สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน ZIGBEE

ELECTRICAL EQUIPMENT CONTROL SYSTEM BY USING ZIGBEE

T104332

โดย

นางสาวกฤตภัค เกตุเรือง 48010021

นายกษิดิ์เดช ชื่นชูผล 48010033

นางสาวกาญจนา ลิ้มวัฒนาชัย 48010041

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 104332
วัน,เดือน,ปี - 2 พ.ย. 2552

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

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน ZIGBEE

ELECTRICAL EQUIPMENT CONTROL SYSTEM BY USING ZIGBEE

โดย

นางสาวกฤตภัค เกตุเรือง 48010021

นายกษิดิ์เดช ชื่นชูผล 48010033

นางสาวกาญจนา ลิ้มวัฒนาชัย 48010041

ภาควิชา
วิศวกรรมโทรคมนาคม
ผ่านการตรวจรูปเล่มแล้ว
(ลงชื่อ).....ผู้ตรวจ

อาจารย์ที่ปรึกษา

ดร.พิสิฐ บุญศรีเมือง

ผศ. สุรพล บุญจันทร์

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2551

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน ZIGBEE

ELECTRICAL EQUIPMENT CONTROL SYSTEM BY USING ZIGBEE

ผู้จัดทำ

1. นางสาวกฤตภัค เกตุเรือง 48010021
2. นายกษิดิ์เดช ชื่นชูผล 48010033
3. นางสาวกาญจนา ลิ้มวัฒนาชัย 48010041

อาจารย์ที่ปรึกษา

(ดร.พิสิฐ บุญศรีเมือง)

อาจารย์ที่ปรึกษา

(ผศ. สุรพล บุญจันทร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน ZIGBEE

ELECTRICAL EQUIPMENT CONTROL SYSTEM BY USING ZIGBEE

โดย นางสาวกฤตภัค เกตุเรือง 48010021

นายกษิดิ์เดช ชื่นชูผล 48010033

นางสาวกาญจนา ลิ้มวัฒนาชัย 48010041

อาจารย์ที่ปรึกษา ดร.พิสิฐ บุญศรีเมือง

ผศ. สุรพล บุญจันทร์

บทคัดย่อ

โครงงานนี้นำเสนอระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน Zigbee ประกอบด้วย 2 ส่วนคือ ส่วนที่1 คอมพิวเตอร์ที่เชื่อมต่อกับโมดูลZigbee ส่วนที่ 2 ชุดไมโครคอนโทรลเลอร์ที่เชื่อมต่อโมดูล Zigbeeและอุปกรณ์ไฟฟ้า หลักการทำงาน คือ คำสั่งในการควบคุมอุปกรณ์ไฟฟ้าจะถูกส่งจากคอมพิวเตอร์ ไปยังโมดูล Zigbee ทางฝั่งส่งที่ถูกเชื่อมต่ออยู่ โมดูลZigbee ของชุดไมโครคอนโทรลเลอร์ทางฝั่งรับ จะรับคำสั่งที่ส่งมาและนำไปควบคุมอุปกรณ์ไฟฟ้า

ABSTRACT

This project presents electrical equipment control by Zigbee that includes 2 sections. The first section consists of computer with Zigbee. And second section consists of microcontroller with Zibee module. This system controls the electrical equipment via computer by using Zigbee. The command from computer will send to microcontroller with Zigbee module. The microcontroller uses the received command to control the electrical equipments.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1 บทนำ

1.1 ความสำคัญและที่มา 1

1.2 จุดประสงค์ 1

1.3 ขอบเขตของโครงงาน 2

1.4 ผลที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีและหลักการ

2.1 Zigbee 3

2.1.1 ชนิดอุปกรณ์ของ ZigBee 3

2.1.2 การประยุกต์ใช้งาน ZigBee 4

2.1.3 โครงสร้างของโพรโทคอล ZigBee 5

2.1.4 ขั้นตอนการทำงานของโพรโทคอล ZigBee 5

2.2 โมดูล XBee Serial 2 OEM RF 7

2.2.1 ลักษณะทางกายภาพ 8

2.2.2 การติดตั้งใช้งาน 9

2.2.3 ขาสัญญาณ 9

2.2.4 คุณสมบัติทางไฟฟ้า 10

2.2.5 การสื่อสารผ่านทางพอร์ตอนุกรม 11

2.2.6 ข้อมูลแบบอนุกรม 11

2.2.7 บัฟเฟอร์ของพอร์ตอนุกรม 12

2.2.8 ZigBee Network Communication 13

2.2.9 ZigBee Device Addressing 13

2.1.10 การกำหนดตำแหน่งที่อยู่ที่ทำงานร่วมกัน (Coordinator Addressing) 13

2.1.11 การกำหนดตำแหน่งที่อยู่ที่แบบบอร์ดคาสต์ (Broadcast Addressing) 13

2.3 AT89C5131 14

2.3.1 พอร์ตรับ/ส่งข้อมูลแบบอนุกรม (Serial I/O Port) 14

2.3.2 Framing error detection 14

2.4 Delphi 22

2.4.1 Comport Library คอมโพเนนต์จัดการสื่อสารข้อมูลอนุกรม 22

2.5 สเต็ปเปอร์มอเตอร์ (STEPPER MOTOR) 25

2.5.1 ชนิดของสเต็ปเปอร์มอเตอร์ 26

2.5.2 เฟสของสเต็ปเปอร์มอเตอร์ (Stepper Motor Phase) 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.5.3 การพันลวดบนสเตเตอร์ของสเต็ปเปอร์มอเตอร์ 28

2.5.4 ชนิดของสเต็ปเปอร์มอเตอร์แบ่งตามลักษณะสายที่ใช้ต่อกับวงจรขับ 28

2.5.5 วิธีการขับ (Driving) หรือวิธีการกระตุ้นเฟส (Phase Excitation) ของสเต็ปเปอร์มอเตอร์ 30

2.6 L298 32

2.7 PC 817 และ PC 847 33

บทที่ 3 การออกแบบ

3.1 การออกแบบระบบ 35

3.1.1 ฝั่งคอมพิวเตอร์ 35

3.1.2 ฝั่งไมโครคอนโทรลเลอร์ 35

3.1.3 การเชื่อมต่อระหว่างฝั่งคอมพิวเตอร์และฝั่งไมโครคอนโทรลเลอร์โดยผ่านโมดูล Zigbee 35

3.1.4 แสดงบล็อกไดอะแกรมการทำงานของระบบ 35

3.2 การออกแบบคำสั่งควบคุมของอุปกรณ์ไฟฟ้า 36

3.3 การออกแบบการแสดงสถานะของอุปกรณ์ไฟฟ้า 36

3.4 การออกแบบการทำงานของโปรแกรมควบคุมอุปกรณ์ไฟฟ้า 36

3.4.1 การออกแบบโปรแกรมฝั่งคอมพิวเตอร์ 36

3.4.2 การออกแบบโปรแกรมฝั่งไมโครคอนโทรลเลอร์ 41

3.5 การออกแบบวงจรด้านรับและด้านส่งของโมดูล Zigbee 45

3.6 วงจรโฟโต้คอปเปอร์ LTV 847 และ PC 817 46

3.7 วงจรขับมอเตอร์โดยใช้ไอซี L298 46

3.8 วงจรตรวจจับกระแส 48

3.9 วงจรควบคุมการทำงานของหลอดไฟ 48

บทที่ 4 ผลการทดลองและวิเคราะห์ผลการทดลอง

4.1 การทดลองการส่งข้อมูลสถานะของอุปกรณ์จากฝั่งไมโครคอนโทรลเลอร์ไปยังฝั่งคอมพิวเตอร์ 50

4.1.1 ฝั่งคอมพิวเตอร์ 50

4.1.2 ฝั่งไมโครคอนโทรลเลอร์ 51

4.2 การทดลองวัดการป้อนสัญญาณไบนารี่ 4 บิตเพื่อขับมอเตอร์ 53

4.3 การทดลองวัดสัญญาณเอาท์พุทของ L298 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

4.4 การทดลองวัดสัญญาณเอาท์พุทของวงจรแปลงไฟจากกระแสสลับ 9 โวลต์ เป็นกระแสตรง ± 5 โวลต์ 56

4.5 การทดลองวัดสัญญาณเอาท์พุทของวงจรตรวจจับกระแสโดยใช้ไอซี ACS712 57

4.6 การทดลองวัดสัญญาณเอาท์พุทของวงจรขยายโดยใช้ออปแอมป์ 58

4.7 การทดลองติดต่อกับโมดูล Zigbee ด้วย AT Command ผ่านโปรแกรม X-CTU 58

4.8 การทดลองการส่ง-รับข้อมูลของโมดูล Zigbee 59

4.9 การทดลองเปลี่ยนตำแหน่งที่อยู่ของ Zigbee 60

บทที่ 5 สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง 61

5.2 วิจารณ์การทดลอง 61

5.3 แนวทางการพัฒนา 61

บรรณานุกรม 62

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 เปรียบเทียบมาตรฐาน IEEE 802.15.4 แต่ละความถี่สำหรับการติดต่อสื่อสาร	4
รูปที่ 2.2 Channel Division	4
รูปที่ 2.3 รูปแบบเครือข่าย IEEE 802.15.4	5
รูปที่ 2.4 IEEE 802.15.4 Stack	5
รูปที่ 2.5 802.15.4/ZigBee, BluetoothและWireless	7
รูปที่ 2.6 ลักษณะทางกายภาพของโมดูล XBee Serial 2 OEM RF	8
รูปที่ 2.7 แสดงการติดตั้งการใช้งานของโมดูล XBee Serial 2 OEM RF	9
รูปที่ 2.8 แสดงขาสัญญาณของโมดูล XBee Serial 2 OEM RF	9
รูปที่ 2.9 แสดงไดอะแกรมการส่ง-รับข้อมูลของการอินเตอร์เฟส UART	11
รูปที่ 2.10 แสดงแพ็คเก็ตข้อมูล UART (เลขฐานสิบ"31") ที่ส่งผ่านโมดูล RF	11
รูปที่ 2.11 แสดงไดอะแกรมการส่ง-รับข้อมูลภายใน	12
รูปที่ 2.12 บล็อกไดอะแกรมFraming Error	14
รูปที่ 2.13 การเลือกอัตราบอร์ด	15
รูปที่ 2.14 อัตราบอร์ดภายใน	16
รูปที่ 2.15 โครงสร้างสเต็ปเปอร์มอเตอร์ชนิดวาริเอเบิลรีลักแตนซ์และสัญญลักษณ์	26
รูปที่ 2.16 สเต็ปเปอร์มอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ต	27
รูปที่ 2.17 สเต็ปเปอร์มอเตอร์ชนิดไฮบริดขนาด 5 เฟส	27
รูปที่ 2.18 ลักษณะการพันขดลวดบนสเตเตอร์	28
รูปที่ 2.19 สัญลักษณ์, โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบไบโพลา 2 เฟส	28
รูปที่ 2.20 สัญลักษณ์, โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบยูนิโพลา 2 เฟส	29
รูปที่ 2.21 สัญลักษณ์ , โครงสร้างมอเตอร์2 เฟสที่พันลวดแบบไบฟิลาและแยกปลายของแต่ละขด	29
รูปที่ 2.22 สัญลักษณ์การต่อสายแบบต่างๆ	29
รูปที่ 2.23 ตัวอย่างการหมุนและการกระตุ้นมอเตอร์แบบไบโพลา 2 เฟส	32
รูปที่ 2.24 แสดงลักษณะของขาL298	33
รูปที่ 2.25 แสดงลักษณะของขา PC817 และ PC847	34
รูปที่ 3.1 แสดงการทำงานของระบบ	35
รูปที่ 3.2 แผนผังการทำงานของโปรแกรมฝั่งคอมพิวเตอร์	37
รูปที่ 3.3 แผนผังการทำงานของโปรแกรมฝั่งคอมพิวเตอร์ (ต่อ)	38
รูปที่ 3.4 แผนผังการทำงานของโปรแกรมฝั่งคอมพิวเตอร์ (ต่อ)	39
รูปที่ 3.5 แผนผังการทำงานของโปรแกรมฝั่งคอมพิวเตอร์ (ต่อ)	40
รูปที่ 3.6 แผนผังการทำงานของส่วนควบคุมอุปกรณ์ไฟฟ้า	41
รูปที่ 3.7 แผนผังการทำงานของส่วนควบคุมอุปกรณ์ไฟฟ้า (ต่อ)	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 3.8 แผนผังการทำงานของส่วนควบคุมอุปกรณ์ไฟฟ้า (ต่อ) 43

รูปที่ 3.9 แผนผังการทำงานของส่วนควบคุมอุปกรณ์ไฟฟ้า (ต่อ) 44

รูปที่ 3.10 แสดงวงจรทำงานของโมดูล Zigbee 45

รูปที่ 3.11 แสดงวงจรแปลงระดับแรงดันสัญญาณอินพุทเป็น 3.3 โวลต์ 45

รูปที่ 3.12 แสดงวงจรโฟโต้คอปเปอร์ LTV 847 และ PC 817 46

รูปที่ 3.13 แสดงวงจรขับมอเตอร์โดยใช้ไอซี L298 47

รูปที่ 3.14 แสดงวงจรตรวจจับกระแส 48

รูปที่ 3.15 แสดงวงจรวงจรควบคุมการทำงานของหลอดไฟ 49

รูปที่ 4.1 แสดงอุปกรณ์ของฝั่งไมโครคอนโทรลเลอร์ 50

รูปที่ 4.2 แสดงอุปกรณ์ของฝั่งคอมพิวเตอร์ 50

รูปที่ 4.3 แสดงสถานะองศาของมอเตอร์เมื่อไมโครคอนโทรลเลอร์ส่งค่าอักขระ “1” 52

รูปที่ 4.4 แสดงสถานะองศาของมอเตอร์เมื่อไมโครคอนโทรลเลอร์ส่งค่าอักขระ “,” 52

รูปที่ 4.5 แสดงสถานะการทำงานของหลอดไฟเมื่อไมโครคอนโทรลเลอร์ส่งค่าอักขระ “b” และ “a” 53

รูปที่ 4.6 แสดงสัญญาณ M1 ที่ขาอินพุทและเอาท์พุทของโฟโต้คอปเปอร์ LTV 847 และ PC 817

(Chanel 1) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 2

(Chanel 2) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 16 54

รูปที่ 4.7 แสดงสัญญาณ M1 ที่ขาอินพุทและเอาท์พุทของโฟโต้คอปเปอร์ LTV 847 และ PC 817

(Chanel 1) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 4

(Chanel 2) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 14 54

รูปที่ 4.8 แสดงสัญญาณ M1 ที่ขาอินพุทและเอาท์พุทของโฟโต้คอปเปอร์ LTV 847 และ PC 817

(Chanel 1) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 6

(Chanel 2) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 12 55

รูปที่ 4.9 แสดงสัญญาณ M1 ที่ขาอินพุทและเอาท์พุทของโฟโต้คอปเปอร์ LTV 847 และ PC 817

(Chanel 1) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 8

(Chanel 2) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 10 55

รูปที่ 4.10 แสดงสัญญาณ M1 และ M2 ที่ขาเอาท์พุทของ L298

(Chanel 1) แสดงลักษณะของสัญญาณ M1

(Chanel 2) แสดงลักษณะของสัญญาณ M2 56

รูปที่ 4.11 แสดงสัญญาณ M3 และ M4 ที่ขาเอาท์พุทของ L298

(Chanel 1) แสดงลักษณะของสัญญาณ M3

(Chanel 2) แสดงลักษณะของสัญญาณ M4 56

รูปที่ 4.12 แสดงสัญญาณอินพุทกระแสสลับ 9 โวลต์ 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 4.13 แสดงสัญญาณเอาท์พุทกระแสตรง ± 5 โวลต์
(Chanel 1) แสดงลักษณะของสัญญาณ +5 โวลต์
(Chanel 2) แสดงลักษณะของสัญญาณ -5 โวลต์ 57

รูปที่ 4.14 แสดงสัญญาณเอาท์พุทของวงจรตรวจจับกระแสโดยใช้ไอซี ACS712 58

รูปที่ 4.15 แสดงสัญญาณเอาท์พุทของวงจรขยายโดยใช้ออปแอมป์ 58

รูปที่ 4.16 แสดงการติดต่อโมดูล Zigbee โดยใช้ AT Command 59

รูปที่ 4.17 แสดงสัญญาณเปรียบเทียบกันระหว่างข้อมูลของฝั่งส่งกับข้อมูลของฝั่งรับ
(Chanel 1) แสดงลักษณะของสัญญาณที่ CH1
(Chanel 2) แสดงลักษณะของสัญญาณที่ CH2 59

รูปที่ 4.18 แสดงสัญญาณที่วัดจากขา Tx ของ Zigbee ที่ส่ง OK กลับมา 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงการเปรียบเทียบคุณสมบัติของ Zigbee, Wifi และ Bluetooth 6

ตารางที่ 2.2 แสดงคุณสมบัติของโมดูล XBee Serial 2 OEM RF 8

ตารางที่ 2.3 แสดงขาสัญญาณของโมดูล XBee Serial 2 OEM RF 9

ตารางที่ 2.4 แสดงคุณสมบัติทางไฟฟ้าของโมดูล XBee Serial 2 OEM RF 10

ตารางที่ 2.5 การเลือกอัตราบอร์ดสำหรับUART 15

ตารางที่ 2.6 แสดงรีจิสเตอร์ที่ใช้การสื่อสารแบบอนุกรม 16

ตารางที่ 2.7 ตัวอย่างการคำนวณเมื่อ X2 = 1, SMOD1 = 1, SPD = 1 17

ตารางที่ 2.8 ตัวอย่างการคำนวณเมื่อ X2 = 0, SMOD1 = 1, SPD = 0 18

ตารางที่ 2.9 ตารางรีจีสเตอร์T2CON 19

ตารางที่ 2.10 รีจีสเตอร์PCON 20

ตารางที่ 2.11 ตาราง BDRCON Register 21

ตารางที่ 3.1 คำสั่งของอุปกรณ์ไฟฟ้าที่ส่งจากโปรแกรมฝั่งคอมพิวเตอร์ไปยังไมโครคอนโทรลเลอร์ 36

ตารางที่ 3.2 การป้อนสัญญาณไบนารี่เพื่อขับมอเตอร์ด้วยไอซี L298 แบบฟอร์เวิร์ส 47

ตารางที่ 3.3 การป้อนสัญญาณไบนารี่เพื่อขับมอเตอร์ด้วยไอซี L298 แบบรีเวิร์ส 47

ตารางที่ 4.1 คำสั่งที่ใช้ในการตรวจสอบเงื่อนไขการทำงานและการทำงานของมอเตอร์ 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทสำคัญอย่างมากในชีวิตประจำวัน และการที่เทคโนโลยีมีการพัฒนาอย่างต่อเนื่องจึงทำให้อุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการติดต่อสื่อสารมีประสิทธิภาพมากขึ้น อีกทั้งยังส่งผลให้มีการพัฒนาการส่งข้อมูล เดิมทีการส่งข้อมูลระหว่าง 2 แห่งจะต้องใช้การเชื่อมต่อผ่านสายแต่ด้วยการพัฒนาทำให้การส่งข้อมูลระหว่างกันไม่ต้องใช้สายแล้ว ส่งผลให้ผู้ใช้งานได้รับความสะดวกสบายมากขึ้น ด้วยเหตุนี้ผู้ผลิตบางรายจึงนำเทคโนโลยีการสื่อสารไร้สายมาประยุกต์ใช้ในการควบคุมอุปกรณ์และแนวโน้มในอนาคตอาจจะมีการผลิตและใช้งานควบคุมอุปกรณ์แบบไร้สายเพิ่มมากขึ้น โดย Zigbee ถือเป็นโมดูลไร้สายประเภทหนึ่งที่มีประสิทธิภาพสูง ราคาถูก สามารถใช้งานได้ยาวนาน จึงมีแนวคิดในการนำ Zigbee มาใช้งานในการควบคุมอุปกรณ์แบบไร้สาย

ปัจจุบันคอมพิวเตอร์เป็นสิ่งจำเป็นและมีบทบาทสำคัญมาก ดังจะเห็นได้จากยอดขายคอมพิวเตอร์และโน้ตบุ๊คที่เพิ่มสูงขึ้นทุกปีและแทบทุกครัวเรือนจะต้องมีคอมพิวเตอร์ไว้ใช้งาน จึงมีแนวคิดในการจัดทำโครงงานนี้ขึ้น โดยใช้โปรแกรมคอมพิวเตอร์ควบคุมอุปกรณ์ไฟฟ้าแบบไร้สาย

โครงงานนี้เป็นการควบคุมอุปกรณ์ไฟฟ้าโดยควบคุมจากคอมพิวเตอร์ซึ่งจะส่งข้อมูลคำสั่งผ่านทางโมดูล Zigbee ไปยังชุดไมโครคอนโทรลเลอร์เพื่อควบคุมอุปกรณ์ รวมทั้งไมโครคอนโทรลเลอร์จะส่งสถานะของอุปกรณ์กลับมาแสดงบนคอมพิวเตอร์ ซึ่งระบบการทำงานแบ่งเป็น 3 ส่วนดังนี้

ส่วนที่ 1 โปรแกรมควบคุมอุปกรณ์ไฟฟ้า

เป็นส่วนอินเตอร์เฟสระหว่างผู้ใช้งานและส่วนควบคุมอุปกรณ์ไฟฟ้า โปรแกรมจะแสดงวันที่, เวลาและสถานะอุปกรณ์ไฟฟ้าที่ควบคุม สามารถควบคุมอุปกรณ์ไฟฟ้า ได้โดยผ่านหน้าต่างนี้ ซึ่งทำการส่งและรับข้อมูลผ่านโมดูล Zigbee แล้วแสดงสถานะของอุปกรณ์ไฟฟ้า

ส่วนที่ 2 ส่วนควบคุมอุปกรณ์ไฟฟ้า

ส่วนนี้จะเป็นการควบคุมอุปกรณ์ไฟฟ้าด้วยการใช้ไมโครคอนโทรเลอร์ ไมโครคอนโทรลเลอร์จะรับคำสั่งจากโปรแกรมควบคุมอุปกรณ์ไฟฟ้า เพื่อไปควบคุมอุปกรณ์ไฟฟ้า รวมทั้งส่งสถานะของอุปกรณ์ไฟฟ้า กลับไปยังโปรแกรมควบคุมอุปกรณ์ไฟฟ้าด้วย

ส่วนที่ 3 การเชื่อมต่อระหว่างโปรแกรมควบคุมหลอดไฟและส่วนควบคุมหลอดไฟโดยผ่านโมดูล Zigbee

โมดูล Zigbee จะส่งผ่านคำสั่งระหว่างโปรแกรมคอมพิวเตอร์และส่วนควบคุมอุปกรณ์ไฟฟ้า โดยจะมีโมดูล Zigbee 2 ตัว ตัวที่ 1 จะเชื่อมต่อกับคอมพิวเตอร์ที่มีโปรแกรมควบคุมอุปกรณ์ไฟฟ้า และตัวที่ 2 เชื่อมต่อกับไมโครคอนโทรลเลอร์ที่เป็นส่วนควบคุมอุปกรณ์ไฟฟ้า

1.2 จุดประสงค์

จุดมุ่งหมายในการทำโครงงานนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เพื่อศึกษาการใช้งาน Zigbee

2) เพื่อศึกษาการส่งผ่านข้อมูลโดยใช้ไมโครคอนโทรลเลอร์

3) สามารถควบคุมอุปกรณ์ไฟฟ้าโดยผ่านโปรแกรมควบคุมได้

1.3 ขอบเขตของโครงงาน

ระบบควบคุมอุปกรณ์ไฟฟ้าผ่านโมดูล Zigbee มีการทำงานหลักดังนี้

1) คอมพิวเตอร์เชื่อมต่อกับโมดูล Zigbee

คอมพิวเตอร์แสดงสถานะการทำงานของอุปกรณ์ไฟฟ้าและส่งคำสั่งเพื่อควบคุมอุปกรณ์ไฟฟ้าโดยผ่านโมดูล Zigbee

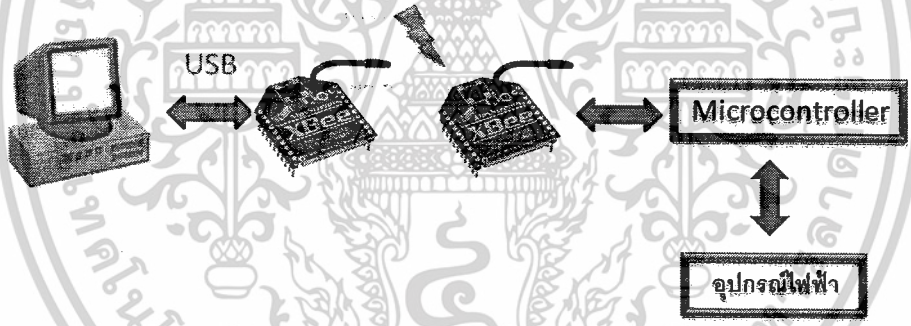
2) ชุดไมโครคอนโทรลเลอร์ที่เชื่อมต่อโมดูล Zigbee และอุปกรณ์ไฟฟ้า

ชุดไมโครคอนโทรลเลอร์รับคำสั่งจากคอมพิวเตอร์เพื่อควบคุมการทำงานของอุปกรณ์ไฟฟ้าพร้อมทั้งส่งสถานะของอุปกรณ์ไฟฟ้ากลับไปยังคอมพิวเตอร์ด้วยโดยผ่านโมดูล Zigbee

1.4 ผลที่คาดว่าจะได้รับ

1) ระบบที่พัฒนาสามารถใช้งานได้จริง

2) ทำให้เกิดความรู้ ความเข้าใจในการใช้โปรแกรม Delphi, การส่งผ่านข้อมูลควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์, การใช้งานและการเชื่อมต่อโมดูล Zigbee



รูปที่ 1.1 แสดงการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 Zigbee

ชื่อ ZigBee ได้มาจากพฤติกรรมการสื่อสารของผึ้ง โดยผึ้งจะบินแบบซิกแซ็ก และจะให้ข้อมูลข่าวสารระหว่างผึ้งด้วยกันที่เกี่ยวกับ ตำแหน่ง ระยะทาง และทิศทางของอาหารที่พวกมันกำลังหาอยู่ ZigBee ถูกสร้างขึ้นในการทำระบบเครือข่ายไร้สายส่วนบุคคล (WPAN) อยู่ภายใต้มาตรฐาน IEEE 802.15.4 โดยมาตรฐานนี้ใช้งานสำหรับการสื่อสารความเร็วต่ำ ใช้กำลังไฟฟ้าน้อย อุปกรณ์ราคาถูก และมีคุณสมบัติการจัดการตัวเองได้ เป็นเทคโนโลยีไร้สายที่ร่วมกันสื่อสารข้อมูลผ่านเซ็นเซอร์ขนาดเล็กจิ๋วจำนวนเป็นพันๆ หมื่นๆ ชิ้นที่ฝังอยู่ตามส่วนต่างๆ ในอาคาร สำนักงาน โรงงาน หรือแม้แต่ในบ้าน

การทำงานของ ZigBee จะเป็นการรับ-ส่งคลื่นสัญญาณข้อมูล ผ่านชิปเล็กจิ๋วนี้จุดต่อจุดไปเรื่อยๆ จนถึงปลายทางที่ต้องการดาวน์โหลดข้อมูลลงในเครื่องคอมพิวเตอร์เพื่อใช้ในการวิเคราะห์ข้อมูล ข้อมูลที่ได้อาจจะเป็นการวัดอุณหภูมิ การเคลื่อนไหวของสิ่งมีชีวิต จับปริมาณมลพิษในอากาศ ปริมาณน้ำ ท่อแก๊สโดยใช้พลังงานแสงอาทิตย์หรือแบตเตอรี่ขนาดเล็กที่กินไฟน้อยมาก

ลักษณะของ ZigBee คือมีทางเข้าช่องสัญญาณ โดยการใช้ Carrier Sense Multiple Access with Collision Avoidance (CSMA - CA) หรือมีทางเข้าช่องสัญญาณหลายๆ ทาง เพื่อหลีกเลี่ยงการชนกัน ระยะทางโดยทั่วไปประมาณ 50 เมตร มี topology แบบ star , peer-to-peer, mesh ทั้งนี้แต่ละอุปกรณ์จะมีแอดเดรส ที่มีความยาว 64 หรือ 16 บิต (รองรับได้64,000 อุปกรณ์)
มาตรฐาน IEEE 802.15.4 ที่ถูกกำหนดไว้ใน ชั้น Physical layer มีอยู่ 3 ความถี่คือ
1. 2.4-2.4835 GHz bit rate 250 kb/s มีอยู่ 16 ช่องสัญญาณ คือช่องสัญญาณที่ 11-26
2. 868-870 MHz bit rate 20 kb/s มีอยู่ 1 ช่องสัญญาณ คือช่องสัญญาณที่ 0
3. 902-928 MHz bit rate 40 kb/s มีอยู่ 10 ช่องสัญญาณ คือ ช่องสัญญาณที่ 1-10
โดยความถี่ 2.4-2.4835 GHz สามารถใช้งานได้ทั่วโลก และความถี่ 868-870 MHz และ 902-928 MHz ใช้งานได้ในพื้นที่ของอเมริกาเหนือ , ยุโรป , ออสเตรเลีย และนิวซีแลนด์

2.1.1 ชนิดอุปกรณ์ของ ZigBee

แบบ Physical Device มี 2 ประเภท คือ

1) Full Function Device : FFD เป็น เราเตอร์ที่เป็นสื่อกลางในการส่งข้อมูลจากอุปกรณ์อื่นๆ ใช้พลังงานจาก power line ทำงานได้ในทุก Topology และสามารถทำเป็นจุดเชื่อมต่อกันได้

2) Reduced Function Device : RFD เหมาะแก่การเชื่อมต่อภายในเครือข่าย ใช้พลังงานจากแบตเตอรี่ ไม่สามารถถ่ายทอดข้อมูลจากอุปกรณ์อื่นๆ ได้ ทำได้ง่ายในเครือข่ายที่เป็นแบบ star

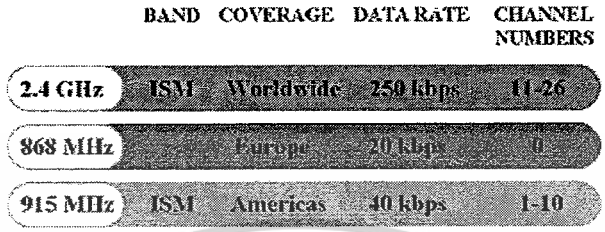
แบบ Logical Device มี 3 ประเภท คือ

1) ZigBee Coordinators เป็นจุดที่ประสานเชื่อมต่อกัน ทำหน้าที่เริ่มจัดการเชื่อมโครงข่าย จัดการโหนดในโครงข่ายและเก็บข่าวสารของโนดในโครงข่าย

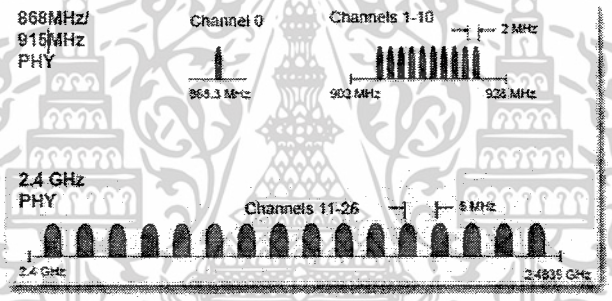
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ZigBee Routers ทำหน้าที่จัดการเส้นทางของข้อความที่ส่งผ่านภายในโครงข่ายโหนดใด ๆ

3) ZigBee End Devices เป็นโหนดที่อยู่ในส่วนของผู้ใช้งาน โดยสามารถเป็นได้ทั้งแบบ RFD และ FFD



รูปที่ 2.1 เปรียบเทียบมาตรฐาน IEEE 802.15.4 แต่ละความถี่สำหรับการติดต่อสื่อสาร



รูปที่ 2.2 Channel Division

2.1.2 การประยุกต์ใช้งาน ZigBee

แบ่งแยกตามประเภทของข้อมูลข่าวสาร มีอยู่ 3 แบบคือ

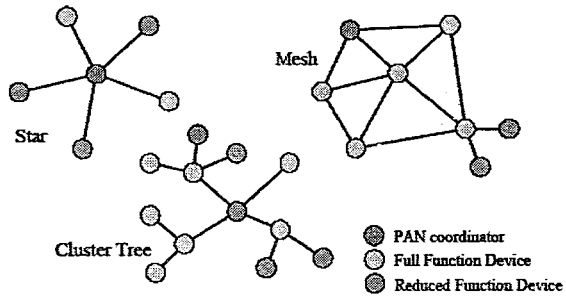
1) ข้อมูลแบบ Periodic ข้อมูลเป็นช่วงเวลา โปรแกรมสามารถควบคุมอัตราการส่ง และตัวตรวจจับสัญญาณกระตุ้น เช็คข้อมูลและทำให้ข้อมูลไม่เคลื่อนไหว ใช้สำหรับเซนเซอร์และมิเตอร์

2) ข้อมูลแบบ Intermittent เป็นลักษณะที่มีการส่งผ่านข้อมูลเมื่อมีการใช้งาน เช่น สวิตช์ไฟไร้สาย

3) ข้อมูลแบบ Repetitive low latency ใช้ในงานที่ต้องการ latency น้อย ๆ โดยการสื่อสารจะใช้วิธีจัดสรรช่องเวลา และสามารถใช้กลไกแบบ GTS เพื่อรับประกันคุณภาพของการบริการ นำไปใช้ในงาน เช่น เมาส์ไร้สาย

โพรโทคอล ZigBee ถูกออกแบบมาเฉพาะในส่วนของ Application layer, Application support layer และ Network layer เท่านั้น แต่ใช้ MAC layer และ Physical layer ตามมาตรฐาน IEEE 802.15.4 ชั้น Network นี้สามารถรองรับได้ 3 Topologies คือแบบ Star, Mesh และ Cluster-tree แสดงในรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

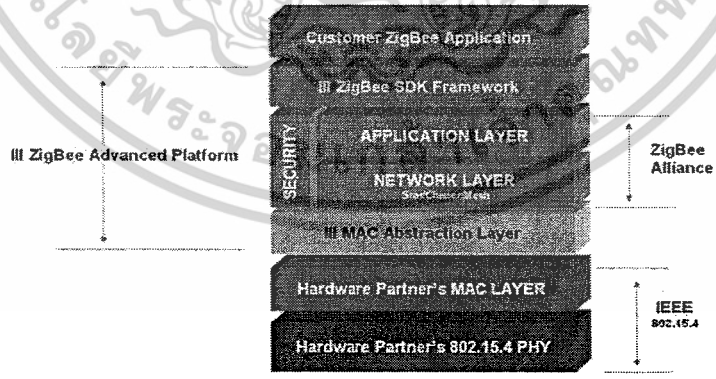


รูปที่ 2.3 รูปแบบเครือข่าย IEEE 802.15.4

ในการเชื่อมต่อแบบ Star นั้น จะใช้งานกับอุปกรณ์พื้นฐาน เช่นเซนเซอร์ ที่มีช่วงการทำงานได้นาน ส่วนการเชื่อมต่อแบบ Mesh หรือ peer-to-peer นั้น เป็นการส่งผ่านข้อมูลภายในโครงข่ายของอุปกรณ์จะส่งออกไปในหลายเส้นทาง และมีระดับของความเชื่อถือและแบ่งแยกได้สูง ส่วนแบบ Cluster-tree จะเป็นเป็นการรวมกันของแบบ Star กับ Mesh เพื่อที่จะได้ประโยชน์จากทั้งสองแบบคือมีระดับของความเชื่อถือได้และรองรับโหนดที่สามารถใช้พลังงานจากแบตเตอรี่ได้ยาวนาน

2.1.3 โครงสร้างของโพรโทคอล ZigBee

Application layer เป็นชั้นที่มีส่วนของ Endpoint อยู่ เรียกว่า Application framework โดยมี ZigBee Device Object (ZDO) ทำหน้าที่ในการจัดการในการเข้าถึงและใช้งาน Application layer , Application -support sub-layer ทำหน้าที่ในการสร้างเฟรมของ Application layer และทำหน้าที่ในการรับส่งข้อมูลรวมถึงการจัดการด้านต่างๆ ที่เกี่ยวข้องกับ Application layer , Network layer ทำหน้าที่ใช้ในการ routing ข้อมูลจากต้นทางไปยังปลายทางที่อาจอยู่ภายในเครือข่ายเดียวกันหรือต่างเครือข่ายกัน



รูปที่ 2.4 IEEE 802.15.4 Stack

2.1.4 ขั้นตอนการทำงานของโพรโทคอล ZigBee

1) ขั้นตอนการทำงานของ ZigBee coordinator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ZigBee coordinator จะเริ่มต้นเครือข่าย โดยการตรวจสอบการใช้ช่องสัญญาณวิทยุภายในบริเวณรอบๆ ถ้ามีช่องสัญญาณที่ไม่ถูกใช้โดย Coordinator ตัวอื่น ก็สามารถเริ่มต้นเครือข่ายได้ หลังจากนั้น Coordinator ก็จะทำหน้าที่เป็นศูนย์กลางของเครือข่าย รองรับการเข้าร่วมเครือข่ายของ ZigBee end-device และรองรับการร้องขออื่นๆ ตามมาตรฐานด้วยเช่นกัน

2) ขั้นตอนการทำงานของ ZigBee end-device

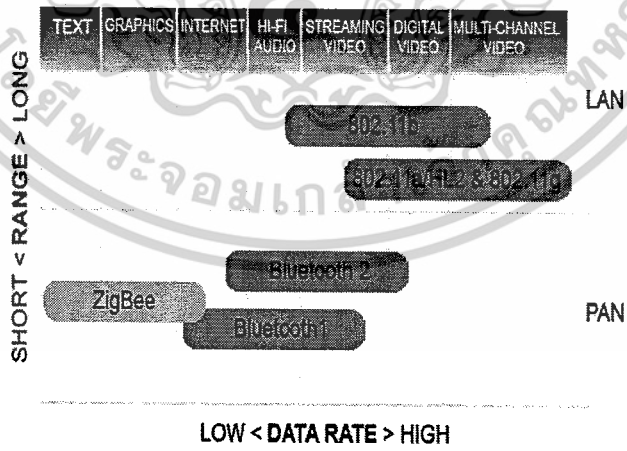
ZigBee end-device จะเริ่มต้นการทำงานโดยการร้องขอการเข้าร่วมเครือข่ายไปยัง Coordinator ประจำเครือข่ายนั้นๆ โดยการตรวจสอบผ่านช่องสัญญาณต่างๆ ว่า Coordinator ใช้ช่องสัญญาณใดอยู่เมื่อเข้าร่วมเครือข่ายแล้ว End-device จึงสามารถทำการร้องขอคำสั่งอื่นๆ ผ่านทาง Coordinator ได้ เช่น การส่งข้อความทั่วไป (Message), การร้องขอการ Binding (Binding request), การขอออกจากเครือข่าย

ตารางที่ 2.1 แสดงการเปรียบเทียบคุณสมบัติของ Zigbee, Wifi และ Bluetooth

คุณสมบัติ	Zigbee	Wifi	Bluetooth
มาตรฐาน	802.15.4	802.11b	802.15.1
ย่านความถี่ที่ใช้งาน	2.4GHz	2.4GHz	2.4GHz
จำนวนช่องสัญญาณ	ประมาณ16 ช่องสัญญาณ		แถบยุโรปและอเมริกา แบ่งออกเป็น 79 ช่องสัญญาณ ญี่ปุ่นแบ่งออกเป็น23 ช่องสัญญาณ
ระยะทำการ	ภายในอาคาร 40 m ภายนอกอาคาร 120 m	50-100 เมตร	5-10 เมตร
อัตราการส่งถ่ายข้อมูล	1200 - 115200 bps	11 Mb/s	1Mbps
การใช้พลังงาน	ต่ำ คือ 0.1 มิลลิวัตต์	ปานกลาง	ต่ำ คือ 0.1วัตต์
การใช้งาน	ส่งข้อมูลในจำนวนที่ไม่มากมักใช้กับlighting, sensors, RC peripherals	ติดต่อกันโดยการเชื่อมโยง สามารถขนส่งข้อมูลได้ในจำนวนมากมักใช้กับบริษัทและhome access points	ใช้กับอุปกรณ์ที่มีขนาดเล็ก เนื่องจากใช้การขนส่งข้อมูลในจำนวนที่ไม่มากเช่น Telecom audioใช้แทนสายcable
ระบบป้องกัน	128 บิต AES และ Application layer	Authentication Service Set ID (SSID)	ใช้การป้อนรหัสก่อนการเชื่อมต่อ และ ป้อ[illegible]ัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ผู้ใช้งานกำหนด		การดักสัญญาณระหว่างสื่อสารโดยจะสลับช่องสัญญาณไปมา64บิต, 128บิต
ขนาด	เล็ก	ปานกลาง	เล็ก
ราคา	ถูก	ปานกลาง	สูง
ความเสี่ยงต่อการถูกรบกวน	สูง	สูง	สูง
Complexity	ง่าย	ยากมาก	ยาก
Power Profile	หลายปี	หลายชั่วโมง	หลายวัน
Nodes/Master	64000	32	7
Latency	30 มิลลิวินาที	Enumerationสูงสุด 3 วินาที	Enumeration สูงสุด10 วินาที
Extend ability	ได้	อาจจะได้	ไม่ได้
TX	35mA	400+mA	40mA
Standby	3 ไมโครแอมป์	20 มิลลิแอมป์	200 ไมโครแอมป์
หน่วยความจำ	32-60 กิโลไบต์	100+ กิโลไบต์	100+ กิโลไบต์
Topology	Mesh networking Star networking Cluster tree networking	Point to multi-point	Point to multi-point



รูปที่ 2.5 802.15.4/ZigBee, BluetoothและWireless

2.2 โมดูล XBee Serial 2 OEM RF

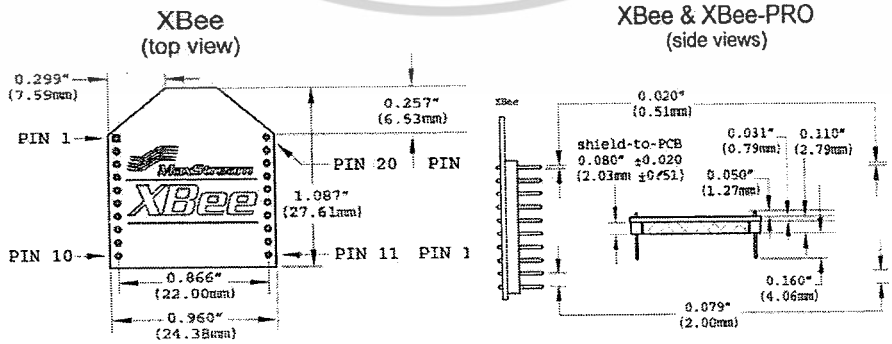
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูล XBee Serial 2 OEM RF ถูกออกแบบใช้งานภายในโปรโตรคอล Zigbee รองรับการใช้เป็นเซ็นเซอร์ไร้สายกำลังไฟต่ำ ราคาถูก โมเดลนี้ต้องการกำลังไฟต่ำและใช้รับส่งข้อมูลระหว่างอุปกรณ์ทางไกล

ตารางที่ 2.2 แสดงคุณสมบัติของโมดูล XBee Serial 2 OEM RF

คุณสมบัติ	XBee Series 2
คุณสมบัติทั่วไป	
ระยะภายในอาคาร	40 เมตร
ระยะภายนอกอาคาร	120 เมตร
กำลังการส่ง	2 mw
อัตราการส่งข้อมูล	250,000 bps
อัตราการส่งข้อมูลผ่านพอร์ตอนุกรม	1200-230,400 bps
ความต้องการกำลังไฟฟ้า	
ไฟเลี้ยง	2.8-3.4 โวลต์
กระแส Tx	40 mA(@3.3v)
กระแส Rx	40 mA(@3.3v)
ช่วงความถี่การใช้งาน	2.4 GHz
ขนาด	2.438 cm × 2.761 cm
อุณหภูมิการทำงาน	-40 ถึง 85°
เสาอากาศ	Integrated Whip,chip,RPSMA
เน็ตเวิร์กและความปลอดภัย	
รองรับโครงสร้าง	Point-to-point,point-to-multipoint,peer-to-peer
จำนวนช่องสัญญาณ	16 ช่อง

2.2.1 ลักษณะทางกายภาพ

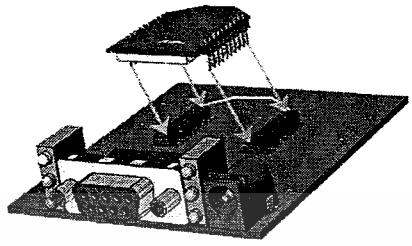


รูปที่ 2.6 ลักษณะทางกายภาพของโมดูล XBee Serial 2 OEM RF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

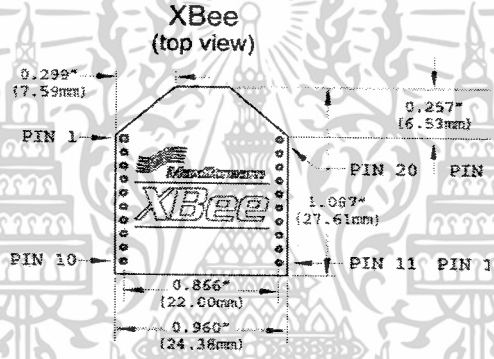
2.2.2 การติดตั้งใช้งาน

XBee Series 2 RF เมื่อใช้งานจะติดตั้งบนฐาน (socket) และไม่ต้องการการบัดกรีเมื่อใช้งานบนบอร์ด XBee รองรับการอินเตอร์เฟส RS232 และ USB ฐานติดตั้งมี 20 ขา



รูปที่ 2.7 แสดงการติดตั้งการใช้งานของโมดูล XBee Serial 2 OEM RF

2.2.3 ขาสัญญาณ



รูปที่ 2.8 แสดงขาสัญญาณของโมดูล XBee Serial 2 OEM RF

ตารางที่ 2.3 แสดงขาสัญญาณของโมดูล XBee Serial 2 OEM RF

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN/CONFIG	Input	UART Data In
4	DIO8	Either	Digital I/O 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns
6	PWM0/RSSI/DIO10		Output PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM/DIO11	Either	Digital I/O 11
8	[reserved]	-	Do not connect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9	DTR/SLEEP RQ/DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	DIO4	Either	Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	[reserved]	-	Do not connect
15	Associate / DIO5	Either	Associated Indicator, Digital I/O 5
16	RTS / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1 20
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

2.2.4 คุณสมบัติทางไฟฟ้า

ตารางที่ 2.4 แสดงคุณสมบัติทางไฟฟ้าของโมดูล XBee Serial 2 OEM RF

Symbol	Parameter	Condition	Min	Typical	Max	Units
VIL	Input Low Voltage	All Digital Inputs	-	-	0.2 * VCC	V
VIH	Input High Voltage	All Digital Inputs	0.8 * VCC	-	0.18* VCC	V
VOL	Output Low Voltage	IOL = 2 mA, VCC >= 2.7 V	-	-	0.18*VCC	V
VOH	Output High Voltage	IOH = -2 mA, VCC >= 2.7 V	-	-	0.82*VCC	V
IIIN	Input Leakage Current	VIN = VCC or GND, all inputs, per pin	-	-	0.5uA	uA
TX	Transmit Current	VCC = 3.3 V	-	45	-	mA
RX	Receive Current	VCC = 3.3 V	-	50	-	mA
PWR-DWN	Power-down Current	SM parameter = 1	-	< 10	-	uA

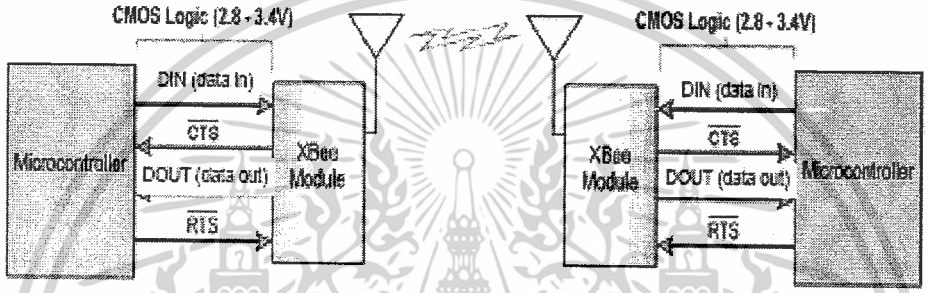
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 การสื่อสารผ่านทางพอร์ตอนุกรม

XBee series2 OEM RF โมดูลทำการอินเตอร์เฟสกับไมโครคอนโทรลเลอร์ทางพอร์ตอนุกรมระดับลอจิกแบบอะซิงโครนัส (Logic-Level Asynchronous) การผ่านทางพอร์ตอนุกรมของตัวมันนี้ ทำให้โมดูลสามารถติดต่อกับ UART ที่มีลอจิกและแรงดันที่ตรงกันได้ หรือ ผ่านทางตัวแปลงระดับแรงดันไปยังอุปกรณ์แบบอนุกรมอื่นๆ เช่น MAX stream, RS 232 หรือ USB อินเตอร์เฟสบอร์ด

UART Data flow

อุปกรณ์ที่มี UART อินเตอร์เฟสสามารถเชื่อมต่อโดยตรงกับขาของ RF โมดูล ดังแสดงในรูปข้างล่าง

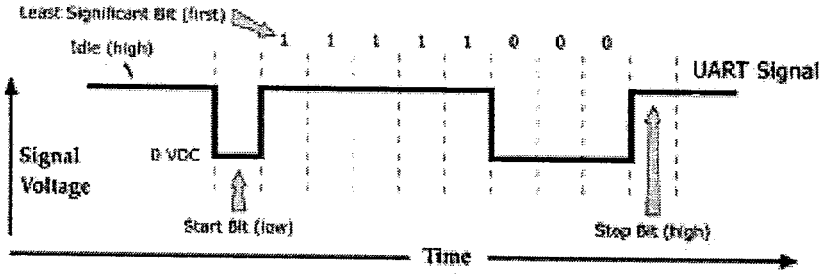


รูปที่ 2.9 แสดงไดอะแกรมการส่ง-รับข้อมูลของการอินเตอร์เฟส UART

2.2.6 ข้อมูลแบบอนุกรม

ข้อมูลจะเข้าสู่โมดูล UART ผ่านทาง DIN (ขา 3) เป็นสัญญาณอนุกรมแบบอะซิงโครนัส สัญญาณควรจะมีสถานะลอจิก 1 (high) เมื่อไม่มีข้อมูลส่งเข้ามา แต่ละไบท์ของข้อมูลประกอบด้วยบิตเริ่มต้นลอจิก 0 (low), ข้อมูลแบบอนุกรม 8 บิต และบิตหยุดสถานะลอจิก 1 (high)

โมดูล UART มีฟังก์ชันการทำงานเช่น ตรวจสอบระดับพาริตี้ (parity) และช่วงเวลา (timing) ซึ่งงานเหล่านี้ล้วนมีความสำคัญสำหรับการรับ-ส่งข้อมูล การรับ-ส่งข้อมูลเหล่านี้ UART ทั้ง 2 อันที่ถูกนำใช้ต้องมีการตั้งค่าที่ตรงกัน เช่น บอดเรต (baud rate), บิตพาริตี้ (parity), บิตเริ่มต้น (start bits), บิตหยุด (stop bits) และบิตข้อมูล (data bits)



รูปที่ 2.10 แสดงแพ็คเก็ตข้อมูล UART (เลขฐานสิบ"31") ที่ส่งผ่านโมดูล RF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 บัฟเฟอร์ของพอร์ตอนุกรม

XBee series2 โมดูลจะเตรียมบัฟเฟอร์ขนาดเล็กๆไว้มากมายเพื่อเก็บข้อมูลแบบอนุกรมและข้อมูล RF บัฟเฟอร์รับข้อมูลแบบอนุกรม (Serial receive buffer) จะเก็บข้อมูลตัวอักษรที่เข้ามาและเก็บไว้ จนกระทั่งตัวอักษรเหล่านี้สามารถนำไปใช้ได้ บัฟเฟอร์ส่งข้อมูลของแบบอนุกรม (serial transmit buffer) จะสะสมข้อมูลที่ได้รับผ่านทางลิงค์ RF ที่ต้องการส่งผ่าน UART ออกไป

บัฟเฟอร์รับข้อมูลแบบอนุกรม (serial receive buffer)

เมื่อข้อมูลแบบอนุกรม เข้าไปยัง RFโมดูล ผ่านทาง Din (ขา3) ข้อมูลจะถูกบันทึกไว้ในบัฟเฟอร์รับข้อมูลแบบอนุกรม จนกระทั่ง ข้อมูลนำไปทำงานต่อไปได้

Hardware Flow Control (CTS) เมื่อบัฟเฟอร์รับข้อมูลแบบอนุกรม (Serial receive buffer) ได้ 17 ไบท์ หากเกิดความผิดพลาด โมดูลจะเซ็ตค่า CTS (High) เพื่อส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ให้หยุดการส่งข้อมูล CTS จะเซ็ตค่าอีกครั้ง หลังจากบัฟเฟอร์ข้อมูลแบบอนุกรมมีหน่วยความจำ 34 ไบท์

บัฟเฟอร์ส่งข้อมูลแบบอนุกรม (Serial transmit buffer)

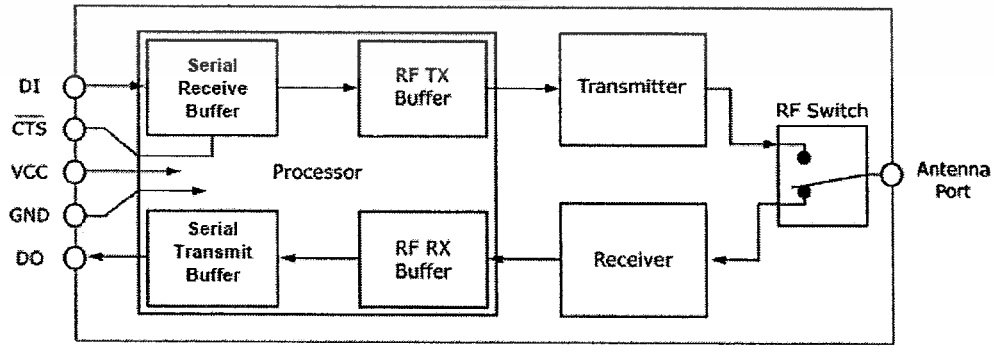
เมื่อได้รับข้อมูล RF ข้อมูลจะย้ายไปที่บัฟเฟอร์ส่งข้อมูลแบบอนุกรม (Serial transmit buffer) และส่งออกไปผ่านพอร์ตอนุกรม ถ้าบัฟเฟอร์ส่งข้อมูลแบบอนุกรมเต็ม ข้อมูลทั้งหมดในแพ็คเก็ตรับ RF (Receive RF packet) ก็จะไม่เข้ากับบัฟเฟอร์ส่งข้อมูลแบบอนุกรม แพ็คเก็ตข้อมูล RF ทั้งหมดจะถูกตัดทิ้ง

Hardware Flow Control (RTS) ถ้า RTSได้อนุญาตให้ส่งผ่านข้อมูลได้ (D6 (DI06 กำหนดค่าเป็น 1)) ข้อมูลจะไม่ส่งออกไปที่บัฟเฟอร์สำหรับส่งข้อมูลแบบอนุกรมนานเท่ากับ RTS ไม่ถูกใช้

กรณีที่บัฟเฟอร์สำหรับส่งข้อมูลแบบอนุกรมเต็มและจะมีการตัดแพ็คเก็ตทิ้ง

1) ถ้าอัตราข้อมูล RF มีค่าสูงกว่าอัตราอินเตอร์เฟสข้อมูลของโมดูล โมดูลจะสามารถรับข้อมูลเร็วกว่าที่จะส่งออกไปยังไมโครคอนโทรลเลอร์

2) ถ้าไมโครคอนโทรลเลอร์ไม่อนุญาตให้โมดูลส่งข้อมูลออกจากบัฟเฟอร์สำหรับส่งข้อมูลของพอร์ตอนุกรมเพราะว่าถูกพักโดย hardware flow control



รูปที่ 2.11 แสดงไดอะแกรมการส่ง-รับข้อมูลภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.8 ZigBee Network Communication

Zigbee รองรับการกำหนดตำแหน่งที่อยู่และ Application layer addressing อุปกรณ์จะมีการกำหนดตำแหน่งที่อยู่เฉพาะเพื่อใช้แยกปลายทางที่จะส่งข้อมูล Application layer addressing จะแสดง Application ที่รับไว้เรียกว่า End point ของ Zigbee แสดงโดยข้อความเรียกว่า กลุ่ม ID

2.2.9 ZigBee Device Addressing

โปรโตรคอล 802.15.4 ใช้เป็นโปรโตรคอลของ ZigBee โดยมีการกำหนดเป็น 2 ประเภทคือ

- ตำแหน่งที่อยู่เน็ตเวิร์ก 16 บิต
- ตำแหน่งที่อยู่ 64 บิต

<u>ตำแหน่งที่อยู่เน็ตเวิร์ก 16 บิต</u>

- ตำแหน่งที่อยู่เน็ตเวิร์ก 16 บิตถูกกำหนดเป็นโหนดเมื่อมีการใช้ในเน็ตเวิร์ก ตำแหน่งที่อยู่เน็ตเวิร์กจะถูกกำหนดเฉพาะโหนดในเน็ตเวิร์กนั้น อย่างไรก็ตามตำแหน่งที่อยู่เน็ตเวิร์กไม่คงที่สามารถเปลี่ยนแปลงได้

เงื่อนไขที่แสดงถึงสาเหตุที่ต้องมีการเปลี่ยนตำแหน่งที่อยู่เน็ตเวิร์ก

1) เมื่ออุปกรณ์สุดท้ายไม่สามารถติดต่อกับอุปกรณ์ที่เกี่ยวข้อง อุปกรณ์นี้อาจต้องออกจากเน็ตเวิร์กและต้องเข้าร่วมเน็ตเวิร์กใหม่

2) ถ้าชนิดของอุปกรณ์เปลี่ยนจากเราท์เตอร์ไปอุปกรณ์สุดท้าย หรือในทำนองเดียวกัน อุปกรณ์นี้อาจต้องออกจากเน็ตเวิร์กและต้องเข้าร่วมกับอุปกรณ์ใหม่

ZigBee ต้องการข้อมูลเพื่อส่งตำแหน่งที่อยู่เน็ตเวิร์ก 16 บิตของอุปกรณ์ปลายทาง ZigBee ต้องการตำแหน่ง 16 บิต ก่อนการส่งข้อมูล

<u>ตำแหน่งที่อยู่ 64 บิต</u>

แต่ละโหนดมีตำแหน่งที่อยู่ 64 บิต ตำแหน่งที่อยู่ 64 บิตจะมีเฉพาะแต่ละโหนดและจะไม่เปลี่ยนแปลง

2.1.10 การกำหนดตำแหน่งที่อยู่ที่ทำงานร่วมกัน (Coordinator Addressing)

แบบการทำงานร่วมกัน จะใช้ตำแหน่งที่อยู่ 64 บิตหรือข้อความ NI หาก ZigBee ที่ทำงานร่วมกันมีตำแหน่งที่อยู่เน็ตเวิร์กเป็น 0 มันจะกำหนดค่าตำแหน่งที่อยู่ 16 บิต

การส่งค่าไปยังโมดูลที่ทำงานร่วมกันใช้ค่าตำแหน่งที่อยู่ 16 บิต

กำหนดค่าตำแหน่งปลายทางของโมดูลตามด้านล่างนี้

DL (Destination Low Address) =0

DH (Destination High Address) =0

2.1.11 การกำหนดตำแหน่งที่อยู่ที่แบบบอร์ดคาสต์ (Broadcast Addressing)

การส่งแบบบอร์ดคาสต์จะส่งค่าตำแหน่งที่อยู่ 64 บิตของ 0x0000FFFF โมดูล RF จะรับแพ็ค-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็ตที่ให้ค่าตำแหน่งที่อยู่แบบบอร์ดคาสต์ เมื่อมีการตั้งค่าการทำงานในโหมดบอร์ดคาสต์ โมดูลที่รับจะส่งค่า ACKs (Acknowledgement) ส่งแพ็คเก็ตแบบบอร์ดคาสต์ไปยังทุกโมดูล กำหนดค่าตำแหน่งที่อยู่ปลายทางของโมดูลที่ทำการส่งตามนี้

DL (Destination Low Address) =0x0000FFFFF

DH (Destination High Address) =0x00000000

2.3 AT89C5131

2.3.1 พอร์ตรับ/ส่งข้อมูลแบบอนุกรม (Serial I/O Port)

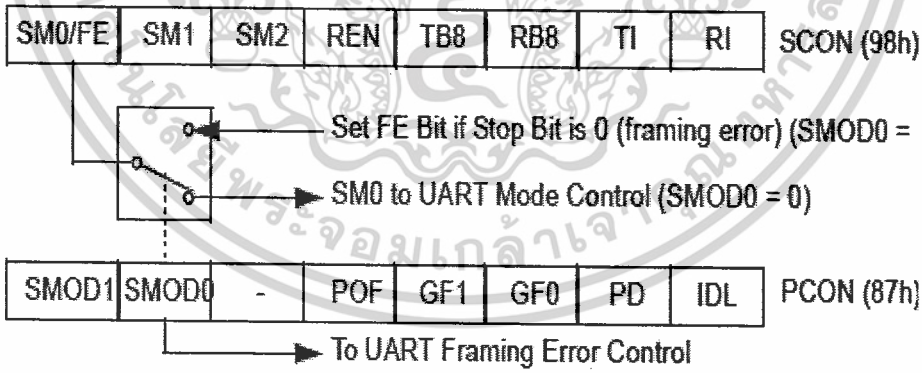
พอร์ตรับ/ส่งข้อมูลแบบอนุกรม ใน AT89C5131 ซึ่งเข้ากันได้กับ พอร์ตรับ/ส่งข้อมูลแบบอนุกรม ใน 80C52 มีทั้งโหมดซิงโครนัสและอะซิงโครนัส ซึ่งจัดการกับตัวรับอะซิงโครนัสโดยทั่วไปและตัวส่ง (UART) ใน3โหมดฟูลดูเพล็กซ์ (โหมด 1,2และ3) การส่งและรับอะซิงโครนัสสามารถเกิดขึ้นได้ในเวลาเดียวกันและที่อัตราบอร์ดต่างกัน

พอร์ตรับ/ส่งข้อมูลแบบอนุกรม ได้เพิ่มคุณสมบัติดังต่อไปนี้

- Framing error detection
- Automatic address recognition

2.3.2 Framing error detection

Framing bit error detection เป็นการเตรียมสำหรับโหมดอะซิงโครนัสทั้ง 3โหมด (โหมด1, 2 และ3) ทำให้สามารถมีรูปแบบการค้นพบเฟรมบิตผิดพลาดโดยเซ็ต SMOD ให้มีลอจิกเป็น 0 ในรีจีสเตอร์ PCON



รูปที่ 2.12 บล็อกไดอะแกรมFraming Error

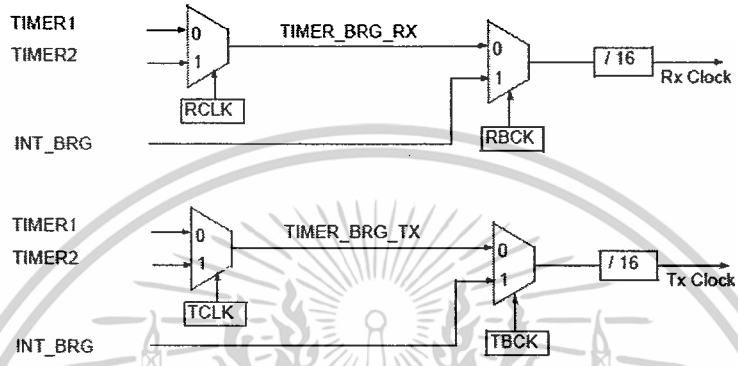
ตัวรับตรวจสอบเฟรมข้อมูลแต่ละเฟรมที่รับเข้ามาจนถึงบิตหยุดที่ถูกต้อง บิตหยุดที่ไม่ถูกต้องอาจเป็นผลมาจากตัวรบกวนบนสายอนุกรมหรือจากการส่งโดย CPU 2 ตัวในเวลาเดียวกัน ถ้าไม่พบบิตหยุดที่ถูกต้อง Framing Error bit (FE) ในรีจีสเตอร์ SCON จะถูกเซต ซอฟแวร์อาจจะตรวจสอบบิต FE หลังจากมีการรับรองการตรวจสอบข้อมูลผิดพลาด การเซตค่าเพียงใช้ซอฟแวร์หรือการรีเซตก็สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลียร์บิต FE ต่อมารับเฟรมกับบิตหยุดไม่สามารถเคลียร์บิต FEได้ เมื่อรูปแบบ FE ถูกใช้งาน เพิ่ม RI บนบิตหยุดแทนที่ข้อมูลบิตสุดท้าย

การเลือกอัตราบอร์ดสำหรับUARTสำหรับโหมด1และ3

อัตราบอร์ดสำหรับส่งและรับสัญญาณนาฬิกาสามารถถูกเลือกต่างหากผ่านทางรีจีสเตอร์ T2CON และBDRCON



รูปที่ 2.13 การเลือกอัตราบอร์ด

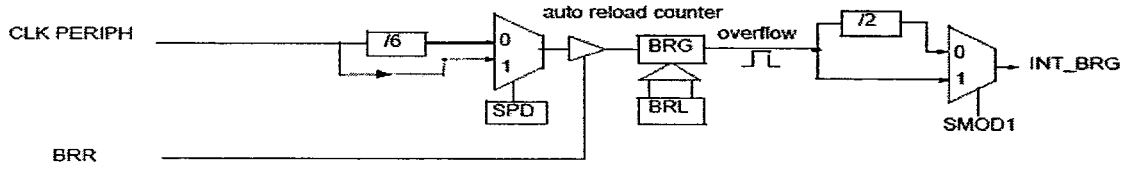
ตารางที่ 2.5 การเลือกอัตราบอร์ดสำหรับUART

TCLK (T2CON)	RCLK (T2CON)	TBCK (BDRCON)	RBCK (BDRCON)	Clock Source UART Tx	Clock Source UART Rx
0	0	0	0	Timer 1	Timer 1
1	0	0	0	Timer 2	Timer 1
0	1	0	0	Timer 1	Timer 2
1	1	0	0	Timer 2	Timer 2
X	0	1	0	INT_BRG	Timer 1
X	1	1	0	INT_BRG	Timer 2
0	X	0	1	Timer 1	INT_BRG
1	X	0	1	Timer 2	INT_BRG
X	X	1	1	INT_BRG	INT_BRG

Internal Baud Rate Generator (BRG)

เมื่อแหล่งกำเนิดอัตราบอร์ดภายในถูกใช้ อัตราบอร์ดถูกกำหนดโดยBRGซึ่งขึ้นอยู่กับค่าบรรจุใหม่ของ BRL ค่าของบิต SPD (Speed Mode) ใน รีจีสเตอร์BDRCON และค่าของบิตSMOD1บิตในรีจีส-เตอร์PCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 อัตราบอร์ดภายใน

สูตรอัตราบอร์ดสำหรับUART

$$Baud_Rate = \frac{2^{SMOD1} \times F_{CLK\ PERIPH}}{2 \times 2 \times 6^{(1-SPD)} \times 16 \times [256 - (BRL)]}$$

$$(BRL) = 256 - \frac{2^{SMOD1} \times F_{CLK\ PERIPH}}{2 \times 2 \times 6^{(1-SPD)} \times 16 \times Baud_Rate}$$

รีจีสเตอร์SCON-อนุกรมSCONควบคุมรีจีสเตอร์ (98h)

7	6	5	4	3	2	1	0
FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI

ตารางที่ 2.6 แสดงรีจิสเตอร์ที่ใช้การสื่อสารแบบอนุกรม

Bit Number	Bit Mnemonic	คำจำกัดความ
7	FE	Framing Error bit (SMOD0 = 1) เคลียร์เพื่อรีเซตข้อผิดพลาด, ไม่เคลียร์โดยบิตหยุดที่ถูกต้อง เซ็ตค่าโดยฮาร์แวร์เมื่อพบบิตหยุดที่ไม่ถูกต้อง SMOD0ต้องถูกเซ็ต เพื่อให้สามารถเข้าไปยังบิตFEได้
	SM0	พอร์ตอนุกรมโหมดบิต 0 อ้างถึง SM1 สำหรับการเลือโหมดพอร์ตอนุกรม SMOD0ต้องถูกเคลียร์เพื่อเข้าไปยังบิต SM0
6	SM1	พอร์ตอนุกรมโหมดบิต1 SM0 SM1 Mode Description Baud Rate 0 0 0 Shift Register FCPU PERIPH/6 0 1 1 8-bit UART Variable 1 0 2 9-bit UART FCPU PERIPH/32 or/16 1 1 3 9-bit UART Variable
5	SM2	พอร์ตอนุกรมโหมด2บิต/ Multiprocessor Communication Enable bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

		เคลียร์เพื่อ disable รูปแบบการติดต่อมัลติโปรเซสเซอร์ เซ็ตเพื่อ enable รูปแบบการติดต่อมัลติโปรเซสเซอร์ในโหมด2และ3, และ สุดท้ายโหมด1 บิตนี้ควรถูกเคลียร์ในโหมด0
4	REN	Reception Enable bit เคลียร์เพื่อdisable serial reception เซ็ตเพื่อ enable serial reception
3	TB8	ตัวส่งบิต8/บิตที่9เพื่อส่งในโหมด2และ3 เคลียร์เพื่อส่งลอจิก 0 ในบิตที่9
2	RB8	ตัวรับบิต8/บิตที่9รับในโหมด2และ3 เคลียร์โดยฮาร์แวร์ถ้าบิตที่9รับลอจิก0 เซ็ตโดยฮาร์แวร์ถ้าบิตที่9รับลอจิก1 ในโหมด1, ถ้า SM2 = 0, RB8เป็นบิตหยุดที่รับมา, ในโหมด0 RB8 ไม่ใช้
1	TI	ส่งInterrupt flag เคลียร์เพื่อรับรู้อินเตอร์รัปต์ เซ็ตโดยฮาร์ดแวร์ที่บิตสุดท้ายของบิตเวลาที่8ในโหมด0 หรือที่บิตเริ่มต้นของบิตหยุดในโหมดอื่น
0	RI	รับ Interrupt flag เคลียร์เพื่อรับรู้อินเตอร์รัปต์ เซ็ตโดยฮาร์ดแวร์ที่บิตสุดท้ายของบิตเวลาที่8ในโหมด0ดังรูป32และ33ในโหมดอื่น

Reset Value = 0000 0000b

Bit addressable

ตารางที่ 2.7 ตัวอย่างการคำนวณเมื่อ X2 = 1, SMOD1 = 1, SPD = 1

Baud Rates	F_{OSCA} = 16.384 MHz		F_{OSCA} = 24 MHz	
	BRL	Error (%)	BRL	Error (%)
115200	247	1.23	243	0.16
57600	238	1.23	230	0.16
38400	229	1.23	217	0.16
28800	220	1.23	204	0.16
19200	203	0.63	178	0.16
9600	149	0.31	100	0.16
4800	43	1.23	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
104332

ตารางที่ 2.8 ตัวอย่างการคำนวณเมื่อ X2 = 0, SMOD1 = 1, SPD = 0

Baud Rates	F_{OSCA} = 16.384 MHz		F_{OSCA} = 24 MHz	
	BRL	Error (%)	BRL	Error (%)
4800	247	1.23	243	0.16
2400	238	1.23	230	0.16
1200	220	1.23	202	3.55
600	185	0.16	152	0.16

เครื่องกำเนิดอัตราบอร์ดสามารถถูกใช้สำหรับโหมด 1 หรือ 3 แต่สำหรับโหมด0 สำหรับUART ที่บิต SRCในรีจีสเตอร์ BDRCON

UART Registers

SADEN – รีจีสเตอร์Slave Address Mask สำหรับ UART (B9h)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Reset Value = 0000 0000b

SADDR – รีจีสเตอร์Slave Address สำหรับ UART (A9h)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Reset Value = 0000 0000b

SBUF – รีจีสเตอร์Serial Buffer สำหรับ UART (99h)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Reset Value = XXXX XXXXb

BRL – รีจีสเตอร์บรรจุอัตราบอร์ดใหม่ สำหรับเครื่องกำเนิดอัตราบอร์ดภายใน UART (9Ah)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Reset Value = 0000 0000b

T2CON - Timer 2 Control Register (C8h)

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.9 ตารางรีจีสเตอร์T2CON

Bit Number	Bit Mnemonic	คำจำกัดความ
7	TF2	Timer 2 overflow Flag ต้องถูกเคลียร์โดยซอร์ฟแวร์ เซ็ตโดยฮาร์ดแวร์บน Timer 2 overflow ถ้า RCLK = 0 และ TCLK = 0
6	EXF2	Timer 2 External Flag เซ็ตเมื่อ capture หรือ reload ถูกทำให้เกิดโดย negative transitionบนขา T2EX ถ้า EXEN2 = 1 เมื่อเซ็ต เป็นเหตุให้CPUมักชี้ไปยังอินเตอร์รัปต์ Timer 2 เมื่ออินเตอร์รัปต์ Timer 2 enabled ต้องถูกเคลียร์โดยซอร์ฟแวร์EXF2 ซึ่งไม่ทำให้เกิดอินเตอร์รัปต์ขึ้น/ลงในโหมด counter (DCEN = 1)
5	RCLK	รับบิตสัญญาณนาฬิกาสำหรับ UART เคลียร์เพื่อใช้ Timer 1 overflow ดังที่รับสัญญาณนาฬิกาสำหรับพอร์ตอนุกรมในโหมด1หรือ3 เซ็ตเพื่อใช้ Timer 2 overflow ดังที่รับสัญญาณนาฬิกาสำหรับพอร์ตอนุกรมในโหมด1หรือ3
4	TCLK	Transmit Clock bit for UART เคลียร์เพื่อใช้ Timer 1 overflow ดังที่ส่งสัญญาณนาฬิกาสำหรับพอร์ตอนุกรมในโหมด1หรือ3 เซ็ตเพื่อใช้ Timer 2 overflowดังที่ส่งสัญญาณนาฬิกาสำหรับพอร์ตอนุกรมในโหมด1หรือ3
3	EXEN2	Timer 2 External Enable bit เคลียร์ไม่สนใจเหตุการณ์บนขาT2EXสำหรับการปฏิบัติการ Timer 2 เซ็ตเป็นเหตุให้ capture หรือ reload เมื่อ negative transitionบนขา T2EXที่ถูกพบ ถ้า Timer 2ไม่ถูกใช้สัญญาณนาฬิกาพอร์ตอนุกรม
2	TR2	Timer 2 Run control bit เคลียร์เพื่อปิด Timer 2 เซ็ตเพื่อเปิด Timer 2
1	C/T2#	Timer/Counter 2 select bit เคลียร์สำหรับขั้นตอน timer (ป้อนจากระบบสัญญาณนาฬิกาภายใน: FCLK PERIPH) เซ็ตสำหรับขั้นตอน counter(ป้อนจากขาinput T2,การลดขอบ trigger)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ต้องเป็น0สำหรับสัญญาณนาฬิกาออกจากโหมด
0	CP/RL2#	Timer 2 Capture/Reload bit ถ้า RCLK = 1 หรือ TCLK = 1, CP/RL2#ถูกละเลย และ timerถูกใช้แรงไปยัง Auto-reload บน Timer 2 overflow เคลียร์ Auto-reload บน Timer 2 overflows หรือ negative transitionsบนขา T2EX ถ้า EXEN2 = 1 เซ็ต capture บน negative transitionsบน ขาT2EX ถ้า EXEN2 = 1

Reset Value = 0000 0000b

Bit addressable

PCON - Power Control Register (87h)

7	6	5	4	3	2	1	0
SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL

ตารางที่ 2.10 รีจีสเตอร์PCON

Bit Number	Bit Mnemonic	คำจำกัดความ
7	SMOD1	Serial port Mode bit 1 for UART เซ็ตเลือกเพิ่มอัตราบอร์ดเป็น2เท่าในโหมด1, 2 หรือ 3
6	SMOD0	Serial port Mode bit 0 for UART เคลียร์เลือกบิต SM0ในรีจีสเตอร์ SCON เซ็ตเลือกบิต FE ในรีจีสเตอร์SCON
5	-	Reserved ค่าอ่านจากบิตนี้ซึ่งไม่แน่นอน ห้ามเซ็ตบิตนี้
4	POF	Power-Off Flag เคลียร์รู้จักชนิดของรีเซ็ตตัวต่อไป เซ็ตโดยฮาร์แวร์เมื่อเพิ่ม VCCจาก0ถึง nominal voltage สามารถถูกเซ็ตได้โดยซอร์ฟแวร์ด้วย
3	GF1	General-purpose Flag เคลียร์โดยผู้ใช้งาน สำหรับการใช้งานทั่วไป เซ็ตโดยผู้ใช้งาน สำหรับการใช้งานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2	GF0	General-purpose Flag เคลียร์โดยผู้ใช้งาน สำหรับการใช้งานทั่วไป เซ็ตโดยผู้ใช้งาน สำหรับการใช้งานทั่วไป
1	PD	Power-down Mode Bit เคลียร์โดยฮาร์ดแวร์เมื่อเกิดรีเซ็ตขึ้น เซ็ต เข้าสู่โหมด power-down
0	IDL	Idle Mode Bit เคลียร์โดยฮาร์ดแวร์เมื่อมีอินเตอร์รัปต์หรือรีเซ็ตเกิดขึ้น เซ็ต เข้าสู่โหมด idle

Reset Value = 00X1 0000b

Not bit addressable

Power-off flag รีเซ็ตค่าจะมีค่าเป็น1เท่านั้น หลังจากมีกำลัง(cold reset) ซึ่งwarm reset ไม่มีผลกับค่าของบิตนี้

BDRCON - Baud Rate Control Register (9Bh)

7	6	5	4	3	2	1	0
-	-	-	BRR	TBCK	RBCK	SPD	SRC

ตารางที่ 2.11 ตาราง BDRCON Register

Bit Number	Bit Mnemonic	คำจำกัดความ
7	-	Reserved ค่านี้อ่านจากบิตนี้ซึ่งไม่แน่นอน ห้ามเซ็ตบิตนี้
6	-	Reserved ค่านี้อ่านจากบิตนี้ซึ่งไม่แน่นอน ห้ามเซ็ตบิตนี้
5	-	ค่านี้อ่านจากบิตนี้ซึ่งไม่แน่นอน ห้ามเซ็ตบิตนี้
4	BRR	Baud Rate Run Control bit เคลียร์เพื่อหยุดตัวกำเนิดอัตราบอร์ดภายใน เซ็ตเพื่อเริ่มตัวกำเนิดอัตราบอร์ดภายใน
3	TBCK	Transmission Baud rate Generator Selection bit for UART เคลียร์เพื่อเลือก Timer 1 หรือ Timer 2 สำหรับตัวกำเนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		อัตราบอร์ด เซ็ตเพื่อเลือกตัวกำเนิดอัตราบอร์ดภายใน
2	RBCK	Reception Baud Rate Generator Selection bit for UART เคลียร์เพื่อเลือก Timer 1 หรือ Timer 2 สำหรับตัวกำเนิดอัตราบอร์ด เซ็ตเพื่อเลือกตัวกำเนิดอัตราบอร์ดภายใน
1	SPD	Baud Rate Speed Control bit for UART เคลียร์เพื่อเลือกกำเนิดอัตราบอร์ดอย่างช้า เซ็ตเพื่อเลือกกำเนิดอัตราบอร์ดอย่างเร็ว
0	SRC	Baud Rate Source select bit in Mode 0 for UART เคลียร์เพื่อเลือก FOSC/12 เหมือนดังตัวกำเนิดอัตราบอร์ด (FCLK PERIPH/6 in X2 mode) เซ็ตเพื่อเลือกตัวกำเนิดอัตราบอร์ดภายในสำหรับ UARTs ในโหมด0

Reset Value = XXX0 0000b

Not bit addressable

2.4 Delphi

เป็นซอฟต์แวร์ตัวแปลภาษาปาสกาล (Pascal) ตัวหนึ่ง ที่พัฒนาโดยบริษัท Borland ซึ่งรูปแบบของภาษานั้น ได้ถูกออกแบบระบบภาษาใหม่ เพื่อสนับสนุนการเขียนโปรแกรมแบบเชิงวัตถุ (Object oriented programming หรือเรียกแบบย่อว่า OOP) การเขียนโปรแกรมแบบเชิงวัตถุนั้น มีความแตกต่างกับการเขียนโปรแกรมแบบโครงสร้างอยู่ด้วยกันหลายจุด ซึ่งส่วนใหญ่แล้ว ความสามารถของโปรแกรมเชิงวัตถุ ยังคงต้องการความรู้ของการเขียนโปรแกรมแบบโครงสร้างเป็นพื้นฐาน ทั้งนี้เนื่องจากภาษาโปรแกรมเชิงวัตถุเป็นภาษาที่ออกแบบมาเพื่อเพิ่มประสิทธิภาพของการเขียนโปรแกรมในรูปแบบเดิม คือ ให้ง่ายต่อการพัฒนาต่อ และสามารถนำไปใช้ได้ (reusable)

สำหรับการเขียนโปรแกรมกับ Delphi นั้น จะใช้แนวทางการเขียนโปรแกรมแบบ Event Driven ซึ่งเป็นการเขียนโปรแกรมในลักษณะ ที่ว่า เมื่อมีเหตุการณ์อย่างใดอย่างหนึ่งเกิดขึ้น กับตัวโปรแกรม เราก็จะมีวิธีจัดการกับเหตุการณ์นั้นๆ ด้วยการเขียนคำสั่งต่างๆ รองรับเหตุการณ์ที่เกิดขึ้น นั่นเอง

2.4.1 Comport Library คอมโพเนนต์จัดการสื่อสารข้อมูลอนุกรม

สำหรับการทดลองเขียนโปรแกรมเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ MCS-51 และเช่นนี้ โปรแกรมDelphi ผ่านพอร์ตอนุกรมนั้น จะต้องใช้คอมโพเนนต์พิเศษเพื่อช่วยให้การติดต่อเกิดขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Comport Library คือไลบรารีที่รวบรวมฟังก์ชันต่าง ๆที่เกี่ยวข้องกับการสื่อสารข้อมูลอนุกรมของคอมพิวเตอร์เพื่อใช้งานร่วมกับDelphi ใน Comport Library มีชุดคอมโพเนนต์ที่ใช้คือ

1. คอมโพเนนต์ comport

ทำหน้าที่จัดการรับส่งข้อมูลผ่านพอร์ตอนุกรม เป็นคอมโพเนนต์ที่ไม่แสดงตัวในขณะรัน

คุณสมบัติที่ใช้งานบ่อย

Connect กำหนดให้คอมโพเนนต์ comport เชื่อมต่อการสื่อสารข้อมูลอนุกรม
กำหนดให้เป็น “TRUE” เพื่อเชื่อมต่อ (Connect)
กำหนดให้เป็น “FALSE” เพื่อยกเลิกการเชื่อมต่อ (Disconnect)
ค่า Default จะอยู่ที่ “FALSE”

Port กำหนดพอร์ตที่ใช้ในการสื่อสารข้อมูลอนุกรม ค่าdefault อยู่ที่ COM1

Baud Rate กำหนดอัตราบอร์ดในการสื่อสารข้อมูลอนุกรมค่า default อยู่ที่ br9600

อีเวนต์ที่ใช้งานบ่อย

OnRxChar เป็นอีเวนต์เมื่อเกิดเหตุการณ์รับข้อมูลเข้ามาทางพอร์ตอนุกรม ถ้าเปรียบเทียบกับไมโครคอนโทรลเลอร์คือ เกิดการอินเตอร์รัปต์จากพอร์ตอนุกรม ผู้พัฒนาจะต้องเขียนโปรแกรมตอบสนองไว้ภายในส่วนของอีเวนต์นี้ ซึ่งทำได้โดยการคลิกเลือกที่คอมโพเนนต์ comport แล้วดับเบิ้ลคลิกที่ส่วนของอีเวนต์ OnRxChar เพื่อเข้าไปกำหนดโปรแกรมตอบสนอง

Method ที่ใช้งานบ่อย

1) ส่งข้อมูลอักขระออกทางพอร์ตอนุกรมด้วยฟังก์ชัน Write

รูปแบบ

```
Function Write(Const Buffer; Count: Integer): Integer;
```

โดยที่ Buffer คือตัวแปรชนิดจำนวนเต็มที่นำมารับข้อมูลจากพอร์ตอนุกรม
Count คือตัวแปรเก็บจำนวนข้อมูลอักขระที่ส่งออกทางพอร์ตอนุกรม

ตัวอย่าง

```
var dat: byte;
dat :=$30;
comport1.Write(dat,1);          // ส่งข้อมูล 30H ออกพอร์ตอนุกรม
```

2) ส่งข้อมูลอักขระออกทางพอร์ตอนุกรมด้วยฟังก์ชัน WriteStr

รูปแบบ

```
function WriteStr(const Str: String): Integer;
```

โดยที่ Str คือสายอักขระที่ต้องการส่งออกไปทางพอร์ตอนุกรมของคอมพิวเตอร์

ตัวอย่าง

```
comport1.WriteStr('0');              // ส่งอักขระ “0” ออกทางพอร์ตอนุกรม
comport1.WriteStr('hello');          // ส่งสายอักขระ “hello” ออกทางพอร์ตอนุกรม
```

3) อ่านข้อมูลแบบจำนวนเต็มที่เข้ามาทางพอร์ตอนุกรมด้วยฟังก์ชัน Read

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ

```
function Read(var Buffer; Count: Integer): Integer;
```

โดยที่ Buffer คือตัวแปรชนิดจำนวนเต็มที่นำมารับข้อมูลจากพอร์ตอนุกรม

Count คือตัวแปรเก็บจำนวนข้อมูลอักขระที่ส่งออกทางพอร์ตอนุกรม

การใช้งานฟังก์ชันนี้มักจะมีการเรียกอ่านข้อมูลที่อีเวนต์ OnRxChar ซึ่งเป็นเหตุการณ์ตอบสนองคำสั่งภายใน เมื่อมีการรับข้อมูลเข้ามาทางพอร์ตอนุกรมคล้ายกับฟังก์ชันตอบสนองอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์

ตัวอย่าง

```
var dat: byte;                  // ตัวแปรบัฟเฟอร์เก็บข้อมูลที่เข้ามาทางพอร์ตอนุกรม
comport1.Read(dat,1);           // อ่านข้อมูลที่เข้ามาทางพอร์ตอนุกรม
edit1.text :=inttostr(dat);     // แสดงค่าข้อมูลแบบจำนวนเต็มที่edit1
```

ตัวอย่างการเรียกฟังก์ชัน Read ภายในอีเวนต์ OnRxChar ของคอมโพเนนต์ comport

```
procedure TForm1.Comport1RxChar (Sender: Object; Count: Integer);
var dat: byte;                  // ตัวแปรบัฟเฟอร์เก็บข้อมูลที่เข้าทางพอร์ตอนุกรม
begin
comport1.Read (dat,count);  // อ่านข้อมูลที่เข้ามาทางพอร์ตอนุกรม
end;
```

4) อ่านข้อมูลแบบสายอักขระ (สตริง : String) ที่เข้ามาทางพอร์ตอนุกรมด้วยฟังก์ชัน Readstr

รูปแบบ

```
function Readstr(const Str: String): Integer;
```

โดยที่ Str คือสายอักขระที่ต้องการส่งออกไปทางพอร์ตอนุกรมของคอมพิวเตอร์

การเรียกใช้งานฟังก์ชันนี้มักจะเรียกอ่านข้อมูลที่อีเวนต์ OnRxChar ซึ่งเป็นเหตุการณ์ตอบสนองคำสั่งภายในเมื่อมีการรับข้อมูลเข้ามาทางพอร์ตอนุกรมของคอมพิวเตอร์คล้ายกับฟังก์ชันตอบสนองการอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์

ตัวอย่าง

```
var mystr: string;              // ประกาศตัวแปร mystr เป็นแบบ string
comport1.Readstr(mystr);        // อ่านข้อมูลสายอักขระที่เข้าทางพอร์ตอนุกรมมาเก็บที่mystr
```

ตัวอย่างการเรียกฟังก์ชัน Readstr ภายในอีเวนต์ OnRxChar ของ comport

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var mystr: string;              //  ตัวแปรบัฟเฟอร์เก็บข้อมูลสายอักขระที่เข้ามาทางพอร์ต
                                //อนุกรมของคอมพิวเตอร์
 Begincomport1.Read(mystr,count);                    // อ่านข้อมูลสายอักขระที่เข้ามาทางพอร์ตอนุกรมเก็บที่mystr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
edit1.text := mystr;                    // แสดงสายอักขระที่รับเข้ามาที่คอมโพเนนต์
edit1
end;
```

5) กำหนดพารามิเตอร์ของการสื่อสารข้อมูลอนุกรม

ทำได้โดยการเรียกโพซิเยอร์ showsetupdialog เพื่อเปิด Dialog Setup สำหรับกำหนดพารามิเตอร์ของการสื่อสารข้อมูลอนุกรม ดังตัวอย่างต่อไปนี้

```
comport1.ShowSetupDialog;
```

2. คอมโพเนนต์ comled

เป็นคอมโพเนนต์แสดงสถานการณ์ทำงานในรูปของหลอดแสดงผล LED ในลักษณะต่าง ๆ ต้องใช้งานร่วมกับคอมโพเนนต์ comport สำหรับการสื่อสารข้อมูลอนุกรม โดยการกำหนดจากค่าคุณสมบัติ (Property) ที่ใช้งานบ่อย

คุณสมบัติ

ในหนึ่งโปรเจ็กต์สามารถใช้คอมโพเนนต์ comled ได้หลายดวง ส่วนคุณสมบัติที่มักใช้งานบ่อย ๆได้แก่

Comport ใช้กำหนดค่าอ้างอิงกับคอมโพเนนต์ comport1 ที่ใช้ร่วมกันในการแสดงผลสัญญาณของการสื่อสารข้อมูลอนุกรมนั้น ๆ จะต้องถูกกำหนดก่อนใช้งาน มิฉะนั้นจะไม่มีการแสดงผลใด ๆ

LedSignal ใช้เลือกชนิดของสัญญาณที่จะนำมาแสดงผลที่ comled กำหนดได้เป็น

- IsConn : กำหนดให้ comled แสดงสถานะการเชื่อมต่อของพอร์ตอนุกรม
- IsCTS : กำหนดให้ comled แสดงสถานะของสัญญาณ CTS
- IsDSR : กำหนดให้ comled แสดงสถานะของสัญญาณ DSR
- IsRing : กำหนดให้ comled แสดงสถานะของสัญญาณ Ring
- IsRLSD : กำหนดให้ comled แสดงสถานะของสัญญาณRLSD
- IsRx : กำหนดให้ comled แสดงสถานะของสัญญาณRx
- IsTx : กำหนดให้ comled แสดงสถานะของสัญญาณ Tx

สำหรับค่า Default ของคุณสมบัติ LedSignal อยู่ที่ IsConn kind ใช้เปลี่ยนสีของคอมโพเนนต์ comled

2.5 สเต็ปเปอร์มอเตอร์ (STEPPER MOTOR)

สเต็ปเปอร์มอเตอร์ เป็นมอเตอร์ที่มีการหมุนเป็นสเต็ป (STEP) เมื่อจ่ายกระแสไฟฟ้าไปยังขดลวด

ที่พันอยู่บนสเตเตอร์(Stator) ในลักษณะเป็นพัลส์ (Pulse) ในกรณีที่จ่ายกระแสไฟฟ้าให้กับขดลวดขดนั้นของสเต็ปเปอร์มอเตอร์ตลอดเวลาก็จะเกิดการหมุนเพียงหนึ่งสเต็ปเท่านั้นซึ่งต่างจาก DC มอเตอร์เมื่อจ่ายกระแสไฟฟ้าให้ ก็จะหมุนตามจนกว่าจะหยุดจ่ายกระแสไฟฟ้าให้

ข้อดีของสเต็ปเปอร์มอเตอร์เมื่อเปรียบเทียบกับ DC มอเตอร์

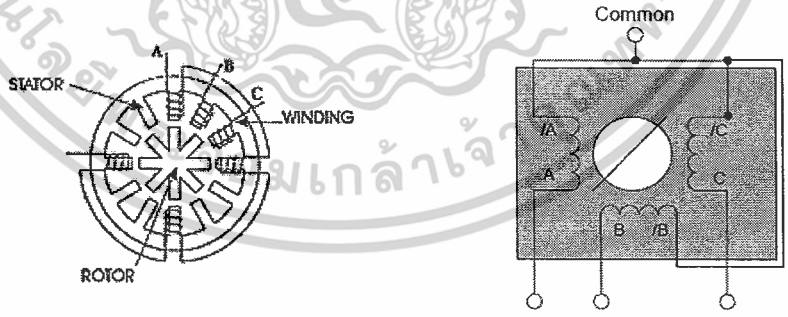
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สามารถใช้ในงานควบคุมตำแหน่งในลักษณะวงจรควบคุมแบบเปิด (Open Loop Control) ได้ โดยที่ไม่
ต้องการสัญญาณป้อนกลับ(Feedback signal) แต่อาศัยการนับจำนวนของพัลซ์ที่ส่งไปควบคุมการหมุนแทน
2. ไม่มีส่วนของแปรงถ่านที่จะสึกหรอและไม่เกิดการสปาร์คที่แปรงถ่านซึ่งอาจก่อให้เกิดสัญญาณรบกวน

2.5.1 ชนิดของสเต็ปเปอร์มอเตอร์

1) วาริเอเบิลรีลักแตนซ์ (Variable Reluctance : VR)

โรเตอร์(Rotor) ทำด้วยเหล็กอ่อนรูปทรงกระบอกและทำเป็นลักษณะฟัน(teeth) สเตเตอร์(Stator)จะมีลวดพัน และจะทำเป็นลักษณะของฟันเช่นกัน เมื่อจ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จะเกิดเป็นขั้วแม่เหล็กที่ฟันของสเตเตอร์และเหนี่ยวนำ ให้ฟันของโรเตอร์เกิดเป็นขั้วแม่เหล็กที่มีขั้วตรงกันข้ามกับสเตเตอร์ทำให้ดึงดูดกันเกิดการหมุนของโรเตอร์ขึ้น มอเตอร์ชนิดนี้โดยปกติจะมีขนาด 3 เฟส ในบางครั้งอาจพบ ถึง 4 เฟส มอเตอร์ชนิดนี้ถ้าไม่จ่ายกระแสไฟฟ้าให้กับขดลวดบนสเตเตอร์ ตัวโรเตอร์จะไม่เกิดแรงดึงดูดกับสเตเตอร์ มอเตอร์ชนิดนี้ไม่นิยมนำ ไปใช้ในงานอุตสาหกรรมแต่จะถูกนำไปใช้กับงานที่มีขนาดเล็ก เช่น Micro-positioning table เป็นต้น เพราะไม่มีส่วนที่เป็นแม่เหล็กถาวร ดังนั้นในขณะไม่จ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จึงไม่เกิดแรงดึงดูด วิธีการขับ(Driving)หรือการกระตุ้นเฟส(Phase Excitation)จะทำดังนี้คือ ต่อปลายด้านcommon เข้ากับแหล่งจ่ายไฟขั้วบวก(+) แล้วทำการสวิทช์ให้ปลายด้าน A , B , C ต่อลงกราวด์(Ground)ตามลำ ดับ ทีละปลายแล้วทำ เช่นนี้เรื่อยไป แต่ถ้าต้องการให้หมุนกลับก็สวิทช์ย้อนกลับ และในการอธิบายต่อจากนี้ไปจะไม่ขอกล่าวถึงมอเตอร์ชนิดนี้อีก



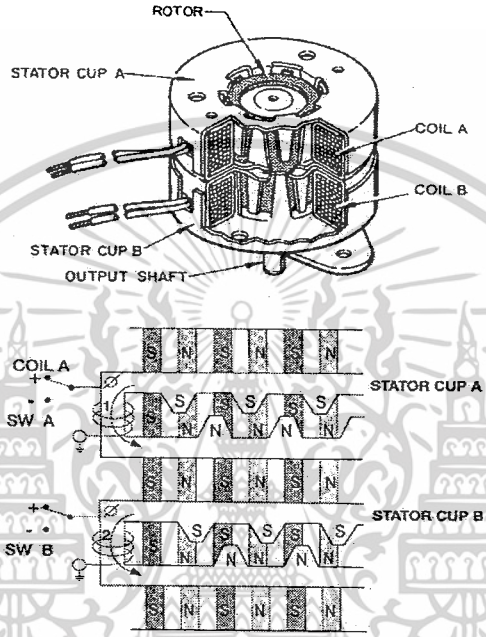
รูปที่ 2.15 โครงสร้างสเต็ปเปอร์มอเตอร์ชนิดวาริเอเบิลรีลักแตนซ์และสัญญลักษณ์

2) แบบแม่เหล็กถาวร (Permanent Magnet : PM)

โรเตอร์(Rotor) ทำด้วยแม่เหล็กถาวรรูปทรงกระบอกเรียบสเตเตอร์(Stator)จะมีขดลวดพันและก็จะทำเป็นฟัน เมื่อจ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จะเกิดเป็นขั้วแม่เหล็กที่ฟันของสเตเตอร์และจะดึงดูดกับขั้วของแม่เหล็กถาวรที่โรเตอร์ทำให้เกิดการหมุนของโรเตอร์ขึ้น มอเตอร์ชนิดนี้โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

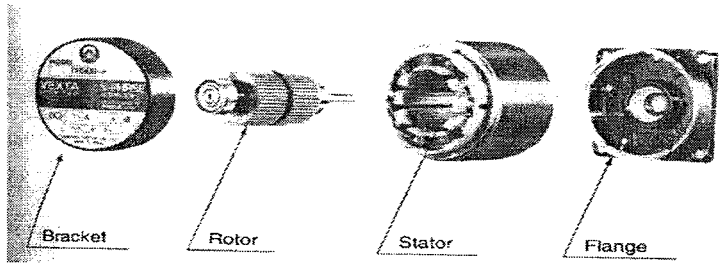
จะมีตั้งแต่ขนาด 2 เฟสขึ้นไปมอเตอร์ชนิดนี้ไม่นิยมนำไปใช้ในงานอุตสาหกรรมแต่จะถูกนำไปใช้กับอุปกรณ์คอมพิวเตอร์เช่นตัวขับวงล้อที่ใช้หมุนเพื่อเลื่อนกระดาษของเครื่องพิมพ์ เป็นต้น เพราะความเร็วต่ำ แรงบิดต่ำ และนอกจากนี้ด้วยโครงสร้างของมอเตอร์ชนิดนี้ทำให้มุมที่หมุนไปแต่ละสเต็ป(Step Angle)ไม่ละเอียดเช่น สเต็ปละ 3.6 ,7.5 , 15 ,18 องศาเป็นต้น มอเตอร์ชนิดนี้ถึงไม่จ่ายกระแสไฟฟ้าให้กับขดลวดบนสเตเตอร์ (Stator) ตัวโรเตอร์จะเกิดแรงดึงดูดกับสเตเตอร์ซึ่งเกิดจากอำนาจของแม่เหล็กถาวรที่โรเตอร์ทำให้หมุนได้ยาก จำนวนขั้วแม่เหล็กที่โรเตอร์สามารถนับได้จากจำนวนขั้วแม่เหล็กที่จะเกิดขึ้นจากกระแสไฟฟ้าไหลผ่านขดลวดที่สเตเตอร์ชุดใดชุดหนึ่ง



รูปที่ 2.16 สเต็ปเปอร์มอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ต

3) แบบผสม (Hybrid : HB)

ใช้หลักการทำงานของทั้งสองแบบมาออกแบบโดยที่สเตเตอร์จะคล้ายกับแบบVR ส่วนโรเตอร์จะคล้ายแบบPM แต่จะทำ เป็นฟัน มอเตอร์แบบนี้นิยมใช้ในงานอุตสาหกรรมเพราะแรงบิดสูง ความละเอียดของสเต็ปในการหมุนสูง , ความเร็วสูงกว่าสองแบบที่กล่าวมาแล้วมอเตอร์ชนิดนี้โดยปกติจะมีขนาด 2 เฟส ถึง 5 เฟส และมอเตอร์ชนิดนี้ได้มีการพัฒนาให้มีประสิทธิภาพ เหนือกว่าเดิมไปอีกโดยให้ชื่อว่า “Enhanced Hybrid” ซึ่งจะไม่อธิบายโครงสร้างในเอกสารนี้



รูปที่ 2.17 สเต็ปเปอร์มอเตอร์ชนิดไฮบริดขนาด 5 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

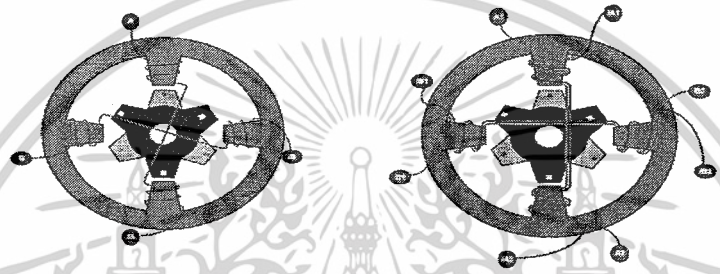
2.5.2 เฟสของสเต็ปเปอร์มอเตอร์ (Stepper Motor Phase)

หมายถึง จำนวนขดลวดที่พันอยู่บนสเตเตอร์ซึ่งแยกออกจากกันอย่างอิสระ รูปที่ 2.17 แสดงตัวอย่างมอเตอร์ขนาด 3 เฟส ในกรณีของมอเตอร์แบบยูนิโพลา 2 เฟสนั้นมักถูกจะเรียกเป็นมอเตอร์ขนาด 4 เฟสก็เพราะขดลวดที่พันอยู่บนสเตเตอร์แต่ละชุดจะมี 2 ขด จึงเข้าใจว่ามี 4 ขดลวด แต่ถ้าพิจารณากันจริงจะพบว่าขดลวดทั้งสองนั้นเป็นขดลวดขดเดียวแต่มีจุดต่อตรงกลางขดเท่านั้นดังแสดงในรูปที่ 2.22

2.5.3 การพันลวดบนสเตเตอร์ของสเต็ปเปอร์มอเตอร์ แบ่งออกได้ 2 แบบคือ

1) การพันแบบใช้ลวดเส้นเดียว (Unifilar)

2) การพันแบบใช้ลวด 2 เส้นพันไปด้วยกัน (Bifilar)

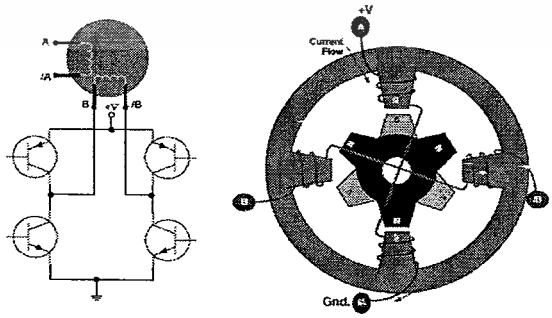


รูปที่ 2.18 ลักษณะการพันขดลวดบนสเตเตอร์

2.5.4 ชนิดของสเต็ปเปอร์มอเตอร์แบ่งตามลักษณะสายที่ใช้ต่อกับวงจรขับ แบ่งออกได้ 2 แบบคือ

1) แบบไบโพลา (Bipolar)

ขดลวดที่สเตเตอร์แต่ละชุดจะไม่มีจุดร่วม การต่อเข้ากับวงจรขับจะใช้ปลายทั้งสองด้านของขดลวดแต่ละชุด การทำให้เกิดขั้วแม่เหล็กที่สเตเตอร์ทำได้โดยการจ่ายกระแสไฟจากปลายด้านหนึ่งไปยังปลายอีกด้านหนึ่งของขดลวดและการเปลี่ยนขั้วแม่เหล็กที่สเตเตอร์ชุดเดียวกันนี้ก็ทำได้โดยสลับทิศทางการไหลของกระแสไฟฟ้านั่นเอง ดังนั้นวงจรขับที่ใช้จึงจำเป็นต้องสามารถกลับทิศทางการไหลของกระแสได้ กรณีเป็น
มอเตอร์ 2 เฟสจะมีสายที่ใช้ต่อขับกับวงจร 4 สาย

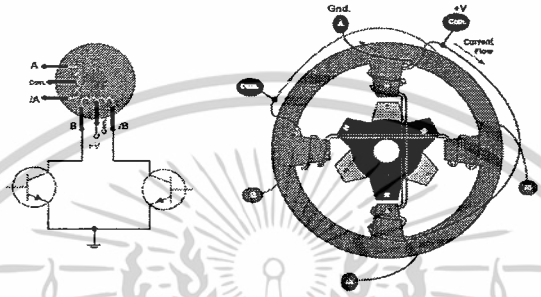


รูปที่ 2.19 สัญลักษณ์, โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบไบโพลา 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

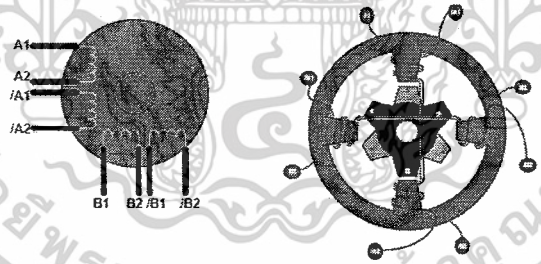
2) แบบยูนิโพลา (Unipolar)

ขดลวดที่สเตเตอร์แต่ละชุดจะมีจุดร่วม การพันขดลวดจะพันในแบบ Bifilar การต่อเข้ากับวงจรขับจะใช้ปลายของขดลวดแต่ละด้านต่อเข้ากับวงจรขับและใช้จุดร่วมต่อเข้ากับขั้วบวกของแหล่งจ่ายไฟ การทำให้เกิดขั้วแม่เหล็กที่สเตเตอร์ทำได้โดยการจ่ายกระแสไฟให้ไหลจากจุดร่วมลงกราวด์มาครบวงจรที่ปลายด้านหนึ่งของขดลวด การเปลี่ยนขั้วแม่เหล็กที่สเตเตอร์ชุดเดียวกันนี้ก็ทำได้โดยเปลี่ยนการจ่ายกระแสไฟฟ้าจากขดหนึ่งไปยังอีกขดหนึ่งของขดลวดที่พันอยู่บนสเตเตอร์ชุดเดียวกัน ดังนั้นวงจรขับจึงเป็นวงจรสวิทช์เพื่อทำให้จ่ายกระแสไฟที่ผ่านขดลวดครบวงจรเท่านั้น กรณีเป็นมอเตอร์ 2 เฟสจะมีสายที่ใช้ต่อขับกับวงจร 5 หรือ 6 สาย



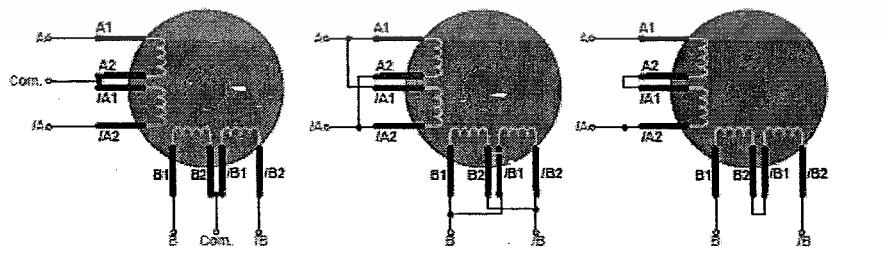
รูปที่ 2.20 สัญลักษณ์, โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบยูนิโพลา 2 เฟส

สเต็ปเปอร์มอเตอร์ 2 เฟสที่มีการพันลวดแบบไบฟิลาและแยกปลายแต่ละขดออกจากกันดังนั้นสามารถเลือกต่อเป็นแบบต่างๆได้ดังนี้



รูปที่ 2.21 สัญลักษณ์, โครงสร้างมอเตอร์2 เฟสที่พันลวดแบบไบฟิลาและแยกปลายของแต่ละขด

<u>ตัวอย่างการต่อสาย</u>



รูปที่ 2.22 สัญลักษณ์การต่อสายแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ที่ผลิตออกมาจำหน่ายและมีการพันลวดแบบไบโพลาบางรุ่นให้ผู้ใช้เลือกต่อเองเป็นแบบต่างๆแต่บางรุ่นก็ต่อภายในมาให้เสร็จ

2.5.5 วิธีการขับ (Driving) หรือวิธีการกระตุ้นเฟส (Phase Excitation) ของสเต็ปเปอร์มอเตอร์

การกระตุ้นเฟสของสเต็ปเปอร์มอเตอร์ คือ การจ่ายกระแสไฟฟ้าไปยังขดลวดที่สเตเตอร์ของแต่ละเฟสเพื่อทำ ให้มอเตอร์หมุนนั่นเอง แบ่งออกเป็น 3 วิธีคือ

1) การกระตุ้นเฟสแบบ One phase excitation หรือ Wave หรือ Half Drive

การกระตุ้นเฟสแบบนี้ทำได้โดยการจ่ายกระแสไฟฟ้าไปยังขดลวดครั้งละหนึ่งขดบนสเตเตอร์ดังนี้

Step	Unipolar		Bipolar	
	Supply		Supply	
	+V	Gnd.	+V	Gnd.
1	Com.	A	A	/A
2	Com.	B	B	/B
3	Com.	/A	/A	A
4	Com.	/B	/B	B

2) การกระตุ้นเฟสแบบ Two phase excitation หรือ Full Step

การกระตุ้นเฟสแบบนี้ทำได้โดยการจ่ายกระแสไฟฟ้าไปยังขดลวดครั้งละสองขดที่อยู่ใกล้กันบนสเตเตอร์ดังนี้

Step	Unipolar		Bipolar	
	Supply		Supply	
	+V	Gnd.	+V	Gnd.
1	Com.	A, B	A, B	/A, /B
2	Com.	/A, B	/A, B	A, /B
3	Com.	/A, /B	/A, /B	A, B
4	Com.	A, /B	A, /B	/A, B

3) การกระตุ้นเฟสแบบ One – Two phase excitation หรือ Half Step

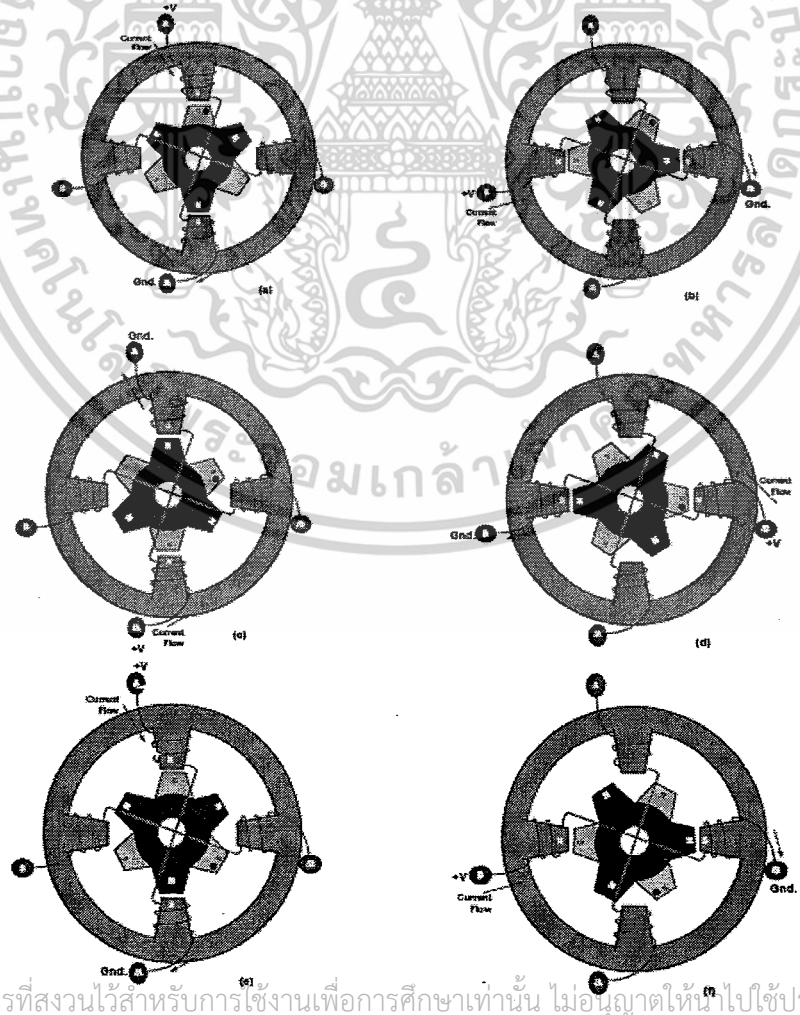
การกระตุ้นเฟสแบบนี้ทำได้โดยการจ่ายกระแสไฟฟ้าไปยังขดลวดครั้งละสองขดที่อยู่ใกล้กันบนสเตเตอร์สลับกับการจ่ายกระแสไฟฟ้าครั้งละหนึ่งขดบนสเตเตอร์ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

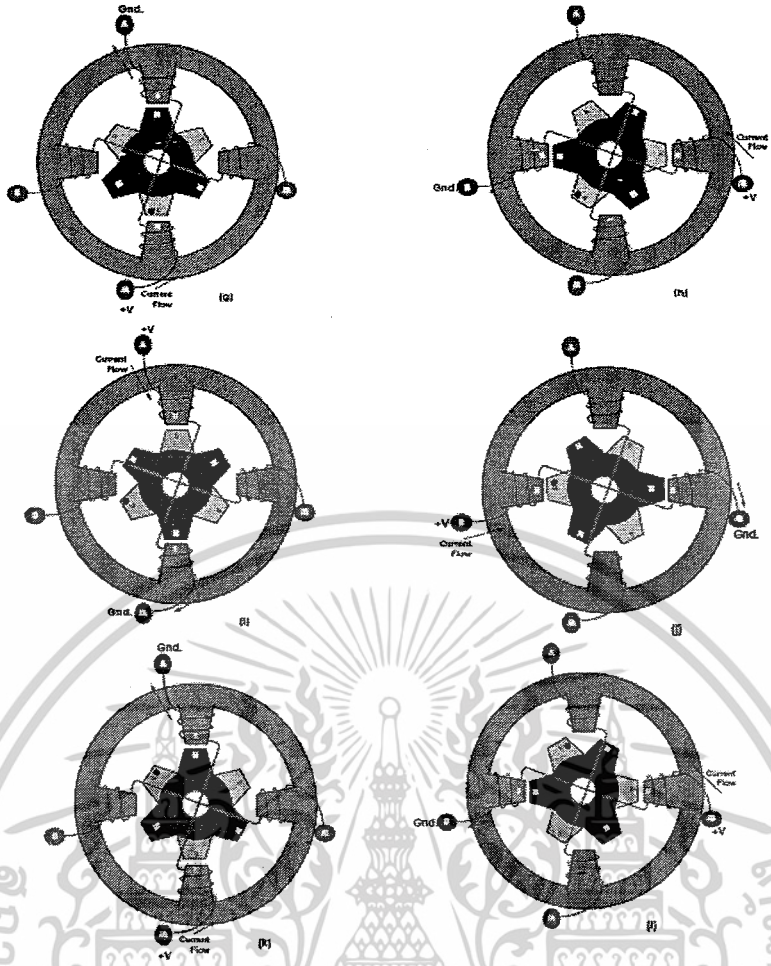
Step	Unipolar		Bipolar	
	Supply		Supply	
	+V	Gnd.	+V	Gnd.
1	Com.	A, B	A, B	/A, /B
2	Com.	B	B	/B
3	Com.	/A, B	/A, B	A, /B
4	Com.	/A	/A	A
5	Com.	/A, /B	/A, /B	A, B
6	Com.	/B	/B	B
7	Com.	A, /B	A, /B	/A, B
8	Com.	A	A	/A

หมายเหตุ 1.ค่าคงที่ (f) ที่ใช้ในการคำนวณหา Step per Rev. วิธีที่1และ 2 มีค่าเท่ากับ 1 ส่วนวิธีที่ 3 มีค่าเท่ากับ 2

ตัวอย่างการหมุนและการกระตุ้นมอเตอร์แบบไบโพลา 2 เฟส



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 ตัวอย่างการหมุนและการกระตุ้นมอเตอร์แบบไบโพลา 2 เฟส

2.6 L298

เป็นตัวไดร์ฟแบบ dual full-bridge ซึ่งไดร์ฟโวลต์เตจและกระแสสูง ๆ ถูกออกแบบตามมาตรฐาน TTL

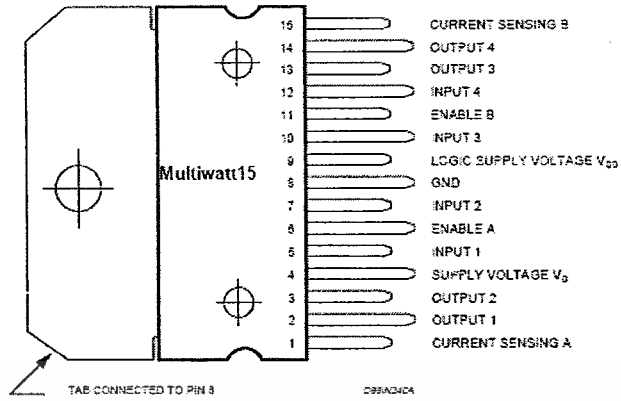
logic levels และใช้ไดร์ฟinductive loads เช่น รีเลย์, โซลีนอยด์, DC มอเตอร์ และสเต็ปปิ้งมอเตอร์ มีอินพุตenable 2 ขา เป็นตัวเงื่อนไขว่าจะenable หรือdisableอุปกรณ์ของสัญญาณอินพุต emittersของ lower transistorsแต่ละบริดจ์จะถูกเชื่อมโยงถึงกันและตรงกับเทอร์มินอลภายนอกสามารถถูกใช้สำหรับตัวติดต่อของsensing resistorภายนอก logicทำงานเมื่อป้อนแหล่งจ่ายไฟอินพุตโวลต์ต่ำ

<u>ลักษณะทั่วไป</u>

- ทำงานเมื่อป้อนsupply voltage สูงสุดถึง 46 v
- กระแสไฟตรงสูงสุดถึง4A
- Saturation voltage ต่ำ
- การป้องกันอุณหภูมิสูง ๆ
- ลอจิก “0” อินพุตโวลเตจสูงสุดถึง1.5v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของขา



รูปที่ 2.24 แสดงลักษณะของขาL298

2.7 PC 817 และ PC 847

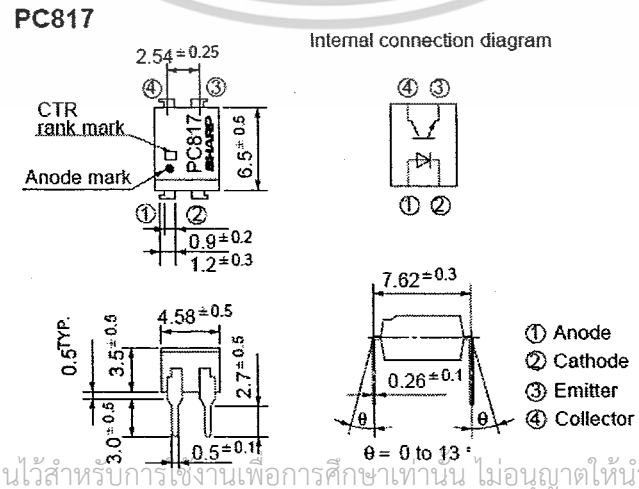
ลักษณะทั่วไป

อัตราส่วนกระแสส่งผ่าน I_F = 5mA , V_{CE}=5V , แยกโวลต์สูงระหว่างอินพุตและเอาท์พุต Viso : 5,000 Vrms ซึ่ง PC 817 เป็นชนิด 1-channal และ PC 847 เป็นชนิด 4-channal

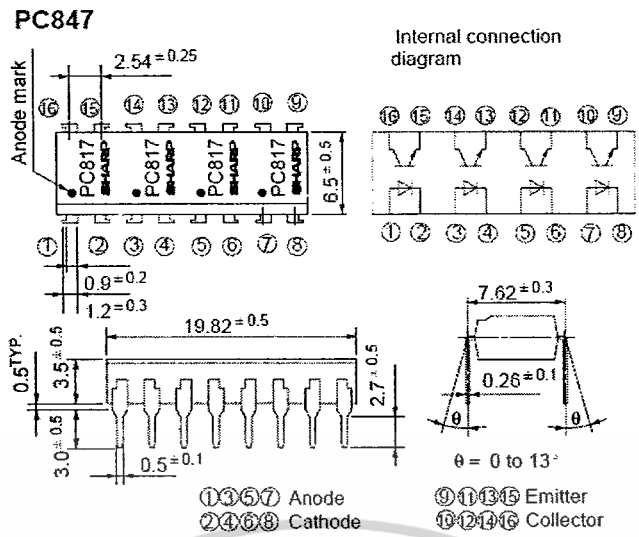
การประยุกต์ใช้

- ใช้กับเทอร์มินอลคอมพิวเตอร์
- เครื่องมือวัด
- ใช้กับอุปกรณ์อิเล็กทรอนิกส์ภายในบ้าน เช่น พัดลม, เครื่องทำความร้อน เป็นต้น
- ใช้กับเครื่องถ่ายเอกสาร, automatic vending machines
- เครื่องมือทางการแพทย์, อุปกรณ์ทางฟิสิกส์และเคมี
- การส่งผ่านสัญญาณระหว่างวงจรของ different potentials และ impedances

ลักษณะของขา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 แสดงลักษณะของขา PC817 และ PC847

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

3.1 การออกแบบระบบ

การออกแบบระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน Zigbee แบ่งการทำงานหลักออก 3 ส่วนดังนี้

3.1.1 ฝั่งคอมพิวเตอร์

เป็นส่วนอินเตอร์เฟสระหว่างผู้ใช้งานและคอมพิวเตอร์ โดยส่วนนี้จะเป็นโปรแกรมคอมพิวเตอร์แบ่งการทำงานออกเป็น 3 ส่วนคือ

1. ส่วนควบคุมการใช้งานของอุปกรณ์ไฟฟ้า

เป็นส่วนที่ใช้ควบคุมการทำงานของอุปกรณ์ไฟฟ้าซึ่งในโครงงานจะนำเสนอการควบคุมมอเตอร์และหลอดไฟโดยจะแสดงหน้าจอการควบคุมการหมุนมอเตอร์ทางซ้ายและทางขวาตามองศาต่างๆ และทำการส่งค่าองศาที่ต้องการไปยังส่วนควบคุมอุปกรณ์ไฟฟ้า

2. ส่วนแสดงผลการทำงานของอุปกรณ์ไฟฟ้า

แสดงสถานะการทำงานมอเตอร์ว่าขณะนั้นทำงานอยู่ที่องศาใด (สถานะจะถูกแสดงเป็นข้อความที่คอมโพเนนต์ Label1 และ Label2 ของหน้าต่างโปรแกรม) และแสดงสถานะของหลอดไฟในปัจจุบัน โดยจะร้องขอค่าสถานะจากส่วนควบคุมอุปกรณ์ไฟฟ้า

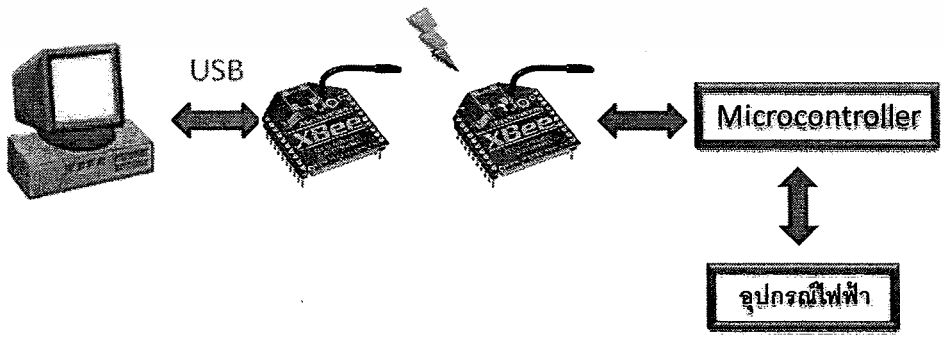
3.1.2 ฝั่งไมโครคอนโทรลเลอร์

ส่วนนี้จะเป็นการควบคุมอุปกรณ์ไฟฟ้าด้วยการใช้ไมโครคอนโทรเลอร์ ไมโครคอนโทรลเลอร์จะรับคำสั่งจากคอมพิวเตอร์และทำตามคำสั่งนั้น

3.1.3 การเชื่อมต่อระหว่างฝั่งคอมพิวเตอร์และฝั่งไมโครคอนโทรลเลอร์โดยผ่านโมดูล Zigbee

โมดูล Zigbee จะใช้ในการส่งผ่านคำสั่งระหว่างฝั่งคอมพิวเตอร์และฝั่งไมโครคอนโทรลเลอร์ โดยจะมีโมดูล Zigbee 2 ตัว ตัวที่ 1 จะเชื่อมต่อกับคอมพิวเตอร์ที่มีโปรแกรมควบคุมอุปกรณ์ไฟฟ้าและตัวที่ 2 เชื่อมต่อกับไมโครคอนโทรลเลอร์ที่เป็นส่วนควบคุมอุปกรณ์ไฟฟ้า โมดูล Zigbee จะทำการส่งคำสั่งหากันผ่านความถี่ 2.4 GHz

3.1.4 แสดงบล็อกไดอะแกรมการทำงานของระบบ



รูปที่ 3.1 แสดงการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบคำสั่งควบคุมของอุปกรณ์ไฟฟ้า

เมื่อโปรแกรมฝั่งคอมพิวเตอร์รับค่าคำสั่งจากผู้ใช้งานจะส่งคำสั่งไปยังไมโครคอนโทรลเลอร์เพื่อควบคุมอุปกรณ์ไฟฟ้า (โครงงานนี้นำเสนออุปกรณ์ไฟฟ้าคือมอเตอร์และหลอดไฟ)

3.3 การออกแบบแสดงสถานะของอุปกรณ์ไฟฟ้า

เมื่อไมโครคอนโทรลเลอร์ได้รับอักขระ "R" ที่เป็นคำสั่งอ่านค่าสถานะปัจจุบันของอุปกรณ์ไฟฟ้า ไมโครคอนโทรลเลอร์ก็จะทำการส่งสถานะมุมของมอเตอร์กลับไปยังฝั่งคอมพิวเตอร์ดังตารางนี้

ตารางที่ 3.1 ข้อมูลสถานะของอุปกรณ์ไฟฟ้าที่ไมโครคอนโทรลเลอร์ทำการส่งกลับไปยังโปรแกรมควบคุมอุปกรณ์ไฟฟ้า

คำสั่ง	เงื่อนไขการทำงาน		
	รอบการหมุน	องศา	ทิศทางการหมุน
0	0	0°	
1	1	12.5°	ฟอร์เวิร์ส
2	2	25°	ฟอร์เวิร์ส
3	3	37.5°	ฟอร์เวิร์ส
4	4	50°	ฟอร์เวิร์ส
5	5	62.5°	ฟอร์เวิร์ส
6	6	75°	ฟอร์เวิร์ส
7	7	87.5	ฟอร์เวิร์ส
8	8	100°	ฟอร์เวิร์ส
/	1	12.5°	รีเวิร์ส
.	2	25°	รีเวิร์ส
-	3	37.5°	รีเวิร์ส
,	4	50°	รีเวิร์ส
+	5	62.5°	รีเวิร์ส
*	6	75°	รีเวิร์ส
)	7	87.5	รีเวิร์ส
(	8	100°	รีเวิร์ส
a	หลอดไฟทำงาน		
b	หลอดไฟไม่ทำงาน		

3.4 การออกแบบการทำงานของโปรแกรมควบคุมอุปกรณ์ไฟฟ้า

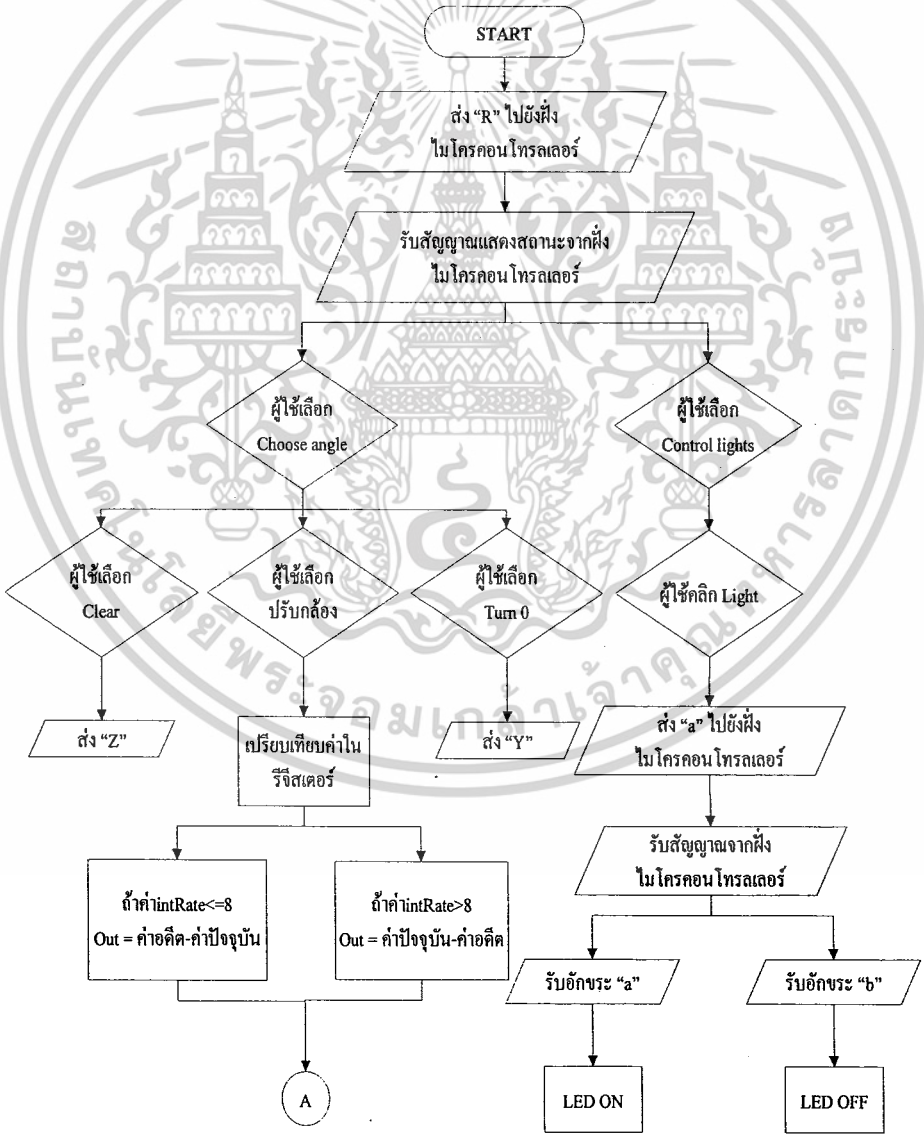
3.4.1 การออกแบบโปรแกรมฝั่งคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบส่วนของโปรแกรมฝั่งคอมพิวเตอร์ เริ่มต้นโดยเมื่อเปิดโปรแกรม โปรแกรมจะส่งอักขระ “R” ไปยังฝั่งไมโครคอนโทรลเลอร์ เพื่อตรวจสอบสถานะอุปกรณ์ จากนั้นจึงรับค่าสถานะกลับมาแสดงบนหน้าจอ ผู้ใช้งานทำการเลือกควบคุมกล้องหรือควบคุมหลอดไฟ

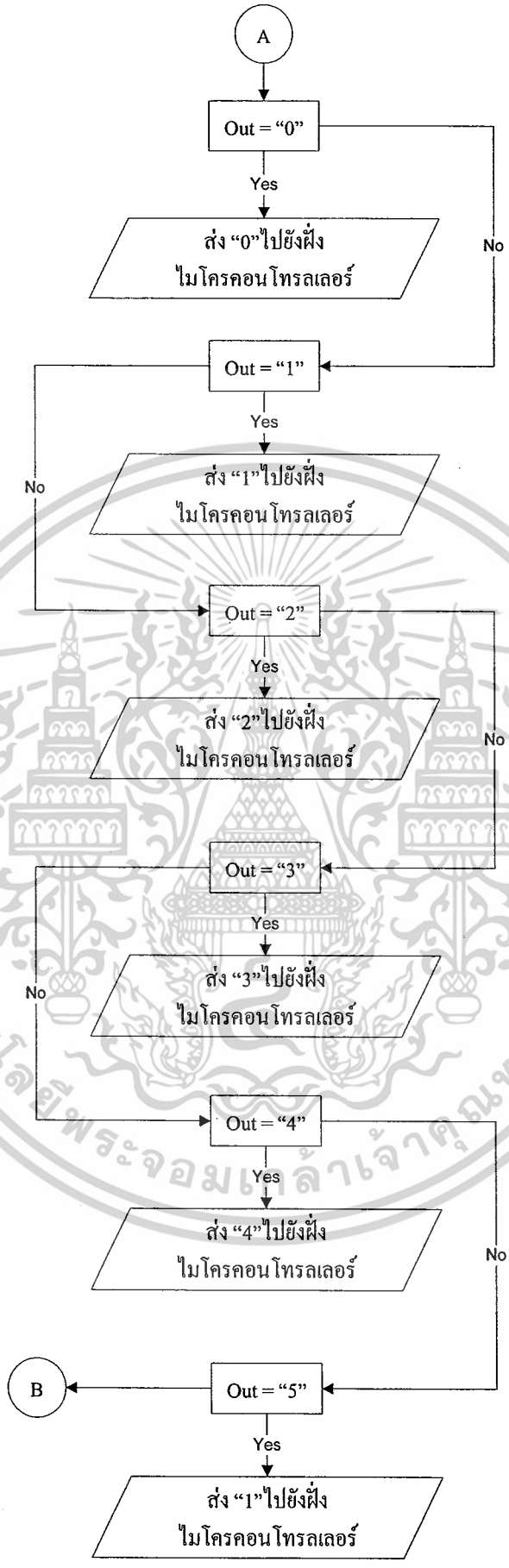
- ถ้าผู้ใช้งานเลือกควบคุมกล้อง โดยทำการเลื่อนแถบตามองศาที่ต้องการให้กล้องหมุน โปรแกรมจะทำการเทียบค่าในรีจีสเตอร์แล้วส่งค่าอักขระตามเงื่อนไขไปยังฝั่งไมโครคอนโทรลเลอร์ หรือเลือก Clear เพื่อเคลียร์ค่าในEEPROM โดยโปรแกรมจะส่งอักขระ “Z” หรือ เลือกTurn0 เพื่อให้กล้องหมุนกลับมาที่ 0 องศา โดยโปรแกรมจะส่งอักขระ “Y”
- ถ้าผู้ใช้งานเลือกควบคุมหลอดไฟ เมื่อคลิกLight จะส่งอักขระ “a” ไปยังฝั่งไมโครคอนโทรลเลอร์ จากนั้นรอรับอักขระจากฝั่งไมโครคอนโทรลเลอร์ ถ้ารับมาเป็นอักขระ “a” แสดงสถานะไฟติด ถ้าเป็นอักขระ “b” แสดงสถานะไฟดับ

การออกแบบโปรแกรมนี้เป็นดังรูปที่ 3.2 – 3.5



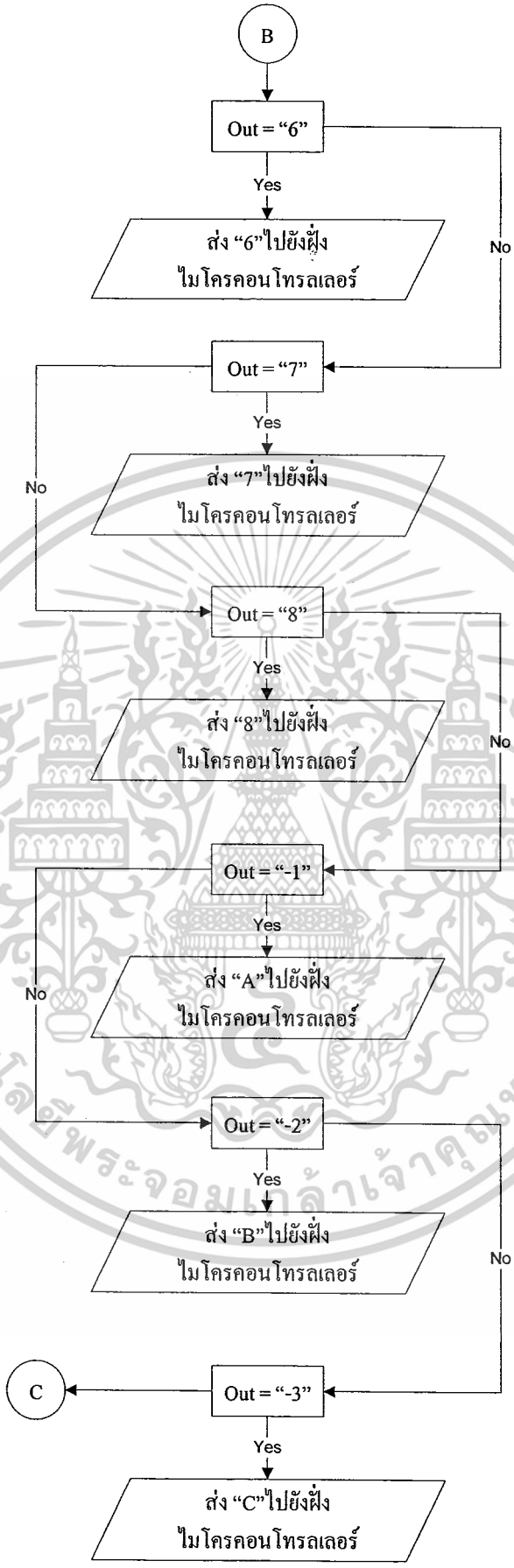
รูปที่ 3.2 แผนผังการทำงานของโปรแกรมฝั่งคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



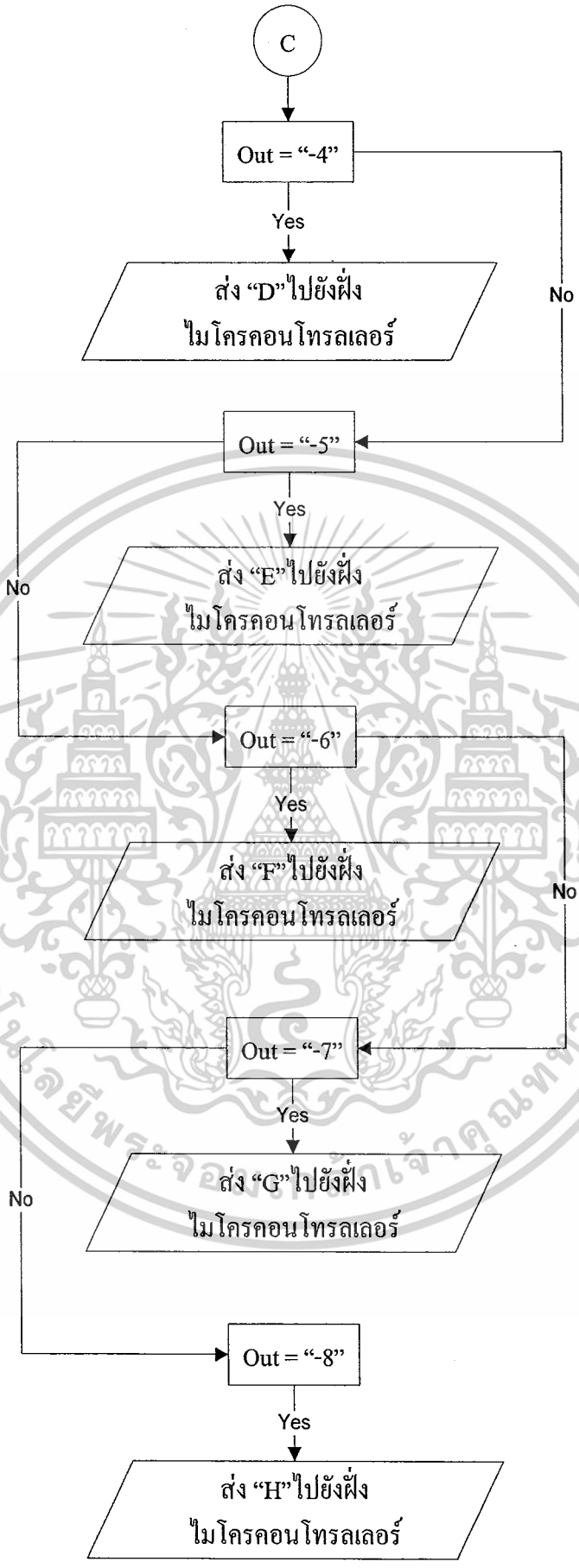
รูปที่ 3.3 แผนผังการทำงานของโปรแกรมฝั่งคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แผนผังการทำงานของโปรแกรมฝั่งคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

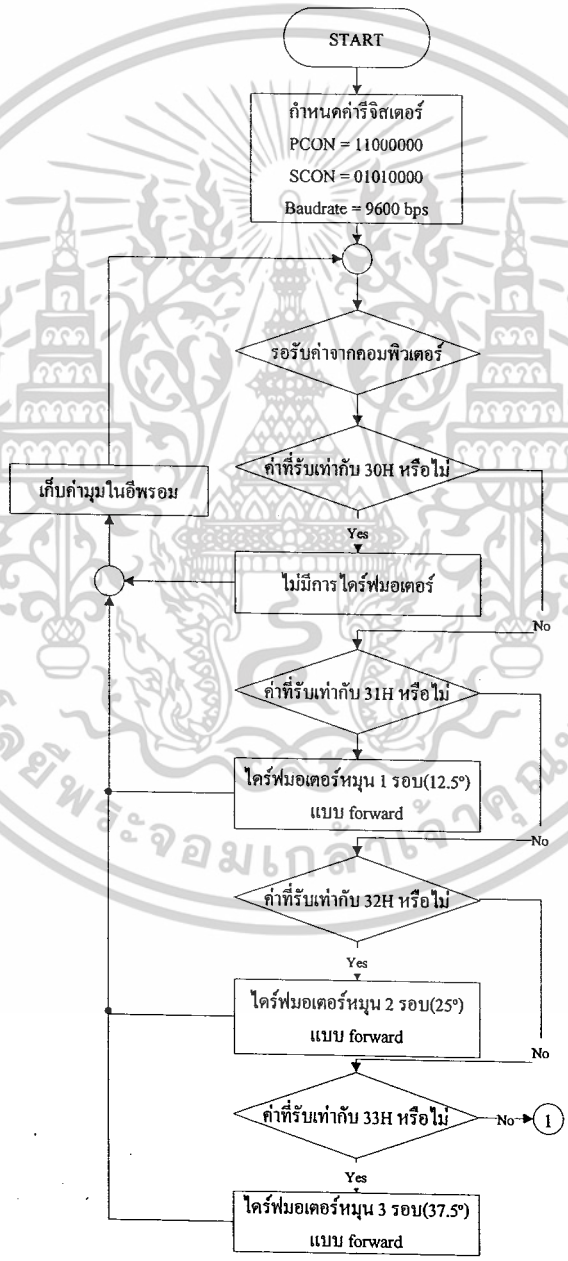


รูปที่ 3.5 แผนผังการทำงานของโปรแกรมฝั่งคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การออกแบบโปรแกรมส่วนฝั่งไมโครคอนโทรลเลอร์

การออกแบบโปรแกรมทางฝั่งไมโครคอนโทรลเลอร์ การทำงานเริ่มต้นจากการกำหนดบอร์ดเรทของไมโครคอนโทรลเลอร์ให้ตรงกับโมดูล Zigbee ก่อนเพื่อใช้ในการรับส่งคำสั่งจากโปรแกรมทางฝั่งคอมพิวเตอร์ เมื่อกำหนดค่าแล้วจะทำการรอรับคำสั่ง เมื่อมีคำสั่งมาจะทำการตรวจสอบเงื่อนไขการทำงาน เช่น ถ้ารับค่า R จะทำการเรียกค่าสถานะปัจจุบันของอุปกรณ์ไฟฟ้าจากวงจรตรวจสอบการทำงานและส่งค่าไปยังฝั่งไมโครคอนโทรลเลอร์ยกตัวอย่างส่ง 1 แสดงว่าขณะนี้มอเตอร์อยู่ที่มุม 12.5 องศาทางด้านฟอร์เวิร์ส และเมื่อมีคำสั่งอื่นมาอีกไมโครคอนโทรลเลอร์ก็จะตรวจสอบเงื่อนไขและทำงานตามคำสั่งต่อไป ดังรูปที่ 3.6 – 3.9

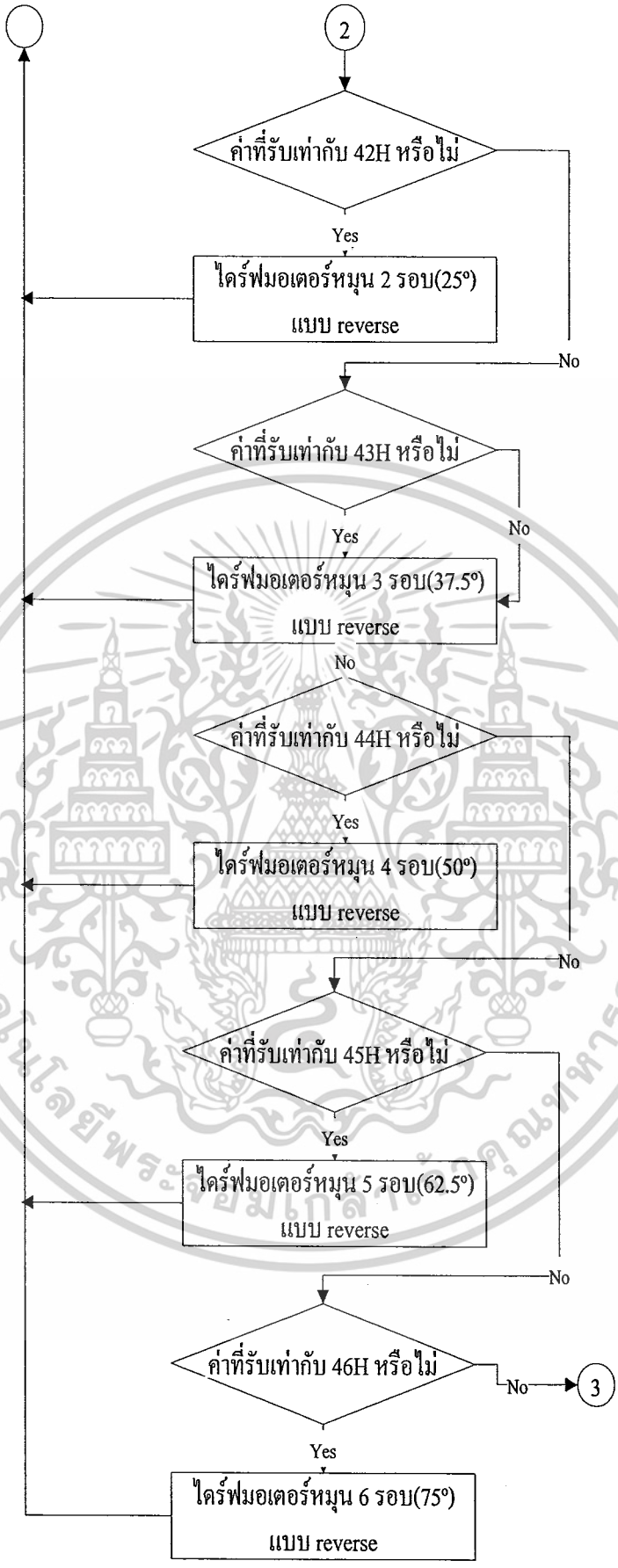


รูปที่ 3.6 แผนผังการทำงานของส่วนควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.7 แผนผังการทำงานของส่วนควบคุมอุปกรณ์ไฟฟ้า (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แผนผังการทำงานของส่วนควบคุมอุปกรณ์ไฟฟ้า (ต่อ)

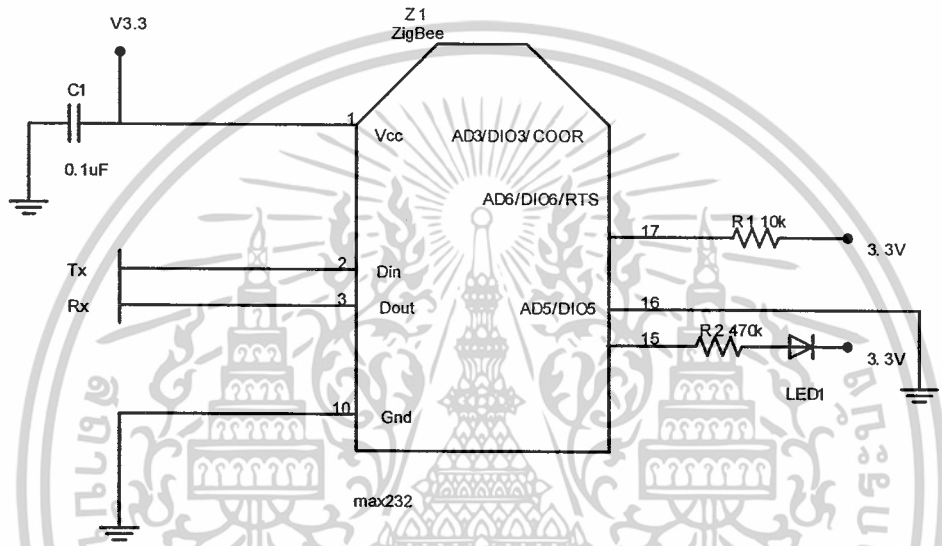
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 แผนผังการทำงานของส่วนควบคุมอุปกรณ์ไฟฟ้า (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

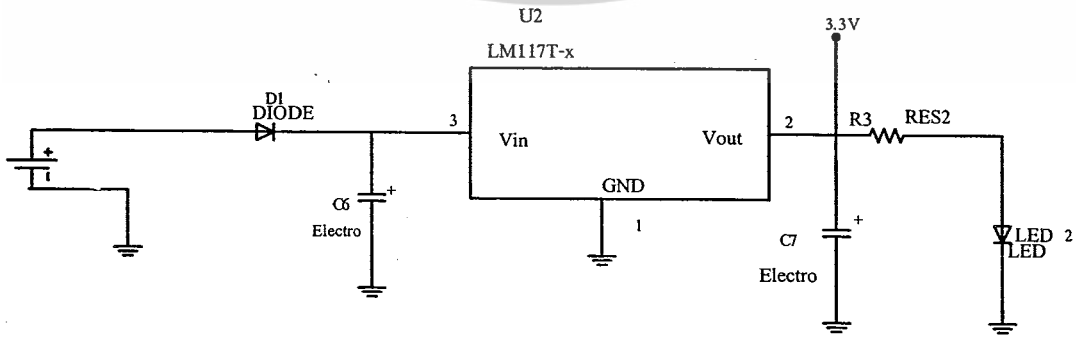
3.5 การออกแบบวงจรด้านรับและด้านส่งของโมดูล Zigbee

การทำงานของวงจรคือ เมื่อทำการป้อนไฟเลี้ยงแก่วงจรจะผ่านเข้าไอซี LM117T-X ก่อนเพื่อป้องกันการป้อนไฟเลี้ยงเกินเนื่องจากโมดูล Zigbee ใช้ไฟเลี้ยงเพียง 3.3 โวลต์ จากนั้นสัญญาณทางฝั่งคอมพิวเตอร์จะเข้าไอซี Max 232 เพื่อแปลงระดับแรงดันก่อนจะเข้าสู่โมดูล Zigbee โมดูล Zigbee จะส่งสัญญาณเป็นคลื่นวิทยุความถี่ 2.4 GHz ออกทางขา Tx และเข้าจะผ่านทางขา Rx ของตัวโมดูล Zigbee ในฝั่งไมโครคอนโทลเลอร์เพื่อประมาวผลการทำงานต่อไป หากต้องการส่งสัญญาณจากฝั่งไมโครคอนโทลเลอร์ไปหาฝั่งคอมพิวเตอร์ก็จะเป็นกระบวนการเดียวกันเพียงแต่ทิศทางตรงกันข้ามเท่านั้น



รูปที่ 3.10 แสดงวงจรทำงานของโมดูล Zigbee

เป็นวงจรที่ใช้แปลงระดับแรงดัน คือ เมื่อป้อนไฟเลี้ยง 5 โวลต์แก่วงจรจะต้องผ่านไอซี LM117T-X ก่อนเพื่อป้องกันการป้อนไฟเลี้ยงเกินเนื่องจากโมดูล Zigbee ใช้ไฟเลี้ยงเพียง 3.3 โวลต์ หากป้อนแรงดันเกินระดับนี้โมดูล Zigbee จะเกิดการเสียหาย

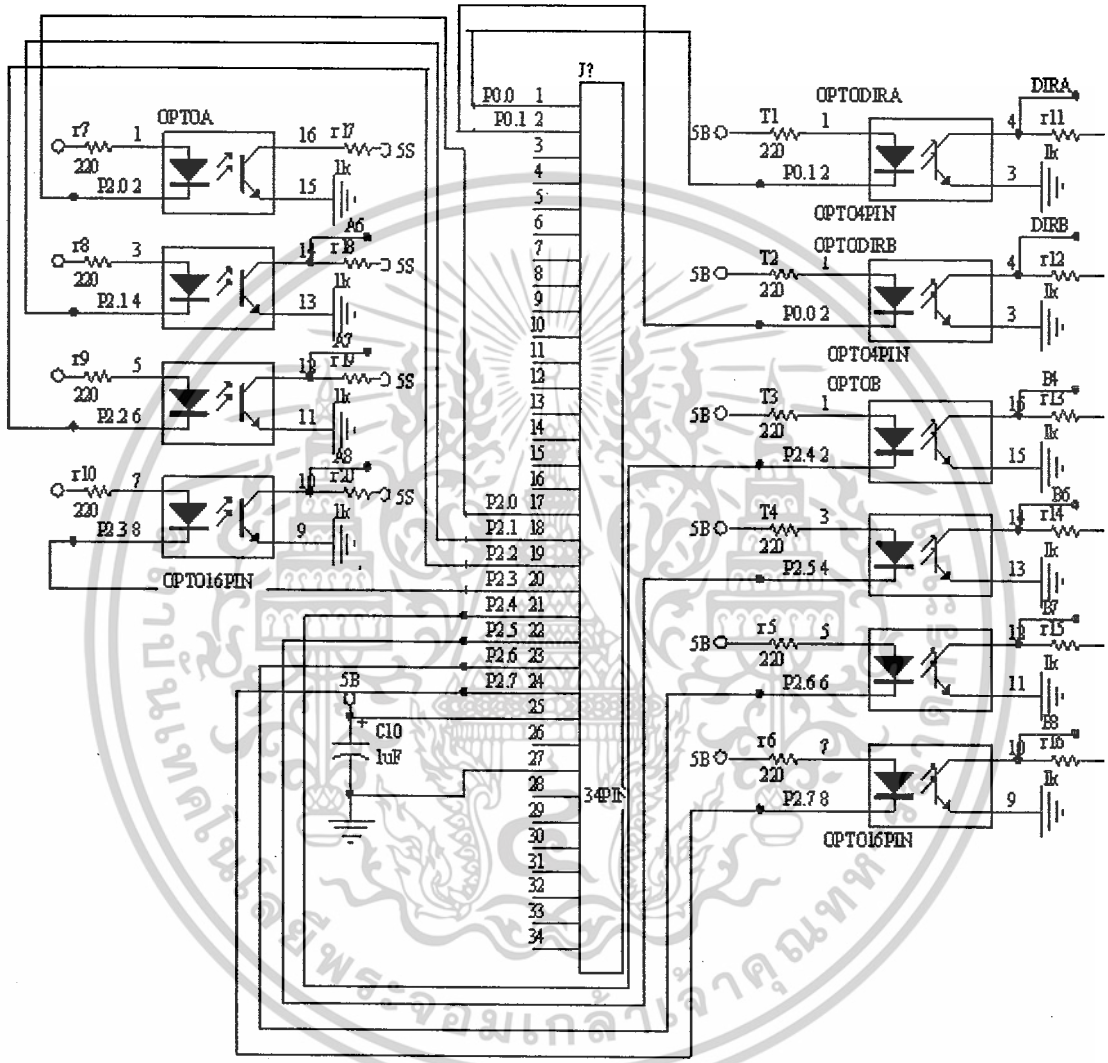


รูปที่ 3.11 แสดงวงจรแปลงระดับแรงดันสัญญาณอินพุทเป็น 3.3 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 วงจรโฟโต้คอปเปอร์ LTV 847 และ PC 817

หลักการทำงานของวงจรโฟโต้คอปเปอร์ LTV 847 และ PC 817 จะเป็นวงจรที่ใช้สำหรับแยกกราวน์ระหว่างพอร์ตไมโครคอนโทรลเลอร์และวงจรขับมอเตอร์โดยใช้ไอซี L298 เพื่อป้องกันกระแสของวงจรขับมอเตอร์เข้าไปยังพอร์ตและอาจทำให้พอร์ตเสียหายได้ โดยรับอินพุทไบนารี่จากพอร์ตไมโครคอนโทรลเลอร์ 6 บิต เอาท์พุทจะออกมาเหมือนกับอินพุทที่เข้าไป



รูปที่ 3.12 แสดงวงจรโฟโต้คอปเปอร์ LTV 847 และ PC 817

3.7 วงจรขับมอเตอร์โดยใช้ไอซี L298

หลักการทำงานของวงจร L298 ต้องทำการป้อนไฟเลี้ยง VSS 5 โวลต์ และ VS เป็นโวลเตจที่มาจากวงจรโฟโต้คอปเปอร์ LTV 847 และ PC 817 เมื่อทำการป้อนสัญญาณอินพุทที่ออกมาจากวงจรโฟโต้คอปเปอร์ LTV 847 และ PC 817 ซึ่งจะได้สัญญาณไบนารี่ 6 บิต โดยป้อนสัญญาณดังตารางด้านล่าง เมื่อนำมอเตอร์ต่อที่ขาเอาท์พุทและทำการป้อนสัญญาณก็จะสามารถขับมอเตอร์ให้หมุนไปตามทิศทางที่ต้องการ

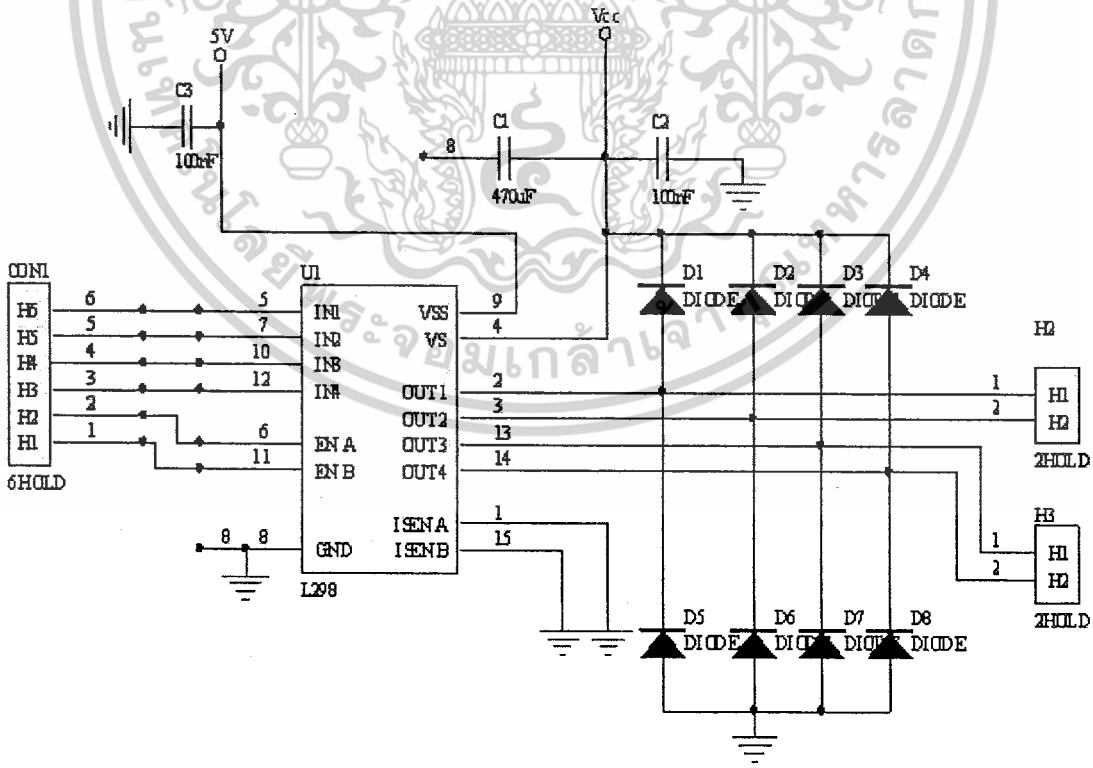
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 การป้อนสัญญาณไบนารี่เพื่อขับมอเตอร์ด้วยไอซี L298 แบบฟอร์เวิร์ส

ENB	ENA	IN4	IN3	IN2	IN1
1	1	1	1	0	0
1	1	0	1	1	0
1	1	0	0	1	1
1	1	1	0	0	1

ตารางที่ 3.3 การป้อนสัญญาณไบนารี่เพื่อขับมอเตอร์ด้วยไอซี L298 แบบรีเวิร์ส

ENB	ENA	IN4	IN3	IN2	IN1
1	1	1	0	0	1
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	1	0	0

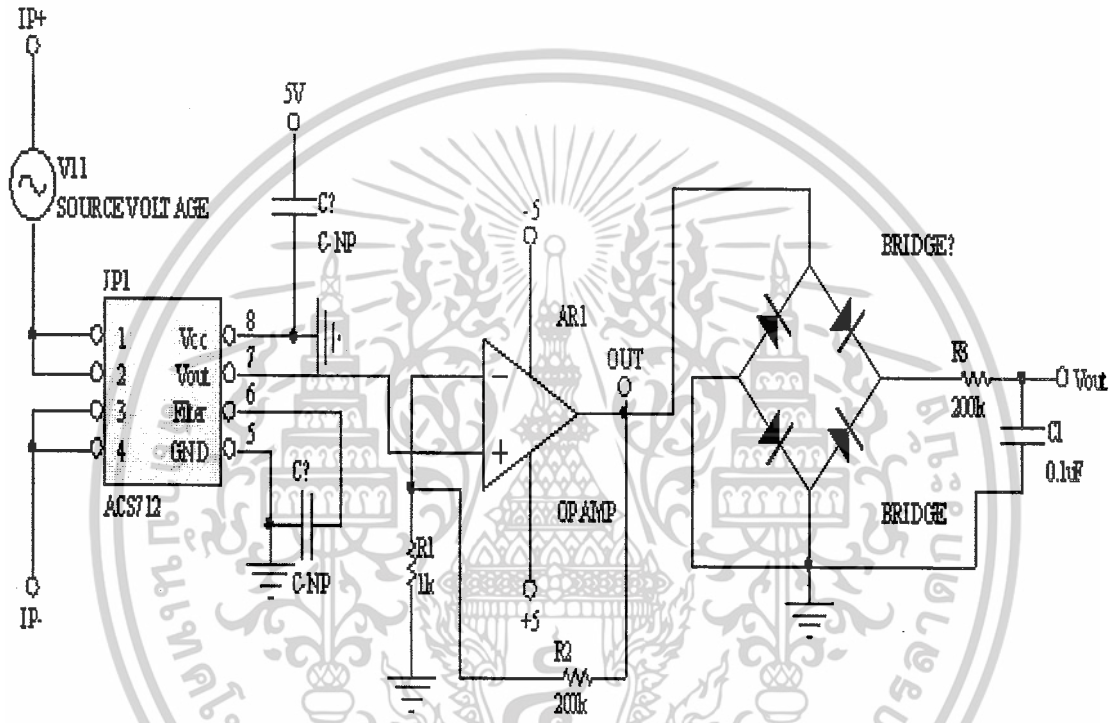


รูปที่ 3.13 แสดงวงจรขับมอเตอร์โดยใช้ไอซี L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 วงจรตรวจจับกระแส

การทำงานของวงจรตรวจจับกระแส วงจรนี้จะต่ออนุกรมกับหลอดไฟเพื่อนำกระแสมาเป็นตัวขับการทำงาน กระแสจะไหลผ่านไอซี ACS712 ที่ต้องมีไฟเลี้ยง 5 โวลต์ เอาท์พุทที่ได้จะเป็นโวลเตจ 200 มิลลิโวลต์โดยถ้าไม่มีกระแสจะไม่มีเอาท์พุทออกมา นำเอาท์พุทที่ได้ผ่านวงจรขยายโดยใช้ออปแอมป์เพื่อขยายโวลเตจขึ้นจะได้เอาท์พุทออกมาประมาณ 1 โวลต์ นำไปผ่านวงจรบริดจ์เพื่อแปลงจากโวลเตจกระแสสลับเป็นโวลเตจกระแสตรงและผ่านวงจรกรองความถี่สูงได้เอาท์พุทประมาณ 3.5โวลต์ เพื่อนำไปใช้กับพอร์ตไมโครคอนโทรลเลอร์

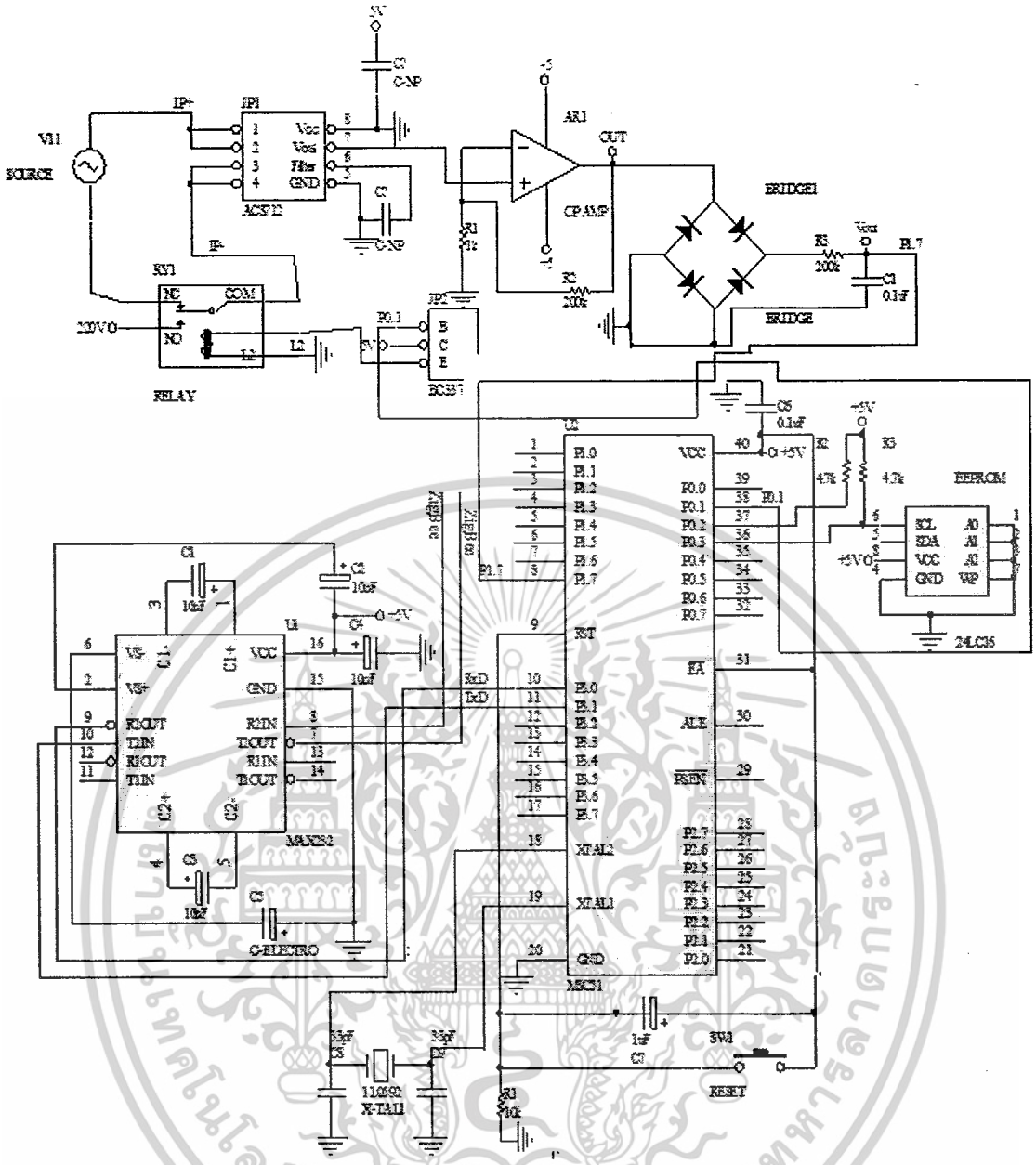


รูปที่ 3.14 แสดงวงจรตรวจจับกระแส

3.9 วงจรควบคุมการทำงานของหลอดไฟ

การทำงานของวงจรนี้คือ เมื่อไมโครคอนโทรลเลอร์ได้รับคำสั่งการทำงานของหลอดไฟแล้วผ่านทางขา Tx ของ MAX 232 ที่เชื่อมต่อขา Tx ของไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์จะทำการประมวลผลและควบคุมการทำงานของหลอดไฟทางพอร์ต P0.1 เมื่อผู้ใช้สั่งให้หลอดไฟติดพอร์ต P0.1 จะป้อนแรงดันเป็น 1 (5 โวลต์) ทำให้ทรานซิสเตอร์ขับกระแสไปกระตุ้นให้รีเลย์ทำงาน หน้าคอนแทก NO จะทำงานทำให้หลอดไฟติดซึ่งหลอดไฟจะเชื่อมต่อกับวงจรตรวจจับกระแสต่อไป แต่ถ้าผู้ใช้งานต้องการปิดหลอดไฟพอร์ต P0.1 จะเป็น 0 (0โวลต์) ทำให้ทรานซิสเตอร์ไม่ขับกระแสไปกระตุ้นให้รีเลย์ทำงาน หน้าคอนจะเป็น NC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



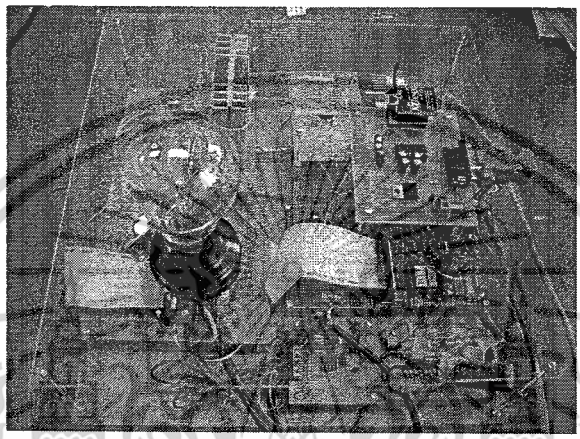
รูปที่ 3.15 แสดงวงจรวงจรควบคุมการทำงานของหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

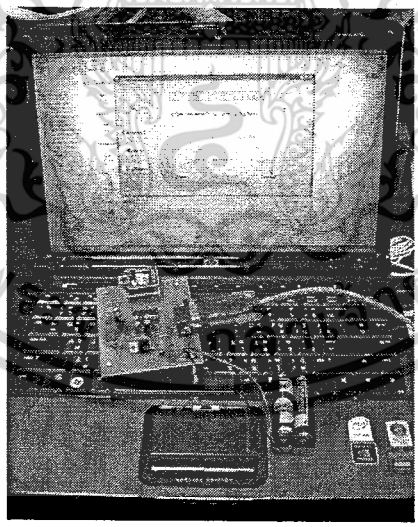
บทที่ 4
ผลการทดลองและวิเคราะห์ผลการทดลอง

4.1 การทดลองการส่งข้อมูลสถานะของอุปกรณ์จากฝั่งไมโครคอนโทรลเลอร์ไปยังฝั่งคอมพิวเตอร์

ในการทดลองนี้ มีวัตถุประสงค์เพื่อทดสอบการทำงานของโปรแกรมในการตรวจสอบสถานะอุปกรณ์ มีการรับ-ส่ง ข้อมูลระหว่างฝั่งรับและฝั่งส่ง แล้วนำค่ามาแสดงบนหน้าจอฝั่งคอมพิวเตอร์ ซึ่งมีอุปกรณ์ดังรูปที่ 4.1 และ 4.2



รูปที่ 4.1 แสดงอุปกรณ์ของฝั่งไมโครคอนโทรลเลอร์



รูปที่ 4.2 แสดงอุปกรณ์ของฝั่งคอมพิวเตอร์

4.1.1 ฝั่งคอมพิวเตอร์

เริ่มต้นเปิดโปรแกรม ฝั่งคอมพิวเตอร์จะทำการส่งอักขระ “R” ไปยังฝั่งไมโครคอนโทรลเลอร์เพื่อเรียกสถานะในปัจจุบันของอุปกรณ์ไฟฟ้า สถานะจะถูกแสดงบนหน้าต่างของโปรแกรมซึ่งมี 2 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือหน้าControl lights และChooseangleหลังจากนั้นเมื่อผู้ใช้เลือกการใช้งานควบคุมอุปกรณ์คอมพิวเตอร์ จะทำการส่งคำสั่งไปยังฝั่งไมโครคอนโทรลเลอร์ดังตารางที่ 4.1 ผ่านทางโมดูล Zigbee

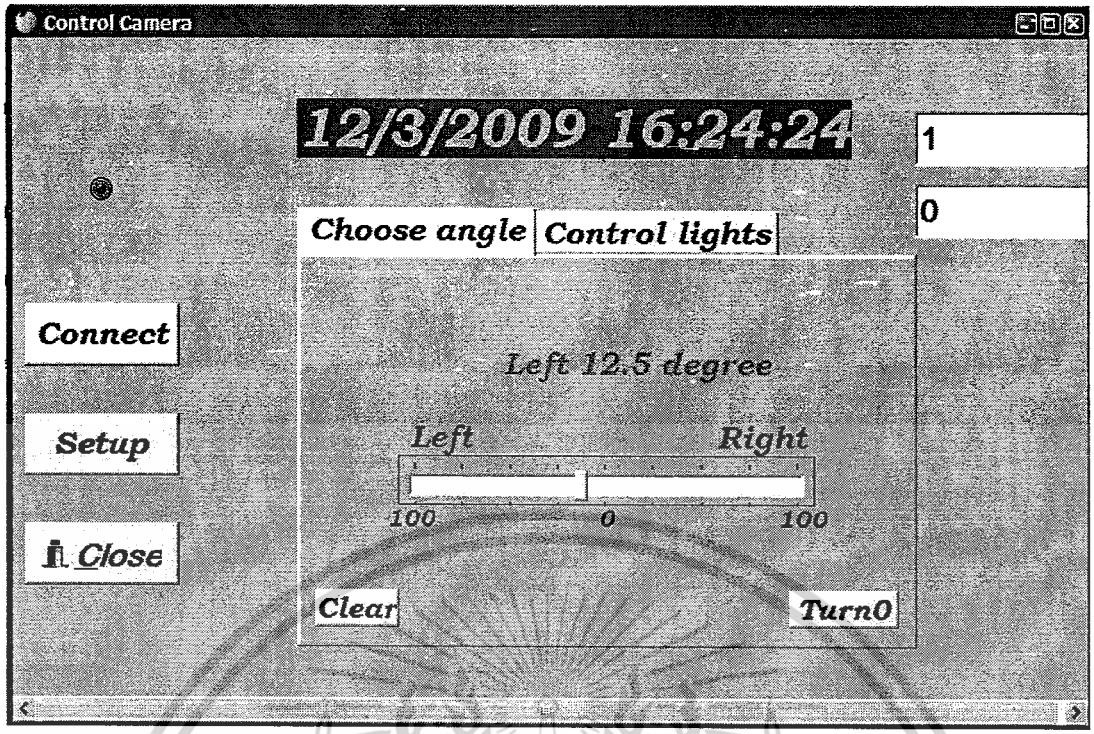
4.1.2 ฝั่งไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์หลังจากได้รับคำสั่งจากฝั่งคอมพิวเตอร์และนำคำสั่งเข้ามายังพอร์ตอนุกรม ไมโครคอนโทรลเลอร์จะตรวจสอบเงื่อนไขการทำงานดังตารางที่ 4.1 และทำงานตามเงื่อนไขนั้น

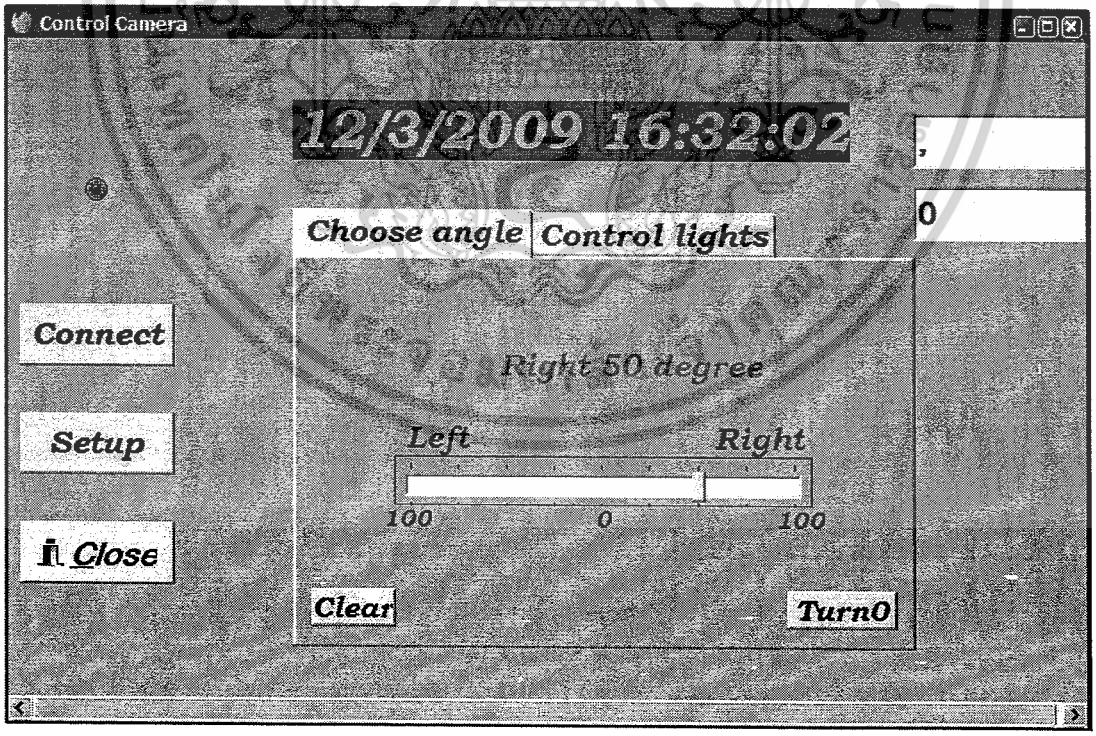
ตารางที่ 4.1 คำสั่งที่ใช้ในการตรวจสอบเงื่อนไขการทำงานและการทำงานของมอเตอร์

คำสั่ง	เงื่อนไขการทำงาน		
	รอบการหมุน	องศา	ทิศทางการหมุน
0	0	0°	
1	1	12.5°	ฟอร์เวิร์ส
2	2	25°	ฟอร์เวิร์ส
3	3	37.5°	ฟอร์เวิร์ส
4	4	50°	ฟอร์เวิร์ส
5	5	62.5°	ฟอร์เวิร์ส
6	6	75°	ฟอร์เวิร์ส
7	7	87.5	ฟอร์เวิร์ส
8	8	100°	ฟอร์เวิร์ส
A	1	12.5°	รีเวิร์ส
B	2	25°	รีเวิร์ส
C	3	37.5°	รีเวิร์ส
D	4	50°	รีเวิร์ส
E	5	62.5°	รีเวิร์ส
F	6	75°	รีเวิร์ส
G	7	87.5	รีเวิร์ส
H	8	100°	รีเวิร์ส
a	กลับสถานะหลอดไฟ		
Z	ลบค่ามุมของมอเตอร์ที่เก็บอยู่ในรีจิสเตอร์ของอีพรอม		
Y	หมุนมอเตอร์ให้กลับไปที่ตำแหน่ง 0°		
R	อ่านค่าสถานะของมอเตอร์และหลอดไฟ		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

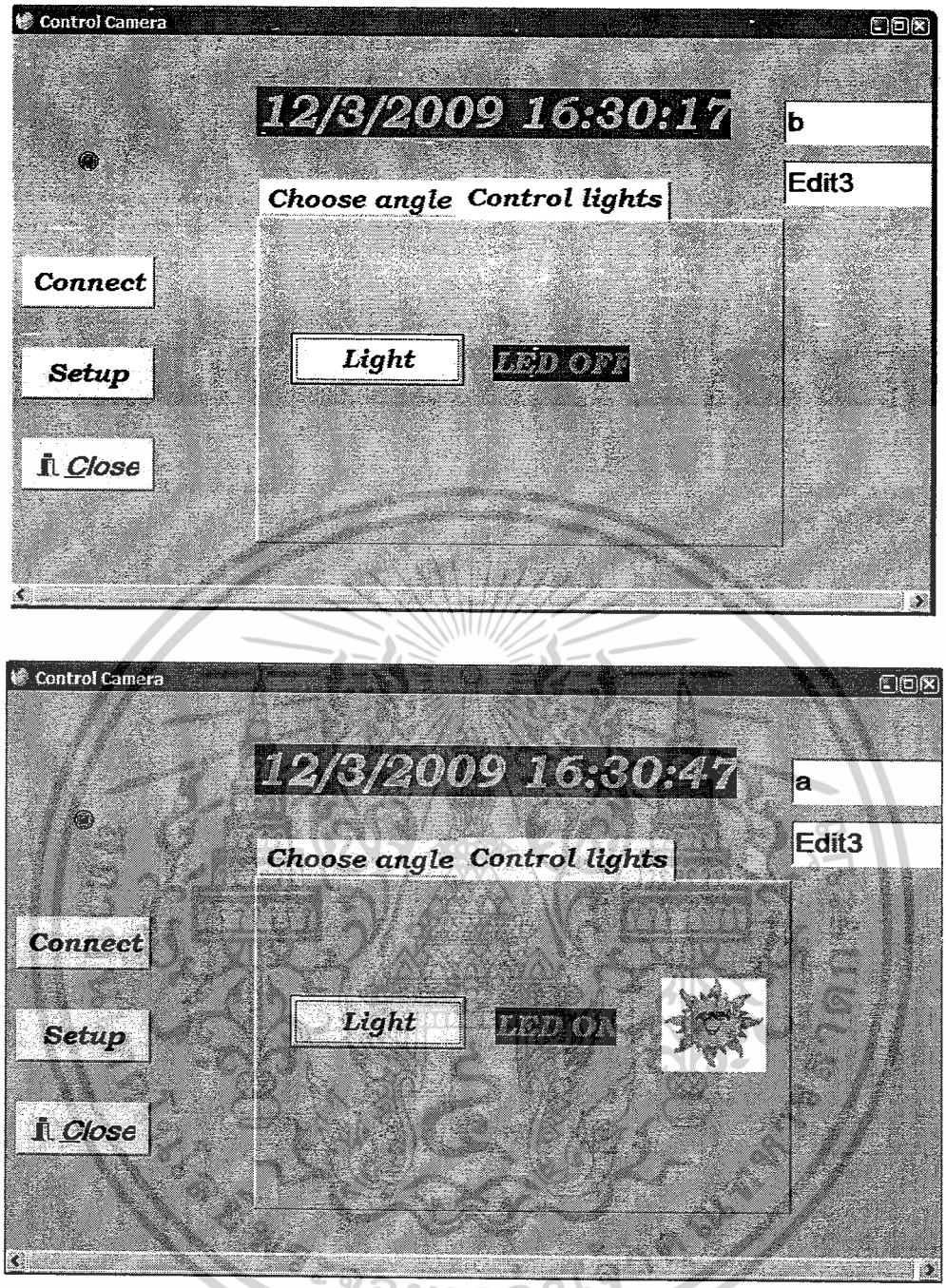


รูปที่ 4.3 แสดงสถานะองศาของมอเตอร์เมื่อไมโครคอนโทรลเลอร์ส่งค่าอักขระ “1”



รูปที่ 4.4 แสดงสถานะองศาของมอเตอร์เมื่อไมโครคอนโทรลเลอร์ส่งค่าอักขระ “,”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

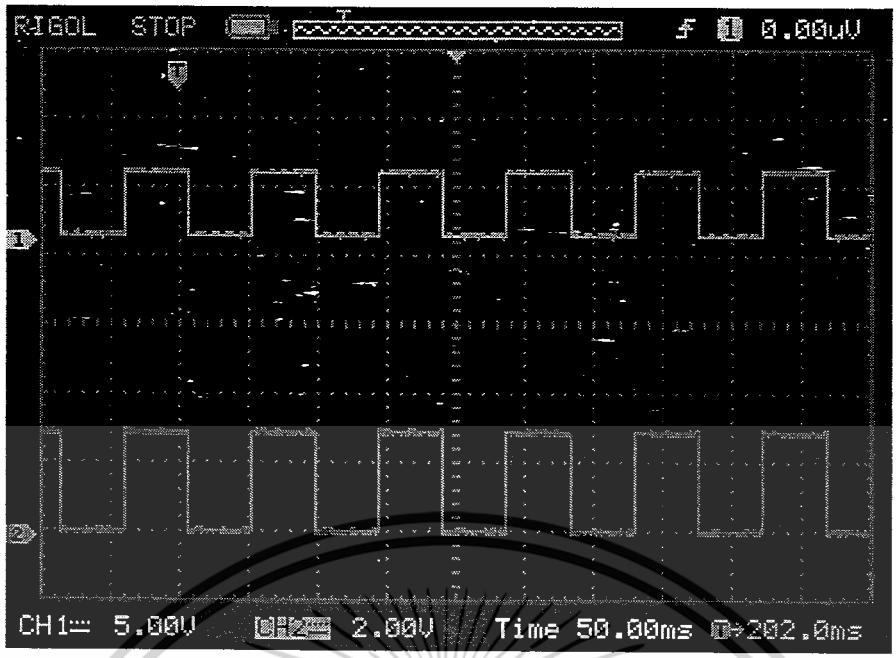


รูปที่ 4.5 แสดงสถานะการทำงานของหลอดไฟเมื่อไมโครคอนโทรลเลอร์ส่งค่าอักขระ “b” และ “a”

4.2 การทดลองวัดการป้อนสัญญาณไบนารี่ 4 บิตเพื่อขับมอเตอร์

เมื่อส่งสัญญาณไบนารี่ 4 บิตออกจากไมโครอนโทรลเลอร์เข้าขาของโฟโต้คอปเปอร์ LTV 847 และ PC 817 แล้วทำการจับสัญญาณที่ขาอินพุทและเอาท์พุทจะได้สัญญาณ M1, M2, M3, M4 ตามลำดับ โดยขา En1 และ En2 ต้องป้อนไฟเลี้ยง 5 โวลต์ แสดงกราฟรูปที่ 4.4 – 4.7 เป็นการป้อนอินพุทแบบฟอร์เวิร์สเพื่อให้มอเตอร์หมุนไปข้างหน้าโดยป้อนเรียงจากภาพ (4.4) - (4.5) - (4.6) - (4.7) แต่ถ้าเป็นการหมุนแบบรีเวิร์สเพื่อให้มอเตอร์หมุนไปด้านตรงข้ามจะป้อนเรียงจากภาพ (4.7) - (4.6) - (4.5) - (4.4)

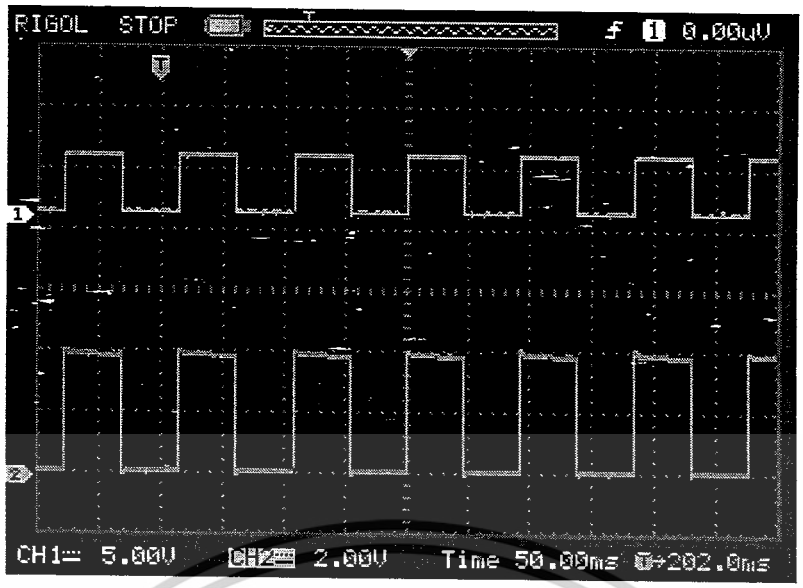
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



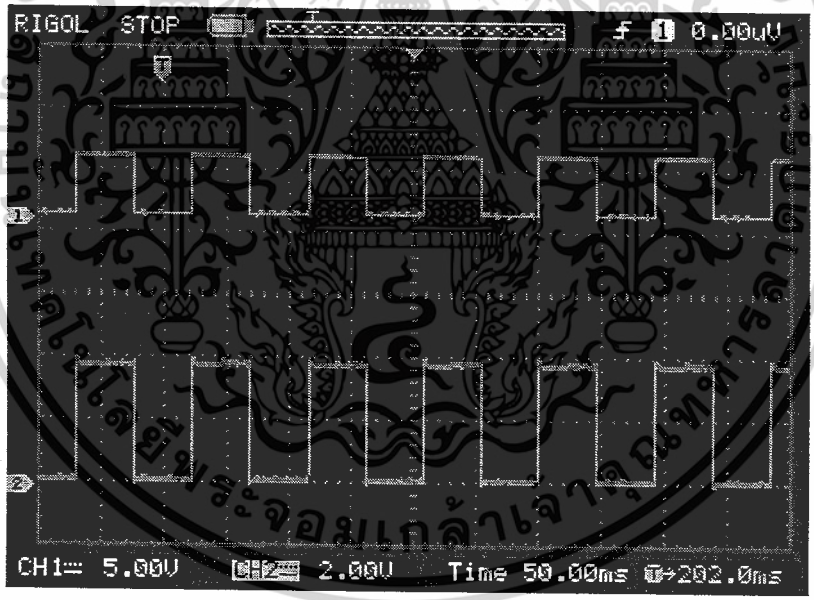
รูปที่ 4.6 แสดงสัญญาณ M1 ที่ขาอินพุทและเอาท์พุทของโฟโต้คอปเปอร์ LTV 847 และ PC 817
(Chanel 1) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 2
(Chanel 2) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 16

รูปที่ 4.7 แสดงสัญญาณ M1ที่ขาอินพุทและเอาท์พุทของโฟโต้คอปเปอร์LTV 847 และ PC 817
(Chanel 1) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 4
(Chanel 2) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงสัญญาณ M1 ที่ขาอินพุทและเอาท์พุทของโฟโต้คอปเปอร์ LTV 847 และ PC 817
(Chanel 1) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 6
(Chanel 2) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 12

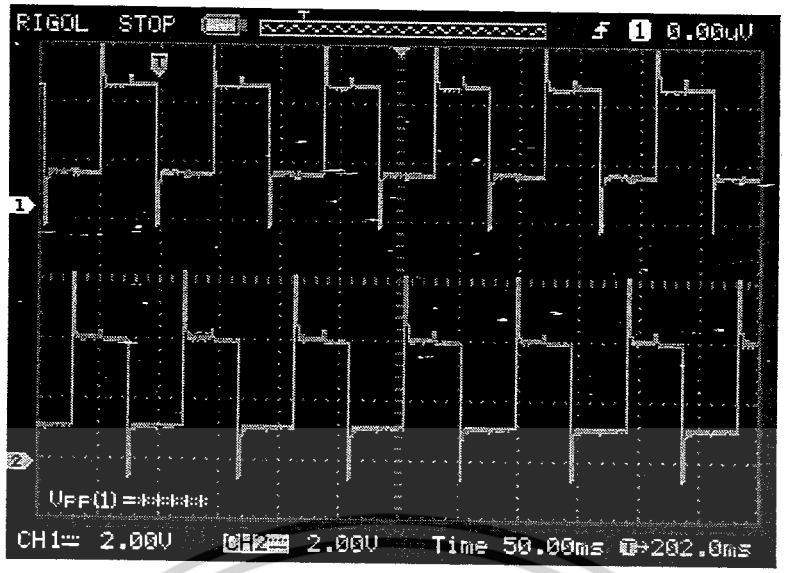


รูปที่ 4.9 แสดงสัญญาณ M1 ที่ขาอินพุทและเอาท์พุทของโฟโต้คอปเปอร์ LTV 847 และ PC 817
(Chanel 1) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 8
(Chanel 2) แสดงลักษณะของสัญญาณขณะทำการวัดที่ขา 10

4.3 การทดลองวัดสัญญาณเอาท์พุทของ L298

เมื่อส่งสัญญาณไบนารี่ 4 บิตจากโฟโต้คอปเปอร์ LTV 847 และ PC 817 เข้าขาของ L298 แล้ว ทำการจับสัญญาณที่ออกมาจะได้สัญญาณ M1, M2, M3, M4 ตามลำดับ

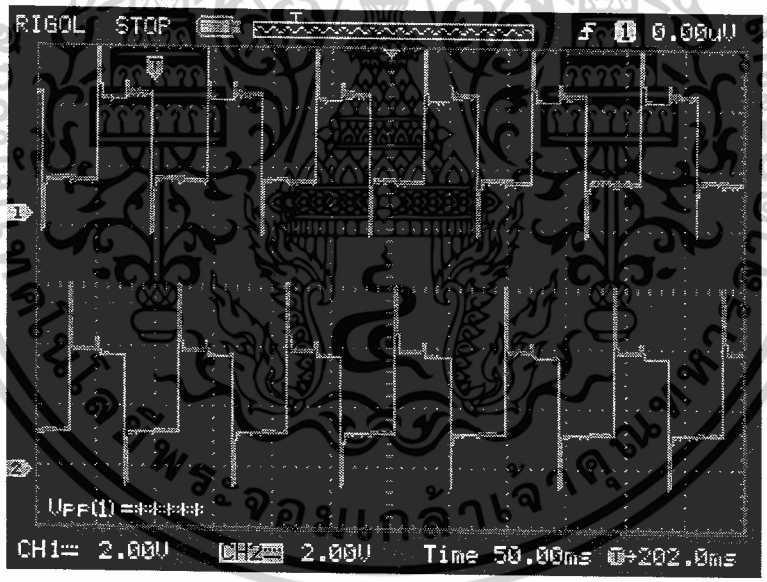
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงสัญญาณ M1 และ M2 ที่ขาเอาท์พุทของ L298

(Chanel 1) แสดงลักษณะของสัญญาณ M1

(Chanel 2) แสดงลักษณะของสัญญาณ M2



รูปที่ 4.11 แสดงสัญญาณ M3 และ M4 ที่ขาเอาท์พุทของ L298

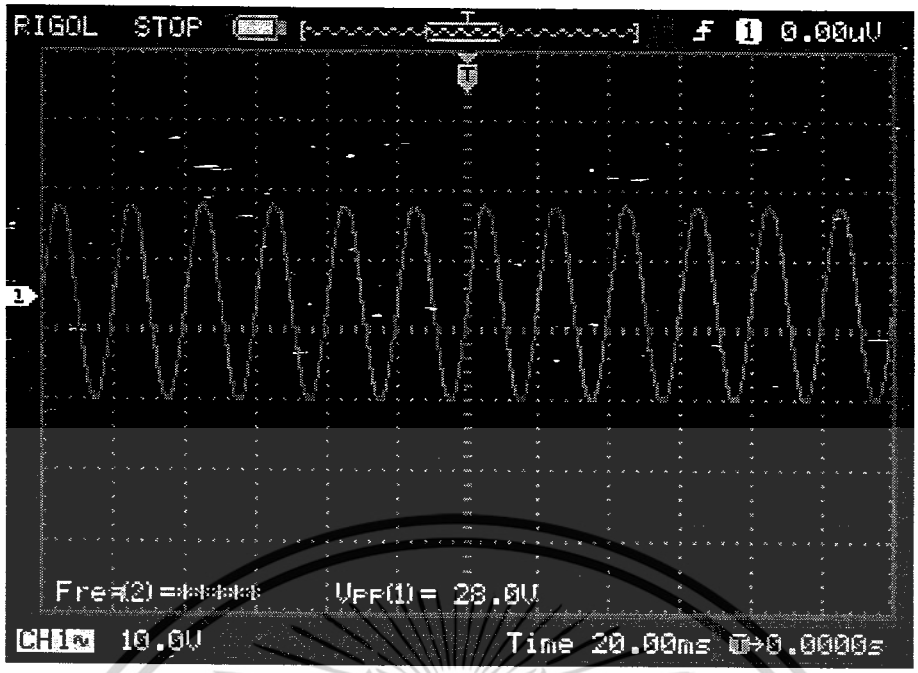
(Chanel 1) แสดงลักษณะของสัญญาณ M3

(Chanel 2) แสดงลักษณะของสัญญาณ M4

4.4 การทดลองวัดสัญญาณเอาท์พุทของวงจรแปลงไฟจากกระแสสลับ 9 โวลต์ เป็นกระแสตรง ± 5 โวลต์

จากการทดลองเมื่อทำการป้อนสัญญาณกระแสสลับ 9 โวลต์จากหม้อแปลงผ่านวงจรแปลงสัญญาณเป็นกระแสตรง ± 5 โวลต์ ทำการวัดขาเอาท์พุทที่ออกจากวงจรจะได้สัญญาณกระแสตรงเป็น ± 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงสัญญาณอินพุทกระแสสลับ 9 โวลต์

รูปที่ 4.13 แสดงสัญญาณเอาท์พุทกระแสตรง ± 5 โวลต์

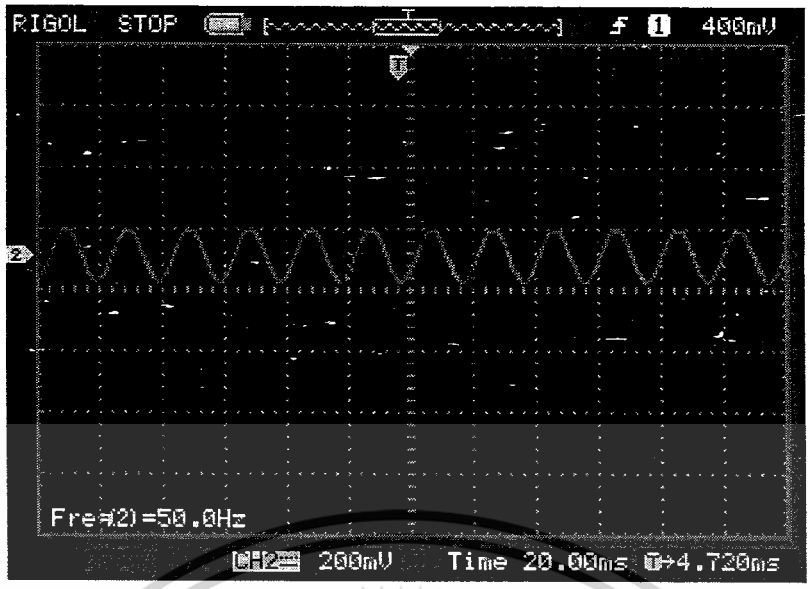
(Chanel 1) แสดงลักษณะของสัญญาณ +5 โวลต์

(Chanel 2) แสดงลักษณะของสัญญาณ -5 โวลต์

4.5 การทดลองวัดสัญญาณเอาท์พุทของวงจรตรวจจับกระแสโดยใช้ไอซี ACS712

จากการทดลองเมื่อทำการป้อนสัญญาณกระแสจากหลอดไฟโดยต่อแบบอนุกรมเข้าวงจรตรวจจับกระแสโดยใช้ไอซี ACS712 เมื่อมีกระแสเข้าวงจรจะได้เอาท์พุท 200 มิลลิโวลต์แต่ถ้าไม่มีกระแสเข้าวงจรจะได้ 0 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงสัญญาณเอาท์พุทของวงจรตรวจจับกระแสโดยใช้ไอซี ACS712

4.6 การทดลองวัดสัญญาณเอาท์พุทของวงจรขยายโดยใช้ออปแอมป์

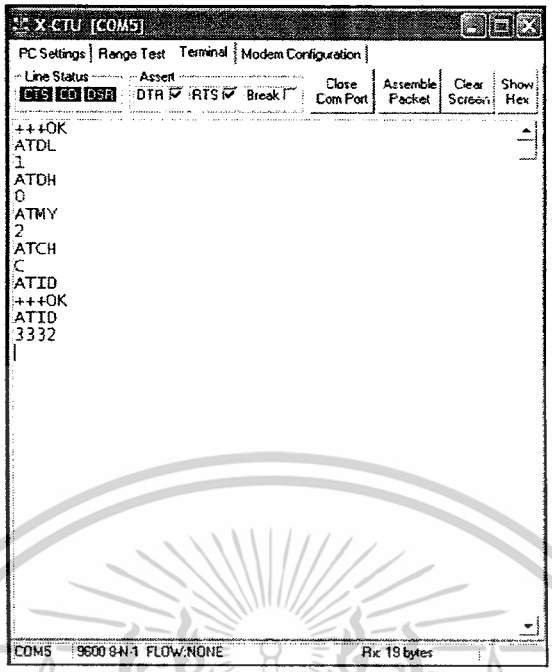
จากการทดลองเมื่อทำการป้อนสัญญาณ 200 มิลลิโวลต์ที่ออกจากวงจรตรวจจับกระแสโดยใช้ไอซี ACS712 ผ่านวงจรขยายโดยใช้ออปแอมป์จะขยายสัญญาณอินพุทจาก 200 มิลลิโวลต์เป็น 1 โวลต์

รูปที่ 4.15 แสดงสัญญาณเอาท์พุทของวงจรขยายโดยใช้ออปแอมป์

4.7 การทดลองติดต่อกับโมดูล Zigbee ด้วย AT Command ผ่านโปรแกรม X-CTU

ทำการทดลองติดต่อกับโมดูล Zigbee ด้วย AT Command โดยทำการเปิดโปรแกรม X-CTU หรือ Hyperterminal แล้วพิมพ์ AT Command ที่ต้องการ ซึ่งในการทดลองนี้จะอ่านค่าที่เคยกำหนดไว้เพื่อตรวจสอบว่ากำหนดค่าได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 แสดงการติดต่อโมดูล Zigbee โดยใช้ AT Command

4.8 การทดลองการส่ง-รับข้อมูลของโมดูล Zigbee

เมื่อทำการจับสัญญาณTxของฝั่งส่งและRxของฝั่งรับของโมดูลZigbee ที่ส่ง-รับข้อมูลกันด้วย Oscilloscope ได้ผลการทดลองแสดงดังรูป

รูปที่ 4.17 แสดงสัญญาณเปรียบเทียบกันระหว่างข้อมูลของฝั่งส่งกับข้อมูลของฝั่งรับ

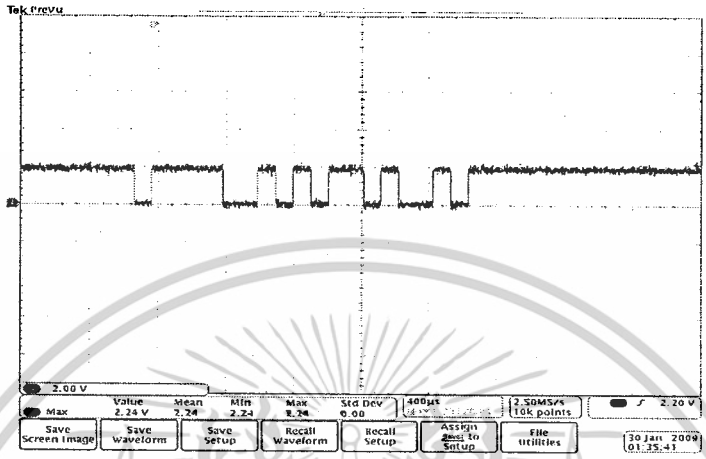
(Chanel 1) แสดงลักษณะของสัญญาณที่ CH1

(Chanel 2) แสดงลักษณะของสัญญาณที่ CH2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 การทดลองเปลี่ยนตำแหน่งที่อยู่ของ Zigbee

เมื่อทำการติดต่อกับ Zigbee ด้วยคำสั่ง +++ Zigbee จะตอบรับ OKกลับมา จากนั้นจึงส่งคำสั่งเพื่อเปลี่ยนตำแหน่งที่อยู่ของ Zigbee โดยใช้คำสั่ง ATDL แล้วตามด้วยตำแหน่งที่ต้องการเปลี่ยน เพื่อทำการทดสอบว่าโมดูล Zigbee สามารถรับคำสั่งที่ส่งไปได้จึงวัดสัญญาณตอบรับ OK ที่ขา Tx ของ Zigbee



รูปที่ 4.18 แสดงสัญญาณที่วัดจากขาTx ของ Zigbee ที่ส่ง OK กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและวิจารณ์การทดลอง

5.1 สรุปผลการทดลอง

โครงงานเรื่องระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน ZIGBEE จากการทดลองในแต่ละส่วนของระบบพบว่าระบบสามารถใช้งานได้ดังนี้

5.1.1 ฝั่งคอมพิวเตอร์

ผู้ใช้งานสามารถควบคุมอุปกรณ์ไฟฟ้าคือ มอเตอร์และหลอดไฟได้ โดยสามารถควบคุมการหมุนมอเตอร์ตามองศาต่างๆ และควบคุมการทำงานของหลอดไฟพร้อมแสดงสถานะปัจจุบันของอุปกรณ์ไฟฟ้า

5.1.2 ฝั่งไมโครคอนโทรลเลอร์

สามารถควบคุมการทำงานของอุปกรณ์ไฟฟ้าได้โดยใช้ไมโครคอนโทรลเลอร์ควบคุม คือ หลังจากที่ไมโครคอนโทรลเลอร์รับคำสั่งจากฝั่งคอมพิวเตอร์แล้วสามารถควบคุมการทิศทางการหมุนของมอเตอร์และการทำงานของหลอดไฟได้อย่างถูกต้องพร้อมทั้งส่งค่ากลับไปยังฝั่งคอมพิวเตอร์เพื่อแสดงสถานะของอุปกรณ์ไฟฟ้า

5.1.3 การเชื่อมต่อระหว่างฝั่งคอมพิวเตอร์และฝั่งไมโครคอนโทรลเลอร์โดยผ่านโมดูล Zigbee

Zigbee ทั้งสองตัวที่ต่อระหว่างฝั่งคอมพิวเตอร์และฝั่งไมโครคอนโทรลเลอร์สามารถรับส่งข้อมูลจากฝั่งคอมพิวเตอร์, ฝั่งไมโครคอนโทรลเลอร์และรับส่งข้อมูลระหว่างกันได้

5.2 วิจารณ์การทดลอง

ในการทดลองนี้พบว่าการควบคุมทิศทางการหมุนของมอเตอร์โดยตัวขับมอเตอร์หรือไอซี L298 เมื่อขับมอเตอร์หมุนในเวลานานไอซีจะร้อนขึ้น และการใช้หรีดสวิช์ตรวจสอบการหมุนของมอเตอร์ต้องใช้แม่เหล็กที่มีแม่เหล็กแรงพอสมควร

5.3 แนวทางการพัฒนา

โครงงานนี้ต้องปรับปรุงระบบให้สามารถรองรับการใช้งานอื่น ๆ ได้มากขึ้นคือนำไปใช้ควบคุมอุปกรณ์ไฟฟ้าได้หลายชิ้นมากขึ้น และปรับปรุงให้วงจรมีขนาดเล็กลงเพื่อสะดวกในการนำไปใช้งานจริง รวมทั้งปรับปรุงให้ระยะทางในการควบคุมให้มากขึ้นด้วยโดยการเพิ่มตัวรีพีสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] สมยศ จุณณะปิยะ,*การประยุกต์ใช้งานไมโครคอนโทรลเลอร์*,พิมพ์ครั้งที่ 5,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ,2546

[2] สัจจะ จรัสรุ่งรวีวร,จักรพงษ์ สุขประเสริฐ, *เริ่มต้นอย่างมืออาชีพด้วย Delphi 7 ฉบับสมบูรณ์*, พิมพ์ครั้งที่ 1,อินโฟเพรส,2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการทดลองในส่วนของฝั่งคอมพิวเตอร์

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, CPortCtl, CPort,ExtCtrls, jpeg, ComCtrls, Buttons,
  TeeProcs, TeEngine, Chart;
type
  TForm1 = class(TForm)
    ComPort1: TComPort;
    ComLed1: TComLed;
    Button4: TButton;
    Timer1: TTimer;
    lblDateTime: TLabel;
    PageControl1: TPageControl;
    TabSheet2: TTabSheet;
    TrackBar1: TTrackBar;
    BitBtn1: TBitBtn;
    connect: TButton;
    Label6: TLabel;
    Label8: TLabel;
    Label12: TLabel;
    Label14: TLabel;
    Edit1: TEdit;
    Edit3: TEdit;
    Label1: TLabel;
    TabSheet3: TTabSheet;
    Button1: TButton;
    Label2: TLabel;
    Image4: TImage;
    Label3: TLabel;
    Label4: TLabel;
    Button2: TButton;
    Button3: TButton;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  procedure FormCreate(Sender: TObject);
  procedure Comport1RxChar(Sender: TObject; Count: Integer);
  procedure Button4Click(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure ComPort1AfterOpen(Sender: TObject);
  procedure connectClick(Sender: TObject);
  procedure TrackBar1Change(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);

 private
  { Private declarations }
 public
  { Public declarations }
 end;
var  cam : string;
var cam1 : integer;
var intRate : integer;
var intRate1 : integer;
 var haha : integer;
 Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject);
begin
comport1.WriteStr('R');
end;
//////////////////////RECEIVE/////////////////////////////////////
procedure TForm1.Comport1RxChar(Sender: TObject; Count: Integer);
 var  cam : string;
 begin
 comport1.ReadStr(cam,count);
 edit1.Text := cam;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
comport1.Read(haha,count) ;
if cam = '0' then
begin
label14.Caption := '0 degree' ;
TrackBar1.Position := 8;
end;
//////////////////////FORWARD/////////////////////////////////////
if cam = '1' then
begin
label14.Caption := 'Left 12.5 degree';
end;
 if cam = '2' then
begin
label14.Caption := 'Left 25 degree';
end;
 if cam = '35' then
begin
label14.Caption := 'Left 37.5 degree';
end;
 if cam = '4' then
begin
label14.Caption := 'Left 50 degree';
end;
 if cam = '5' then
begin
label14.Caption := 'Left 62.5 degree';
end;
 if cam = '6' then
begin
label14.Caption := 'Left 75 degree';
end;
 if cam = '7' then
begin
label14.Caption := 'Left 87.5 degree';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
 if cam = '8' then
begin
label14.Caption := 'Left 100 degree';
end;

//////////////////////REVERSE////////////////////////////////
 if cam = '/' then
begin
label14.Caption := 'Right 12.5 degree';
end;
 if cam = '.' then
begin
label14.Caption := 'Right 25 degree';
end;
 if cam = '-' then
begin
label14.Caption := 'Right 37.5 degree';
end;
 if cam = ',' then
begin
label14.Caption := 'Right 50 degree';
end;
 if cam = '+' then
begin
label14.Caption := 'Right 62.5 degree';
end;
 if cam = '*' then
begin
label14.Caption := 'Right 75 degree';
end;
 if cam = ')' then
begin
label14.Caption := 'Right 87.5 degree';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
     if cam = '(' then
    begin
    label14.Caption := 'Right 100 degree';
    end;
     if haha = $27 then
    begin
    label14.Caption := 'Right 112.5 degree';
    end;
     if cam = '&' then
    begin
    label14.Caption := 'Right 125 degree';
    end;

 ////////////////////Control Lights/////////////////////////////////
if  cam = 'a' then
  begin
       Label2.Caption := 'LED ON';
       Image4.Visible:=True;
      end;
  if  cam = 'b'  then
  begin
       Label2.Caption := 'LED OFF';
       Image4.Visible:=False;
      end;
 end;
procedure TForm1.Button4Click(Sender: TObject);
begin
comport1.ShowSetupDialog;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
 lblDateTime.Caption := DateTimeToStr(Now);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.ComPort1AfterOpen(Sender: TObject);
begin
 comport1.WriteStr('R');
end;
procedure TForm1.connectClick(Sender: TObject);
begin
    comport1.Connected:=not comport1.Connected;
    TrackBar1.Enabled:=comport1.Connected;
    Button1.Enabled:=comport1.Connected;
    Button2.Enabled:=comport1.Connected;
end;
//////////////////////WRITE//////////////////////////
procedure TForm1.TrackBar1Change(Sender: TObject);
var qq : integer;
var Out : integer;
var intR : integer;
begin
qq := 0;
     intRate := TrackBar1.Position ;
      if TrackBar1.Position = 8 then
      begin
          cam1 := intRate;
          end;
      if intRate <= 8  then
      begin
            Out :=  cam1 - intRate;
            if   Out = 0   then
     begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('0');
            Out := 0    ;
            edit3.Text:= inttostr(Out);
  repeat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
 qq := qq+1;
end;
until qq = 80;
      end;
  if  Out = 1  then
  begin
         intR := intRate;
         cam1 := intR;
         comport1.WriteStr('1');
         Out := 1   ;
         edit3.Text:= inttostr(Out);
 repeat
begin
qq := qq+1;
end;
until qq = 80;
     end;
   if  Out = 2  then
     begin
         intR := intRate;
         cam1 := intR;
         comport1.WriteStr('2');
         Out := 2;
         edit3.Text:= inttostr(Out);
 repeat
begin
 qq := qq+1;
end;
until qq = 80;
      end;
                if  Out = 3  then
     begin
         intR := intRate;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            cam1 := intR;
            comport1.WriteStr('3');
            Out := 3;
            edit3.Text:= inttostr(Out);
             repeat
 begin
qq := qq+1;
 end;
 until qq = 80;
 end;
                    if   Out = 4   then
       begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('4');
            Out := 4;
            edit3.Text:= inttostr(Out);
             repeat
 begin
qq := qq+1;
end;
until qq = 80;
        end;
                    if   Out = 5   then
      begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('5');
            Out := 5;
            edit3.Text:= inttostr(Out);
repeat
 begin
 qq := qq+1;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
until qq = 80;
        end;
                if  Out = 6  then
       begin
           intR := intRate;
           cam1 := intR;
           comport1.WriteStr('6');
           Out := 6;
           edit3.Text:= inttostr(Out);
            repeat
begin
  qq := qq+1;
end;
until qq = 80;
        end;
                if  Out = 7  then
       begin
           intR := intRate;
           cam1 := intR;
           comport1.WriteStr('7');
           Out := 7;
           edit3.Text:= inttostr(Out);
repeat
begin
  qq := qq+1;
end;
until qq = 80;
        end;
                if  Out = 8  then
       begin
           intR := intRate;
           cam1 := intR;
           comport1.WriteStr('8');
           Out := 8;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            edit3.Text:= inttostr(Out);
repeat
 begin
 qq := qq+1;
 end;
 until qq = 80;
        end;
                    if   Out = -1   then
       begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('A');
            Out := -1;
            edit3.Text:= inttostr(Out);
             repeat
 begin
qq := qq+1;
 end;
 until qq = 80;
        end;
                    if   Out = -2   then
       begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('B');
            Out := -2;
            edit3.Text:= inttostr(Out);
 repeat
 begin
  qq := qq+1;
 end;
 until qq = 80;
        end;
                    if   Out = -3   then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('C');
            Out := -3;
            edit3.Text:= inttostr(Out);
repeat
begin
qq := qq+1;
end;
until qq = 80;
        end;
                if  Out = -4  then
        begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('D');
            Out := -4;
            edit3.Text:= inttostr(Out);
            repeat
begin
qq := qq+1;
end;
until qq = 80;
        end;
                if  Out = -5  then
        begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('E');
            Out := -5  ;
            edit3.Text:= inttostr(Out);
            repeat
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  qq := qq+1;
 end;
 until qq = 80;
        end;
                if  Out = -6  then
       begin
          intR := intRate;
          cam1 := intR;
         comport1.WriteStr('F');
          Out := -6;
          edit3.Text:= inttostr(Out);
repeat
 begin
 qq := qq+1;
 end;
 until qq = 80;
       end;

                if  Out = -7  then
      begin
          intR := intRate;
          cam1 := intR;
          comport1.WriteStr('G');
          Out := -7;
          edit3.Text:= inttostr(Out);
           repeat
begin
  qq := qq+1;
end;
until qq = 80;
       end;
                if  Out = -8  then
     begin
         intR := intRate;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            cam1 := intR;
            comport1.WriteStr('H');
            Out := -8;
            edit3.Text:= inttostr(Out);
repeat
begin
qq := qq+1;
end;
until qq = 80;
      end;
                  if intRate >= 9 then
    begin
        Out := cam1 - intRate;
    if  Out = 0   then
  begin
        intR := intRate;
        cam1 := intR;
        comport1.WriteStr('0');
        Out := 0   ;
        edit3.Text:= inttostr(Out);
repeat
begin
qq := qq+1;
end;
until qq = 80;
      end;
                if  Out = 1   then
 begin
        intR := intRate;
        cam1 := intR;
        comport1.WriteStr('1');
        Out := 1   ;
        edit3.Text:= inttostr(Out);
         repeat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
qq := qq+1;
end;
until qq = 80;
      end;
                if  Out = 2  then
     begin
        intR := intRate;
        cam1 := intR;
        comport1.WriteStr('2');
        Out := 2;
        edit3.Text:= inttostr(Out);
repeat
begin
qq := qq+1;
end;
until qq = 80;
      end;

                      if  Out = 3  then
     begin
        intR := intRate;
        cam1 := intR;
        comport1.WriteStr('3');
        Out := 3;
        edit3.Text:= inttostr(Out);
         repeat
begin
 qq := qq+1;
end;
until qq = 80;
      end;
                if  Out = 4  then
     begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('4');
            Out := 4;
            edit3.Text:= inttostr(Out);
repeat
begin
  qq := qq+1;
end;
until qq = 80;
        end;
                    if   Out = 5   then
        begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('5');
            Out := 5;
            edit3.Text:= inttostr(Out);
   repeat
begin
  qq := qq+1;
end;
until qq = 80;
        end;
                    if   Out = 6   then
        begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('6');
            Out := 6;
            edit3.Text:= inttostr(Out);
     repeat
begin
  qq := qq+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
until qq = 80;
      end;
                if  Out = 7   then
     begin
         intR := intRate;
         cam1 := intR;
         comport1.WriteStr('7');
         Out := 7;
         edit3.Text:= inttostr(Out);
          repeat
begin
  qq := qq+1;
end;
until qq = 80;
      end;
                  if   Out = 8   then
     begin
         intR := intRate;
         cam1 := intR;
         comport1.WriteStr('8');
         Out := 8;
         edit3.Text:= inttostr(Out);
          repeat
begin
  qq := qq+1;
end;
until qq = 80;
      end;
                if   intRate1 = -1   then
     begin
         intR := intRate;
         cam1 := intR;
         comport1.WriteStr('A');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            intRate1 := -1;
            edit3.Text:= inttostr(intRate1);
            repeat
begin
  qq := qq+1;
end;
until qq = 80;
        end;
                    if   Out = -2   then
       begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('B');
            Out := -2;
            edit3.Text:= inttostr(Out);
repeat
begin
  qq := qq+1;
end;
until qq = 80;
        end;
                    if   Out = -3   then
       begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('C');
            Out := -3;
            edit3.Text:= inttostr(Out);
repeat
begin
  qq := qq+1;
end;
until qq = 80;
        end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if  Out = -4  then
        begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('D');
            Out := -4;
            edit3.Text:= inttostr(Out);
    repeat
    begin
     qq := qq+1;
    end;
    until qq = 80;
          end;
                if  Out = -5  then
        begin
            intR := intRate;
            cam1 := intR;
            comport1.WriteStr('E');
            Out := -5  ;
            edit3.Text:= inttostr(Out);
repeat
    begin
     qq := qq+1;
    end;
    until qq = 80;
          end;
                if  Out = -6  then
        begin
            intR := intRate;
            cam1 := intR;
           comport1.WriteStr('F');
            Out := -6;
            edit3.Text:= inttostr(Out);
    repeat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  begin
    qq := qq+1;
  end;
  until qq = 80;
        end;
                  if  Out = -7   then
       begin
           intR := intRate;
           cam1 := intR;
           comport1.WriteStr('G');
           Out := -7;
           edit3.Text:= inttostr(Out);
  repeat
  begin
    qq := qq+1;
  end;
  until qq = 80;
        end;
                  if  Out = -8   then
       begin
           intR := intRate;
           cam1 := intR;
           comport1.WriteStr('H');
           Out := -8;
           edit3.Text:= inttostr(Out);
  repeat
  begin
    qq := qq+1;
  end;
  until qq = 80;
        end;
end;
 end;
procedure TForm1.Button1Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
comport1.WriteStr('a');
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
comport1.WriteStr('Z');
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
 comport1.WriteStr('Y');
end;
end.
```

โปรแกรมที่ใช้ในการทดลองในส่วนของฝั่งไมโครคอนโทรลเลอร์

```
SDA         BIT     P0.2
SCL         BIT     P0.3
FLAG        EQU     02FH
I2C_ACK     BIT     FLAG.0
RX_ERR      BIT     FLAG.1
I2C_ADDR    EQU     034H
I2C_DATA    EQU     035H
EEP_AL      EQU     02CH
EEP_AH      EQU     02DH
EEP_DATA    EQU     02BH
BUFFER      EQU     036H
ANGLE       EQU     040H
AT24C512_ID EQU     10100000B
;*****************************************
        ORG 000H
        LJMP    MAIN
CKCON0      EQU     8FH
BRL         EQU     9AH
BDRCON      EQU     9BH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IEN0            EQU     0A8H
IPL0            EQU     0B8H
;----------   TWI SFRs ----------
SSCON       EQU    93H
SSCS        EQU    94H
SSDAT       EQU    95H
SSADR       EQU    96H
;*********************************************************************
;       DEFINE SECTION
;*********************************************************************
BUFFER_RX    EQU    70H
COUNT_RX     EQU    6FH
;*********************************************************************
        ORG    0023H
        LJMP   TxISR
        ORG    0030H
MAIN:   MOV    SP,#256-32          ; define stack = 32 byte
        MOV    P1,#0FFH
        MOV    P2,#00H
        ORL    CKCON0,#00000001B       ; 6 CLK PERIOD/MACHIN
        MOV    SP,#256-32              ; define stack = 32 byte
        ORL    PCON,#11000000B         ; ENABLE THE FRAMING BIT ERROR DETECTION,
        DOUBLE BAUD RATE
        MOV    IPL0,#00010000B         ; AND CHECK FE BIT OF SCON SFR
        ORL    BDRCON,#00001110B   ; SPD=1,RBCK =1, TBCK =1
        MOV    BRL,#100                ; U CAN SEE THIS VALUE FROM PAGE 71 OF
DATASHEET
        ORL    BDRCON,#00010000B   ; BRR = 1 , START BAUD RATE RUN
        SETB   TR1
        MOV    SCON,#01010000B         ; uart in mode 1 (8 bit), REN=1
        MOV    IEN0,#90H
        SJMP   $
;.....................................................
        SETB   SDA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SETB    SCL
;.............COMPARE.....................
CLEAR:CJNE    A,#5AH,READ_1
        MOV     A,#00H
        ACALL WRITE
        ACALL READ
        ACALL Tx
        RETI
;.....................................
;.............READ.....................
READ_1:         CJNE    A,#52H,LED
        ACALL READ
        ACALL Tx
        MOV     A,P0
        ANL     A,#00000001B
        CJNE    A,#00000001B,BB
        MOV     A,#61H
        ACALL REPLY
        NOP
        RETI
BB:     MOV     A,#62H
        ACALL REPLY
        NOP
        RETI
;.....................................
LED:    CJNE    A,#61H,APP
        CPL     P0.0
        MOV     A,P0
        ANL     A,#00000001B
        CJNE    A,#00000001B,BB1
        MOV     A,#61H
        ACALL REPLY
        NOP
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BB1:    MOV    A,#62H
        ACALL REPLY
        NOP
        RET
        RETI
;.....
APP:    CJNE   A,#59H,D
        ACALL READ
        ADD    A,#30H
;       ACALL Tx
;.....FORWARD COMPARE.....
        CJNE   A,#31H,APP2
        MOV    A,#41H
        ACALL ONE
        RETI
APP2:   CJNE   A,#32H,APP3
        MOV    A,#42H
        ACALL ONE
        RETI
APP3:   CJNE   A,#33H,APP4
        MOV    A,#43H
        ACALL ONE
        RETI
APP4:   CJNE   A,#34H,APP5
        MOV    A,#44H
        ACALL ONE
        RETI
APP5:   CJNE   A,#35H,APP6
        MOV    A,#45H
D:      ACALL ONE
        RETI
APP6:   CJNE   A,#36H,APP7
        MOV    A,#46H
        ACALL ONE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RETI

APP7:   CJNE    A,#37H,APP8
        MOV     A,#47H
        ACALL ONE
        RETI
APP8:   CJNE    A,#38H,APP9
        MOV     A,#48H
        ACALL ONE
        RETI
APP9:   CJNE    A,#39H,APP10
        MOV     A,#49H
        ACALL ONE
        RETI
APP10:  CJNE    A,#40H,APP01
        MOV     A,#50H
        ACALL ONE
        RETI
;.....REVERSE COMPARE.....
APP01:  CJNE    A,#2FH,APP02
        MOV     A,#31H
        ACALL ONE
        RETI
APP02:  CJNE    A,#2EH,APP03
        MOV     A,#32H
        ACALL ONE
        RETI
APP03:  CJNE    A,#2DH,APP04
        MOV     A,#33H
        ACALL ONE
        RETI
APP04:  CJNE    A,#2CH,APP05
        MOV     A,#34H
        ACALL ONE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RETI
APP05:  CJNE    A,#2BH,APP06
        MOV     A,#35H
        ACALL ONE
        RETI
APP06:  CJNE    A,#2AH,APP07
        MOV     A,#61H
        ACALL ONE
        RETI
APP07:  CJNE    A,#29H,APP08
        MOV     A,#37H
        ACALL ONE
        RETI
APP08:  CJNE    A,#28H,APP09
        MOV     A,#38H
        ACALL ONE
        RET
APP09:  CJNE    A,#27H,APP010
        MOV     A,#39H
        ACALL ONE
        RETI
APP010:CJNE     A,#26H,FF
        MOV     A,#40H
        ACALL ONE
FF:     RETI
;.....
ONE:    CJNE    A,#31H,TWO
        NOP
        MOV     A,#01
        ACALL FORWARD
        ACALL READ
        ADD     A,#01
        ACALL WRITE
        ACALL READ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
TWO:    CJNE    A,#32H,THREE
        NOP
        MOV     A,#02
        ACALL FORWARD
        ACALL READ
        ADD     A,#02
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
THREE: CJNE     A,#33H,FOUR
        NOP
        MOV     A,#03
        ACALL FORWARD
        ACALL READ
        ADD     A,#03
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
FOUR:  CJNE     A,#34H,FIVE
        NOP
        MOV     A,#04
        ACALL FORWARD
        ACALL READ
        ADD     A,#04
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
FIVE:   CJNE    A,#35H,SIX
        NOP
        MOV     A,#05
        ACALL FORWARD
        ACALL READ
        ADD     A,#05
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
SIX:    CJNE    A,#36H,SEVEN
        NOP
        MOV     A,#06
        ACALL FORWARD
        ACALL READ
        ADD     A,#06
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
SEVEN:  CJNE    A,#37H,EIGHT
        NOP
        MOV     A,#07
        ACALL FORWARD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACALL READ
        ADD     A,#07
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
EIGHT: CJNE     A,#38H,NINE
        NOP
        MOV     A,#08
        ACALL FORWARD
        ACALL READ
        ADD     A,#08
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
NINE:   CJNE    A,#39H,TEN
        NOP
        MOV     A,#09
        ACALL FORWARD
        ACALL READ
        ADD     A,#09
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
TEN:    CJNE    A,#40H,ONE1
        NOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV    A,#10
        ACALL FORWARD
        ACALL READ
        ADD    A,#10
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
ONE1:  CJNE    A,#41H,TWO1
        NOP
        MOV    A,#01
        LCALL REVERSE
        ACALL READ
        DEC    A
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
TWO1:  CJNE    A,#42H,THREE1
        NOP
        MOV    A,#02
        LCALL REVERSE
        ACALL READ
        DEC    A
        DEC    A
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;.....
THREE1:     CJNE    A,#43H,FOUR1
            NOP
            MOV     A,#03
            LCALL REVERSE
            ACALL READ
            DEC     A
            DEC     A
            DEC     A
            ACALL WRITE
            ACALL READ
            ACALL Tx
            ACALL LEAD_SWITCH
            RETI
;.....
FOUR1: CJNE     A,#44H,FIVE1
            NOP
            MOV     A,#04
            LCALL REVERSE
            ACALL READ
            DEC     A
            DEC     A
            DEC     A
            DEC     A
            ACALL WRITE
            ACALL READ
            ACALL Tx
            ACALL LEAD_SWITCH
            RETI
;.....
FIVE1:  CJNE    A,#45H,SIX1
            NOP
            MOV     A,#05
            LCALL REVERSE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACALL READ
        DEC   A
        DEC   A
        DEC   A
        DEC   A
        DEC   A
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
SIX1:   CJNE  A,#46H,SEVEN1
        NOP
        MOV   A,#06
        LCALL REVERSE
        ACALL READ
        DEC   A
        DEC   A
        DEC   A
        DEC   A
        DEC   A
        DEC   A
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
SEVEN1:       CJNE  A,#47H,EIGHT1
        NOP
        MOV   A,#07
        LCALL REVERSE
        ACALL READ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
EIGHT1:         CJNE    A,#48H,NINE1
        NOP
        MOV     A,#08
        LCALL REVERSE
        ACALL READ
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
NINE1: CJNE     A,#49H,TEN1
        NOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,#09
        LCALL REVERSE
        ACALL READ
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        ACALL WRITE
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
;.....
TEN1:   CJNE    A,#50H,PP
        NOP
        MOV     A,#10
        LCALL REVERSE
        ACALL READ
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        DEC     A
        ACALL WRITE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACALL READ
        ACALL Tx
        ACALL LEAD_SWITCH
        RETI
        NOP
        RETI
PP:     RET
;.....
;;.....EEPROM.....
;*************RECIEVE*****************************
TxISR:  MOV   A,SBUF
        CLR   RI
        ACALL CLEAR
        NOP
        RETI

;.....
Tx:     ADD   A,#30H
        MOV   SBUF,A
        JNB   TI,$
        CLR   TI
        NOP
        RET
;.....
WRITE:  MOV   EEP_AH,#00H          ;WRITE
        MOV   EEP_AL,#00H
        MOV   EEP_DATA,A
        ACALL AT24C512_WR
        ACALL DELAY_100MS
        ACALL DELAY_100MS
        RET
READ:   MOV   EEP_AH,#00H          ;READ
        MOV   EEP_AL,#00H
        ACALL AT24C512_RD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACALL DELAY_100MS
        ACALL DELAY_100MS
        MOV   A,EEP_DATA
        RET
;**********AT24C512_RD*************************
AT24C512_RD: MOV   I2C_ADDR,#AT24C512_ID
             ACALL I2C_SLAVE
             JB    I2C_ACK,AT24C512_RD_EXIT

             MOV   I2C_DATA,EEP_AH
             ACALL I2C_DATA_WR

             MOV   I2C_DATA,EEP_AL
             ACALL I2C_DATA_WR

             MOV   I2C_ADDR,#AT24C512_ID+1
             ACALL I2C_SLAVE

             ACALL I2C_DATA_RD
             MOV   EEP_DATA,I2C_DATA
             ACALL I2C_NACK_BIT
             ACALL I2C_STOP
AT24C512_RD_EXIT:  RET
;**************AT24C512_WR*****************************
AT24C512_WR: MOV   I2C_ADDR,#AT24C512_ID
             ACALL I2C_SLAVE
             JB    I2C_ACK,AT24C512_WR_EXIT

             MOV   I2C_DATA,EEP_AH
             ACALL I2C_DATA_WR

             MOV   I2C_DATA,EEP_AL
             ACALL I2C_DATA_WR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     I2C_DATA,EEP_DATA
                ACALL I2C_DATA_WR

                ACALL I2C_STOP
AT24C512_WR_EXIT:   RET
;**************I2C DATA WRITE*********************
I2C_DATA_WR:PUSH    ACC
                SETB    I2C_ACK
                MOV     A,I2C_DATA
                MOV     R5,#008
I2C_DATA_WR_1:  RLC A
                MOV     SDA,C
                ACALL I2C_CLK
                DJNZ    R5,I2C_DATA_WR_1
                SETB    SDA
                ACALL I2C_DELAY
                SETB    SCL
                ACALL I2C_DELAY
                JB      SDA,I2C_DATA_WR_2
                CLR     I2C_ACK
I2C_DATA_WR_2:      CLR     SCL
                POP     ACC
                RET
;**************I2C DATA READ**************
I2C_DATA_RD: PUSH   ACC
                CLR     A
                MOV     R5,#008
I2C_DATA_RD_1:      ACALL I2C_DELAY
                SETB    SCL
                ACALL I2C_DELAY
                MOV     C,SDA
                RLC     A
                CLR     SCL
                DJNZ    R5,I2C_DATA_RD_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     I2C_DATA,A
                POP     ACC
                RET
;***************8I2C SLAVE CONNECT*****************
I2C_SLAVE:      PUSH    ACC
                SETB    I2C_ACK
                MOV     A,I2C_ADDR
                ACALL I2C_START
                MOV     R5,#008
I2C_SLAVE_1:    RLC     A
                MOV     SDA,C
                ACALL I2C_CLK
                DJNZ    R5,I2C_SLAVE_1

                SETB    SDA
                ACALL I2C_DELAY
                SETB    SCL
                ACALL I2C_DELAY
                JB      SDA,I2C_SLAVE_2
                CLR     I2C_ACK
I2C_SLAVE_2:    CLR     SCL
                POP     ACC
                RET

;*********************I2C START CONDITION************
I2C_START:      JNB     SCL,I2C_START_1
                CLR     SCL

I2C_START_1:    SETB    SDA
                SETB    SCL

                ACALL I2C_DELAY
                CLR     SDA
                ACALL I2C_DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR     SCL
                RET
;************************I2C STOP CONDITION***************
I2C_STOP:       JNB     SCL,I2C_STOP_1
                CLR     SCL
I2C_STOP_1:     CLR     SDA
                ACALL I2C_DELAY
                SETB    SCL
                ACALL I2C_DELAY
                SETB    SDA
                RET
;********************I2C CLOCK*****************************
I2C_CLK:        ACALL I2C_DELAY
                SETB    SCL
                ACALL I2C_DELAY
                CLR     SCL
                RET
;************************I2C ACKNOWLEDGE**************
I2C_ACK_BIT:    CLR     SDA
                ACALL I2C_DELAY
                ACALL I2C_CLK
                SETB    SDA
                RET
;**********************I2C NOT ACKNOWLEDGE***************
I2C_NACK_BIT:SETB   SDA
                ACALL I2C_DELAY
                ACALL I2C_CLK
                SETB    SCL
                RET
;*************DUMMY DELAY TIME**************************
I2C_DELAY:      MOV     R6,#00CH
I2C_DELAY_1:    NOP
                NOP
                DJNZ    R6,I2C_DELAY_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET
;***********************************************
;***********************************************
DELAY_100MS: MOV    R7,#100
DELAY_100MS_1:      MOV    R6,#0E6H
DELAY_100MS_2:      NOP
                NOP
                DJNZ    R6,DELAY_100MS_2
                DJNZ    R7,DELAY_100MS_1
                RET
;*********FORWARD*************
FORWARD:CJNE        A,#00H,ROTATE
        NOP
        RET

ROTATE:         MOV    P2,#11110000B
        ACALL DELAY_100ms;1
        MOV    P2,#10110100B
        ACALL DELAY_100ms;2
        MOV    P2,#10100101B
        ACALL DELAY_100ms;3
        MOV    P2,#11100001B
        ACALL DELAY_100ms;4
        MOV    P2,#11110000B
        ACALL DELAY_100ms;1
        MOV    P2,#10110100B
        ACALL DELAY_100ms;2
        MOV    P2,#10100101B
        ACALL DELAY_100ms;3
        MOV    P2,#11100001B
        ACALL DELAY_100ms;4
        DEC    A
        AJMP   FORWARD
;.....
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
REVERSE:CJNE A,#00H,ROTATE1
        NOP
        RET
ROTATE1:MOV P2,#11100001B
        ACALL DELAY_100ms;1
        MOV   P2,#10100101B
        ACALL DELAY_100ms;2
        MOV   P2,#10110100B
        ACALL DELAY_100ms;3
        MOV   P2,#11110000B
        ACALL DELAY_100ms;4
        MOV   P2,#11100001B
        ACALL DELAY_100ms;1
        MOV   P2,#10100101B
        ACALL DELAY_100ms;2
        MOV   P2,#10110100B
        ACALL DELAY_100ms;3
        MOV   P2,#11110000B
        ACALL DELAY_100ms;4
        DEC   A
        AJMP  REVERSE
;*********************************************
DELAY_1s:     MOV   R5,#100
DELAY_1s_1:   ACALL DELAY_10ms
              DJNZ  R5,DELAY_1s_1
              RET
;*******************************************
DELAY_10ms:   MOV   R7,#020
DELAY_10ms_1:       MOV   R6,#230
DELAY_10ms_2:       NOP
              NOP
              DJNZ  R6,DELAY_10ms_2
              DJNZ  R7,DELAY_10ms_1
              RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;****************************************
LEAD_SWITCH:    MOV   A,P1
                ANL   A,#03FH
                CJNE  A,#3EH,ANGLE_90
                MOV   A,#35H
                MOV   P1,A
                ACALL REPLY
                ACALL DELAY_100ms
                MOV   P1,#0FFH
                ACALL OO
ANGLE_90:       CJNE  A,#3DH,ANGLE_45R
                MOV   A,#40H
                MOV   P1,A
                ACALL REPLY
                ACALL DELAY_100ms
                MOV   P1,#0FFH
                ACALL OO
ANGLE_45R:      CJNE  A,#3BH,ANGLE_90R
                MOV   A,#45H
                MOV   P1,A
                ACALL REPLY
                ACALL DELAY_100ms
                MOV   P1,#0FFH
                ACALL OO
ANGLE_90R:      CJNE  A,#37H,OO
                MOV   A,#50H
                MOV   P1,A
                ACALL REPLY
                ACALL DELAY_100ms
                MOV   P1,#0FFH
                ACALL OO
OO:             RET
                ;=========================
REPLY:          CLR   TI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   SBUF,A
        JNB   TI,$
        CLR   TI
        RET
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้