สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# การควบคุมอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10

# POWER LINE COMMUNICATION WITH X10

นางสาวพรพิสุทธิ์ นุกูลราษฎร์

นายพรรณเชษฐ์ พวงนาค

นายพลเทพ นามพระจันทร์

เลขหมู่.....
เลขทะเบียน 119192
วัน,เดือน,ปี - 6 ส.ค. 2554

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# POWER LINE COMMUNICATION WITH X10

**PORNPISUT NUKULRAD**

**PANNACHET POUNGNAK**

**PONTHEP NAMPRAJAN**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT**

**OF THE REQUIREMENT FOR THE DEGREE OF**

**BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING**

**FACULTY OF ENGINEERING**

**KING MONGKUT'S INSTITUE OF TECHNOLOGY LARDKRABANG**

**ACADEMIC YEAR 2010**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**สาขาวิชาวิศวกรรมการวัดคุม**

**คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ใบรับรองปริญญานิพนธ์**

.....

**หัวข้อปริญญานิพนธ์** การควบคุมอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10

POWER LINE COMMUNICATION WITH X10

**นักศึกษาผู้จัดทำ**

| | | |
|---|---|---|
| นางสาวพรพิสุทธิ์ นุกูลราษฎร์ | รหัสนักศึกษา | 50011052 |
| นายพรรณเชษฐ์ พวงนาค | รหัสนักศึกษา | 50011054 |
| นายพลเทพ นามพระจันทร์ | รหัสนักศึกษา | 50011062 |

**ปริญญา** วิศวกรรมศาสตรบัณฑิต

**สาขาวิชา** วิศวกรรมการวัดคุม

**ปีการศึกษา** 2553

| อาจารย์ผู้ควบคุมปริญญานิพนธ์ | ลายมือชื่อ |
|---|---|
| อาจารย์กฤษณ์ เสมอพิทักษ์ | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หัวข้อปริญญานิพนธ์** การควบคุมอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10

POWER LINE COMMUNICATION WITH X10


**นักศึกษาผู้จัดทำ** นางสาวพรพิสุทธิ์ นุกูลราษฎร์ รหัสนักศึกษา 50011052

นายพรรณเชษฐ์ พวงนาค รหัสนักศึกษา 50011054

นายพลเทพ นามพระจันทร์ รหัสนักศึกษา 50011062

**อาจารย์ที่ปรึกษา** อาจารย์กฤษณ์ เสมอพิทักษ์


**ปีการศึกษา** 2553

## บทคัดย่อ


ในปัจจุบันนี้มนุษย์ต้องการความสะดวกสบายในการดำรงชีวิต และเครื่องอำนวยความสะดวกสบายในปัจจุบันนี้จะใช้ไฟฟ้าในการสั่งการทำงานทั้งนั้น และเนื่องจากการเปิดหรือปิดอุปกรณ์ไฟฟ้า ไม่สามารถเปิด-ปิดอุปกรณ์ไฟฟ้าได้หลาย ๆ ตัวในที่เดียวกัน เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านสายเอซีทำให้เรามีความสะดวกสบายมากขึ้น และเป็นประโยชน์อย่างเห็นได้ชัด โดยใช้หลักการในการส่งผ่านสายเอซีคือ การมอดูเลตสัญญาณแบบเอฟเอสเค และแบ่งการทำงานออกเป็น 2 ส่วนใหญ่ ๆ คือ ภาคส่งสัญญาณควบคุม และภาครับสัญญาณควบคุม




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Thesis Title** Power line Communication with X10

**Authors** Miss Pornpisut Nukulrad

Mr. Pannachet Pongnak

Mr. Ponthep Namprajan

**Thesis Advisor** Mr. Krit Smerpitak

**Year** 2010

## ABSTRACT

Nowaday, human needs more and more the convenient life. For example, many electrical appliances are located in many places in the house. Each one needs a switch to turn on and turn off itself. Therefore, this project tries to overcome the mentions defect by designing an appliance controller via AC line. All the controlled electrical appliances will be functioned on demand via AC line. the controlling signal will be modulated and demodulated in the form of Frequency shift keying (FSK). This designed system is consisted of two parts, one is the signal transmitter and the other is the signal receiver.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์จากปัจจัยหลาย ๆ อย่าง ซึ่งทางคณะผู้จัดทำขอขอบคุณคณะวิศวกรรมศาสตร์ และภาควิชาวิศวกรรมการวัดคุม ที่ได้ให้ประสบการณ์ในการใช้ชีวิตในมหาวิทยาลัย ซึ่งการทำปริญญานิพนธ์ฉบับนี้ต้องขอขอบพระคุณเป็นอย่างสูงสำหรับ อาจารย์กฤษณ์ เสมอพิทักษ์ รองศาสตราจารย์ประภาษ อุคคกิมาพันธุ์ อาจารย์ธีรวัฒน์ เทพมณี และอาจารย์ท่านอื่น ๆ ที่ให้คำแนะนำปรึกษาแก่คณะผู้จัดทำมาโดยตลอด อีกทั้งยังเอื้อเฟื้ออุปกรณ์ ห้องทำโครงงาน เครื่องมืออุปกรณ์ต่าง ๆ ตลอดจนความช่วยเหลือทั้งในด้านทุนทรัพย์ และแรงงาน ซึ่งทางคณะผู้จัดทำรู้สึกซาบซึ้งเป็นอย่างยิ่ง จึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้ ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ภายในภาควิชาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยให้กำลังใจเสมอมา ทั้งยังให้การสนับสนุนช่วยเหลือในทุก ๆ เรื่อง ขอขอบพระคุณบิดา มารดา ที่ให้การสนับสนุน และเป็นกำลังใจเสมอมา รวมถึงการช่วยเหลือในทุก ๆ เรื่อง

สุดท้ายนี้คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าปริญญานิพนธ์ฉบับนี้ จะเป็นประโยชน์ต่อผู้ที่สนใจศึกษาค้นคว้าบ้างไม่มากก็น้อยตามความเหมาะสม และสามารถนำไปใช้ศึกษา พัฒนาให้เกิดประโยชน์ต่อไป

**คณะผู้จัดทำ**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า


**บทคัดย่อภาษาไทย** ..... I

**บทคัดย่อภาษาอังกฤษ** ..... II

**กิตติกรรมประกาศ** ..... III

**สารบัญ** ..... IV

**สารบัญตาราง** ..... VIII

**สารบัญภาพ** ..... IX

**บทที่ 1 บทนำ** ..... 1

1.1 ความสำคัญของปริญญานิพนธ์ ..... 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ ..... 2

1.3 ขอบเขตของปริญญานิพนธ์ ..... 2

1.4 ขั้นตอนการศึกษา ..... 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ ..... 2

**บทที่ 2 ทฤษฎีและหลักการ** ..... 3

2.1 Power Line Communications: PLC ..... 3

2.1.1 ส่วนประกอบทางกายภาพ (Physical Components of BPL) ..... 4

2.1.2 รูปแบบการเชื่อมต่อโครงข่ายระบบไฟฟ้า (Grid Topology) ..... 5

2.1.3 สถาปัตยกรรมโครงข่าย (Network Architectures) ..... 8

2.1.4 ข้อจำกัดทางเทคนิค (Technical Limitations) ..... 11

2.2 การมอดูเลตสัญญาณ (Signal Modulation) ..... 13

2.2.1 การมอดูเลตสัญญาณอนาล็อก ..... 13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า


2.3 ความรู้ทั่วไปเกี่ยวกับ X10 ..... 16

2.3.1 ประวัติความเป็นมา ..... 16

2.3.2 ทฤษฎีของ X10 ..... 17

2.4 สภาพแวดล้อมของสายไฟฟ้ากำลัง ..... 21

2.4.1 ปัญหาการลดทอนของสัญญาณสื่อสาร ..... 21

2.4.2 ปัญหาสัญญาณรบกวนต่าง ๆ ในระบบไฟฟ้ากำลัง ..... 22

2.5 มาตรฐานที่ต้องคำนึงถึงในการสื่อสารบนสายไฟฟ้ากำลัง ..... 22

2.5.1 การเลือกความถี่ ..... 22

2.5.2 ค่าแรงดันในการส่ง ..... 22

2.6 ไมโครคอนโทรเลอร์ ..... 23

2.6.1 คุณสมบัติของ PIC16F648A ..... 23

2.6.2 หน้าที่การทำงานของแต่ละขา ..... 24

2.6.3 ขั้นตอนการพัฒนาโปรแกรม ..... 26

2.7 การสื่อสารผ่านทางพอร์ตอนุกรม (Serial Port Communication) ..... 28

2.7.1 พื้นฐานการสื่อสารแบบอนุกรม ..... 28

2.7.2 รูปแบบของการสื่อสารข้อมูลอนุกรม ..... 29

2.8 การใช้งานพอร์ตอนุกรม RS232 ..... 31

2.8.1 รายละเอียดของสายสัญญาณ ..... 33

2.8.2 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9 ..... 33

2.8.3 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม ..... 35



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า


2.8.4 อัตราเร็วในการสื่อสารข้อมูลอนุกรม ..... 36

2.8.5 ระดับสัญญาณของ RS232 ..... 37

2.9 โปรแกรมวิชวล เบสิก (Visual Basic) ..... 38

2.9.1 คุณสมบัติเด่น ของวิชวล เบสิก ..... 39

2.9.2 การเขียนโปรแกรมติดต่อและควบคุม Serial Port ด้วย Visual Basic ..... 40

2.9.3 การกำหนดคุณสมบัติ ( Property ) ที่สำคัญในการใช้งาน MSComm ให้สามารถต่อกับพอร์ตได้ ..... 41

**บทที่ 3 หลักการทำงานและการออกแบบ ..... 43**

3.1 โครงสร้างของระบบการส่งข้อมูลไปควบคุมอุปกรณ์ไฟฟ้า ..... 43

3.2 โฟลว์ชาร์ตการทำงานของภาคส่ง และภาครับ ..... 46

3.3 หลักการในการมอดูเลต ..... 47

3.3.1 การทำงานของภาคส่ง ..... 47

3.3.2 การทำงานของภาครับ ..... 48

3.3.3 ภาคควบคุม ..... 49

3.3.4 ภาคจ่ายไฟ ..... 49

3.4 โปรแกรมควบคุมผ่านเครื่องคอมพิวเตอร์ ..... 50

**บทที่ 4 การทดลองและผลการทดลอง ..... 52**

4.1 ทดลองการทำงานของวงจรภาคส่ง ..... 52

4.2 ทดลองการทำงานของวงจรภาครับ ..... 55



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า

4.3 ทดลองการสั่งงานจากโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์
กับ วงจรภาครับ และภาคส่ง ..... 59

**บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ ..... 65**

5.1 สรุปและวิจารณ์ผลการทดลอง ..... 65

5.1.1 การทดลองการทำงานของวงจรภาคส่ง ..... 65

5.1.2 การทดลองการทำงานของวงจรภาครับ ..... 65

5.2 ปัญหาของโครงงาน ..... 65

5.3 ข้อเสนอแนะ ..... 66

**บรรณานุกรม ..... 67**

**ภาคผนวก ..... 68**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


**ตารางที่** **หน้า**

2.1 House Code ..... 19

2.2 Key Code ..... 20

2.3 ช่วงความถี่ที่ใช้ได้ตามมาตรฐาน EN50065-1 ..... 22

2.4 ตารางแสดงความแตกต่างของพอร์ตขนาน และพอร์ตอนุกรม ..... 29

2.5 ตารางแสดงรายละเอียดของสายสัญญาณต่าง ๆ ของพอร์ตอนุกรม ..... 32

3.1 ตารางแสดงอุปกรณ์และหน้าที่ของระบบ ..... 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ


ภาพที่ หน้า

2.1 รูปแบบการเชื่อมต่อแบบบัส ..... 6

2.2 รูปแบบการเชื่อมต่อแบบดาว ..... 7

2.3 รูปแบบการเชื่อมต่อแบบวงแหวน ..... 8

2.4 สถาปัตยกรรมโครงข่ายเส้นใยนำแสงและสายไฟฟ้าแรงดันต่ำ ..... 9

2.5 สถาปัตยกรรมโครงข่ายสายไฟฟ้าแรงดันปานกลางต่ำและสายไฟฟ้าแรงดันต่ำ ..... 10

2.6 สถาปัตยกรรมโครงข่ายสายไฟฟ้าแรงดันปานกลางและโครงข่ายไร้สาย ..... 11

2.7 การมอดูเลตเชิงเลขทางแอมพลิจูด (ASK) ..... 15

2.8 การมอดูเลตเชิงเลขทางความถี่ (FSK) ..... 15

2.9 การมอดูเลตเชิงเลขทางเฟส (PSK) ..... 16

2.10 การส่งข้อมูลบน AC-Line ของ X10 ที่ความถี่ 50 เฮิร์ต ..... 17

2.11 ชุดคำสั่งระบุแอดเดรสและฟังก์ชั่น ..... 18

2.12 การส่งข้อมูลแบบบิตแล้วตามด้วย compliment ของบิตนั้น (ไม่รวม Start Code) ..... 21

2.13 PIC16F648A ..... 23

2.14 โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC เบอร์ PIC16F648A ..... 25

2.15 โฟลว์ชาร์ตขั้นตอนการพัฒนาโปรแกรม ..... 26

2.16 โปรแกรม PICkit สำหรับโปรแกรมลงใน ไมโครคอนโทรลเลอร์ ตระกูล PIC ..... 27

2.17 บอร์ดสำหรับโปรแกรมไมโครคอนโทรลเลอร์ ตระกูล PIC ..... 27

2.18 แสดงการส่งข้อมูลขนาด 8 บิตแบบอนุกรมพร้อมด้วย บิตเริ่มต้น บิตพาริตี้ บิตหยุด ด้วยความเร็ว 9600 บิตต่อวินาที ..... 30

2.19 สัญญาณการสื่อสารแบบซิงโครนัส ..... 30

2.20 ลักษณะการคอนเนคเตอร์แบบ D-Type ..... 31



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ(ต่อ)

**ภาพที่** **หน้า**


2.21 พอร์ตอนุกรมของ PC DB9 ตัวผู้ (Male) ..... 31

2.22 พอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย(Female) ..... 31

2.23 DB9 ตัวผู้ (มองจากด้านหลัง) ..... 32

2.24 แบบ NULL Modem ..... 33

2.25 แบบ 3 Wire ..... 34

2.26 แบบ 3 Wire None Flow Control ..... 34

2.27 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที ..... 37

2.28 ระดับสัญญาณของ RS232C และระดับสัญญาณของ TTL ในสถานะลอจิก 1 และ 0 ..... 37

2.29 หน้าต่างแสดงอุปกรณ์และเครื่องมือพื้นฐานของวิชวล เบสิก ..... 39

2.30 เพิ่มคอมโพเนนต์ MSComm ..... 40

2.31 ไดอะล็อก Component เลือกที่รายการ MSComm ..... 41

3.1 โครงสร้างของระบบนั้นจะประกอบไปด้วย ..... 43

3.2 โฟลว์ชาร์ตการทำงานของภาคส่ง และภาครับ ..... 46

3.3 วงจรภาครับ และภาคส่ง X10 ..... 47

3.4 ส่วนควบคุมด้วย MCU ..... 48

3.5 ภาคจ่ายไฟ ..... 49

3.6 แผนผังของการวัดระดับแรงดันไฟฟ้า ..... 50

3.7 หน้าตาของโปรแกรมควบคุมอุปกรณ์ ..... 50

3.8 โฟลว์ชาร์ตการทำงานของโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์ ..... 51

4.1 แสดงการวัดสัญญาณจุดตัดศูนย์ที่ขา 6 ของ Op Amp 741 ..... 52

4.2 สัญญาณจุดตัดศูนย์ ..... 53



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ (ต่อ)

**ภาพที่** **หน้า**


4.3 แสดงการวัดสัญญาณที่ขา Collector ของ BC-547 ..... 54

4.4 แสดงสัญญาณความถี่ 120 กิโลเฮิร์ต ..... 55

4.5 แสดงการวัดสัญญาณที่ขา X10-RX ของภาครับ ..... 56

4.6 แสดงขาที่ใช้ในการส่งข้อมูลที่เป็นสัญญาณ TTL เข้าไปที่ขา X10-RX ของภาครับ ..... 57

4.7 แสดงการรับข้อมูลที่เป็นสัญญาณ TTL เข้าไปที่ขา X10-RX ของภาครับ ..... 57

4.8 แสดงการส่งข้อมูลแบบต่อเนื่อง ..... 58

4.9 แสดงการส่งข้อมูลแบบต่อเนื่องโดยทำการปรับ Time/Div ให้ลดลง ..... 59

4.10 แสดงการสั่งงานผ่านทางโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์
ให้กับ House A ทำงาน ..... 60

4.11 แสดงการสั่งงานให้ A01 ทำงาน ..... 61

4.12 แสดงการสั่งงานให้ A02 ทำงาน ..... 62

4.13 แสดงการสั่งงานให้ A03 ทำงาน ..... 63

4.14 แสดงการต่ออุปกรณ์ที่ใช้กับการสั่งงาน โดยใช้คำสั่ง Bright และ Dimmer ..... 64



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันนี้ มีความคิดมากมายเพื่อให้การดำรงชีวิตประจำวันมีความสะดวกสบายมากขึ้น ซึ่งหนึ่งในนั้นก็ คือ การควบคุมอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10 ซึ่งผู้ที่อยู่อาศัยสามารถสั่งงานอุปกรณ์ไฟฟ้าภายในบ้านให้สามารถ เปิด-ปิด ได้ตามความต้องการของผู้ที่อยู่อาศัย โดยสามารถสั่งงานจากเครื่องควบคุมได้เลย ซึ่งเครื่องควบคุมนี้อาจจะเป็นเครื่องควบคุมระยะไกล สามารถสั่งงานได้จากโต๊ะทำงานที่ใช้ประจำ หรือสถานที่ต่าง ๆ ภายในบ้านที่มีความสะดวกสบายสำหรับทุก ๆ คนภายในบ้านในการใช้งาน โดยที่ควบคุมอุปกรณ์และเครื่องใช้ไฟฟ้าต่าง ๆ ผ่านโปรแกรมบนหน้าจอคอมพิวเตอร์ ซึ่งแนวความคิดนี้มีประโยชน์อย่างมากสำหรับบ้านที่มีผู้อยู่อาศัยที่เป็นคนชรา หรือ ผู้พิการ

โครงงานนี้จะกล่าวถึงการควบคุมอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10 โดยสามารถควบคุมการ เปิด-ปิด อุปกรณ์ไฟฟ้าภายในบ้านด้วยการสั่งงานจากจุด ๆ เดียว จึงทำให้เป็นการประหยัดเวลา รูปแบบการทำงานนั้นจะใช้หลักการในการส่งข้อมูลด้วยสัญญาณอนาล็อกในการส่งข้อมูลดิจิตอล เป็นการเปลี่ยนคุณลักษณะของคลื่นรูปซายน์ไปตามการเปลี่ยนแปลงของข้อมูลที่ต้องการส่ง หรือเรียกอีกอย่างหนึ่งว่า การมอดูเลตสัญญาณดิจิตอล ซึ่งในโครงงานนี้ใช้สัญญาณแบบการมอดูเลตเลขเชิงความถี่ (Frequency Shift Keying : FSK) ทำให้ความถี่เปลี่ยนแปลงตามบิตข้อมูล โดยให้แอมพลิจูด และเฟสคงที่

รูปแบบการส่งข้อมูลจะใช้การส่งสัญญาณนาฬิกาที่ได้จากจุดตัดศูนย์ (zero crossing point) เป็นจุดเปรียบเทียบ โดยค่าของข้อมูลไบนารี 1 จะส่งสัญญาณบัสความถี่ 120 กิโลเฮิรตซ์ (KHz) จำนวน 3 ลูก หลังจากจุดตัดศูนย์ โดยที่ไม่จำเป็นว่าจะเป็นขอบขาขึ้นหรือขอบขาลงของสัญญาณนาฬิกา และค่าของข้อมูลไบนารี 0 จะเป็นการเว้นช่องว่างของสัญญาณเบิซท์โดยที่ 1 ไซเคิลนั้นจะสามารถส่งข้อมูลได้ 2 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาหลักการทำงานของการควบคุมอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10
2. เพื่อออกแบบอุปกรณ์ควบคุมการทำงานของอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10

## 1.3 ขอบเขตของปริญญานิพนธ์

ศึกษาการทำงาน และออกแบบอุปกรณ์ควบคุมการทำงานของอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10 และสามารถสั่งงานอุปกรณ์ไฟฟ้าผ่าน AC-Line ได้ตามมาตรฐานของ X10

## 1.4 ขั้นตอนการศึกษา

1. ศึกษาหลักการทำงานของมาตรฐาน X10
2. ออกแบบอุปกรณ์โดยใช้โปรแกรม Protel 99 SE
3. สร้างอุปกรณ์ภาครับและภาคส่ง
4. เขียนโปรแกรมเพื่อควบคุมการทำงานของอุปกรณ์โดยใช้โปรแกรม Visual Basic
5. จัดทำปริญญานิพนธ์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถออกแบบและสร้างอุปกรณ์ที่สามารถควบคุมการทำงานของอุปกรณ์ไฟฟ้าผ่าน AC-Line ตามมาตรฐานของ X10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎีและหลักการ

## 2.1 Power Line Communications: PLC

เทคโนโลยีการสื่อสารผ่านสายไฟฟ้า (Power Line Communications : PLC) คือการที่ส่งสัญญาณข้อมูลในช่วงความถี่ต่าง ๆ ไปบนสัญญาณไฟฟ้าซึ่งมีความถี่ 50 – 60 เฮิรตซ์ (Hz) ในระยะแรก เทคโนโลยีการสื่อสารผ่านสายไฟฟ้าถูกประยุกต์ใช้กับการสื่อสารข้อมูลที่มีความเร็วไม่มาก ระดับกิโลบิตต่อวินาที (kbps) ของช่วงความถี่ต่ำ (Low frequency : LF) ระดับ 30 - 300 กิโลเฮิรตซ์ (kHz) โดยส่วนใหญ่จะใช้ในการควบคุมและเพิ่มประสิทธิภาพของระบบไฟฟ้า เช่น การควบคุมอุปกรณ์จ่ายไฟฟ้า การอ่านหน่วยมาตรวัดแบบอัตโนมัติ (Automatic Meter Reading : AMR) การตรวจสอบภาวะไฟฟ้าขัดข้อง หรือการควบคุมโหลดไฟฟ้า ฯลฯ รวมทั้งได้มีการประยุกต์ใช้เทคโนโลยีการสื่อสารผ่านสายไฟฟ้า เพื่อควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ ภายในบ้านเช่น ควบคุมระบบแสงสว่าง ระบบรักษาความปลอดภัย ระบบสัญญาณแจ้งเตือนอัคคีภัย และระบบปรับอากาศเป็นต้น ต่อมาเมื่อความต้องการใช้งานระบบสื่อสารความเร็วสูงเพิ่มมากขึ้น จึงได้มีการพัฒนาเทคโนโลยีทางด้านการสื่อสารผ่ายสายไฟฟ้าให้สามารถรองรับการสื่อสารข้อมูลความเร็วสูง (Broadband communication) ในระดับเมกะบิตต่อวินาที (Mbps) ในช่วงความถี่สูง (High frequency : HF) ระดับ 1 – 40 เมกะเฮิรตซ์ (MHz) เช่น การสื่อสารข้อมูลแบบมัลติมีเดียหรือสื่อประสม (Multimedia) การใช้งานอินเทอร์เน็ตความเร็วสูง จึงเป็นที่รู้จักในชื่อเทคโนโลยีการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า

นอกจากนี้ความต้องการใช้บริการอินเทอร์เน็ตความเร็วสูงเพิ่มมากขึ้นอย่างรวดเร็ว และเทคโนโลยีการสื่อสารข้อมูลแบบต่าง ๆ อาจไม่เหมาะสมในการให้บริการอย่างครอบคลุมและทั่วถึงในเวลาต่อไป โดยเฉพาะในพื้นที่ชนบทที่อยู่ห่างไกล ดังนั้นเทคโนโลยีการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าจึงมีความสำคัญเพิ่มขึ้น สำหรับการประยุกต์ใช้เพื่อให้บริการในส่วนโครงข่ายเข้าถึง (Access Network) สำหรับการสื่อสารเนื่องจากสายไฟฟ้าเป็นโครงสร้างพื้นฐานที่มีการติดตั้งใช้งานอย่างครอบคลุมพื้นที่ส่วนใหญ่อยู่แล้ว จึงทำให้ไม่ต้องเสียค่าใช้จ่ายในการเดินสายส่งสัญญาณใหม่ แต่ไม่ว่าอย่างไรก็ตามเทคโนโลยีการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้านั้น ก็มีข้อจำกัดเกี่ยวกับปัญหาของสัญญาณรบกวน โดยสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า ซึ่งในสายไฟฟ้าจะเป็นแหล่งกำเนิดสัญญาณรบกวนไปยังระบบสื่อสารอื่น ๆ ที่ใช้งานย่านความถี่เดียวกัน เนื่องจากสายไฟฟ้าไม่สามารถป้องกันการแพร่กระจายคลื่นและสัญญาณสื่อสาร บรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าจะถูกรบกวนจากสัญญาณภายนอกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเช่นกัน รวมทั้งมีการรบกวนเนื่องจากระบบไฟฟ้าซึ่งอาจมีการเชื่อมต่อใช้งานที่ไม่ถูกต้อง จากสาเหตุดังกล่าวจึงทำให้สัญญาณการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า เกิดการลดทอนอย่างรวดเร็วและต้องใช้อุปกรณ์ทวนสัญญาณจำนวนมาก เพื่อเพิ่มความแรงของสัญญาณให้อยู่ในระดับที่เหมาะสม

### 2.1.1 ส่วนประกอบทางกายภาพ (Physical Components of BPL)

โครงสร้างของโครงข่ายระบบไฟฟ้า อันประกอบด้วยสายส่งไฟฟ้าซึ่งมีแรงดันสามระดับ ได้แก่ แรงดันสูง 110 – 400 กิโลโวลท์ (kV) แรงดันปานกลาง 10 – 30 กิโลโวลท์ และแรงดันต่ำ 220 – 440 โวลท์ (V) เชื่อมต่อกันผ่านหม้อแปลงซึ่งทำหน้าที่ปรับลดระดับแรงดัน โดยหม้อแปลงจะทำหน้าที่เสมือนตัวกรองสัญญาณความถี่ต่ำผ่าน (low-pass filter) ซึ่งจะยอมให้สัญญาณที่มีความถี่ประมาณไม่เกิน 20 กิโลเฮิรตซ์ ผ่านได้เท่านั้น ทำให้สัญญาณข้อมูลการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าในช่วงความถี่สูงไม่สามารถผ่านหม้อแปลงได้โดยตรง นอกจากนั้นสายส่งไฟฟ้ายังถูกออกแบบมาให้มีความเหมาะสมสำหรับการส่งสัญญาณในช่วงความถี่ต่ำ และจะเกิดการลดทอนสัญญาณเพิ่มขึ้นเมื่อส่งสัญญาณที่มีความถี่สูงขึ้น ดังนั้นจะพบว่าคุณลักษณะของระบบไฟฟ้าจึงมีความสัมพันธ์กับการใช้งานเทคโนโลยีการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าเป็นอย่างมาก โดยส่วนประกอบที่สำคัญของระบบไฟฟ้าได้แก่

- **สายส่งไฟฟ้า (Transmission line)**

สัญญาณการสื่อสารบรอดแบนด์ความเร็วสูง ผ่านสายไฟฟ้าจะถูกส่งผ่านสายส่งไฟฟ้าไปยังปลายทางได้อย่างถูกต้อง ก็ต่อเมื่อสัญญาณทุกค่าความถี่ใช้งานเกิดการลดทอนเท่ากันและเกิดการประวิง (Delay) เท่ากัน หรือกล่าวได้ว่าความเร็วเฟสของสัญญาณ (Phase velocity) และค่าคงที่การลดทอนของสัญญาณ (Attenuation constant) จะต้องมีค่าคงที่ตลอดทั้งย่านความถี่ใช้งานของสัญญาณ สายส่งไฟฟ้าแรงดันสูงมีความยาว ตั้งแต่หลายสิบถึงหลายร้อยกิโลเมตร แบ่งออกเป็นสายส่งสองลักษณะ คือ สายส่งข้ามศีรษะ (overhead lines) และสายส่งใต้ดิน (ground cables)

สายส่งไฟฟ้าแรงดันปานกลาง หรือแรงดันต่ำ (Medium Voltage/Low Voltage) ประกอบด้วยสายส่งข้ามศีรษะในปริมาณใกล้เคียงกับสายส่งใต้ดิน สายส่งไฟฟ้าแรงดันปานกลางมีความยาวประมาณ 5 – 25 กิโลเมตรและสายส่งไฟฟ้าแรงดันต่ำมีความยาวประมาณ 100 – 500 เมตร

นอกจากนี้ความต้านทานเชิงซ้อน (Impedance) ของสายส่งไฟฟ้าก็มีผลกระทบต่อการส่งสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า โดยถ้าความต้านทานเชิงซ้อนของช่องสัญญาณไม่เข้าคู่กับความต้านทานเชิงซ้อนของแหล่งกำเนิดสัญญาณแล้วนั้น ความต้านทานเชิงซ้อนของโหลดของสัญญาณก็จะก่อให้เกิดการสะท้อนกลับของสัญญาณที่จุดที่เชื่อมต่อ และมีสัญญาณบางส่วนที่ผ่านไปถึงภาครับ จะทำให้สัญญาณที่ได้รับมีค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio : SNR) ลดต่ำลงและมีอัตราบิตผิดพลาด (Bit Error Rate : BER)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มสูงขึ้น โดยค่าสัมประสิทธิ์การสะท้อนซึ่งแสดงอัตราส่วนระหว่างกำลังสัญญาณที่สะท้อนกลับต่อกำลังสัญญาณที่มาถึงจุดเชื่อมต่อ มีความสัมพันธ์กับความต้านทานเชิงซ้อนของสายส่งไฟฟ้าที่เชื่อมต่อกัน ดังนั้นในการส่งสัญญาณจำเป็นต้องคำนึงถึงค่าความต้านทานเชิงซ้อนของสายไฟฟ้าที่เชื่อมต่อกัน เพื่อให้สัญญาณที่ภาครับมีความประสิทธิภาพเพียงพอต่อการใช้งาน

โดยทั่วไปโครงข่ายระบบไฟฟ้าจะประกอบด้วยสายเคเบิลชนิดต่าง ๆ มาเชื่อมต่อกันเป็นจำนวนมาก เนื่องจากมีการต่อขยายและพัฒนาระบบไฟฟ้าเพิ่มขึ้นตลอดเวลาจึงทำให้การเข้าคู่ความต้านทางเชิงซ้อน (Impedance matching) เพื่อใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าจะต้องคำนึงถึงความต้านทางเชิงซ้อน ที่มีการเปลี่ยนแปลงของสายส่งไฟฟ้าแต่ละส่วนที่เชื่อมต่อกัน นอกจากนั้นความต้านทางเชิงซ้อนของสายส่งไฟฟ้ายังมีค่าเปลี่ยนแปลงตามเวลา (time-variant) เนื่องจากการเพิ่ม หรือลดจำนวนโหลดไฟฟ้าที่เชื่อมต่ออยู่กับสายส่งไฟฟ้า จึงอาจส่งผลให้เกิดการไม่เข้าคู่ความต้านทางเชิงซ้อน (Impedance mismatch) ซึ่งในการใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า

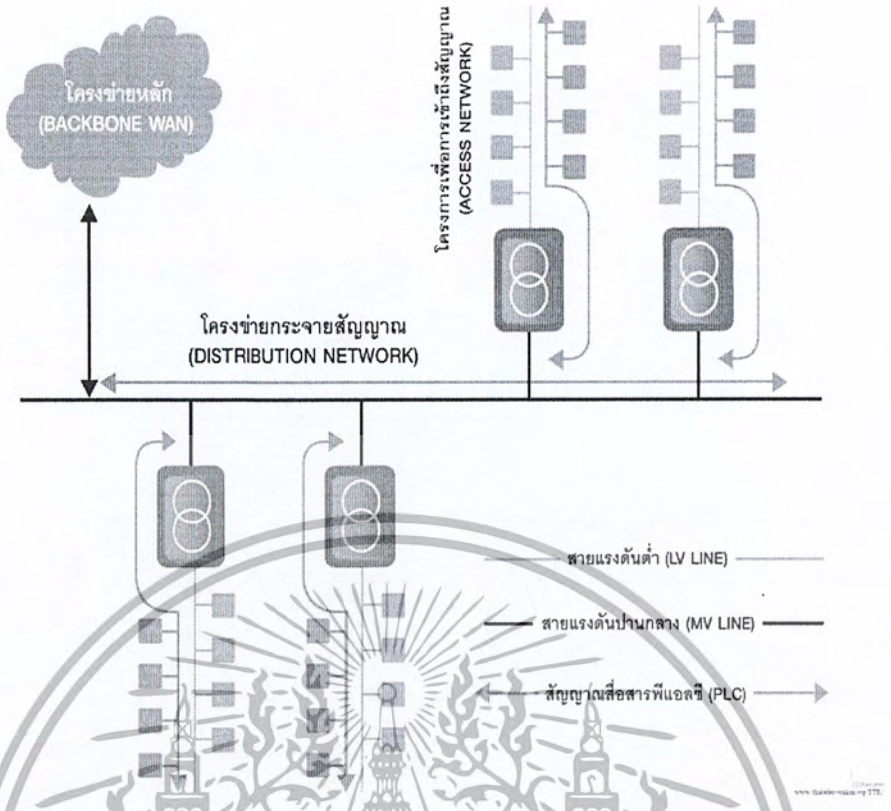
### 2.1.2 รูปแบบการเชื่อมต่อโครงข่ายระบบไฟฟ้า (Grid Topology)

โดยทั่วไปจะมีรูปแบบการเชื่อมต่อโครงข่ายระบบไฟฟ้าสามรูปแบบ ดังนี้

- **รูปแบบการเชื่อมต่อแบบบัส (Bus topology)**

ภาพที่ 2.1 แสดงรูปแบบการเชื่อมต่อแบบบัสซึ่งประกอบด้วยสายส่งไฟฟ้าแรงดันปานกลางหนึ่งเส้น และสายส่งไฟฟ้าแรงดันต่ำจำนวนหนึ่งที่เชื่อมต่อกับสายส่งไฟฟ้าแรงดันปานกลางเพื่อส่งสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าไปยังผู้ใช้ การเชื่อมต่อรูปแบบนี้ใช้เงินลงทุน ในการก่อสร้างระบบไม่สูงมากแต่มีความน่าเชื่อถือต่ำ เนื่องจากเมื่อสายส่งไฟฟ้าแรงดันปานกลางเกิดความผิดพลาด ผู้ใช้จำนวนมากที่อยู่ในแนวผ่านของสายที่เกิดความผิดพลาดนั้นจะไม่สามารถใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

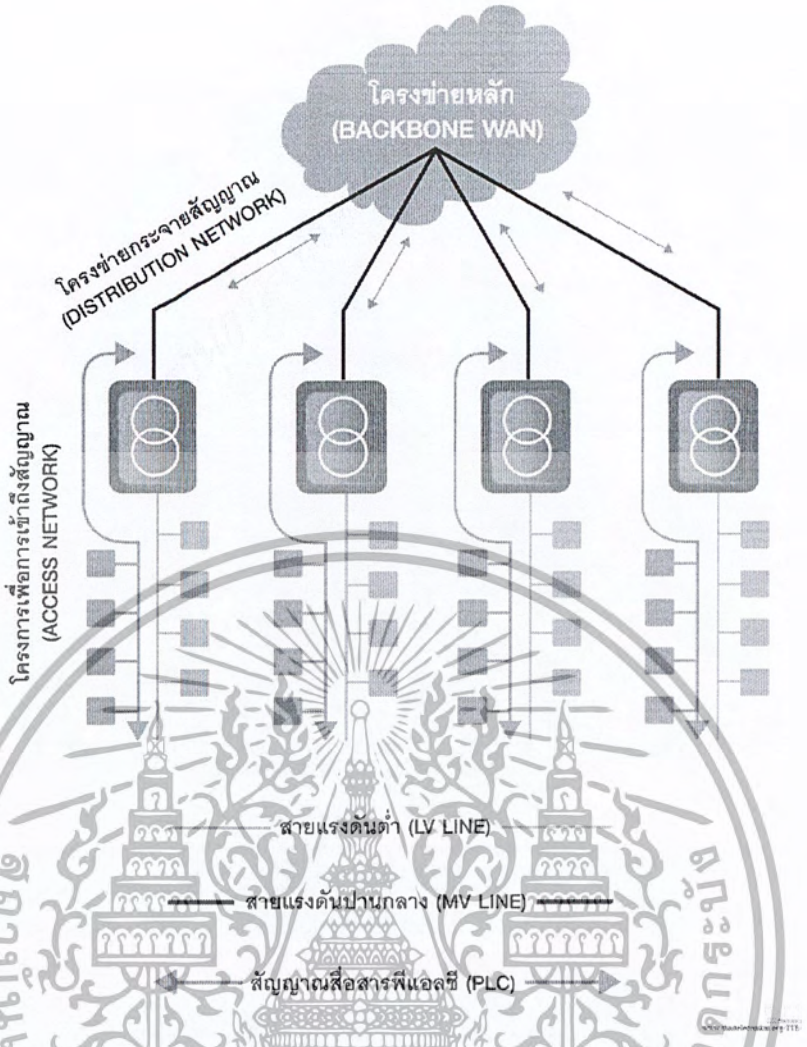



**ภาพที่ 2.1** รูปแบบการเชื่อมต่อแบบบัส

- **รูปแบบการเชื่อมต่อแบบดาว (Star topology)**

ภาพที่ 2.2 แสดงรูปแบบการเชื่อมต่อแบบดาว ซึ่งจะพบว่าแต่ละโครงข่ายเข้าถึง ในการใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าซึ่งจะถูกเชื่อมต่อเข้ากับ โครงข่ายแกนหลัก (Backbone network) แบบแยกออกจากกัน ดังนั้นเมื่อเกิดความผิดพลาดขึ้นที่สายส่งไฟฟ้าแรงดันปานกลาง จำนวนผู้ใช้ที่ไม่สามารถใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า จะถูกจำกัดอยู่เพียงภายในพื้นที่ของโครงข่ายเข้าถึงที่เกิดความผิดพลาดขึ้นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

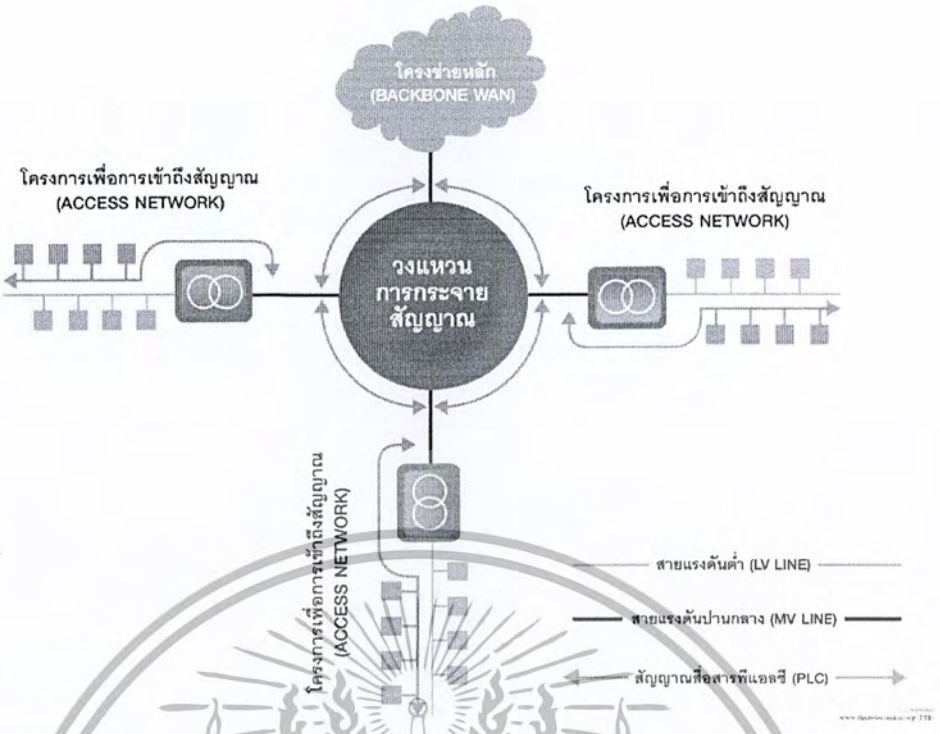



**ภาพที่ 2.2** รูปแบบการเชื่อมต่อแบบดาว

- **รูปแบบการเชื่อมต่อแบบวงแหวน (Ring topology)**

ภาพที่ 2.3 แสดงรูปแบบการเชื่อมต่อแบบวงแหวน ซึ่งเป็นรูปแบบการเชื่อมต่อ ที่มีความน่าเชื่อถือสูงที่สุดเนื่องจากเมื่อสายส่งไฟฟ้าแรงดันปานกลางในวงแหวนเกิดความผิดพลาดสัญญาณสื่อสารบรอดแบนด์ ความเร็วสูงผ่านสายไฟฟ้าจะถูกส่งต่อไปยังผู้ใช้ผ่านอีกเส้นทางหนึ่ง ของวงแหวนโดยอัตโนมัติภายในเวลาอันสั้น (วินาที)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 2.3** รูปแบบการเชื่อมต่อแบบวงแหวน

### 2.1.3 สถาปัตยกรรมโครงข่าย (Network Architectures)

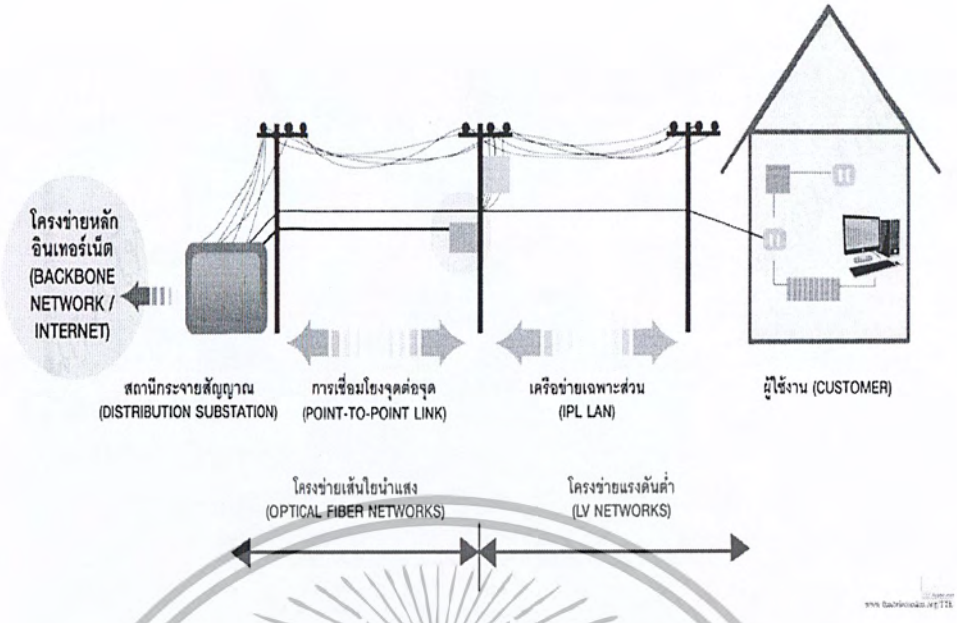
สถาปัตยกรรมโครงข่ายสำหรับการให้บริการ การสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าสามารถแบ่งออกได้เป็นรูปแบบต่าง ๆ ได้แก่

- **สถาปัตยกรรมโครงข่ายเส้นใยนำแสง (Fiber optic) และสายไฟฟ้าแรงดันต่ำ (Low-voltage line)**

โครงข่ายนี้ข้อมูลจะถูกส่งจากโครงข่ายแกนหลักหรือโครงข่ายอินเทอร์เน็ตซึ่งเชื่อมต่ออยู่ที่สถานีไฟฟ้า ผ่านโครงข่ายเส้นใยนำแสงที่พาดผ่านเสาไฟฟ้าไปยังเสาไฟฟ้าตำแหน่งซึ่งมีหม้อแปลงไฟฟ้าแรงดันปานกลางหรือแรงดันต่ำ ติดตั้งอยู่และมีอุปกรณ์เปลี่ยนสัญญาณแสงไฟฟ้า (optical-electrical converter) ซึ่งจะทำหน้าที่แปลงสัญญาณแสงเป็นสัญญาณไฟฟ้า จากนั้นข้อมูลจึงจะถูกส่งผ่านสายไฟฟ้าแรงดันต่ำต่อไปยังผู้ใช้ ดังแสดงในภาพที่ 2.4

ข้อดีของสถาปัตยกรรมโครงข่ายแบบนี้คือ สายเส้นใยนำแสงสามารถรองรับการสื่อสารข้อมูลที่มีความกว้างของแถบความถี่ขนาดใหญ่ และมีความน่าเชื่อถือสูงกว่าการสื่อสารข้อมูลผ่านสายไฟฟ้าแรงดันปานกลาง รวมทั้งไม่เกิดการแพร่กระจายคลื่นวิทยุไปรบกวนระบบ สื่อสารอื่น ๆ แต่มีข้อจำกัด คือ จะต้องลงทุนติดตั้งอุปกรณ์เปลี่ยนสัญญาณแสงไฟฟ้าที่เสาไฟฟ้าซึ่งจะมีราคาสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

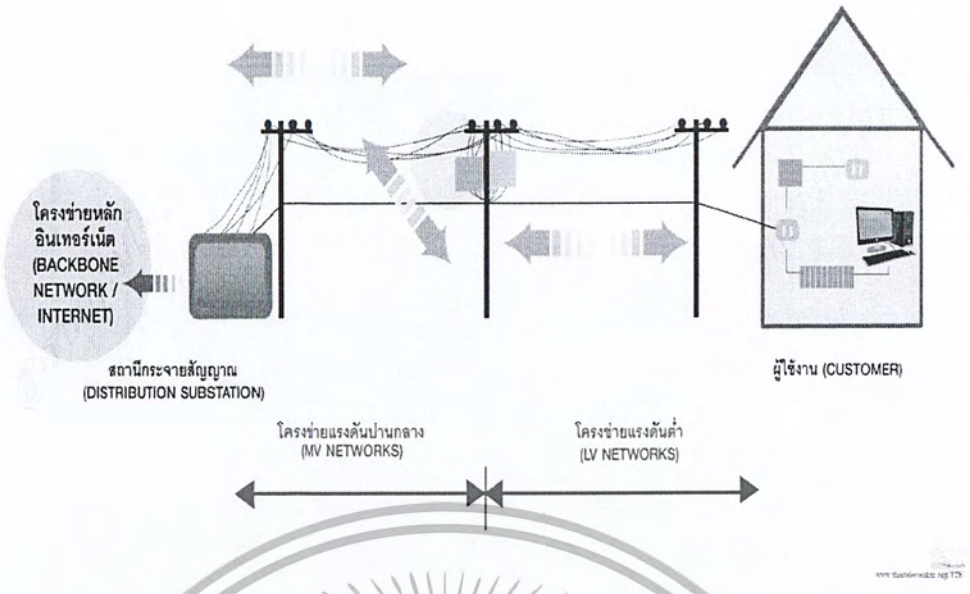



**ภาพที่ 2.4** สถาปัตยกรรมโครงข่ายเส้นใยนำแสงและสายไฟฟ้าแรงดันต่ำ

- **สถาปัตยกรรมโครงข่ายสายไฟฟ้าแรงดันปานกลางต่ำและสายไฟฟ้าแรงดันต่ำ**

โครงข่ายลักษณะนี้ข้อมูลจะถูกส่งจากโครงข่ายแกนหลักผ่านสายไฟฟ้าแรงดันปานกลาง ไปยังเสาไฟฟ้าที่มีหม้อแปลงไฟฟ้าแรงดันปานกลางหรือแรงดันต่ำติดตั้งอยู่ จากนั้นสัญญาณจะสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า โดยที่สัญญาณนั้นจะต้องผ่านอุปกรณ์บายพาสหรือส่งผ่าน (bypass-transformer) เพื่อข้ามหม้อแปลงไฟฟ้าแรงดันปานกลาง หรือแรงดันต่ำไปยังสายไฟฟ้าแรงดันต่ำก่อนส่งต่อไปยังผู้ใช้ เนื่องจากสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า ซึ่งมีความถี่ช่วงที่สูงนั้นก็จะทำให้ไม่สามารถผ่านหม้อแปลงไฟฟ้าแรงดันปานกลางหรือแรงดันต่ำได้โดยตรงดังแสดงในภาพที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



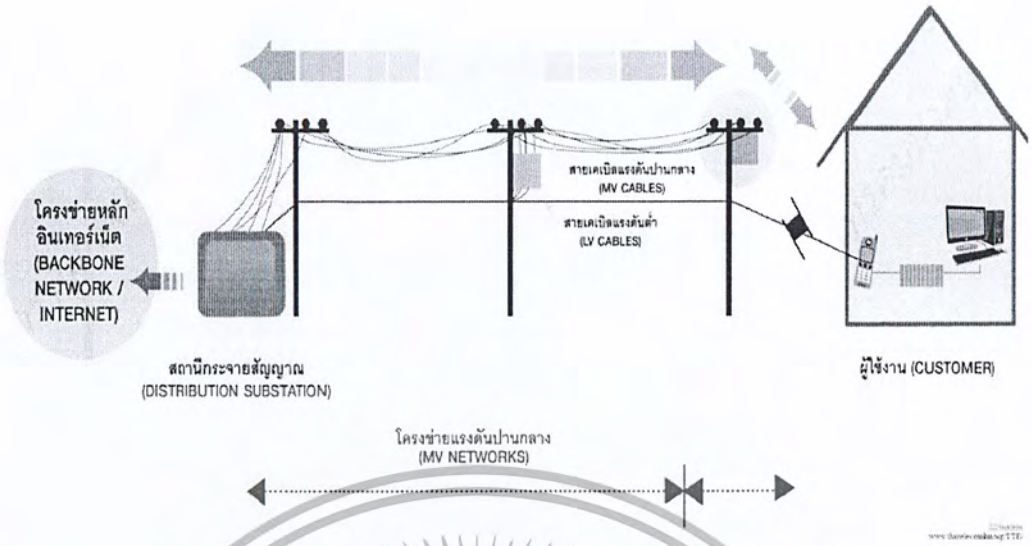

**ภาพที่ 2.5** สถาปัตยกรรมโครงข่ายสายไฟฟ้าแรงดันปานกลางต่ำและสายไฟฟ้าแรงดันต่ำ

ข้อดีของสถาปัตยกรรมโครงข่ายแบบนี้ คือ ผู้ให้บริการไฟฟ้าที่ไม่มีโครงข่ายเส้น ใยนำแสงขนาดใหญ่สามารถให้บริการการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าได้ แต่ก็จะมีข้อจำกัด คือ อุปกรณ์ส่งผ่านมีราคาสูง และเนื่องจากสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าถูกส่งผ่านทั้งสายไฟฟ้าแรงดันปานกลางและสายไฟฟ้าแรงดันต่ำ จึงเกิดการแพร่กระจายสัญญาณรบกวนไปยังระบบสื่อสารอื่น ๆ เป็นปริมาณมาก

- **สถาปัตยกรรมโครงข่ายสายไฟฟ้าแรงดันปานกลางและโครงข่ายไร้สาย (Wireless network)**

โครงข่ายนี้ข้อมูลจะถูกส่งจากโครงข่ายแกนหลักสายไฟฟ้าแรงดันปานกลางไปยังเสาไฟฟ้าที่มีหม้อแปลงไฟฟ้าแรงดันปานกลางหรือแรงดันต่ำติดตั้งอยู่ จากนั้นข้อมูลจะถูกส่งผ่านคลื่นวิทยุไปยังผู้ใช้แทน การส่งสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าแรงดันต่ำ ดังแสดงในภาพที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 2.6** สถาปัตยกรรมโครงข่ายสายไฟฟ้าแรงดันปานกลาง และโครงข่ายไร้สาย

ข้อดีของสถาปัตยกรรมโครงข่ายนั้นก็คือจะมีสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า ในโครงข่ายสายไฟฟ้าแรงดันปานกลางจะได้รับผลกระทบจากสัญญาณรบกวนน้อยกว่าและมีความน่าเชื่อถือสูงกว่าในโครงข่ายสายไฟฟ้าแรงดันต่ำ และเนื่องจากมีจำนวนโนด (node) ที่เชื่อมต่ออยู่น้อยกว่า นอกจากนั้นผู้ให้บริการไฟฟ้าก็ไม่ต้องลงทุนสูงในการสร้างโครงข่ายเส้นใยนำแสง แต่มีข้อจำกัด คือ จะต้องลงทุนติดตั้งอุปกรณ์สื่อสารไร้สายกับเสาอากาศจำนวนมาก และอาจมีความกว้างแถบความถี่ไม่เพียงพอสำหรับให้บริการ เนื่องจากอาจมีผู้ใช้บริการจำนวนมากต่อหนึ่งหม้อแปลงแบบไฟฟ้าแรงดันปานกลางหรือแรงดันต่ำ

### 2.1.4 ข้อจำกัดทางเทคนิค (Technical Limitations)

การใช้งานเทคโนโลยีการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า จะมีข้อจำกัดทางเทคนิคที่สำคัญดังต่อไปนี้

**- สัญญาณรบกวน**

ในช่วงความถี่ที่ใช้งานตามปกติของเทคโนโลยีการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า (1–40 เมกะเฮิรตซ์) ตรงกับช่วงความถี่ใช้งานของระบบวิทยุสมัครเล่น ระบบสื่อสารของทหาร วิทยุควบคุมการบิน ฯลฯ ดังนั้นประเด็นเรื่องความเข้ากันได้ทางแม่เหล็กไฟฟ้า (Electro Magnetic Compatibility : EMC) จึงมีความสำคัญสำหรับการใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า เนื่องจากสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า จะแผ่คลื่นแม่เหล็กไฟฟ้าไปรอบทิศทาง ซึ่งก่อให้เกิดการรบกวนระบบสื่อสารวิทยุอื่น ๆ โดยสายไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเปรียบเสมือนเป็นสายอากาศที่แพร่กระจายคลื่นแม่เหล็กไฟฟ้า เนื่องจากสายไฟฟ้าไม่มีการป้องกันการแพร่กระจายคลื่น (untwisted and unshielded) ซึ่งต่างจากสายเคเบิลของระบบสื่อสารโดยทั่วไป วิธีการแก้ปัญหาความเข้ากันได้ทางแม่เหล็กไฟฟ้านั้นจะกระทำโดยการจำกัดกำลังส่งของสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า แต่ก็จะมีผลทำให้ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนซึ่งแสดงคุณภาพของสัญญาณนั้นลดลงอย่างรวดเร็ว ดังนั้นวิธีการแก้ปัญหาที่เหมาะสมจึงควรจะสามารถแก้ปัญหาความเข้ากันได้ทางแม่เหล็กไฟฟ้า ไปพร้อม ๆ กับรักษาระดับค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนไม่ให้มีค่าต่ำเกินไป ซึ่งวิธีการการระบุความถี่ (Frequency notching) เป็นวิธีแก้ปัญหาที่เหมาะสม ซึ่งวิธีนี้จะจำกัดกำลังส่งของสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าให้เหลือต่ำที่สุดเฉพาะในบางช่วงความถี่ใช้งานที่ก่อให้เกิดการรบกวนกับระบบสื่อสารอื่น ๆ ที่กำลังใช้งานอยู่ ดังนั้นช่วงความถี่ใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า จึงลดลงเท่ากับช่วงความถี่ที่ระบุหรือว่ากำหนดไว้ (notched frequency)

เนื่องจากปัญหาความเข้ากันได้ทางแม่เหล็กไฟฟ้า จะทำให้ไม่สามารถส่งสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าด้วยกำลังส่งสูงได้ สัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าจึงมีความไวต่อการรบกวนในรูปแบบต่าง ๆ ทั้งจากระบบสื่อสารที่ใช้งานในช่วงความถี่เดียวกันกับสัญญาณสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า และจากเครื่องใช้ ไฟฟ้าขนาดใหญ่ เช่น สัญญาณรบกวนในสายไฟฟ้า ซึ่งเกิดจากเครื่องกำเนิดไฟฟ้า ดังนั้นจึงสามารถใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าได้ในระยะทางไม่ไกล (ประมาณ 2 กิโลเมตร ในโครงข่ายไฟฟ้าแรงดันปานกลางและ 250 เมตร ในโครงข่ายสายไฟฟ้าแรงดันต่ำ) อย่างไรก็ตามมีการใช้เทคนิคการเข้ารหัสแก้ไขความผิดพลาดล่วงหน้า (Forward Error Correction : FEC) เพื่อแก้ไขสัญญาณที่ถูกรบกวนให้ถูกต้องมากที่สุด และที่เป็นอีกแนวทางหนึ่งในการเพิ่มคุณภาพของสัญญาณ แต่ก็จะส่งผลให้ต้องใช้ความกว้างแถบความถี่เพื่อส่งข้อมูลเพิ่มขึ้น เนื่องจากเทคนิคนี้จะทำให้ข้อมูลมีขนาดเพิ่มขึ้น

- **กฎระเบียบในการใช้งานย่านความถี่**

เนื่องจากการใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า ก่อให้เกิดสัญญาณรบกวนต่อระบบสื่อสารวิทยุอื่น ๆ หน่วยงานที่มีหน้าที่กำกับดูแลการใช้งานคลื่นความถี่วิทยุจึงได้ออกกฎระเบียบเพื่อควบคุมการใช้งานการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้าเช่น FCC Part 15 Subpart G จากหน่วยงานเอฟซีซี (Federal Communications Commission : FCC) ประเทศสหรัฐอเมริกา และ CISPR standard 22 จากหน่วยงานซีไอเอสพีอาร์ (International special committee on radio interference : CISPR) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การประวิงแบบกลุ่ม (Group delay)**

การประวิงแบบกลุ่มของเทคโนโลยีการสื่อสารบรอดแบนด์ความเร็วสูงที่ผ่านสายไฟฟ้าเกิดขึ้นเมื่อสัญญาณการสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า ในช่วงความถี่ที่แตกต่างกันเกิดการประวิงและการลดทอนที่แตกต่างกัน ดังนั้นเมื่อสัญญาณไปถึงปลายทางจึงทำให้เกิดความผิดเพี้ยนเชิงแอมพลิจูดหรือขนาดและความผิดเพี้ยนเชิงเฟส ซึ่งจะทำให้เกิดอัตราบิตผิดพลาดเพิ่มสูงขึ้น

## 2.2 การมอดูเลตสัญญาณ (Signal Modulation)

เมื่อมีความต้องการที่จะส่งสัญญาณเสียงหรือข้อมูลผ่านช่องทางการสื่อสารจำเป็นอย่างยิ่งที่จะต้องอาศัยพลังงานไฟฟ้าช่วยพาสัญญาณเหล่านั้นให้เคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่งวิธีการหรือขั้นตอนข้างต้นในการเพิ่มพลังงานไฟฟ้าดังกล่าวเราเรียกว่าการมอดูเลต (Modulation) พลังงานไฟฟ้าซึ่งมีความถี่สูงและคงที่รวมทั้งมีแอมพลิจูด (ขนาด) สูงด้วยนั้นเราเรียกว่าสัญญาณคลื่นพาห์ (Signal Carrier)

อุปกรณ์สำหรับมอดูเลตสัญญาณ (Modulator) จะสร้างสัญญาณคลื่นพาห์ และรวมเข้ากับสัญญาณข้อมูลเพื่อให้สัญญาณมีความแรงพอที่จะส่งผ่านสื่อกลางไปยังอีกจุดหนึ่งที่อยู่ไกลออกไปได้และเมื่อถึงปลายทางก็จะมีอุปกรณ์ ซึ่งทำหน้าที่แยกสัญญาณคลื่นพาห์ออกให้เหลือเพียงสัญญาณข้อมูลเราเรียกวิธีการแยกสัญญาณนี้ว่าการดีมอดูเลต (Demodulation) เรื่องการมอดูเลตสัญญาณเป็นเรื่องที่สำคัญมากในการสื่อสารข้อมูลการเลือกวิธีการมอดูเลตและการดีมอดูเลตที่เหมาะสมจะช่วยให้ทำการส่งข้อมูลข่าวสารได้อย่างมีประสิทธิภาพ

### 2.2.1 การมอดูเลตสัญญาณอนาล็อก

วิธีการมอดูเลตสัญญาณอนาล็อกเพื่อส่งผ่านไปในช่องทางสื่อสารอนาล็อกนั้นมี 3 วิธีด้วยกัน คือ

- **การมอดูเลตทางแอมพลิจูด (AM)**

การมอดูเลตแบบการมอดูเลตทางแอมมพลิจูด นั้นจะเป็นวิธีการที่ดั้งเดิมและสะดวกที่สุดความถี่ของสัญญาณคลื่นพาห์จะคงที่และสูงกว่าความถี่ของสัญญาณข้อมูล เพื่อที่จะให้สามารถพาสัญญาณข้อมูลไปได้ในระยะทางไกล ๆ ซึ่งจะทำให้เห็นว่าได้ว่าสัญญาณ AM ที่มอดูเลตแล้วจะมีความถี่เท่ากับความถี่ของสัญญาณคลื่นพาห์โดยมีขนาด หรือแอมพลิจูดของสัญญาณเปลี่ยนแปลงไปตามแอมพลิจูดของสัญญาณข้อมูล

ข้อเสียของการมอดูเลตแบบ AM คือ แบนด์วิดท์ของสัญญาณ AM เป็นย่านความถี่ที่ไม่สูงนักทำให้สัญญาณรบกวน จากภายนอกสามารถเข้ามารบกวนได้ง่ายคลื่นพาห์จะตก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- การมอดูเลตทางความถี่ (FM)**

ตรงกันข้ามกับการมอดูเลตแบบมอดูเลตทางแอมมพลิจูด สัญญาณมอดูเลตแบบการมอดูเลตทางความถี่ ก็จะมีแอมพลิจูดคงที่แต่ความถี่ของสัญญาณจะไม่คงที่เปลี่ยนแปลงไปตามความถี่ของสัญญาณข้อมูล

ข้อเสียของการมอดูเลตแบบการมอดูเลตทางความถี่ คือ ต้องการแบนด์วิดท์ที่จะมีขนาด ที่กว้าง เนื่องจากสัญญาณข้อมูลมีหลายความถี่ดังนั้นจึงต้องหาวัสดุที่เป็นสายสื่อสารที่มีขนาดของแบนด์วิดท์กว้างทำให้ราคาของสายสื่อสารสูงขึ้นตามไปด้วย

**- การมอดูเลตทางเฟส (PM)**

ในขณะที่การมอดูเลตแบบมอดูเลตทางแอมมพลิจูด และ การมอดูเลตทางความถี่ เป็นที่นิยมใช้ในการกระจายเสียงทางวิทยุวิธีการมอดูเลตแบบการมอดูเลตทางเฟสกลับ นิยมใช้กันในการแพร่ภาพสีทางทีวีวิธีการมอดูเลตแบบการมอดูเลตทางเฟส ออกจะเป็นเรื่องที่ยุ่งยากกว่าแต่ก็เป็นวิธีที่ดีในการส่งสัญญาณข้อมูลคอมพิวเตอร์ที่ต้องความเร็วในการส่งข้อมูลสูงนิยมใช้การมอดูเลตแบบการมอดูเลตทางเฟส โดยส่งผ่านระบบโทรศัพท์ ในการมอดูเลตแบบการมอดูเลตทางเฟส ซึ่งครึ่งรอบของสัญญาณเราคิดเป็นมุมเฟสเท่ากับ 180 องศาและเมื่อครบรอบจะคิดเป็น 360 องศา สัญญาณมอดูเลตจะมีการเปลี่ยนมุมของเฟสทุกครั้งที่มุมเฟสของสัญญาณข้อมูลต่างจากมุมเฟสของสัญญาณคลื่นพาห์เท่ากับ 180 องศา

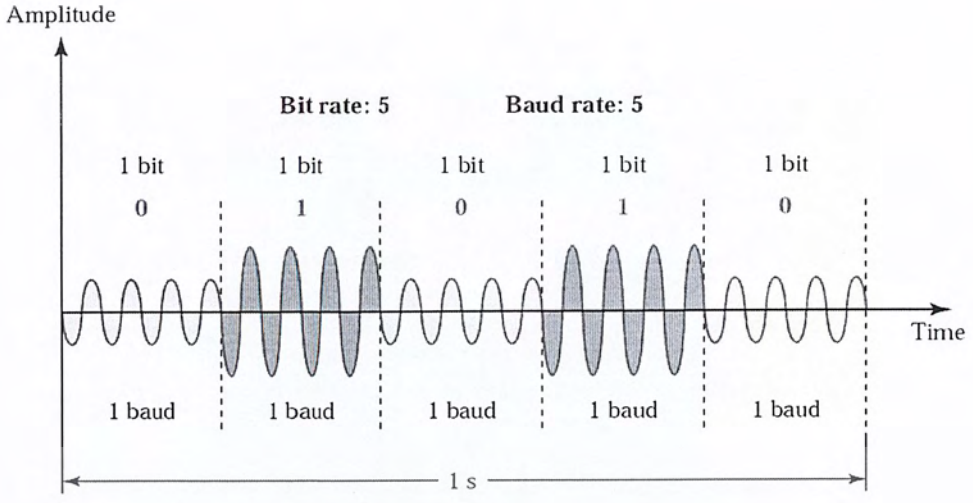
**- การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก (D/A)**

ในปัจจุบันนั้นการส่งสัญญาณแบบข้อมูลดิจิตอลโดยผ่านช่องทางสื่อสารแบบอนาล็อกที่เราคุ้นเคยกัน กับการส่งข้อมูลคอมพิวเตอร์ผ่านทางเครือข่ายโทรศัพท์สาธารณะเครือข่ายโทรศัพท์ถูกออกแบบมาเพื่อทำการสลับสวิตช์ และจะส่งสัญญาณอนาล็อกซึ่งเป็นย่านความถี่ของเสียงหรือประมาณ 300-3400 เฮิรตซ์ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่แปลงสัญญาณข้อมูลดิจิตอลให้เป็นสัญญาณอนาล็อกย่านความถี่เสียงเราเรียกว่าโมเด็ม ( MODEM : Modulator – DEModulator ) การแปลงสัญญาณข้อมูลดิจิตอลให้เป็นสัญญาณอนาล็อกนั้นมีอยู่ด้วยกัน 3 วิธีคือ

**1. การมอดูเลตเชิงเลขทางแอมพลิจูด (ASK)**

ความถี่ของคลื่นพาห์ (Carrier Wave) ซึ่งก็จะทำหน้าที่นำสัญญาณอนาล็อกผ่านตัวกลางสื่อสารนั้นจะคงที่ ลักษณะของสัญญาณมอดูเลตเมื่อค่าของบิตของสัญญาณข้อมูลดิจิตอลมีค่าเป็น 1 ขนาดของคลื่นพาห์จะสูงขึ้นกว่าปกติ และเมื่อบิตมีค่าเป็น 0 ขนาดของลงกว่าปกติการมอดูเลตเชิงเลขทางแอมพลิจูด มักจะไม่ค่อยได้รับความนิยมเพราะจะถูกรบกวนจากสัญญาณอื่นได้ง่าย

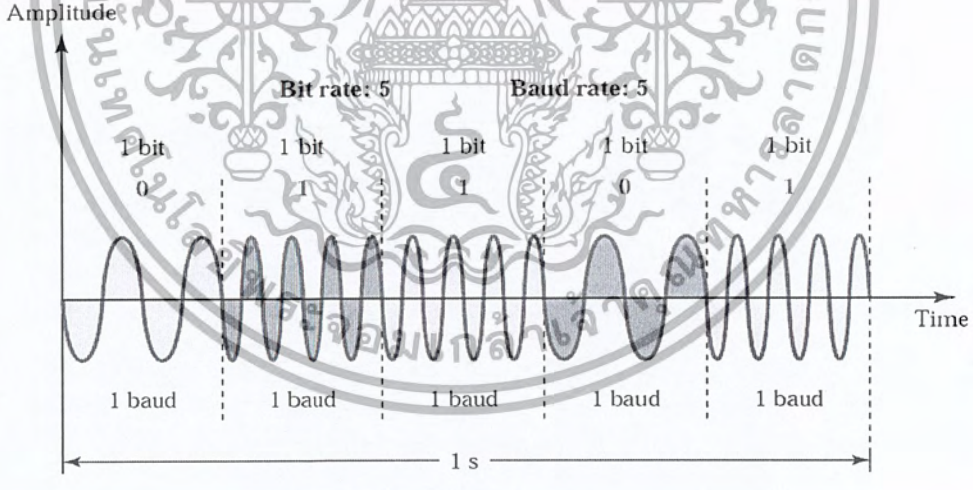
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 2.7** การมอดูเลตเชิงเลขทางแอมพลิจูด

**2. การมอดูเลตเชิงเลขทางความถี่ (FSK)**

ในการมอดูเลตแบบการมอดูเลตเชิงเลขทางความถี่ ขนาดของคลื่นพาห์จะไม่มีการเปลี่ยนแปลง แต่ส่วนที่จะเปลี่ยนแปลง คือ ความถี่ของคลื่นพาห์นั้น เมื่อบิตมีค่าเป็น 1 ความถี่ของคลื่นพาห์จะสูงกว่าปกติและเมื่อบิตมีค่าเป็น 0 ความถี่ของคลื่นพาห์ก็จะต่ำกว่าปกติ



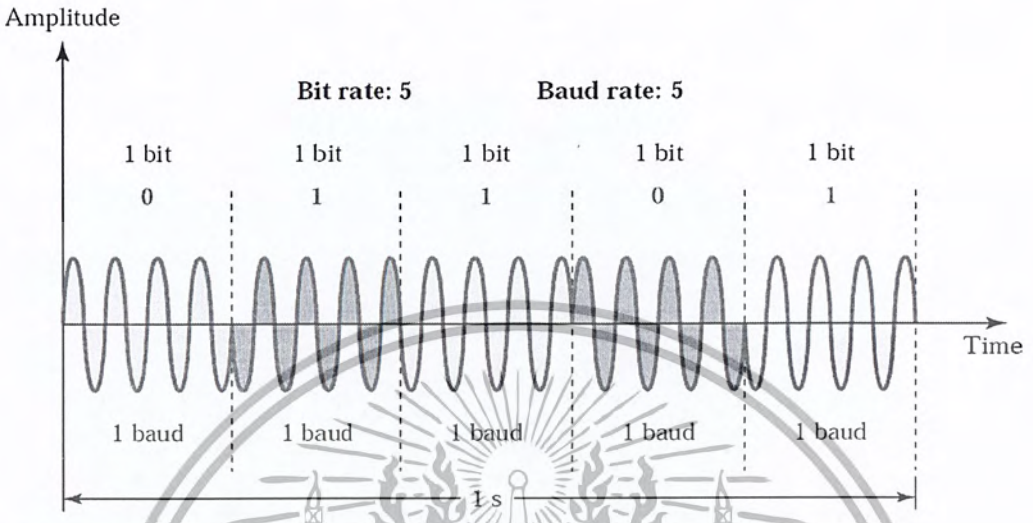

**ภาพที่ 2.8** การมอดูเลตเชิงเลขทางความถี่

**3. การมอดูเลตเชิงเลขทางเฟส (PSK)**

หลักการของแบบมอดูเลตเชิงเลขทางเฟสคือค่าของขนาดและความถี่ขอคลื่นพาห์จะไม่มีการเปลี่ยนแปลงแต่ที่จะเปลี่ยน คือ เฟสของสัญญาณ โดยเมื่อมีการเปลี่ยนแปลงสภาวะของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตจาก 1 ไปเป็น 0 หรือเปลี่ยนจาก 0 ไปเป็น 1 เฟสของคลื่นจะเปลี่ยน (Shift) ไป 180 องศา ด้วย โดยที่หลักการของการมอดูเลตเชิงเลขทางเฟสนั้นสามารถทำได้ทั้งแบบ 2 เฟส ดังต่อไปนี้ (0, 90,180 และ 270 องศา) และแบบ 8 เฟส (0, 45, 90, 135, 180, 225, 270 และ 315 องศา)




**ภาพที่ 2.9** การมอดูเลตเชิงเลขทางเฟส

## 2.3 ความรู้ทั่วไปเกี่ยวกับ X10

X10 คือ มาตรฐานการสื่อสารของอุปกรณ์ไฟฟ้า อุปกรณ์ส่องสว่าง ระบบรักษาความปลอดภัย อุปกรณ์เพื่อความสะดวกสบาย รวมถึงอุปกรณ์เพื่อความบันเทิงอย่างทีวี เครื่องเสียง ภายในบ้านโดยผ่านทางสายไฟฟ้า 220Vac/50Hz ที่มีอยู่ภายในบ้าน ดังนั้นไม่ว่าจะเป็นบ้านที่สร้างขึ้นใหม่ หรือบ้านที่อยู่อาศัยมานานแล้วก็สามารถทำได้ง่ายโดยไม่ต้องเดินสายไฟเพิ่มเติม ทำให้สามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้อย่างสมบูรณ์แบบ และมีอุปกรณ์ให้เลือกหลากหลายตามความต้องการ

### 2.3.1 ประวัติความเป็นมา

เทคโนโลยี X10 เริ่มแรกนิยมใช้ในประเทศอเมริกาและเริ่มแพร่หลายในประเทศแถบยุโรป รวมทั้งมีการเจริญเติบโตอย่างรวดเร็วในประเทศแถบเอเชีย เนื่องจากเริ่มมีการผลิตให้เหมาะสมกับกระแสไฟฟ้า 220Vac/50Hz ซึ่งสามารถใช้ในประเทศไทยได้ลักษณะการใช้งานจะเป็นแบบ plug-in คือ สามารถใช้งานได้ทันทีโดยไม่ต้องมีการติดตั้งหรือเดินสายไฟเพิ่ม

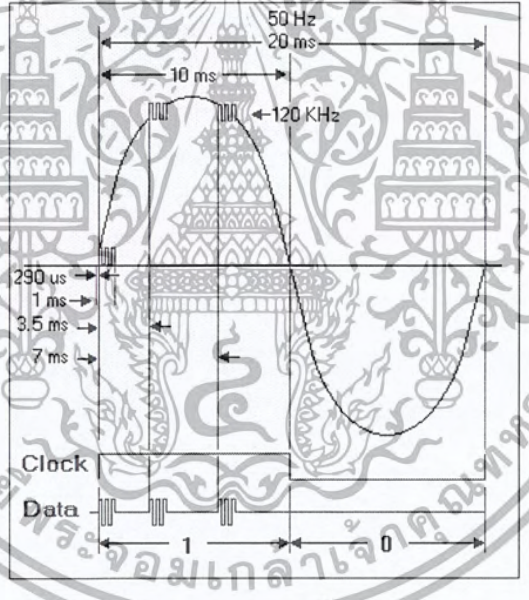
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ทำไม X10 จึงเป็นที่นิยมเพราะมีอุปกรณ์ให้เลือกใช้หลากหลาย สามารถใช้ต่อขยายได้อย่างไร้ขีดจำกัด ติดตั้งง่ายโดยไม่ต้องเดินสายไฟเพิ่มทำให้ลดค่าใช้จ่ายในการติดตั้ง ง่ายและสะดวกในการควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน

### 2.3.2 ทฤษฎีของ X10

การส่งข้อมูลของ X10 จะอาศัยการสร้างสัญญาณนาฬิกาที่ได้จากจุดตัดศูนย์ โดยที่ค่าของข้อมูลไบนารี 1 จะส่งสัญญาณ Burst มีความถี่ 120 กิโลเฮิรตซ์ จำนวน 3 ลูก หลังจากจุดตัดศูนย์ โดยไม่จำเป็นว่าจะเป็นขอบขาขึ้นหรือขอบขาลงของสัญญาณนาฬิกา และค่าของข้อมูลไบนารี 0 จะเป็นการเว้นสัญญาณของช่องว่างของสัญญาณเบิซท์ ดังนั้นใน 1 ไซเคิลจะสามารถส่งข้อมูลได้ 2 บิต รายละเอียดของระยะเวลาสามารถดูได้จากภาพที่ 2.1




**ภาพที่ 2.10** การส่งข้อมูลบน AC-Line ของ X10 ที่ความถี่ 50 เฮิรตซ์

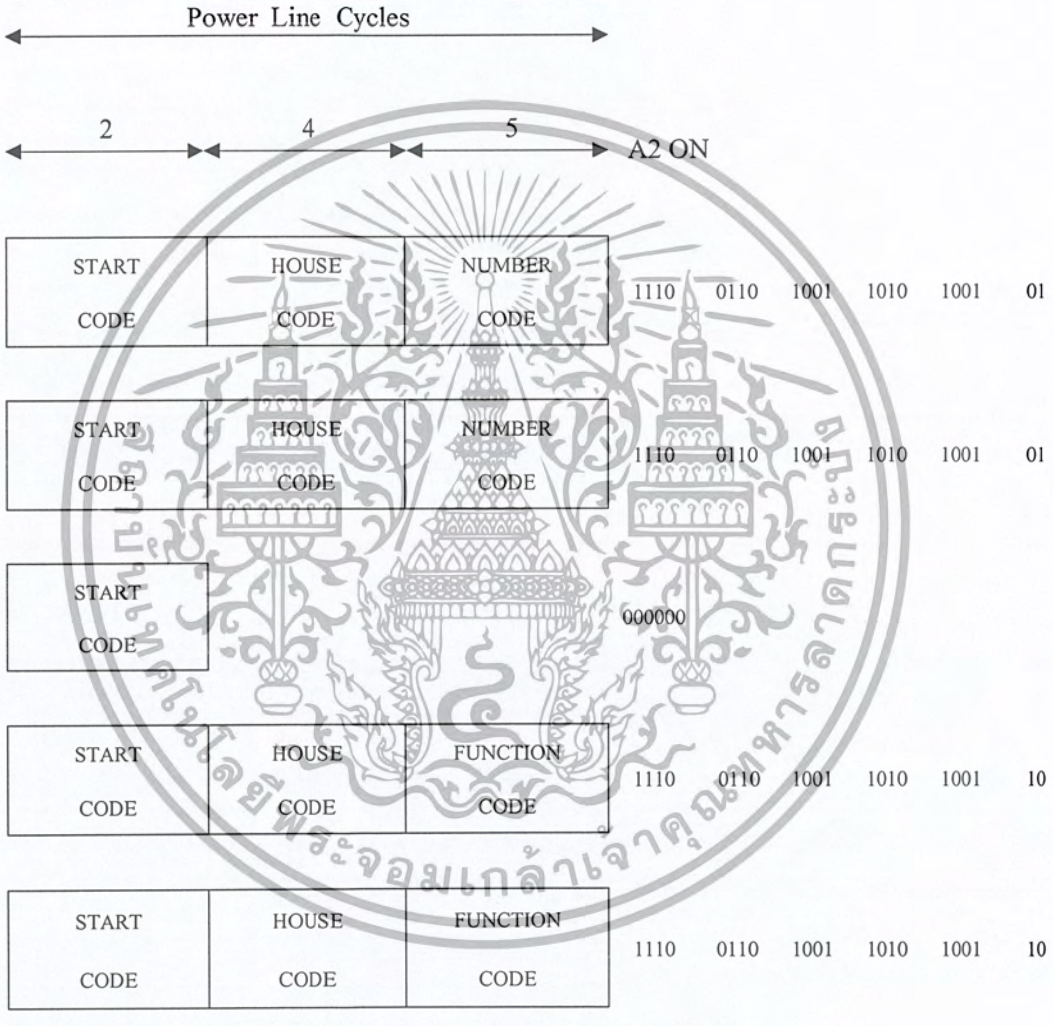
**- โครงสร้างของ Block ข้อมูล**

โครงสร้างของ Block ข้อมูลจะประกอบไปด้วย Start Code, House Code และ Key Code คือ เมื่อ Block ข้อมูลระบุแอดเดรส Key Code จะหมายถึง Number Code แต่ถ้า Block ข้อมูลระบุฟังก์ชัน Key Code จะหมายถึง Function Code โดยที่ Start Code เป็นข้อมูลไบนารี 4 บิต คือ 1110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
119192

- **วิธีการส่ง Block ข้อมูลชุดคำสั่ง**

Block ข้อมูลระบุแอดเดรสจะต้องตามด้วย Block ข้อมูลฟังก์ชัน เช่น Block ข้อมูลระบุแอดเดรสนั้นจะประกอบด้วย Start Code, House Code และ Number Code (ส่ง 2 ครั้งติดต่อกัน) ช่องว่าง (GAP) คือ ข้อมูล 000000 (เว้นอย่างน้อย 3 ไซเคิล) และ Block ข้อมูลระบุฟังก์ชั่นประกอบด้วย Start Code, House Code และ Function Code (ส่ง 2 ครั้งติดต่อกัน)




**ภาพที่ 2.11** ชุดคำสั่งระบุแอดเดรสและฟังก์ชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.1** House Code

| House | House Codes | | | |
|---|---|---|---|---|
| **Addresses** | **H1** | **H2** | **H4** | **H8** |
| A | 0 | 1 | 1 | 0 |
| B | 1 | 1 | 1 | 0 |
| C | 0 | 0 | 1 | 0 |
| D | 1 | 0 | 1 | 0 |
| E | 0 | 0 | 0 | 1 |
| F | 1 | 0 | 0 | 1 |
| G | 0 | 1 | 0 | 1 |
| H | 1 | 1 | 0 | 1 |
| I | 0 | 1 | 1 | 1 |
| J | 1 | 1 | 1 | 1 |
| K | 0 | 0 | 1 | 1 |
| L | 1 | 0 | 1 | 1 |
| M | 0 | 0 | 0 | 0 |
| N | 1 | 0 | 0 | 0 |
| O | 0 | 1 | 0 | 0 |
| P | 1 | 1 | 0 | 0 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

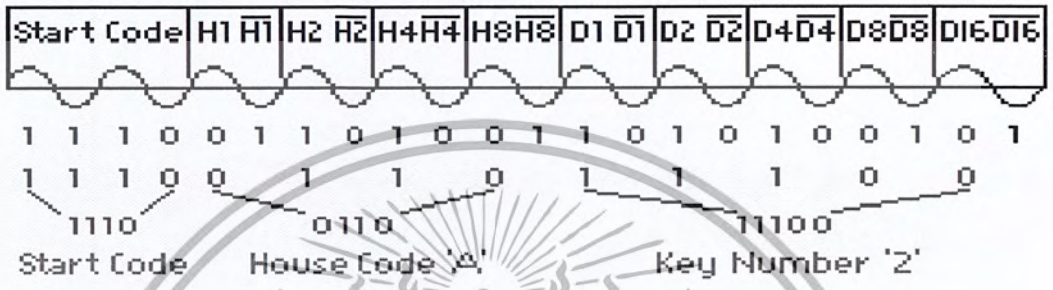
**ตารางที่ 2.2** Key Code

| Unit address | Key Codes | | | | |
|---|---|---|---|---|---|
| | Number Code | | | | |
| | D1 | D2 | D4 | D8 | D16 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 2 | 1 | 1 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 |
| 4 | 1 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 1 | 0 |
| 6 | 1 | 0 | 0 | 1 | 0 |
| 7 | 0 | 1 | 0 | 1 | 0 |
| 8 | 1 | 1 | 0 | 1 | 0 |
| 9 | 0 | 1 | 1 | 1 | 0 |
| 10 | 1 | 1 | 1 | 1 | 0 |
| 11 | 0 | 0 | 1 | 1 | 0 |
| 12 | 1 | 0 | 1 | 1 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 |
| 14 | 1 | 0 | 0 | 0 | 0 |
| 15 | 0 | 1 | 0 | 0 | 0 |
| 16 | 1 | 1 | 0 | 0 | 0 |
| **Function Codes** | **Function Code** | | | | |
| All Units Off | 0 | 0 | 0 | 0 | 1 |
| All Units On | 0 | 0 | 0 | 1 | 1 |
| On | 0 | 0 | 1 | 0 | 1 |
| Off | 0 | 0 | 1 | 1 | 1 |
| Dim | 0 | 1 | 0 | 0 | 1 |
| Bright | 0 | 1 | 0 | 1 | 1 |
| All Lights Off | 0 | 1 | 1 | 0 | 1 |
| Extended Code | 0 | 1 | 1 | 1 | 1 |
| Hail Request | 1 | 0 | 0 | 0 | 1 |
| Hail Acknowledge | 1 | 0 | 0 | 1 | 1 |
| Pre-set Dim | 1 | 0 | 1 | X | 1 |
| Extended Code | 1 | 1 | 0 | 0 | 1 |
| (Analog) | | | | | |
| Status = On | 1 | 1 | 0 | 1 | 1 |
| Status = Off | 1 | 1 | 1 | 0 | 1 |
| Status Request | 1 | 1 | 1 | 1 | 1 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- วิธีการส่งข้อมูลของ House code และ Key code**

หลังจากที่ส่ง Start Code ไปแล้วนั้นส่วนของ Key Code (Number code และ Function code) จะต้องส่งแบบบิตแล้วตามด้วย compliment (ส่วนกลับ) ของบิตนั้น เช่น House code A = 0110 compliment = 1001 และ Number Code 2 =11100 มี compliment = 00011 ดังนั้นการส่งข้อมูลแอดเดรสของ A2 จะเป็น 1110 + 01101001 + 1010100101




**ภาพที่ 2.12** การส่งข้อมูลแบบบิตแล้วตามด้วย compliment ของบิตนั้น (ไม่รวม Start Code)

## 2.4 สภาพแวดล้อมของสายไฟฟ้ากำลัง

เนื่องจากสายส่งของระบบไฟฟ้ากำลังถูกออกแบบมาเพื่อใช้ในการส่งสัญญาณไฟฟ้ากำลังที่ความถี่ 50 Hz เท่านั้น โดยไม่คำนึงถึงการนำมาส่งสัญญาณสื่อสารความถี่สูง ดังนั้นปัญหาสำคัญที่อาจจะเกิดขึ้นกับการส่งสัญญาณสื่อสารดังนี้

### 2.4.1 ปัญหาการลดทอนของสัญญาณสื่อสาร

เกิดจากการเปลี่ยนแปลงอิมพีแดนซ์ของสายไฟ ซึ่งอิมพิแดนซ์จะเปลี่ยนแปลงเนื่องมาจากอุปกรณ์ที่ต่อเข้ากับระบบไฟฟ้า การที่จุดเชื่อมต่อของสายไฟฟ้ามีอิมพิแดนซ์ไม่เท่ากับสัญญาณสื่อสารบางส่วนจะสะท้อนกลับทำให้เกิดการลดทอน จากการติดตั้งสายไฟฟ้าหลายเส้นทาง เนื่องจากระบบไฟฟ้ากำลังจะมีการส่งสัญญาณไฟฟ้าไปตามสายไฟฟ้าหลายเส้นทาง และความยาวของสายไฟฟ้าก็จะต่างกัน สัญญาณสื่อสารที่ส่งไปในสายไฟฟ้าเส้นต่าง ๆ จะมีเฟสต่างกัน หากมาบรรจบกันได้แล้ว สัญญาณอาจถูกลดทอนหรืออาจหายไปที่จุดบรรจบนั้นมีสัญญาณสื่อสารจากสองเส้นทางที่มีเฟสต่างกัน 180 องศาพอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.2 ปัญหาสัญญาณรบกวนต่าง ๆ ในระบบไฟฟ้ากำลัง

ในระบบไฟฟ้ากำลังมีสัญญาณรบกวนมากมายที่สำคัญ คือ สัญญาณรบกวนจากการทำงานของวงจร DIMMER ทำให้เกิดไปรส์ขนาด 20 V ถึง 50 V ในช่วงความถี่ 100 เฮิรตซ์ ถึง 120 เฮิรตซ์ สัญญาณรบกวนจาก SWITCHING POWER SUPPLY จะทำให้เกิดสัญญาณรบกวนในช่วงความถี่ 20 กิโลเฮิรตซ์ ถึง 1 เมกะเฮิรตซ์ สัญญาณจากการส่งของ INTERCOM ซึ่งจะส่งคลื่นพาห์ในช่วงความถี่ 150 kHz ถึง 500 kHz และจากการทำงานของโหลดประเภทอินดักทีฟจะเกิดสัญญาณรบกวนอิมพัลส์ขึ้นในระบบสายส่งระบบไฟฟ้ากำลัง

อิมพีแดนซ์ภายในสายส่งมีค่าประมาณ 10 – 100 โอห์ม ขนาดของตัวเก็บประจุภายในสายประมาณ 10 – 1000 พิโคฟารัด ต่อ 1000 ฟุต ขนาดความต้านทานในสายประมาณ 0.1 – 1 โอห์ม ต่อ 1000 ฟุต ความต้านทานในสายยังไม่มีความแน่นอนขึ้นอยู่กับจำนวนอุปกรณ์ไฟฟ้าอีกด้วย

## 2.5 มาตรฐานที่ต้องคำนึงถึงในการสื่อสารบนสายไฟฟ้ากำลัง

มาตรฐาน EN50065-1 นั้นเป็นมาตรฐานที่เกี่ยวข้องกับการสื่อสารบนสายไฟฟ้ากำลังมีข้อกำหนดดังนี้

### 2.5.1 การเลือกความถี่

มาตรฐาน EN50065-1 กำหนดความถี่ของการส่งสัญญาณสื่อสารบนสายไฟฟ้ากำลังไว้ที่ความถี่ 3 กิโลเฮิรตซ์ ถึง 148 กิโลเฮิรตซ์ ซึ่งออกเป็น 4 ช่วงความถี่ดังนี้

**ตารางที่ 2.3** ช่วงความถี่ที่ใช้ได้ตามมาตรฐาน EN50065-1

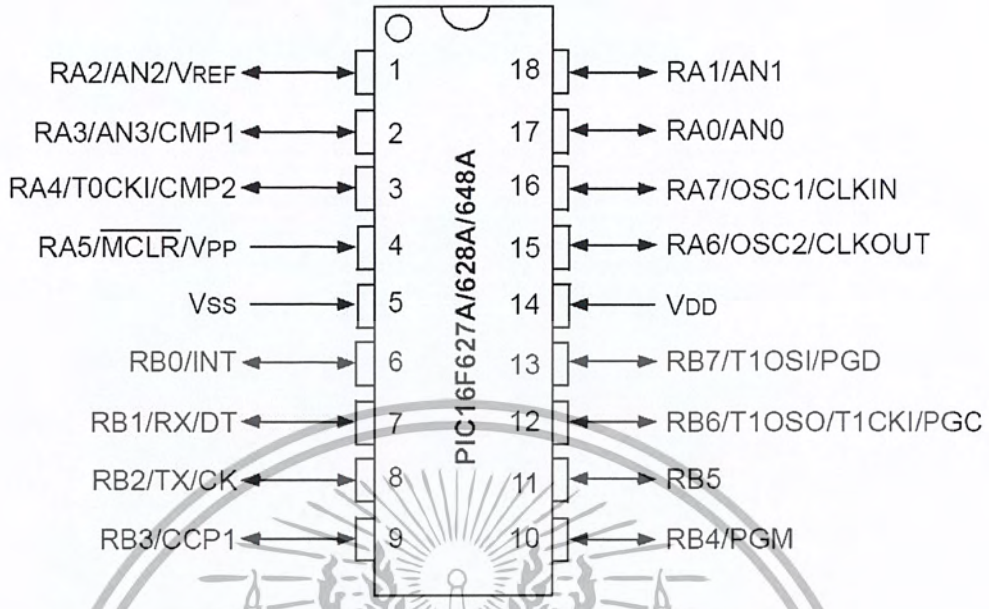
| Electricity suppliers<br>3 kHz – 95 kHz | Home systems<br>95 kHz – 125 kHz | Home system<br>125 kHz – 140 kHz | Alarm & security<br>140 kHz – 148 kHz |
|---|---|---|---|
| | Free access | Access rule | |

### 2.5.2 ค่าแรงดันในการส่ง

ค่าแรงดันในการส่งภายในที่อยู่อาศัยกำหนดไว้ไม่เกิน 3.56 Vpp ส่วนในสถานที่พิเศษ เช่น โรงงานอุตสาหกรรมถูกกำหนดไว้ไม่เกิน 23.8 Vpp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 ไมโครคอนโทรเลอร์




**ภาพที่ 2.13** PIC16F648A

ไมโครคอนโทรเลอร์ PIC16F648A เป็นไมโครคอนโทรเลอร์ที่มีขนาดเล็ก และเป็นไอซี 18 ขา ที่มีการใช้งานที่ง่าย โดยที่จะมีรูปแบบคำสั่งและสถาปัตยกรรมโดยทั่วไปเหมือนกับไมโครคอนโทรลเลอร์ตระกูล PIC16F

### 2.6.1 คุณสมบัติของ PIC16F648A

- มีคำสั่งให้ใช้งาน 35 คำสั่ง
- คำสั่งหนึ่งๆใช้เวลาทำงาน 1 ถึง 2 ไซเคิล
- หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 4KWord (1 word=14 บิต)
- มี RAM ขนาด 256 ไบต์ ให้เราใช้งาน
- มี EEPROM ขนาด 256 ไบต์
- ตอบสนองกับอินเตอร์รัพได้ทั้งหมด 14 แหล่ง
- มี Stack ให้ใช้ได้สูงสุด 8 ระดับ
- มีระบบ Power On Reset, Power Up Timer, Oscillator Start-up timer
- Watchdog timer
- มีระบบ Code Protection
- มีโหมดประหยัดพลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งานคือ อาจจะใช้ XTAL หรือ วงจร RC ก็ได้
- สามารถโปรแกรมด้วยไฟ +5VDC ได้
- ใช้การโปรแกรมแบบ In-Circuit Serial Programming
- ทำงานที่ไฟเลี้ยง 2VDC ถึง 5.5VDC
- Current Sink และ Current Source อยู่ที่ 25mA
- มี Timer/Counter
- มีโมดูล Capture/Compare/PWM
- มี A-TO-D Converter แบบ 10 บิต
- มีระบบ USART สำหรับต่อกับ การสื่อสารแบบ RS232 หรือดีกว่า
- มีระบบตรวจระดับไฟเลี้ยง (Brown-out reset)

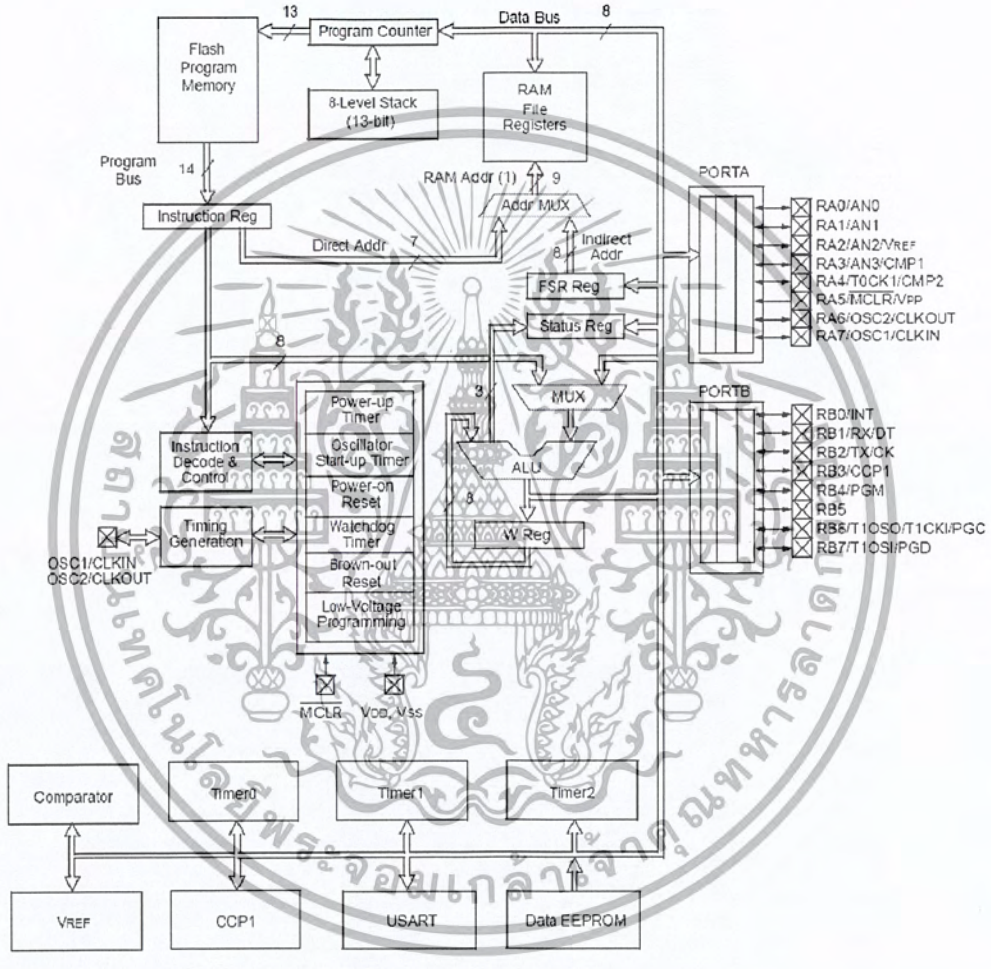
**2.6.2 หน้าที่การทำงานของแต่ละขา**

- OSC1/CLKIN : Oscillator crystal input /External clock source input.
- OSC2/CLKOUT : Oscillator crystal output.
  ทั้งสองขานี้จะต่อกับ crystal หรือ resonator ในกรณีที่อยู่ใน crystal oscillator mode (ใช้สัญญาณนาฬิกาจากภายนอก)
- MCLR(inv) : Master clear(reset)input /Programming voltage input.
  เมื่อขานี้เป็น LOW แล้ว MCU จะถูก reset อีกหน้าที่หนึ่งของขานี้ก็คือ จะเป็น input ของ voltage programming ขณะที่ทำการ program ตัว MCU
- RA0 - RA3 : Bi-directional I/O port. เป็นพอร์ตแบบ สองทิศทาง คือ เลือกให้เป็นINPUTหรือOUTPUTก็ได้อย่างใดอย่างหนึ่ง
- RA4/T0CKI : Bi-directional I/O port. เป็นพอร์ตแบบสองทิศทาง คือ เลือกให้เป็น INPUT หรือ OUTPUT ก็ได้อย่างใดอย่างหนึ่ง สัญญาณนาฬิกาเพื่อป้อนให้กับ Timer 0 ซึ่งอยู่ภายใน MCU ในกรณีที่เลือกว่าแหล่งของสัญญาณนาฬิกาที่ป้อนให้กับ Timer 0 ให้ใช้จากภายนอก
- RB0/INT : Bi-directional I/O port. เป็นพอร์ตแบบสองทิศทาง คือ เลือกให้เป็น INPUT หรือ OUTPUT ก็ได้อย่างใดอย่างหนึ่ง อีกหน้าที่หนึ่งก็คือ External interrupt pinรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ interrupt เมื่อเกิดการเปลี่ยนแปลงของสัญญาณที่ขานี้

- RB1 - RB7 : Bi-directional I/O port. เป็นพอร์ตแบบสองทิศทาง คือ เลือกให้เป็น INPUT หรือ OUTPUT ก็ได้อย่างใดอย่างหนึ่ง
- VSS : Ground
- VDD : Positive supply (+2.0V to +5.5V)



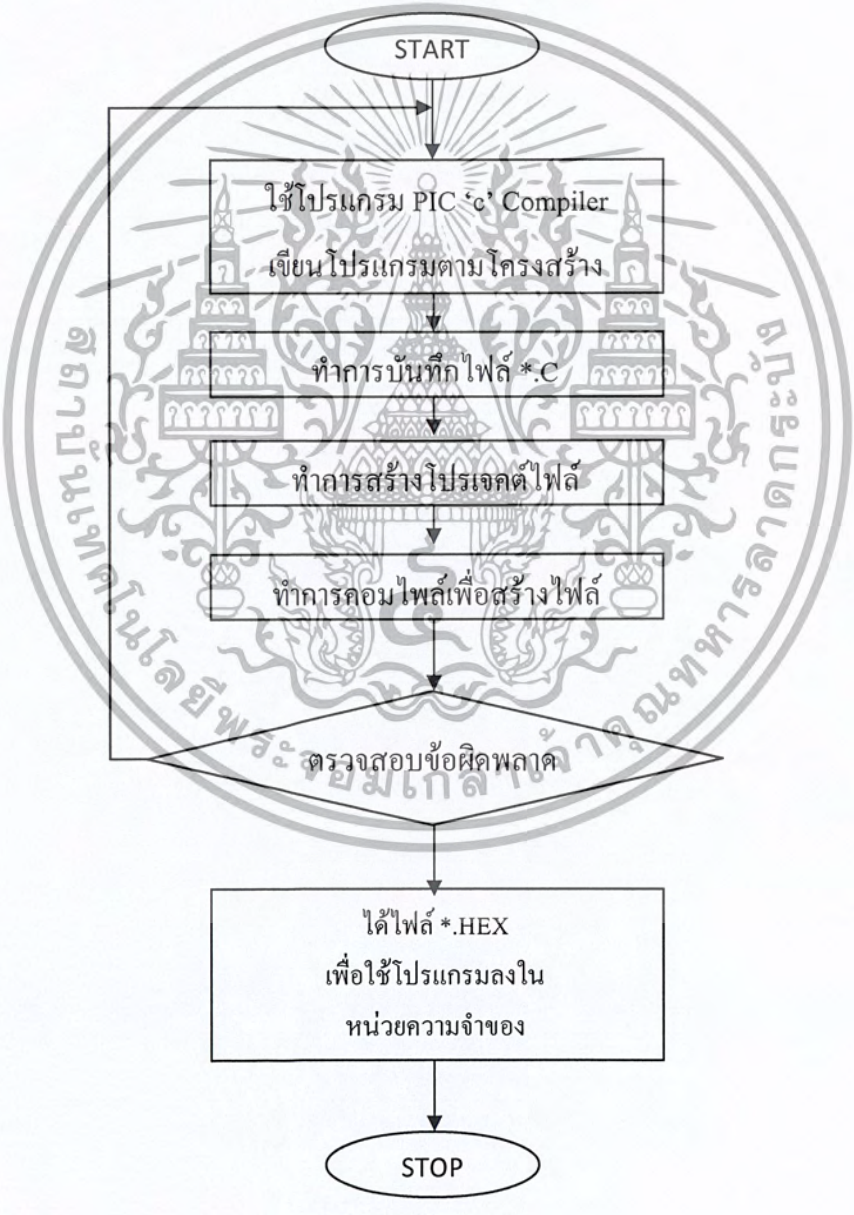

Note 1: Higher order bits are from the Status register.

**ภาพที่ 2.14** โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC เบอร์ PIC16F648A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

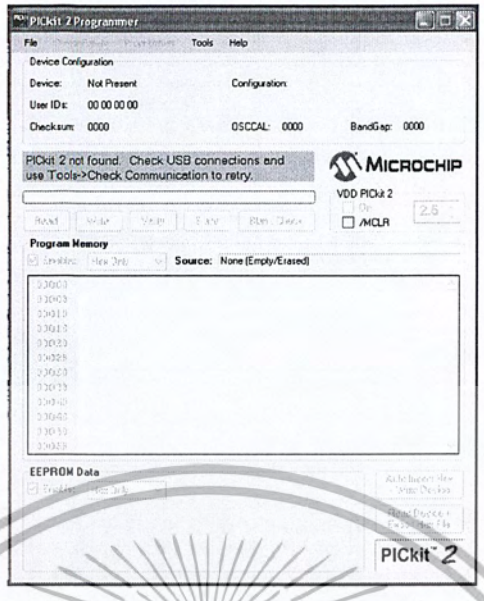
### 2.6.3 ขั้นตอนการพัฒนาโปรแกรม

การพัฒนาโปรแกรมเริ่มจากใช้โปรแกรม PIC 'C' Compiler เขียนโปรแกรมโดยใช้โครงสร้างโปรแกรมภาษาซี จากนั้นทำการบันทึก สร้างโปรเจคต์ไฟล์ และทำการคอมไพล์โปรแกรม เมื่อทำการคอมไพล์โปรแกรมเสร็จและได้ไฟล์ฐานสิบหก ซึ่งเป็นไฟล์ *.HEX จากนั้นจึงนำไปเขียนหรือ โปรแกรมลงในหน่วยความจำของไมโครคอนโทรลเลอร์โดยใช้เครื่องโปรแกรมไมโครคอนโทรเลอร์ หรือการโปรแกรมผ่านพอร์ตอนุกรม ซึ่งแสดงขึ้นตอนการพัฒนาโปรแกรมดังภาพ

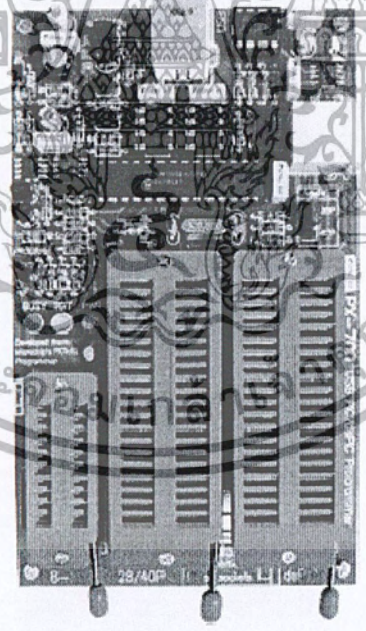



**ภาพที่ 2.15** โฟลว์ชาร์ตขั้นตอนการพัฒนาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 2.16** โปรแกรม PICkit สำหรับโปรแกรมลงในไมโครคอนโทรลเลอร์ตระกูล PIC



**ภาพที่ 2.17** บอร์ดสำหรับโปรแกรมไมโครคอนโทรลเลอร์ตระกูล PIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 การสื่อสารผ่านทางพอร์ตอนุกรม (Serial Port Communication)

ในอดีตการเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์จะมีความเฉพาะตัวเป็นอย่างมาก โดยควบคุมผ่านภาษาแอสแซมบลีซึ่งเป็นภาษาที่ผูกติดกับฮาร์ดแวร์มาก แม้จะให้ผลการทำงานดีแต่การเขียนโปรแกรมและการแก้ไขโปรแกรมนั้นทำได้ยาก เพราะภาษาแอสแซมบลีเป็นภาษาที่เข้าใจยาก และเป็นภาษาที่ไม่เป็นโครงสร้าง ต่อมาได้มีการนำโปรแกรมใหม่ ๆ เพื่อนำมาใช้ในการเขียนโปรแกรมไม่ว่าจะเป็นภาษาซี ปาสคาลซึ่งภาษาซีได้รับความนิยมมาก เพราะจะมีประสิทธิภาพใกล้เคียงกับภาษาแอสแซมบลี และเข้าใจได้ง่ายกว่า แต่ในปัจจุบันการเขียนโปรแกรมส่วนใหญ่จะเป็นการเขียนโปรแกรมแบบติดต่อผู้ใช้ ซึ่งจะเขียนได้ง่ายและรวดเร็ว ทำให้ภาษาโปรแกรมอย่างวิชวล เบสิกเป็นที่นิยมอย่างมาก ในการนำมาใช้ติดต่อควบคุมฮาร์ดแวร์ ปัจจุบันได้รับการพัฒนาให้ครอบคลุมเกือบหมดแล้ว

### 2.7.1 พื้นฐานการสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมในเครื่องคอมพิวเตอร์นั้น จะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน ทั้งนี้เนื่องจากการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตขนานนั้นสามารถส่งข้อมูลได้ครั้งละหลาย ๆ บิตพร้อมกัน ทำให้ผลการสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน

**ตารางที่ 2.4** ตารางแสดงความแตกต่างของพอร์ตขนานและพอร์ตอนุกรม

| พอร์ตขนาน (Parallel Port) | พอร์ตอนุกรม (Serial Port) |
|---|---|
| สร้างวงจรแล้วไม่สามารถใช้งานร่วมกันได้ | สร้างวงจรแล้วบางวงจรจะไม่สามารถใช้งานร่วมกันได้ |
| พอร์ตขนานมีการส่งข้อมูลแบบขนานคือได้ทีละหลาย ๆ บิต พร้อม ๆ กัน | พอร์ตอนุกรมมีการส่งข้อมูลแบบทีละบิต |
| การส่งข้อมูลพอร์ตขนานมีความรวดเร็วในการส่งข้อมูลแต่ละชุด | การส่งข้อมูลแบบอนุกรมต้องใช้ระยะเวลานานในกาส่งข้อมูลแต่ละชุด |
| การส่งข้อมูลของพอร์ตขนานเป็นการส่งข้อมูลในระยะที่ไม่ไกลมากนัก | การส่งข้อมูลได้ในระยะทางไกล ๆ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่การส่งข้อมูลแบบอนุกรมนั้นมีข้อที่เหนือกว่าการส่งข้อมูลแบบขนาน คือ การสามารถส่งข้อมูลได้ระยะทางที่ไกลกว่าแบบขนาน อีกทั้งสายสัญญาณที่ใช้ยังมีน้อยกว่าการส่งข้อมูลแบบขนานอีกด้วย

การสื่อสารแบบอนุกรมสามารถแบ่งออกได้เป็น 3 รูปแบบ คือ

1. Simplex สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว
2. Half- Duplex สามารถส่งข้อมูลไปยังปลายทางและสามารถรับข้อมูลจากปลายทางได้ แต่ไม่สามารถส่งข้อมูลและรับข้อมูลได้ในเวลาเดียวกัน
3. Full- Duplex สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

### 2.7.2 รูปแบบของการสื่อสารข้อมูลอนุกรม

สามารถแบ่งประเภทการสื่อสารตามลักษณะสัญญาณในการส่งได้ 2 แบบคือ

- **การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส**

เป็นวิธีการรับและส่งข้อมูล โดยไม่ต้องอาศัยสัญญาณนาฬิกาส่งร่วมไปด้วย แต่จะใช้อัตราความเร็วของจำนวนข้อมูลต่อวินาที และจะทำการเพิ่มบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง เพื่อจะได้ทำการตรวจสอบข้อมูลได้อย่างถูกต้องมากยิ่งขึ้นแสดงดังรูปที่ ซึ่งประกอบด้วยกัน 4 ส่วน คือ

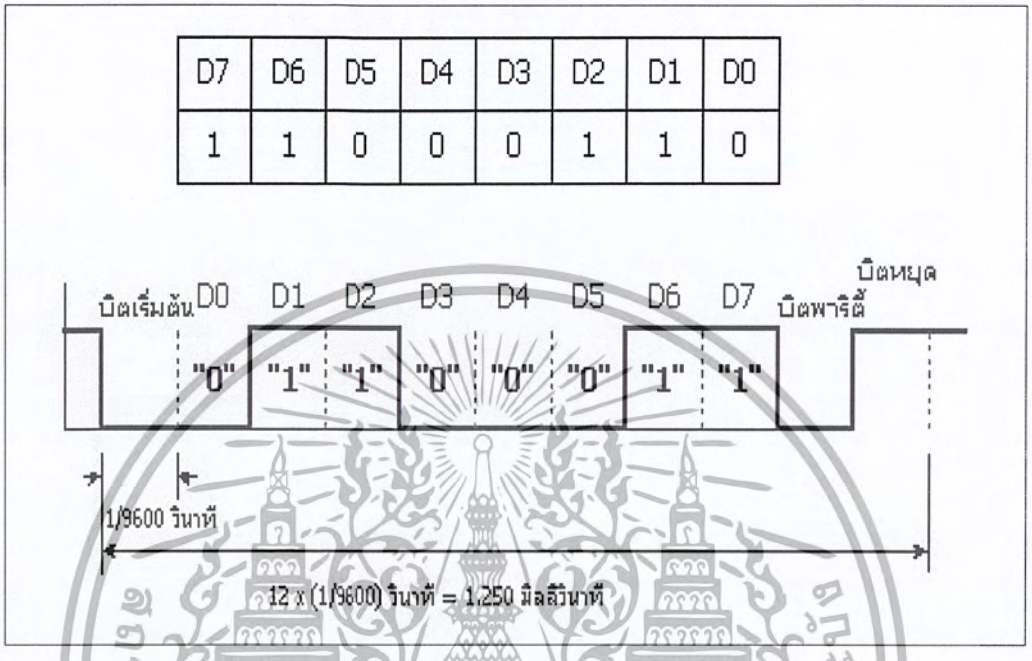
1. บิตเริ่มต้น (Start bit) จะมีขนาด 1 บิตจะเป็นระดับลอจิกตรงกันข้ามกับ ระดับลอจิกของสภาวะสายสื่อสารขณะที่ยังไม่มีการส่งข้อมูล

2. บิตข้อมูล (Data Bit) จะเริ่มจากบิตที่มีนัยสำคัญต่ำสุดก่อน หรือบิต LSB ก่อน โดยข้อมูลที่จะส่งอาจจะมีขนาด 5, 6, 7 หรือ 8 บิตก็ได้

3. บิตแสดงสภาวะเลขคู่หรือเลขคี่ (Parity Bit) มีขนาด 1 บิตโดยบิตนี้จะนำไปต่อท้ายกับบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของข้อมูลที่เป็น “1” โดยเลือกการส่งข้อมูลแบบพาริตี้คู่หรือพาริตี้คี่ ตัวอย่างถ้ากำหนดให้มีการส่งข้อมูลแบบพาริตี้คู่ แต่ข้อมูลมีเลข 1 เป็นจำนวนคี่ ก็จะให้พาริตี้นี้เป็น “1” เพื่อจะได้จำนวนเลข “1” เป็นคู่ ซึ่งทางด้านรับก็ต้องมีการตรวจสอบจำนวนข้อมูลที่รับเข้ามาเป็น “1” รวมทั้งบิตพาริตี้ 1 บิต ถ้ามีค่า “1” เป็นจำนวนคู่ แสดงว่าข้อมูลที่รับเข้ามาถูกต้อง สามารถกำหนดการรับและส่งข้อมูลเป็นแบบ NONE โดยไม่ต้องมีการตรวจสอบพาริตี้บิตก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

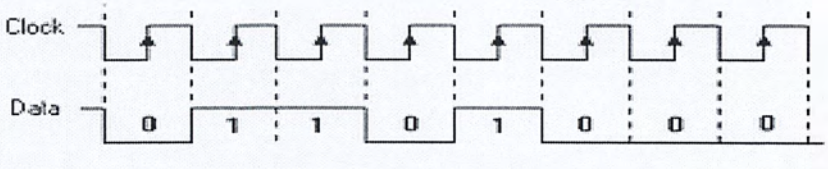
4. บิตสุดท้ายหรือบิตหยุด (Stop Bit) เป็นการระบุขอบเขตของการสิ้นสุดข้อมูล โดยจะทำให้ขาข้อมูล โดยจะทำให้ขาข้อมูลมีสถานะลอจิกเป็น “1” ซึ่งอาจมีจำนวนมากกว่า 1 บิต ก็ได้ เช่น 1 บิต 1.5 บิต หรือ 2 บิต เป็นต้น




**ภาพที่ 2.18** แสดงการส่งข้อมูลขนาด 8 บิตแบบอนุกรมพร้อมด้วย บิตเริ่มต้น บิตพาริตี้ บิตหยุด ด้วยความเร็ว 9600 บิตต่อวินาที

- **การสื่อสารแบบซิงโครนัส**

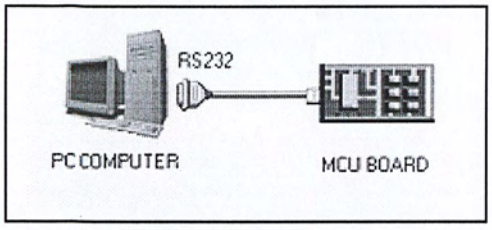
การรับส่งข้อมูลจะมีสัญญาณนาฬิกา ซึ่งเป็นตัวกำหนดจังหวะเวลาการส่งข้อมูลร่วมอยู่ด้วยอีกเส้นหนึ่งใช้คู่กับสัญญาณข้อมูล ตัวอย่างเช่น การส่งสัญญาณจากคีย์บอร์ด




**ภาพที่ 2.19** สัญญาณการสื่อสารแบบซิงโครนัส

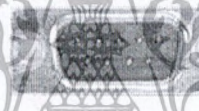
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8 การใช้งานพอร์ตอนุกรม RS232




**ภาพที่ 2.20** ลักษณะการคอนเนคเตอร์แบบ D-Type

หัวต่อแบบ D-Type ที่ใช้ในการสื่อสารแบบอนุกรมของเครื่องคอมพิวเตอร์นั้น จะมีอยู่ 2 แบบ คือ แบบ 9 ขา และ 25 ขา บางครั้งเราจะเรียกว่า DB9 และ DB25 ซึ่งหัวต่อของทั้ง 2 ชนิดจะมีลักษณะการทำงานของสัญญาณเหมือนกันแต่การจัดเรียงไม่เหมือนกัน



**ภาพที่ 2.21** พอร์ตอนุกรมของ PC DB9 ตัวผู้ (Male)

**ภาพที่ 2.22** พอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย(Female)

พอร์ตอนุกรมของ PC จะเป็นคอนเนคเตอร์แบบ DB9 ตัวผู้ (Male) และพอร์ตอนุกรมของอุปกรณ์ภายนอกจะเป็นคอนเนคเตอร์แบบ DB9 ตัวเมีย (Female) ซึ่งการจัดขาของคอนเนคเตอร์อนุกรมแบบ DB9 และหน้าที่ในการใช้งานต่าง ๆ แสดงในตารางที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาพที่ 2.23** DB9 ตัวผู้ (มองจากด้านหลัง)

**ตารางที่ 2.5** ตารางแสดงรายละเอียดของสายสัญญาณต่าง ๆ ของพอร์ตอนุกรม

| Pin | Description | Type |
|---|---|---|
| 1 | Data Carrier Detect (DCD) | Input |
| 2 | Received Data (RXD) | Input |
| 3 | Transmitted Data (TXD) | Output |
| 4 | Data Terminal Ready (DTR) | Output |
| 5 | Signal Ground (GND) | Input |
| 6 | Data Set Ready (DSR) | Input |
| 7 | Request To Sent (RTS) | Output |
| 8 | Clear To Send (CTS) | Input |
| 9 | Ring Indicator (RI) | Input |

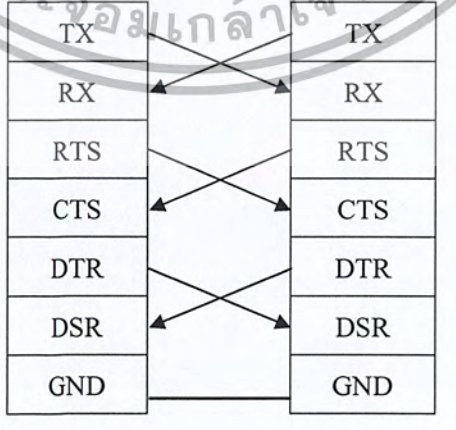
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.8.1 รายละเอียดของสายสัญญาณ

| | |
|---|---|
| Transmit Data : TD | ใช้สำหรับส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์ |
| Receive Data : RD | ใช้สำหรับส่งข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์ |
| Request to Sent : RTS | ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทาง เพื่อร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลกลับมากลับมา |
| Clear to Sent : CTS | ใช้สำหรับตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อด้วยพร้อมที่จะรับข้อมูลหรือไม่ โดยจะคอยรับสัญญาณ RTS เมื่อทุกอย่างพร้อมก็จะทำการส่งข้อมูลออกทางขา TD |
| Data Set Ready : DSR | ใช้สำหรับตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง จะใช้คู่กับขา DTR |
| Signal Ground : SG | เป็นกราวด์ระบบ |
| Carrier Detect : CD | ขานี้จะ Active เมื่อมีการส่งสัญญาณ Carrier จากโมเด็ม |
| Data Terminal Ready : DTR | ใช้สำหรับบอกให้อุปกรณ์อุปกรณ์ปลายทางรับรู้ว่าต้องติดต่อด้วยขา DTR นี้ต้องเชื่อมต่อกับขา DTR นี้ต้องเชื่อมต่อกับขา DTR ของอุปกรณ์ปลายทาง |
| Ring Indicator : RI | ขานี้จะ Active เมื่อโมเด็มได้รับสัญญาณเรียกเข้าจากสายโทรศัพท์ |

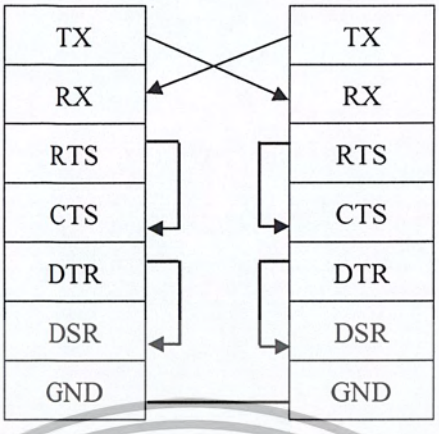
### 2.8.2 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9

แบ่งเป็น 3 รูปแบบดังรูป

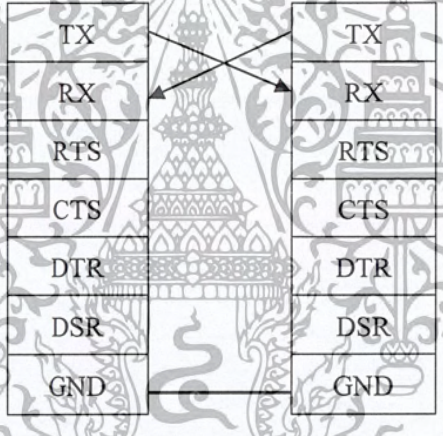



**ภาพที่ 2.24** แบบ NULL Modem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 2.25** แบบ 3 Wire




**ภาพที่ 2.26** แบบ 3 Wire None Flow Control

- TX = เป็นขาส่งข้อมูล
- RX = เป็นขารับข้อมูล
- RTS = เป็นขาที่ส่งสถานะไปยังตัวรับว่าต้องการส่งข้อมูล เมื่อต้องการส่งข้อมูลจะ ON จน กระทั่งส่ง Data ออกทางขา TX จนเสร็จจึงจะ OFF
- CTS = เป็นขาที่รอรับสถานะ จาก RTS ของอุปกรณ์ที่ต่ออยู่ด้วย
- DTR = เป็นขาที่แสดงสถานะว่า Port นั้นเปิดอยู่หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DSR = เป็นขาที่ใช้ตรวจเช็ค สถานะ DTR ของอุปกรณ์ที่เชื่อมต่ออยู่ด้วย
- GND = Signal Ground

### 2.8.3 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม

สื่อสารแบบอนุกรมที่นิยมใช้กับเครื่องคอมพิวเตอร์นั้น เป็นการสื่อสารข้อมูลแบบอะซิงโครนัส คือ ต้องใช้สายสัญญาณเส้นเดียวทำหน้าที่ส่งส่วนที่เป็นข้อมูลและส่วนที่ใช้ควบคุมการส่งข้อมูล ดังนั้นข้อมูลที่อ่านได้แต่ละบิตจากการส่งข้อมูลแบบอนุกรมจึงจะต้องแยกแยะว่าใช้สำหรับวัตถุประสงค์ใด โดยสามารถแบ่งเป็น 4 ส่วน คือ

1. Start Bit ขนาด 1 บิต
2. บิตข้อมูล (Data Character) ขนาด 7 บิต หรือ 8 บิต
3. Parity Bit ขนาด 1 บิต
4. Stop Bit ขนาด 1 บิต หรือ 2 บิต

แต่ละตัวอักษรที่ถูกส่งออกไปเป็นกลุ่มจะประกอบไปด้วยบิตเริ่มต้น บิตข้อมูล บิตพาริตี้ และบิตจบ แต่ละส่วนมีหน้าที่ดังนี้

- Start Bit หรือ บิตเริ่มต้น : จะใส่ที่จุดเริ่มต้นเสมอเพื่อเป็นการเตือนว่าอุปกรณ์ฝ่ายรับว่าข้อมูลกำลังจะมาถึง
- Data Character หรือ บิตข้อมูล : การส่งบิตข้อมูลจะส่งเป็นกลุ่ม ๆ โดยทั่วไปจะส่ง 7 หรือ 8 บิต ซึ่งเพียงพอสำหรับการส่ง Ascii Word
- Parity Bit หรือ บิตพาริตี้ : ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ส่งบิตพาริตี้จะใส่เข้าไปทั้งตัวรับและตัวส่งโดยที่ตัวรับและตัวส่งจะต้องรู้ว่าใช้พาริตี้แบบไหนในการส่งข้อมูลซึ่งหลักการในการส่งพาริตี้มีหลายแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

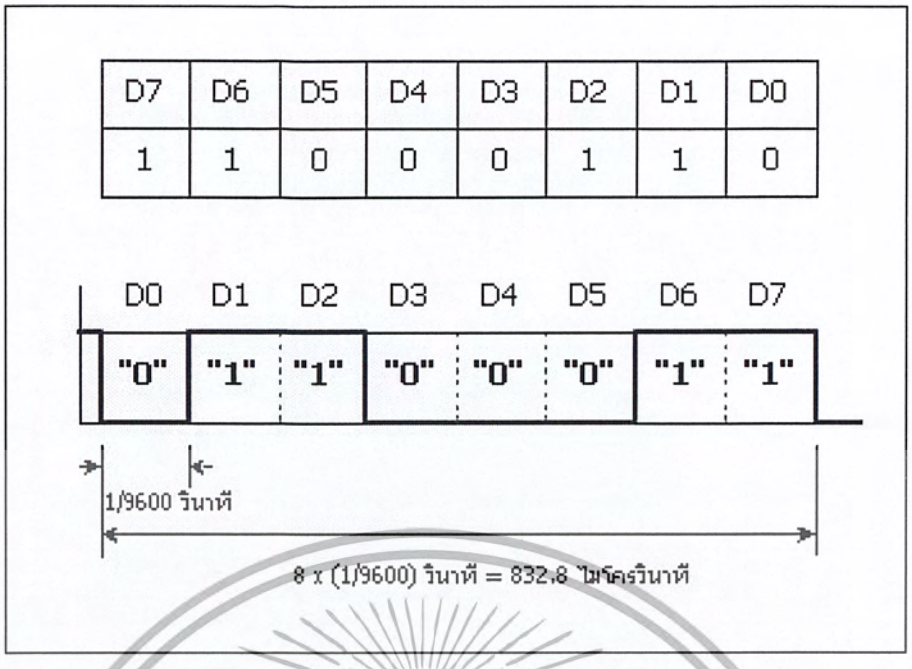
1. พาริตี้คู่ (Even Parity) ค่าของบิตพาริตี้นี้เมื่อรวมกับทุก ๆ บิตของข้อมูลแล้วจะต้องมีบิตที่เป็นเลข 1 เป็นเลขคู่
2. พาริตี้คี่ (Odd Parity) ค่าพาริตี้นี้เมื่อรวมกับทุก ๆ บิตของข้อมูลแล้วจะต้องมีจำนวนบิตที่เป็นเลข 1 เป็นเลขคี่
3. ไม่มีพาริตี้ (None Parity) ถ้าตั้งบิตพาริตี้เป็น None ทั้งภาครับและส่งจะไม่มีการตรวจสอบพาริตี้

- Stop Bit หรือบิตจบ : เป็นบิตที่ส่งมาปิดท้ายข้อมูล

**2.8.4 อัตราเร็วในการสื่อสารข้อมูลอนุกรม**

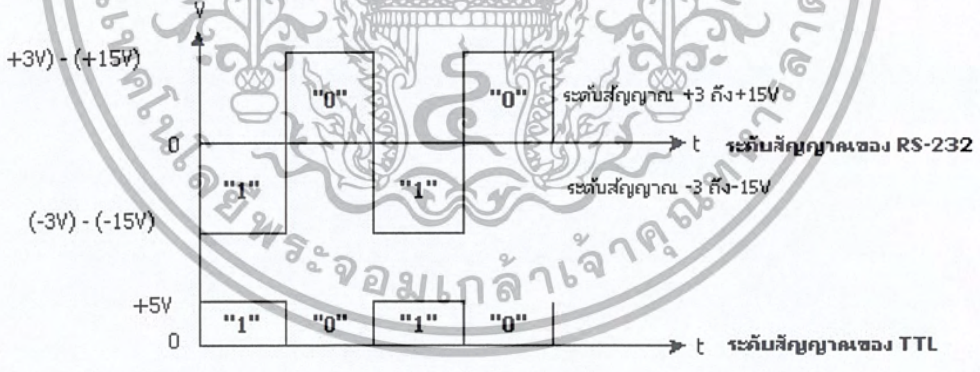
ในการสื่อสารข้อมูลแบบอนุกรมเพื่อที่จะรับ หรือส่งข้อมูลนั้นจะเป็นลักษณะของกลุ่มข้อมูล ดังนั้นอัตราความเร็วจะต้องมีค่าเท่ากันระหว่างการรับและการส่ง ซึ่งโดยทั่วไปจะระบุความเร็วของจำนวนบิตในการรับและส่งข้อมูลเป็นจำนวนของบิตที่จะส่งใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด (Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที เช่น 300, 1200, 2400, 4800 และ 9600 บิตต่อวินาทีดังภาพที่ 2.12 ถ้าหากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที จะใช้เวลาในการรับส่งข้อมูลทั้ง 8 บิต จะมีค่าเท่ากับ 8*104.1 หรือ 832.8 ไมโครวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 2.27** แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

### 2.8.5 ระดับสัญญาณของ RS232




**ภาพที่ 2.28** ระดับสัญญาณของ RS232C และระดับสัญญาณของ TTL
ในสถานะลอจิก 1 และ 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณรบกวนที่เกิดขึ้นในสายนำสัญญาณ มักจะมีแรงดันเป็นบวกเมื่อเทียบกับกราวด์ เพื่อป้องกันสัญญาณรบกวนจึงออกแบบแรงดันของโลจิก 1 เป็นลบ คืออยู่ในช่วงของ -3V ถึง +15 V

ส่วนแรงดันของลอจิก 0 อยู่ในช่วง +3V ถึง +15V และเหตุที่ระดับสัญญาณของ RS232 อยู่ในช่วง +15V ถึง -15V เพื่อให้ต่อสายสัญญาณไปได้ไกลขึ้น ดังนั้นจึงจำเป็นต้องมีวงจรเปลี่ยนระดับแรงดันของ RS232 มาเป็นระดับแรงดันของ TTL

## 2.9 โปรแกรมวิชวล เบสิก (Visual Basic)

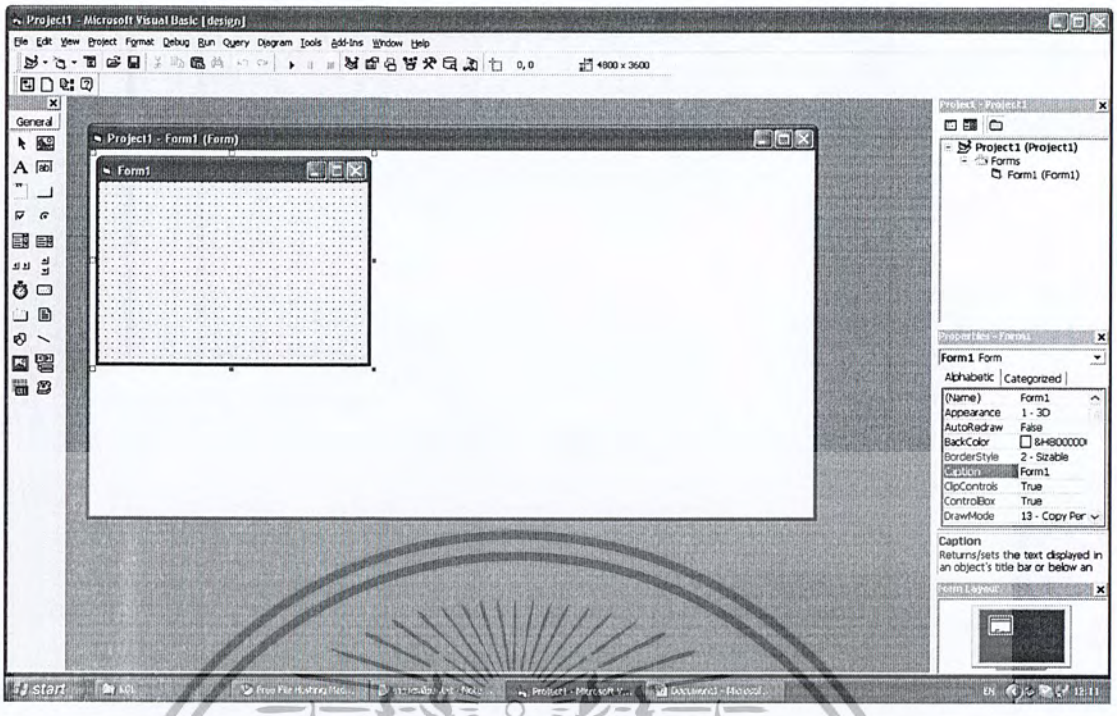
โปรแกรมวิชวล เบสิกเป็นโปรแกรมที่ใช้กันอย่างกว้างขวางในปัจจุบันสามารถรองรับกับโปรแกรมต่าง ๆ ของบริษัทไมโครซอฟท์ได้เป็นอย่างดี แม้แต่ในไมโครซอฟท์ออฟฟิศเองก็ยังต้องใช้วิชวล เบสิกสำหรับเขียนโปรแกรมต่อเติมเพื่อการทำงานในระดับสูง หรือแก้ไขส่วนที่บกพร่องเล็ก ๆ น้อย ๆ ของโปรแกรม

ในส่วนของภาษาที่ใช้ในการเขียนนั้นในด้านกราฟิกพัฒนามาจากภาษาเบสิก (Basic) ซึ่งภาษาที่มีมาตั้งแต่เดิมในสมัยที่ยังใช้ ดอส (Dos) ในการพัฒนาโปรแกรม ซึ่งในตอนนั้นมีภาษาเบสิกหลายตัว เช่น QBasic, MBasic เป็นต้น

ในโครงงานนี้เลือกใช้โปรแกรมวิชวล เบสิกเวอร์ชัน 6.0 ซึ่งเป็นเวอร์ชันที่นิยมกันมากในปัจจุบัน ทั้งในงานทางด้านกราฟิกต่าง ๆ รวมไปถึงการนำมาใช้ในการควบคุมในงานอุตสาหกรรมประเภทต่าง ๆ อีกด้วย

นอกจากนั้นยังรวมไปถึงแหล่งความรู้ทางด้านบุคคลหรือเอกสาร ซึ่งรวมทั้งหนังสือมีอยู่มากมายสำหรับเวอร์ชัน 6.0 ดังนั้นเมื่อเกิดปัญหาจะสามารถแก้ไขได้โดยไม่ลำบาก รวมถึงการเพิ่มทักษะในการเขียนโค้ดต่าง ๆ ให้งานออกมาดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 2.29** หน้าต่างแสดงอุปกรณ์และเครื่องมือพื้นฐานของวิชวล เบสิก

### 2.9.1 คุณสมบัติเด่น ของวิชวล เบสิก

นอกจากความสะดวกสบายในการใช้และศึกษาโปรแกรมแล้ว วิชวล เบสิก มีข้อดีพอสรุปได้ดังนี้

1. ใช้เขียนโปรแกรมได้ทั้งวินโดว์และเว็บวิชวล เบสิกในเวอร์ชันหลัง ๆ ที่ได้รับการพัฒนาให้สามารถเขียนโปรแกรมให้ทำงานได้ทั้งบนวินโดว์ และบนอินเตอร์เน็ต โดยที่ผู้สนใจเขียนโปรแกรมบนวินโดว์สามารถประยุกต์ใช้ความรู้ที่มีอยู่ในการพัฒนาเว็บแอพพลิเคชันได้ง่าย โดยที่สามารถเรียนรู้ได้อย่างรวดเร็ว

2. แก้ไขโปรแกรมที่เขียนขึ้นได้โดยง่ายในบางครั้งการเขียนโปรแกรมอาจจะมีข้อผิดพลาด การแก้ไขหรือปรับปรุงในภายหลังของวิชวล เบสิกนั้นสามารถทำได้ง่าย

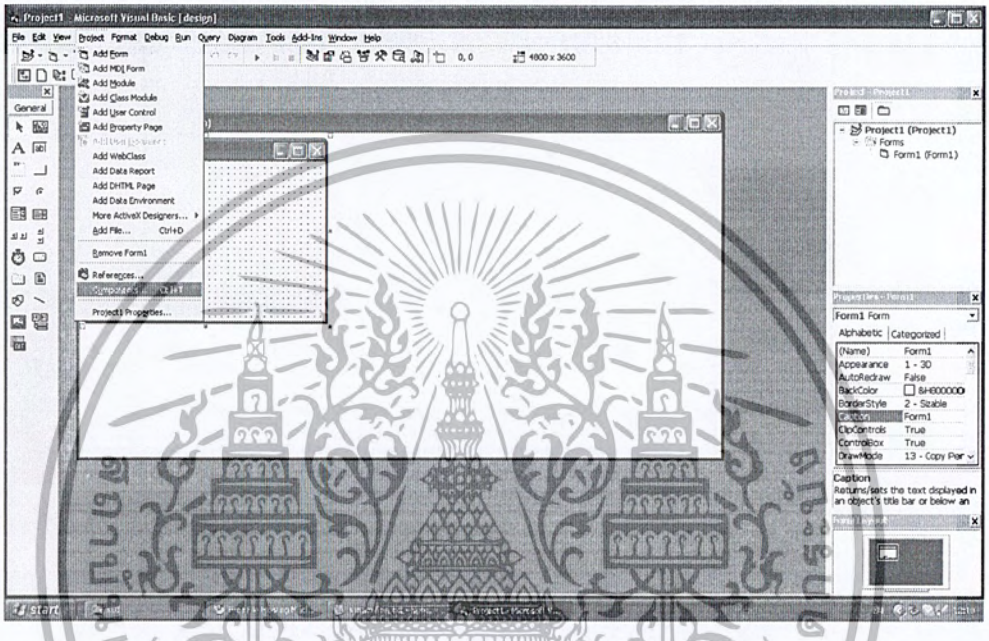
3. มีเอกสารให้ศึกษาอ้างอิงมาก จากที่ได้กล่าวมาแล้วว่าโปรแกรมวิชวล เบสิกเป็นที่นิยมใช้มากในปัจจุบัน เมื่อมีปัญหาหรือข้อสงสัยก็สามารถหาคำตอบได้จากเอกสาร รวมถึงจากเว็บไซท์ต่าง ๆ

4. โปรแกรมที่มีอนาคตอย่างวิชวล เบสิกนั้นเป็นภาษาเขียนโปรแกรมที่มีไมโครซอฟท์ที่ถือว่าเป็นภาษาที่สำคัญมากจึงถูกพัฒนาอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.9.2 การเขียนโปรแกรมติดต่อและควบคุม Serial Port ด้วย วิชวล เบสิก

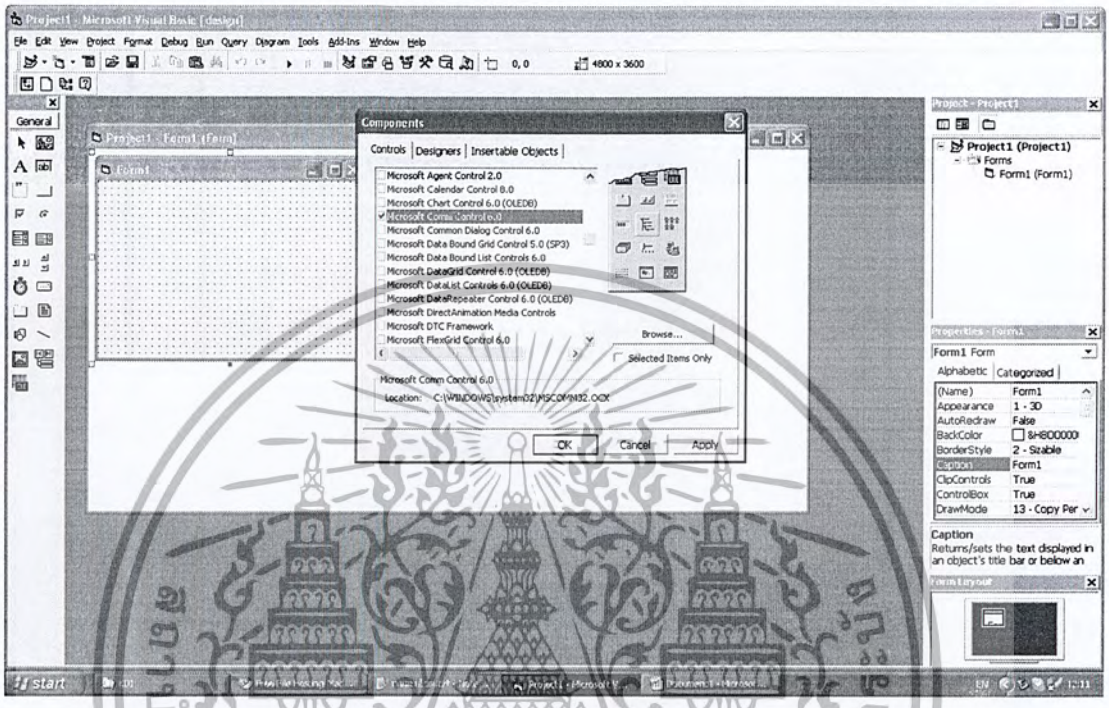
คอนโทรลที่สำคัญในการทำให้วิชวล เบสิกนั้นสามารถสื่อสารผ่านพอร์ตอนุกรมได้ก็คือ คอนโทรล MSComm ในการใช้งาน MSComm นั้นเราจะต้องทำการเพิ่มคอนโทรลนี้เข้าไปใน Tool Box ของโปรแกรมวิชวล เบสิกซึ่งสามารถกระทำได้โดยคลิกที่เมนู Project แล้วเลือก Component ดังรูป




**ภาพที่ 2.30** เพิ่มคอมโพเนนต์ MSComm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจะปรากฏไดอะล็อก Component ขึ้นมาจากนั้นให้คลิกเลือกที่ Microsoft Comm Control 6.0 แล้วคลิกที่ปุ่ม OK เมื่อคลิกที่ปุ่ม OK ดังรูป




**ภาพที่ 2.31** ไดอะล็อก Component เลือกที่รายการ MSComm

จากนั้นก็ปรากฏภายใน Toolbox จะมีไอคอนรูปโทรศัพท์ซึ่งเป็นไอคอนของคอนโทรล MSComm ปรากฏขึ้นมาให้เราเลือกใช้งาน

### 2.9.3 การกำหนดคุณสมบัติ ( Property ) ที่สำคัญในการใช้งาน MSComm ให้สามารถต่อกับพอร์ตได้

1. Property ชื่อ CommPort ใช้ในการกำหนดหมายเลขของพอร์ตอนุกรมที่เราต้องการติดต่อ ตัวอย่าง ถ้าเรากำหนดให้การเขียนโปรแกรมติดต่อกับพอร์ต Com1 ซึ่งจะเขียนเป็น MSComm1.CommPort = 1

2. Property ชื่อ Settings คือการตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอดของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่าง ๆ ดังนี้ กำหนดอัตรา Baud Rate หรือความเร็วในการส่งข้อมูล มีหน่วยเป็นบิตต่อวินาที, พาริตี้, จำนวนของบิตข้อมูล, จำนวนของบิตปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท้าย ตัวอย่าง เรากำหนดให้มีการเขียนโปรแกรมใช้งานที่ Baud Rate = 9600 บิตต่อวินาที ไม่มีพาริตี้ จำนวนบิตข้อมูลเท่ากับ 8 บิต และมีบิตปิดท้าย 1 บิต MSComm1.Settings = "9600, N, 8,1"

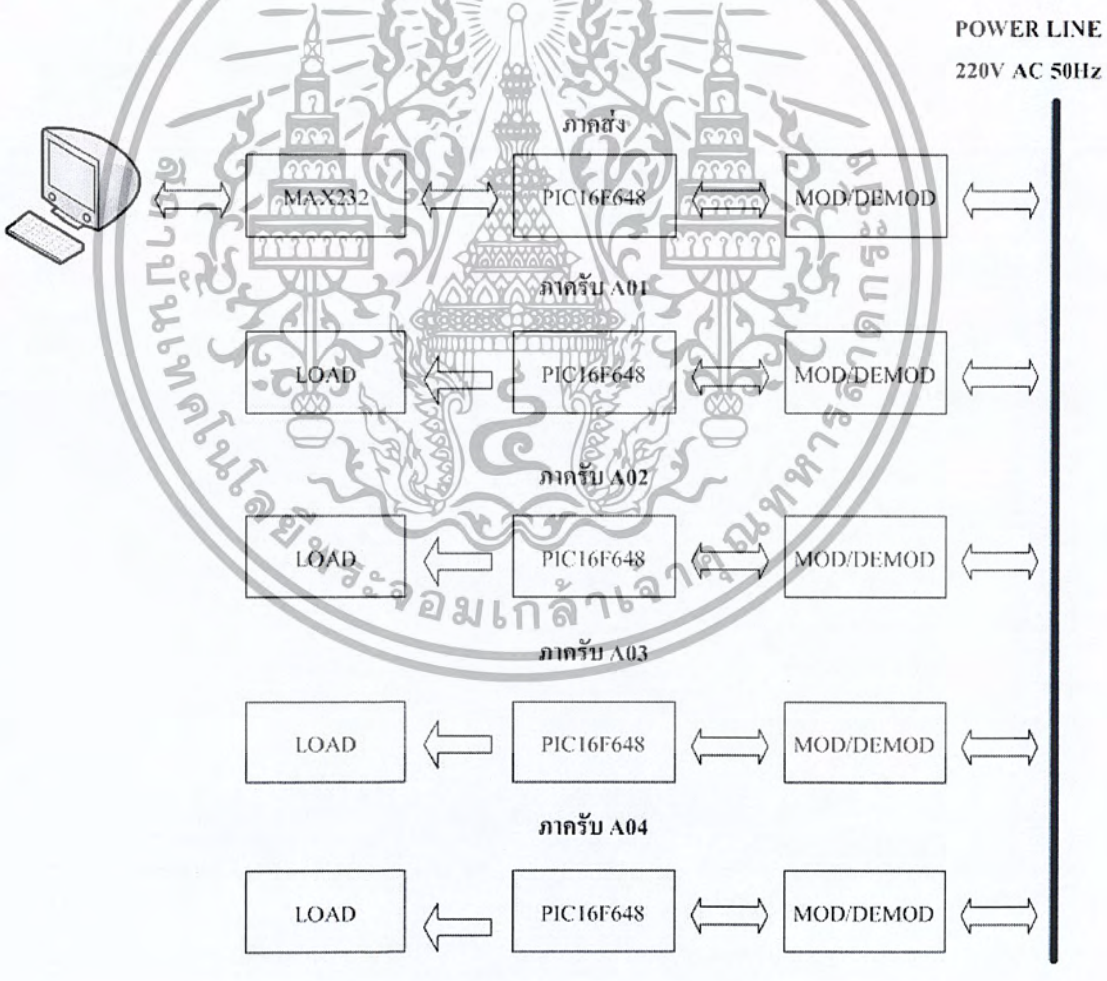
3. Property ชื่อ PortOpen ใช้สำหรับเปิดและปิดการใช้งานพอร์ตอนุกรม ตัวอย่าง เราจะเปิดใช้งานพอร์ตอนุกรมให้กำหนดค่า Value เป็น True เขียนโค้ดได้ดังนี้ MSComm1.PortOpen = True แต่ถ้าต้องการปิดพอร์ตอนุกรมให้กำหนดค่า Value เป็น False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# หลักการทำงานและการออกแบบ

ปริญญานิพนธ์ฉบับนี้ได้เริ่มทำจากการศึกษาหลักการของการมอดูเลตสัญญาณ แบบการมอดูเลตเชิงเลขทางความถี่ โดยใช้มาตรฐานX10 ในการสื่อสารข้อมูลผ่าน AC-line และมีการใช้ไมโครคอนโทรลเลอร์ในการสั่งงาน ให้อุปกรณ์ไฟฟ้าทำงานตามมาตรฐานX10ได้ โดยนำวงจรที่ได้มาทำการเขียนลายวงจรลงในแผ่นปริ้นต์ โดยได้นำโปรแกรม Protel 99 SE มาใช้เพื่อนำวงจรดังกล่าวมาใช้ในการรับส่งข้อมูลในการที่จะนำไปใช้ควบคุมอุปกรณ์ไฟฟ้าต่อไป

## 3.1 โครงสร้างของระบบการส่งข้อมูลไปควบคุมอุปกรณ์ไฟฟ้า




**ภาพที่ 3.1** โครงสร้างของระบบการสื่อสารผ่าน AC-line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.1** ตารางแสดงอุปกรณ์และหน้าที่ของระบบ

| อุปกรณ์ | หน้าที่ |
|---|---|
| Computer | จะทำหน้าที่เป็นตัวรับคำสั่งจากผู้ใช้งาน โดยที่จะส่งข้อมูลที่ได้รับมาผ่านพอร์ต-อนุกรมไปยังตัว Micro Controller PIC16F648A ทำการส่งข้อมูลไปยังชุดควบคุมทางภาครับ สำหรับในส่วนของการแสดงผลนั้นจะทำโดย ที่จะส่งผ่านโปรแกรม Visual C++ ซึ่งสามารถใช้ mouse และ keyboard ในการเลือกตำแหน่ง และคำสั่งที่จะสั่งได้ง่าย และสะดวก |
| Micro Controller PIC16F648A Transmitter | จะทำหน้าที่รับคำสั่งจากตัวคอมพิวเตอร์แล้วทำการกำหนด codes ซึ่งเป็นลักษณะที่ชุดควบคุมทางด้านภาครับสามารถอ่านเข้าใจและตีความเพื่อนำไปปฏิบัติได้ในการรับข้อมูลนั้น เพื่อที่จะรับผ่านพอร์ตอนุกรม (Serial Port RS-232) แล้วทำการสร้าง codes เพื่อให้สอดคล้องกับคำสั่งจากตัวคอมพิวเตอร์ |
| Modulate and coupling ชุด Modulate | จะทำหน้าที่ส่งสัญญาณควบคุมที่ได้รับ เพื่อที่นำมา Coupling เข้าสู่สายไฟเอซีโดยจะมีการมอดดูเลตเป็นแบบการมอดูเลตเชิงเลขทางความถี่ |
| Demodulate and Decoupling | จะทำหน้าที่คอยรับสัญญาณควบคุมที่ถูกส่งมายัง Power line เพื่อจะมาทำการ Decoupling และ Demodulate แล้วหลังจากที่ทำการ Demodulate แล้วจะทำการขยายสัญญาณ เพื่อให้สัญญาณที่ได้กลับมาเป็นสัญญาณที่จะสามารถนำไปใช้ประมวลผลและสั่งงานได้ |

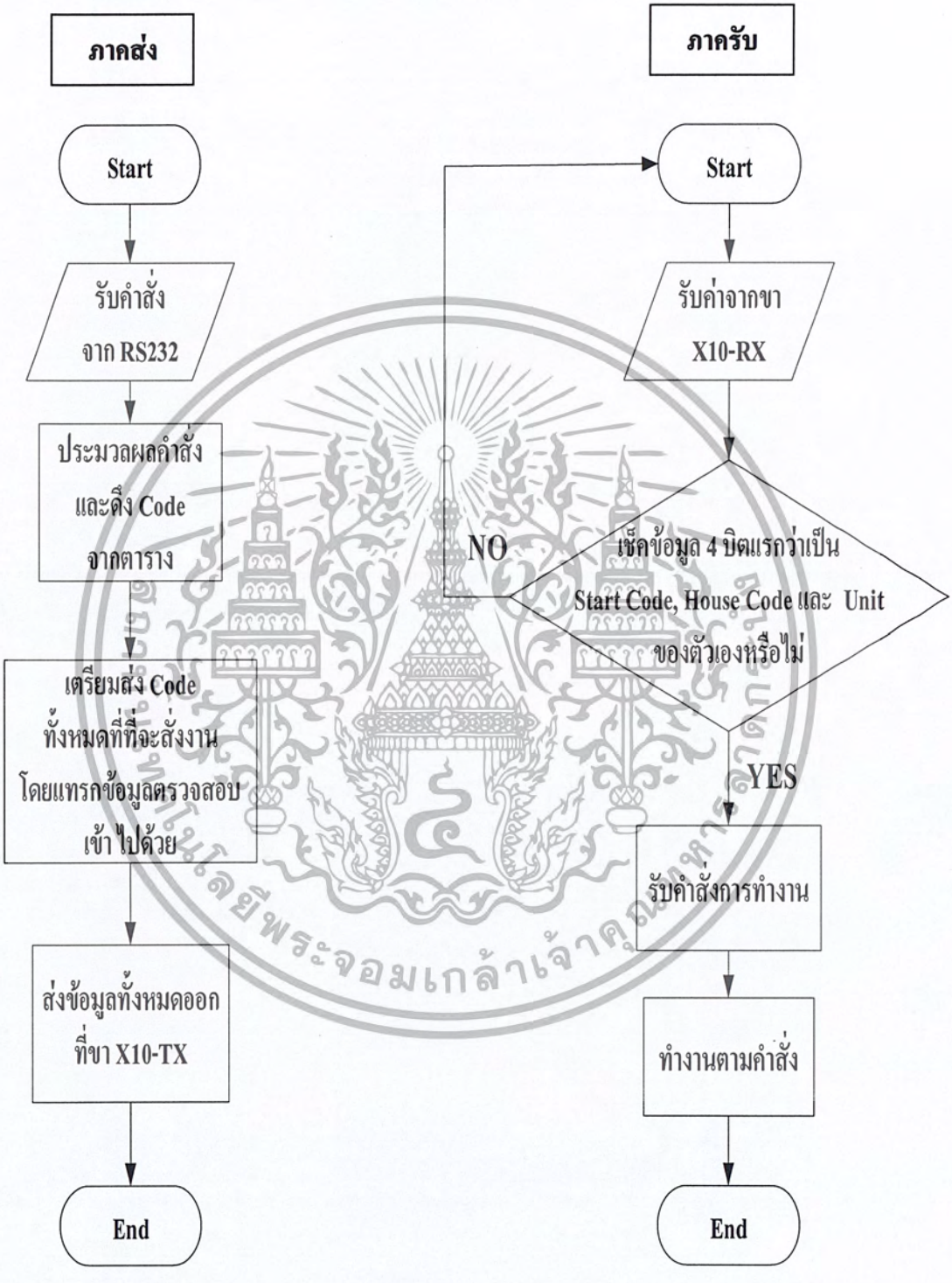
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.1** ตารางแสดงอุปกรณ์และหน้าที่ของระบบ (ต่อ)

| อุปกรณ์ | หน้าที่ |
|---|---|
| Micro controller PIC16F648A Receiver | ทำหน้าที่รับสัญญาณแล้วจะทำการประมวลผลคำสั่งที่เข้ามาเพื่อสร้างสัญญาณไป Trig ตัว Relay ของ Load Switch ต่อไป |
| Load Switch | จะทำการควบคุมนั้น เราจะควบคุมเฉพาะคำสั่งเปิด-ปิด เท่านั้น ซึ่งวงจร Load Switch นี้สามารถที่จะนำไปต่อกับอุปกรณ์ไฟฟ้าภายนอก |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

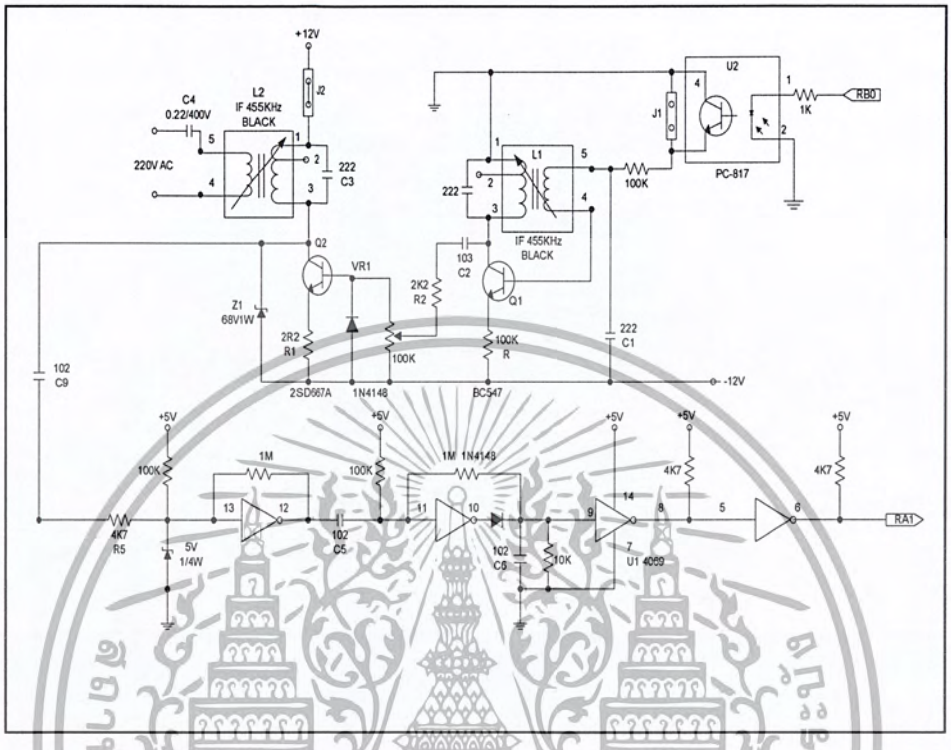
## 3.2 โฟลว์ชาร์ตแสดงการทำงานของภาคส่ง และภาครับ




**ภาพที่ 3.2** โฟลว์ชาร์ตการทำงานของภาครับ และภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.3 หลักการในการมอดูเลต




**ภาพที่ 3.3** วงจรภาครับ และภาคส่ง X10

### 3.3.1 การทำงานของภาคส่ง

วงจรภาคส่งประกอบด้วย OSC Q1 BC547 และ L1 ซึ่งทำหน้าที่เป็นทรานฟอร์เมอร์ เพื่อขยายสัญญาณให้แก่ IF 455 กิโลเฮิรตซ์ ในวงจรภาครับ AM (ตัวสีดำ) ซึ่งต่ออยู่กับตัว C ที่มีค่าความจุ 222 ไมโครฟารัด จะมีหน้าที่ทำให้ความถี่ต่ำลงมาเท่ากับ 120 กิโลเฮิรตซ์ เอาท์พุตที่ได้จากขา C ของ Q1 จะถูกส่งต่อไปยังภาคขยาย

สัญญาณของ Q2 ที่ทำหน้าที่รวมสัญญาณเข้ากับสายไฟ AC ผ่าน L2 ซึ่งทำหน้าที่เป็นทรานฟอร์เมอร์ เพื่อขยายสัญญาณให้แก่ IF เช่นเดียวกัน ในการควบคุมวงจร OSC นั้นจะควบคุมโดยขา RB0 ผ่าน IC เบอร์ PC-817 และ VR1 จะใช้สำหรับปรับแรงดันให้ลดลง ในกรณีส่งสัญญาณแบบต่อเนื่อง เพื่อใช้ในการทดสอบโมดูลอื่น ๆ ไม่เช่นนั้นอาจจะทำให้ Q2 ร้อนจัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

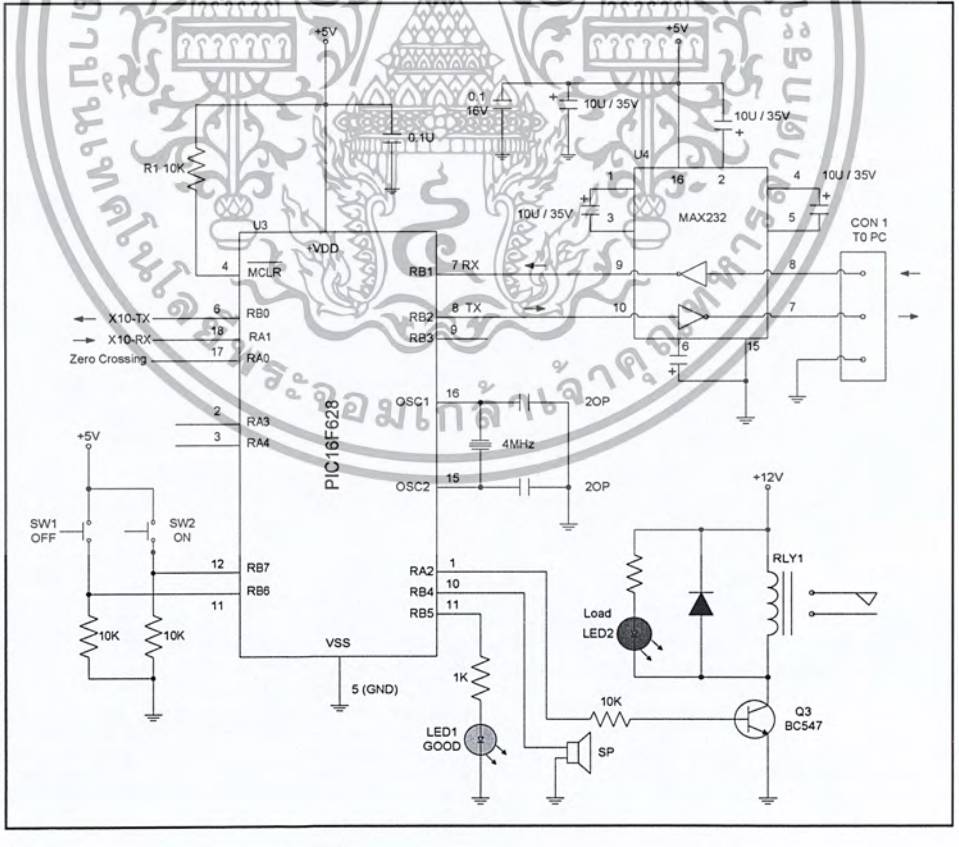
### 3.3.2 การทำงานของภาครับ

วงจรภาครับในขณะที่ไม่ได้ส่งสัญญาณนั้น L2 จะทำหน้าที่เป็นวงจรจูน และสัญญาณที่ได้จากขาที่ 3 ของ L2 จะถูกขยาย และปรับระดับสัญญาณให้มีระดับเดียวกับ TTL โดย U1 (4069) เพื่อส่งให้กับ MCU ที่ขา RA1

สัญญาณจะเข้ามาทางขา C ของ Q1 ผ่าน C9, R5 (High Pass) ซึ่งถูกป้องกันแรงดันเกินด้วยซีเนอร์ 5 โวลท์ เข้ามาทางขา 13 ของ U1-1 (13, 12) ซึ่งต่อไว้กับ R 100 กิโลโอห์ม เพื่อ Pull Up อินพุตไว้ไม่ให้ลอย มีหน้าที่ขยายสัญญาณเบื้องต้น และถูกป้อนกลับแบบ Negative ที่ขา 13, 12 ด้วย R 1 เมกะโอห์ม เพื่อควบคุมอัตราขยาย โดยให้เอาท์พุตออกทางขา 12 ผ่าน C5(102) ที่ทำหน้าที่ให้เฉพาะสัญญาณที่เป็นสัญญาณ AC ผ่านเท่านั้น

ส่วน U1-2(11, 10) ซึ่งทำหน้าที่ขยายสัญญาณอีกต่อหนึ่ง ที่เอาท์พุต U1-2 (ขา 10) ต่อไว้กับไดโอด และ C6 ทำหน้าที่เป็น Envelope Detector ซึ่งจะตัดสัญญาณ 120 กิโลเฮิรตซ์ เพื่อให้เหลือแต่สัญญาณดิจิตอล และป้อนกลับแบบ Negative ด้วย R 1 M ไปยังอินพุตที่ขา 11, 10

สัญญาณที่ได้จะผ่าน U1-3 (9, 8) ซึ่งมีหน้าที่ปรับสัญญาณรูปคลื่นให้คมชัด



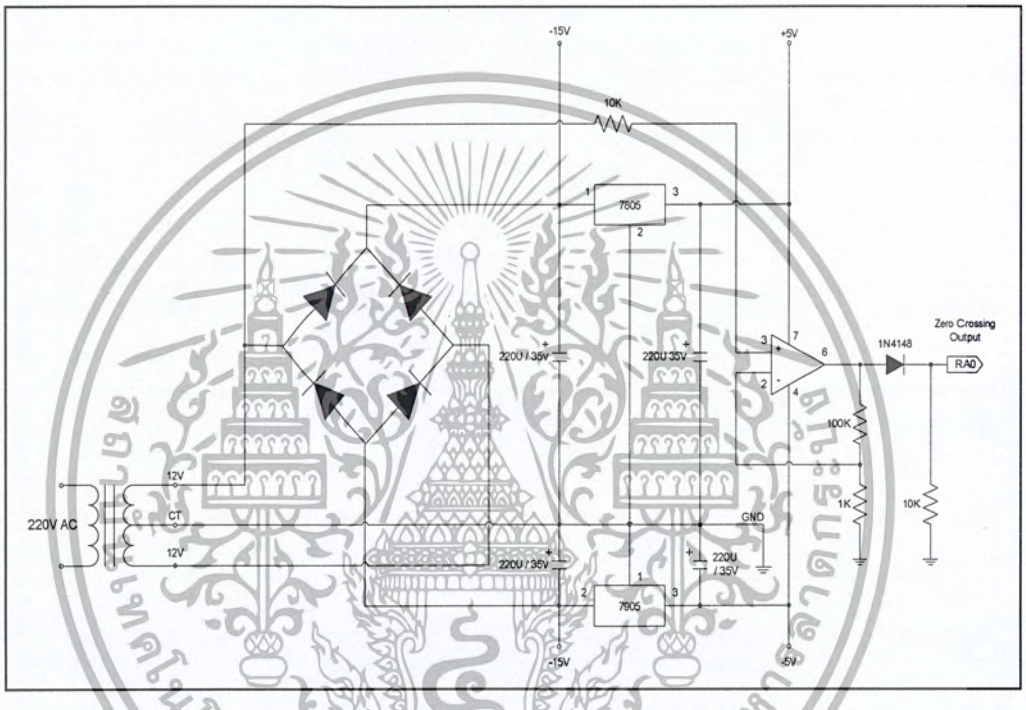

**ภาพที่ 3.4** ส่วนควบคุมด้วย MCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.3 ภาคควบคุม

วงจรภาคควบคุมจะประกอบไปด้วย Micro Controller Unit โดยใช้ PIC เบอร์ 16F648A ซึ่งทำหน้าที่

- วิเคราะห์สัญญาณ และแสดงผลผ่าน RS232 โดยใช้ IC MAX232
- รับคำสั่งจาก RS232 และแสดงข้อมูลผ่าน AC-Line
- ตรวจสอบการกดปุ่ม ON-OFF เพื่อควบคุมอุปกรณ์ไฟฟ้าผ่าน RLY1

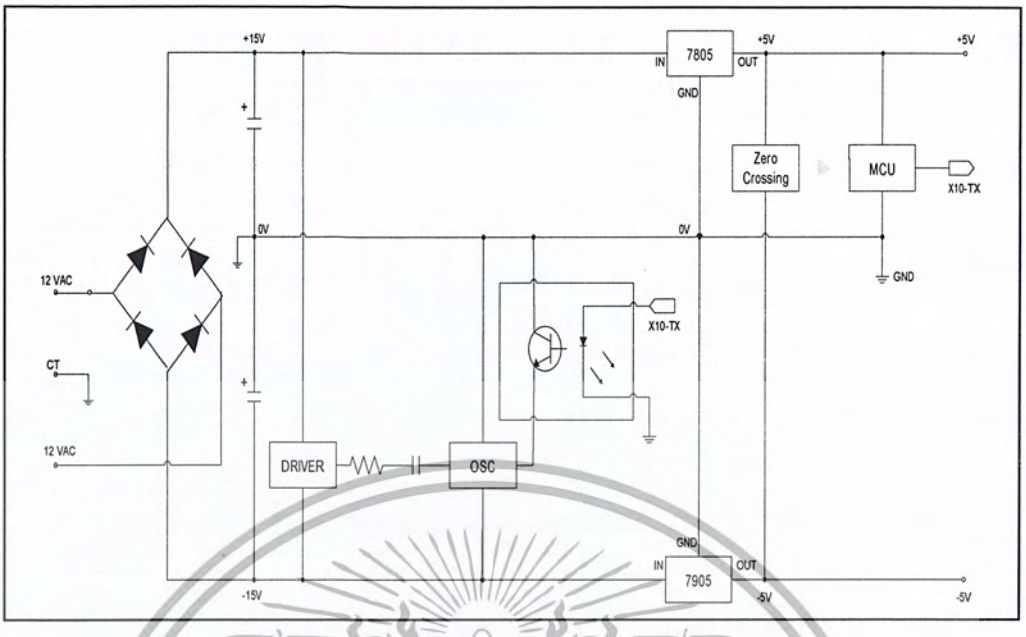



**ภาพที่ 3.5** ภาคจ่ายไฟ

### 3.3.4 ภาคจ่ายไฟ

วงจรภาคจ่ายไฟ จะเป็นส่วนสร้างสัญญาณ เพื่อใช้ในการตรวจสอบจุดตัดศูนย์ โดยใช้ OPAMP 741 ทำหน้าที่เป็นวงจร Schmitt Trigger

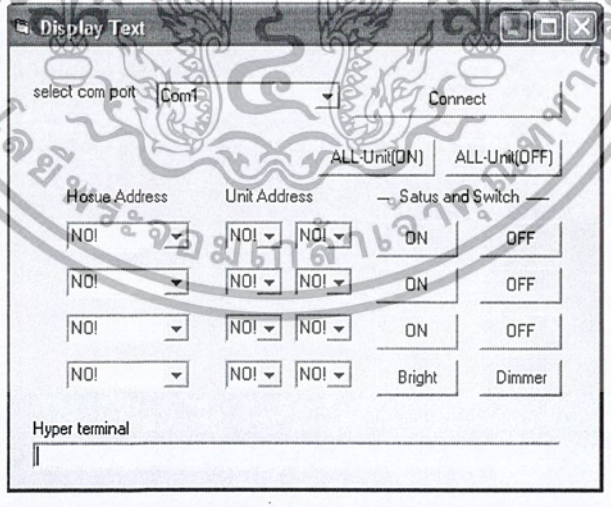
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 3.6** แผนผังของการวัดระดับแรงดันไฟฟ้า

## 3.4 โปรแกรมควบคุมอุปกรณ์ผ่านทางเครื่องคอมพิวเตอร์

พัฒนาโปรแกรมสั่งงานผ่านคอมพิวเตอร์ด้วยโปรแกรม วิชวล เบสิก เวอร์ชัน 6.0




**ภาพที่ 3.7** หน้าตาของโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาพที่ 3.8** โฟลว์ชาร์ตแสดงการทำงานของโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์

ความสามารถของโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์

1. สามารถเลือก COMPORT ที่ใช้ในการติดต่อกับไมโครคอนโทรลเลอร์ได้
2. สั่งงานแบบ ALL-Unit คือสั่งงานทุกตัวที่ต่อในเครือข่ายเป็นเป็น House เดียวกัน
3. สั่งงานแบบทีละ Unit สามารถเลือกชื่อ Unit ที่ต้องการส่งงานได้
4. มีฟังก์ชั่น ON, OFF, Bright, Dimmer
5. สามารถดูข้อความการทำงานในลักษณะของ Hyper terminal ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

# การทดลองและผลการทดลอง

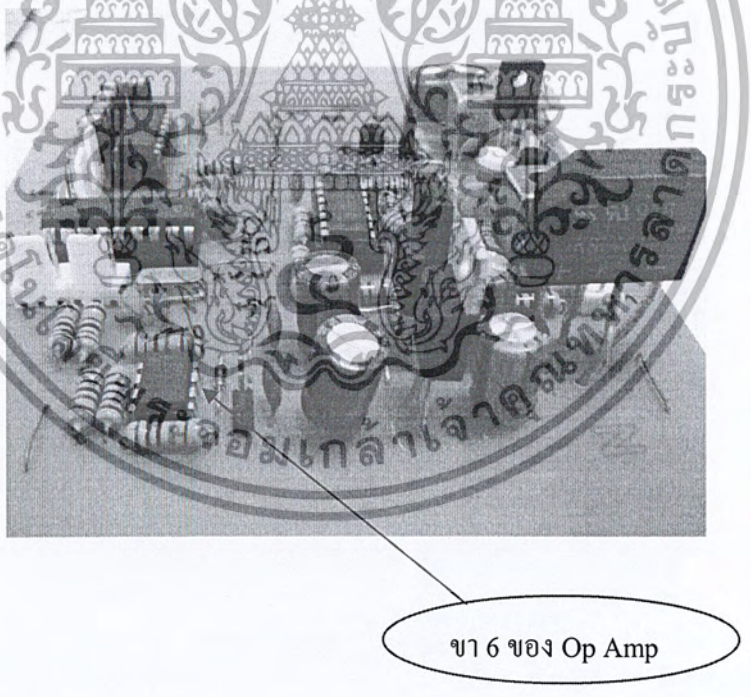
## 4.1 ทดลองการทำงานของวงจรภาคส่ง

### วิธีทำการทดลอง

การทดลองสัญญาณจุดตัดศูนย์ ของภาคส่ง

- นำสัญญาณไฟฟ้ากระแสสลับเข้าที่ขา 3 ของ Op Amp 741 เพื่อต้องการนำความถี่ที่ได้จาก AC-line มาเปรียบเทียบเพื่อให้ไมโครคอนโทรลเลอร์ส่งข้อมูลตามความถี่ของ AC-line

- วัดค่าที่ขา 6 ของ Op Amp 741 จะได้สัญญาณจุดตัดศูนย์ เป็นสัญญาณที่ใช้ควบคุมการส่งข้อมูลของไมโครคอนโทรลเลอร์ เข้าไปใน AC-line



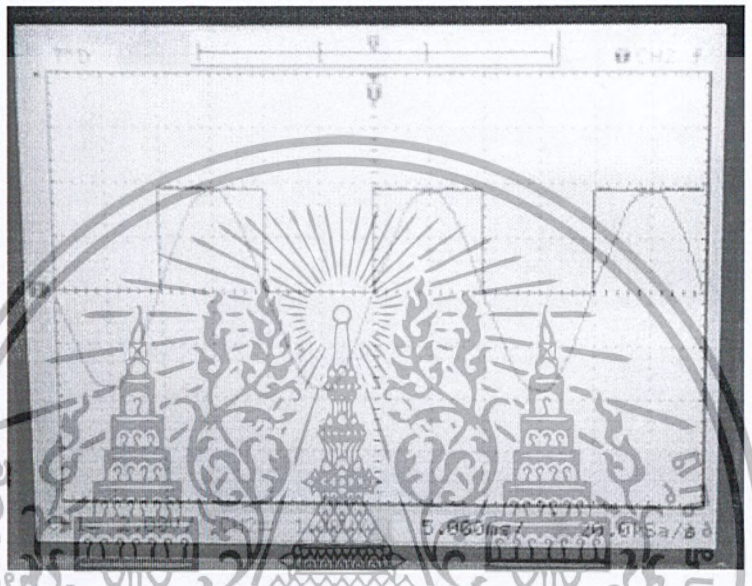

**ภาพที่ 4.1** แสดงการวัดสัญญาณจุดตัดศูนย์ที่ขา 6 ของ Op Amp 741

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองสัญญาณจุดตัดศูนย์ ของภาคส่ง

- เมื่อใช้ออสซิลโลสโคป วัดสัญญาณที่ขา 6 ของ Op Amp 741 จะได้สัญญาณเป็นแรงดันพัลส์ ดังภาพที่ 4.2

- เมื่อใช้ออสซิลโลสโคป วัดสัญญาณจาก AC-line จะปรากฏรูปคลื่นซายน์ที่มีความถี่ 50 เฮิรตซ์ ตัดกับสัญญาณของจุดตัดศูนย์



**ภาพที่ 4.2** สัญญาณจุดตัดศูนย์

Ch 1 คือ แรงดันพัลส์จากวงจรจุดตัดศูนย์

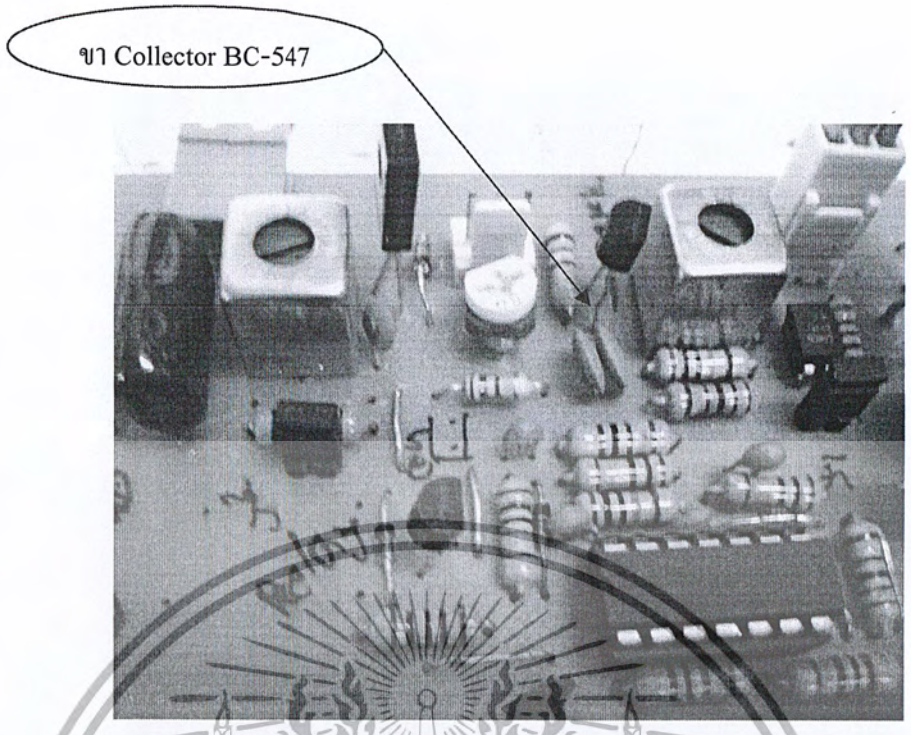
Ch 2 คือ AC-line

การทดลองส่งความถี่ 120 กิโลเฮิรตซ์ ของภาคส่ง

- ทำการปรับความถี่ที่ IF 455 กิโลเฮิรตซ์ ให้ได้ 120 กิโลเฮิรตซ์

- วัดค่าที่ขา Collector ของTransistor BC-547 จะมีสัญญาณความถี่ 120 กิโล-เฮิรตซ์ เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

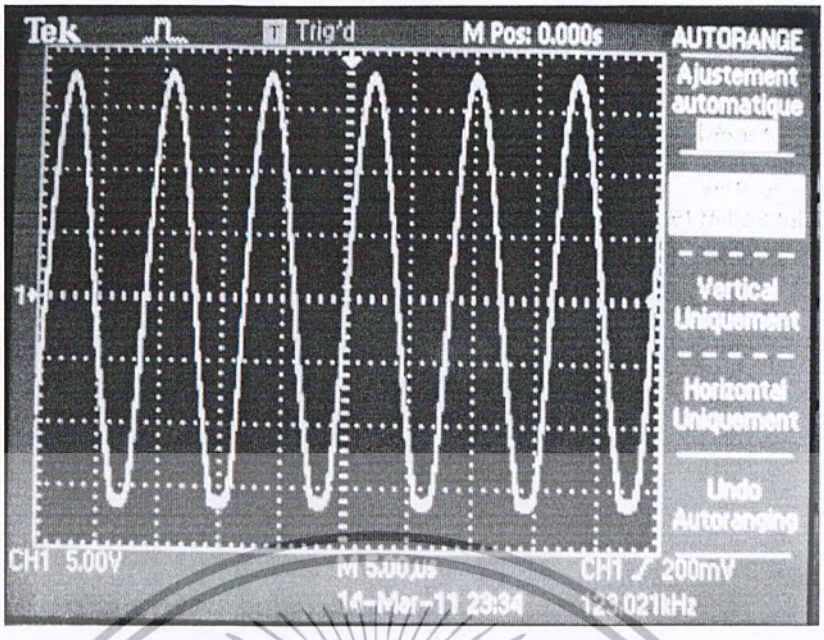



**ภาพที่ 4.3** แสดงการวัดสัญญาณที่ขา Collector ของ BC-547

ผลการทดลองส่งความถี่ 120 กิโลเฮิรตซ์ ของภาคส่ง

- วัดค่าที่ขา Collector ของTransistor BC-547 จะมีสัญญาณความถี่ 120 กิโล-เฮิรตซ์เกิดขึ้น
- จากการวัดสัญญาณดังภาพที่ 4.3 จะมีสัญญาณปรากฏขึ้นดังภาพที่ 4.4
- ทำให้เราสามารถนำสัญญาณที่ได้จาก IF 455 กิโลเฮิรตซ์ มาใช้ในการมอดูเลต เข้าไปใน AC-line ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 4.4** แสดงสัญญาณความถี่ 120 กิโลเฮิรตซ์

Ch 1 คือ สัญญาณความถี่ 120 กิโลเฮิรตซ์

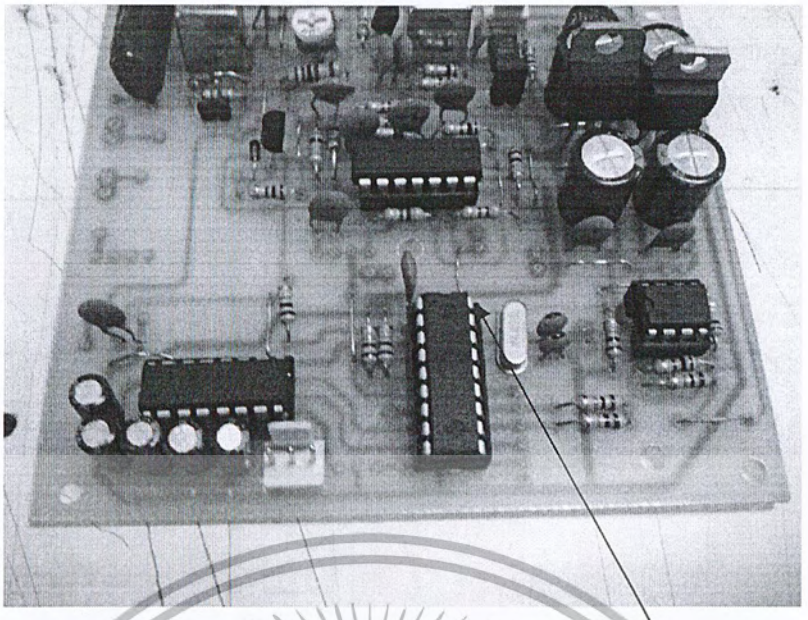
## 4.2 ทดลองการทำงานของวงจรภาครับ

**วิธีการทดลอง**

การทดลองรับสัญญาณของ X10 ของภาครับ

- จากการที่เขียนโปรแกรมเพื่อป้อนให้ไมโครคอนโทรลเลอร์ส่งสัญญาณจากขา X10-Tx ของภาคส่งไปยังภาครับเพื่อสั่งงานให้ภาครับทำงานตามมาตรฐานของ X10
- ในการส่งข้อมูลนั้นเมื่อข้อมูลที่ต้องการส่งข้อมูลเป็นลอจิก 1 จะทำให้มีการส่งความถี่ 120 กิโลเฮิรตซ์ เข้าไปในสายไฟฟ้ากระแสสลับ 3 ลูกคลื่น โดยจะส่งข้อมูลตามจังหวะของจุดตัดศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



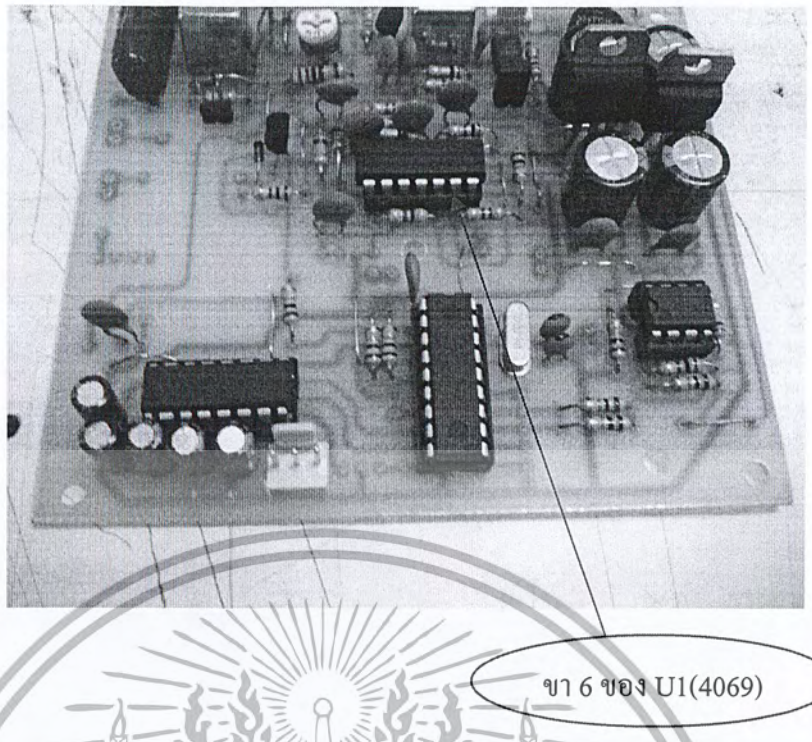

**ภาพที่ 4.5** แสดงการวัดสัญญาณที่ขา X10-RX ของภาครับ

ผลการทดลองการทำงานของวงจรภาครับ

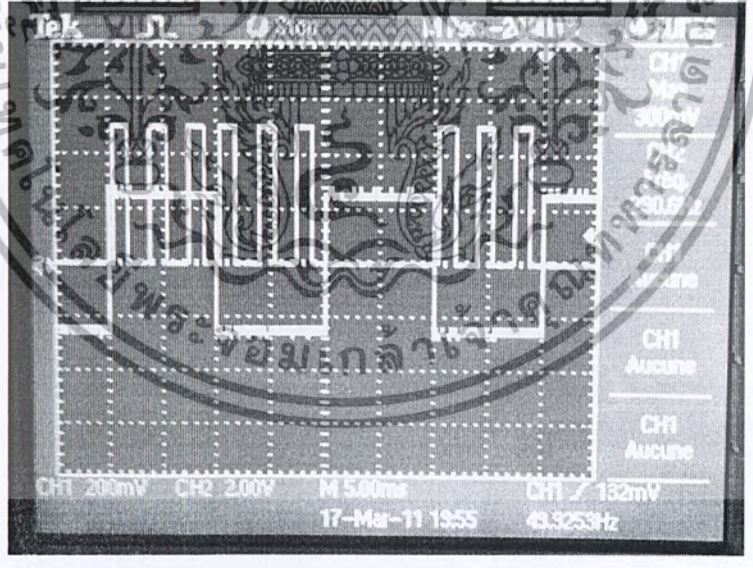
- ในการรับข้อมูลนั้นเมื่อข้อมูลที่ต้องการส่งข้อมูลเป็นลอจิก 1 จะทำให้มีการส่งความถี่ 120 กิโลเฮิรตซ์ เข้าไปในสายไฟฟ้ากระแสสลับ 3 ลูกคลื่น โดยจะส่งข้อมูลตามจังหวะของจุดตัดศูนย์

- เมื่อมีการรับสัญญาณมาจาก AC-line จะถูกแปลงสัญญาณความถี่ 120 กิโล-เฮิรตซ์ โดยจะเปลี่ยนเป็นสัญญาณ TTL ด้วย U1(4069) ที่ขา 6 และจะส่งสัญญาณ TTL ที่ได้ไปยังขา X10-RX ของภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 4.6** แสดงขาที่ใช้ในการส่งข้อมูลที่เป็นสัญญาณ TTL เข้าไปที่ขา X10-RX ของภาครับ



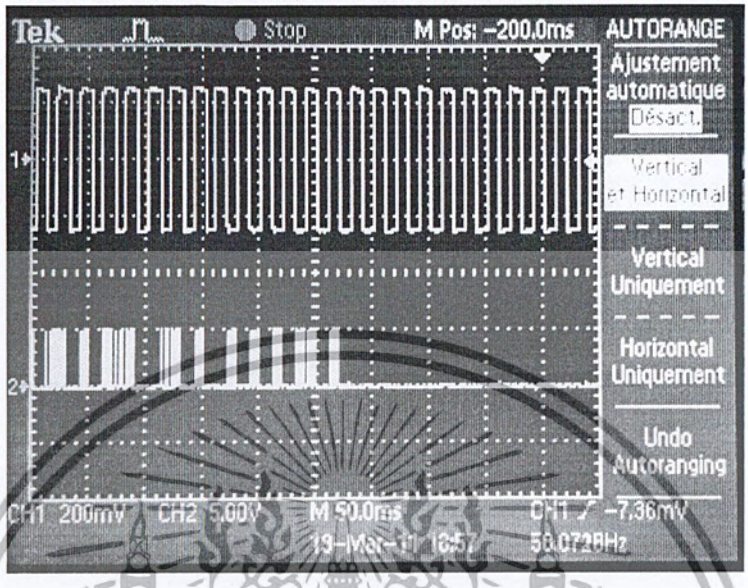

**ภาพที่ 4.7** แสดงการรับข้อมูลที่เป็นสัญญาณ TTL เข้าไปที่ขา X10-RX ของภาครับ

Ch 1 คือ สัญญาณที่ได้จากวงจรจุดตัดศูนย์

Ch 2 คือ สัญญาณTTL ที่มาจากการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วัดที่ขา X10-Rx ของภาครับ จะเห็นได้ว่ามีการส่งข้อมูลเข้ามาที่ขาของภาครับอย่างต่อเนื่อง ตามคำสั่งการทำงานของการส่งข้อมูลตามมาตรฐานของ X10



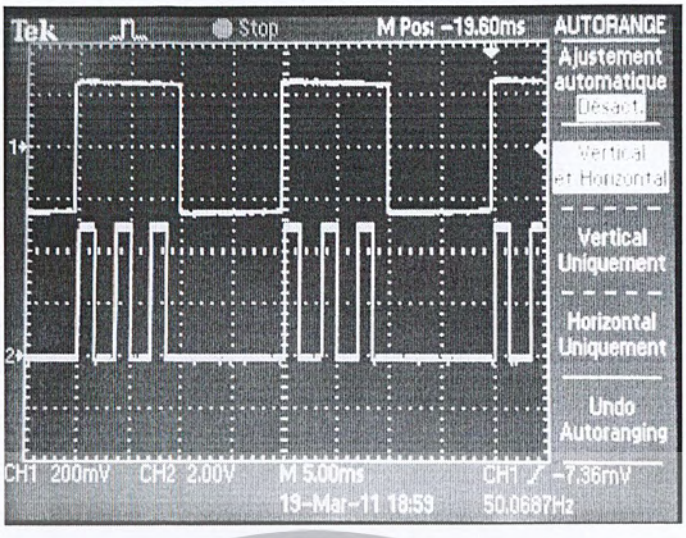

**ภาพที่ 4.8** แสดงการส่งข้อมูลแบบต่อเนื่อง

Ch 1 คือ สัญญาณที่ได้จากวงจรจุดตัดศูนย์

Ch 2 คือ สัญญาณ TTL ที่มาจากการส่งข้อมูล

- ทำการปรับ Time/Div ให้ลดลงเพื่อให้เห็นการส่งข้อมูลของสัญญาณ TTL จะเห็นได้ว่าในการส่งข้อมูลนั้นจะใช้สัญญาณจุดตัดศูนย์ในการให้จังหวะเวลาในการส่งข้อมูลด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 4.9** แสดงการส่งข้อมูลแบบต่อเนื่องโดยทำการปรับ Time/Div ให้ลดลง

Ch 1 คือ สัญญาณที่ได้จากวงจรจุดตัดศูนย์

Ch 2 คือ สัญญาณ TTL ที่มาจากการส่งข้อมูล

## 4.3 ทดลองการสั่งงานจากโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์กับวงจรภาครับ และภาคส่ง

**วิธีการทดลอง**

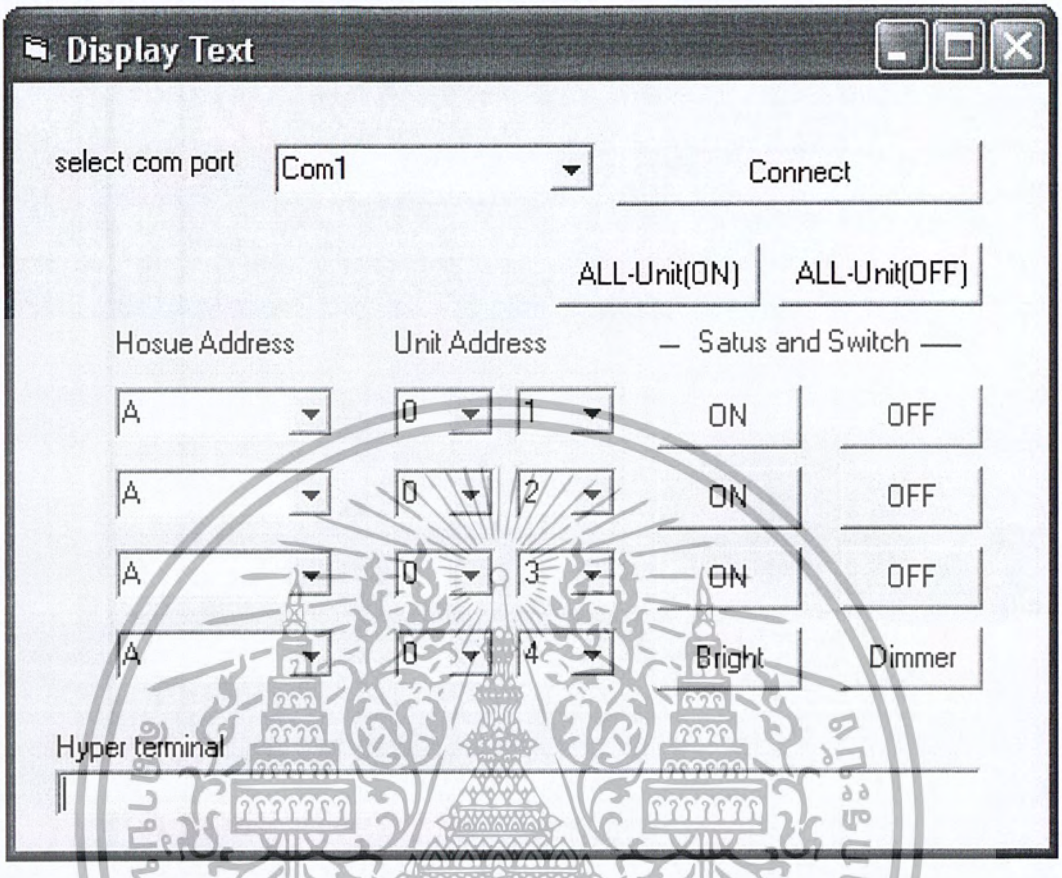
การทดลองสั่งงานให้ House Address A รับคำสั่งจากโปรแกรมควบคุมอุปกรณ์ผ่านทางคอมพิวเตอร์ โดยใช้โปรแกรม Proteus ในการแสดงการทำงาน

- เมื่อเลือกให้ House A ทำงานแล้วจะสามารถเลือกให้ Unit Address ที่ 01 02 และ 03 ทำงานได้

- โดย Unit Address ที่เราเลือกนั้นจะสามารถทำงานได้ต่อเมื่อมีการกดที่ปุ่มของโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์ว่าจะให้ On หรือ Off และยังสามารถเลือกได้อีกว่าจะให้ทั้ง Unit On หรือ Off โดยกดที่ปุ่ม ALL-Unit On หรือ ALL-Unit Off ได้อีกด้วย

- สามารถเลือกการใช้งานในรูปแบบของการหรี่ไฟได้ ด้วยการกดปุ่ม Bright หรือ Dimmer

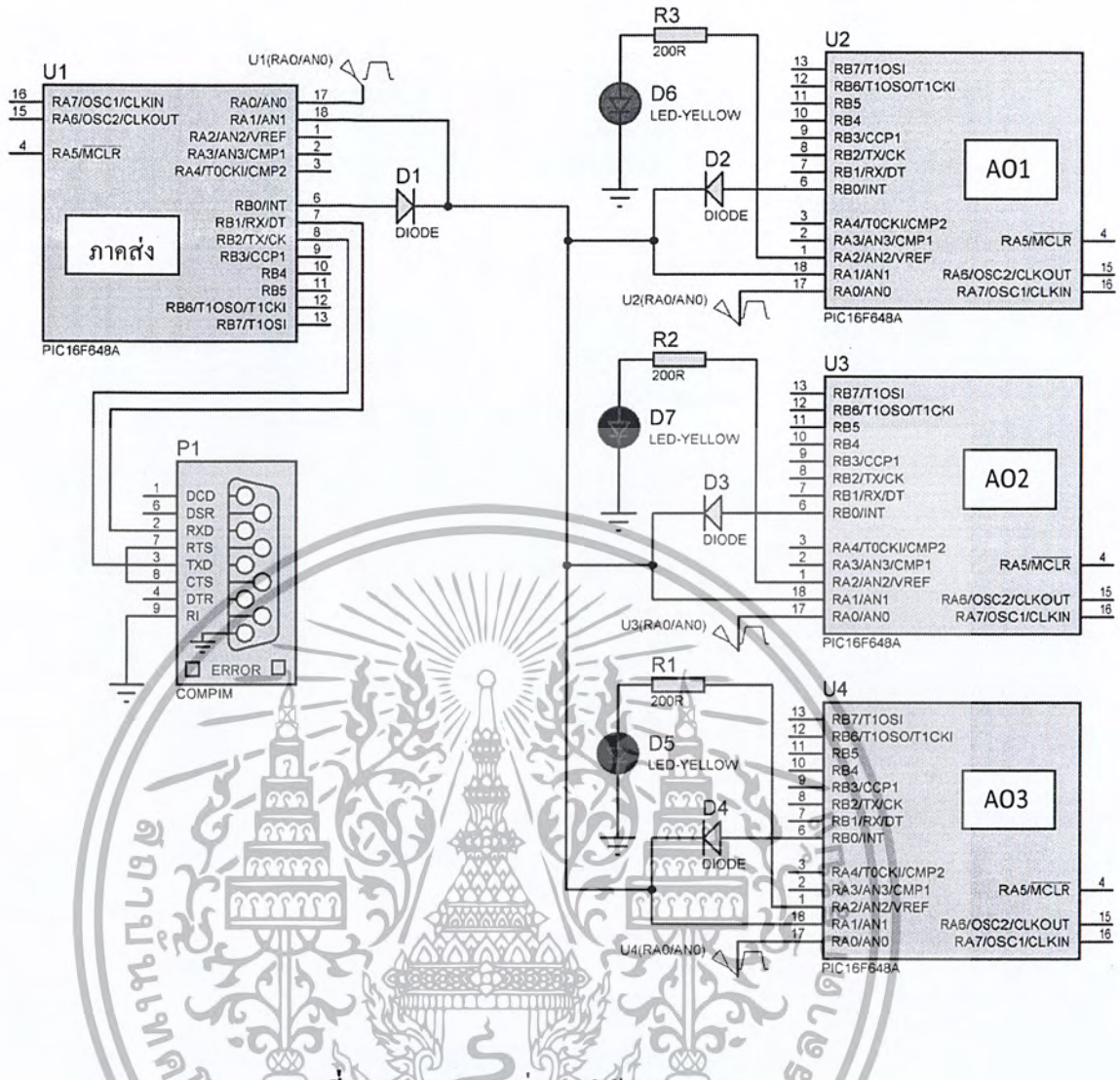
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 4.10** แสดงการสั่งงานผ่านทางโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์ ให้กับ House A ทำงาน

ผลการทดลองสั่งงานให้ House Address A รับคำสั่งจากโปรแกรมควบคุมอุปกรณ์ผ่านทางคอมพิวเตอร์ โดยใช้โปรแกรม Proteus ในการแสดงการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

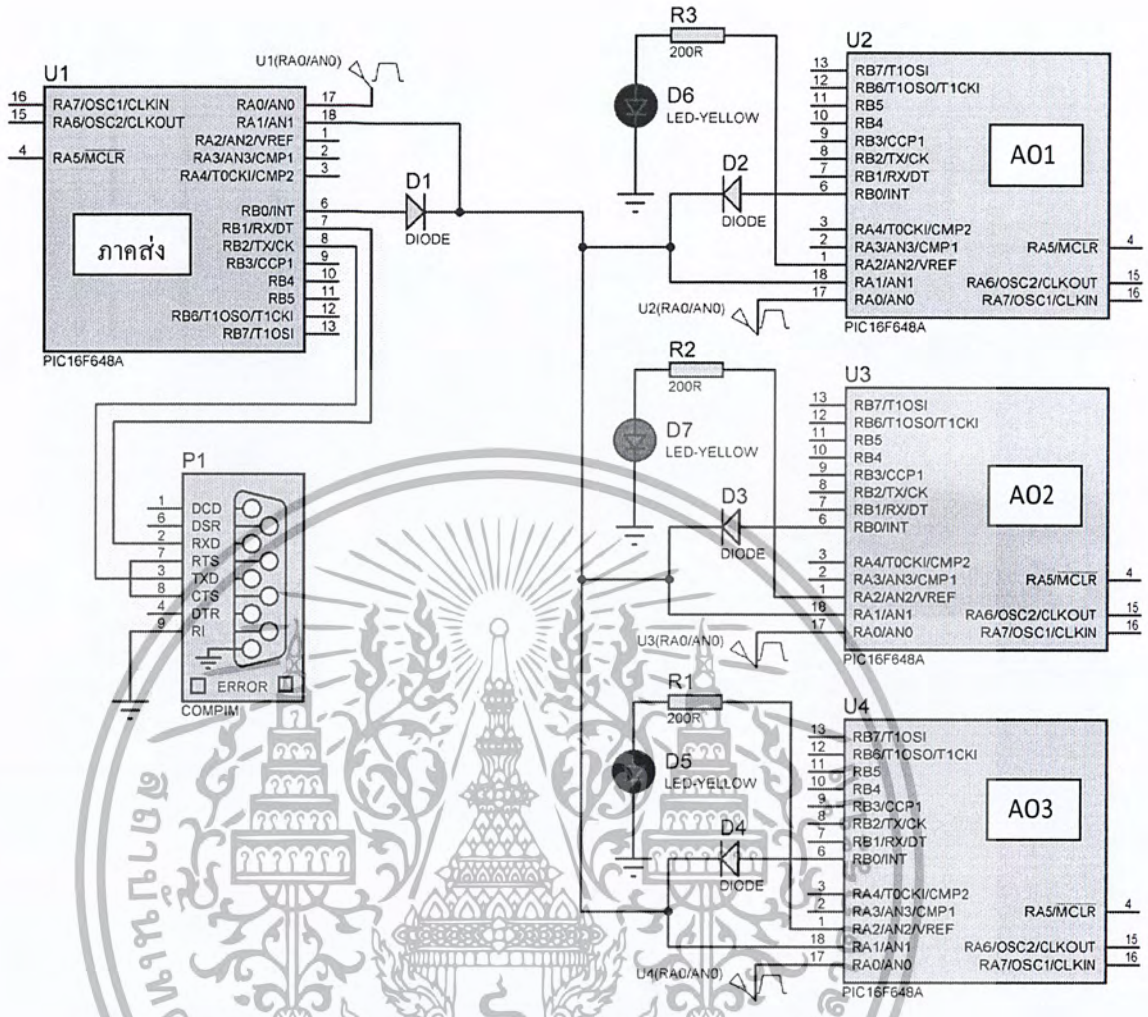



**ภาพที่ 4.11** แสดงการสั่งงานให้ A01 ทำงาน

- เมื่อกด (A01 ON) ในโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์ จะทำให้มีการแสดงผลการทำงานโดยหลอด LED ที่ต่ออยู่กับขา RA2 ของ PIC16F648A จะติดซึ่งสามารถนำสัญญาณ ที่ได้ไปใช้ในการควบคุมอุปกรณ์ไฟฟ้าอื่น ๆ ได้

- โดยเราจะใช้โปรแกรม Proteus ในการแสดงการทำงานโดยให้ Unit Address A04 เป็นภาคส่ง และ A01 A02 และ A03 เป็นภาครับ โดยใช้สายสัญญาณเส้นเดียวแทน AC-line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

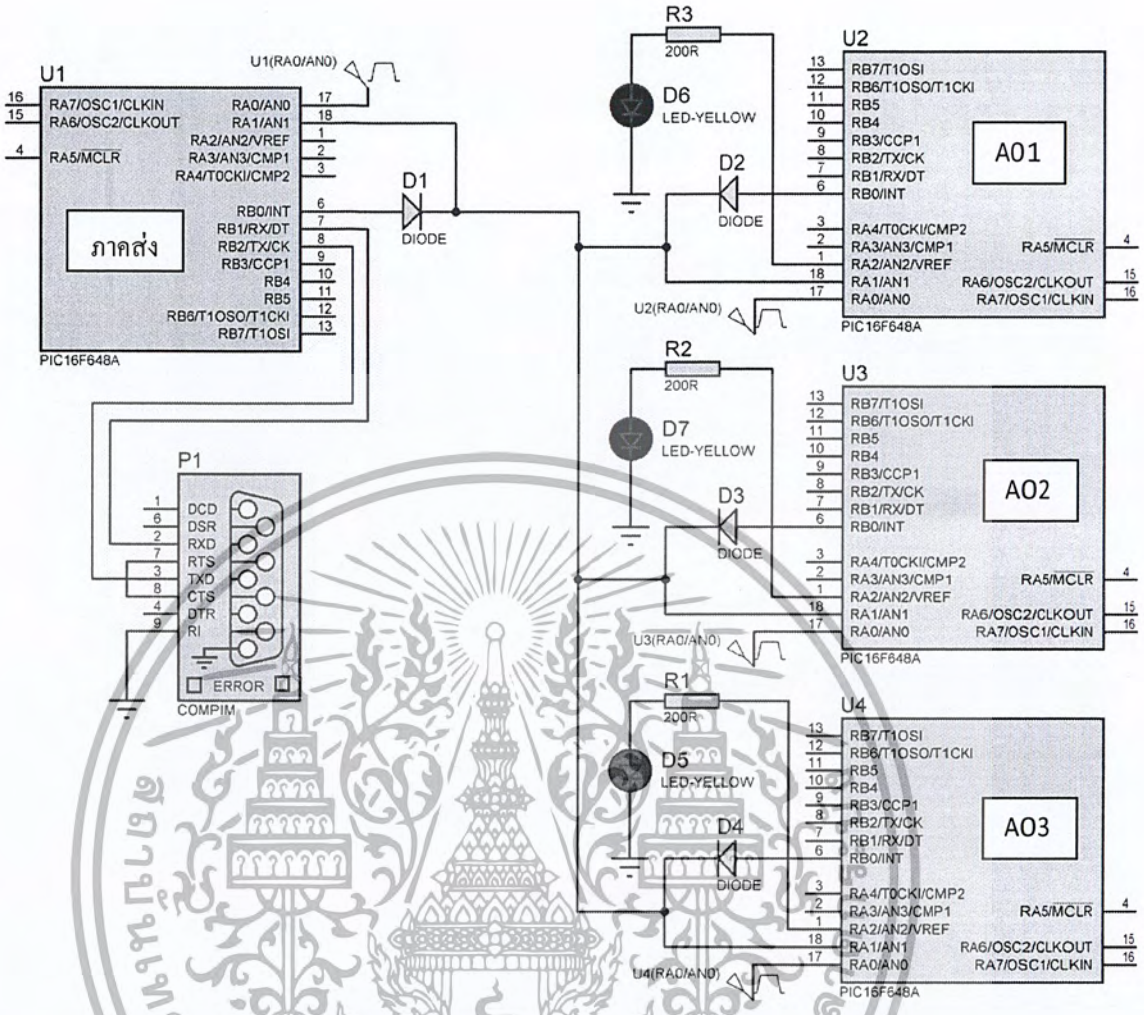



**ภาพที่ 4.12** แสดงการสั่งงานให้ A02 ทำงาน

- เมื่อกด (A02 ON) ในโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์ จะทำให้มีการแสดงผลการทำงานโดยหลอด LED ที่ต่ออยู่กับขา RA2 ของ PIC16F648A จะติดซึ่งสามารถนำสัญญาณ ที่ได้ไปใช้ในการควบคุมอุปกรณ์ไฟฟ้าอื่น ๆ ได้

- โดยเราจะใช้โปรแกรม Proteus ในการแสดงการทำงานโดยให้ Unit Address A04 เป็นภาคส่ง และ A01 A02 และ A03 เป็นภาครับ โดยใช้สายสัญญาณเส้นเดียวแทน AC-line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

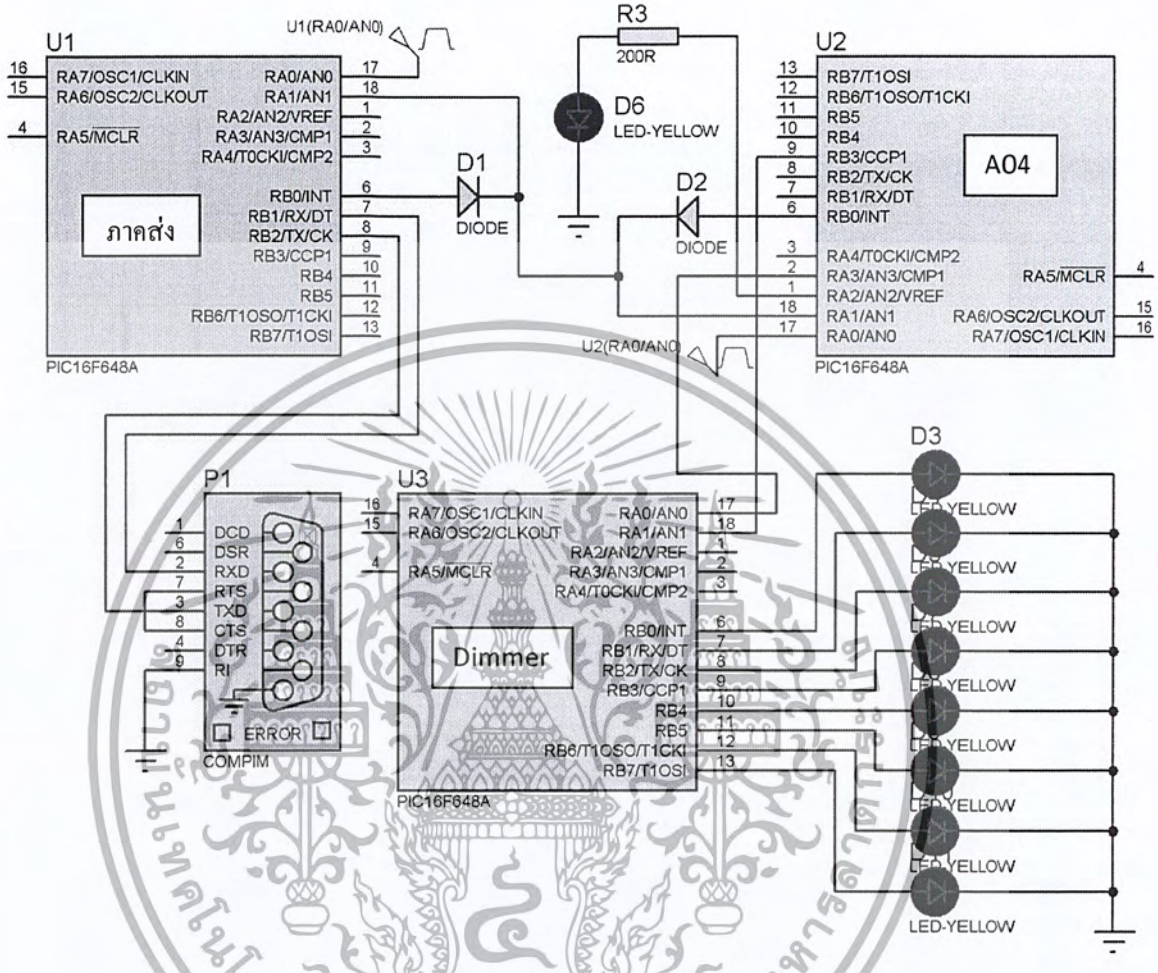



**ภาพที่ 4.13** แสดงการสั่งงานให้ A03 ทำงาน

- เมื่อกด (A03 ON) ในโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์ จะทำให้มีการแสดงผลการทำงานโดยหลอด LED ที่ต่ออยู่กับขา RA2 ของ PIC16F648A จะติดซึ่งสามารถนำสัญญาณ ที่ได้ไปใช้ในการควบคุมอุปกรณ์ไฟฟ้าอื่น ๆ ได้

- โดยเราจะใช้โปรแกรม Proteus ในการแสดงการทำงานโดยให้ Unit Address A04 เป็นภาคส่ง และ A01 A02 และ A03 เป็นภาครับ โดยใช้สายสัญญาณเส้นเดียวแทน AC-line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 4.14** แสดงการต่ออุปกรณ์ที่ใช้กับการสั่งงานโดยใช้คำสั่ง Bright และ Dimmer

- เมื่อกด Bright ที่ A04 ในโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์ จะทำให้หลอด LED ที่ต่ออยู่ค่อย ๆ สว่างขึ้นทีละดวง

- เมื่อกด Dimmer ที่ A04 ในโปรแกรมควบคุมอุปกรณ์ ผ่านทางคอมพิวเตอร์ จะทำให้หลอด LED ที่ต่ออยู่ค่อย ๆ ดับลงทีละดวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

## 5.1 สรุปและวิจารณ์ผลการทดลอง

### 5.1.1 การทดลองการทำงานของวงจรภาคส่ง

จากผลการทดลองจะเห็นได้ว่ามีสัญญาณความถี่ 120 กิโลเฮิรตซ์ ออกมาจากขาของ Transistor BC 547 ซึ่งจะนำความถี่ที่ได้ นั้นไปสั่งงานให้ Transistor 2SD667A ทำงานที่ความถี่ 120 กิโลเฮิรตซ์ เช่นเดียวกันจะทำให้เกิดการมอดูเลตความถี่ 120 กิโลเฮิรตซ์ ไปบนความถี่ 50 เฮิรตซ์ ของ AC-line ได้ แต่เนื่องจากมีปัญหาเรื่องการมอดูเลตสัญญาณ ทำให้ไม่สามารถส่งความถี่ 120 กิโลเฮิรตซ์ ไปกลับความถี่ของ AC-line จึงได้มีการทดลองส่งข้อมูลเพื่อให้อุปกรณ์ภาครับทำงาน โดยใช้สายสัญญาณในการส่งข้อมูล จากการทดลองส่งข้อมูลจากภาคส่ง จะมีสัญญาณเป็นสัญญาณพัลส์ 3 ลูกคลื่นเพื่อสั่งงานในขณะที่เราส่งข้อมูลเป็นลอจิก 1 ส่วนลอจิก 0 จะไม่มีสัญญาณพัลส์เกิดขึ้น เมื่อเราได้ลองสั่งงานตามมาตรฐานของ X10 นั้น อุปกรณ์ที่ทำงานเป็นภาครับ จะทำงานตามคำสั่งของโปรโตคอล X10 ที่มาจากการสั่งงานของอุปกรณ์ที่ทำงานเป็นภาคส่ง

### 5.1.2 การทดลองการทำงานของวงจรภาครับ

จากการทดลองจะเห็นได้ว่า เมื่อมีสัญญาณมาจาก AC-line แล้ว จะนำสัญญาณนั้นมาดีมอดูเลต โดยใช้ U1 (4069) ในการแปลงสัญญาณ 120 กิโลเฮิรตซ์ มาเป็นสัญญาณ TTL เพื่อสั่งงานให้ภาครับทำงาน จากการทดลองจะเห็นได้ว่า เมื่อมีสัญญาณเป็นพัลส์เข้ามาที่ขา X10-RX จะทำให้ไมโครคอนโทรเลอร์ทำงานตามคำสั่งที่ได้จากการส่งของภาคส่ง

## 5.2 ปัญหาของโครงงาน

- เนื่องจาก IF 455 กิโลเฮิรตซ์ นั้นไม่ค่อยเป็นที่นิยมในการใช้มอดูเลตสัญญาณเนื่องจากไม่ปลอดภัยในการใช้งาน และในปัจจุบันไม่เป็นที่นิยมใช้งานแล้วจึงทำให้การหาอุปกรณ์ เพื่อนำมาใช้ในการมอดูเลตสัญญาณนั้นหาได้ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เนื่องจากอุปกรณ์ที่ใช้ในการสร้างความถี่ 120 กิโลเฮิรตซ์ ที่จะทำหน้าที่ส่งความถี่ 120 กิโลเฮิรตซ์ เข้าไปในสายไฟฟ้ากระแสสลับนั้นไม่สามารถหาซื้อได้ตรงตามสเปคที่ต้องการใช้งาน จึงทำให้ไม่สามารถสร้างวงจรมอดูเลตสัญญาณ 120 กิโลเฮิรตซ์ได้

## 5.3 ข้อเสนอแนะ

ในการแก้ไขปัญหาเรื่องของการมอดูเลตนั้น อาจใช้การสั่งบอร์ดทดลองในเรื่องการมอดูเลตสัญญาณผ่านสายไฟฟ้ากระแสสลับโดยมาตรฐานX10 เพื่อที่จะใช้ทดลองในการส่งข้อมูลผ่านสายไฟฟ้ากระแสสลับได้ แต่เนื่องจากการสั่งบอร์ดทดลองนั้นมีระยะเวลาในการจัดส่งที่นาน และมีค่าใช้จ่ายสูง จึงทำให้ไม่เหมาะแก่การนำมาใช้ในโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


[1] Behrouz A. Forouzan,Sophia Chung Fegan.**Data Communication and Networking**. กรุงเทพมหานคร:สำนักพิมพ์ท้อป.2550.

[2] ธเนตร นามโมรา,ธรรมรัตน์ แซ่โค้ว. **"ระบบควบคุมอุปกรณ์ไฟฟ้าผ่านสายไฟบ้าน."** ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.2546.

[3] นพพล สุนทรชิ้นฑ์,ประชา ดวงใจ. **"ระบบไฟฟ้าในบ้าน."** ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.2544.

[4] เจนจบ วีระพานิชเจริญ. **"การสื่อสารบรอดแบนด์ความเร็วสูงผ่านสายไฟฟ้า."**[Online].Avilable : http://www.thaitelecomkm.org/TTE/topic/attach/Broadband_over_Power_Lines/index.php

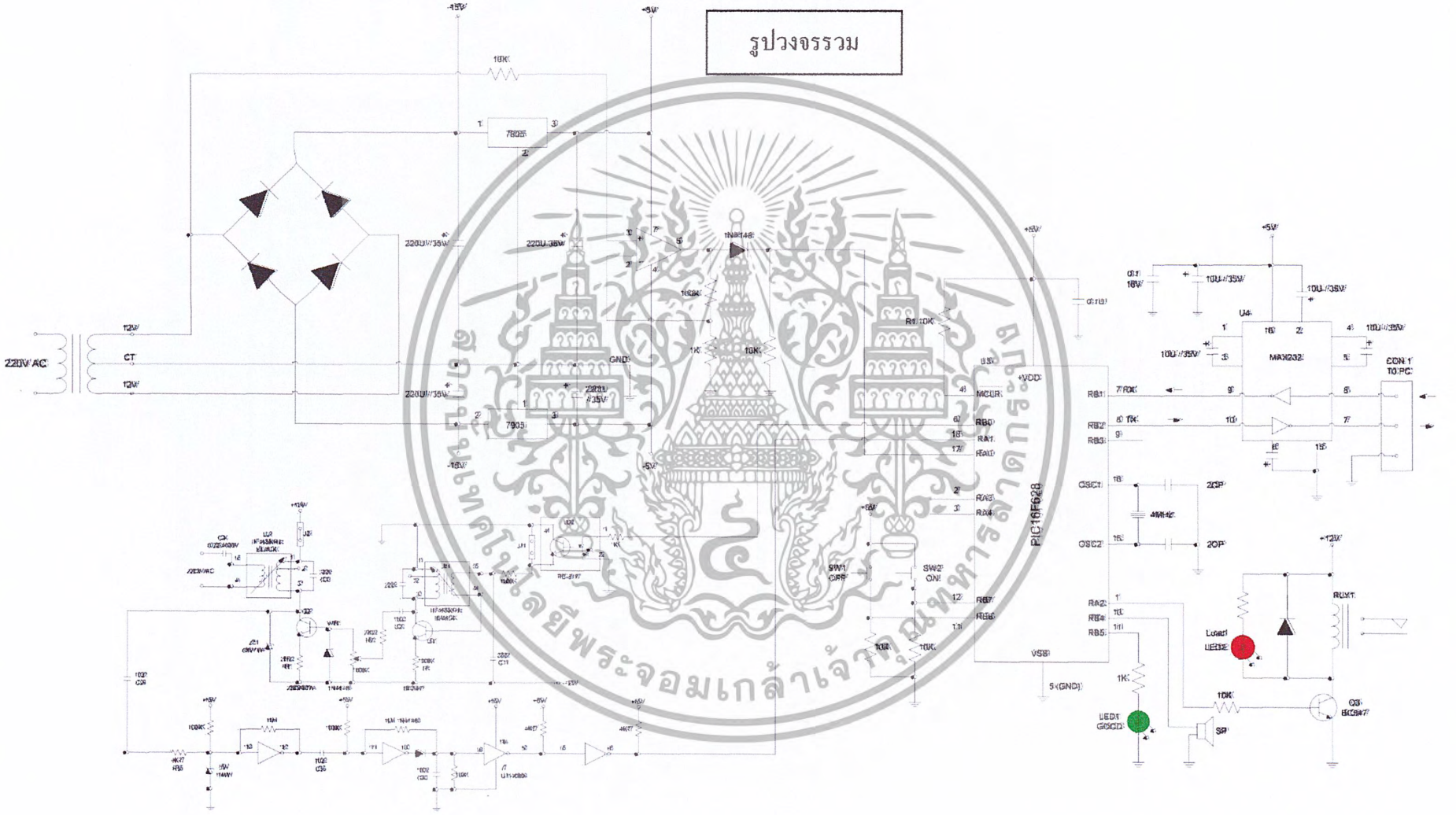


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

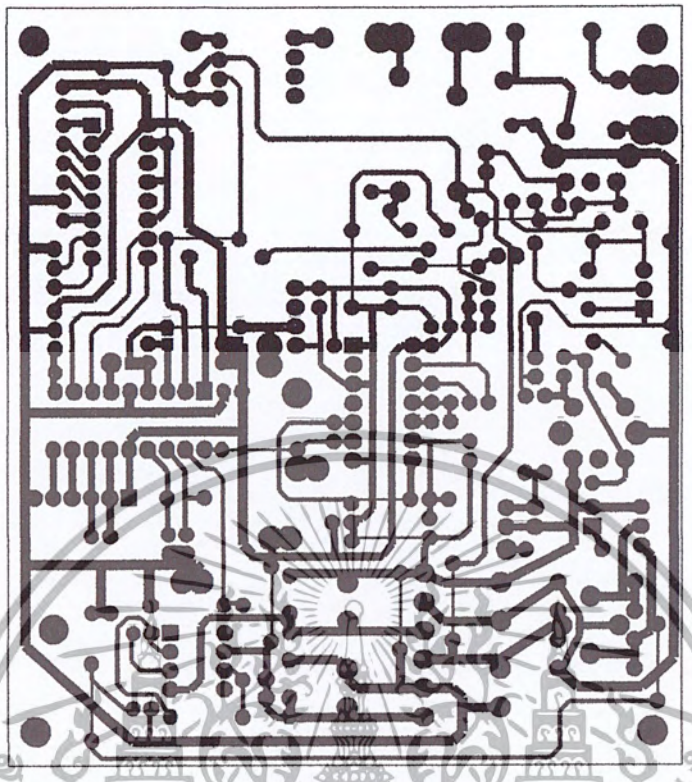
# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

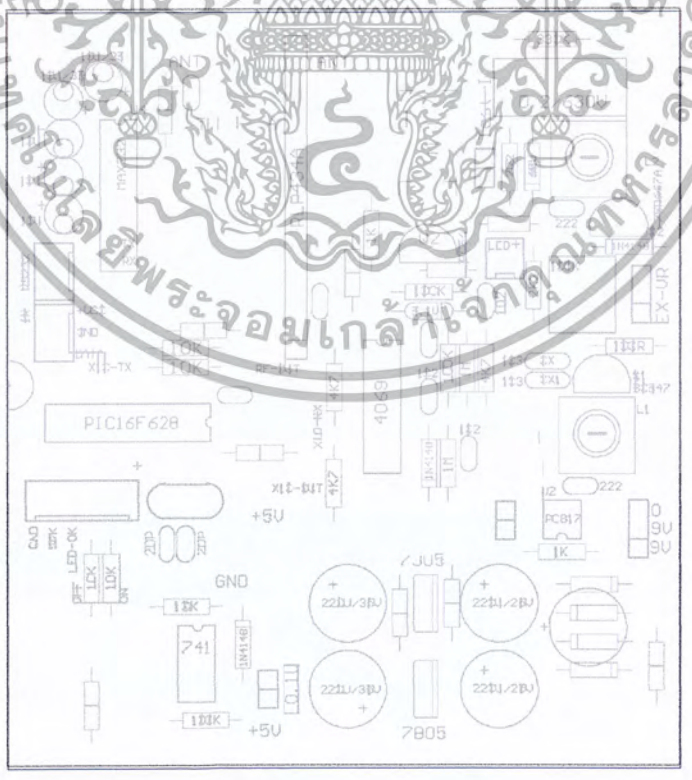
# รูปวงจรรวม


220V AC
CT
12V
7805
7905
1N4148
MAX232
PIC16F628
MCLR
VDD
VSS
OSC1
OSC2
4MHz
LED1
LED2
Load1
SW1
SW2
SP
BC847
RELAY
10K
1K
2200U/35V
10U/35V
0.1U
18P

ลายวงจรที่ใช้ในการกัดแผ่นปริ้นต์

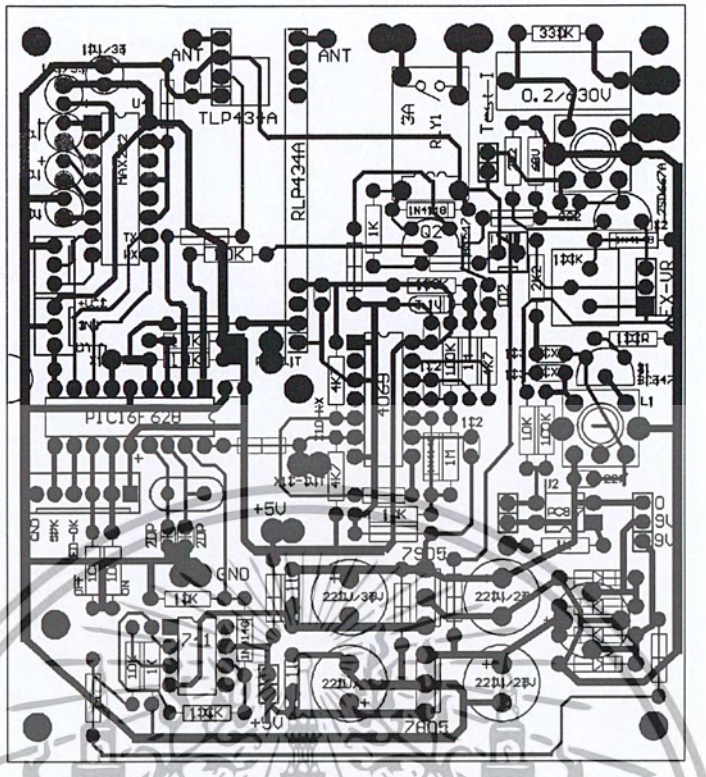


รูปลายปริ้นต์ด้านล่าง




รูปการวางอุปกรณ์ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปลายวงจรและการวางอุปกรณ์ในแผ่นปริ้นต์

รายการอุปกรณ์ที่ใช้ในการทำวงจรภาครับ และภาคส่ง

| อุปกรณ์ | จำนวน |
|---|---|
| Zener diode 68V 1W | 1 |
| Transistor 2sd667a | 1 |
| C 222 | 3 |
| C 102 | 3 |
| C 103 | 1 |
| Diode 1N4148 | 2 |
| Zener diode 5V ¼ W | 1 |
| BC 547 | 1 |
| PC 817 | 1 |
| HEF4069UB | 1 |
| R 1 M | 2 |
| R 100 K | 3 |
| R 10 K | 1 |
| R 100 | 1 |
| R 2.2 | 1 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| R 2.2K | 1 |
| R 4.7 K | 3 |
| VR 100 K | 1 |

รายการอุปกรณ์ที่ใช้ในการทำวงจรไมโครคอนโทรลเลอร์

| อุปกรณ์ | จำนวน |
|---|---|
| switch | 2 |
| R 10 K | 4 |
| BC 547 | 1 |
| Crystals | 1 |
| Relay | 1 |
| R 1 K | 1 |
| C 0.1 | 2 |
| C 20 p | 2 |
| Max232 | 1 |
| PIC16F648A | 1 |
| C 10 U 35 V | 5 |
| Diode 1N4148 | 1 |

รายการอุปกรณ์ที่ใช้ในการทำวงจรจ่ายไฟ และ จุดตัดศูนย์

| อุปกรณ์ | จำนวน |
|---|---|
| หม้อแปลง 12 V | 1 |
| bridge | 1 |
| 7805 | 1 |
| 7905 | 1 |
| Op amp 741 | 1 |
| C 220 U 35 V | 4 |
| R 10 K | 2 |
| R 100 K | 1 |
| R 1 K | 1 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**โค้ดที่ใช้ในการสั่งงานในส่วนของไมโครคอนโทรลเลอร์**

```
//RA (11110011)
//0   ZERO-Crossing
//1   X10-RX
//2   X10-OUTPUT
//3   UHF-TX
//4   UHF-RX
//5   (Any input)
//
//RB (11000010)
//0   X10-TX
//1   RS232 RX
//2   RS232 TX
//3   X10EX   (X10-Lamp/IR)
//4   Speaker
//5   LED-OK
//6   SW-ON
//7   SW-OFF

#include <16F648A.h>
//#include <16F628.h>
#include <string.h>
#include <stdlib.h>

#use delay(clock=4000000)
#use rs232(baud=9600,parity=N,xmit=PIN_B2,rcv=PIN_B1,bits=8,FORCE_SW)
#fuses XT,PUT,BROWNOUT,MCLR,NOWDT,NOPROTECT,NOLVP

#case
#use fast_io(A)
#define X10_ZERO_CROSS  PIN_A0
#define X10READPIN      PIN_A1
#define X10OUT          PIN_A2
#define X10UHFTX        PIN_A3
#define X10UHFRX        PIN_A4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define X10WRITEPIN     PIN_B0
#define X10EX           PIN_B3
#define SPEAKER         PIN_B4
#define LEDOK           PIN_B5
#define SWOFF           PIN_B6
#define SWON            PIN_B7

#define X10WAIT         800
#define X10_STARTBIT    0B11100000

//X-10 Function Code
#define X10FN_ALL_UN_OFF    1
#define X10FN_ALL_UN_ON     2
#define X10FN_ON            3
#define X10FN_OFF           4
#define X10FN_DIM           5
#define X10FN_BRIGHT        6
#define X10FN_ALL_LG_OFF    7
#define X10FN_EXT_CODE      8
#define X10FN_HAIL_REQUEST  9
#define X10FN_HAIL_ACK      10
#define X10FN_PRE_SET_DIM   11
#define X10FN_EXT_CODE_AG   12
#define X10FN_STATUS_ON     13
#define X10FN_STATUS_OFF    14
#define X10FN_STATUS_REQ    15

#define PORTA           0x05
#define PORTB           0x06

#define KEYDELAY        50
#define KEY_ON          0x80
#define KEY_OFF         0x40
#define CMDSIZE         6

//Sound
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define   Speaker         PORTB       //Speaker Port
#define   Spkbit          0B00010000    //Speaker bit is   ra.0

#rom  0x2100={'A',0x04}       //Init EEPROM

//Compliment of House & Keycode (is same)
int const X10NumCode[16]
={0x69,0xA9,0x59,0x99,0x56,0x96,0x66,0xA6,0x6A,0xAA,0x5A,0x9A,0x55,0x95,0x65,0xA5};
//House Code
char const X10HouseCode[16]={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P'};
//Compliment of Function Code
int const X10FnCode[15]
={0x55,0x56,0x59,0x5A,0x65,0x66,0x69,0x6A,0x95,0x96,0x99,0xA5,0xA6,0xA9,0xAA};
//char const StrFn[12][22]={"ALL Units Off","ALL Units On","On","Off","Dim","Bright","All Lights
Off","Extended Code","Hail Request","Hail Acknowledge",
//"Pre-set Dim","Extended Code(Analog)","Status = On","Status = Off","Status Request"};

int StartCode,HouseCode[2],KeyCode[2],kType[2];
int rHouse,rNumber,rFunction;
char StrFunc[22];
char PKStart,RxHouse;
char CmdBuff[CMDSIZE+1];
int PKIndex;
int RxNo,RxFn;

//&sound
//*** Sound ***
//Working send out put to toggle Spkbit
//XTAL 4MHz
void Sound(char freq,char dura)
{
int dl,cl;
 #asm

  MOVLW   50
  MOVWF   dl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SndNx:
  MOVF   freq,W
  MOVWF   cl
  MOVLW   Spkbit
  XORWF   Speaker,F
SndLp:
  NOP
  NOP
  NOP
  DECFSZ   cl,F
  GOTO   SndLp

  DECFSZ   dl,F
  GOTO   SndNx

  MOVLW   50
  MOVWF   dl
  DECFSZ   dura,F
  GOTO   SndNx

   //BCF    Speaker,7     //Clear Speaker bit low
 #endasm
// output_low(PIN_B7);
}
void SoundOK(void)
{
 Sound(25,20);
}
void SoundHL(void)
{
 Sound(25,75);
}
void SoundFn(void)
{
 Sound(20,35);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void SoundERR(void)
{
 Sound(210,10);
}
void SoundSuccess(void)
{
 Sound(25,10);
 delay_ms(50);
 Sound(25,10);
 delay_ms(50);
 Sound(25,10);
}
void SoundPress(void)
{
 Sound(35,10);
}

//END Sound

char ReadPortA(void)
{
 #asm
   movf  PORTA,0   //mov to W
   movwf _RETURN_
 #endasm
}
char ReadPortB(void)
{
 #asm
   movf  PORTB,0    //mov to W
   movwf _RETURN_
 #endasm
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//X-10 Function ************************************
//Store Function detail in StrFunc[];
//Input number of Function 1-n
void StrFuncMsg(int n)
{
    switch(n)
    {
   case 1:
        strcpy(StrFunc,"ALL Units Off");
      break;
   case 2:
        strcpy(StrFunc,"ALL Units On");
      break;
   case 3:
        strcpy(StrFunc,"ON");
      break;
   case 4:
        strcpy(StrFunc,"OFF");
      break;
   case 5:
        strcpy(StrFunc,"Dim");
      break;
   case 6:
        strcpy(StrFunc,"Bright");
      break;
   case 7:
        strcpy(StrFunc,"All Lights Off");
      break;
   case 8:
        strcpy(StrFunc,"All Lights Off");
      break;
   case 9:
        strcpy(StrFunc,"Hail Request");
      break;
   case 10:
        strcpy(StrFunc,"Hail Acknowledge");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
    case 11:
         strcpy(StrFunc,"Pre-set Dim");
        break;
    case 12:
         strcpy(StrFunc,"Extended Code(Analog)");
        break;
    case 13:
         strcpy(StrFunc,"Status = On");
        break;
    case 14:
         strcpy(StrFunc,"Status = Off");
        break;
    case 15:
         strcpy(StrFunc,"Status Request");
        break;
    default:
        strcpy(StrFunc,"Unknow");
     }
}
//-------------------------------------
void InitialChip(void)
{
  setup_comparator(NC_NC_NC_NC);  //Input Digital
  output_a(0);
  output_b(0B00000100);
  set_tris_a(0B11110011);
  set_tris_b(0B11000010);
}

//Find House
char FindHouseCode(int H)
{
int i;

   for(i=0;i<16;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
     if(H==X10NumCode[i])
       return X10HouseCode[i];
    }
  return 'X';
}
int FindKeyCode(int K)
{
int i;

    for(i=0;i<16;i++)
    {
     if(K==X10NumCode[i])
       return (i+1);
    }
  return 0;
}
//Find Function Code
//Return    0    not found
//         (i+1) found in Table
int FindFnCode(int F)
{
int i;

    for(i=0;i<15;i++)
    {
     if(F==X10FnCode[i])
     {
       //printf("-%X-",X10FnCode[i]);
       return (i+1);
     }
    }
  return 0;
}
//Check bit 1 of Start code  1110
//Return 0 is not found, 1 is found
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int X10READY(void)
{
 return(input(X10READPIN));
}
//Check ZeroCrossing
void CheckZeroCross()
{
 if(input(X10_ZERO_CROSS))          //If = 1
    while(input(X10_ZERO_CROSS)) ;    //wait 0
 else
    while(!input(X10_ZERO_CROSS)) ;   //if = 1 wait 0
}

//Read Data n bit form AC-Line Sync with ZeroCrossing
//Input  n
//Output XData
int X10ReadBit(int n)
{
int xdata,i;

  xdata=0;
  for(i=0;i<n;++i)
  {
   xdata |= input(X10READPIN);    //Save bit
   CheckZeroCross();
   delay_us(X10WAIT);
   if(i!=(n-1))                  //Check done
    xdata <<=1;                  //Shift Left one
  }
 return xdata;
}
//Check Start Code if good
//Read n Record in to buffer HouseCode[],KeyCode[],kType;
int X10ReadBuff(int n)
{
int i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Start Code
  for(i=0;i<n;i++)
  {
   StartCode=X10ReadBit(4);
   if(StartCode!=0B00001110)
    return 0;

   HouseCode[i]=X10ReadBit(8);
   KeyCode[i]=X10ReadBit(8);
   kType[i]=X10ReadBit(2);
  }

 return 1;
}

//Read Record
//Return 0  not Valid
//      1  Valid Record and Get Data in rHouse,rNumber,rFunction and get StrFunc
int X10Read(void)
{
int i,R;

  R=0;
  if(X10ReadBuff(2))
  {
   for(i=0;i<2;i++)
   {
    switch(kType[i])
    {
    //Address (Get HouseCode and NumberCode)
    case 1:
       output_high(LEDOK);
       if(i==1)
       {
        if(HouseCode[0]==HouseCode[1] && KeyCode[0]==KeyCode[1] && kType[0]==kType[1])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          {
           rHouse=FindHouseCode(HouseCode[1]);
           rNumber=FindKeyCode(KeyCode[1]);
          }
         }
        break;
      //Function (Get HouseCode and FunctionCode)
      case 2:
        if(i==1)
        {
         if(HouseCode[0]==HouseCode[1] && KeyCode[0]==KeyCode[1] && kType[0]==kType[1])
         {
          rFunction=FindFnCode(KeyCode[1]);
          StrFuncMsg(rFunction);
          output_low(LEDOK);
          R=1;
         }
        }
        break;
    }
   }

  }

 return R;
}
//X10 write 1 bit 3 time Sync with Zero Crossing
void X10WriteBit(int b)
{
int i;
//@
  CheckZeroCross();
  delay_us(500);

  for(i=0;i<3;i++)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    output_bit(X10WRITEPIN,b);
    delay_us(1400);
    output_low(X10WRITEPIN);
    if(i<2)
     delay_ms(2);
   }
}
//X10 write 1 byte (MSB write)
//Input Data  data
//     nBit
void X10WriteByte(int Data,int nBit)
{
int i;
BOOLEAN xBit;

  for(i=0;i<nBit;++i)
  {
   xBit=shift_left(&Data,1,0);
   X10WriteBit(xBit);
  }
}
//X10 write Block
//Input is  House,Number,Function
void X10Write(char House,int Number,int Fn)
{
int HouseCode,i;

  //printf("<\r\n");
  HouseCode=House-0x41;                  //Change to index
 //Address+Number twice
  for(i=0;i<2;i++)
  {
   X10WriteByte(X10_STARTBIT,4);           //Start bit
   X10WriteByte(X10NumCode[HouseCode],8);    //House A-P
   X10WriteByte(X10NumCode[Number-1],8);     //Number
   X10WriteByte(0B01000000,2);             //Alway 01
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    X10WriteByte(0B00000000,6);              //Gap
  //Address+Function twice
    for(i=0;i<2;i++)
    {
     X10WriteByte(X10_STARTBIT,4);           //Start bit
     X10WriteByte(X10NumCode[HouseCode],8);  //House
     X10WriteByte(X10FnCode[Fn-1],8);        //Function
     X10WriteByte(0B10000000,2);             //Alway 10
    }
}
//X10 write Only
//Input is  House,Function
void X10WriteFn(char House,int Fn)
{
int HouseCode,i;

   //printf("<\r\n");
    HouseCode=House-0x41;                   //Change to index
  //Address+Function twice
    for(i=0;i<2;i++)
    {
     X10WriteByte(X10_STARTBIT,4);          //Start bit
     X10WriteByte(X10NumCode[HouseCode],8); //House
     X10WriteByte(X10FnCode[Fn-1],8);       //Function
     X10WriteByte(0B10000000,2);            //Alway 10
    }
}
//Execute Function code
void X10FuncExe(int House,int Number,int Fn)
{
int H,N,P;

  //Get Address form EEPROM House & Number
    H=read_eeprom(0);
    N=read_eeprom(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Private Function
  if(H==House && N==Number)
  {
   switch(Fn)
   {
    case X10FN_ON:                  //ON
     output_high(X10OUT);
     break;
    case X10FN_OFF:                 //OFF
     output_low(X10OUT);
     break;
    case X10FN_STATUS_REQ:                  //12 ON ,13 OFF
      P=ReadPortA();
      P&=0B00000100;
      if(P==0B00000100)
        X10Write(H,N,X10FN_STATUS_ON);     //X10Write(H,N,12);
      else
        X10Write(H,N,X10FN_STATUS_OFF);
      break;
   }
  }

//ALL Unit Function
  switch(Fn)
  {
   case X10FN_ALL_UN_OFF:           //All Units Off
   case X10FN_ALL_LG_OFF:           //All Lights Off
    output_low(X10OUT);
    break;

   case X10FN_ALL_UN_ON:            //All Units On
    output_high(X10OUT);
     break;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
//Util**********************************

int ScanKey(void)
{
int D1,D2,Key;
int Cn;

  Key=0;
  D1=ReadPortB();          //Read again
  D1&=0B11000000;

  if(D1==0)                //No Key press
   return 0;
  else
  {
   delay_ms(KEYDELAY);     //Delay for test
   D2=ReadPortB();         //Read again
   D2&=0B11000000;
   if(D1==D2)
   {
    Key=1;                 //Key press is normal
    Cn=0;
    do
    {
     //restart_wdt();
     delay_ms(30);
     D2=ReadPortB();       //Read again
     D2&=0B11000000;
     if(Key==1)            //Normal press only
     {
      Cn++;
      if(Cn==100)
      {
       output_low(X10OUT);
       SoundFn();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Key=2;                //Key press is FnKey
      Cn=0;
     }
    }
    //while(D2!=0 && Key==2);  //wait for FnKey Release
   }
   while(D2!=0);            //Wait for Release Key

   if(Key==1)
   {
    switch(D1)
    {
     case KEY_ON:
        output_high(X10OUT);
        break;
     case KEY_OFF:
        output_low(X10OUT);
        break;
    }
    SoundPress();
   }

  }
 }
return Key;
}
void SetAddress(void)
{
unsigned int32 Cn;
  Cn=0;
  while(TRUE)
  {
   if(X10READY())
   {
    if(X10Read())
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      write_eeprom (0,rHouse);
      write_eeprom (1,rNumber);
      printf("NEW ADDRESS=%C%02d\r\n",read_eeprom(0),read_eeprom(1));
      SoundSuccess();
      break;
     }
    }
    Cn++;              //wait for 5 sec
    if(Cn>=250000)
    {
     SoundERR();
     printf("Setup Abort\r\n");
     break;
    }
   }
}
void PrintHelp(void)
{
   printf("\r\nEnter House,Number-Function(6 Character)\r\n");
   printf("Format HNN-FF\r\n");
   printf("   H=House in upper case (A-P)\r\n");
   printf("   NN=Number with Zero (01-16)\r\n");
   printf("   FF is Function with Zero (01-15)\r\n");
   printf("\r\nExample-1\r\nA01-03<ENTER>  A01(ON)\r\n");
   printf("A01-04<ENTER>  A01(OFF)\r\n");
   printf("\r\nEnter House,Function(3 Character)\r\n");
   printf("Format HFF\r\n");
   printf("   H=House in upper case (A-P)\r\n");
   printf("   FF is Function with Zero (01-15)\r\n");

   printf("\r\nExample-2\r\nA01<ENTER>  A(All Units Off)\r\n");
   printf("A05<ENTER>  A(Dim)\r\nA06<ENTER>  A(Bright)\r\n");
   printf("Number Function Valid is 1,2,5,6,7\r\n");
   printf("Pres ? for Help\r\n");
}
int CheckCmd3(char *Str)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
int len,i,R;
char Buff[3];
   RxHouse=CmdBuff[0];
//Check Limit of Function
   R=1;
   if(CmdBuff[1]<'0'|| CmdBuff[1]>'9')
     R=0;
   else
   {
    if(CmdBuff[2]<'0'|| CmdBuff[2]>'9')
     R=0;
   }
//Check Value of Number & Function
   if(R==1)
   {
    //Test Function valid
     strncpy(Buff,CmdBuff+1,2);
     Buff[2]=0;
     RxFn=atoi(Buff);
     switch(RxFn)
     {
      case 1:    //All Units Off
      case 2:    //All Units On
      case 5:    //Dim
      case 6:    //Bright
      case 7:    //All Lights Off
         break;
      default:
          R=0;
     }
   }
return R;
}
int CheckCmd6(char *Str)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int len,i,R;
char Buff[3];
   RxHouse=CmdBuff[0];
//Check Limit of Number & Function
   R=1;
   for(i=1;i<6;i++)
   {
    switch(i)
    {
     case 1:
     case 2:
     case 4:
     case 5:
        if(CmdBuff[i]<'0'|| CmdBuff[i]>'9')
         R=0;
        break;
     case 3:
        if(CmdBuff[i]!='-')
         R=0;
        break;
    }
   }
//Check Value of Number & Function
   if(R==1)
   {
    //Test number valid
    strncpy(Buff,CmdBuff+1,2);
    Buff[2]=0;
    RxNo=atoi(Buff);
    if(RxNo<1 || RxNo>16)
    {
     R=0;
     //printf("\r\nNumber Error\r\n");
    }
    else
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      strncpy(Buff,CmdBuff+4,2);
      Buff[2]=0;
      RxFn=atoi(Buff);
      if(RxFn<1 || RxFn>15)
      {
       R=0;
       //printf("\r\nFunction Error\r\n");
      }
     }
    }

 return R;
}
void UARTReady(void)
{
char c,R;
int len;
//signed int R;
   if(kbhit())
   {
    c = getc();
    if(PKStart==0)
    {
     if(c>='A'&& c<='P')
     {
      PKStart=1;
      PKIndex=0;
      CmdBuff[PKIndex]=c;
      PKIndex++;
      //printf("Start\r\n");
      printf("%c",c);
     }
     else
     {
      if(c=='?')
       PrintHelp();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 else
   printf("House code Error or not use Upper Letter\r\n");
 }
}
else
{
 if(c=='\r'||c=='\n')        //check <ENTER> End command
 {
  CmdBuff[PKIndex]=0;
  printf("\r\n");
  len=strlen(CmdBuff);
  if(len!=6 && len!=3)
   printf("Command Length Error\r\n");
  else
  {
   //Check Record is Valid
    if(len==3)
     R=CheckCmd3(CmdBuff);
    else
    {
     if(len==6)
      R=CheckCmd6(CmdBuff);
    }
    if(R==0)
     printf("Command Error\r\n");
    else
    {
     if(len==3)   //House,Function
     {
      printf("Sending...%c%02d\r\n",RxHouse,RxFn);
      X10WriteFn(RxHouse,RxFn);
      if(RxFn==1||RxFn==2)
       X10FuncExe(RxHouse,RxNo,RxFn);
     }
     if(len==6)   //House,Number,Function
     {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            printf("Sending...%c%02d-%02d\r\n",RxHouse,RxNo,RxFn);
            X10Write(RxHouse,RxNo,RxFn);
            if(RxFn==3||RxFn==4)
             X10FuncExe(RxHouse,RxNo,RxFn);
           }
          }
        }
        PKStart=0;
        PKIndex=0;
       }
       else
       {
        printf("%c",c);
        CmdBuff[PKIndex]=c;       //store in buffer
        PKIndex++;
      //Test buffer full
        if(PKIndex==CMDSIZE+1)
        {
         CmdBuff[PKIndex]=0;
          //DoCmd(CmdBuff);
         //else
          //printf("Command Error\r\n");
         printf("Buffer Full\r\n");
         PKStart=0;
         PKIndex=0;
        }
       }
      }
    }
}
//int X10flag;
void main(void)
{
int Key;
  InitialChip();
  output_high(LEDOK);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   delay_ms(500);
   output_low(LEDOK);
   printf("X10-LAB DEMO-2 ");
//default rHouse & rNumber
   rHouse=read_eeprom(0);
   rNumber=read_eeprom(1);
   printf("ADDRESS=%C%02d\r\n",rHouse,rNumber);
   printf("Press ? for Help\r\n");
   SoundSuccess();

   PKStart=0;
   PKIndex=0;
   while (TRUE)
   {
    UARTReady();
    //Check X10 Line Status
    if(X10READY())
    {
     if(X10Read())
     {
      switch(rFunction)
      {
       case 1:
       case 2:
          printf("(Fn%02d)-%s\r\n",rFunction,StrFunc);
          break;
       case 3:
       case 4:
       case 5:
       case 6:
       case 7:
       case 8:
       case 9:
       case 10:
       case 11:
       case 12:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      case 13:
      case 14:
      case 15:
         printf("%c%02d(Fn%02d)-%s\r\n",rHouse,rNumber,rFunction,StrFunc);
         break;
      default:
         printf("UnKnow Function\r\n");
      }
      X10FuncExe(rHouse,rNumber,rFunction);
      SoundOK();
      }
     }

     Key=ScanKey();
     if(Key==2)
     {
      output_high(LEDOK);
      printf("Setup Address\r\n");
      SetAddress();
      output_low(LEDOK);
     }

   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้