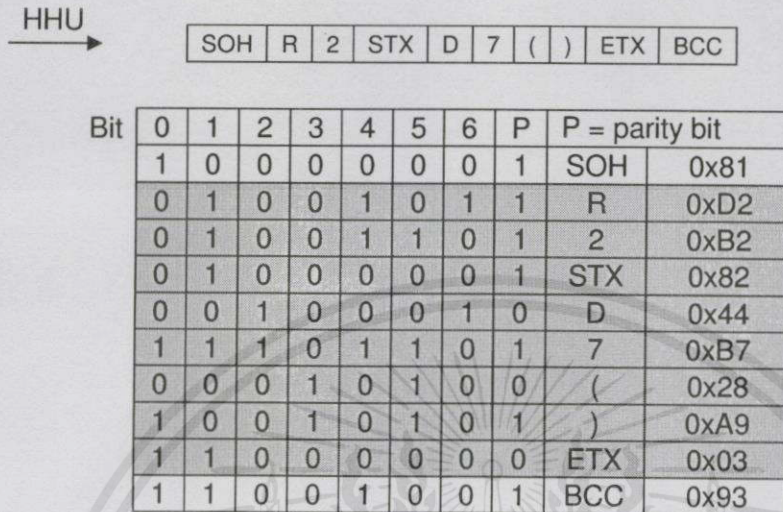
1.2.3 If an STX character appears after the summation has been started by SOH, the STX character shall be included in the summation as if it were a text character.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Example of block check character calculation

Assume that the read command message has the stream of character like this;



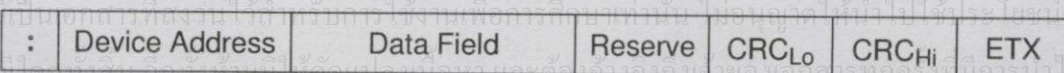
From the figure, the highlight area is the field of character for generating block check character. Each bit of block check character is the longitudinal even parity of each column of the block, including the block check character, shall be even. So from the example the block check character from the calculation will be (93h).

2. Cyclical Redundancy Check (CRC-16) calculation

The Cyclical Redundancy Check (CRC-16) field is two bytes, containing a 16-bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, that's means the communication error are occurs. Program will flush message and do not operate any operation and any response message.

Placing the CRC into the Message

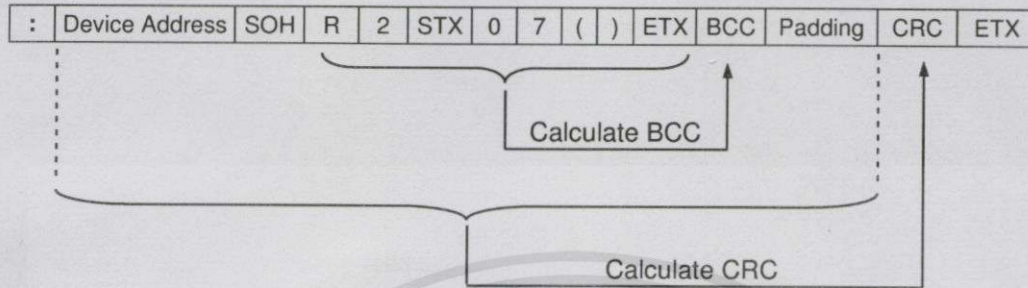
When the 16-bit CRC (two 8-bit bytes) is transmitted in the message, the low-order byte will be transmitted first, followed by the high-order byte. For example, if the CRC value is 1241 hex (0001 0010 0100 0001) the CRC value will be place into message as follow.



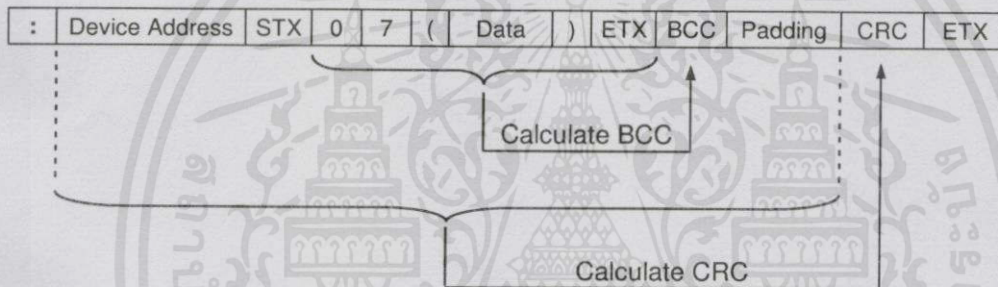
41 12

BCC and CRC calculation field.

Send



Receive



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Annex B

RS-485 Address

The meter RS-485 address assign as follows

- If 3rd digits from the last digit of ID no. that showing on nameplate is even number, the meter RS-485 address will be "*0xx*" by "*xx*" is the last 2 digits of ID no. If the last 2 digits of ID no. are "*00*", meter RS-485 address will be "*200*".

- If 3rd digits from the last digit of ID no. that showing on nameplate is odd number, the meter RS-485 address will be "*1xx*" by "*xx*" is the last 2 digits of ID no.

Example:

ID no.	3 rd digit from last digit of ID no.	RS-485 address
0275 <u>348</u>	<u>Odd</u>	<u>148</u>
0275 <u>448</u>	<u>Even</u>	<u>48</u>
0275 <u>548</u>	<u>Odd</u>	<u>148</u>
0276 <u>000</u>	<u>Even</u>	<u>200</u>
0276 <u>100</u>	<u>Odd</u>	<u>100</u>
0700 <u>030</u>	<u>Even</u>	<u>30</u>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน



นาย นพดล ปาลีภัย

เกิดเมื่อวันที่ 12 สิงหาคม 2535 สถานที่เกิดจังหวัดเชียงใหม่

สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนสันป่าตองวิทยาคม ปัจจุบันกำลัง

ศึกษาระดับปริญญาตรี ภาควิชาวิศวกรรมไฟฟ้าสถาบันเทคโนโลยีพระจอมเกล้าเจ้า

คุณทหารลาดกระบัง
Email: nopadon.paleekui@gmail.com



นางสาว นสณีย์ ยวงโย

เกิดเมื่อวันที่ 11 มีนาคม 2536 สถานที่เกิดจังหวัดกรุงเทพมหานคร

สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนปรีณสรอยแยลส์วิทยาลัย ปัจจุบัน

กำลังศึกษาระดับปริญญาตรี ภาควิชาวิศวกรรมไฟฟ้าสถาบันเทคโนโลยีพระจอม

เกล้าเจ้าคุณทหารลาดกระบัง
Email: ns.youngyai@gmail.com



นางสาว เบนจพร ก้อนผา

เกิดเมื่อวันที่ 11 กุมภาพันธ์ 2536 สถานที่เกิดจังหวัดกรุงเทพมหานคร

สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนสตรีศรีนครปฐมบำเพ็ญ ปัจจุบันกำลัง

ศึกษาระดับปริญญาตรี ภาควิชาวิศวกรรมไฟฟ้าสถาบันเทคโนโลยีพระจอมเกล้าเจ้า

คุณทหารลาดกระบัง
Email: benjaporn.konpha@gmail.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้