ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS

SMS - CONTROLLED ELECTRIC APPLIANCES

นางสาวจันทร์จิรา ต๊ะมาธง รหัส 46380011

นางสาวมรกต ศิริรัตน์ รหัส 46380035

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 9 มี.ค. 2553
เลขทะเบียน 14942425
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร 9253 S
2550

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS			
ผู้ดำเนินโครงงาน	นางสาวจันทร์จิรา	ต๊ะมาธง	รหัส	46380011
	นางสาวมรกต	ศิริรัตน์	รหัส	46380035
อาจารย์ที่ปรึกษา	ดร.ไพศาล มุณีสว่าง			
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2550			

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.ไพศาล มุณีสว่าง)

.....กรรมการ

(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ

(อาจารย์จิราพร พุกสุข)

หัวข้อโครงงาน	ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS			
ผู้ดำเนินงาน	นางสาวจันทร์จิรา ต๊ะมาธง		รหัส	46380011
	นางสาวมรกต ศิริรัตน์		รหัส	46380035
อาจารย์ที่ปรึกษา	ดร.ไพศาล มุณีสว่าง			
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2550			

.....

บทคัดย่อ

จากความต้องการควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน โดยไม่จำกัดเวลา สถานที่ และระยะทาง ทำให้เกิดการประยุกต์ใช้เทคโนโลยีสื่อสารแบบไร้สาย ควบคู่กับการทำงานของ Microcontroller โดยเทคโนโลยีสื่อสารแบบไร้สายในปัจจุบันที่มีประสิทธิภาพ และเป็นเทคโนโลยีที่รองรับในโทรศัพท์เคลื่อนที่ทั่วๆ ไป มีค่าใช้บริการที่ไม่สูงมากนัก ก็คือบริการส่งข้อความสั้น (Short Message Service) หรือ SMS ก่อให้เกิดแนวคิดในการออกแบบระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS

โครงงานนี้เป็นการจำลองการควบคุมอุปกรณ์ไฟฟ้าในบ้านโดยการใช้บริการส่งข้อความสั้น (Short Message Service) หรือ SMS จากโทรศัพท์มือถือ มาเป็นช่องทางในการส่งสัญญาณควบคุมไปยังเครื่องรับข้อความ (GR 64 GSM/GPRS Module) โดยได้เชื่อมต่อเข้ากับชุดแปลสัญญาณควบคุมที่สร้างขึ้นโดย Microcontroller Single Board (STARTC-51) แปลสัญญาณดังกล่าว เป็นสัญญาณทางไฟฟ้า ที่สามารถนำไปควบคุมการปิด/เปิดอุปกรณ์ไฟฟ้าเพื่อควบคุมการทำงานของระบบ โดยอาศัยวงจร ET-OPTO RELAY 4 ทำหน้าที่เป็นสวิตซ์ ปิด/เปิดอุปกรณ์

Project Title	SMS - CONTROLLED ELECTRIC APPLIANCES.		
Name	Ms. Janjira	Tamatong	ID. 46380011
	Ms. Morakot	Sirirat	ID. 46380035
Project Advisor	Paisarn Muneesawang, Ph.D.		
Major	Computer Engineering.		
Department	Electrical and Computer Engineering.		
Academic Year	2007		

ABSTRACT

The wireless communications system known as "SMS" (or Short Message Service) is a standard technology on mobile phone that is easier, cheaper and more efficient than other standards. SMS technology can also be applied with a Microcontroller so that home electric appliances can controlled remotely. It means that people can control the electric appliances overcome time, place and distance limitations.

This project presents the simulation of electric controlled system using message in SMS as a command via a GR 64 GSM/GPRS Module. The GR 64 GSM/GPRS Module is connected to a Microcontroller single board (STARTC-51) used to translate messages stored in the GR 64 GSM/GPRS Module to electrical signals. Electrical signals from the Microcontroller are send out to a control circuit to control electric appliances by using ET-OPTO RELAY 4 as a switch of appliances.

กิตติกรรมประกาศ

การจัดทำโครงงานระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS สำเร็จลุล่วงไปได้ด้วยดี โดยทั้งนี้ คณะผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาโครงงาน ดร.ไพศาล มุณีสว่าง เป็นอย่างสูง ที่ให้คำปรึกษา แนวความคิด แนวทางในการแก้ไขปัญหา ตลอดจนคำแนะนำต่าง ๆ อันเป็นประโยชน์อย่างยิ่งต่อโครงงานนี้

ขอบคุณ คุณอนันต์ ภูกิตติกุล จากบริษัท ศิลารีเสิร์ช จำกัด ที่ให้ข้อมูลดีๆ และเป็นประโยชน์สำหรับโครงงานนี้อย่างมาก

ขอบคุณทางหอพักนริศา ที่ให้ความเอื้อเฟื้อเผื่อแผ่มาโดยตลอด

ขอบคุณพี่รหัส ที่คอยช่วยเหลือและแนะนำ ทั้งในเรื่องการเรียนและการจัดทำโครงงานในครั้งนี้

ขอกราบขอบพระคุณบิดา มารดา ญาติพี่น้อง ที่ช่วยดูแล เป็นกำลังใจ และให้ความรัก ความอบอุ่นตลอดเวลา โดยเฉพาะทุนทรัพย์จากบิดาและมารดาที่เอื้อหนุนตลอดมา

จันทร์จิรา ตั๊ะมาธง

มรกต ศิริรัตน์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูปภาพ ..... ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบข่ายของโครงงาน 2

1.4 ขั้นตอนวิธีการดำเนินงาน 2

1.5 ผลที่คาดว่าจะได้รับ 3

1.6 งบประมาณที่ใช้ 3

บทที่ 2 หลักการและทฤษฎีเบื้องต้น

2.1 หลักการทำงานของการใช้ SMS ควบคุมอุปกรณ์ไฟฟ้า 4

2.2 หลักการรับ- ส่ง SMS 5

2.3 PDU Mode 6

2.4 การติดต่อโดยใช้ชุดคำสั่ง AT Command 9

2.5 Serial communication 11

2.6 ส่วนประกอบของวงจร 16

2.6.1 Microcontroller Single Board (STARTC-51) 16

2.6.2 GR-64 (GR64 GSM/GPRS Module) 20

2.6.3 ET-OPTO RELAY 4 22

2.6.4 ภาษา Assembly 24

สารบัญ(ต่อ)

หน้า

บทที่ 3 วิธีดำเนินงานโครงงานวิศวกรรม

3.1 การวิเคราะห์ระบบควบคุมอุกรณ์ไฟฟ้าด้วย SMS 28

3.2 การออกแบบระบบควบคุมอุกรณ์ไฟฟ้าด้วย SMS 28

3.3 การพัฒนาระบบควบคุมอุกรณ์ไฟฟ้าด้วย SMS 28

3.3.1 โปรแกรมควบคุมการทำงาน 29

3.3.2 การนำวงจร ET – Opto Relay 4 มาใช้งาน 34

3.4 การทดสอบและประเมินผลการดำเนินโครงงาน 36

บทที่ 4 การทดสอบและการวิเคราะห์การทำงาน

4.1 การควบคุมให้หลอดไฟทำงาน 37

4.2 การควบคุมให้หลอดหยุดไฟทำงาน 38

4.3 การควบคุมให้หลอดไฟที่ทำงานอยู่หยุดทำงานและควบคุมให้หลอดไฟที่ไม่ทำงาน เริ่มทำงานในการส่งข้อความเพียงครั้งเดียว 38

4.4 การควบคุมให้ระบบส่งข้อความแจ้งผลการทำงานกลับมายังผู้ใช้ 39

บทที่ 5 บทสรุป

5.1 สรุปผลการดำเนินโครงงาน 40

5.2 ปัญหาและแนวทางแก้ไข 40

5.3 ข้อจำกัด 41

5.4 ข้อเสนอแนะในการพัฒนาต่อไป 41

เอกสารอ้างอิง 42

ภาคผนวก ก AT Commands 43

ภาคผนวก ข Assembly Code 45

ภาคผนวก ค การดำเนินงานของโครงงาน 60

ประวัติผู้เขียนโครงงาน 71

สารบัญตาราง

ตารางที่ หน้า

2.1 ความหมายของรหัส SMS ที่ได้รับ 7

2.2 ความหมายของรหัส SMS ที่ได้รับ 8

2.3 ตารางค่าดงที่ ๆ กำหนดจากการใช้ ไทม์เมอร์1 เมื่อใช้อัตราความเร็วในการส่งข้อมูล มาตรฐาน 15

2.4 ตารางแสดงจุดเชื่อมต่อสัญญาณของบอร์ด ET-OPTO RELAY 4 กับ ไมโครคอนโทรเลอร์ 25

สารบัญรูปภาพ

รูปที่ หน้า

2.1 หลักการทำงานของการใช้ SMS ควบคุมอุปกรณ์ไฟฟ้า....4

2.2 แสดงหลักการรับส่ง SMS....5

2.3 ข้อมูลใน SMS ในรูปแบบ PDU MODE....6

2.4 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ"0"....11

2.5 ขาสัญญาณพอร์ตอนุกรมของ PC (DB9)....13

2.6 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem....13

2.7 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น....13

2.8 โครงสร้างของ Microcontroller Single Board (STARTC-51)....18

2.9 แสดงวงจรของ Microcontroller Single Board (STARTC-51)....19

2.10 โครงสร้างของ GR 64 GSM/GPRS Module....20

2.11 แสดงวงจรของ GR 64 GSM/GPRS Module....21

2.12 โครงสร้างของบอร์ด ET-OPTO RELAY 4....22

2.13 แสดงวงจรของ ET-OPTO RELAY4 ขนาด 1 ช่อง....22

2.14 แสดงวงจรของ ET-OPTO RELAY4....24

2.15 แสดงการจัดเรียงสัญญาณของขั้ว IDE 10 Pin CPA 6 Pin....25

3.1 ตัวอย่างการทำงานของโปรแกรม....30

3.2 แผนผังการทำงานโปรแกรม....32

3.3 แผนผังแสดงขั้นตอนในการพัฒนาโปรแกรมสำหรับควบคุมอุปกรณ์ไฟฟ้า....33

3.4 แสดงการควบคุมอุปกรณ์ไฟฟ้าของ STARTC-51 ผ่าน วงจร ET – Opto Relay 4....34

3.5 แสดงภาพ12B PORT ของ STARTC-51 และ ขั้ว IDE-10 Pin ของ วงจร ET – Opto Relay 4....34

3.6 แสดงการเชื่อมต่อของ 12B PORT บน STARTC-51 และ ขั้ว IDE-10 Pin บน วงจร ET – Opto Relay 4....35

3.7 แสดงการตรวจสอบการทำงานของ STARTC-51 และ วงจร ET – Opto Relay 4....35

3.8 การประกอบอุปกรณ์ทั้งหมดเข้าด้วยกัน และสร้างแบบจำลอง....36

3.9 แบบจำลองการทำงานของอุปกรณ์ไฟฟ้า....36

สารบัญรูปภาพ (ต่อ)

รูปที่ **หน้า**

4.1 แสดงผลการทำงานของคำสั่ง NUOBCE 37
4.2 แสดงผลการทำงานของคำสั่ง NUOC1357E 38
4.3 แสดงผลการทำงานของคำสั่ง NUO1357C2468E 38
4.4 แสดงผลการทำงานของคำสั่ง NUO86C31SE และข้อความแจ้งผลการทำงานที่ส่งกลับมา 39
4.5 แสดงผลการทำงานของคำสั่ง NUOACSE และข้อความแจ้งผลการทำงานที่ส่งกลับมา 39

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

จากเทคโนโลยีในปัจจุบัน ทำให้เกิดเครื่องอำนวยความสะดวกในชีวิตประจำวันขึ้นอย่างมากมาย อุปกรณ์เครื่องใช้ไฟฟ้าเป็นเครื่องอำนวยความสะดวกอีกชนิดหนึ่ง ซึ่งมีความสำคัญและมีบทบาทอย่างยิ่งกับชีวิตประจำวันของทุกคน ยิ่งเทคโนโลยีมีพัฒนาการไปมากเท่าไร เครื่องใช้ไฟฟ้ายิ่งก็เข้ามามีบทบาทกับชีวิตประจำวันของเรามากขึ้นเท่านั้น แต่ในขณะเดียวกันเมื่อเครื่องใช้ไฟฟ้ามีคุณประโยชน์กับเราอย่างมากมาย แต่หากเราไม่ระมัดระวัง หรือประมาทเลินเล่อ เครื่องใช้ไฟฟ้าทั้งหลาย ก็ก่อโทษอย่างมหาศาลให้เราได้เช่นกัน เช่น หากเราเปิดเครื่องใช้ไฟฟ้าบางชนิดทิ้งไว้ อาจทำให้เกิดเพลิงไหม้ และเพลิงอาจลุกลามจนเหนือการควบคุม เผาไหม้ทำลายทรัพย์สิน ก่อความเสียหายอย่างมหาศาล

แต่ด้วยพัฒนาการของเทคโนโลยี และการประยุกต์เข้ากับการรักษาความปลอดภัยในการใช้งานเครื่องใช้ไฟฟ้า จึงได้มีการประดิษฐ์ระบบควบคุมเครื่องใช้ไฟฟ้าในบ้านระยะไกล เพื่อควบคุมเครื่องใช้ไฟฟ้าจากนอกบ้าน ทั้งนี้เพื่ออำนวยความสะดวกในการควบคุมเครื่องใช้ไฟฟ้า และป้องกันเหตุร้าย ที่จะเกิดจากการเปิดเครื่องใช้ไฟฟ้าทิ้งไว้ หรือการขัดข้องของเครื่องใช้ไฟฟ้าบางชนิด อันนำมาซึ่งความสูญเสียอย่างมากมาย

ในส่วนของโครงงานนี้ เป็นการศึกษาวงจรควบคุมอุปกรณ์ไฟฟ้าโดยอาศัยบริการส่งข้อความสั้น (Short Message Service: SMS) เป็นช่องทางในการส่งสัญญาณเพื่อควบคุมอุปกรณ์ไฟฟ้าจากนอกบ้าน ซึ่งจากโครงงานของนายวรายุทธ คัมภีราวัฒน์ และ นางสาวสุชาดา ไชยสุข [1] ที่ใช้ทฤษฎีของบริการส่งข้อความสั้น ประยุกต์เข้ากับการใช้งาน Microcontroller โดยมีการใช้โทรศัพท์เคลื่อนที่เป็นตัวรับสัญญาณ และใช้ Microcontroller เบอร์ AT90S2313 ในตระกูล AVR เป็นวงจรถอดรหัสสัญญาณ และควบคุมอุปกรณ์ไฟฟ้า ในส่วนของโครงงานนี้จะประยุกต์ทฤษฎีดังกล่าว โดยจะใช้ Microcontroller เบอร์ P89C51RD2BN ในตระกูล MCS-51 ซึ่งจะมีประโยชน์คือ เป็น Microcontroller ในตระกูลที่เป็นที่นิยม มีการพัฒนาอย่างต่อเนื่อง มีทางเลือกที่หลากหลาย และทำให้การโปรแกรมมีความง่าย มีขนาดเล็ก ทำงานได้รวดเร็ว ทำให้สามารถพัฒนาโปรแกรมให้มีความสามารถมากขึ้นกว่าเดิมได้ คือ ให้มีการแจ้งสถานะการทำงานกลับไปยังผู้ใช้งาน นอกจากนี้ยังใช้ GR-64 เป็นตัวรับสัญญาณแทนการใช้โทรศัพท์เคลื่อนที่ทั่ว ๆ ไป เนื่องจาก GR-64 เป็นอุปกรณ์ที่ออกแบบมาเพื่อใช้ในการรับ- ส่งข้อมูลผ่านทาง SMS , CSD , GPRS และสามารถรองรับการสื่อสารผ่านพอร์ตอนุกรมได้ ทำให้การรับ-ส่งสัญญาณมีความเสถียรมากขึ้น ซึ่งการเลือกใช้อุปกรณ์ดังกล่าวจะเป็นประโยชน์มาก สำหรับการพัฒนาในขั้นต่อ ๆ ไป สำหรับผู้ที่สนใจ

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อสามารถควบคุมอุปกรณ์ไฟฟ้า จากนอกบ้านได้
2. เพื่อป้องกันเหตุร้ายอันจะเกิดจากความประมาทของมนุษย์
3. ช่วยประหยัดพลังงานไฟฟ้า
4. ช่วยอำนวยความสะดวกในการ ปิด-เปิด เครื่องใช้ไฟฟ้าได้ในระยะไกล
5. เพื่อใช้เป็นข้อมูลสำหรับใช้เป็นแนวทางในการศึกษาและพัฒนาระบบได้ต่อไปในอนาคต

1.3 ขอบข่ายของโครงงาน

1. ศึกษาหลักการ การสื่อสารผ่านพอร์ตอนุกรมระหว่างอุปกรณ์ที่เกี่ยวข้อง
2. ออกแบบระบบสำหรับควบคุมอุปกรณ์ไฟฟ้า
3. เชื่อมต่อวงจร STARTC-51 และ GR64
4. ออกแบบการทำงานของระบบควบคุมผ่านทาง Microcontroller Single Board (STARTC-51)
5. ออกแบบวงจรที่จะใช้ในการควบคุมอุปกรณ์ไฟฟ้าโดยอาศัยสัญญาณ TTL ของ RELAY จาก ET – Opto Relay 4

1.4 ขั้นตอนการดำเนินงาน

กิจกรรม	2550									2551
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.
1. ศึกษาหาข้อมูลเรื่อง - หลักการรับ-ส่ง SMS - การสื่อสารผ่านพอร์ตอนุกรม - หลักการและวิธีการใช้งานของ Microcontroller Single Board (STARTC-51) - วิธีการใช้งานของ GR-64 และ ET - Opto Relay 4 - หลักการเขียนโปรแกรมภาษา Assembly	←	—	—	→						

กิจกรรม	2550									2551
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.
2. เชื่อมต่อวงจรของ STARTC-51, GR64 และ ET -Opto Relay 4				←	→					
3. วิเคราะห์และออกแบบการทำงานของระบบควบคุมผ่านทางSTARTC-51				←	→	→				
4. เขียนโปรแกรมสำหรับการทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าผ่านทาง Microcontroller					←	→	→			
5. ทดสอบการทำงานของระบบควบคุมอุปกรณ์ไฟฟ้า ด้วย SMS						←	→	→		
6. พัฒนา แก้ไขและปรับปรุงระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS							←	→	→	
7. จัดทำรายงาน								←	→	→

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถควบคุมอุปกรณ์ไฟฟ้าจากนอกบ้านได้
2. อำนวยความสะดวกในการควบคุมอุปกรณ์ไฟฟ้า โดยไม่จำกัดเวลา และสถานที่ และระยะทาง
3. รู้ถึงหลักการทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS มากขึ้น
4. สามารถนำระบบควบคุมที่ได้ไปพัฒนาและประยุกต์ใช้ในงานอื่น ๆ ต่อไปได้ในอนาคต

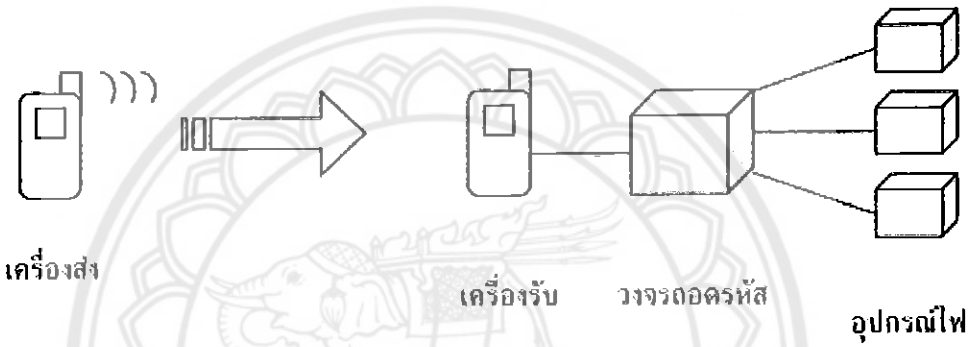
1.6 งบประมาณที่ใช้

ค่าวัสดุ-อุปกรณ์	6,500 บาท
ค่าเอกสาร	1,500 บาท
รวมทั้งสิ้น	8,000 บาท

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 หลักการทำงานของการใช้ SMS ควบคุมอุปกรณ์ไฟฟ้า [2]

หลักการการทำงานคือ ส่งข้อความสั้นจากโทรศัพท์มือถือเครื่องหนึ่ง ไปยังโทรศัพท์มือถือเครื่องรับ ซึ่งได้ต่อเชื่อมกับอุปกรณ์ที่สามารถถอดรหัสข้อความที่ส่งมา และแปลข้อความนั้นเป็นสัญญาณทางไฟฟ้า จากนั้นจึงส่งไปควบคุมอุปกรณ์เครื่องใช้ไฟฟ้า



รูปที่2.1 หลักการทำงานของการใช้ SMS ควบคุมอุปกรณ์ไฟฟ้า

จากการเรียนการสอนในรายวิชาการเชื่อมต่อไมโครคอมพิวเตอร์และไมโครคอนโทรลเลอร์ (Microcontroller and Microcomputer Interfacing) รหัสวิชา 305381 ทำให้ได้ศึกษาการใช้งาน Microcontroller Single Board (STARTC-51) จึงได้มีการนำ Microcontroller Single Board (START C-51) มาประยุกต์ใช้เป็นตัวควบคุมอุปกรณ์ไฟฟ้าในโครงงานนี้ เพื่อเป็นการพัฒนาความรู้ที่มีอยู่เดิมและนำอุปกรณ์ประกอบการศึกษามาใช้งานให้เกิดประโยชน์ขึ้นไปอีก และนอกจากนี้ยังได้มีการนำ GR 64 GSM/GPRS Module มาเป็นเครื่องรับข้อความ แทนการใช้โทรศัพท์มือถือทั่ว ๆ ไป เนื่องจากอุปกรณ์ GR 64 GSM/GPRS Module นั้น มีคุณสมบัติที่เหมาะสม มีฟังก์ชันการทำงานครบสมบูรณ์และยังสามารถนำไปพัฒนาให้เกิดประโยชน์อื่นๆ ได้ในอนาคต ส่วนการควบคุมการเปิด/ปิด ของอุปกรณ์ไฟฟ้านั้น ได้นำวงจร ET - OPTO RELAY 4 มาใช้งานในลักษณะของหน้าสัมผัส ON/OFF และเป็นวงจรแบบ OPTO-ISOLATE ในการควบคุมการทำงาน เพื่อป้องกันสัญญาณรบกวนต่าง ๆ ที่เกิดจากการทำงานของ RELAY เพื่อให้ STARTC-51 นั้นสามารถทำงานได้เสถียรมากขึ้น

2.2 หลักการรับ-ส่ง SMS

SMS : Short Massage Service เป็นการให้บริการส่งข้อความสั้นจากโทรศัพท์มือถือต้นทางผ่านชุมสายไปยังโทรศัพท์มือถือปลายทาง โดยสามารถส่งได้สูงสุด 160 ตัวอักษรต่อครั้ง ตามข้อกำหนดมาตรฐานขององค์การ ETSI (European Telecommunications Standards Institute)

ระบบ SMS ในระบบเครือข่ายโทรศัพท์มือถือ รองรับโดยระบบ GSM (Global System for Mobile Communication), TDMA (time division multiple access) และ CDMA (code division multiple access)

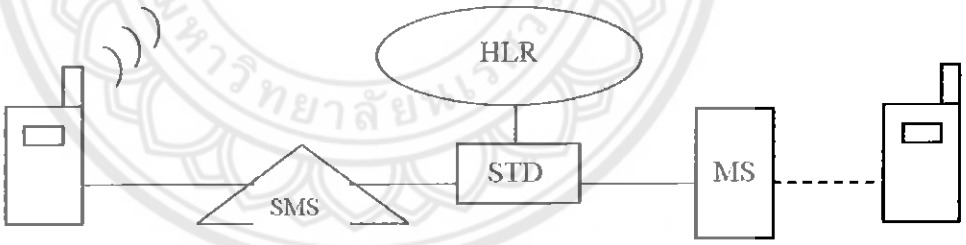
เมื่อ SMS ถูกส่งจากโทรศัพท์มือถือเครื่องหนึ่ง ข้อความนั้นจะถูกส่งไปที่ Shot Message Service Center (SMSC) จากนั้นจึงจะส่งไปยังโทรศัพท์มือถือเครื่องรับอีกทอดหนึ่ง โดยมีกระบวนการดังนี้

1. SMSC จะส่ง SMS Request ไปยัง Home location register (HLR) เพื่อหาตำแหน่งของผู้รับ

2. เมื่อ HLR ได้รับสัญญาณ Request ก็จะส่งสถานะของผู้รับ (subscriber's status) กลับมายัง SMSC คือ

   1.) สถานะของเครื่องรับ Inactive หรือ Active

   2.) ตำแหน่งของเครื่องรับ



รูป 2.2 แสดงหลักการรับส่ง SMS

ถ้าสถานะของเครื่องรับเป็น Inactive แล้ว SMSC จะเก็บข้อความไว้ช่วงเวลาหนึ่ง และเมื่อใดที่เครื่องรับมีสถานะ Active แล้ว HLR จะส่ง SMS Notification ไปยัง SMSC และ SMSC ก็จะตอบรับข้อความนั้นไว้ จากนั้น SMSC จะส่งผ่านข้อความไปในรูปแบบ Short Message Delivery Point-to-Point ไปยังระบบบริการ โดยระบบจะทำการเรียกไปยังเครื่องรับ และถ้าเครื่องรับมีการตอบรับกลับมา ข้อความก็จะถูกส่งตามไปและ SMSC จะได้รับการตอบยืนยันว่า ข้อความได้ถูกรับโดยปลายทางเรียบร้อยแล้ว หลังจากนั้นข้อความจะมีสถานะเป็น SENT และจะไม่ถูกส่งอีก

โหมดของการรับส่งข้อมูล SMS

โหมดของการรับส่งข้อมูล SMS แบ่งออกเป็น 2 โหมด คือ

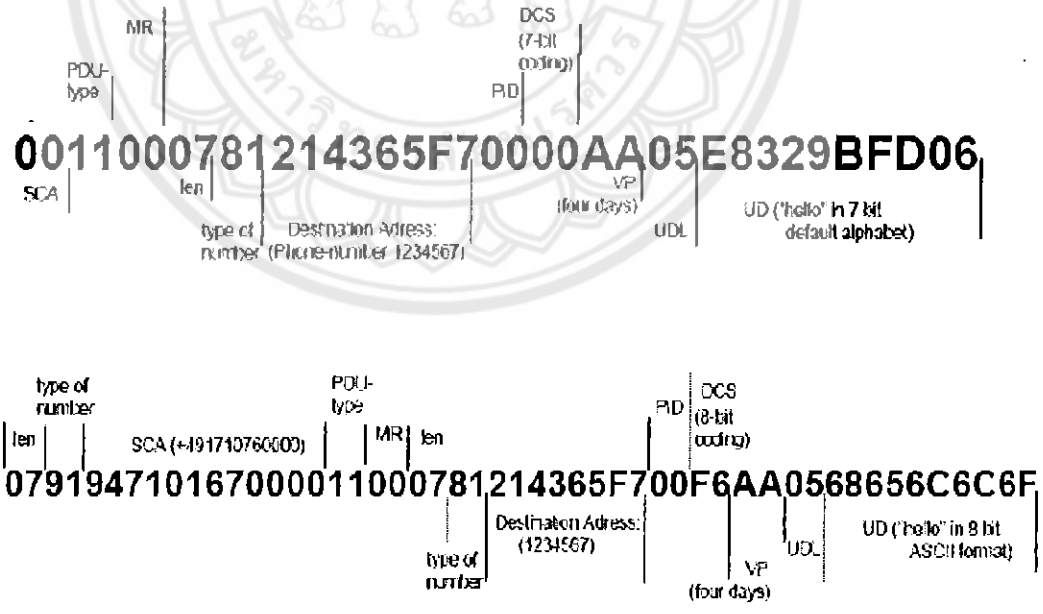
1. Text Mode
2. PDU Mode (Protocol Description Unit Mode)

การส่งข้อความในรูปแบบ Text Mode เป็นการส่งข้อมูลในรูปของตัวอักษรได้โดยตรง และไม่จำเป็นต้องนำมาเข้ารหัสก่อนส่งออก ซึ่งจะใช้โมดูล GR 64 เป็นตัวรองรับการส่งแบบ Text Mode ผ่านทางชุดคำสั่ง AT Command ส่วน PDU Mode: PACKET DATA UNIT เป็นรูปแบบการส่งข้อความ SMS อีกรูปแบบหนึ่งที่ต้องมีการเข้ารหัสข้อมูลที่สลับซับซ้อน

2.3 PDU MODE

PDU : Protocol Description Unit คือ รูปแบบของข้อมูลที่ใช้ในการส่งข้อความสั้น ซึ่งเกิดจากการนำข้อมูลดิจิตอลมาเข้ารหัสเพื่อใช้ในการส่งผ่านข้อมูลผ่าน Air Interface โดยการรับและส่งข้อมูล SMS ในแบบ PDU MODE จะมีส่วนประกอบและโครงสร้างที่แตกต่างกัน

PDU CODE ข้อมูล SMS ที่อยู่ใน PDU MODE ประกอบด้วยเลขฐาน10 และเลขฐาน 16 โดยตัวเลขแต่ละคู่เรียกว่า Octet



รูปที่ 2.3 ข้อมูลใน SMS ในรูปแบบ PDU MODE

2.3.1 การรับข้อมูล SMS

ข้อมูลที่ได้รับจะประกอบด้วยข้อมูลของผู้ส่ง , ข้อมูล SMS Service Center (SMSC) , Time-Stamp และอื่นๆ ตามด้วยส่วนของข้อความซึ่งจะอยู่ท้ายสุด

ตัวอย่าง

0691801813079304 0A91807131107000004030212194348 20AE8329BFD4697D9E CC37

ความหมายของรหัสในแต่ละ Octet หรือ กลุ่ม Octet เป็นไปตามตาราง ดังนี้

ตารางที่ 2.1 ความหมายของรหัส SMS ที่ได้รับ

กลุ่มตัวเลข 8 บิต (Octet)	รายละเอียด
06	ความยาวของ SMSC Information 6 Octet
91	รูปแบบของเลขหมาย SMSC 91 หมายถึงเลขหมายแบบสากล
80 18 13 07 93	เลขหมาย SMSC ซึ่งจะเป็นเลขฐาน 10 สลับ nibble ในกรณีนี้เลขหมายจริงของ Service Center คือ+66818310739
04	First octet of these SMS-DELIVER message
0A	ความยาวของเลขหมายผู้ส่ง (0A hex = 10 ตัว)
91	รูปแบบของเลขหมายผู้ส่ง 91 หมายถึงเลขหมายแบบสากล
80 71 31 10 70	เลขหมายผู้ส่ง เป็นเลขฐาน 10 สลับ nibble ในกรณีนี้เลขหมายจริงของผู้ส่งคือ +66871131007
00	TP-PID. (Protocol identifier)
00	TP-DCS (Data coding scheme) 00 คือเข้ารหัสข้อความแบบ 7 bits Default Alphabet
40 30 21 21 94 34 82	TP-SCTS. ข้อมูล Time stamp สลับ nibble
0A	TP-UDL. User data length จำนวนตัวอักษรของข้อความที่ส่งในที่นี้คือ 10 ตัว
E8329BFD4697D9ECC37	TP-UD. ข้อความ “hellohello” ที่เข้ารหัสแบบตัวอักษร 7 bits

2.3.2 การส่งข้อมูล SMS

ข้อมูลที่ส่งจะประกอบด้วย ข้อมูลของ SMSC , หมายเลขของผู้รับ , ช่วงเวลาหมดอายุของข้อความ , และ อื่นๆ ปิดท้ายด้วยข้อมูลในส่วนของข้อความสั้น

ตัวอย่าง

0011000A9180737899750000AA0AE8329BFD4697D9EC37

ความหมายของรหัสในแต่ละ Octet หรือ กลุ่ม Octet เป็นไปตามตาราง ดังนี้

ตารางที่ 2.2 ความหมายของรหัส SMS ที่ส่ง

กลุ่มตัวเลข 8 บิต (Octet)	รายละเอียด
00	ความยาวของ SMSC Information โดยที่ 00 หมายถึงให้ใช้ SMSC Information ที่เก็บอยู่ภายในเครื่อง (ปกติแล้วเครื่องที่สามารถส่ง SMS ได้มีข้อมูล SMSC ภายในเครื่องอยู่แล้ว)
11	First octet of the SMS-SUBMIT message
00	TP-message-Reference โดยที่ 00 คือให้เครื่องตั้งหมายเลขอ้างอิงข้อความขึ้นเอง
0A	Address-Length คือความยาวของเลขหมายผู้รับ (10 ตัว)
91	Type-of-Address. (91 indicates international format of the phone number)
80 73 78 99 75	เบอร์โทรศัพท์เคลื่อนที่ปลายทางที่ทำการเข้ารหัสแบบสลับแล้ว โดยหมายเลขที่แท้จริงคือ 083-7879957
00	TP-PID (Protocol identifier) เป็น 00
00	TP-DCS (Data coding scheme) เป็น 00
AA	TP-Validity-Period โดยที่ AA หมายถึง ช่วงเวลาหมดอายุของข้อความ 4 วัน ถ้าภายในช่วงเวลานี้ยังส่งไม่ถึงปลายทาง ข้อความจะถูกยกเลิกโดยอัตโนมัติ
0A	TP-User-Data-Length คือความยาวของข้อความก่อนแปลงรหัส นั่นคือ "hellohello" = 10 ตัวอักษร
E8329BFD4697D9EC37	TP-User-Data เป็นข้อมูลในส่วนของข้อความตัวอักษรแบบ 7 บิต นั่นคือคำว่า "hellohello" ที่ผ่านการแปลง (เข้ารหัส) เป็นข้อมูลแบบ 8 บิต แล้ว

การถอดรหัสตัวอักษรชนิด 7 บิต

ในส่วนของข้อมูลที่เป็นข้อความสั้นในกรณีที่เป็นตัวอักษรภาษาอังกฤษ อักษรแต่ละตัวจะใช้รหัสขนาด7 บิต (7 bits default alphabet) และสำหรับอักษรภาษาอื่นๆ จะใช้ตัวอักษรชนิดอื่น เช่น 8 บิต หรือ 16 บิต สำหรับข้อความสั้นภาษาไทยนั้น ใช้รหัสตัวอักษรแบบ UNICODE ในที่นี้เราจะกล่าวถึงการถอดรหัสข้อความสั้นภาษาอังกฤษเท่านั้นโดยมีขั้นตอนการถอดรหัส ดังนี้

1.) นำ PDU CODE ในส่วนของข้อความสั้น (TP-UD) ซึ่งเป็นเลขฐาน 16 มาเขียนเป็นเลขฐาน 2 ทีละ Octet

2.) ตัวอักษรแรกเกิดจากบิตที่ 0 ถึง 6 ของ Octet แรก ตัวอักษรถัดมาเกิดจากบิตที่ 0 ถึง 5 ของOctet ที่ 2 และนำบิตที่ 7 ของ Octet ที่ 1 มาต่อท้าย ตัวอักษรถัดไปเกิดจากบิตที่ 0 ถึง 4 ของOctet ที่ 3 และนำบิตที่ 7 และ 6 ของ Octet ที่ 2 มาต่อท้าย ทำเช่นนี้ไปเรื่อยๆ

3.) นำรหัสตัวอักษร 7 บิต จากข้อสอง เพิ่มบิตที่ 8 ด้วย '0' นำไปเทียบกับตารางรหัส ASCII โปรแกรม Microcontroller ที่เกี่ยวข้องกับการถอดรหัส PDU เป็น ASCII

2.4 การติดต่อโดยใช้ชุดคำสั่ง AT Command

ความหมายของ AT Command

AT Command เป็น ชุดคำสั่งมาตรฐาน ที่สามารถใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็ม หรือ อุปกรณ์ DTE (Data Terminal Equipment) เพื่อโต้ตอบตั้งค่าหรือสั่งงานอุปกรณ์เหล่านั้น ให้ทำงานตามที่ต้องการ

GSM AT Command

GSM AT Command เป็นชุดคำสั่งมาตรฐานที่ใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็มหรืออุปกรณ์ DTE (Data Terminal Equipment) ในชุดคำสั่งพื้นฐานนั้นบริษัท Hayes ได้เป็นผู้ออกแบบคิดค้นเพื่อใช้กับโมเด็มของตนเองและต่อมาบริษัทผู้ผลิตมือถือยี่ห้อต่างๆก็ได้พัฒนามาใช้กับผลิตภัณฑ์ของตนเอง เป็นเหตุให้คำสั่งพิเศษบางคำสั่งไม่เหมือนกันในผลิตภัณฑ์ต่างยี่ห้อกัน และความสามารถของโทรศัพท์ในบางรุ่นจะไม่รองรับคำสั่งดังกล่าว เนื่องจากไม่ได้มีวงจรส่วนของโมเด็มบรรจุอยู่ภายใน

ตัวอย่างคำสั่งที่เป็น BASIC AT COMMAND

AT ตรวจสอบความพร้อมของอุปกรณ์ ถ้าสามารถติดต่อกันได้อุปกรณ์จะตอบกลับมาว่า OK

ATDT โทรไปยังเลขหมายปลายทาง (phone number)

ATH วางสาย

ATA รับสาย

AT COMMAND ที่เกี่ยวข้องกับการรับ-ส่ง SMS

1.) Message Format (AT+CMGF) เป็นคำสั่งกำหนดรูปแบบของข้อความที่จะให้แสดงออกมา โดย

AT+CMGF = 1 คือ แสดงข้อความในรูปแบบ TEXT

AT+CMGF = 0 คือ แสดงข้อความในรูปแบบ PDU CODE

2.) List Message (AT+CMGL) เป็นคำสั่งที่ให้แสดงข้อความในสถานะต่างๆ โดยจะแสดงข้อความทั้งหมด นั้นสถานะนั้น ดังนี้

AT+CMGL=0 คือ แสดงข้อความที่ได้รับแต่ยังไม่ได้อ่าน ("REC UNREAD")

AT+CMGL=1 คือ แสดงข้อความที่ได้รับและอ่านแล้ว ("REC READ")

AT+CMGL=2 คือ แสดงข้อความที่เก็บไว้และยังไม่ได้ส่ง ("STO UNSENT")

AT+CMGL=3 คือ แสดงข้อความที่เก็บไว้และส่งออกไปแล้ว ("STO SENT")

AT+CMGL=4 คือ แสดงข้อความทั้งหมด ("ALL")

หมายเหตุ .... หากกำหนด Message Format เป็น PDU CODE จะต้องเลือกสถานะโดยใช้ตัวเลข 0 ถึง4 แต่หากกำหนด Message Format เป็น Text จะต้องเลือกสถานะโดยใช้ตัวอักษรที่วงเล็บด้านหลัง

3.) Read Message (AT+CMGR) เป็นคำสั่งที่ใช้อ่านข้อความที่เฉพาะเจาะจงได้โดยระบุตำแหน่งที่ข้อความนั้นถูกเก็บไว้

4.) Set the Message center number (AT+CSCA= "XXX") เป็นคำสั่งที่ใช้ระบุหมายเลขของ SMSC เพื่อจะทำการส่ง SMS ต่อไปได้

5.) Send Message (AT+CMGS= "YYY") เป็นคำสั่งที่ใช้ระบุเลขหมายปลายทางที่ต้องการส่ง SMS ไปถึง

6.) Delete Message (AT+CMGD=index ของข้อความที่จะลบ) เป็นคำสั่งลบข้อความ เช่น "AT+CMGD=1" หมายความว่าลบข้อความใน index ที่ 1

การส่ง SMS นั้นจะใช้ไมโครคอนโทลเลอร์ในการส่งโดยจะส่งชุดคำสั่ง AT-COMMAND เข้าไปทางพอร์ตอนุกรมของ modem ที่ต่ออยู่กับไมโครคอนโทรเลอร์ แต่ก่อนที่ส่งนั้นต้องทำการกำหนด อัตรา Baud Rate ให้ตรงกับ modem เสียก่อน หลังจากนั้นจะทดสอบคำสั่งพื้นฐานของชุดคำสั่ง AT-COMMAND เช่น AT+CMGL ในการตรวจเช็คข้อความที่มีการนำเข้า หลังจากนั้นจะทำการส่ง AT+CMGS เพื่อระบุหมายเลขปลายทาง เป็นต้น

2.5 SERIAL COMMUNICATION [3]

Serial Communication คือ การรับ-ส่งข้อมูลระหว่างอุปกรณ์ผ่านสายสัญญาณ โดยส่งผ่านข้อมูลไปทีละบิตต่อเนื่องกัน แบ่งได้เป็น 2 แบบ คือ

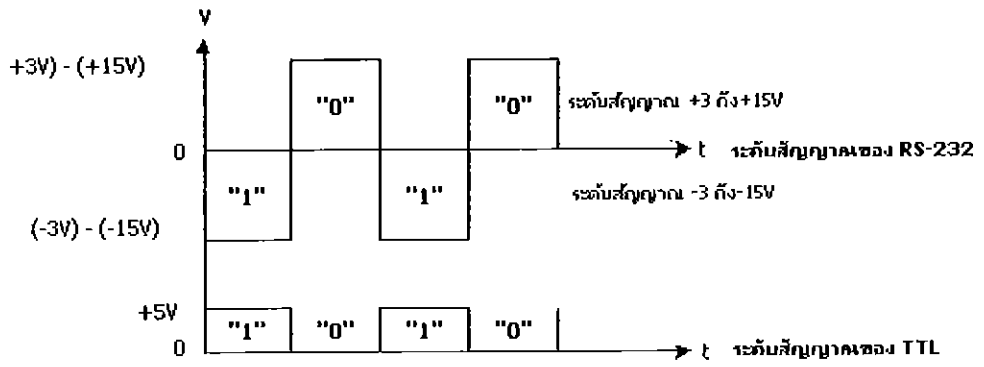
1. Synchronous มีการส่งสัญญาณ clock ไปพร้อมกับข้อมูล
2. Asynchronous ไม่มีการส่งสัญญาณ clock ไปกับข้อมูล

ในการสื่อสารรับ-ส่งข้อมูลในระยะไกลนั้น จำเป็นต้องใช้ Data Communication Equipment (DCE) อย่างเช่น Modem เพื่อให้ส่งข้อมูลได้ไกลขึ้น แต่สำหรับการสื่อสารในระยะไกลไม่เกิน 50 ฟุต ไม่จำ เป็นต้องใช้ DCE เรียกว่า การสื่อสารแบบ Null-modem และในการติดต่อสื่อสารข้อมูลระหว่างอุปกรณ์ จำเป็นจะต้องมีมาตรฐานในการเชื่อมต่อวงจร ซึ่งมาตรฐานที่ใช้กันมากที่สุดก็คือ มาตรฐาน RS232-C

การเชื่อมต่อพอร์ตอนุกรมตามมาตรฐาน RS-232-C

การกำหนดมาตรฐานการเชื่อมต่อแบบอนุกรม EIA RS-232 (x) เป็นมาตรฐานอุตสาหกรรมโดยคณะกรรมการสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association) ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง เพื่อให้มีการใช้งานในการเชื่อมต่อที่สอด -คล้องกันระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ การรับส่งสัญญาณจะกำหนดความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุตโดยมีระดับ สัญญาณตั้งแต่ 3 โวลต์ จนถึง 15 โวลต์ สำหรับลอจิก "0" และมีระดับแรงดันที่ -3 โวลต์ จนถึง -15 โวลต์ สำหรับลอจิก "1" ดังแสดงในรูป

ดังนั้นสังเกตได้ว่าจะมีระดับแรงดันที่ใช้ในสถานะลอจิก "0" และ ลอจิก "1" แตกต่างออกไปจากระบบไอซีดิจิตอลทั่วๆไป การต่อใช้งานจึงต้องมีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรงดันจาก 0 - 5 โวลต์ จากไมโครคอนโทรลเลอร์ ให้เป็นระดับแรงดันที่สูงกว่า +3 หรือต่ำกว่า - 3 โดยจะมีไอซีสำเร็จรูปพร้อมใช้งาน หรืออาจจะต่อวงจรจากทรานซิสเตอร์ได้



รูป 2.4 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ "0"

การสื่อสารข้อมูลแบบอนุกรมระหว่างคอมพิวเตอร์และอุปกรณ์อื่นๆ เกือบทั้งหมดเป็นแบบ Asynchronous ดังนั้นเราจะกล่าวถึงรายละเอียดเกี่ยวกับการสื่อสารข้อมูลแบบ Asynchronous เท่านั้น

Asynchronous Serial Communication

สำหรับการสื่อสารแบบอะซิงโครนัสนั้นจะใช้สายสัญญาณเพียงตัวเดียวแต่จะใช้รูปแบบในการส่งข้อมูล หรือ Bit Pattern เป็นตัวกำหนดว่าส่วนไหนเป็นตัวเริ่มต้นข้อมูล ส่วนไหนเป็นตัวข้อมูล ส่วนไหนจะเป็นตัวตรวจสอบความถูกต้องของข้อมูล และส่วนไหนเป็นส่วนปิดท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาครับและภาคส่ง ซึ่งจะมีอุปกรณ์พิเศษที่เรียกว่า UART หรือ Universal Asynchronous Receiver/Transmitter คอยควบคุมการรับและการส่งข้อมูล

การส่งข้อมูลจะมีลักษณะเป็น FRAME โดยแต่ละ frame ประกอบด้วย

START | DATA | PARITY | STOP

Start bit จำนวน 1 บิต โดยทั่วไปจะเป็น Low (0)

Data จำนวน 5-8 บิต

Parity bit จำนวน 0-1 บิต เป็นการตรวจสอบความถูกต้องของข้อมูล กำหนดเป็น even หรือ odd

Stop bit จำนวน 1-2 บิต โดยทั่วไปจะเป็น high (1)

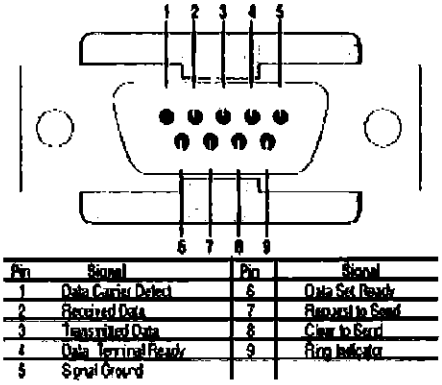
สัญลักษณ์ที่ใช้บอกลักษณะของข้อมูล เช่น 7E1: data 7 bits, even parity, 1 stop bit

8N1: data 8 bits, non-parity, 1 stop bit

เนื่องจากการสื่อสารแบบ Asynchronous ไม่มีการส่งสัญญาณ Clock ไปกับข้อมูล จึงเป็นต้องเซตสัญญาณความถี่สัญญาณ Clock ทางด้าน Rx และ Tx ให้เท่ากันเพื่อให้รับ-ส่งข้อมูลระหว่างกันได้ เรียกว่าค่า Bit rate หรือ Baud

อัตราการส่งข้อมูล (Baud rate) คือความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิทต่อวินาที เช่น 300, 1,200, 2,400, 4,800, 9,600, 14,400, 19,200, 38,400, 56,000 เป็นต้น ซึ่งการเลือกอัตราการส่งข้อมูลจะขึ้นอยู่กับชนิดของสายสัญญาณ, ระยะทาง, และปริมาณสัญญาณรบกวน

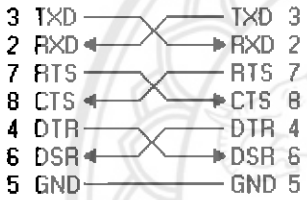
สำหรับการสื่อสารแบบอนุกรมบน PC รวมถึง Microcontroller จะถูกจัดการโดยส่วนที่เรียกว่า UART (Universal Asynchronous Receiver / Transmitter) ซึ่งจะทำหน้าที่รับ - ส่งข้อมูล Asynchronous โดยแปลงข้อมูลในรูปขนาน เป็นอนุกรม Asynchronous แล้วค่อยส่งออก จากนั้นภาครับก็จะรับอนุกรม Asynchronous มาแปลงเป็นขนานอีกทีหนึ่ง ในการเชื่อมต่อสายสัญญาณเข้ากับอุปกรณ์อนุกรมนิยมใช้หัวต่อชนิด DB9 และDB25 ซึ่งในที่นี้ จะใช้เฉพาะหัวต่อชนิด DB9 ดังรูปที่ 2.5



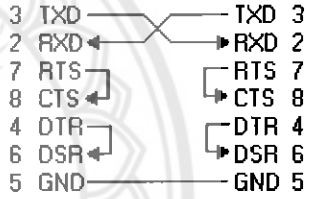
Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

รูปที่ 2.5 ขาสัญญาณเพอร์ตอนุกรมของ PC (DB9)

การเชื่อมต่ออุปกรณ์ ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9



รูปที่ 2.6 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem



รูปที่ 2.7 การต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น

การทำงานของขาสัญญาณ DB9

TXD เป็นขาที่ใช้ส่งข้อมูล

RXD เป็นขาที่ใช้รับข้อมูล

DTR แสดงสภาวะพอร์ตว่าเปิดใช้งาน, DSR ตรวจสอบว่าพอร์ต ที่ติดต่อด้วย เปิดอยู่หรือไม่

- เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าต้องการติดต่อด้วย
- ในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่

RTS แสดงสภาวะพอร์ตว่าต้องการส่งข้อมูล, CTS ตรวจสอบว่าพอร์ตที่ติดต่ออยู่ ต้องการส่งข้อมูลหรือไม่

- เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF
- ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่

GND ขา ground

การสื่อสารผ่านพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

เนื่องจากไมโครคอนโทรเลอร์ MCS-51 มีพอร์ตที่ใช้สำหรับในการสื่อสารที่สามารถรับ - ส่งข้อมูลแบบอนุกรมได้โดยที่ไม่ต้องทำการต่อวงจรเพิ่มเลย ซึ่งพอร์ตสื่อสารแบบอนุกรมใน MCS-51 นั้นสามารถทำงานในแบบ full duplex คือสามารถที่จะรับและส่งข้อมูลได้ในเวลาเดียวกัน และตัวไมโครคอนโทรลเลอร์จะมีการบัฟเฟอร์ข้อมูลให้ด้วย ซึ่งพอร์ตสื่อสารแบบอนุกรมใน MCS-51 นั้นจะประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิตสองตัว คือ

- รีจิสเตอร์สำหรับรับข้อมูลที่ส่งเข้ามาจากภายนอก
- รีจิสเตอร์สำหรับส่งข้อมูลจาก MCS-51 ออกไปภายนอก

โดยที่รีจิสเตอร์ทั้ง 2 ตัวนั้นจะมีตำแหน่งเดียวกันในรีจิสเตอร์สำหรับใช้งานเฉพาะ คือ จะตรงกับตำแหน่งรีจิสเตอร์ใช้งานเฉพาะ SBUF (ตำแหน่ง 99H) ในหน่วยความจำ สำหรับเก็บข้อมูลภายในชิปที่ใช้เป็นรีจิสเตอร์ใช้งานเฉพาะ ซึ่งการเข้าถึงข้อมูลในรีจิสเตอร์แต่ละตัวนั้น MCS-51 จะทำการตรวจสอบรหัสคำสั่งว่าผู้ใช้ต้องการติดต่อกับรีจิสเตอร์ตัวใด เพราะในการเขียนข้อมูลไปไว้ในรีจิสเตอร์ SBUF หมายถึงการโหลดข้อมูลไปที่รีจิสเตอร์สำหรับส่งข้อมูล เพื่อส่งข้อมูลออกไปภาย - นอก ส่วนการอ่านข้อมูลจากรีจิสเตอร์ SBUF จะหมายถึงการนำค่าที่รับเข้ามาจากภายนอกมาเก็บไว้ในรีจิสเตอร์สำหรับรับข้อมูลมาใช้งาน

รีจิสเตอร์ที่เกี่ยวข้องในการทำงานของพอร์ตอนุกรม

รีจิสเตอร์ SCON (98H)

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

รีจิสเตอร์ TMOD (89H)

Bit	7	6	5	4	3	2	1	0
Name	GATE (1)	C/T (1)	M1 (1)	M0 (1)	GATE (0)	C/T (0)	M1 (0)	M0 (0)

รีจิสเตอร์ TCON (88H)

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

ตารางที่ 2.3 ตารางค่าคงที่ๆ กำหนดจากการใช้ ไทม์เมอร์1 เมื่อใช้อัตราความเร็วในการส่งข้อมูลมาตรฐาน

Baud rate	Fosc (MHz)	บิต SMOD	Timer1		
			บิต C/T	MODE	Reload Value
62.5 K	12.00	1	0	2	FFH
19.25 K	11.059	1	0	2	FDH
9600	11.059	0	0	2	FDH
4800	11.059	0	0	2	FAH
2400	11.059	0	0	2	F4H
1200	11.059	0	0	2	E8H
137.5	11.059	0	0	2	1DH
110	6	0	0	2	72H
110	12	0	0	1	FEEBH

- รีจิสเตอร์บัฟเฟอร์ SBUF (Serial data buffer register) มีตำแหน่งอยู่ที่ 99H แบ่งออกเป็นสองส่วนคือ ส่วนที่ใช้เป็นบัฟเฟอร์ในการส่งข้อมูล และส่วนที่ใช้เป็นบัฟเฟอร์ในการรับข้อมูล
- รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม SCON (Serial port Control Register) มีตำแหน่งอยู่ที่ 98H ,บิต TI และบิต RI
- รีจิสเตอร์เลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์ TMOD (Timer/Counter Mode Control Register) มีตำแหน่งอยู่ที่ 89H
- รีจิสเตอร์ TH1 (Timer 1 High) กำหนดค่าของไทเมอร์ 1 เพื่อเลือก baud rate มีตำแหน่งอยู่ที่ 8Dh
- รีจิสเตอร์ควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ TCON (Timer/Counter Control Register) มีตำแหน่งอยู่ที่ 88H และบิต TR1 เป็นบิตควบคุม การทำงานของไทเมอร์

การกำหนดค่าเริ่มต้นสำหรับการใช้งานพอร์ตอนุกรม (Initial Serial Port)

สำหรับในการทดลองจะใช้ Baud Rate 9600, ข้อมูลรับ-ส่งมีขนาด 8 บิต, 1 Stop bit และไม่มีตรวจสอบ Parity บิต ซึ่งเป็นมาตรฐานของการใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์โดยทั่วไป

2.6 ส่วนประกอบของวงจร

2.6.1 Microcontroller Single Board (STARTC-51) [4-5]

Microcontroller

Microcontroller Single Board (STARTC-51) เป็นไมโครคอนโทรลเลอร์ในตระกูลMCS-51 ซึ่งเป็นตระกูลที่มีทางเลือกจากผู้ผลิตชิพมากที่สุดในโลก และเป็นไมโครคอนโทรลเลอร์ที่ออกแบบมาเพื่อสนองความต้องการของผู้ใช้แบบสำเร็จในไอซีตัวเดียว คือ มีสายสัญญาณอินพุตและเอาต์พุตภายในตัว พอร์ตของอินพุตและเอาต์พุตบัฟเฟอร์ที่ใช้เชื่อมต่อกับวงจรภายนอก (interface) และสายสัญญาณควบคุมอื่น ๆ ที่ใช้สำหรับแยกสายสัญญาณข้อมูลกับสายสัญญาณกำหนดตำแหน่งหน่วยความจำ และยังมีชุดคำสั่งพิเศษเพื่อจัดการข้อมูลเพิ่มขึ้นอีกนอกจากนั้นยังมีวงจรนับเวลาและตั้งเวลาด้วย

ข้อสำคัญ คือ มีการพัฒนาภายในไมโครคอนโทรลเลอร์ให้มีหน่วยความจำเป็นแบบ Flash (Flash Memory) ทำให้สามารถโปรแกรมข้อมูลในหน่วยความจำโปรแกรมได้โดยไม่ต้องถอดตัวไมโครคอนโทรลเลอร์ออกจากวงจร เรียกว่าการโปรแกรมภายในวงจร (In-System Programming) และมีการติดต่อแบบ ISP ทำให้การพัฒนาและการแก้ไขปรับปรุงโปรแกรมทำได้สะดวก

หน่วยความจำของ STARTC-51

หน่วยความจำของ STARTC-51 ประกอบด้วย 3 ส่วนดังนี้

1. Internal RAM ขนาด 256 Byes และส่วน SFR ภายในตัวชิพ 89C51RD2
2. Program Memory (Flash) ขนาด 64 Kbytes ภายในตัวชิพ 89C51RD2 (เป็นพื้นที่สำหรับให้ผู้พัฒนาโปรแกรมจำนวน 48 Kbytes)
3. Data Memory (SRAM) ขนาด 768 Byes ภายในตัวชิพ 89C51RD2 หรือขยายเพิ่มเติมเป็น 8,32 Kbytes ด้วย RAM ภายนอกเบอร์ 6264 , 62256

ส่วนประกอบหลัก

ส่วนประกอบหลักทาง Hardware ของ STARTC-51 ที่เกี่ยวข้องมีดังนี้

MCU เป็นหัวใจการทำงานของบอร์ดเรียกว่าไมโครคอนโทรเลอร์ คือเบอร์ 89C51RD2 ขนาด 40 ขา คุณสมบัติคือ มีพื้นที่ ROM ที่เป็น Flash Memory ขนาด 64 Kbytes ที่สามารถ Download Code โปรแกรมจากเครื่อง PC ผ่านพอร์ทอนุกรมของตัว MCU ลงสู่ Flash ROM ได้ทันทีโดยไม่ต้องใช้เครื่อง BURN ใด ๆ มาช่วยให้ยุ่งยาก และยังช่วยให้การพัฒนาโปรแกรมทำได้ง่ายขึ้นและสะดวกสบายขึ้นโดยไม่ต้องถอดตัวชิพออกจากบอร์ดเลย เรียกวิธีการโปรแกรมชิพแบบนี้ว่า ISP (In-System Programming)

MEMORY เป็นหน่วยความจำบนบอร์ดที่สามารถใช้งานได้ทั้งหมด จะแยกอยู่ในตำแหน่งต่างๆ บนบอร์ด ดังนี้

Internal RAM คือมีพื้นที่หน่วยความจำ RAM ภายในตัว MCU ขนาด 256 Byes อ้างถึงตำแหน่ง Address ตั้งแต่ 00-FFH โดยการเข้าถึงหน่วยความจำทำได้ดังนี้คือ ที่ Address 00-7FH จะเข้าถึงแบบ Direct หรือ Indirect (MOV 30H, 42H) ส่วน Address 80H-DFH จะต้องเข้าถึงแบบ Indirect เท่านั้น (MOV @R0, A)

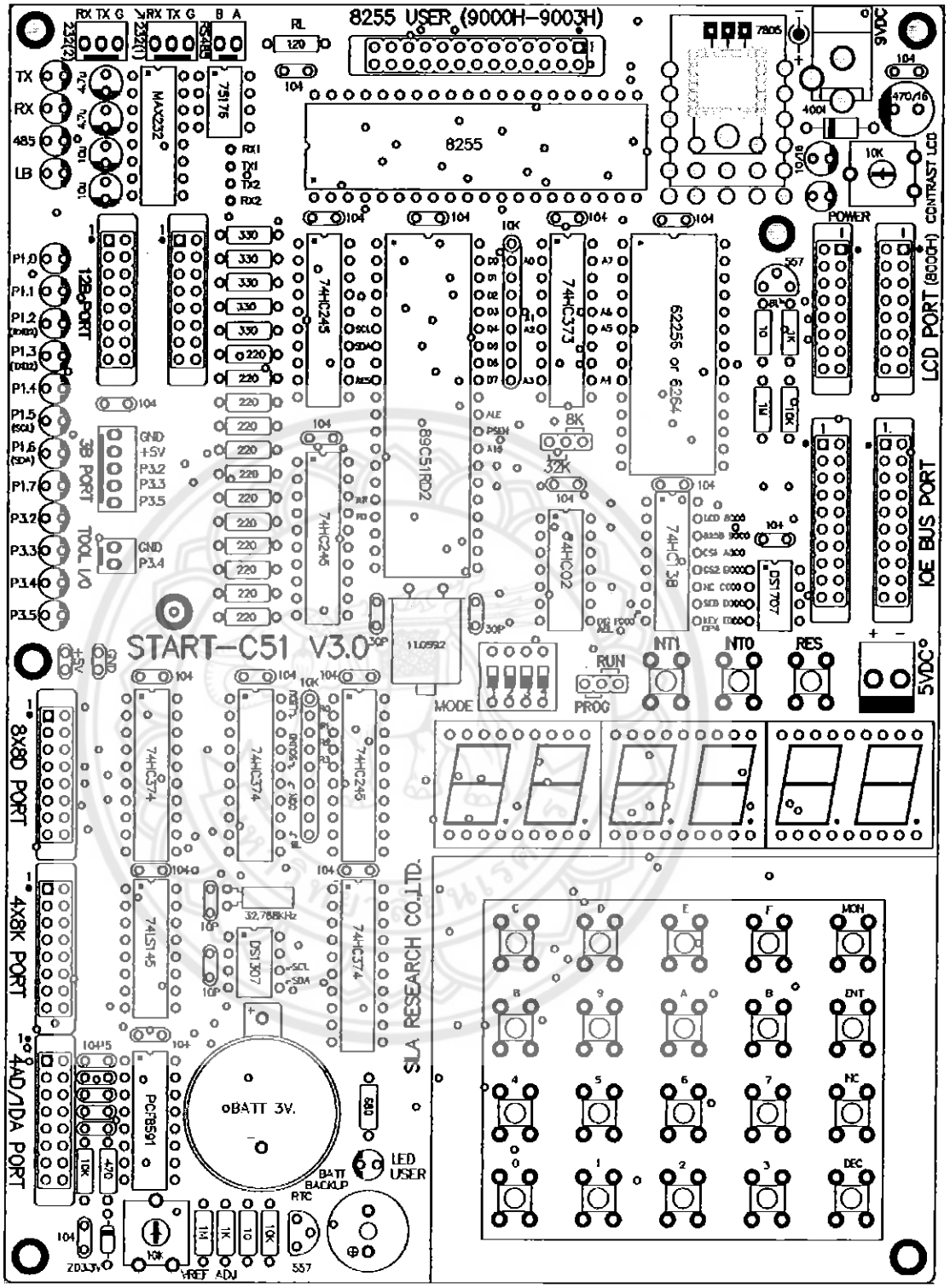
Data Memory (Internal) คือพื้นที่หน่วยความจำ RAM เพิ่มเติมภายในตัว 89C51RD2 อ้างถึงตำแหน่ง Address ตั้งแต่ 000-2FFH การเข้าถึงหน่วยความจำนี้จะกระทำได้โดยเข้าถึงแบบ Index Addressing (MOVX @DPTR,A OR MOV @R0,A) และต้อง Clear Flag EXT-RAM=0 ที่ ตำแหน่ง Address 8EH ก่อน

Program Memory คือพื้นที่ส่วนที่ใช้เก็บ Code Program จะอยู่ภายในตัว MCU ขนาด 64 Kbytes มี Address ตั้งแต่ 0000-FFFFH รวมถึงพื้นที่ที่ใช้เก็บโปรแกรม MONITER ของบอร์ดที่ Address C000-FFFFH ส่วนพื้นที่ที่ผู้ใช้สามารถใช้ได้อยู่ที่ Address ตั้งแต่ 0000-BFFFH ขนาด 48 Kbyes และโปรแกรมที่ผู้ใช้พัฒนาขึ้น การโหลดจะโหลดโปรแกรมเข้าส่วนนี้จะต้องDownload ผ่านพอร์ท RS-232

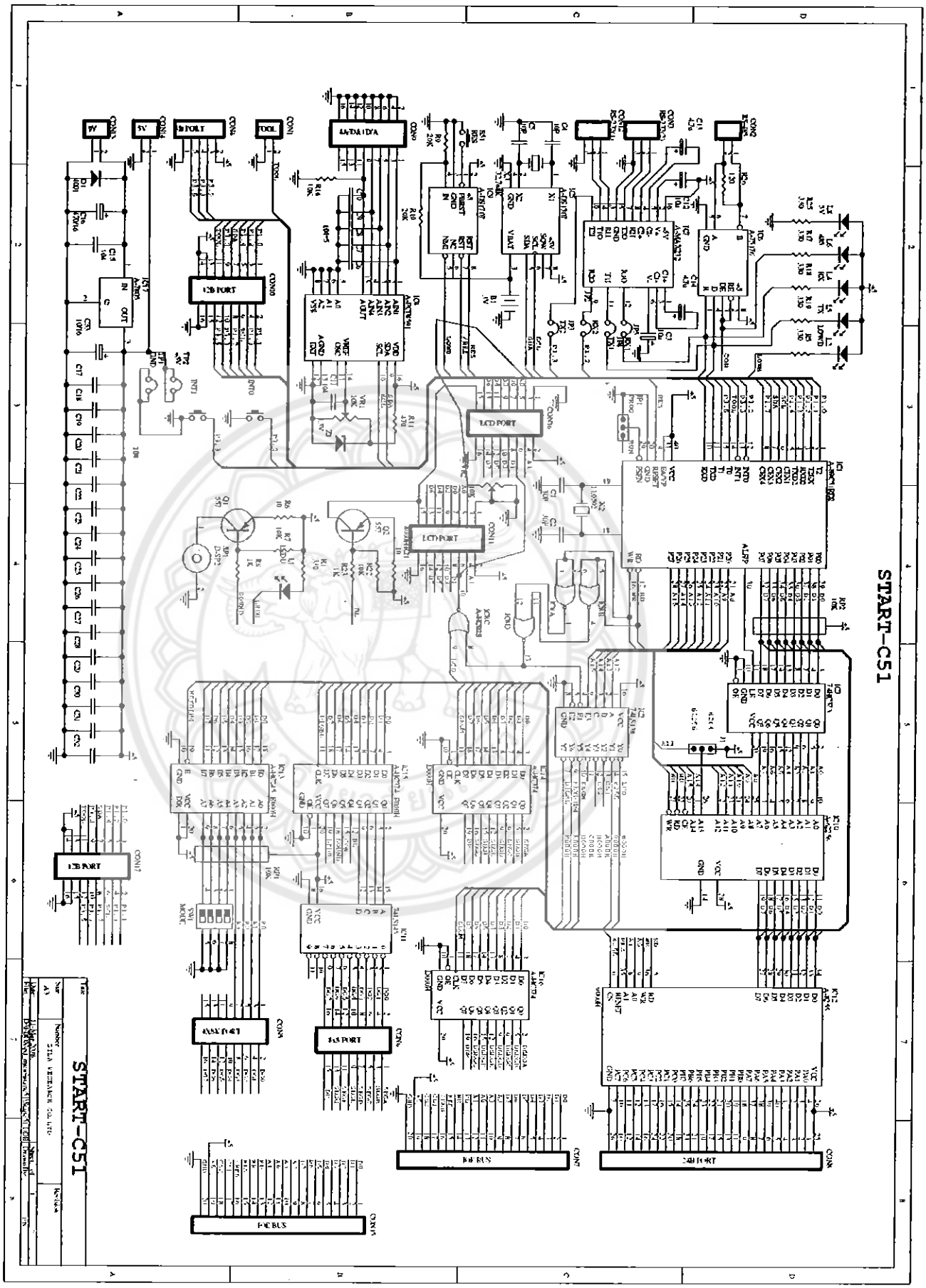
RS-232 PORT คือพอร์ทสื่อสารข้อมูลแบบอนุกรม ซึ่งมีอยู่ 2 พอร์ทตามมาตรฐาน RS-232 ได้แก่ ขา RXD1 TXD1 และ RXD2 TXD2 (กรณีที่ใช้ MCU DS89C420 และต้องบัดกรี Jump เมื่อต้องการใช้งาน RXD2 TXD2) ผ่านชิพเบอร์ MAX 232 เพื่อปรับระดับแรงดันไฟ การใช้งานจะใช้เป็นพอร์ทสื่อสารข้อมูลกับเครื่อง PC และใช้เพื่อโปรแกรมโค้ดเข้า Flash ROM ของ MCU ทางพอร์ท RXD1 และ TXD1 ด้วย

SUPPLY ขั้วต่อไฟเลี้ยงบอร์ด สามารถป้อนเข้าได้ 2 ทาง คือ 9 VCD จะใช้กับ Adapter ที่มากับบอร์ดแรงดันจะถูก Regulate ด้วย 7805 อีกที กับอีกทางคือที่ขั้ว 5 VCD จะต้องเป็นแรงดันที่คงที่ 5V เท่านั้น และจะไม่ผ่าน Regulate ภายในบอร์ด กรณีที่ใช้ 9 VCD ที่ช่อง 5 VCD จะสามารถจ่ายไฟไปยังบอร์ดอื่นได้อีกแต่จะต้องกินกรแสไม่เกิน 300mA

Area STARTC-51 นี้ถูกออกแบบให้มี PCB Working สำหรับเสียบลอยขึ้นมาจากบอร์ดหลัก โดยจะเชื่อมพอร์ท 12B PORT, พอร์ท LCD และพอร์ท IOE BUS ให้พร้อมต่อการใช้งานได้ทันที ซึ่งทำให้สะดวกในการต่อวงจรทดลองมากขึ้น



รูป 2.8 โครงสร้างของ Microcontroller Single Board (STARTC-51)



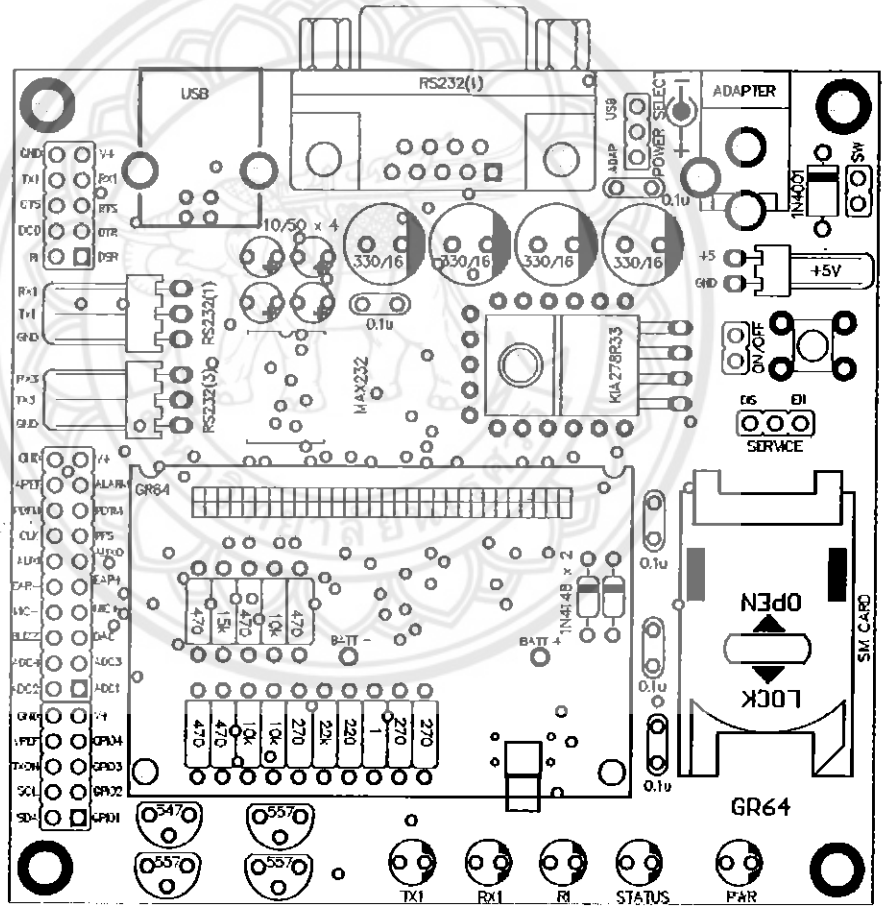
รูป 2.9 แสดงวงจรของ Microcontroller Single Board (STARTC-51)

2.6.2 GR-64 (GR 64 GSM/GPRS Module) [6]

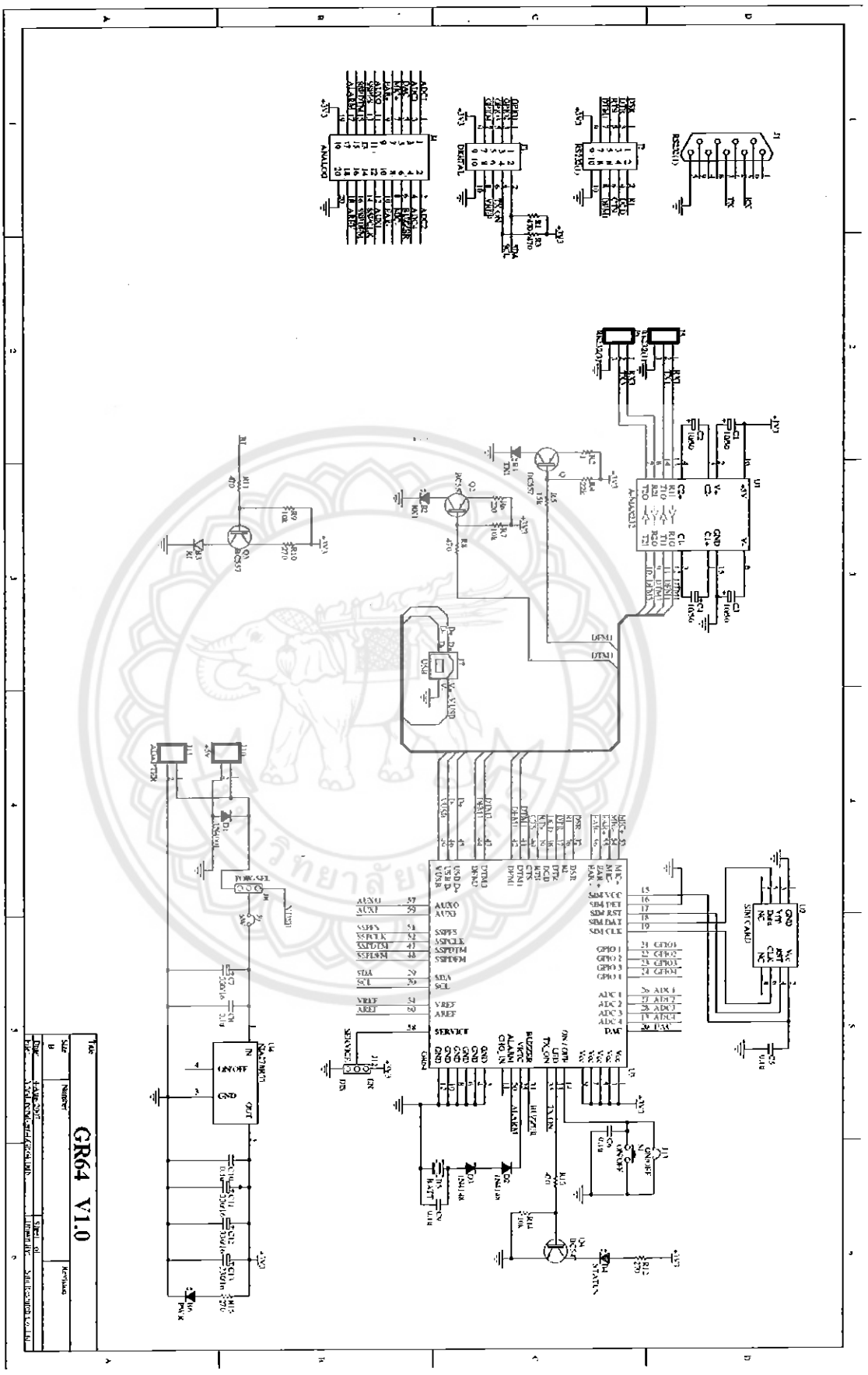
GR 64 เป็นโมดูล GSM / GPRS จะใช้กับ GSM-Sim มือถือได้ทุกเครือข่าย รองรับ 4 ความถี่ ได้แก่ 850/900/1800/1900 MHz

GR 64 เป็นเหมือนโมเด็มไร้สายที่ถูกพัฒนาขึ้นเพื่อใช้ในการรับ-ส่งข้อมูลโดย SMS , CSD , GPRS หรือใช้อย่างโทรศัพท์เคลื่อนที่ทั่ว ๆ ไป แต่มีลักษณะพิเศษตรงที่ว่า GR 64 สามารถรองรับการสื่อสารผ่านพอร์ตอนุกรมได้ จึงนำมาใช้เป็นเครื่องรับข้อความแทนโทรศัพท์เคลื่อนที่ทั่ว ๆ ไป

ในการใช้งาน GR 64 นี้สามารถติดต่อโดยผ่านทาง RS - 232 (ขั้ว 3 Pin หรือ DB9) ด้วยชุดคำสั่ง AT COMMAND ซึ่งสามารถใช้โปรแกรม STERM ซึ่งในที่นี้จะใช้โปรแกรม HyperTerminal ในการทำงาน



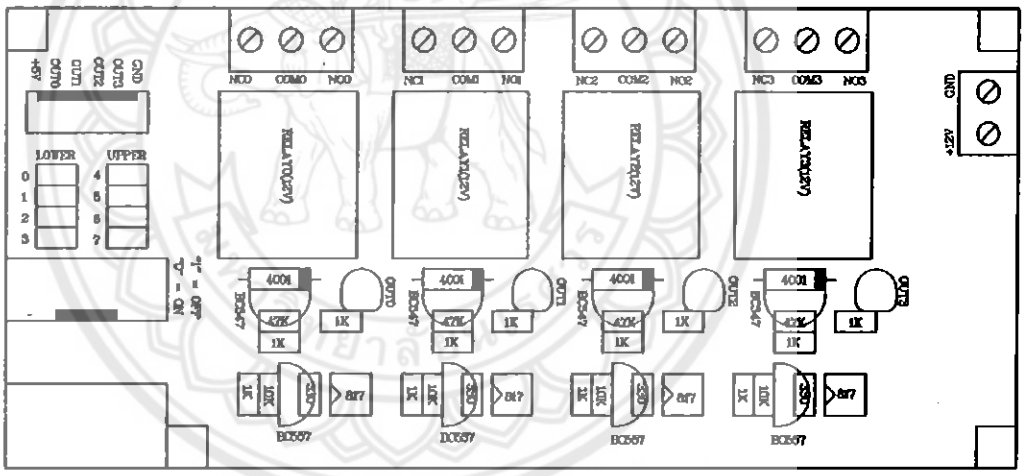
รูป 2.10 โครงสร้างของ GR 64 GSM/GPRS Module



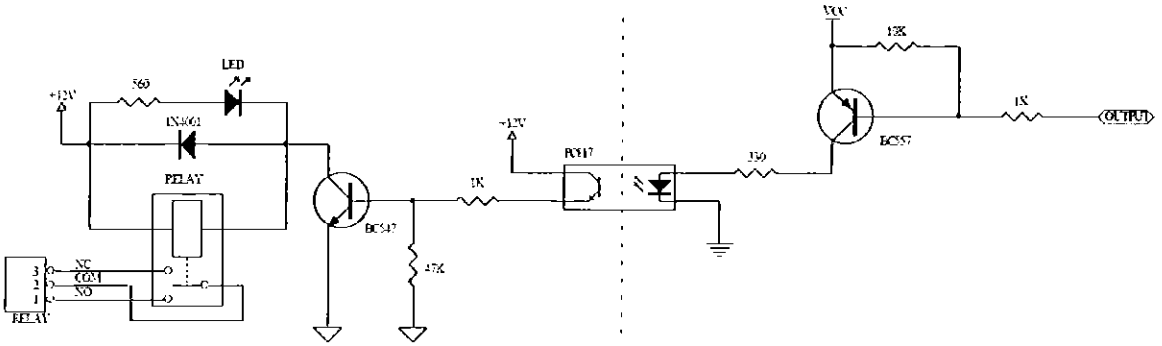
รูป 2.11 แสดงวงจรของ GR 64 GSM/GPRS Module

2.6.3 ET-OPTO RELAY 4 [7-8]

บอร์ด ET-OPTO RELAY 4 เป็นชุด OPTO-ISOLATE OUTPUT แบบหน้าสัมผัส RELAY ขนาด 4 ช่อง ใช้สำหรับควบคุมการทำงานของอุปกรณ์ไฟฟ้าต่าง ๆ ในลักษณะของหน้าสัมผัส ON/OFF โดยใช้สัญญาณในการควบคุมการทำงานของ RELAY ด้วยสัญญาณโลจิก TTL ซึ่งวงจรสำหรับควบคุมการทำงานของ RELAY นั้น จะใช้วงจรแบบ OPTO-ISOLATE ในการควบคุมการทำงาน เพื่อป้องกันสัญญาณรบกวนต่าง ๆ ที่เกิดจากการทำงานของ RELAY และอุปกรณ์ output ที่ควบคุมการทำงานจากหน้าสัมผัส RELAY ที่จะย้อนกลับมารบกวนการทำงานของวงจรควบคุมหรือไมโครคอนโทรเลอร์ต่างๆ ทำให้ประสิทธิภาพการทำงานของระบบไมโครคอนโทรลอร์ต่าง ๆ มีความน่าเชื่อถือมากยิ่งขึ้น โดยการทำงานของวงจรนั้นจะออกแบบให้การทำงานของ RELAY ทำงานด้วยสภาวะโลจิก “0” และหยุดการทำงานด้วยสภาวะโลจิก “1” เพื่อป้องกันไม่ให้ RELAY ทำงานในขณะที่ระบบไมโครคอนโทรลเลอร์เกิดการ RESET หรือสัญญาณควบคุมของบอร์ด ET-OPTO RELAY 4 ถูกปล่อยลอยอยู่



รูป 2.12 โครงสร้างของบอร์ด ET-OPTO RELAY 4



รูป 2.13 แสดงวงจรของ ET-OPTO RELAY4 ขนาด 1 ช่อง

การทำงานของบอร์ด

การใช้งานบอร์ด ET-OPTO RELAY 4 เพื่อควบคุมการทำงานของอุปกรณ์ต่างๆ ในลักษณะของหน้าสัมผัสสวิตช์นั้น จะต้องมีการป้อนแรงดันไฟตรงขนาด +12VCD เพื่อใช้เป็นไฟเลี้ยงขดลวดของ RELAY ด้วย ซึ่งแรงดัน +12VCD นี้ ควรจะเป็นแรงดันไฟตรงที่ได้จากหม้อแปลง ซึ่งแยกขดกับชุดที่จะจ่ายให้กับบอร์ดไมโครคอนโทรเลอร์ด้วย เพื่อป้องกันการรบกวนที่จะเกิดขึ้นกับวงจรไมโครคอนโทรลเลอร์ ในขณะที่ RELAY หรืออุปกรณ์ output ที่ถูกควบคุมจาก RELAY ทำงาน

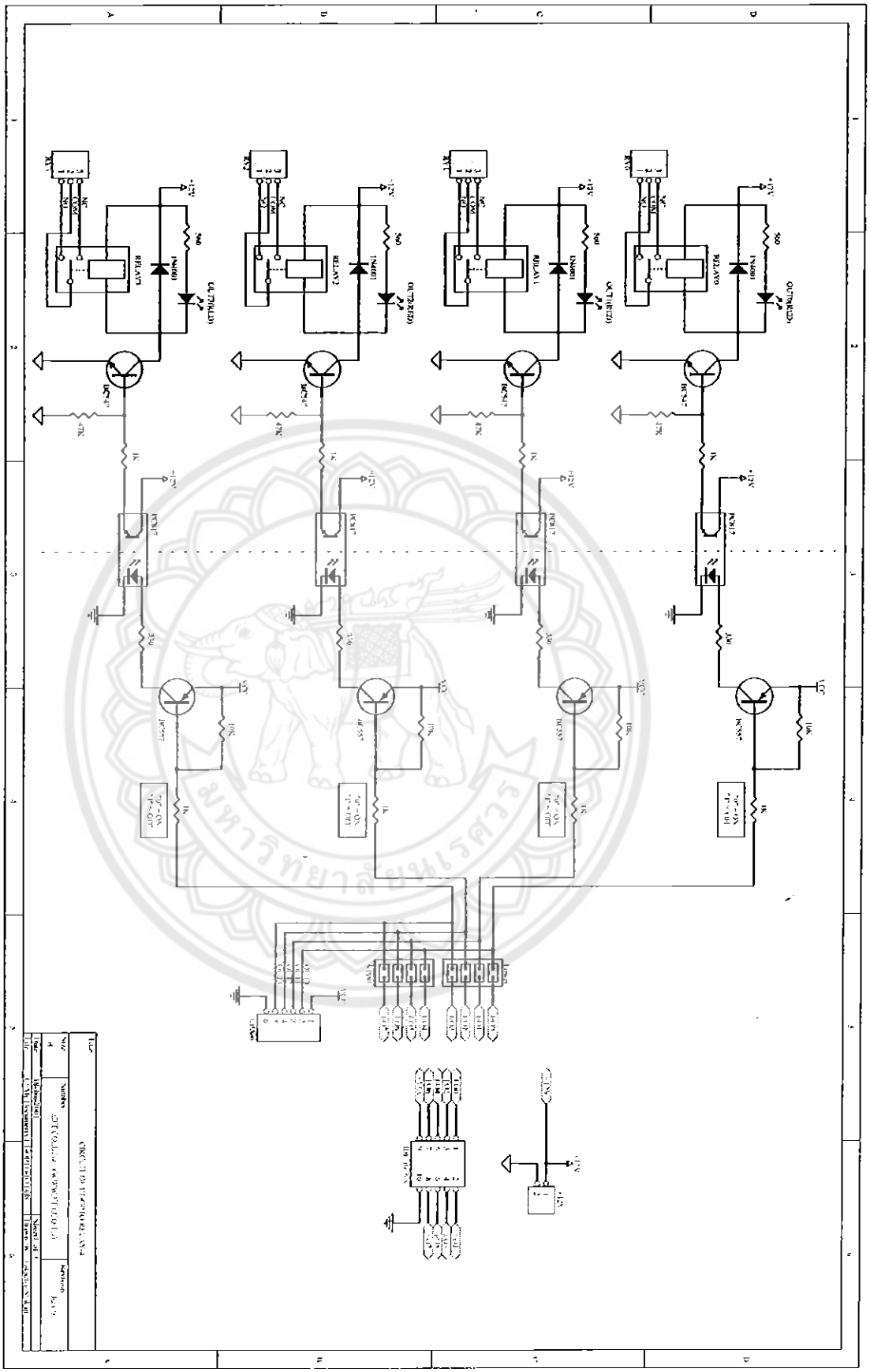
ซึ่งการทำงานของบอร์ดจะเริ่มจากเมื่อมีการป้อนสถานะโลจิก "0" ให้กับจุด OUTPUT ของวงจร ซึ่งจะส่งผลให้ขา BASE ของทรานซิสเตอร์ BC557 ที่ต่ออยู่กับภาค Output ของ OPTO-ISOLATE (PC817) ได้รับการไบอัสที่ถูกต้อง ทำให้ทรานซิสเตอร์นำกระแสซึ่งจะทำให้มีแรงดัน +VCC (+5VDC) ไหลออกจากขา Emitter ไปยัง Collector ผ่านตัวต้านทาน 330 โอห์ม เพื่อจำกัดกระแสให้กับ OPTO ISOLATE (PC817) ซึ่งส่งผลให้วงจร OPTO ISOLATE ทำงาน เป็นผลให้เกิดมีกระแสไหลจากแหล่งจ่าย +12VCD ผ่านภาค OUTPUT ของ OPTO ISOLATE (PC817) โดยแรงดัน+12VCDนี้จะถูกจำกัดขนาดของกระแสด้วยตัวต้านทานค่า 1KOHM ก่อนที่จะป้อนให้กับขา Base ของทรานซิสเตอร์ BC547 ที่ต่ออยู่กับวงจรภาค Output ของ OPTO ISOLATE เป็นผลให้ทรานซิสเตอร์ BC547 นำกระแส ซึ่งก็จะทำให้มีกระแสไฟฟ้าไหลผ่านขดลวดของ RELAY ทำให้ RELAY และ LED ทำงาน

โดยในการประยุกต์ใช้งานบอร์ดนั้น จะใช้สำหรับควบคุมการทำงานของอุปกรณ์ต่าง ๆ ที่ต้องการควบคุมการทำงานด้วยวิธีการ ON/OFF ของหน้าสัมผัส

โดยลักษณะของหน้าสัมผัส RELAY จะมีให้เลือกใช้งาน 2 แบบคือ NO และ NC ในการต่อใช้งานจะต้องเลือกแบบใดแบบหนึ่ง โดยจะต่อใช้งานร่วมกับขา COM (Common)

- NO หรือ Normal Open ซึ่งหมายถึงหน้าสัมผัสแบบปกติเปิดและจะทำงานเมื่อ RELAY เริ่มต้นทำงาน
- NC หรือ Normal Close ซึ่งหมายถึงหน้าสัมผัสแบบปกติปิดและจะหยุดทำงานเมื่อRELAY เริ่มทำงาน

สำหรับการเชื่อมต่อสัญญาณ TTL ไปให้กับวงจรภาค Output ของไมโครคอนโทรเลอร์นั้น จะสามารถกระทำได้ 2 แบบ ด้วยกันคือ ต่อผ่านขั้วต่อแบบ IDE ขนาด 10 Pin โดยใช้สายแพร์เป็นสายนำสัญญาณ โดยสามารถเลือกกำหนดสัญญาณ TTL ที่ได้จากการทำงานของบอร์ด ได้ 2 กลุ่มคือ LOWER และ UPPER ส่วนอีกวิธีหนึ่งจะสามารถกระทำได้โดยต่อสัญญาณผ่านขั้วต่อแบบ CPA ขนาด 6 Pin

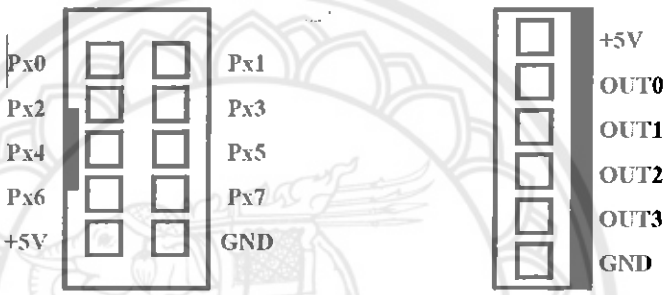


รูป 2.14 แสดงวงจรของ ET-OPTO RELAY4

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
| 4942425

ตาราง 2.4 ตารางแสดงจุดเชื่อมต่อสัญญาณของบอร์ด ET-OPTO RELAY 4 กับไมโครคอนโทรเลอร์

ขั้วต่อ Output (Terminal)	LOWER		UPPER	
	IDE 10 Pin	CPA 6 Pin	IDE 10 Pin	CPA 6 Pin
RELAY0	Px0	OUT0	Px4	OUT0
RELAY1	Px1	OUT1	Px5	OUT1
RELAY2	Px2	OUT2	Px6	OUT2
RELAY3	Px3	OUT3	Px7	OUT3



รูป 2.15 แสดงการจัดเรียงสัญญาณของขั้ว IDE 10 Pin CPA 6 Pin

2.6.4 ภาษา Assembly [9]

ภาษา Assembly เป็นภาษาที่ใช้สัญลักษณ์ในการสื่อความหมาย ภาษา Assembly มีลักษณะคำสั่งที่ขึ้นกับเครื่องคอมพิวเตอร์ที่ใช้งานและมีการแปลคำสั่งให้เป็นภาษาเครื่อง นอกจากภาษาเครื่อง และภาษา Assembly แล้ว ยังมีภาษาระดับสูง เช่น Basic Cobol FORTRAN ซึ่งเป็นภาษาที่มีคำสั่งใกล้เคียงกับภาษาอังกฤษมาก ซึ่งทำให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้สะดวกและรวดเร็วแต่ว่าโปรแกรมที่เขียนด้วยภาษาระดับสูงต้องใช้เนื้อที่เก็บในหน่วยความจำเป็นจำนวนมาก อีกทั้ง ทำงานได้ช้ากว่าภาษา Assembly ดังนั้นภาษาระดับสูงจึงไม่นิยมที่จะนำมาประยุกต์ใช้กับการทำงานที่ระบบการควบคุมที่มีความสำคัญมาก ภาษา Assembly จึงเหมาะกับโปรแกรมที่ใช้เนื้อที่ในหน่วยความจำไม่มากนัก ทั้งทำงานได้รวดเร็วในการควบคุมการทำงานของเครื่องคอมพิวเตอร์ได้โดยตรง

โครงสร้างของโปรแกรมAssembly

โปรแกรม Assembly จะถูกเก็บไว้ในหน่วยความจำซึ่งอยู่ในรูปแบบของเลขไบนารี่ที่เรียกว่าภาษา Machine ซึ่งเป็นภาษาที่สามารถติดต่อให้คอมพิวเตอร์เข้าใจได้ ภาษา Machine นี้จะจัดให้อยู่

ในรูปแบบที่เป็นเลขฐานสิบหก(HEX)เช่น คำสั่ง 8 บิต 11101011B (B-ไบนารี่) เขียนได้เป็น 0EBH (H-ฐานสิบหก) แต่ก็เป็นการที่จะเข้าใจความหมายได้ยากในการใช้งาน การที่จะทำความเข้าใจภาษา Machine จะมีการใช้สัญลักษณ์ (Symbols) ที่เรียกว่า Mnemonics เพื่อแทนความหมายของคำสั่ง เช่น MOV A,#67H หมายความว่านำข้อมูลค่าคงที่ 67H ไปเก็บไว้ใน reg. A) โปรแกรมที่เขียนด้วยรหัส Mnemonics เรียกว่า ภาษา Assembly และก่อนที่จะให้ CPU ทำงานตามโปรแกรมที่เขียนด้วยภาษา Assembly ได้ แต่ต้องเปลี่ยนให้เป็นภาษา Machine ก่อนโดยใช้โปรแกรม Assembler

รูปแบบของภาษา Assembly

โปรแกรมภาษา Assembly ประกอบด้วยส่วนต่าง ๆ ดังนี้

1. Machine Instructions คือชุดคำสั่งภาษาเครื่องโดยแทนด้วยรหัส Mnemonics เช่น MOV ,ANL เป็นต้น
2. Assembler Directives หรือคำสั่งเทียมเป็นคำสั่งที่กำหนดขึ้นโดยไม่ต้องถูกเปลี่ยนแปลงเป็นรหัสภาษาเครื่อง มีไว้สำหรับกำหนดโครงสร้างโปรแกรม ,สัญลักษณ์ ,ข้อมูลต่าง ๆ เช่น ORG ,EQU เป็นต้น
3. Assembler Controls เป็นชุดคำสั่งในการควบคุมของตัว Assembler
4. Comments เป็นการเขียนคำอธิบายเข้าไปในโปรแกรม

ในการเขียนโปรแกรมในภาษา Assembly ในแต่ละบรรทัดมีส่วนประกอบที่สำคัญ 4 ส่วน คือ Label Mnemonics Operand Comments โดยรูปแบบทั่วไปจะเป็นดังนี้

[Label:] Mnemonics [Operand] [,Operand] [...] [; Comments]

Label Field

ในภาษาแอสแซมบลี Label จะเป็นตัวแทนตำแหน่งของชุดคำสั่งหรือข้อมูลที่ตามหลัง Label อยู่ ในคำสั่งกระโดดหรือคำสั่งต่าง ๆที่ต้องมีการอ้างตำแหน่ง สามารถแทนค่าตำแหน่งด้วย Label ได้เลย เช่น SJMP SKIP เป็นการกระโดดไปที่ตำแหน่งที่แทนด้วย Label SKIP

โดยทั่วไปแล้ว Label Field แทนได้สองความหมายคือ แทนค่าตำแหน่งและเป็น "Symbol" ซึ่งจะแทนค่าสัญลักษณ์ ถ้าหากจะใช้เป็น Label จะตามด้วยเครื่องหมาย Colon " : " ถ้าเป็น Symbol จะตามด้วยคำสั่งเทียม พิจารณาคำสั่งต่อไปนี้จะใช้ PAR แทน Symbol และ START แทน Label

```
PAR      EQU  500          ; " PAR " เป็นสัญลักษณ์แทนด้วยค่า 500
START :  MOV , #0FFH      ; " START " เป็นลาเบลจะมีความหมายแทน
                            ตำแหน่งที่เก็บคำสั่ง MOV
```

การตั้งชื่อจะต้องขึ้นต้นด้วยตัวอักษร "A-Z" หรือเครื่องหมาย " _ ", " . " จะเป็นตัวอักษรจาก A ถึง Z หรือตัวเลขจาก 0 ถึง 9 หรือเครื่องหมาย " ? " และจะยาวกี่ตัวอักษรก็ได้ แต่เมื่อรวมทั้งบรรทัดแล้วต้องไม่เกิน 255 ตัวอักษร

Mnemonics Field

ส่วน Mnemonics Field เป็นชุดคำสั่งหรือคำสั่งเทียมก็ได้ ซึ่งจะเขียนตามหลัง Label ตัวอย่างของชุดคำสั่งได้แก่ ADD , MOV , DIV , INC ตัวอย่างคำสั่งเทียมได้แก่ ORG , EQU , DB

Operand Field

ในการเขียน Mnemonics เป็นตัวบอกว่าจะให้ Mnemonics ทำอะไร เช่น MOV จะเป็นตัวบอกว่าให้ย้ายข้อมูล แต่ยังไม่มีตำแหน่งที่จะย้ายว่าจะย้ายจากไหนไปไหน ส่วนที่จะบอกว่าจะให้ทำกับอะไรเรียกว่า Operand ซึ่งจะเขียนตามหลังรหัส Mnemonics ใน Operand Field นี้ประกอบไปด้วย Address หรือ Data ที่จะใช้กับชุดคำสั่ง ซึ่งอาจจะเป็น Label ที่แทนตำแหน่งของข้อมูลหรือ Symbol ที่แทนค่าข้อมูลด้วยก็ได้ คำสั่งส่วนใหญ่จะมี Operand Field ตามหลัง แต่บางคำสั่งจะไม่มี เช่น คำสั่ง RET

Comment Field

เป็นการเขียนคำอธิบายแต่ละบรรทัดหรือหมายเหตุนั้นและที่สำคัญจะต้องใช้เครื่องหมาย ";" (Semi-Colon) นำหน้าด้วยทุกครั้ง

บทที่ 3
วิธีการดำเนินงานโครงงานวิศวกรรม

ในการจัดทำโครงงานเรื่อง ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS ประกอบไปด้วยวิธีการดำเนินงาน 4 ขั้นตอน ดังนี้

3.1 การวิเคราะห์ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS

3.2 การออกแบบระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS

3.3 การพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS

3.4 การทดสอบและประเมินผลการดำเนินโครงงาน

3.1 การวิเคราะห์ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS

ในขั้นแรกของการดำเนินโครงงาน คือการวิเคราะห์ระบบควบคุมอุปกรณ์ไฟฟ้า ดังนี้

1. วิเคราะห์ปัญหา อันเป็นที่มาของระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS
2. วิเคราะห์องค์ประกอบของระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS จากเอกสารและงานวิจัยที่เกี่ยวข้อง
3. วิเคราะห์การทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS โดยพิจารณาจากความต้องการของระบบ

3.2 การออกแบบระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS

ในขั้นตอนของการออกแบบระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS แบ่งออกเป็น 2 ขั้นตอน คือ

1. การออกแบบวงจรควบคุมการทำงานของอุปกรณ์ไฟฟ้า
2. การออกแบบการทำงานของโปรแกรม

3.3 การพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS

ในขั้นตอนของการพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS นั้น สามารถแบ่งการพัฒนาออกเป็นสองด้านคือ

1. การพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS ด้านวงจรควบคุมอุปกรณ์ไฟฟ้า
2. การพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS ด้านการโปรแกรม

โดยเริ่มจากการพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS ด้านโปรแกรมควบคุมการทำงานของอุปกรณ์ไฟฟ้า ดังนี้

3.3.1 โปรแกรมควบคุมการทำงาน

หลังจากทดสอบจนแน่ใจแล้วว่า STARTC-51 และ GR-64 สามารถสื่อสารกัน ผ่านทางพอร์ทอนุกรม RS-232 ได้แล้ว ขั้นต่อไปคือการพัฒนาโปรแกรมภาษา Assembly เพื่อควบคุมการทำงานของ LED P1.0 – P1.7 บน START-C51 โดยการนำชุดคำสั่ง AT COMMAND มาประยุกต์ใช้งานกับโปรแกรมภาษา Assembly เริ่มต้นจากกำหนดรูปแบบข้อความที่จะใช้ส่งเข้ามา ซึ่งได้กำหนดไว้ดังนี้

NUOxxxxCxxxxSE

NU หมายถึง Start เป็นการเริ่มต้นรูปแบบข้อความ

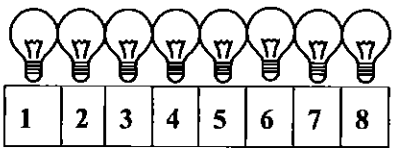
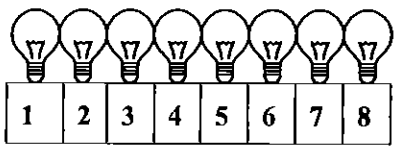
O หมายถึง Open เป็นรูปแบบในการสั่งให้เปิดไฟ

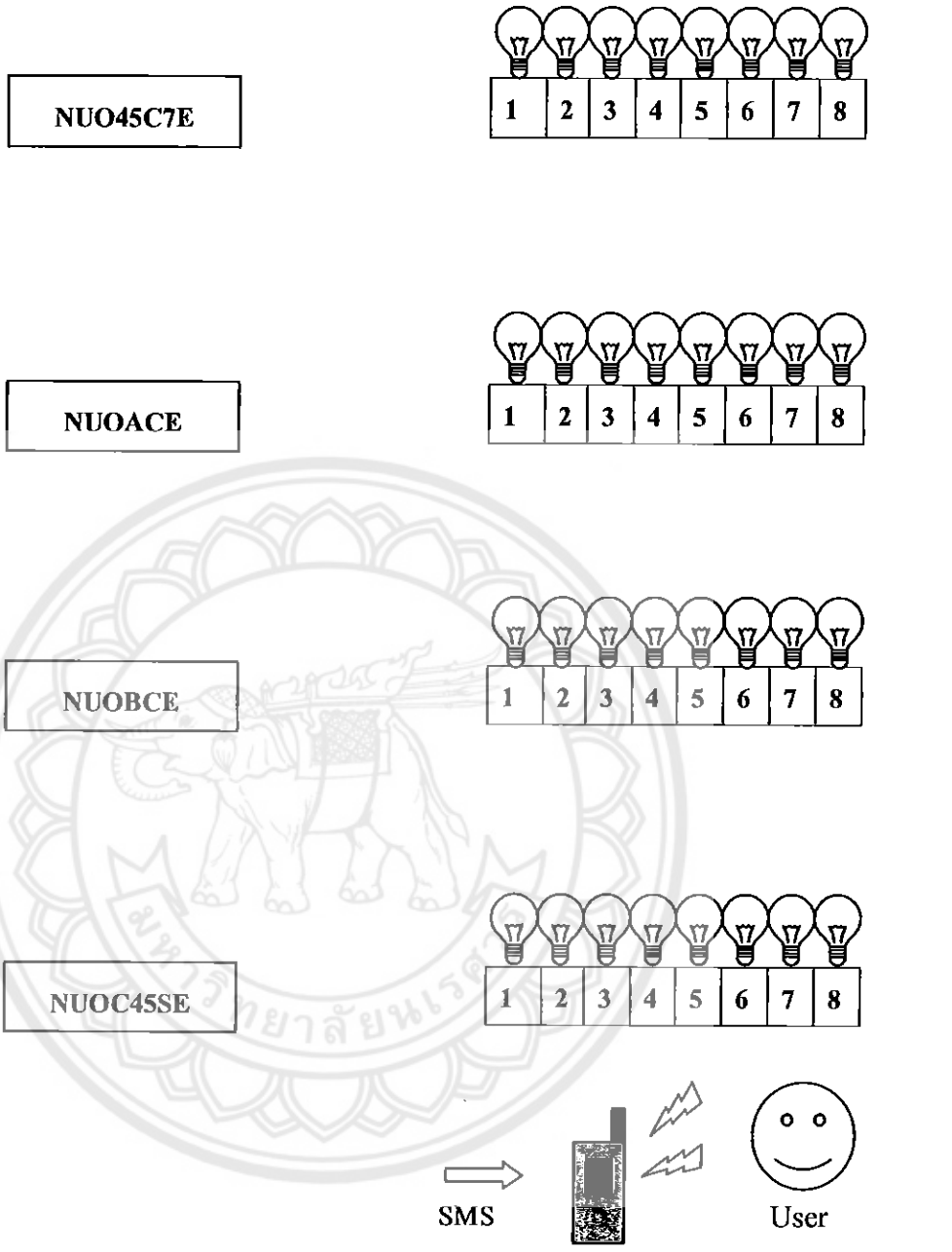
C หมายถึง Close เป็นรูปแบบในการสั่งให้ปิดไฟ

S หมายถึง Sent เป็นการแจ้งผลการทำงานหลังการสั่งการ

E หมายถึง End เป็นการสิ้นสุดข้อความ

นอกจากนี้ได้กำหนดรูปแบบข้อความที่สั่งให้ LED P1.0 – P1.7 ติดสว่างด้วยข้อความ **NUOBCE** และสั่งให้ LED P1.0 – P1.7 ดับด้วยข้อความ **NUOACE** และเมื่อแทรกอักษร "S" ลงในข้อความ STARTC-51 จะต้องสั่งงาน GR-64 ส่งข้อความกลับไปแจ้งผลการทำงานยังผู้ใช้ด้วย ซึ่งจะสามารถควบคุมการติดดับของ LED P1.0 – P1.7 บน START-C51 ดังนี้

ข้อความที่ส่ง	การแสดงผลของ LED P1.0 – P1.7
NUO12345CE	
NUO678C12345E	



และมีการส่งข้อความแจ้งผลการทำงานกลับไปยังผู้ใช้ด้วย

แทนการดับของ LED แทนการสว่างของ LED

รูปที่ 3.1 ตัวอย่างการทำงานของโปรแกรม

ขั้นตอนการทำงานของโปรแกรม

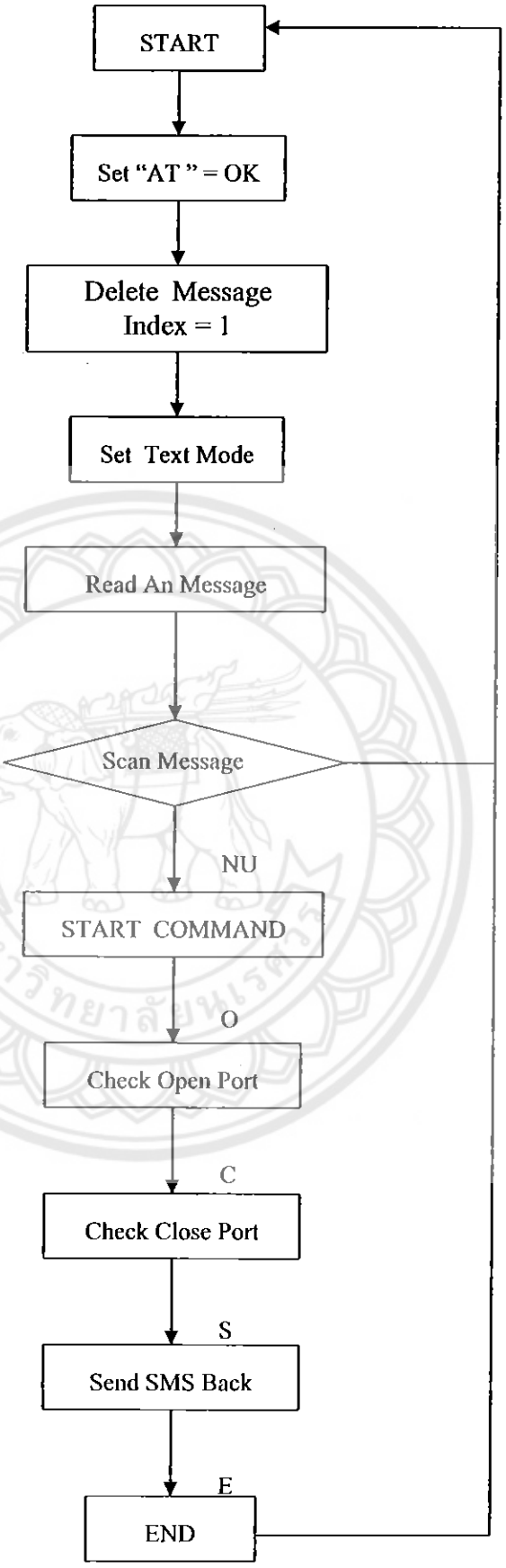
เมื่อทดลองส่ง SMS จากมือถือโดยทำการส่ง ข้อความ NUO12345C23SE แล้วตัว GR64 จะได้รับ SMS จากศูนย์ข้อความนำมาเก็บไว้ จากนั้นเมื่อ START-C51 จะคอยส่งชุดคำสั่ง AT Command มาทำการตรวจสอบว่ามีข้อความใหม่เข้าหรือไม่ โดย START-C51 จะส่ง Command มาสอบถามตามลำดับต่อไปนี้ เริ่มจากคำสั่ง AT เป็นการบอกให้ GR64 ว่าเริ่มการใช้งานคำสั่ง หลังจากนั้นแล้ว START-C51 จะทำการ SET MODE ในการรับส่งข้อมูลให้เป็น TEXT MODE โดยการส่งคำสั่ง AT+CMGL="1"ต่อจากนั้นจะส่งคำสั่ง AT+CMGL="ALL" เพื่อเป็นการอ่านข้อความทั้งหมดมาเมื่อ GR64 ได้รับคำสั่งแล้วจะ ส่งข้อมูลที่เป็น TEXTให้ START-C51 โดยประกอบไปด้วยข้อมูลต่างๆ ดังนี้

AT+CMGL="ALL"

+CMGL: 1"REC READ","+66892676962",,

NUO12345C23SE

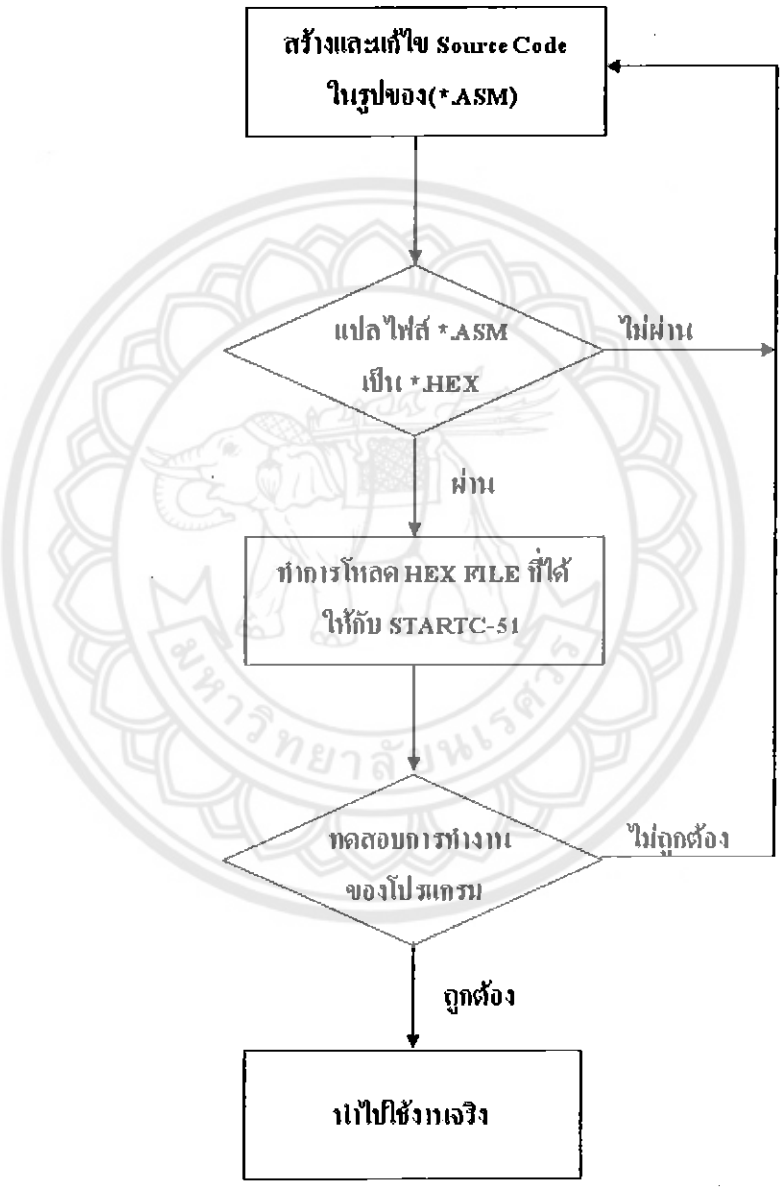
ตัว START-C51 จะทำการประมวลผลโดยการเลือกประมวลผลเฉพาะคำ NUO เป็นต้นไป ซึ่งหลักการประมวลผลเป็นไปดังนี้ เมื่อ START-C51 ได้รับข้อมูลมาแล้วประมวลผลเจอ NUO จะเป็นการเริ่มทำงานในโหมด Open โดยจะทำการประมวลตัวเลขที่ส่งมาตัวแรกเป็นเลข 1 ก็จะทำการ Setport P1.0 ให้มี LOGIC = 0 ผลคือทำให้ LED P1.0 ติด ต่อจากนั้นจะทำการประมวลตัวถัดไปคือ 2 ก็จะทำการ Setport P1.1 ให้มี LOGIC = 0 ผลคือทำให้ได้ LED P1.1 ติด และจะทำไปเลยๆจนประมวลผลพบค่าC คือเป็นการทำงานในโหมด Close เมื่อประมวลผลพบ 2 ก็จะทำการ Setport P1.1 ให้มี LOGIC = 1 ผลคือทำให้ LED P1.1 ดับ ต่อจากนั้นก็จะประมวลผลพบค่า 3 จะทำการ Setport P1.2 ให้มี LOGIC = 1 ผลคือทำให้ LED P1.2 ดับ ในลำดับถัดไปประมวลผลพบ S ซึ่งเป็นการทำงานในโหมด การส่ง SMS ไปยังหมายเลขที่ตั้งไว้เพื่อเป็นการยืนยันการทำงาน หลังจากนั้นจะเป็นการจบการทำงานในโหมด END หลังจากจบการทำงานในโหมด END แล้ว START-C51 จะส่งคำสั่ง AT+CMGD=1 เพื่อทำการลบ SMS ในลำดับที่ 1 หรือ SMS ข้อความที่เข้ามาใหม่เพื่อเป็นการ CLEAR ข้อมูลเก่าทิ้งไปเพื่อรอรับ SMS ใหม่เข้ามาแทนและเมื่อเสร็จแล้ว ก็จะเริ่มกลับไปทำงานเริ่มต้นอย่างนี้ไปเรื่อยๆ สรุปผลการทำงานคือ จะมี LED P1.0,P1.3,P1.4 ติดสว่าง



รูปที่ 3.2 แผนผังการทำงานโปรแกรม

ความสามารถของโปรแกรมคือไม่ต้องเรียงลำดับตัวเลขจากน้อยไปมาก สามารถสั่งให้ส่งข้อความกลับไปยังผู้ใช้หรือไม่ก็ได้ โดยจะไม่มีผลต่อการทำงานในส่วนของการควบคุมการติดดับของ LED P1.0 – P1.7 ทั้งนี้ในส่วนของรายละเอียดโปรแกรมได้แสดงไว้ที่ภาคผนวก

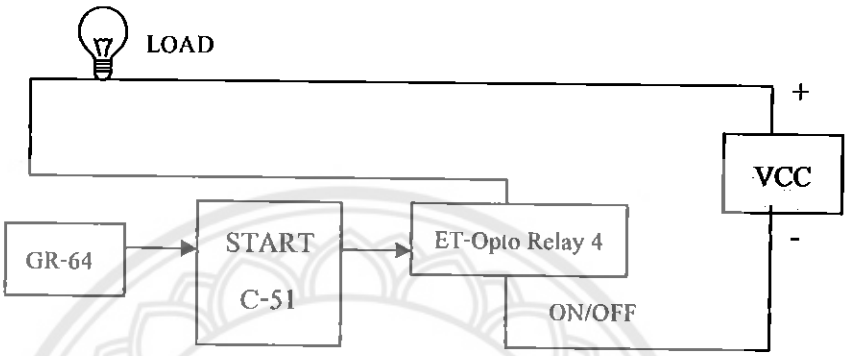
ซึ่งการพัฒนาโปรแกรมสำหรับใช้ในการควบคุมอุปกรณ์ไฟฟ้านั้น จะมีลำดับขั้นตอนในการพัฒนาโปรแกรมดังแผนผังต่อไปนี้



รูปที่ 3.3 แผนผังแสดงขั้นตอนในการพัฒนาโปรแกรมสำหรับควบคุมอุปกรณ์ไฟฟ้า

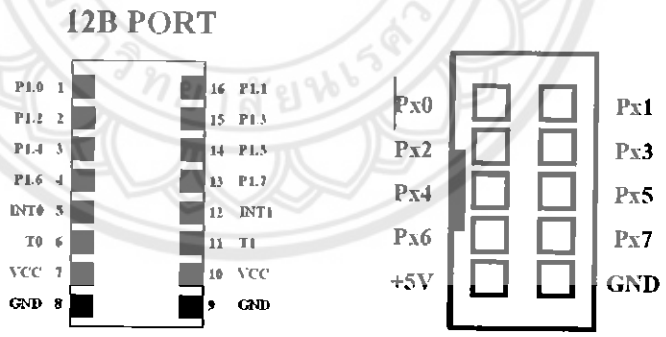
3.3.2 การนำวงจร ET – Opto Relay มาใช้งาน

เมื่อออกแบบโปรแกรมควบคุมการทำงานได้เสร็จสมบูรณ์แล้ว ในขั้นต่อไปคือการการนำวงจร ET – Opto Relay 4 มาเชื่อมต่อกับ STARTC-51 เพื่อทำหน้าที่ควบคุมการทำงานของอุปกรณ์ไฟฟ้า ในลักษณะของหน้าสัมผัส ON/OFF ดังรูป โดยจะให้หลอดไฟแทนอุปกรณ์ไฟฟ้าที่ต้องการควบคุม



รูปที่ 3.4 แสดงการควบคุมอุปกรณ์ไฟฟ้าของ STARTC-51 ผ่าน วงจร ET – Opto Relay 4

ซึ่งการเชื่อมต่อ สามารถทำได้โดยการใช้ Output จาก 12B PORT ของ STARTC-51 มาเชื่อมต่อกับ Input ของวงจร ET – Opto Relay 4 ผ่านขั้ว IDE-10 Pin ดังนี้



รูปที่ 3.5 แสดงภาพ12B PORT ของ STARTC-51 และ ขั้ว IDE-10 Pin ของวงจร ET – Opto Relay 4

ในวงจร ET – Opto Relay 4 นั้นสามารถควบคุมการทำงานของหลอดไฟได้ 4 หลอด ในโครงงานนี้จะใช้หลอดไฟจำนวน 8 หลอด ในการจำลองการควบคุมการทำงานของอุปกรณ์ไฟฟ้า

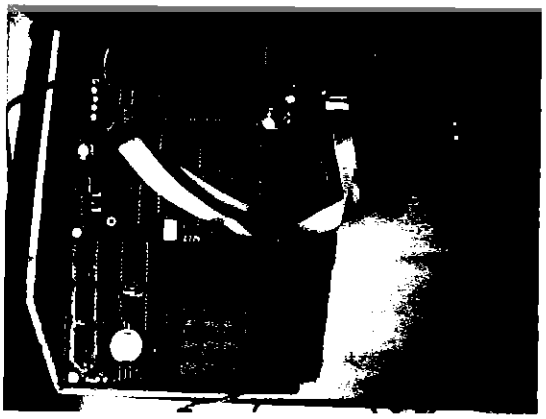
ดังนั้นจึงจำเป็นจะต้องใช้วงจร ET – Opto Relay 4 จำนวนสองวงจร โดยจะมีการจับคู่พอร์ท Px0, Px1, Px2, Px3,+5V และ GND ของขั้ว IDE-10 Pin บนวงจร ET – Opto Relay 4 ตัวที่ 1 กับพอร์ท P1.0, P1.1, P1.2, P1.3,VCC และ GND ของ 12B PORT บน STARTC-51 เพื่อควบคุมการ

ทำงานของหลอดไฟดวงที่ 1,2,3 และ 4 จับคู่พอร์ท Px0, Px1, Px2, Px3,+5V และ GND ของขั้ว IDE-10 Pin บนวงจร ET – Opto Relay 4 ตัวที่ 2 กับพอร์ท P1.4, P1.5, P1.6, P1.7,VCC และ GND ของ 12B PORT บน STARTC-51 เพื่อควบคุมการทำงานของหลอดไฟดวงที่ 5,6,7 และ 8

เมื่อจับคู่ขั้วพอร์ทและเชื่อมต่อเสร็จแล้ว จะเป็นดังรูป

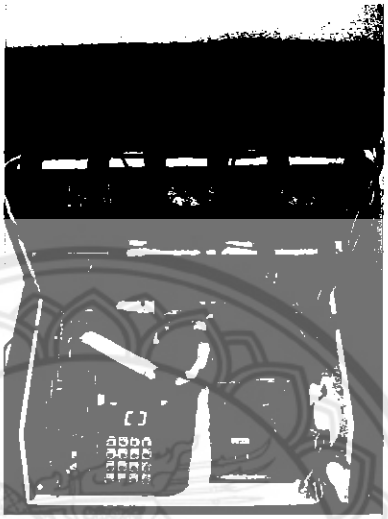
รูปที่ 3.6 แสดงการเชื่อมต่อของ 12B PORT บน STARTC-51 และ ขั้ว IDE-10 Pin บนวงจร ET – Opto Relay 4

สามารถทดสอบการทำงานของวงจร ET – Opto Relay 4 ในเบื้องต้นว่าทำงานได้ถูกต้องหรือไม่ โดยอาศัยระบบ SELF-TEST ของ STARTC-51 โดยการกดปุ่ม RESET และ INT1 พร้อมกันค้างไว้ จากนั้นค่อยๆ ปล่อยปุ่ม RESET ที่กดไว้ และปุ่ม INT1 ตามลำดับ จากนั้นจึงกดเลือกปุ่มหมายเลข 3 เพื่อทดสอบ 12B PORT จากนั้นสังเกตการทำงานของ STARTC-51 และ วงจร ET – Opto Relay 4 ว่าทำงานสัมพันธ์กันหรือไม่ โดยสังเกตจากไฟแสดงสถานะ การทำงานของ LED P1.0-P1.7 และไฟแสดงสถานะ การทำงานของ วงจร ET – Opto Relay 4



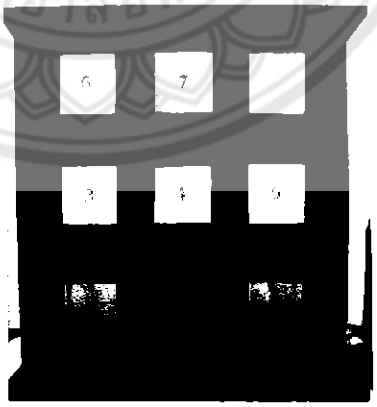
รูปที่ 3.7 แสดงการตรวจสอบการทำงานของ STARTC-51 และ วงจร ET – Opto Relay 4

จากรูปจะเห็นได้ว่าเมื่อไฟแสดงสถานะ การทำงานของ LED P1.2 บน STARTC-51 ทำงานแล้ว ไฟแสดงสถานะการทำงานของ Opto Relay ก็จะทำงานด้วย นั่นคือ STARTC-51 และ วงจร ET – Opto Relay 4 ทำงานสัมพันธ์กัน แล้วขั้นต่อไปคือการประกอบอุปกรณ์ทั้งหมดเข้าด้วยกัน และสร้างแบบจำลอง และทดสอบการทำงานต่อไป



รูปที่ 3.8 การประกอบอุปกรณ์ทั้งหมดเข้าด้วยกัน และสร้างแบบจำลอง

สำหรับแบบจำลองที่สร้างขึ้นนั้น ได้จำลองเป็นรูปแบบบ้านเพื่อแสดงภาพพจน์ที่ชัดเจน และต้องกำหนดเป็นเลขหมายประจำหลอดไฟแต่ละหลอดดังรูป



รูปที่ 3.9 แบบจำลองการทำงานของอุปกรณ์ไฟฟ้า

3.4 การทดสอบและประเมินผลการดำเนินโครงงาน

หลังจากดำเนินการพัฒนาโครงงานสำเร็จเป็นรูปร่างแล้ว ในขั้นต่อไปคือการทดสอบและประเมินผลการดำเนินโครงงาน ซึ่งจะอธิบายในบทต่อ ๆ ไป

บทที่ 4
การทดสอบและการวิเคราะห์การทำงาน

หลังจากประกอบอุปกรณ์ทั้งหมดเข้าด้วยกัน และสร้างแบบจำลองเรียบร้อยแล้วสามารถที่จะทดสอบการทำงานของระบบได้ โดยการกำหนดรูปแบบของข้อความที่ต้องการสั่งงานควบคุมอุปกรณ์ไฟฟ้าซึ่งก็คือหลอดไฟในแบบจำลอง จากนั้นจึงส่งข้อความไปยังหมายเลขโทรศัพท์ ที่อยู่บน GR-64 แบบต่อเนื่อง แล้วสังเกตผลการทำงาน

สำหรับระบบการทำงานสามารถแบ่งออกเป็น 4 ส่วนคือ

4.1 การควบคุมให้หลอดไฟทำงาน

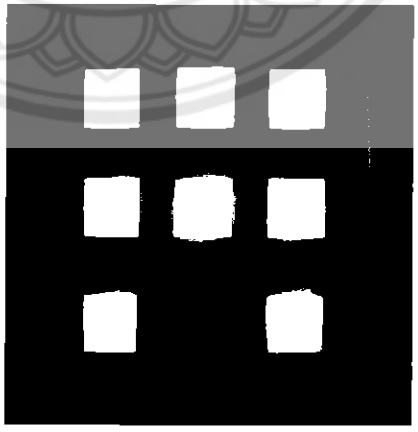
4.2 การควบคุมให้หลอดไฟหยุดทำงาน

4.3 การควบคุมให้หลอดไฟที่ทำงานอยู่หยุดทำงาน และควบคุมให้หลอดไฟที่ไม่ทำงานเริ่มทำงาน ในการส่งข้อความเพียงครั้งเดียว

4.4 การควบคุมให้ระบบส่งข้อความแจ้งผลการทำงานกลับมายังผู้ใช้

4.1 การควบคุมให้หลอดไฟทำงาน

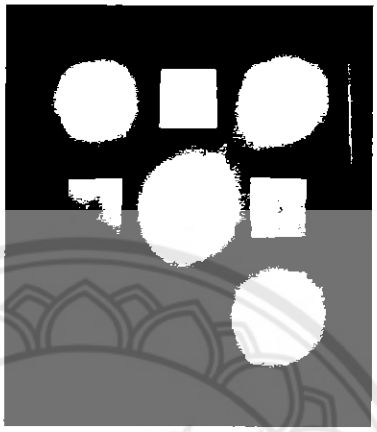
ทดสอบได้ด้วยคำสั่ง NUOBCE หมายถึง สั่งงานให้หลอดไฟทั้งหมดทำงาน เมื่อส่งข้อความไปยังหมายเลขโทรศัพท์บน GR-64 แล้ว STARTC-51จะทำการประมวลผลสักครู่หนึ่ง แล้วสั่งให้ ET – Opto Relay 4 ทำงาน จะเห็นว่าหลอดไฟทุกดวงเริ่มทำงาน



รูปที่ 4.1 แสดงผลการทำงานของคำสั่ง **NUOBCE**

4.2 การควบคุมให้หลอดไฟหยุดทำงาน

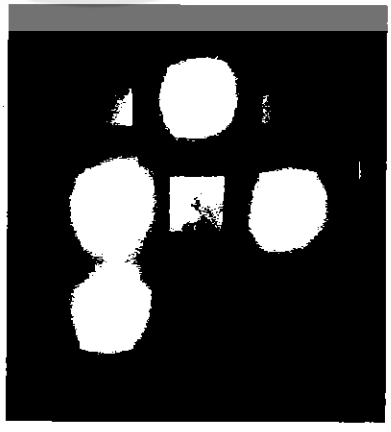
หลังจากหลอดไฟทุกดวงเริ่มทำงานแล้ว ขั้นต่อไปคือทดสอบด้วยคำสั่ง **NUOC1357E** หมายถึงสั่งงานให้หลอดไฟดวงที่ 1, 3, 5 และ 7 หยุดทำงาน ผลการทดสอบที่ได้คือ หลอดไฟดวงที่ 1, 3, 5 และ 7 จะหยุดทำงาน



รูปที่ 4.2 แสดงผลการทำงานของคำสั่ง NUOC1357E

4.3 การควบคุมให้หลอดไฟที่ทำงานอยู่หยุดทำงาน และควบคุมให้หลอดไฟที่ไม่ทำงานเริ่มทำงานในการส่งข้อความเพียงครั้งเดียว

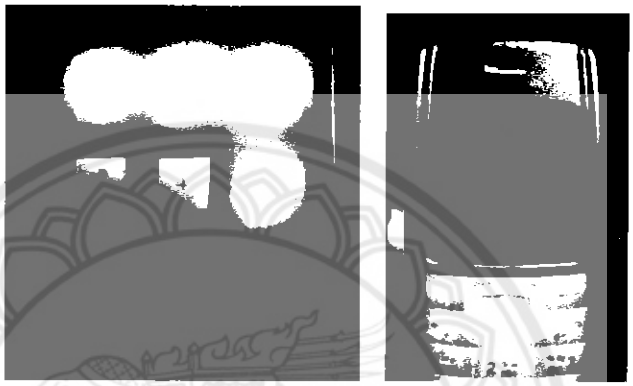
ในขั้นต่อไปจะเป็นการทดสอบโดยการสั่งให้หลอดไฟดวงที่ทำงานอยู่หยุดทำงาน และสั่งให้หลอดไฟที่หยุดทำงานไปแล้ว กลับมาทำงานอีกครั้งหนึ่ง ด้วยคำสั่ง **NUO1357C2468E** หมายถึง สั่งงานให้หลอดไฟดวงที่ 1 , 3 , 5 และ 7 ทำงาน และสั่งงานให้หลอดไฟดวงที่ 2 , 4 , 6 และ 8 หยุดทำงาน ผลการทำงานที่ได้คือ



รูปที่ 4.3 แสดงผลการทำงานของคำสั่ง **NUO1357C2468E**

4.4 การควบคุมให้ระบบส่งข้อความแจ้งผลการทำงานกลับมายังผู้ใช้

ขณะนี้หลอดไฟดวงที่ 1 , 3 , 5 และ 7 กำลังทำงานอยู่นี้ และในขั้นต่อไปจะทำการทดสอบด้วยคำสั่ง NUO86C31SE หมายถึงสั่งงานให้หลอดไฟดวงที่ 6 และ 8 ทำงาน และสั่งงานให้หลอดไฟดวงที่ 1 และ 3 หยุดทำงาน เพื่อให้เห็นว่าโปรแกรมสามารถทำงานได้โดยไม่จำเป็นต้องเรียงลำดับตัวเลขจากน้อยไปหามาก พร้อมกันนี้ให้แจ้งผลการทำงานกลับมาด้วย ผลการทำงานคือ หลอดไฟที่ทำงานจะเป็นหลอดไฟดวงที่ 5 , 6 , 7 และ 8 และมีการส่งข้อความแจ้งผลการทำงานกลับมายังผู้ใช้



รูปที่ 4.4 แสดงผลการทำงานของคำสั่ง NUO86C31SE และข้อความแจ้งผลการทำงานที่ส่งกลับมา

ในขั้นต่อไปคือทดสอบการหยุดการทำงานของหลอดไฟทั้งหมด แล้วให้ระบบส่งข้อความแจ้งผลการทำงานกลับมา เพื่อให้เห็นว่าระบบมีความเสถียรในการส่งข้อความแจ้งผลการทำงานกลับมายังผู้ใช้ ซึ่งจะส่งข้อความทดสอบด้วยคำสั่ง NUOACSE หมายถึงสั่งงานให้หลอดไฟทั้งหมดหยุดทำงาน พร้อมกันนี้ให้แจ้งผลการทำงานกลับมาด้วย ผลการทำงานที่ได้คือหลอดไฟทั้งหมดหยุดทำงาน และมีการส่งข้อความแจ้งผลการทำงานกลับมายังผู้ใช้ แสดงให้เห็นว่าระบบสามารถทำงานได้ถูกต้องตามข้อความที่กำหนดให้ และมีการตอบสนองที่ดีในระดับหนึ่ง

รูปที่ 4.5 แสดงผลการทำงานของคำสั่ง NUOACSE และข้อความแจ้งผลการทำงานที่ส่งกลับมา

บทที่ 5
บทสรุป

ในการทำโครงงานระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS คณะผู้จัดทำโครงงานได้แบ่งหัวข้อในบทนี้เป็น 4 หัวข้อด้วยกัน ดังนี้

5.1 สรุปผลการดำเนินโครงงาน

5.2 ปัญหาและแนวทางการแก้ไข

5.3 ข้อจำกัดของระบบ

5.4 ข้อเสนอแนะในการพัฒนาต่อไป

5.1 สรุปผลการดำเนินโครงงาน

จากการทำโครงงานระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS ได้ข้อสรุปจากการดำเนินงาน ดังนี้

1. ได้ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS ที่สามารถนำไปประยุกต์ใช้จริงในอนาคต
2. สามารถอำนวยความสะดวกในการควบคุมอุปกรณ์ไฟฟ้า โดยไม่จำกัดเวลา และสถานที่และระยะทาง
3. ได้รับความรู้ความเข้าใจ ในหลักการทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS มากยิ่งขึ้น
4. ได้รับความรู้ความเข้าใจ ในหลักการรับ- ส่ง SMS, การโปรแกรมภาษา Assembly และการเชื่อมต่อไมโครคอมพิวเตอร์และไมโครคอนโทรลเลอร์ มากยิ่งขึ้น
5. ได้นำความรู้พื้นฐานที่มีอยู่เดิม และอุปกรณ์ประกอบการเรียนการสอนมาใช้ให้เกิดประโยชน์มากยิ่งขึ้น

5.2 ปัญหาและแนวทางการแก้ไข

จากการทำโครงงานระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS พบปัญหาและอุปสรรคต่างๆ ดังนี้

1. ข้อความที่ส่งได้รับล่าช้า เกิดจากความล่าช้าของผู้ให้บริการเครือข่าย ในอนาคตเมื่อเครือข่ายที่ให้บริการมีการพัฒนาหรือเจริญเติบโตมากขึ้น ปัญหานี้อาจหมดไป
2. ในการเริ่มการทำงานของโปรแกรม อาจมีบางช่วงที่ STARTC-51 และ GR-64 ขาดการติดต่อกัน ทำให้ระบบไม่สามารถทำงานตามข้อความที่ส่ง เกิดจากความไม่พร้อมของ GR-64 แก้ไขได้โดย ก่อนการเริ่มโปรแกรมควรรอให้ GR-64 มีสถานะที่พร้อมใช้งานก่อน โดย GR-64 จะใช้เวลาในการเตรียมความพร้อมประมาณ 1 นาที

3. ระบบมีขนาดใหญ่เกิน และมีอุปกรณ์หลายอย่างที่ไม่ได้ถูกใช้งานบน STARTC-51 แนวทางแก้ไขคือ ควรมีการพัฒนาระบบต่อไป โดยเลือกเฉพาะไอซีที่จำเป็นต่อการใช้งาน เพื่อลดขนาดของระบบ และปรับปรุงแพคเกจให้เหมาะสมแก่การใช้งานจริง ในอนาคต
4. การพัฒนาระบบมีความล่าช้าเนื่องจากการขาดความรู้ความเข้าใจในการเลือกอุปกรณ์รับข้อความ โดยโทรศัพท์มือถือรุ่นที่มีคุณสมบัติเหมาะสมแก่การพัฒนาโครงงาน นั้นหาได้ยาก เนื่องจากในปัจจุบันการเชื่อมต่อโทรศัพท์มือถือส่วนใหญ่จะเชื่อมต่อผ่านทาง USB Port ซึ่งไม่สามารถทำงานผ่านพอร์ทอนุกรมได้ แก้ไขได้โดยเลือกอุปกรณ์ที่รองรับการทำงานผ่านพอร์ทอนุกรม และมีคุณสมบัติในการรับ - ส่งข้อความ อย่างเช่น อุปกรณ์ GR-64 ซึ่งสามารถนำไปพัฒนาต่อไปได้ในอนาคต
5. เทคนิคที่ใช้ยากที่จะทำการเข้าใจ แนวทางแก้ไขคือ ขอคำปรึกษาจากเพื่อนหรืออาจารย์ที่ปรึกษาเพื่อประหยัดเวลาในการทำงาน

5.3 ข้อจำกัดของระบบ

1. ระบบไม่สามารถเตือนได้ว่า มีอุปกรณ์ใดทำงานอยู่บ้างในกรณีที่ผู้ใช้ลืมเปิดอุปกรณ์บางอย่างทิ้งไว้
2. ระบบไม่สามารถตรวจสอบได้ว่า มีอุปกรณ์ใดทำงานอยู่ หรือหยุดทำงานไปแล้วบ้าง
3. ระบบไม่สามารถบันทึกหมายเลขโทรศัพท์มือถืออื่นๆ ในการที่จะส่งข้อความกลับไปได้ สามารถส่งข้อความกลับไปยังหมายเลขที่ตั้งไว้ในโปรแกรมได้เพียงอย่างเดียว
4. ระบบสามารถควบคุมหลอดไฟได้แค่ 8 ดวง เนื่องความสามารถของ Hardware

5.4 ข้อเสนอแนะในการพัฒนาต่อไป

1. ระบบควรจะมีการโต้ตอบกับผู้ใช้มากขึ้น อาทิเช่น การแจ้งสถานะของอุปกรณ์ที่จะทำการควบคุม การแจ้งสถานะของระบบในกรณีที่ระบบเกิดขัดข้อง หรือการตั้งเวลาปิด/เปิดอุปกรณ์ เป็นต้น
2. ระบบควรมีการตอบสนองต่อผู้ใช้แบบทันที (real-time) อย่างเช่น สามารถตรวจสอบได้ทันทีว่ามีอุปกรณ์ใดกำลังทำงานอยู่บ้าง
3. ในการออกแบบระบบควบคุมประเภทนี้ สิ่งที่สำคัญอีกประการหนึ่งคือความปลอดภัย และ ความเชื่อถือได้ ซึ่งเป็นสิ่งที่สมควรมีการพัฒนาในส่วนนี้ควบคู่ไปด้วย
4. ในอนาคตระบบการสื่อสารไร้สายน่าจะเข้ามามีบทบาทกับชีวิตประจำวันอย่างมาก ดังนั้นการพัฒนาระบบที่เกี่ยวข้องน่าจะเป็นประโยชน์อย่างมากเช่นกัน

เอกสารอ้างอิง

[1] นายวรายุทธ คัมภีราวัฒน์ และ นางสาวสุชาดา ไชยสุข. "ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS." วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น. 2547

[2] "เครื่องส่งข้อความเตือนกันขโมยรถยนต์." [ออนไลน์]. เข้าถึงได้จาก : http://www.elecnet.chandra.ac.th/research/paper/2548/2548_150393211.doc. เข้าถึงเมื่อวันที่ 28 พฤษภาคม 2550.

[3] ทวีชัย ภูรีทิพย์. ไขปัญหา RS-232. กรุงเทพมหานคร : บริษัท เอช. เอ็น. กรุ๊ป จำกัด. 2538

[4] บริษัท ศิลารีเสิร์ช จำกัด. "START-51 USER'S MANUAL." [คู่มือการใช้งาน]. กรุงเทพมหานคร :

[5] อุดม จีนประดับ. ไมโครคอนโทรลเลอร์ MCS-51. ครั้งที่ 1. กรุงเทพมหานคร : ศูนย์ผลิตตำรา สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ. 2541.

[6] บริษัท ศิลารีเสิร์ช จำกัด. "คู่มือการใช้งาน GR64 GSM/GPRS Module." [CD-ROM]. กรุงเทพมหานคร :

[7] บริษัท อี ที ที จำกัด. "คู่มือการใช้งานบอร์ด ET-OPTO RERAY 4." [คู่มือการใช้งาน]. กรุงเทพมหานคร :

[8] พันธ์ศักดิ์ พุฒิมานิตพงศ์ และ ไฉน บุญเพ็ชร. ทฤษฎีวงจรดิจิตอล. กรุงเทพมหานคร : ศูนย์ส่งเสริมอาชีวะ.

[9] ธีรวัฒน์ ประกอบผล. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์. กรุงเทพมหานคร : สำนักพิมพ์ สสท. 2543.

ภาคผนวก ก

AT Commands

ชุดคำสั่ง AT Commands

1. Call Control

Command	Description
ATA	Answer Command
ATD	Dial Command
ATH	Hang Up Call
ATL	Monitor Speaker Loudness
ATX	Call Progress Monitoring Control
ATO	Return to Online Data Mode
AT+CHUP	Hang up Call
AT+CLCC	List Current Calls
AT+CSNS	Single Numbering Scheme
AT+CSTA	Select Type of Phone Number
AT+CVHU	Voice Hang-Up
AT+VTD	DTMF tone duration
AT+VTS	DTMF and Tone Generation

2. Short Message Services - Point to Point

Command	Description
AT*E2SMSRI	Ring indicator for SMS
AT+CGSMS	Select Service for MO SMS Messages
AT+CMGF	Message Format
AT+CMGW	Write Message to Memory
AT+CMGC	Send Command
AT+CMGS	Send Message
AT+CMSS	Send From Storage
AT+CMGD	Delete Message

AT+CMGL	List Message
AT+CMGR	Read Message
AT+CNMI	New Message Indications to TE
AT+CMTI	New Message Indication Unsolicited Response
AT+CPMS	Preferred Message Storage
AT+CSCA	Service Centre Address
AT+CSCS	Select Character Set
AT+CSDH	Show Text Mode Parameters
AT+CSMP	Set Text Mode Parameters
AT+CSMS	Select Message Service
AT*E2CMGA	Modify message attribute
AT*E2CMGL	List message, without marking message Read
AT*E2CMGR	Read Message without Read mark

3. Short Message Services - Cell Broadcast

Command	Description
AT+CSCB	Select Cell Broadcast Message Type
AT*EMBOX	Mailbox Numbers
AT*EMWI	Message Waiting Indication

ภาคผนวก ข

Assembly Code

```
ORG 0000H
MOV SCON,#52H
MOV TMOD,#20H
MOV TH1,#0FAH
SETB TR1

SEND_AT:
MOV DPTR,#AT
LOOP_AT:JNB TI,LOOP_AT
CLR TI
CLR A
MOVC A,@A+DPTR
JZ END_AT
MOV SBUF,A
CLR TI
INC DPTR
LCALL DELAY
SJMP LOOP_AT
END_AT:
LCALL INIT

JUMP:SJMP JUMP1
SEND_CMGD:
;LCALL TIME3
MOV DPTR,#CMGD
LOOP_CMGD:JNB TI,LOOP_CMGD
CLR TI
CLR A
```

```
MOVC A,@A+DPTR
JZ END_CMGD
MOV SBUF,A
CLR TI
INC DPTR
;LCALL DELAY
SJMP LOOP_CMGD
END_CMGD:
LCALL INIT
JUMP1:

SEND_CMGF1:
MOV DPTR,#CMGF1
LOOP_CMGF1:JNB TI,LOOP_CMGF1
CLR TI
CLR A
MOVC A,@A+DPTR
JZ END_CMGF1
MOV SBUF,A
CLR TI
INC DPTR
LCALL DELAY
SJMP LOOP_CMGF1
END_CMGF1:
LCALL INIT

SEND_CMGL:
MOV DPTR,#CMGL
LOOP_CMGL:JNB TI,LOOP_CMGL
CLR TI
CLR A
MOVC A,@A+DPTR
```

```
JZ END3
MOV SBUF,A
CLR TI
INC DPTR
LCALL DELAY
SJMP LOOP_CMGL
END3:

;*****************************CHECK HEADER
INPUT:JNB RI,INPUT
CLR RI
MOV A,SBUF
CJNE A,#4EH,INPUT
LJMP STAT_OPEN1
NOP
LJMP INIT
JMP INPUT

STAT_OPEN1:JNB RI,STAT_OPEN1
CLR RI
MOV A,SBUF
CJNE A,#55H,STAT_OPEN1
;LJMP STAT_OPEN2

STAT_OPEN2:JNB RI,STAT_OPEN2
CLR RI
MOV A,SBUF
CJNE A,#4FH,STAT_OPEN2
LJMP STAT_CHECK
LJMP INIT
;GOTO1:LJMP STAT_CLOSE1
;*************************************END CHECK HEADER
```

```
;************************************START CHECK NUMBER PORT OPEN
STAT_CHECK:JNB RI,STAT_CHECK
CLR RI
MOV A,SBUF
CJNE A,#31H,STAT_CHECK2
LCALL OPENPOPRT1_1
CLR RI
LOOP1:JNB RI,LOOP1

STAT_CHECK2:
MOV A,SBUF
CJNE A,#32H,STAT_CHECK3
LCALL OPENPOPRT1_2
CLR RI
LOOP2:JNB RI,LOOP2

STAT_CHECK3:
MOV A,SBUF
CJNE A,#33H,STAT_CHECK4
LCALL OPENPOPRT1_3
CLR RI
LOOP3:JNB RI,LOOP3

STAT_CHECK4:
MOV A,SBUF
CJNE A,#34H,STAT_CHECK5
LCALL OPENPOPRT1_4
CLR RI
LOOP4:JNB RI,LOOP4

STAT_CHECK5:
MOV A,SBUF
```

```
CJNE A,#35H,STAT_CHECK6
CALL OPENPOPRT1_5
CLR RI
LOOP5:JNB RI,LOOP5

STAT_CHECK6:
MOV A,SBUF
CJNE A,#36H,STAT_CHECK7
CALL OPENPOPRT1_6
CLR RI
LOOP6:JNB RI,LOOP6

STAT_CHECK7:
MOV A,SBUF
CJNE A,#37H,STAT_CHECK8
CALL OPENPOPRT1_7
CLR RI
LOOP7:JNB RI,LOOP7

STAT_CHECK8:
MOV A,SBUF
CJNE A,#38H,STAT_CHECK_OA
CALL OPENPOPRT1_8
CLR RI
LOOP8:JNB RI,LOOP8

STAT_CHECK_OA:
MOV A,SBUF
CJNE A,#41H,STAT_CHECK_OB
CALL STAT_CLOSE_OA
CLR RI
LOOPOC:JNB RI,LOOPOC
```

```
STAT_CHECK_OB:
MOV A,SBUF
CJNE A,#42H,STAT_CHECK_C
CALL STAT_OPEN_OB
CLR RI
LOOPOO:JNB RI,LOOPOO
; **********************************************END CHECK NUMBER PORT OPEN

;**********************************************START CHECK HEAD CLOSE PORT
STAT_CHECK_C:
MOV A,SBUF
CJNE A,#43H,STAT_CHECK_E
CLR RI
SJMP STAT_CHECK_C1

STAT_CHECK_E:
MOV A,SBUF
CJNE A,#45H,GOTO2
CLR RI
LCALL INIT
LJMP SEND_CMGD
GOTO2:LJMP STAT_CHECK

;**************************************START CHECK  NUMBER CLOSE PORT
STAT_CHECK_C1:JNB RI,STAT_CHECK_C1
CLR RI
MOV A,SBUF
CJNE A,#31H,STAT_CHECK_C2
CALL CLOSEPOPRT1_1
CLR RI
LOOP9:JNB RI,LOOP9
```

```
STAT_CHECK_C2:
MOV A,SBUF
CJNE A,#32H,STAT_CHECK_C3
CALL CLOSEPOPRT1_2
CLR RI
LOOP10:JNB RI,LOOP10

STAT_CHECK_C3:
MOV A,SBUF
CJNE A,#33H,STAT_CHECK_C4
CALL CLOSEPOPRT1_3
CLR RI
LOOP11:JNB RI,LOOP11

STAT_CHECK_C4:
MOV A,SBUF
CJNE A,#34H,STAT_CHECK_C5
CALL CLOSEPOPRT1_4
CLR RI
LOOP12:JNB RI,LOOP12

STAT_CHECK_C5:
MOV A,SBUF
CJNE A,#35H,STAT_CHECK_C6
CALL CLOSEPOPRT1_5
CLR RI
LOOP13:JNB RI,LOOP13

STAT_CHECK_C6:
MOV A,SBUF
CJNE A,#36H,STAT_CHECK_C7
```

```
CALL CLOSEPOPRT1_6
CLR RI
LOOP14:JNB RI,LOOP14

STAT_CHECK_C7:
MOV A,SBUF
CJNE A,#37H,STAT_CHECK_C8
CALL CLOSEPOPRT1_7
CLR RI
LOOP15:JNB RI,LOOP15

STAT_CHECK_C8:
MOV A,SBUF
CJNE A,#38H,STAT_CHECK_CE
CALL CLOSEPOPRT1_8
CLR RI
LOOP16:JNB RI,LOOP16

STAT_CHECK_CE:
MOV A,SBUF
CJNE A,#45H,STAT_CHECK_CR
CLR RI
LCALL INIT
LJMP SEND_CMGD

STAT_CHECK_CR:
MOV A,SBUF
CJNE A,#53H,STAT_CHECK_C1
CLR RI
LCALL INIT
LJMP RE_MESSAG
;*******************************END CHECK CLOSE PORT
```

```
;*******************************RE MESSAG CSMP+CMGS+TEXT
RE_MESSAG:

SEND_CMGD2:
MOV DPTR,#CMGD
LOOP_CMGD2:JNB TI,LOOP_CMGD2
CLR TI
CLR A
MOVC A,@A+DPTR
JZ END_CMGD2
MOV SBUF,A
CLR TI
INC DPTR
;LCALL DELAY
SJMP LOOP_CMGD2
END_CMGD2:
LCALL INIT

RE_CSMP:
MOV DPTR,#CSMP
LOOP_CSMP:JNB TI,LOOP_CSMP
CLR TI
CLR A
MOVC A,@A+DPTR
JZ END_CSMP
MOV SBUF,A
CLR TI
INC DPTR
LCALL DELAY
SJMP LOOP_CSMP
END_CSMP:
LCALL INIT
```

```
RE_CMGS:
MOV DPTR,#CMGS
LOOP_CMGS:JNB TI,LOOP_CMGS
CLR TI
CLR A
MOVC A,@A+DPTR
JZ END_CMGS
MOV SBUF,A
CLR TI
INC DPTR
LCALL DELAY
SJMP LOOP_CMGS
END_CMGS:
LCALL INIT

RE_TEXT:
MOV DPTR,#TEXT
LOOP_TEXT:JNB TI,LOOP_TEXT
CLR TI
CLR A
MOVC A,@A+DPTR
JZ END_TEXT
MOV SBUF,A
CLR TI
INC DPTR
LCALL DELAY
SJMP LOOP_TEXT
END_TEXT:
LCALL INIT
LCALL TIME3
LCALL TIME3
```

```
LJMP SEND_AT

RE_MESSAG_END:

;*******************************

OPENPOPRT1_1:
;MOV P1,#11111110B
CLR P1.0
RET

OPENPOPRT1_2:
;MOV P1,#11111101B
CLR P1.1
RET

OPENPOPRT1_3:
;MOV P1,#11111011B
CLR P1.2
RET

OPENPOPRT1_4:
;MOV P1,#11110111B
CLR P1.3
RET

OPENPOPRT1_5:
;MOV P1,#11101111B
CLR P1.4
RET

OPENPOPRT1_6:
```

```
;MOV P1,#11011111B
CLR P1.5
RET

OPENPOPRT1_7:
;MOV P1,#10111111B
CLR P1.6
RET

OPENPOPRT1_8:
;MOV P1,#01111111B
CLR P1.7
RET

CLOSEPOPRT1_1:
;MOV P1,#00000001B
SETB P1.0
RET

CLOSEPOPRT1_2:
;MOV P1,#000000010B
SETB P1.1
RET
CLOSEPOPRT1_3:
;MOV P1,#000000100B
SETB P1.2
RET

CLOSEPOPRT1_4:
;MOV P1,#00001000B
SETB P1.3
RET
```

```
CLOSEPOPRT1_5:
;MOV P1,#00010000B
SETB P1.4
RET

CLOSEPOPRT1_6:
;MOV P1,#00100000B
SETB P1.5
RET

CLOSEPOPRT1_7:
;MOV P1,#01000000B
SETB P1.6
RET

CLOSEPOPRT1_8:
;MOV P1,#10000000B
SETB P1.7
RET

STAT_CLOSE_OA:
MOV P1,#11111111B
RET

STAT_OPEN_OB:
MOV P1,#00000000B
RET

INIT: MOV SCON,#52H
MOV TMOD,#20H
MOV TH1,#0FAH
```

```
SETB TR1
RET

TIME3:
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
RET
```

```
DELAY:    PUSH   07H
          PUSH   06H
DEL1:     MOV    R6,#0FFH
DEL2:     MOV    R7,#0FFH
          DJNZ   R7,$
          DJNZ   R6,DEL2
          POP    06H
          POP    07H
          RET
RET

AT:   DB "AT",0DH,0
CMGF1: DB "AT+CMGF=1",0DH,0
CMGL:  DB "AT+CMGL=",22H,"ALL",22H,0DH,0
CMGL2: DB "AT+CMGL=",22H,"REC READ",22H,0DH,0
CMGD:  DB "AT+CMGD=1",0DH,0   ;delete message index = 1
CSMP:  DB "AT+CSMP=17,167,0,0",0DH,0
CMGS:  DB "AT+CMGS=",22H,"0857182695",22H,0DH,0
TEXT:  DB "COMMAND COMPLETE FROM PROJECT CPE",1AH,0DH,0

END
```

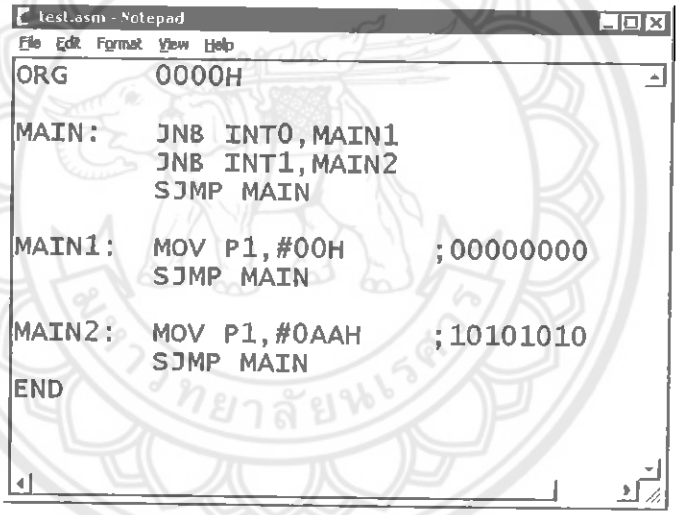
ภาคผนวก ค

การดำเนินงานของโครงงาน

1. การทดลอง Microcontroller Single Board (STARTC-51)

สำหรับการทดลอง Microcontroller Single Board (STARTC-51) จะเป็นการทดลองการโหลด Assembly Code จากเครื่องคอมพิวเตอร์ลงสู่ STARTC-51 โดยมีสามารถติดต่อสื่อสารรับ-ส่งข้อมูลผ่านทางพอร์ทอนุกรม

ในขั้นแรก ต้องทดสอบว่าสามารถใช้ START-C51 เป็นตัวควบคุมการทำงานของอุปกรณ์ไฟฟ้าได้ โดยอาศัย LED P1.0 – P1.7 เพื่อจำลองการทำงานของอุปกรณ์ไฟฟ้า โดยการโปรแกรมภาษา Assembly ควบคุมการติดดับของ LED P1.0 – P1.7 โดยใช้คำสั่งต่อไปนี้



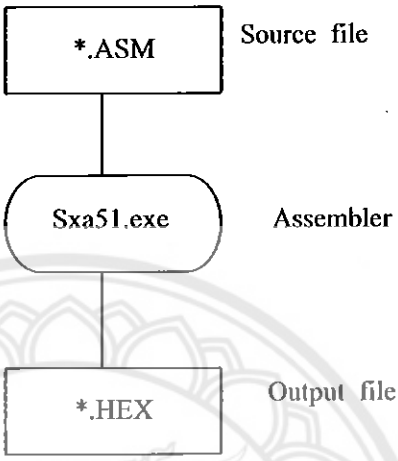
รูปที่ ค.1 แสดงชุดคำสั่งเพื่อควบคุมการติดดับของ LED P1.0 – P1.7 บน START-C51

ชุดคำสั่งดังกล่าวจะทำงานเมื่อกดปุ่ม INT0 และ INT1 โดยเมื่อกดปุ่ม INT0 LED P1.0 – P1.7 จะติดสว่างหมดทุกดวง หมายถึงการสั่งให้อุปกรณ์ไฟฟ้าทำงาน แต่เมื่อกดปุ่ม INT1 จะทำให้ LED P1.1, P1.3, P1.5, P1.7 ดับ หมายถึงการสั่งให้อุปกรณ์ไฟฟ้าหยุดทำงาน

หลักการทดลอง

เริ่มจาก เขียนโปรแกรมสำหรับควบคุมอุปกรณ์ไฟฟ้า โดยสามารถเขียนได้ใน Text Editor ตัวไหนก็ได้ที่ไม่มีอักษรพิเศษปนอยู่ เช่น Q-EDIT, SK, Edit ของ DOS, Edit plus 2 หรือ UltraEdit-32 ฯลฯ แต่ในที่นี้จะเขียนใน Notepad แล้วให้ทำการบันทึกเป็นไฟล์นามสกุล *.ASM ไว้ โดยเขียนเฉพาะคำสั่งเท่านั้น แล้วตัวโปรแกรม SXA51 จะทำการแปลภาษาเครื่องออกมา เป็นเลขฐาน16 สามารถนำมาโปรแกรมกับ Microcontroller ได้เลย

หลักการทดลองเริ่มจากการนำ Assembly Code ที่เขียนไว้ใน Notepad มาทำการ Compile ด้วย Assembler ของ MCU ซึ่งตัว Compiler นั้นจะใช้โปรแกรม Sxa51.exe ที่ให้มากับ Microcontroller Single Board (STARTC-51) เมื่อ Assembler ทำการ Compile แล้ว จากนั้นจะได้ไฟล์ใหม่เป็น op-code file ซึ่งจะอยู่ในรูปของ *.Hex



รูปที่ ค.2 แผนผังแสดงการแปลภาษา Assembly เป็น op-code file โดยโปรแกรม Sxa51.exe

การบรรจุชุดคำสั่งลงใน START-C51

1. การแปลจากภาษา Assembly เป็น op-code โดยอาศัยโปรแกรม Sxa51.exe โปรแกรม Sxa51.exe เป็น Assembler ที่ใช้ในการแปลภาษา Assembly เป็น op-code บนเครื่องคอมพิวเตอร์ โดยสามารถ compile ชุดคำสั่งที่ได้โดยการพิมพ์ในรูปแบบดังนี้

C:\ > SXA51 - <Option> File name.ASM

- <OPTION> แสดงตัวเลือกให้ดังรายการ

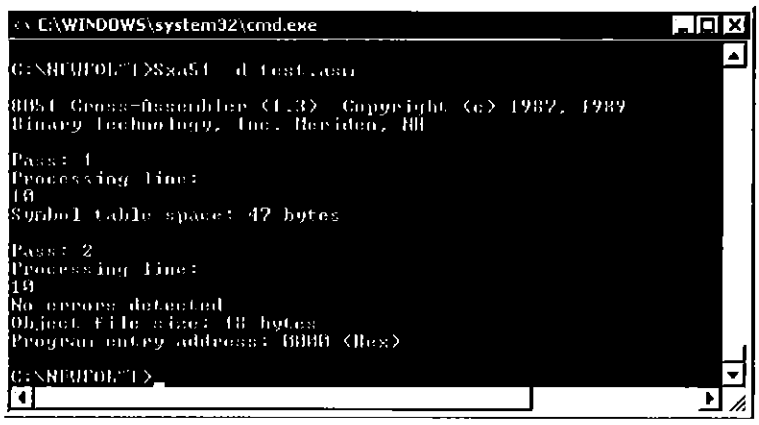
-L = ให้สร้างไฟล์แสดงรายละเอียด Address Op-code โดยจะได้ไฟล์นามสกุล .LST

-N = ไม่สร้างไฟล์เอาต์พุตใดๆ เพื่อความรวดเร็ว

-C = สร้าง Symbol cross-reference ไว้ที่ท้ายของโปรแกรมเพื่อความสะดวกของการหาตำแหน่งของ LABLE

-D = แสดง Process ขณะที่กำลังทำการ Assembler โดยโปรแกรมจะแสดงเลขบรรทัดที่กำลังแปลอยู่เช่น C:\NEWFOL~1>Sxa51 -d test.asm

ซึ่งเมื่อ compile เสร็จแล้ว โปรแกรม Sxa51.exe จะสร้าง op-code file เพื่อใช้บรรจุลงใน START-C51 ให้โดยอัตโนมัติ แสดงดังรูปที่ ค.3



รูปที่ ค.3 แสดงการใช้คำสั่ง C:\NEWFOL~1>Sxa51 -d test.asm

2. จากการที่ทำการ Compile เสร็จจะได้ผลออกมาเป็น

No errors detected ; เป็นการบอกข้อผิดพลาดที่พบ

Object file size : 851 bytes ; เป็นการบอกขนาดของข้อมูล

Program entry address : 0000 (Hex) ; เป็นการบอกถึง address เริ่มต้น

3. การบรรจุ op-code file ที่ได้จากการ compile ลงใน START-C51 โดยอาศัยโปรแกรม Hexsend.exe

โปรแกรม Hexsend.exe เป็นโปรแกรมสำหรับบรรจุ op-code file ที่ได้จากการ compile ลงใน START-C51 โดยส่งผ่านพอร์ทอนุกรม COM1 บนคอมพิวเตอร์ ไปยัง พอร์ทอนุกรม RS-232 บน START-C51 ทำได้โดยการกำหนดอัตราความเร็วของการส่งสัญญาณ (baud rate) ของพอร์ทอนุกรม COM1 ให้ตรงกับอัตราความเร็วของการส่งสัญญาณ (baud rate) ของพอร์ทอนุกรม RS-232 บน START-C51 ซึ่งในที่นี้กำหนดเป็น 9600 baud/second และจะต้องเตรียมความพร้อมของ START-C51 ดังนี้

ตรวจสอบพื้นที่ใน Program Memory ก่อนว่ามีชุดคำสั่งอื่นบรรจุหรือไม่ โดย

กดที่ปุ่ม

สังเกตหน้าจอแสดงผล

จากนั้นกดที่ปุ่ม

ตรวจดูค่าใน Program Memory จำนวน 4 Block (0,1,2,3) ว่าพื้นที่ว่าง หรือไม่ ถ้าหมายเลขใด แสดงจุดไว้แสดงว่า พื้นที่ไม่ว่าง

หากพื้นที่ไม่ว่างให้กดปุ่ม

สังเกตหน้าจอแสดงผล

START-C51 จะใช้เวลาสักครู่ในการล้าง จากนั้นก็จะกลับที่เดิม โดยจะเห็นว่าจุดที่หมายเลขหายไปแล้ว

รูปที่ ค.4 แสดงการเตรียมความพร้อมของ START-C51 ก่อนการบรรจุชุดคำสั่ง

เรียก function เพื่อการบรรจุชุดคำสั่งลง Program Memory โดย

กดที่ปุ่ม

สังเกตหน้าจอแสดงผล

จากนั้นกดที่ปุ่ม

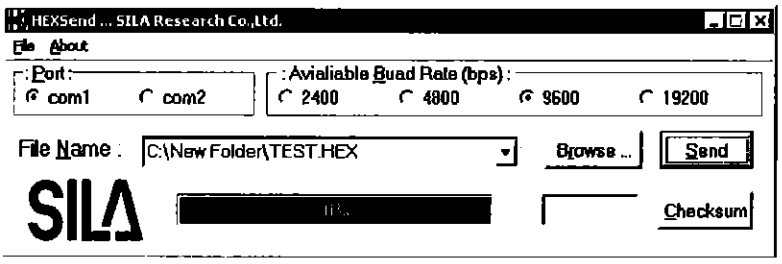
START-C51 จะให้เลือก 0 เพื่อบรรจุชุดคำสั่งลงใน Data Memory หรือ 1 เพื่อบรรจุชุดคำสั่งลงใน Program Memory

กดที่ปุ่ม เพื่อบรรจุชุดคำสั่งลงใน Program Memory

สังเกตหน้าจอแสดงผล ว่าพร้อมเพื่อการโหลดโปรแกรมทางพอร์ทอนุกรม RS-232 แล้ว

รูปที่ ค.5 แสดงการเรียก function เพื่อการบรรจุชุดคำสั่งลง Program Memory บน START-C51

หลังจากเตรียมความพร้อมของบอร์ดแล้วให้เปิดโปรแกรม Hexsend.exe แล้วกำหนดอัตราความเร็วของการส่งสัญญาณ (baud rate) เป็น 9600 ดังรูป



รูปที่ ค.6 แสดงการใช้งานโปรแกรม Hexsend.exe

จากนั้นเลือกไฟล์ op-code ที่เกิดจากการ compile แล้วกด send จะเกิดไฟกระพริบที่หลักแรกของหน้าจอแสดงผลบน START-C51 เมื่อเรียบร้อยแล้วจะเกิดส่งเสียงเตือนและ กลับเข้าสู่สภาวะปกติของหน้าจอแสดงผล เป็นอันเสร็จสิ้นการบรรจุชุดคำสั่งลงใน START-C51

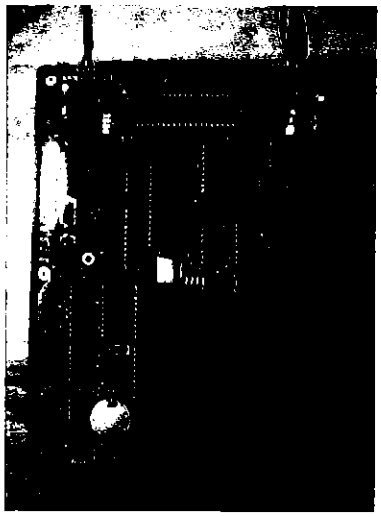
รูปที่ ค.7 หน้าจอแสดงผลบน START-C51 ระหว่าง และหลังจากการบรรจุชุดคำสั่ง

ผลจากการทำงานของชุดคำสั่ง

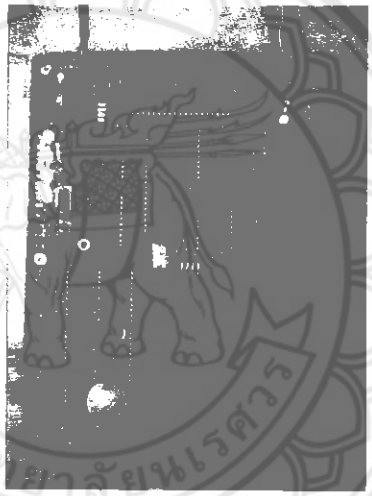
เมื่อบรรจุชุดคำสั่งลงใน START-C51 เรียบร้อยแล้ว สามารถสั่งให้ START-C51 แสดงผลการทำงานได้โดยการกดที่ปุ่ม RUN (F) แล้วป้อน Address เริ่มต้นของชุดคำสั่งที่กำหนดไว้ เพื่อให้โปรแกรมทำงาน จากนั้นกดปุ่ม ENT ดังรูปที่ 3.7

รูปที่ ค.8 การสั่งให้ START-C51 แสดงผลการทำงานของชุดคำสั่ง

จากนั้นสังเกตผลการทำงานของชุดคำสั่ง เมื่อกดปุ่ม INT0 และ INT1



รูปที่ ค.9 แสดงการทำงานของ LED P1.0 – P1.7 เมื่อกดปุ่ม INT0



รูปที่ ค.10 แสดงการทำงานของ LED P1.0 – P1.7 เมื่อกดปุ่ม INT1

จะเห็นได้ว่าเมื่อกดปุ่ม INT0 LED P1.0 – P1.7 จะติดสว่างหมดทุกดวง แต่เมื่อกดปุ่ม INT1 จะทำให้ LED P1.1, P1.3, P1.5, P1.7 ดับ นั่นคือ START-C51 สามารถควบคุมการทำงานของ LED ได้ เปรียบเสมือน START-C51 สามารถควบคุมการทำงานของอุปกรณ์ไฟฟ้าได้ โดยการใช้โปรแกรมภาษา Assembly เป็นตัวควบคุม

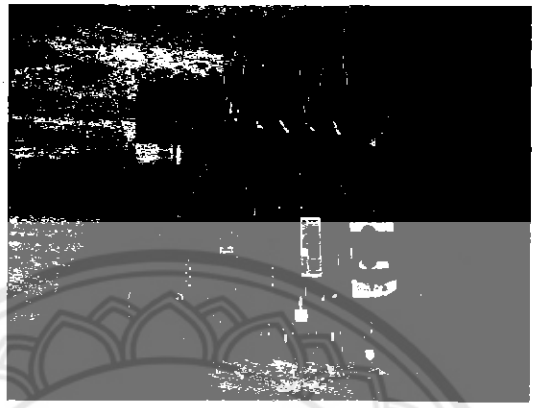
2. การทดลอง GR-64 (GR64 GSM/GPRS Module)

หลักการทดลอง

เพื่อให้แน่ใจว่าสามารถ ใช้งาน GR-64 ผ่านพอร์ทอนุกรม RS-232 บน START-C51 ได้ สามารถทดลองได้โดยการนำ GR-64 มาเชื่อมต่อกับคอมพิวเตอร์ โดยทดลองส่งชุดคำสั่ง AT -

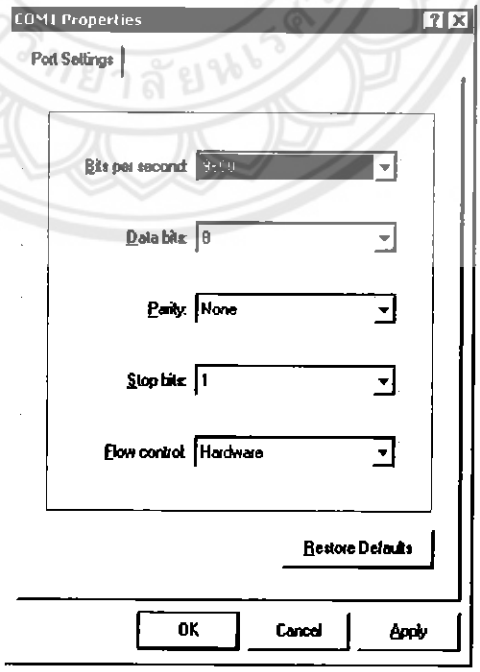
COMMAND ด้วยโปรแกรม Hyper Terminal ผ่านพอร์ทอนุกรม COM1 ของคอมพิวเตอร์ไปยังพอร์ทอนุกรม RS-232 บน GR-64 เพื่อควบคุมการทำงานและดูผลตอบสนองจาก GR-64 ดังนี้

1. การเชื่อมต่อระหว่างคอมพิวเตอร์ และ GR-64 เชื่อมต่อได้โดยผ่านพอร์ทอนุกรม COM1 ของคอมพิวเตอร์ และพอร์ทอนุกรม RS-232 บน GR-64 ดังรูป



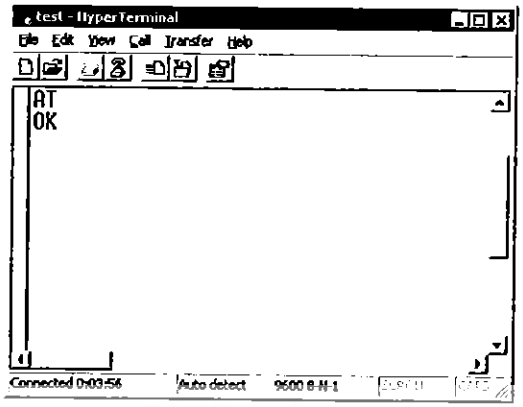
รูปที่ ค.11 แสดงพอร์ทอนุกรม RS-232 บน GR-64

เมื่อเชื่อมต่อเสร็จแล้ว เปิดโปรแกรม Hyper Terminal แล้วทำการกำหนดอัตราความเร็วของการส่งสัญญาณ (baud rate) ระหว่างคอมพิวเตอร์และโมดูล GR-64 ให้ตรงกันคือ 9600 baud/second, ข้อมูลรับ-ส่งมีขนาด 8 บิต, 1 Stop bit และไม่มีตรวจสอบ Parity บิต ดังรูปที่ 3.11



รูปที่ ค.12 แสดงการกำหนดอัตราความเร็วของการส่งสัญญาณ (baud rate)

จากนั้นทดลองพิมพ์คำสั่ง AT (enter) หากระบบสามารถเชื่อมต่อได้ โมดูล GR-64 จะตอบกลับมาว่า OK

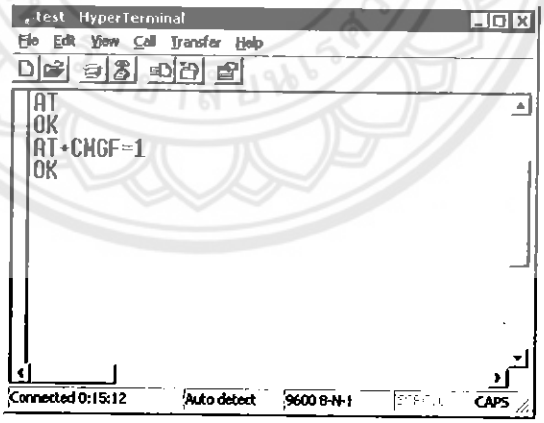


รูปที่ ค.13 แสดงผลตอบสนองจากการพิมพ์คำสั่ง AT

จะเห็นได้ว่าสามารถสื่อสารผ่านพอร์ทอนุกรม COM1 ของคอมพิวเตอร์ และพอร์ทอนุกรม RS-232 บน GR-64 ได้แล้ว

2. การทดลองส่งชุดคำสั่ง AT COMMAND โดยจะใช้ชุดคำสั่งเบื้องต้นในการเลือก Mode การทำงานของ GR-64 โดยในที่นี้จะเลือกเป็นแบบ Text Mode เพื่อให้ง่ายต่อการโปรแกรมภาษา Assembly นั่นคือคำสั่ง

AT+CMGF=1



รูปที่ ค.14 แสดงผลการทำงานของคำสั่ง AT+CMGF=1

ผลตอบสนองของคำสั่ง AT+CMGF=1ที่ตอบกลับมาคือ OK นั่นคือการเลือก Mode การทำงานของ GR-64 เป็น Text Mode ได้แล้ว ในขั้นต่อไปคือการนำ START-C51 มาเชื่อมต่อกับ GR-64 แทนการเชื่อมต่อกับคอมพิวเตอร์ แล้วทดสอบว่า สามารถใช้ START-C51 ควบคุมการทำงานของ GR-64 ได้หรือไม่

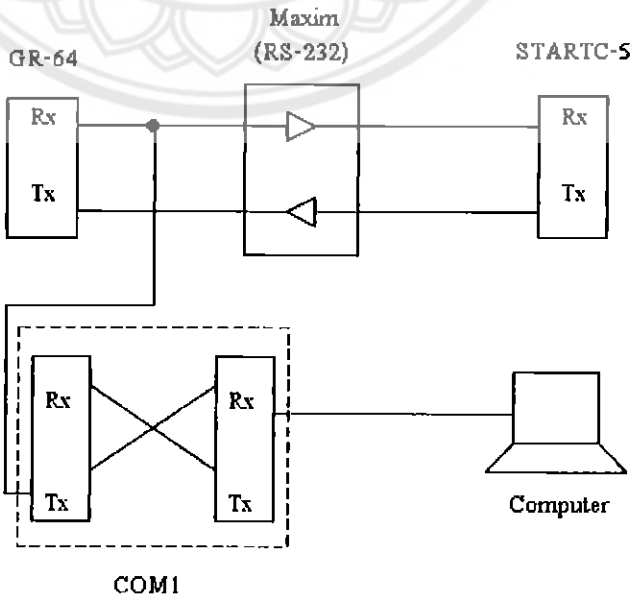
การทดลองเชื่อมต่อวงจร START-C51 และ GR-64

สามารถทำได้โดย เชื่อมต่อผ่านพอร์ทอนุกรม RS-232 บน START-C51 และ พอร์ทอนุกรม RS-232 บน GR-64 ดังรูป

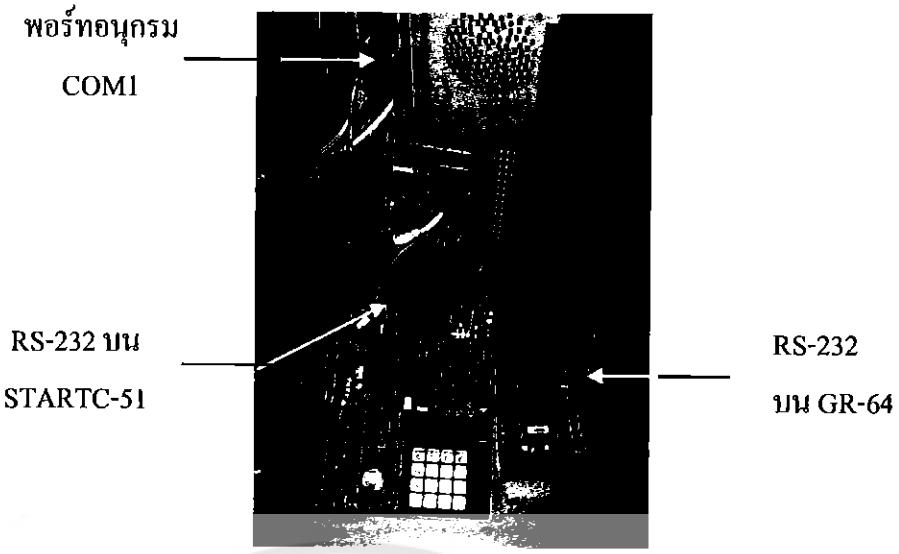
รูปที่ ค.15 การเชื่อมต่อพอร์ทอนุกรม RS-232 บน START-C51 และพอร์ทอนุกรม RS-232 บน GR-64

หลักการทดลอง

ต้องการทราบว่า START-C51 และ GR-64 สามารถสื่อสารกันผ่านพอร์ทอนุกรม RS-232 ทำได้โดยส่งชุดคำสั่ง AT COMMAND จาก START-C51 ไปยัง GR-64 เพื่อติดต่อและสั่งการ เมื่อ GR-64 ตอบกลับมา จะมีการดักสัญญาณที่ส่งกลับมาไปแสดงผลทางจอคอมพิวเตอร์ผ่านโปรแกรม Hyper- Terminal



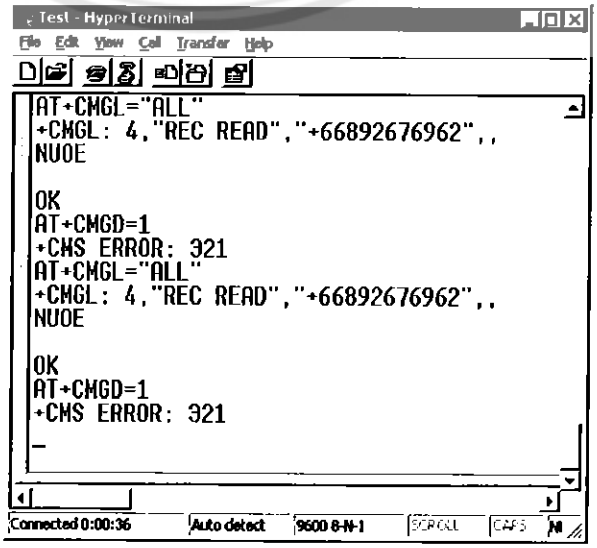
รูปที่ ค.16 Diagram การเชื่อมต่อสายสัญญาณ เพื่อแสดงข้อมูลที่ GR-64 ส่งกลับมา



รูปที่ ค.17 การเชื่อมต่อสายสัญญาณเพื่อแสดงข้อมูลที่ GR-64 ส่งกลับมา

จากนั้นทดสอบการติดต่อสื่อสารกับ GR-64 โดยการโปรแกรมให้ STARTC-51 ส่งคำสั่ง AT- COMMAND ไปยัง GR-64 ดังนี้

1) ตรวจสอบความพร้อมของ GR-64 โดยส่งคำสั่ง AT (ENTER) หาก GR-64 อยู่ในสถานะที่พร้อม จะมีการตอบ OK กลับมา
2) ส่งคำสั่งไปลบข้อความที่ Index 1 โดยส่งคำสั่ง AT+CMGD=1(ENTER)
3) ส่งคำสั่งเพื่อขออ่านข้อมูล SMS จาก GR-64 โดยส่งคำสั่ง AT+CMGL="ALL" (ENTER) ซึ่งจะได้ข้อมูลเป็นข้อความจาก SMS ทั้งหมดกลับมา
4) สั่งให้โปรแกรมวนกลับไปทำงานตั้งแต่ข้อ 1 ใหม่โดยไม่สิ้นสุด เพื่อสามารถมองเห็นการสื่อสารระหว่างอุปกรณ์ที่ชัดเจนยิ่งขึ้น โดยจะได้ผลการทำงานดังนี้



รูปที่ ค.18 แสดงผลการดักสัญญาณการสื่อสารระหว่าง GR-64 กับ STARTC-51

จากผลการทำงานดังกล่าว จะเห็นได้ว่า GR-64 กับ STARTC-51 นั้นสามารถสื่อสารกันได้แล้ว โดยโปรแกรม Hyper Terminal จะแสดงข้อความซ้ำๆ กันไปเรื่อยๆ เกิดจากการทำงานซ้ำๆของโปรแกรม ซึ่งโปรแกรมในส่วนนี้ สามารถนำไปพัฒนาเป็นโปรแกรมควบคุมการทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS ต่อไป

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวจันทร์จิรา ต๊ะมาธง

ภูมิลำเนา 183 หมู่ 7 ต.ศรีถ้อย อ.แม่ใจ จ.พะเยา
รหัสไปรษณีย์ 56130

ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนศิริมาตย์เทวี
- จบระดับมัธยมศึกษาตอนปลายจากโรงเรียนแม่ใจวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : bomberman_56@hotmail.com

ชื่อ นางสาวมรกต ศิริรัตน์

ภูมิลำเนา 55/26 หมู่ 5 ต.ท่าวังทอง อ.เมือง จ.พะเยา
รหัสไปรษณีย์ 56000

ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนเทศบาล 6 (รัฐประชาอุทิศ)
- จบระดับมัธยมศึกษาตอนต้นจากโรงเรียนรัตนโกสินทร์สมโภช
  บางเขน
- จบระดับมัธยมศึกษาตอนปลายจากโรงเรียนพะเยาพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : untouchable_dream35@hotmail.com