ระบบควบคุมความต้องการพลังไฟฟ้า

POWER DEMAND CONTROLLER SYSTEM

สุพจน์ แก้วกรณ์

SUPOD KAEWKORN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

ISBN 974-15-2144-8

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมความต้องการพลังไฟฟ้า

POWER DEMAND CONTROLLER SYSTEM

สุพจน์ แก้วกรณ์

SUPOD KAEWKORN

เลขหมู่.....
เลขทะเบียน.....76669
วัน,เดือน,ปี.....- 6 S.A. 2550

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2549

ISBN 974-15-2144-8

POWER DEMAND CONTROLLER SYSTEM

SUPOD KAEWKORN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

ISBN 974-15-2144-8

COPYRIGHT 2006

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หวขอวิทยานิพนธ์	ระบบควบคุมความต้องการพลังไพ่ฟ้า
นักศึกษา	นายสุพจน์ แก้วกรณ์
รหัสนักศึกษา	44061033
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร. วิจิตร กิณเรศ

บทคัดย่อ

วิทยานิพนธ์ฉบับนีได้นำเสนอ การสร้าง และพัฒนาระบบควบคุมความต้องการพลังไฟฟ้า สำหรับการควบคุมความต้องการพลังไฟฟ้าให้เหมาะสม ความสามารถสูง ราคาถูก ซึงมีฟังก์ชันการทำงานทีตอบสนองความต้องการของตลาด สามารถควบคุม, จัดเก็บข้อมูล, วิเคราะห์ และแสดงผล ระบบมีจุดเด่นตรงทีสามารถลด หรือขยายได้อย่างอิสระให้เหมาะสมกับงาน เนืองจากอุปกรณ์มีลักษณะเป็นโมดูลทีโปรแกรมได้ ติดต่อสือสารระยะไกลผ่านพอร์ทอนุกรม RS485, RS232C หรือโมเด็ม มีโปรแกรมตรวจสอบแก้ไขข้อผิดพลาดของการสือสารข้อมูลทำให้ควบคุมได้ถูกต้องแม่นยำ ระบบเต็มรูปแบบทีนำเสนอประกอบด้วย มิเตอร์วัดความต้องการพลังไฟฟ้า, ตัวควบคุม, ชุดรีเลย์, ตัวตรวจเช็คบันทึกสถานะ, ตัวแปลงการสือสารอนุกรม และคอมพิวเตอร์ อาจจัดระบบให้เหมาะสมกับสภาวะการใช้งานจริงได้โดยการปรับเปลียนโมดูล ตัวควบคุมจะควบคุมความต้องการพลังไฟฟ้าด้วยตัวเองใน 2 ลักษณะ คือ ควบคุมตามเวลา และ ควบคุมด้วยค่าความต้องการพลังไฟฟ้าในช่วงเวลาทีกำหนด หรือควบคุมจากคอมพิวเตอร์โดยตรงในกรณีพิเศษ การควบคุมด้วยตัวเองต้องอ่านค่าความต้องการพลังไฟฟ้าจากมิเตอร์ผ่านช่องทางการสื่อสารที่กำหนดไว้มาเปรียบเทียบกับฐานเวลาจริงภายในแล้วประมวลผลตามทีโปรแกรมไว้ จากนันก็ควบคุมรีเลย์แบบเปิด-ปิด ซึ่งรีเลย์นี้อาจนำไปตัด-ต่อโหลด หรือเครื่องกำเนิดไฟฟ้าก็ได้ตามความเหมาะสม ในขณะเดียวกันก็ทำการจัดเก็บข้อมูลทางไฟฟ้าต่างๆ ลงหน่วยความจำตามช่วงเวลาทีกำหนดด้วย ส่วนคอมพิวเตอร์นันจะทำหน้าทีควบคุมโดยตรง, เก็บบันทึกข้อมูล, วิเคราะห์, แสดงผล และโปรแกรมตัวควบคุมตามความต้องการของผู้ใช้

Thesis Title	Power Demand Controller System
Student	Mr.Supod Kaewkorn
Student ID.	44061033
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2006
Thesis Advisor	Assoc. Prof. Dr. Vijit Kinnares

Abstract

This thesis proposes the implementation and development of a power demand controller system for optimum power demand control offering low cost and many functions for commercial. These functions are control, data record, analysis and monitor. The system can either independently expand or decrement because each device is programmable module. Module can communication via RS485, RS232C or modem furthermore high accuracy by communication error correction. The system consist of demand power meters, demand controller module, relay module, input sensor module, communication adaptor and PC. User can adapt system by add or remove module for optimize. In normal time controller is work in 2 modes are time control and power demand with time control. But also it can direct control by PC. The controller reads demand power via power meters and reads present time via real time clock for processed. Then the data is processed and analyzed. The load or generator is controlled based on ON-OFF technique for relay set according to the given program. In the meantime can store data into memories. The PC is used for direct control, data record, data analysis, monitor and command input according to user requirement.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก รศ.ดร. วิจิตร กิณเรศ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์ และขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ บ.แอมพ์ทรอนอินทรูเม็นท์ (ประเทศไทย) จำกัด ที่ได้สนับสนุนทุน และอุปกรณ์ในการทำวิจัย

ขอขอบคุณ เว็บไซด์ ต่างๆทั้งไทย และ ต่างประเทศ

ขอขอบคุณ การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย สำหรับข้อมูล

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

สุพจน์ แก้วกรณ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ----------I

บทคัดย่อภาษาอังกฤษ----------II

กิตติกรรมประกาศ----------III

สารบัญ----------IV

สารบัญตาราง ----------IX

สารบัญรูป ----------X

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 1

1.3 สมมุติฐานของการศึกษา 2

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย 2

1.5 ขอบเขตการวิจัย 2

1.6 ขั้นตอนของการศึกษา 3

บทที่ 2 ทฤษฎีการวัดและควบคุมพลังไฟฟ้าสูงสุด 4

2.1 บทนำ 4

2.2 การวัดพลังไฟฟ้าสูงสุด 4

2.2.1 การวัดชนิดบวกสะสมภายใน 15 นาที 4

2.2.2 การวัดชนิดเลื่อนค่า (Moving Average) 6

2.3 การควบคุมค่าพลังไฟฟ้าสูงสุด 7

2.3.1 การหาค่าพลังไฟฟ้าสูงสุด 8

2.3.2 การพิจารณาพลังไฟฟ้าสูงสุด 8

2.3.3 การควบคุมค่าพลังไฟฟ้าสูงสุด 9

2.3.4 อุปกรณ์ควบคุมความต้องการพลังไฟฟ้า 10

2.4 การคำนวณค่าทางไฟฟ้าเพื่อเก็บบันทึก 11

2.4.1 ค่าฐานที่ต้องเก็บ 12

2.4.2 การคำนวณ 12

2.5 การคิดค่าไฟฟ้า 14

สารบัญ (ต่อ)

หน้า

2.5.1 อัตรา TOD 14

2.5.2 อัตรา TOU 14

2.5.3 ค่าเพาเวอร์แฟคเตอร์ (P.F.) 14

2.5.4 ค่า Ft. 15

บทที่ 3 ทฤษฎีพื้นฐานของโปรแกรมการติดต่อสื่อสาร 16

3.1 หลักการเขียนโปรแกรม 16

3.2 โปรแกรมไมโครคอนโทรลเลอร์ 17

3.2.1 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์ 17

3.2.2 การเขียนโปรแกรมภาษาแอสแซมบลี 18

3.3 โปรแกรมคอมพิวเตอร์ 20

3.3.1 แอพพลิเคชั่นแบบ Event-Driven 21

3.3.2 ระบบ Message Loop ของ Windows 21

3.3.3 ออบเจ็กต์ (Object) 21

3.3.4 ขั้นตอนการสร้างแอพพลิเคชั่น 23

3.3.5 ลักษณะการเขียนโปรแกรมที่ดี 23

3.3.6 การประยุกต์ใช้ Delphi 23

3.4 ระบบฐานข้อมูล (Database System) 24

3.4.1 ระบบแฟ้มข้อมูล (File System) 24

3.4.2 ความหมายของระบบฐานข้อมูล (Database System) 25

3.4.3 องค์ประกอบของระบบฐานข้อมูล 26

3.4.4 ความเป็นอิสระของข้อมูล (Data Independence) 26

3.4.5 โปรแกรมตัวกลางฐานข้อมูล (Database Management System) 27

3.4.6 หน้าที่ของ DBMS 27

3.4.7 โครงสร้าง และการจัดการไฟล์ข้อมูล Data Dictionary และ File Manager ... 28

3.4.8 ประโยชน์ของฐานข้อมูล 28

3.5 หลักการ RS232C 29

3.5.1 สัญญาณอนุกรม RS232C 30

สารบัญ (ต่อ)

หน้า

3.5.2 รีจีสเตอร์ต่าง ๆ สำหรับควบคุมพอร์ทอนุกรม 31

3.6 หลักการ RS485 32

3.6.1 การคำนวณเพื่อชดเชยค่าอิมพีแดนซ์ ในระบบ RS485 34

3.6.2 การคำนวณเพื่อหาค่าความเร็วในการรับ-ส่ง สูงสุด 35

3.7 หลักการ I^2C 36

3.8 หลักการโมเด็ม 38

3.8.1 การควบคุมโมเด็ม 38

3.8.2 หน้าที่ของขาโมเด็มชนิด RS232C 39

3.9 การกำหนดโปรโตคอล (Protocol) 40

3.10 การป้องกันไม่ให้เกิด Error (Error Prevention) 41

3.10.1 การตรวจจับ Error (Error Detection) 42

3.10.2 วิธีตรวจสอบพาริตี้ (Parity Check) 42

3.10.3 การตรวจจับ Error แบบ Longitudinal Redundancy Check (LRC) 42

3.10.4 การสร้างรหัสตรวจสอบ Cyclic Redundancy Check (CRC) 42

บทที่ 4 การออกแบบและสร้างระบบควบคุมความต้องการพลังไฟฟ้า 44

4.1 บทนำ 44

4.2 ความสามารถของระบบ 44

4.3 การทำงานของระบบควบคุมความต้องการพลังไฟฟ้า 45

4.4 การออกแบบโมดูลแปลง RS232C เป็น RS485 ชนิดแยกกราวด์
(RS232C to RS 485 Converter Isolate Module) 48

4.5 การออกแบบโมดูลตรวจเช็คบันทึกสถานะ (Input Module) 49

4.5.1 คุณสมบัติของโมดูลตัวตรวจเช็คบันทึกสถานะ 50

4.5.2 การป้องกันโมดูลตัวตรวจเช็คบันทึกสถานะ 50

4.5.3 การออกแบบการติดต่อสื่อสาร 51

4.5.4 การออกแบบซอฟแวร์ 53

4.5.5 การเช็คหน่วยความจำ NVRAM 55

4.5.6 การเช็คฐานเวลาจริง 57

สารบัญ (ต่อ)

หน้า

4.5.7 การบันทึกข้อมูลลงหน่วยความจำ 58

4.6 การออกแบบโมดูลรีเลย์ควบคุม (Relay Module) 60

4.7 การออกแบบโมดูลควบคุม และบันทึกความต้องการพลังไฟฟ้า

(Demand Controller Module) 61

4.7.1 การควบคุมในโหมดเวลา 63

4.7.2 การควบคุมในโหมดเวลาร่วมกับค่าความต้องการพลังไฟฟ้า 64

4.7.3 การตัด-ต่อรีเลย์ และคำนวณหารีเลย์ที่ต้องทำงานตัวต่อไป 69

4.7.4 ขั้นตอนการตรวจสอบการตัดและหา Relay ตัวที่ต้องตัด 69

4.7.5 ขั้นตอนการตรวจสอบการต่อและการหา Relay ตัวที่ต้องต่อ 70

4.8 การออกแบบโปรแกรมคอมพิวเตอร์ 71

4.8.1 โปรแกรมติดต่อสื่อสารกับ โมเด็ม 71

4.8.2 โปรแกรมติดต่อสื่อสารกับ โมดูลควบคุม และ

บันทึกความต้องการพลังไฟฟ้า (Demand Controller Module) 72

4.8.3 โปรแกรมติดต่อสื่อสารกับโมดูลตรวจเช็คบันทึกสถานะ (Input Module) 72

4.8.4 โปรแกรมติดต่อสื่อสารกับโมดูลรีเลย์ควบคุม (Relay Module) 72

4.8.5 โปรแกรมฐานข้อมูล (Database) 72

4.8.6 โปรแกรมแสดงผล (Monitor) 72

4.8.7 โปรแกรมคำนวณ และวิเคราะห์ข้อมูล 73

บทที่ 5 การทดลอง 74

5.1 การติดต่อสื่อสาร 75

5.1.1 ทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS232C 75

5.1.2 ทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS458 76

5.1.3 ทดลองรับ-ส่งข้อมูลผ่านโมเด็ม 77

5.1.4 ทดลองรับ-ส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์ผ่านอนุกรม I^2C 78

5.2 การอ่านค่าต่างๆทางไฟฟ้ามาจัดเก็บและแสดงผล 79

5.3 การควบคุมความต้องการพลังไฟฟ้า 82

5.3.1 การทดลองเบื้องต้นกับโหลดหลอดไฟ 82

สารบัญ (ต่อ)

หน้า

5.3.2 การทดลองกับโหลดจริง 85
5.3.3 สรุปขั้นตอนการติดตั้งระบบควบคุมค่าความต้องการพลังไฟฟ้า 91

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ 92
6.1 สรุปผลการวิจัย 92
6.2 ข้อเสนอแนะในการพัฒนาต่อไป 92

บรรณานุกรม 93

ภาคผนวก 94
ภาคผนวก ก. อัตราค่าไฟฟ้าของผู้ใช้ไฟฟ้าประเภทต่างๆ 94
ภาคผนวก ข. คำสั่ง AT COMMAND SET SUMMARY 102
ภาคผนวก ค. MCS-51 Instruction Set 106
ภาคผนวก ง. วงจรที่ใช้ในงานวิจัย 111
ภาคผนวก จ. ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่ 126

ประวัติผู้เขียน 131

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงตัวอย่างค่าดีมานต์แฟคเตอร์ สำหรับอุตสาหกรรมสาขาต่างๆ 9

3.1 ระดับสัญญาณแรงดันอนุกรมมาตรฐาน RS 232C 30

3.2 รายละเอียดของ Connector DB9 31

3.3 ตำแหน่งของรีจีสเตอร์พอร์ทอนุกรม บน IBM PC 32

4.1 การป้องกันฮาร์ดแวร์ของโมดูลตรวจเช็คบันทึกสถานะ 51

4.2 การป้องกันซอฟแวร์ของโมดูลตรวจเช็คบันทึกสถานะ 51

4.3 ฟังก์ชั่นที่ใช้ควบคุมโมดูลตรวจเช็คบันทึกสถานะ 52

5.1 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS232C จำนวน 50 ครั้ง 75

5.2 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS232C จำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง 75

5.3 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS485 จำนวน 50 ครั้ง เวลาหน่วง 0 mS. 76

5.4 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS485 จำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง
เวลาหน่วง 0 mS. 76

5.5 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS485 จำนวน 50 ครั้ง เวลาหน่วง 10 mS. 76

5.6 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS485 จำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง
เวลาหน่วง 10 mS. 77

5.7 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็มจำนวน 50 ครั้ง เวลาหน่วง 0 mS. 77

5.8 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็มจำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง
เวลาหน่วง 0 mS. 77

5.9 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็มจำนวน 50 ครั้ง เวลาหน่วง 10 mS. 78

5.10 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็มจำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง
เวลาหน่วง 10 mS. 78

5.11 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็มจำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง
เวลาหน่วง 20 mS. 78

5.12 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม I^2C จำนวน 50 ครั้ง เวลาหน่วง 20 uS. 79

5.13 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม I^2C จำนวน 50 ครั้ง เวลาหน่วง 1 mS. 79

5.14 แสดงการทดลองควบคุมความต้องการพลังงานที่ค่าความต้องการพลังไฟฟ้าเริ่มต้น 0 W... 83

5.15 ผลการทดลองควบคุมความตองการพลงงานทคาความตองการพลง เพพาเรมตน 200 W... 84

5.16 รายการสำรวจโหลดที่ต่ออยู่ในระบบทั้งหมด 87

สารบัญรูป

รูปที่ หน้า

2.1 ตัวอย่างค่าความต้องการพลังไฟฟ้าสูงสุดจากการวัดชนิดบวกสะสมภายใน 15 นาที 5

2.2 การวัดค่าความต้องการพลังไฟฟ้าสูงสุดชนิดเลื่อนค่าภายใน 15 นาที 7

3.1 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ 17

3.2 แนวคิดของ Message Loop ของ Windows 21

3.3 โครงสร้างระบบแฟ้มข้อมูล 25

3.4 โครงสร้างระบบฐานข้อมูล 25

3.5 ระดับสัญญาณเพื่อการสื่อสารข้อมูลบน IBM PC 30

3.6 ระบบสื่อสารอนุกรม RS485 33

3.7 ระดับสัญญาณไฟฟ้าระหว่างขั้ว A กับ B ของ RS485 34

3.8 การต่ออุปกรณ์บนบัสข้อมูล I^2C 37

3.9 ไดอะแกรมการส่งข้อมูลผ่านระบบโทรศัพท์ 38

3.10 (a-b) การต่อวงจรควบคุมโมเด็ม 39

3.11 รูปแบบมาตรฐานของชุดข้อมูล 40

4.1 ระบบควบคุมความต้องการพลังไฟฟ้า 47

4.2 โมดูลแปลง RS232C เป็น RS 485 ชนิดแยกกราวด์ที่ออกแบบสร้าง และพัฒนาขึ้น 48

4.3 บล็อกไดอะแกรมของโมดูลตัวตรวจเช็คบันทึกสถานะ 49

4.4 โมดูลตัวตรวจเช็คบันทึกสถานะที่ออกแบบสร้าง และพัฒนาขึ้น 49

4.5 รูปแบบการติดต่อสื่อสารกับโมดูลตัวตรวจเช็คบันทึกสถานะ 52

4.6 บล็อกคำสั่งที่มีข้อมูลไบต์เดียว 53

4.7 บล็อกคำสั่งที่มีข้อมูล 6 ไบต์ 53

4.8 บล็อกข้อมูลทวนคำสั่งกลับ 54

4.9 บล็อกข้อมูลตอบกลับที่มีความยาวข้อมูล 8 ไบต์ 55

4.10 บล็อกข้อมูลตอบกลับที่มีความยาวข้อมูล N ไบต์ 55

4.11 ขั้นตอนการเช็คสถานะหน่วยความจำ NVRAM 56

4.12 ขั้นตอนการเช็คสถานะฐานเวลาจริง 57

4.13 การบันทึกส่วนหัวของข้อมูล 58

4.14 การจัดลำดับของหน่วยความจำในขณะที่ (a) ยังไม่วนบันทึก (b) มีการวนบันทึก 59

4.15 การจัดเรียงข้อมูลเพื่อส่งออก 59

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.16 พารามิเตอร์ที่จัดเก็บในหน่วยความจำ EEPROM 60
4.17 บล็อกไดอะแกรมของชุดรีเลย์โมดูล 60
4.18 ชุดรีเลย์โมดูลทีออกแบบสร้าง และพัฒนาขึน 60
4.19 บล็อกไดอะแกรมของโมดูลตัวควบคุมความต้องการพลังไฟฟ้า 61
4.20 โมดูลตัวควบคุมความต้องการพลังไฟฟ้าทีออกแบบสร้าง และพัฒนาขึน 62
4.21 สัญญาณสือสารระหว่างไมโครคอนโทรลเลอร์ 62
4.22 การควบคุมในโหมดเวลา 63
4.23 ข้อมูลทีใช้ควบคุมในโหมดเวลา 63
4.24 โฟวล์ชาร์ทการควบคุมในโหมดเวลา 64
4.25 การควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้า 65
4.26 ค่าทีใช้ควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้า 65
4.27 ค่าพารามิเตอร์ทีใช้ควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้า 66
4.28 ค่าพารามิเตอร์ทีใช้ควบคุม (a) โหมดเวลา (b) โหมดเวลาร่วมกับพลังไฟฟ้า 67
4.29 โฟวล์ชาร์ทการควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้า 68
4.30 บล็อคไดอะแกรม ของโปรแกรมคอมพิวเตอร์ 71
5.1 แสดงการควบคุมความต้องการพลังงานไฟฟ้าทีมีการควบคุมชนิด FIFO 74
5.2 ฟอร์มขอโหลดค่าจากโมดูลภายนอก 79
5.3 ฟอร์มขอโหลดค่าจากเพาเวอร์มิเตอร์ A2000 80
5.4 ฟอร์มแสดงข้อมูลทางไฟฟ้าทีเก็บไว้ในดาต้าเบส 80
5.5 ฟอร์มแสดงกราฟข้อมูลทางไฟฟ้าทีเก็บไว้ในดาต้าเบส 80
5.6 ฟอร์มแสดงข้อมูลจากโมดูลตัวตรวจเช็คบันทึกสถานะทีเก็บไว้ในดาต้าเบส 81
5.7 ฟอร์มแสดงกราฟข้อมูลจากโมดูลตัวตรวจเช็คบันทึกสถานะทีเก็บไว้ในดาต้าเบส 81
5.8 ฟอร์มคำนวณค่าไฟฟ้าอัตรา TOU 82
5.9 ฟอร์มคำนวณค่าไฟฟ้าอัตรา TOD 82
5.10 เพาเวอร์มิเตอร์ รุ่น A2000 ทีใช้ทดสอบ 83
5.11 ทดสอบควบคุมค่าความต้องการพลังไฟฟ้า 83
5.12 One Line Diagram ของการทดลองควบคุมค่าความต้องการพลังไฟฟ้า 86

สารบัญรูป (ต่อ)

รูปที่ หน้า

5.13 (a-b) การทดลองควบคุมค่าความต้องการพลังไฟฟ้า 89
5.14 กราฟการใช้กำลังไฟฟ้า (Load Curve) ใน 1 วัน 90
5.15 กราฟความต้องการพลังไฟฟ้า (Demand) ที่ถูกควบคุมใน 1 วัน 91
5.16 กราฟการทำงานของรีเลย์เทียบกับค่าความต้องการพลังไฟฟ้า 92

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเชื้อเพลิงมีราคาสูงขึ้นเป็นผลให้ค่าไฟฟ้ามีราคาสูงขึ้นตามไปด้วย การควบคุมความต้องการพลังไฟฟ้า [2] เป็นอีกหนึ่งหนทางที่จะสามารถลดค่าใช้จ่าย อีกทั้งยังช่วยชาติประหยัดต้นทุนการผลิตไฟฟ้า เนื่องจากพฤติกรรมการใช้ไฟฟ้าในแต่ละช่วงเวลาไม่เหมือนกัน หากค่าความต้องการพลังไฟฟ้าสูงสุด ในแต่ละวันมีค่ามากกว่าค่าความต้องการพลังไฟฟ้าเฉลี่ยมาก การไฟฟ้าจะต้องสร้างโรงไฟฟ้าให้เพียงพอกับความต้องการพลังไฟฟ้าสูงสุด แต่จะทำงานเฉพาะช่วงที่มีความต้องการพลังไฟฟ้าสูงสุดเท่านั้น อีกทั้งยังใช้เชื้อเพลิงที่มีราคาสูงกว่าปกติ ทำให้เกิด ต้นทุนในการผลิตเพิ่มขึ้น ดังนั้นจึงมีการคิดค่าความต้องการพลังไฟฟ้าสูงสุดกับผู้ใช้ไฟฟ้าด้วย เพื่อชักจูงให้ผู้ใช้ไฟฟ้าปรับเปลี่ยนพฤติกรรมการใช้ไฟฟ้าให้เหมาะสม [6] บทความนี้จึงได้นำเสนอ ระบบควบคุมความต้องการพลังไฟฟ้าชาญฉลาด ที่สามารถควบคุมความต้องการพลังไฟฟ้า, บันทึกข้อมูลทางไฟฟ้า, วิเคราะห์พฤติกรรมของโหลด [5] คำนวณค่าไฟฟ้าในระบบ TOU และ TOD แสดงผลในรูปกราฟ ฯลฯ เพื่อช่วยควบคุมความต้องการพลังไฟฟ้าให้เหมาะสม ตามข้อมูลที่ได้จากการวิเคราะห์มาแล้ว [3], [4] และยังสามารถช่วยลด ค่าพลังงานไฟฟ้าได้อีกด้วย ภายในระบบประกอบด้วยโมดูลย่อยๆหลายโมดูล [1] ซึ่งแต่ละโมดูลทำงานอิสระต่อกัน และสามารถรับคำสั่งจากภายนอกได้ ทำให้ระบบมีความอ่อนตัวสูง ง่ายต่อการปรับเปลี่ยนจำนวนโมดูลตามความเหมาะสม

1.2 ความมุ่งหมาย และวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้มุ่งหวังเพื่อวิจัย และออกแบบสร้างระบบควบคุมความต้องการพลังไฟฟ้าที่สามารถนำไปใช้งานจริง ผลที่ได้จากการวิจัยสามารถใช้เป็นข้อมูลเพื่อขยายความสามารถของระบบเพิ่มขึ้น ให้สอดคล้องกับความต้องการของผู้ใช้งาน พร้อมทั้งปรับปรุงระบบให้มีความถูกต้องแม่นยำสูงขึ้น อันจะเป็นผลให้ได้ระบบควบคุมความต้องการพลังไฟฟ้าที่มีราคาถูก ลดการนำเข้า และมีประสิทธิภาพสูง อีกทั้งยังสามารถปรับการใช้งานให้เหมาะสมกับประเทศไทย

1.3 สมมุติฐานของการศึกษา

เพื่อช่วยลดค่าความต้องการพลังงานไฟฟ้าที่เกิดขึ้นพร้อมกันในช่วงเวลา 15 นาที จะต้องมีระบบตรวจจับค่าความต้องการพลังไฟฟ้าแล้วประมวลผลเพื่อปรับเปลี่ยนช่วงเวลาและปริมาณการใช้โหลดให้เหมาะสม รวมไปถึงการหาแหล่งจ่ายทดแทนในกรณีที่ไม่สามารถปรับเปลี่ยนโหลดได้ ดังนั้นจะต้องสร้างระบบที่ทำงานได้ตามฐานเวลาจริง (Real Time) และค่าความต้องการพลังไฟฟ้าจริงขณะนั้น แต่ความเป็นจริงแล้วระบบตรวจจับกับจุดที่ต้องการควบคุมอาจอยู่ห่างกันมากจึงต้องมีเรื่องของการสื่อสารข้อมูลระยะไกลเข้ามาเกี่ยวข้องด้วย การออกแบบควรจะออกแบบแยกแต่ละส่วนออกเป็นโมดูลย่อยๆแล้วใช้การติดต่อสื่อสารระยะไกลเข้ามาช่วยในการแลกเปลี่ยนข้อมูลเพื่อการควบคุม วิธีนี้จะช่วยให้ระบบมีความอ่อนตัวสูงสามารถนำไปปรับเปลี่ยนใช้ในงานควบคุมอื่นได้ อีกทั้งยังขยายหรือลดระบบได้ง่ายอีกด้วย

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

แนวความคิดที่จะต้องควยคุมค่าความต้องการพลังไฟฟ้านี้เกิดจากการไฟฟ้ามีการเรียกเก็บค่าความต้องการพลังไฟฟ้าจากผู้ใช้ไฟและมีการคิดค่าไฟฟ้าในอัตราที่ไม่เท่ากันในแต่ละช่วงเวลา โดยในช่วงเวลาที่มีความต้องการพลังไฟฟ้าสูง (On Peak) ก็จะคิดค่าไฟฟ้าแพงขึ้นกว่าในช่วงที่มีค่าความต้องการพลังไฟฟ้าต่ำ (Off Peak) (รายละเอียดดูในหัวข้อการคิดค่าไฟฟ้า) ดังนั้นการควบคุมจะต้องคำนึงถึงช่วงเวลาที่การไฟฟ้ากำหนดไว้ตามชนิดการคิดเงินที่เลือกใช้ประกอบกับการเลือกควบคุมโหลดให้เหมาะสมระบบจึงจะมีประสิทธิผลสูงสุด

1.5 ขอบเขตการวิจัย

1. สร้างระบบควบคุมความต้องการพลังไฟฟ้าที่มีความสามารถดังหัวข้อที่ 1.2 (ความสามารถของระบบ)
2. ทดลอง และทดสอบผลจากการออกแบบพร้อมสรุปวิธีการแก้ไข
3. ทดลองใช้งานกับโหลดจริง
4. ออกแบบโมดูลสำเร็จรูปเพื่อนำไปใช้งานจริง

1.6 ขั้นตอนของการศึกษา

1. ศึกษาค้นคว้าหาข้อมูลการควบคุมความต้องการพลังไฟฟ้า และการออกแบบสร้างที่มีในปัจจุบัน
2. นำข้อมูลที่ได้มาวิเคราะห์เพื่อปรับเปลี่ยนให้เหมาะสมกับประเทศไทย
3. ออกแบบคร่าว ๆ ตามความเป็นไปได้
4. เลือกอุปกรณ์ให้เหมาะสมที่มีในท้องตลาดและราคาไม่แพง
5. ปรับเปลี่ยนแบบวงจร และอุปกรณ์ตามความเหมาะสม
6. ลงมือสร้างเครื่องต้นแบบ
7. สรุปปัญหาเบื้องต้น หาวิธีแก้ไข พร้อมทั้งศึกษาหาความรู้เพิ่มเติมเกี่ยวกับการเขียนโปรแกรมให้ครอบคลุมความสามารถของระบบทั้งหมด
8. ลงมือสร้างเครื่องตัวต่อไป
9. ทดลองทดสอบพร้อมแก้ไขจุดบกพร่อง เพิ่มความสามารถของระบบให้ครบถ้วนสมบูรณ์
10. สร้างเครื่องทีจะนำไปใช้จริงให้มีความกะทัดรัดสมบูรณ์
11. ทดสอบคุณสมบัติทั้งหมด
12. แก้ไขข้อบกพร่อง
13. สรุป และวิจารณ์ผลพร้อมทั้งจัดทำวิทยานิพนธ์

บทที่ 2
ทฤษฎีการวัดและควบคุมพลังไฟฟ้าสูงสุด

2.1 บทนำ

พลังงานในรูปต่างๆที่เราใช้อยู่ในชีวิตประจำวัน ไม่ว่าจะเป็น ก๊าซ ถ่านหิน หรือ ไฟฟ้า มีวันที่จะหมดสิ้นลงได้ และการที่จะอนุรักษ์พลังงานที่จำเป็นสำหรับการดำรงชีวิตมนุษย์ให้ได้ใช้เป็นเวลายาวนานนั้นอาจกล่าวได้ว่าขึ้นอยู่กับกิจกรรมทางด้านการควบคุมการใช้พลังงานให้เป็นไปอย่างประหยัดและมีประสิทธิภาพ การควบคุมที่ถูกต้องขึ้นอยู่กับข้อมูลที่ป้อนให้กับกระบวนการควบ ถ้าข้อมูลที่ป้อนมีความถูกต้องแล้วจะส่งผลให้มีการควบคุมที่ถูกต้องแม่นยำตามไปด้วย การวัดจึงเป็นสิ่งที่ขาดไม่ได้เลยในการควบคุม ตามปกติเครื่องวัดจะติดตั้งที่จุดวัดเดียวหรือจำนวนจุดน้อยที่สุดที่สามารถครอบคลุมค่าที่ต้องการวัดทั้งหมด ฉะนั้นข้อสำคัญก็คือตำแหน่งที่ติดตั้งจะต้องเป็นตำแหน่งที่เป็นตัวแทนของสภาพที่ต้องการวัดจริงๆ แต่ต้องคำนึงถึงความสอดคล้องกันกับโหลดที่ต้องการควบคุมด้วย

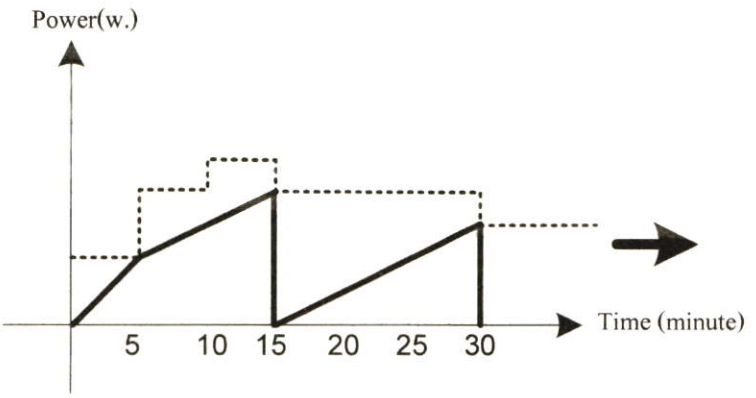
2.2 การวัดพลังไฟฟ้าสูงสุด

การวัดพลังไฟฟ้าสูงสุดมีหลายเทคนิคขึ้นอยู่กับผู้ผลิตในประเทศไทยคิดค่าสูงสุดภายในช่วงเวลา 15 นาที แต่ก็มีบางประเทศเช่นญี่ปุ่นที่คิดค่าสูงสุดภายในช่วงเวลา 30 นาที ในวิทยานิพนธ์ฉบับนี้จะกล่าวถึงวิธีพื้นฐานและเป็นที่นิยมใช้กันอย่างแพร่หลาย คือตรวจจับสัญญาณแรงดัน และกระแสทั้ง 3 เฟสเข้ามาประมวลผลหาค่าความต้องการพลังไฟฟ้าสูงสุด เทคนิคดังกล่าวนำไปสร้างมิเตอร์วัดค่าความต้องการพลังไฟฟ้าสูงสุด (Demand Power Meter) ค่าที่ได้จากการวัดจะถูกส่งให้กับหน่วยจัดเก็บ, หน่วยแสดงผล หรือหน่วยประมวลผลเพื่อควบคุม ด้วยเทคนิคการสื่อสารเฉพาะ

2.2.1 การวัดชนิดบวกสะสมภายใน 15 นาที

การวัดจะเริ่มจากเวลาที่ 0 นำค่าที่วัดได้ในแต่ละช่วงเวลาที่สุ่มค่า (Sampling Time) มาคูณกับเวลาที่ใช้สุ่มค่า แล้วนำมาบวกสะสมกับค่าเดิม ได้ค่ารวมนำมาหารด้วยเวลา 15 นาที เมื่อครบ 15 นาทีจะได้กราฟการใช้พลังงานไฟฟ้าในลักษณะไต่ขึ้นดังรูปที่ 2.1 ความชันแต่ละจุดขึ้นอยู่กับการใช้พลังงานไฟฟ้าในขณะนั้น แต่จะได้ค่าความต้องการพลังไฟฟ้าสูงสุดที่เวลา 15 นาที ถ้าต้องการควบคุมค่าความต้องการพลังไฟฟ้าจะต้องนำค่าที่บวกสะสมในช่วงเวลา 15 นาทีนี้ไปเปรียบเทียบ

กับค่าที่ต้องการควบคุมเพื่อควบคุมโหลดต่อไป ส่วนการหาค่าความต้องการพลังไฟฟ้าสูงสุดต้องนำค่าสุดท้ายที่เวลา 15 นาทีไปเปรียบเทียบค่าสูงสุดก่อนหน้านั้นว่ามีค่ามากกว่าหรือเปล่าถ้ามากกว่าก็นำค่าใหม่นี้ไปเก็บไว้เป็นค่าความต้องการพลังไฟฟ้าสูงสุดแทนค่าเดิมแต่ถ้าหากค่าใหม่มีค่าต่ำกว่าค่าสูงสุดก่อนหน้าก็ไม่ต้องเปลี่ยนแปลงค่าความต้องการพลังไฟฟ้าสูงสุดและเริ่มกระบวนการบวกสะสมในช่วงเวลาต่อไป ค่าความต้องการพลังไฟฟ้าสูงสุดนี้จะต้องถูกรีเซ็ตทุกๆ ต้นเดือนหลังจากมีการจดค่าความต้องการพลังไฟฟ้าสูงสุดไปแล้วเพื่อเริ่มคิดค่าใหม่ในเดือนถัดไป



รูปที่ 2.1 ตัวอย่างค่าความต้องการพลังไฟฟ้าสูงสุดจากการวัดชนิดบวกสะสมภายใน 15 นาที

เมื่อ

..... คือ ค่าพลังไฟฟ้าที่ใช้จริงที่เวลาต่างๆ

——— คือ ค่าพลังไฟฟ้าเฉลี่ยของช่วงเวลา 15 นาที ที่เวลาต่าง ๆ

การคิดค่าดีมานด์หรือค่าพลังไฟฟ้าสูงสุด

$$D = \frac{\sum_{t=1}^{t=15} P_t \times \Delta T}{15} \qquad (2.1)$$

โดยที่

D คือ ค่าพลังไฟฟ้าเฉลี่ยของช่วงเวลา 15 นาที

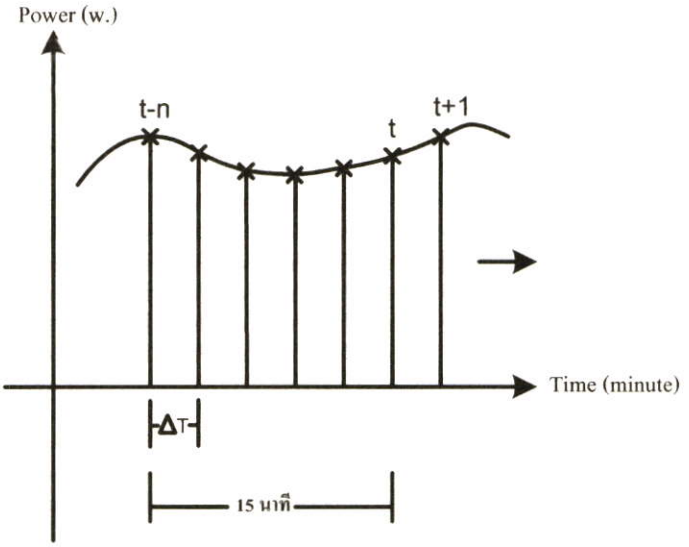
P_t คือ ค่าพลังไฟฟ้าชั่วขณะ

ΔT คือ ช่วงเวลาที่ใช้ไฟฟ้า

การวัดค่าดีมานด์หรือค่าพลังไฟฟ้าสูงสุดใช้ดีมานด์มิเตอร์เป็นเครื่องวัดดีมานด์มิเตอร์ จะวัดค่าพลังไฟฟ้าเฉลี่ยของทุกๆช่วงเวลา 15 นาทีอย่างต่อเนื่องตลอดทั้งเดือน ในหนึ่งเดือนจะวัด "ค่าพลังไฟฟ้าเฉลี่ยของช่วงเวลา 15 นาที" ได้ ประมาณ 2,880 ค่า (จำนวนวันใน 1 เดือน x จำนวนชั่วโมงใน 1 วัน x จำนวนช่วงเวลา 15 นาที ใน 1 ชั่วโมง) แล้วดีมานด์มิเตอร์ จะแสดง "ค่าพลังไฟฟ้าเฉลี่ยของช่วงเวลา 15 นาที" ที่สูงที่สุด เป็นค่าดีมานด์ หรือ ค่าพลังไฟฟ้าสูงสุด ซึ่งใช้ในการคิดค่าไฟฟ้า

2.2.2 การวัดชนิดเลื่อนค่า (Moving Average)

วิธีนี้จะบวกสะสมค่าพลังงานไฟฟ้าที่วัดได้ให้ครบ 15 นาทีก่อนในช่วงเริ่มต้น แล้วนำค่าสูงสุดไปเป็นค่าความต้องการพลังไฟฟ้าสูงสุดเหมือนการวัดชนิดบวกสะสมภายใน15นาที หลังจากนั้นเมื่อมีค่าพลังงานไฟฟ้าที่ทำการสุ่มค่ามาใหม่ให้ตัดค่าที่สุ่มได้ครั้งแรกสุดจากการคำนวณค่าความต้องการพลังไฟฟ้าสูงสุดก่อนหน้านั้นออกแล้วเอาค่าที่สุ่มได้ใหม่มาแทนแล้วคำนวณค่าความต้องการพลังไฟฟ้าสูงสุดอีกครั้งหนึ่งแล้วนำไปเปรียบเทียบกับค่าสูงสุดก่อนหน้านั้นว่ามีค่ามากกว่าหรือเปล่าถ้ามากกว่าก็นำค่าใหม่นี้ไปเก็บไว้เป็นค่าความต้องการพลังไฟฟ้าสูงสุดแทนค่าเดิมแต่ถ้าหากค่าใหม่มีค่าต่ำกว่าค่าสูงสุดก่อนหน้าก็ไม่ต้องเปลี่ยนแปลงค่าความต้องการพลังไฟฟ้าสูงสุด ในการควบคุมจะใช้ค่าความต้องการพลังไฟฟ้าสูงสุดที่คำนวณได้ใหม่นี้มาเปรียบเทียบกับค่าที่ตั้งไว้เพื่อควบคุมต่อไป การวัดชนิดเลื่อนค่านี้ดีกว่าชนิดแรกตรงที่ได้ค่าความต้องการพลังไฟฟ้าสูงสุดโดยที่ไม่ต้องรอให้ครบ 15 นาทีสามารถนำค่านี้ไปใช้ควบคุมค่าความต้องการพลังไฟฟ้าสูงสุดได้เร็วกว่าเพราะทุกค่าที่คำนวณได้เป็นค่าที่บวกสะสมเป็นเวลา 15 นาทีแล้ว



รูปที่ 2.2 การวัดค่าความต้องการพลังไฟฟ้าสูงสุดชนิดเลื่อนค่าภายใน 15 นาที

จากรูปที่ 2.2 ค่าความต้องการพลังไฟฟ้าสูงสุดหาได้จากการรวมค่าพลังงานไฟฟ้าที่ตรวจจับมาในแต่ละช่วงเวลา t+n คูณกับเวลา ΔT แล้วหารด้วยเวลา 15 นาทีเช่นเดียวกับการวัดชนิดบวกสะสม แตกต่างกันตรงที่การบวกจะไม่เริ่มใหม่เมื่อครบเวลา 15 นาที แต่จะนำค่าพลังงานไฟฟ้าที่ตรวจจับได้ใหม่ที่เวลา t+1 มาบวกเข้า พร้อมกับลบค่าพลังงานไฟฟ้าที่เวลา t-n ทิ้ง เปรียบเสมือนค่าพลังงานสะสมในช่วง 15 นาทีถูกเลื่อนไปเรื่อยๆ หากค่าความต้องการพลังไฟฟ้าสูงสุดที่วัดได้มีค่ามากกว่าค่าเก่าก็จะเก็บค่าใหม่ไว้เป็นค่าสูงสุดถ้าน้อยกว่าค่าเก่าก็จะลบทิ้ง ค่าความต้องการพลังไฟฟ้าสูงสุดที่คำนวณได้นี้จะนำไปใช้ควบคุมโหลดให้เหมาะสมต่อไป

2.3 การควบคุมค่าพลังไฟฟ้าสูงสุด

ค่ากำลังงานสูงสุดเป็นตัวกำหนดอัตราค่าความต้องการกำลังงานไฟฟ้า ซึ่งเป็นส่วนประกอบสำคัญของค่าไฟฟ้า เพราะฉะนั้นการควบคุมกำลังงานสูงสุดจึงนับว่ามีความสำคัญมาก ในโรงงานที่มีแต่การเดินเครื่องในเวลากลางวันเท่านั้น บางครั้งจะพบว่า 30-35% ของค่าไฟฟ้าทั้งหมดเป็นส่วนที่คิดจากอัตราค่าความต้องการพลังไฟฟ้าซึ่งมีราคาสูง เพราะฉะนั้นถ้าทำการควบคุมการใช้พลังไฟฟ้าให้เหมาะสม ก็จะสามารถปรับปรุงอัตราค่าไฟฟ้าให้ลดต่ำลงได้ ในโรงงานส่วนใหญ่แล้วยังมีช่องทางที่จะทำได้เหลืออยู่

2.3.1 การหาค่าพลังไฟฟ้าสูงสุด

การวัดการใช้พลังไฟฟ้าสูงสุดจาก Maximum Demand Power Meter ในมิเตอร์นี้จะชี้ค่ากำลังงานที่ใช้จนได้เวลา 15 นาที ถ้ามีการใช้พลังไฟฟ้าสูงกว่าที่ผ่านมาเข็มของมิเตอร์ก็จะดันเข็มชี้บ่งค่าสูงสุดให้สูงขึ้นไปอีกเป็นค่าใหม่ ซึ่งทำให้รู้ว่าพลังไฟฟ้าสูงสุดเท่าที่ผ่านมาเป็นเท่าใด และจะรีเซ็ตค่ากลับเป็น 0 ทุกเดือนหลังจากที่มีการจดค่าเรียบร้อยแล้ว สำหรับช่วงเวลาที่กำหนดให้ชี้ค่าพลังไฟฟ้าสูงสุดสามารถเปลี่ยนได้ เช่น ในกรณีของประเทศญี่ปุ่นก็ใช้ 30 นาที ส่วนในประเทศไทยใช้ 15 นาที

การวัดการใช้กำลังงานสูงสุดโดยใช้เครื่องวัดพลังไฟฟ้าที่บันทึกค่าได้ ทำการวัดค่าได้โดยใช้เครื่องวัดพลังไฟฟ้าที่บันทึกค่าได้ทำการบันทึกค่าการใช้พลังไฟฟ้าไว้บนกระดาษกราฟ ทุก ๆ 30 นาที หรือ ทุก ๆ 15 นาที ก็สามารถตรวจสอบค่าพลังไฟฟ้าสูงสุดว่าเกิดขึ้น ณ เวลาใด

ปัจจุบันมิเตอร์วัดพลังไฟฟ้าสามารถวัดค่าต่างๆทางไฟฟ้าได้มากมายอีกทั้งยังสามารถเก็บบันทึกค่า รวมไปถึงสื่อสารข้อมูลกับอุปกรณ์ภายนอกได้ การควบคุมจำเป็นต้องได้มาซึ่งค่าพลังไฟฟ้าสูงสุดเพื่อที่จะป้อนให้ตัวควบคุม ดังนั้นก่อนเริ่มกระบวนการควบคุมตามโปรแกรมตัวควบคุมจะต้องส่งคำสั่งไปขอค่าพลังไฟฟ้าสูงสุดจากมิเตอร์ เมื่อได้ค่าพลังไฟฟ้าสูงสุดมาแล้วก็นำไปประมวลผลควบคุมต่อไป

2.3.2 การพิจารณาพลังไฟฟ้าสูงสุด

การไฟฟ้านครหลวง จะคิดค่าความต้องการพลังไฟฟ้าตามระดับแรงดัน (ยกเว้นอุตสาหกรรมขนาดใหญ่ในระยะเวลา Off-Peak) ในอัตราต่าง ๆ กันขึ้นอยู่กับประเภทของโรงงานซึ่งแบ่งตามขนาดของความต้องการพลังไฟฟ้า

ในกรณีที่เริ่มเปิดโรงงานใหม่หรือเพิ่มอุปกรณ์จำเป็นต้องคาดคะเนอัตราความต้องการพลังไฟฟ้า หรือดีมานต์แฟคเตอร์ (Demand factor) เพื่อคำนวณค่าพลังไฟฟ้าสูงสุดที่ต้องใช้ดีมานต์แฟคเตอร์คำนวณได้จากสูตรต่อไปนี้

$$\text{ดีมานด์แฟคเตอร์} = \frac{\text{โหลดที่ใช้งานจริง}}{\text{โหลดที่ต่อทั้งระบบ}} \times 100\,(\%) \qquad (2.2)$$

ดีมานต์แฟคเตอร์เปลี่ยนแปลงไปได้มากตามสภาพของเครื่องจักรและลักษณะการใช้เครื่องเพราะฉะนั้นต้องพยายามคาดคะเนค่าดีมานต์แฟคเตอร์ให้เหมาะสมและเชื่อถือได้มากที่สุด เพื่อให้การใช้ไฟฟ้าได้ค่าที่เหมาะสมถูกต้อง

ตารางที่ 2.1 แสดงตัวอย่างค่าดีมานต์แฟคเตอร์ สำหรับอุตสาหกรรมสาขาต่าง ๆ

ชนิดของอุตสาหกรรม	ดีมานต์แฟคเตอร์	ชนิดของอุตสาหกรรม	ดีมานต์แฟคเตอร์ (%)
อาหารสำเร็จรูป	50-56	เหล็ก	40-60
ทอผ้า	55-75	อะลูมิเนียม	50-60
กระดาษ เยื่อกระดาษ	60-75	โลหะ	35-50
เคมี	60-80	ทำเครื่องจักร	35-50
กลั่นน้ำมันปิโตรเลียม	50-70	ต่อเรือ	30-45
เครื่องปั้นดินเผา	65-75		

2.3.3 การควบคุมค่าพลังไฟฟ้าสูงสุด

ในกรณีที่มีแนวโน้มว่าจะมีการใช้พลังไฟฟ้าสูงสุดเกินค่ากำหนด อาจจะทำการป้องกันได้โดยการเลือกตัดอุปกรณ์ไฟฟ้าที่สามารถหยุดได้ชั่วคราว เช่น เครื่องปรับอากาศ เป็นต้น โดยแจ้งเวลาที่จะตัดและเวลาที่เริ่มใช้งานได้ให้หน่วยงานที่เกี่ยวข้องทราบ

ในกรณีที่เป็นห้องเย็น บางโรงงานใช้วิธีตัดสวิตซ์เพื่อหยุดเครื่องทำความเย็น และเครื่องเป่าลม

การควบคุมค่าความต้องการพลังไฟฟ้าสูงสุดให้เหมาะสมพิจารณาได้จากตัวประกอบโหลด (Load factor) [7]

$$LF(\%) = \frac{E \times 100\%}{D_m \times h} \tag{2.3}$$

ที่เช่น โรงงานที่ปฏิบัติงานตลอด 24 ชม. จะมีค่า LF ที่เหมาะสมในย่าน70% ถึง 85% แต่นิยมใช้ค่า 80% เป็นค่าความต้องการพลังไฟฟ้าสูงสุดที่เหมาะสม ดังนั้นโรงงานที่ปฏิบัติงาน 8 ชม. จะมีค่า LF ที่เหมาะสมประมาณ 26.67 %

ค่าความต้องการพลังไฟฟ้าสูงสุดที่เหมาะสมหาได้จากค่า LF ที่เหมาะสม

$$D_m^* = \frac{E \times 100\%}{LF^* \times h} \tag{2.4}$$

โดยที่

LF คือ ตัวประกอบโหลด

LF* คือ ตัวประกอบโหลดที่เหมาะสม

E คือ จำนวนหน่วยพลังงานที่ใช้ในรอบ 1 เดือน (กิโลวัตต์·ชั่วโมง)

Dm คือ ความต้องการพลังไฟฟ้าสูงสุด

D_m^* คือ ความต้องการพลังไฟฟ้าที่เหมาะสม

h คือ จำนวนชั่วโมงที่ทำงานในเดือนนั้น

โดยพิจารณาวิธีการควบคุมดังนี้

1. เปลี่ยนเวลาใช้งานเครื่องใช้ไฟฟ้าเป็นเวลาอื่น ในขณะที่มีการใช้พลังงานสูงสุด
2. มีการเตือนหรือควบคุมระบบการผลิตในช่วงที่มีการใช้กำลังไฟฟ้าสูงสุด
3. ปรับเปลี่ยนเครื่องจักรหรืออุปกรณ์ไฟฟ้าให้มีขนาดการใช้พลังงานน้อยลงแต่ให้เพิ่มช่วงเวลาของการทำงาน
4. ปรับปรุงกระบวนการผลิตเพื่อควบคุมการใช้กำลังไฟฟ้าที่เกิดขึ้นพร้อมกันในเวลา 15 นาที
5. พิจารณาอุปกรณ์ที่มีประสิทธิภาพสูงในการประหยัดพลังงานแทนระบบการผลิตเดิม
6. ประหยัดพลังงานในช่วงที่มีความต้องการกำลังไฟฟ้าสูงสุด
7. ใช้ไฟฟ้าสำรองในขณะที่มีความต้องการไฟฟ้าสูง

2.3.4 อุปกรณ์ควบคุมความต้องการพลังไฟฟ้า

ในโรงงานที่ทำสัญญาการใช้ไฟฟ้าสูงกว่า 500 kW การควบคุมการใช้พลังไฟฟ้าไม่ให้สูงกว่าค่าความต้องการกำลังงานสูงสุดนับว่ามีความสำคัญมาก เพราะถ้าค่าพลังไฟฟ้าสูงสุดมีค่ามากจะส่งผลให้การไฟฟ้าฝ่ายผลิตต้องจัดสร้างโรงไฟฟ้าเพิ่มขึ้น ทำให้สูญเสียต้นทุนการผลิตดังนั้นจึงมีการจัดเก็บค่าความต้องการพลังไฟฟ้าสูงสุดกับโรงงานที่มีขนาดการใช้ไฟฟ้าสูง

การควบคุมความต้องการพลังไฟฟ้าคือ การควบคุมการใช้โหลดไม่ให้เกินค่าความต้องการพลังไฟฟ้าสูงสุด และในขณะเดียวกันก็วางแผนปรับปรุงพลังไฟฟ้าให้มีค่าที่เหมาะสม พร้อมกับเพิ่มโหลดแฟคเตอร์ของโรงงานอีกด้วย การควบคุมความต้องการพลังไฟฟ้าจะได้ผลดีก็ต่อเมื่อสามารถทราบข้อมูลของอุปกรณ์ไฟฟ้าได้อย่างละเอียดถูกต้อง เช่น ต้องรู้ว่าเป็นอุปกรณ์แบบไหน มีขนาดเท่าไหร่ สภาพการใช้งานเป็นอย่างไร (เช่น เดินเครื่องตลอดเวลา เดินบ้างหยุดบ้าง เดินเป็นระยะ ๆ หรือว่าพักการใช้งาน) แล้วจึงนำข้อมูลเหล่านี้มาใช้ในการควบคุมความต้องการกำลังงานไฟฟ้า ซึ่งจะทำให้การควบคุมทำได้ผลดี

เท่าที่ผ่านมามีการใช้ Maximum Demand Power Meter ซึ่งมีเสียงสัญญาณแจ้งการใช้พลังไฟฟ้าเกินติดอยู่ เครื่องชนิดนี้สามารถใช้ในการควบคุมตามเวลาที่ตั้งไว้ได้ แต่ไม่สามารถทำงานควบคุมอย่างต่อเนื่องตลอดเวลาได้ อุปกรณ์ที่แก้ข้อเสียส่วนนี้ก็คือ อุปกรณ์ควบคุมความต้องการพลังไฟฟ้า (Demand Controller)

Demand Controller เป็นอุปกรณ์ที่ใช้ในการตรวจตรา แจ้งเหตุ และบันทึกสภาพการใช้พลังไฟฟ้าและยังสามารถส่งสัญญาณไปควบคุมการทำงานของโหลดโดยอัตโนมัติอย่างเหมาะสม ในกรณีที่มีแนวโน้มว่าจะมีการใช้พลังไฟฟ้าเกิน อุปกรณ์ควบคุมมีหลายชนิด เช่น PLC รีเลย์ โซลิดสเตทรีเลย์ อุปกรณ์เหล่านี้มักรวมกันอยู่ในโมดูลที่สามรถรับคำสั่งควบคุมจากภายนอกได้ ส่วนประมวลผลจะทำหน้าที่อ่านค่าพลังไฟฟ้าสูงสุดมาประมวลผลแล้วส่งข้อมูลมาให้โมดูลควบคุมทำงานอีกทีหนึง การแยกส่วนระหว่างโมดูลควบคุมออกจากส่วนแสดงผลทำให้สะดวกต่อการใช้งานในกรณีที่จุดควบคุมอยู่ห่างกัน

หลักการในการทำงานของ Demand Controller สามารถอธิบายได้ดังนี้

พลังไฟฟ้าจะถูกตรวจตรา และบันทึกไว้โดย Demand Power Meter ที่การไฟฟ้าติดตั้งไว้ โดยคิดจากพลังไฟฟ้าเฉลี่ยทุก 15 นาที ในกรณีที่การใช้โหลดสม่ำเสมอจะสามารถทำการแสดงพลังไฟฟ้าในช่วง 15 นาทีได้อย่างราบเรียบ นอกจากนี้แล้วพอจะคาดคะเนล่วงหน้าเกี่ยวกับพลังไฟฟ้าสูงสุดได้หลังจากสิ้นสุดช่วงเวลาดังกล่าวแล้ว

โดยทั่วไปการใช้โหลดจะเปลี่ยนแปลงอยู่ตลอดเวลา เพราะฉะนั้นถ้าใช้ Demand Power Meter อย่างเดียวก็จะไม่สามารถทำนายได้ว่าการใช้พลังไฟฟ้าในช่วงนั้นๆ จะเป็นอย่างไร แต่จากการใช้หลักการโปรแกรมการควบคุมนี้ทำให้ Demand Controller สามารถทำการควบคุมการใช้พลังไฟฟ้า ให้ไม่เกินค่าที่ตั้งไว้ ไม่ว่าสภาพการใช้โหลดจะเป็นอย่างไรก็ตาม

2.4 การคำนวณค่าทางไฟฟ้าเพื่อเก็บบันทึก

ระบบควบคุมค่าความต้องการพลังไฟฟ้านี้สามารถใช้เป็น Data Logger ได้เพื่อเก็บค่าต่างๆ ทางไฟฟ้า เช่น กระแส, แรงดัน, พลังไฟฟ้า ฯลฯ แต่เนื่องจากโมดูลควบคุมความต้องการพลังไฟฟ้ามีหน่วยความจำที่ใช้บันทึกข้อมูลจำนวนจำกัดการเก็บค่าต่างๆทางไฟฟ้าจึงไม่ควรเก็บทุกค่าลงหน่วยความจำ แต่จะเลือกเก็บเฉพาะค่าฐานที่จำเป็นเท่านั้น ค่าฐานที่ว่าหมายถึงค่าพื้นฐานทางไฟฟ้าที่สามารถนำไปคำนวณหาค่าอื่นๆ ได้ต่อไป เพื่อลดการใช้หน่วยความจำลงเมื่อต้องการลง ถ้าต้องการใช้ข้อมูลนอกเหนือจากค่าฐานก็นำค่าที่จำเป็นไปคำนวณ การบันทึกค่าจะบันทึกค่าฐานตามด้วยวันเวลาปัจจุบันเพื่อบ่งบอกว่าค่าที่วัดได้นี้มาจากเวลาใด

2.4.1 ค่าฐานที่ต้องเก็บ

I_1 = กระแสเฟส 1

I_2 = กระแสเฟส 2

I_3 = กระแสเฟส 3

V_1 = แรงดันเฟส 1

V_2 = แรงดันเฟส 2

V_3 = แรงดันเฟส 3

PF_1 = เพาเวอร์แฟคเตอร์เฟส 1

PF_2 = เพาเวอร์แฟคเตอร์เฟส 2

PF_3 = เพาเวอร์แฟคเตอร์เฟส 3

D_t = ค่าความต้องการพลังไฟฟ้าสูงสุดชั่วขณะ

Q_t = ค่าความต้องการพลังไฟฟ้ารีแอ็คทีฟสูงสุดชั่วขณะ

2.4.2 การคำนวณ

การคำนวณนี้จะเกิดขึ้นบนคอมพิวเตอร์หลังจากที่ได้รับค่าฐานมาแล้วเพื่อลดการใช้งาน CPU ของโมดูลควบคุมความต้องการพลังไฟฟ้า และลดข้อมูลในการสื่อสาร

ค่าเพาเวอร์แฟคเตอร์ที่ได้จากการโหลดมาจากมิเตอร์สามารถนำไปคำนวณหาค่ามุมเฟสได้ดังนี้

$$\theta_1 = \cos^{-1} PF1 \tag{2.5}$$

$$\theta_2 = \cos^{-1} PF_2 \tag{2.6}$$

$$\theta_3 = \cos^{-1} PF_3 \tag{2.7}$$

เมื่อคำนวณค่ามุมเฟสแล้วก็สามารถหาค่ากำลังไฟฟ้าแต่ละเฟสได้จากการนำค่า $\cos\theta$ มาคูณกับแรงดัน และกระแสแต่ละเฟส

$$P_1 = V_1 I_1 \cos\theta_1 \tag{2.8}$$

$$P_2 = V_2 I_2 \cos\theta_2 \tag{2.9}$$

$$P_3 = V_3 I_3 \cos\theta 3 \tag{2.10}$$

กำลังไฟฟ้า 3 เฟส เท่ากับกำลังไฟฟ้าแต่ละเฟสรวมกัน

$$P_{3\phi} = V_1 I_1 + V_2 I_2 + V_3 I_3 \tag{2.11}$$

คำนวณหาค่ากำลังไฟฟ้ารีแอ็คทีฟแต่ละเฟสได้จากการนำค่า $\sin\theta$ มาคูณกับแรงดัน และกระแสแต่ละเฟส

$$Q_1 = V_1 I_1 \sin\theta_1 \tag{2.12}$$

$$Q_2 = V_2 I_2 \sin\theta_2 \tag{2.13}$$

$$Q_3 = V_3 I_3 \sin\theta_3 \tag{2.14}$$

กำไฟฟ้ารีแอ็คทีฟ 3 เฟส เท่ากับกำลังไฟฟ้ารีแอ็คทีฟแต่ละเฟสรวมกัน

$$Q_{3\varphi} = Q_1 + Q_2 + Q_3 \tag{2.15}$$

หามุมเฟสรวมได้จากสมการที 2.16 เพื่อนำไปหาเพาเวอร์แฟคเตอร์ของระบบ 3 เฟส ในสมการที่ 2.17

$$\theta_{3\phi} = \tan^{-1}\left(\frac{Q_{3\phi}}{P_{3\phi}}\right) \tag{2.16}$$

$$PF_{3\phi} = \cos\theta_{3\phi} \tag{2.17}$$

หากระแสนิวทรอน และมุม ได้จากการรวมกันของกระแส 3 เฟส ทางเฟสเซอร์

$$I_n \angle\theta_n = I_1 \angle(0 + \theta_1) + I_2 \angle(-120 + \theta_2) + I_3 \angle(120 + \theta_3) \tag{2.18}$$

$$x = I_1 \cos(0 + \theta_1) + I_2 \cos(-120 + \theta_2) + I_3 \cos(120 + \theta_3) \tag{2.19}$$

$$y = I_1 \sin(0 + \theta_1) + I_2 \sin(-120 + \theta_2) + I_3 \sin(120 + \theta_3) \quad (2.20)$$

$$\theta_n = \tan^{-1}\left(\frac{y}{x}\right) \quad (2.21)$$

2.5 การคิดค่าไฟฟ้า

การคิดค่าไฟฟ้าในปัจจุบันมีการแยกคิดตามขนาด และประเภทของการใช้ไฟฟ้า ตามภาคผนวก ก. ซึ่งจะมีให้เลือกใช้ได้ 2 ชนิด หลักๆคือ TOU และ TOD ทั้งสองชนิดจะแบ่งช่วงเวลาการคิดค่าไฟต่างกันปัจจุบันผู้ใช้ไฟฟ้ารายใหม่จะถูกบังคับให้ใช้อัตรา TOU เท่านั้น ส่วนรายเดิมที่ใช้ อัตราTOD สามารถ เปลี่ยนไปใช้อัตรา TOU ได้ แต่ถ้าเปลี่ยนแล้วไม่สามารถเปลี่ยนกลับไปใช้อัตรา TOD เหมือนเดิมได้

2.5.1 อัตรา TOD

คิดค่าพลังไฟฟ้า (กิโลวัตต์) แตกต่างกันเป็น 3 ช่วงเวลาของวัน และเหมือนกันทุกวัน คือ

* ช่วง On Peak (P) หมายถึง เวลา 18.30 น. - 21.30 น.
* ช่วง Partial Peak (PP) หมายถึงเวลา 08.00 น. - 18.30 น.
* ช่วง Off Peak (OP) หมายถึง เวลา 21.30 น. - 08.00 น.

แต่จะคิดค่าความต้องการพลังไฟฟ้าเฉพาะกิโลวัตต์ที่ใช้ในช่วง Peak และ Partial Peak (คิดเฉพาะส่วนที่เกิน Peak) สำหรับค่าพลังงานไฟฟ้า (หน่วย) เท่ากันตลอดทั้งวัน

2.5.2 อัตรา TOU

คิดค่าพลังงานไฟฟ้าต่างกันตามช่วงเวลาของการใช้ คือ

* ช่วง On Peak (P) หมายถึงเวลา 09.00 น. - 22.00 น.ของวันจันทร์-ศุกร์
* ช่วง Off Peak (OP) หมายถึงเวลา 22.00 น. - 09.00 น. ของวันจันทร์-ศุกร์ และวันเสาร์ อาทิตย์ วันหยุดราชการตามปกติ (ยกเว้นวันหยุดชดเชย) ทั้งวัน แต่ค่าความต้องการพลังไฟฟ้าคงที่โดยคิดเฉพาะกิโลวัตต์ที่ใช้ในช่วง On Peak

2.5.3. ค่าเพาเวอร์แฟคเตอร์ (P.F.)

กฟภ. จะเรียกเก็บค่าเพาเวอร์แฟคเตอร์ (P.F.) กับผู้ใช้ไฟฟ้า ประเภทที่ 3 กิจการขนาดกลาง ประเภทที่ 4 กิจการขนาดใหญ่ และประเภทที่ 5 กิจการเฉพาะอย่าง โดยจะคิดค่า P.F. จากค่ากิโลวาร์สูงสุดเฉพาะในส่วนที่เกินจากร้อยละ 61.97 ของค่าความต้องการพลังงานไฟฟ้าสูงสุดในอัตรา

กิโลวาร์ละ 14.02 บาท (เศษของกิโลวาร์ถ้าไม่ถึง 0.5 กิโลวาร์ตัดทิ้ง ตั้งแต่ 0.5 กิโลวาร์ขึ้นไปคิดเป็น 1 กิโลวาร์)

2.5.4. ค่า Ft

ค่า Ft คือ ค่าความผันแปรที่ปรับเพิ่มขึ้น หรือ ลดลง ตามภาวะต้นทุนการผลิต การส่ง และการจำหน่ายที่อยู่นอกเหนือการควบคุมของการไฟฟ้าที่เปลี่ยนแปลงไปจากต้นทุนที่กำหนดไว้ในค่าไฟฟ้าพื้นฐาน ได้แก่ ค่าเชื้อเพลิงที่ใช้ในการผลิตไฟฟ้า อัตราแลกเปลี่ยนเงินตราต่างประเทศ อัตราเงินเฟ้อ และหน่วยจำหน่ายที่เปลี่ยนแปลงไปจากที่ประมาณการไว้ในการกำหนดอัตราค่าไฟฟ้าฐาน จะคิดกับผู้ใช้ไฟฟ้าทุกประเภทในอัตราเท่ากันทุกหน่วย ค่า Ft โดยปกติจะมีการเปลี่ยนแปลงทุก 4 เดือนเพื่อไม่ให้ค่าไฟฟ้าผันผวนมากเกินไป

บทที่ 3
ทฤษฎีพื้นฐานของโปรแกรมการติดต่อสื่อสาร

อุปกรณ์ฮาร์ดแวร์ (Hardware) จำเป็นต้องมีส่วนควบคุมเพื่อให้สามารถทำงานได้ถูกต้อง การควบคุมมีหลายชนิด เช่น ใช้คนควบคุม, วงจรอิเล็คทรอนิคส์, โปรแกรม ฯลฯ ในปัจจุบันงานควบคุมที่ยุ่งยากซับซ้อน มีจุดอินพุท เอาท์พุท หลายจุด และต่างชนิดกัน ไม่สามารถใช้ระบบพื้นฐานควบคุมได้ หรือควบคุมได้แต่มีขนาดใหญ่ ช้า และขาดความน่าเชื่อถือ เทคโนโลยีดิจิตอลได้พัฒนาให้มีความเร็วและความน่าเชื่อถือสูงขึ้นมาก ง่ายต่อการโปรแกรม ระบบส่วนใหญ่จึงหันมาใช้การควบคุมในแบบดิจิตอล การโปรแกรมคือการเปลี่ยนโครงสร้างหน่วยความจำให้กลายเป็นรหัสคำสั่งที่ใช้ในการควบคุมโดยมีหน่วยประมวลผล (CPU) เป็นตัวอ่านคำสั่งเหล่านั้นไปดำเนินการ ความเข้าใจการทำงานของฮาร์ดแวร์ และเรียนรู้โครงสร้างโปรแกรมจะช่วยให้สามารถเขียนคำสั่งควบคุมได้อย่างถูกต้อง กระชับ และรวดเร็ว

3.1 หลักการเขียนโปรแกรม

การเขียนโปรแกรมควบคุมต้องคำนึงถึงโครงสร้างทางกายภาพของฮาร์ดแวร์เป็นสำคัญ เพราะเป้าหมายของการโปรแกรมคือสัญญาณควบคุมอุปกรณ์ภายนอก ดังนั้นก่อนจะเริ่มเขียนโปรแกรมควรศึกษาส่วนประกอบ และการทำงานของอุปกรณ์ที่ต่อร่วมกับฮาร์ดแวร์ที่ใช้ควบคุมด้วยเสมอ จะทำให้สามารถสร้างคำสั่งไปควบคุมได้ถูกต้อง หลักสำคัญอีกข้อของการเขียนโปรแกรมก็คือ ลักษณะการประมวลผล จำเป็นอย่างยิ่งที่ต้องรู้ว่าโปรแกรมที่เขียนขึ้นถูกประมวลผลในลักษณะไหนเพื่อที่จะสามารถสร้างโปรแกรมให้ตอบสนองการประมวลผลนั้นได้ เช่น โปรแกรมไมโครคอนโทรลเลอร์มีการประมวลผลในลักษณะโครงสร้าง (Structure) คือหน่วยประมวลผลจะอ่านโปรแกรมทีละบรรทัดแล้วนำข้อมูลและคำสั่งนั้นไปประมวลผล หลังจากนั้นก็จะขยับไปอ่านคำสั่งถัดไป เพื่อทำงานต่อ ทำงานอย่างนี้วนไปเรื่อยๆ ส่วนโปรแกรมที่เขียนบนระบบปฏิบัติการ Windows นั้นจะทำงานแบบ Multi Tasking คือแบ่งช่วงเวลาและทรัพยากรทั้งหมดออกเป็นส่วนย่อยๆให้กับโปรแกรมที่รันอยู่ในระบบ คำว่าทรัพยากรหมายถึง หน่วยความจำ และ สิ่งอื่นๆที่ใช้ในการประมวลผลคำสั่ง คล้ายๆกับ เวลาทั้งหมดคือลูกแตงโม 1 ลูก แล้วตัดแบ่งให้เด็กๆ จำนวนเด็กก็คือจำนวนโปรแกรมที่รันอยู่ในระบบ แล้วคนแบ่งก็ผลัดป้อนแตงโมให้เด็กทีละคนเรียงกันไปตามช่วงเวลา ดังนั้นในระบบหนึ่งสามารถที่จะรันโปรแกรมได้หลายโปรแกรม ส่วนหน่วยประมวลผลจะทำหน้าที่ผลัดไปประมวลโปรแกรมตามช่วงเวลาที่เป็นรอบของโปรแกรมนั้นๆ ภายในโปรแกรมยังแบ่งเป็นโมดูลย่อยๆอีกหลายตัวเรียกว่า อ็อบเจ็ค (Object) ซึ่งแต่ละอ็อบ

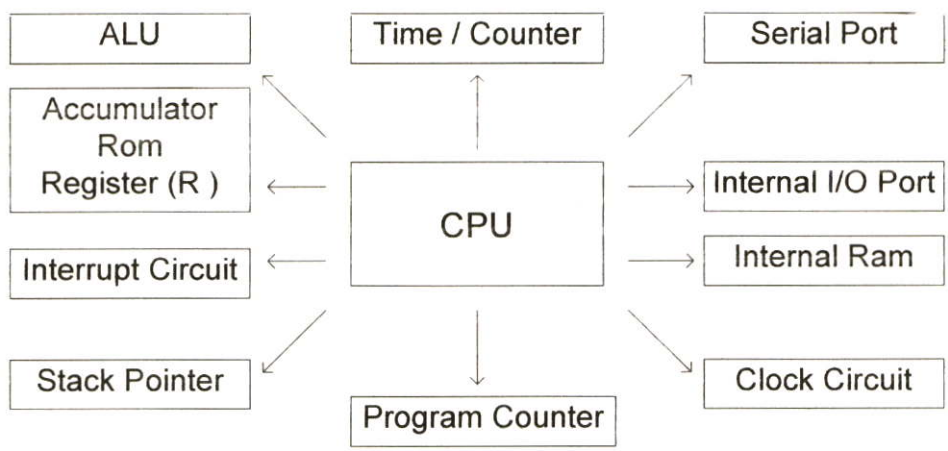
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เจ็คจะทำงานเป็นอิสระต่อกันขึ้นอยู่กับเหตุการณ์ที่เข้ามากระตุ้น ดังนั้นการเขียนโปรแกรมจะต้องเขียนเป็นโปรแกรมย่อยเพื่อจัดการกับเหตุการณ์นั้นๆ การเขียนโปรแกรมลักษณะนี้เรียกว่า การโปรแกรมตามเหตุการณ์พาไป (Even Driven) เมื่อทราบลักษณะการประมวลผลแล้วก็เริ่มเขียนโปรแกรมให้มีลักษณะสอดคล้องกับการประมวลผลนั้นๆ

3.2 โปรแกรมไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์เป็นโปรเซสเซอร์ที่รวมเอาส่วนของหน่วยความจำ ไอโอพอร์ท Timer/Counter และ โมดูลอื่นๆมาบรรจุในชิพเดี่ยวมีชื่อเรียกว่า ซิงเกิลชิพไมโครคอนโทรลเลอร์ (Single Chip Microcontrollers) ปัจจุบันสามารถโปรแกรมได้โดยตรงจากคอมพิวเตอร์ ผ่านพอร์ทอนุกรม หรือพอร์ทขนาน

3.2.1 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์



รูปที่ 3.1 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์

จากรูปที่ 3.1 จะเห็นว่า ไมโครคอนโทรลเลอร์จะประกอบไปด้วย

1. ALU
2. แอคคิวมูเลเตอร์ (ACC)
3. รีจิสเตอร์ (Register)
4. แรมภายในชิพ (Internal RAM)
5. สแตคพอยเตอร์ (SP)
6. หน่วยความจำรอมภายใน (Internal ROM)

76669

7. โปรแกรมเคาน์เตอร์ (PC)
8. ไอโอพอร์ท (I/O Port)
9. วงจรอินเตอร์รัพต์ (Interrupt Circuit)
10. วงจรคล็อค (Clock Circuit)
11. ไทม์เมอร์เคาน์เตอร์ (Timer/Counter)
12. พอร์ทอนุกรม (Serial Port)

โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ตระกูลMCS-51 ในไมโครคอนโทรลเลอร์บางเบอร์ยังอาจใส่โมดูลช่วยอื่นๆเพิ่มเข้าไปได้อีกเพื่อความสะดวกในการใช้งานเฉพาะ เช่น ADC (Analog to Digital Converter), Watch Dog, PWM ฯลฯ

ในวิทยานิพนธ์ฉบับนี้นำเสนอการเขียนโปรแกรมของตระกูล MCS-51 เท่านั้น การเขียนโปรแกรมจำเป็นต้องรู้คำสั่งทั้งหมดซึ่งชุดคำสั่งของ 8051 มีทั้งหมด 256 คำสั่งสามารถดูได้จากภาคผนวก ข. วิธีการแปลงภาษาแอสแซมบลีให้เป็นภาษาเครื่องสามารถดำเนินการได้ 2 แบบ คือ การทำด้วยมือ และใช้โปรแกรมแอสแซมเบลอร์ เช่น SXA-51, Cross-16, Cross-32 ฯลฯ โดยจะเขียนโปรแกรมแล้วพิมพ์เก็บไว้ใน File โดยตั้งชื่อให้มีนามสกุลเป็นจุด ASM โปรแกรมที่ใช้พิมพ์อาจจะใช้ Notepad หรือ Word ก็ได้ (โปรแกรม Editor นั่นเอง)

3.2.2 การเขียนโปรแกรมภาษาแอสแซมบลี

การเขียนโปรแกรมแอสแซมบลีเพื่อใช้กับไมโครคอนโทรลเลอร์ที่ดีจะต้องคำนึงถึงขนาดของหน่วยความจำแรมภายในก่อนเป็นอันดับแรก เนื่องจากหน่วยความจำแรมภายในมีขนาดจำกัด โดยทั่วไปแล้วจะมีขนาด 128 หรือ 256 ไบต์ ขึ้นอยู่กับเบอร์ที่ใช้ บางเบอร์อาจมีแรมส่วนขยาย (XRAM) เพิ่มเข้ามาเพื่อช่วยเก็บข้อมูลเพิ่ม ถ้าไม่พอจะต้องเพิ่มขนาดหน่วยความจำภายนอก วิธีนี้จะทำให้วงจรมีขนาดใหญ่ขึ้น ดังนั้นขั้นตอนแรกของการออกแบบจะต้องกำหนดก่อนว่าวงจรที่ใช้ต้องการใช้หน่วยความจำประมาณเท่าไหร่ ต้องการโมดูลช่วยภายในอะไรบ้างเมื่อกำหนดได้แล้วก็นำไปเปิดแคตตาล็อคหาเบอร์ไมโครคอนโทรเลอร์ที่เหมาะสม เพื่อลดขนาดวงจรให้เล็กลง

หลังจากกำหนดเบอร์ไมโครคอนโทรลเลอร์และต่อวงจรเรียบร้อยแล้ว ก็เริ่มเขียนโปรแกรมโดยต้องกำหนดโครงสร้างโปรแกรมคร่าวๆในรูปแบบโฟวล์ชาร์ทก่อน การเขียนโฟวล์ชาร์ทสามารถหาอ่านได้ตามหนังสือการเขียนโปรแกรมทั่วไป หลังจากนั้นก็เข้าไปวิเคราะห์ว่าแต่ละบล็อกต้องใช้โปรแกรมย่อยอะไรบ้าง และแต่ละโปรแกรมย่อยต้องมีการรับ-ส่งค่าพารามิเตอร์อะไรบ้าง แล้วนำค่าพารามิเตอร์ที่ใช้มาจองหน่วยความจำแรมภายในเพื่อใช้เป็นที่พักข้อมูลสำหรับการคำนวณของโปรแกรมย่อย สามารถสรุปขั้นตอนการออกแบบ และเขียนโปรแกรมได้ดังนี้

1. สรุปคุณสมบัติ และความสามารถของวงจรที่ต้องการออกแบบ

2. ศึกษาค้นคว้าการทำงานของวงจร และค้นหาอุปกรณ์ฮาร์ดแวร์ที่มีขายในท้องตลาดให้มีความเหมาะสมกับวงจรที่ออกแบบมากที่สุด

3. ออกแบบ และสร้างวงจร

4. ออกแบบโครงสร้างวงจรแบบกว้างๆให้เห็นลำดับขั้นการทำงาน แล้วเขียนในรูปโฟล์ทชาร์ท โดยแบ่งให้ชัดเจนว่าส่วนไหนทำงานร่วมกับโปรแกรมหลัก (Main Program) และส่วนไหนทำงานด้วยสัญญาณอินเทอร์รัพท์ (Interrupt)

5. วิเคราะห์การทำงานของแต่ละบล็อก แล้วทำสรุปออกมาว่าแต่ละบล็อกใช้โปรแกรมย่อยอะไรบ้าง ใช้ฟังก์ชั่นอะไรบ้าง มีการรับ-ส่งข้อมูลผ่านตัวแปรกี่ตัว

6. เขียนโฟวล์ชาร์ทย่อยที่แสดงการทำงานของโปรแกรมย่อยแต่ละตัว รวมไปถึงชื่อตัวแปรที่ต้องใช้ในโปรแกรมย่อยนั้นๆด้วยเพื่อให้สามารถรู้ได้ในภายหลังว่าโปรแกรมย่อยนั้นรับค่าผ่านตัวแปรอะไร และคืนค่าผ่านตัวแปรอะไร

7. เริ่มเขียนโปรแกรมทีละบล็อก โดยเริ่มทีละโปรแกรมย่อยแล้วทดสอบด้วยการผ่านค่าให้กับตัวแปรที่ใช้รับค่า แล้วนำค่าที่เป็นคำตอบหรือค่าที่โปรแกรมย่อยต้องคืนค่ากลับมาส่งออกไปแสดงผลยังพอร์ทภายนอก หรืออาจส่งออกทางพอร์ทอนุกรมไปแสดงผลที่คอมพิวเตอร์ก็ได้เพื่อตรวจสอบว่าโปรแกรมที่เขียนสามารถทำงานได้ถูกต้องหรือไม่

8. เมื่อเขียนโปรแกรมย่อยครบแล้วก็เขียนโปรแกรมหลักให้มีลำดับขั้นการทำงานเหมือนกับโฟวล์ชาร์ทที่ออกแบบไว้ข้างต้น พร้อมทั้งสร้างลิ้ง (Link) ให้กับโปรแกรมย่อยที่ตอบสนองอินเทอร์รัพท์ให้ครบ

9. ทดลองรันจริงในระบบ

10. ประเมินผลการทดลอง ทำสรุปการแก้ไขเพิ่มเติม

11. แก้ไข และทดลองซ้ำ

3.3 โปรแกรมคอมพิวเตอร์

โปรแกรมคอมพิวเตอร์ที่จะกล่าวถึงในบทความนี้อ้างถึงโปรแกรม Delphi เท่านั้นเนื่องจากเป็นโปรมแกรมที่ใช้เขียนแอปพลิเคชั่น (Application) บนคอมพิวเตอร์ทั้งหมดในวิทยานิพนธ์ฉบับนี้

โปรแกรมDelphi เป็นโปรแกรมที่สามารถสร้างให้รันข้ามเพลทฟอร์ม (Plat Form) ได้ แต่นิยมสร้างให้ใช้งานบนระบบปฏิบัติการวินโดว์ (Windows) ลักษณะการทำงานของโปรแกรมเป็นโปรแกรมเชิงวัตถุ (OOP) คือ สร้างออกมาในรูปวัตถุที่มีข้อมูล และการกระทำสำเร็จอยู่ในตัวเดียวกัน โปรแกรมหนึ่งๆประกอบไปด้วยวัตถุตั้งแต่ 1 ขึ้นไปทำงานร่วมกัน วัตถุแต่ละตัวถูกกระตุ้นการทำงานจาก วินโดว์เมจเสจ (Windows message)

Delphi นั้นเป็นเครื่องมือพัฒนาแอพพลิเคชั่นแบบ Visual Programming (เหมือนกับ Visual Basic, Visual C++, PowerBuilder ฯลฯ) ซึ่งทำให้เราสามารถเห็นผลลัพธ์การทำงานไปพร้อม ๆ กับการลงมือสร้างแอพพลิเคชั่น

จุดเด่นที่สำคัญมากของความเป็น Visual Programming คือช่วยลดเวลาของการสร้างแอพพลิเคชั่น นั่นเพราะแทนที่เราจะไปทุ่มเวลาไปปรับแต่งส่วนติดต่อผู้ใช้ หรืองานที่ไม่จำเป็นหรืองานซ้ำ ๆ ซาก ๆ เราก็มอบภาระเหล่านี้ให้ โปรแกรมเสีย สำหรับเราก็มุ่งเข้าไปแก้ไขปัญหาที่เป็นหัวใจการทำงานของแอพพลิเคชั่นดีกว่า จุดเด่นที่ว่านั้นพอจะยกมากล่าวได้หลายข้อด้วยกันดังนี้

1. ขนาดไฟล์แอพพลิเคชั่นที่สร้างมีขนาดเล็ก ใช้ทรัพยากรของระบบน้อย และทำงานได้อย่างสมบูรณ์ในตัวโดยไม่ต้องมีไฟล์พิเศษอื่น ๆ เพิ่มเติมนั่นทำให้ความเร็วในการทำงานของแอพพลิเคชั่นที่พัฒนาโดย Delphi สามารถเทียบได้กับการพัฒนาแอพพลิเคชั่นโดยใช้ภาษาซี

2. โปรแกรมเมอร์สามารถสร้างและใช้งานฟังก์ชั่นไลบรารี่ที่เรียกว่า DLL (Dynamic Link Library) ได้ รวมทั้งยังสามารถติดต่อกับ DLL ที่สร้างจากภาษาอื่น ๆ ได้ด้วย

3. ความสามารถในการสร้างแอพพลิเคชั่นสำหรับทำงานบนเครือข่ายอินเทอร์เน็ตเป็นความสามารถที่Delphiในเวอร์ชั่นหลัง ๆ อย่าง Delphi 5.0 เน้นหนักมากขึ้นเพิ่มเติมจากความสามารถในการพัฒนาแอพพลิเคชั่นที่เกี่ยวกับฐานข้อมูลและแอพพลิเคชั่นหรับการใช้งานทั่วไป

4. สามารถเขียนแอสเซมบลี (Assembly) เพื่อติดต่อกับภาคฮาร์ดแวร์ของเครื่องคอมพิวเตอร์ ได้โดยไม่ต้องไปอาศัยพึ่งโปรแกรมอื่น ๆ ช่วย

5. เปิดโอกาสให้โปรแกรมเมอร์สามารถสร้างคอมโพเนนต์ต่าง ๆ ไว้ใช้งานเองได้อย่างอิสระ

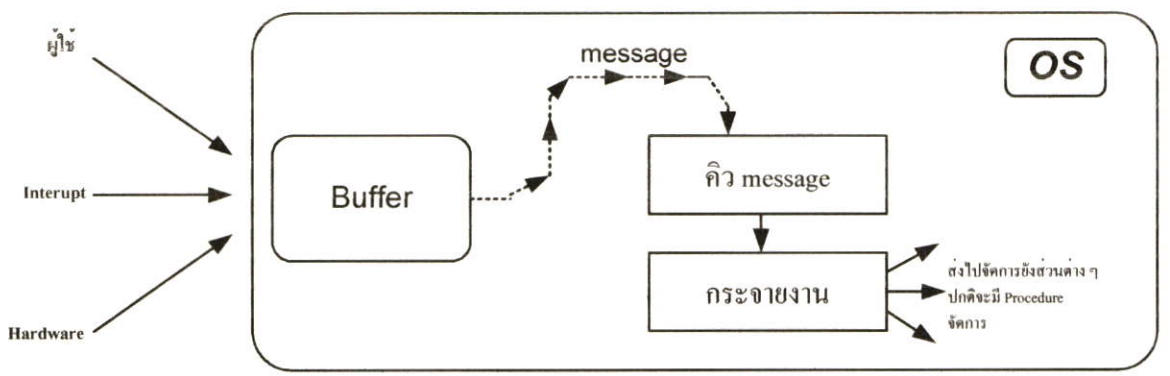
3.3.1 แอพพลิเคชั่นแบบ Event-Driven

แอพพลิเคชั่นที่สร้างจากDelphi นั้นมีวิธีการสร้างที่แตกต่างจากการเขียนโปรแกรมแบบเดิม ๆ ที่เราเคยเรียนรู้มาก่อน หากเราอาจเคยเขียนโปรแกรมจากภาษา BASIC, C หรือ Pascal ก็จะเห็นว่ามีแนวคิดที่ค่อนข้างต่างกัน ซึ่งเราจะเรียกวิธีการที่สร้างแอพพลิเคชั่นด้วย Delphi ว่า Event-Driven (แปลว่า เหตุการณ์พาไป)

Event-Driven ที่จริงก็คือการเขียนโปรแกรมในลักษณะที่ว่า "ถ้ามีเหตุการณ์เกิดขึ้นเราจะจัดการกับมันอย่างไร" เช่น ถ้าผู้ใช้คลิกที่ปุ่ม Exit เราจะทำอย่างไร? เราอาจจะถามผู้ใช้ว่า แน่ใจแล้วนะว่าจะจบการทำงานของโปรแกรม ถ้าผู้ใช้ยืนยันก็จบไป แต่ถ้าไม่ก็ให้แอพพลิเคชั่นทำงานต่อไป เป็นต้น

3.3.2 ระบบ Message Loop ของ Windows

Event-Driven ถือได้ว่าเป็นเหมือนหลักการทำงานของ Windows ก็ว่าได้ ที่จริงแล้วเบื้องหลังการทำงานของ Windows ก็คือจะมีการรอรับการทำงานจากผู้ใช้จากฮาร์ดแวร์ต่างๆ ส่งเหตุการณ์ต่าง ๆ เข้ามาสู่ Message Loop ซึ่งจะพิจารณาว่ามีเหตุการณ์อะไรเกิดขึ้น ซึ่งแต่ละเหตุการณ์ Windows ได้เตรียมวิธีการจัดการแต่ละเหตุการณ์เอาไว้แล้ว



รูปที่ 3.2 แนวคิดของ Message Loop ของ Windows

3.3.3 ออบเจ็กต์ (Object)

แอพพลิเคชั่นประกอบไปด้วย วินโดว์, ปุ่มกด หรือตัวเลือก ซึ่งจะเรียกสิ่งต่าง ๆ ที่นำมาสร้างเป็นแอพพลิเคชั่นนี้ว่า ออบเจ็กต์ (Object) แต่ละออบเจ็กต์จะมีองค์ประกอบสมบูรณ์ด้วยตัวเองกล่าวคือมีข้อมูล และการทำงานเป็นของตัวเองประกอบด้วย

3.3.3.1 พรอพเพอร์ตี้ (Property)

จะเห็นได้ว่าแต่ละออบเจ็กต์มีลักษณะเฉพาะตัวที่แสดงความเป็นออบเจ็กต์ ซึ่งเราเรียกลักษณะเฉพาะตัวนี้ว่า พรอพเพอร์ตี้ (Property)

ในออบเจ็กต์ชนิดเดียวกันก็จะมีพรอพเพอร์ตี้อย่างเดียวกัน แต่อาจจะมีค่าของแต่ละพรอพเพอร์ตี้ที่แตกต่างกัน โดยเราสามารถกำหนดค่าพรอพเพอร์ตี้ได้ใน Object Inspector

3.3.3.2 เมธอด (Method)

นอกจากจะใช้พรอพเพอร์ตี้ในการบอกความแตกต่างของออบเจ็กต์แต่ละตัวแล้วยังมีอีกสิ่งหนึ่งที่ออบเจ็กต์แต่ละตัวมักจะมีนั่นก็คือ เมธอด (Method)

เมธอดก็คือความสามารถในการทำงานอย่างใดอย่างหนึ่งของออบเจ็กต์ เช่น ออบเจ็กต์ Form มีความสามารถในการวาดวงกลมได้เป็นต้น ในการเรียกใช้เมธอดเราจะเรียกโดยการเขียนโปรแกรม

3.3.3.3 อีเวนต์ (Event)

อีเวนต์ก็คือเหตุการณ์ที่เกิดขึ้นกับแต่ละออบเจ็กต์ เช่น ปุ่ม ก็จะมีเหตุการณ์ที่ผู้ใช้คลิกเมาส์ (อีเวนต์ OnClick) หรือกรอบข้อความจะมีเหตุการณ์เมื่อผู้ใช้กรอกข้อความ (อีเวนต์ OnKeypress)

เราสามารถเขียนคำสั่งต่างๆเพื่อจัดการกับเหตุการณ์ที่เกิดขึ้นโดยการเลือกเหตุการณ์ที่ต้องการจากแท็บ Events ใน Object Inspector

3.3.3.4 ออบเจ็กต์ กับ คอมโพเนนต์ของDelphi

ข้อดีของการเขียนโปรแกรมเพื่อสร้างแอพพลิเคชั่นใน Delphi ก็คือ เมื่อเราสมมุติสิ่งต่าง ๆ เป็นออบเจ็กต์ที่มีทั้งพรอพเพอร์ตี้, เมธอดแล้ว เราก็สามารถนำมาใช้งานได้เรื่อย ๆ (Reusable Object) ทำให้ไม่ต้องเสียเวลามาสร้างกันใหม่ทุกครั้งที่สร้างแอพพลิเคชั่น

Delphi จะเรียกออบเจ็กต์ต่าง ๆ ที่สามารถนำกลับมาใช้ในการสร้างแอพพลิเคชั่นว่า คอมโพเนนต์ (Component) ซึ่งจะถูกเก็บไว้ใน Component Palette พร้อมให้เรานำมาใช้งาน

3.3.4 ขั้นตอนการสร้างแอพพลิเคชั่น

ขั้นแรก	ออกแบบการทำงาน และหน้าตาของแอพพลิเคชั่น
ขั้นที่สอง	วางคอมโพเนนต์ที่ต้องการลงในฟอร์ม
ขั้นที่สาม	กำหนดพรอพเพอร์ตี้ให้แต่ละคอมโพเนนต์
ขั้นที่สี่	เขียนคำสั่งกำกับการทำงานของแต่ละคอมโพเนนต์
ขั้นที่ห้า	ทำการคอมไพล์
ขั้นที่หก	ทดสอบการทำงานของแอพพลิเคชั่น
ขั้นที่เจ็ด	บันทึกแอพพลิเคชั่นที่สร้างขึ้นเก็บไว้

3.3.5 ลักษณะการเขียนโปรแกรมที่ดี

การเขียนโปรแกรมเป็นเรื่องของการถ่ายทอดความคิดแล้วนำมาสั่งให้คอมพิวเตอร์ทำงานตามที่เราต้องการ แต่ว่าโปรแกรมที่เราสร้างขึ้นอาจจะไม่ได้ถูกปรับปรุงแก้ไขด้วยตัวเราเอง หรือเราอาจจะต้องทำการเขียนโปรแกรมร่วมกับผู้อื่นแบบเป็นทีม การเขียนโปรแกรมให้อ่านแล้วเข้าใจง่ายจึงเป็นเรื่องที่ควรคำนึง

แม้ว่าลักษณะการเขียนโปรแกรมที่ดีจะตัดสินได้ยากเพราะเราอาจทำได้แต่เพียงรวบรวมลักษณะที่ดีที่ควรทำตามหรือลักษณะที่ไม่ดีที่ควรหลีกเลี่ยงเท่านั้น

3.3.6 การประยุกต์ใช้ Delphi

ในบทความนี้เป็นการแนะนำวิธีการใช้งานซอฟต์แวร์พัฒนาแอพพลิเคชั่นแบบวิชวลที่มีชื่อว่าDelphi โดยเป้าหมายหลักจะมุ่งไปที่การประยุกต์ใช้Delphiกับการเขียนโปรแกรมทางวิศวกรรม Delphiจัดได้ว่าเป็นหนึงในเครื่องมือสำหรับการสร้างแอพพลิเคชันบนระบบปฏิบัติการวินโดวส์ หรือเรียกอีกอย่างว่า เครื่องมือพัฒนาแอพพลิเคชั่นแบบวิชวลโปรแกรมมิ่ง (Visual Programming) ซึ่งเป็นรูปแบบหนึ่งของภาษาคอมพิวเตอร์ที่แตกต่างไปจากภาษาคอมพิวเตอร์แบบเดิม ๆ อย่างภาษาเบสิก (Basic), ภาษาซี (C), หรือภาษาปาสคาล (Pascal)

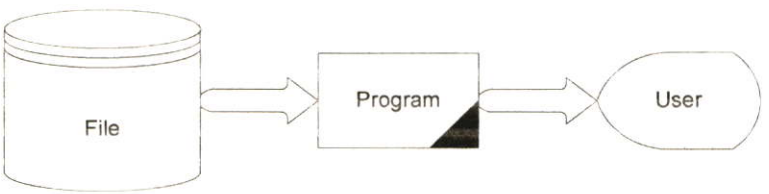
ปัจจุบันการพัฒนาแอพพลิเคชั่นด้วยวิชาโปรแกรมมิ่งได้รับความนิยมอย่างมากเนื่องจากข้อดีตรงที่โปรแกรมเมอร์สามารถจะเห็นผลลัพธ์การทำงานไปพร้อม ๆ กับการลงมือสร้าง อีกทั้งยังมีจุดเด่นในแง่ของการลดเวลาที่ใช้ในการพัฒนาแอพพลิเคชั่นลงได้อย่างมากเนื่องจากโปรแกรมเมอร์จะไม่ต้องไปเสียเวลากับการออกแบบและทำส่วนติดต่อกับผู้ใช้หรือ GUI (Graphic User Interface) หรืองานอื่น ๆ ที่ไม่จำเป็นอีกต่อไป เพราะภาระเหล่านั้นทั้งหมดจะถูกผลักไปให้Delphiแทน ด้วยเหตุนี้โปรแกรมเมอร์จึงสามารถจะทุ่มเวลาให้กับส่วนที่เป็นหัวใจสำคัญของแอพพลิเคชั่นได้อย่างเต็มที่ แน่นอนว่าวิธีนี้ย่อมช่วยให้การพัฒนาแอพพลิเคชั่นบนคอมพิวเตอร์มีประสิทธิภาพมากขึ้น ทั้งยังช่วยให้การทำงานง่ายขึ้นและรวดเร็วมากขึ้นด้วย

3.4 ระบบฐานข้อมูล (Database System)

ข้อมูลที่เก็บอย่างมีระเบียบจะเพิ่มประสิทธิภาพในการเก็บ แก้ไข และค้นหาได้ การนำข้อมูลไปเก็บจะต้องมีการจัดเรียงข้อมูลนั้นๆเป็นหมวดหมู่ที่ชัดเจนและเหมาะสม ได้มีการพัฒนารูปแบบจากจัดเก็บข้อมูลจากการจัดเก็บในหน่วยความจำในตำแหน่งต่างๆมาเป็นจัดเก็บในรูปไฟล์ข้อมูล จนกระทั่งจัดเก็บในรูปดาต้าเบส

3.4.1 ระบบแฟ้มข้อมูล (File System)

ในอดีต องค์กรต่าง ๆ มักจัดเก็บเอกสารไว้ในแฟ้มเอกสารต่าง ๆ ซึ่งมีความเกี่ยวข้องกันทางด้านข้อมูลน้อยหรืออาจไม่มีเลย ขึ้นอยู่กับความต้องการในการใช้ข้อมูลนั้น ๆ เช่น ประวัติการรักษาพยาบาล ที่โดยทั่วไปมักจะแยกเก็บในแฟ้มเอกสารเฉพาะคนไข้แต่ละคน หรือประวัติพนักงาน ที่อาจจัดเก็บในแฟ้มเอกสารแยกตามฝ่ายที่สังกัด ซึ่งข้อมูลของพนักงานแต่ละคนในแต่ละแฟ้มเอกสารจะมีความเกี่ยวข้องกันตามฝ่ายที่สังกัด เช่น แฟ้มเอกสารประวัติพนักงานของฝ่ายธุรการ แฟ้มเอกสารประวัติพนักงานของฝ่ายการเงิน ฯลฯ เป็นต้น แต่ต่อมาเมื่อองค์กรมีขนาดใหญ่ขึ้นจากเดิมที่สามารถค้นหาเอกสารจากแฟ้มเอกสารเพียงแฟ้มเดียว ก็เริ่มต้องค้นหาเอกสารจากแฟ้มเอกสารต่าง ๆ จำนวนมากขึ้น ส่งผลให้การค้นหาเอกสารเป็นงานที่ต้องใช้เวลา และมีความยากลำบากมากขึ้น การจัดเก็บเอกสารในคอมพิวเตอร์จึงถูกริเริ่มนำมาใช้ในองค์กรแทนการจัดเก็บในรูปแบบเดิม แต่การจัดเก็บเอกสารในคอมพิวเตอร์ในยุคแรกๆ ยังคงไม่ค่อยมีประสิทธิภาพมากนัก เนื่องจากยังคงมีรูปแบบการจัดเก็บคล้ายกับการจัดเก็บเอกสารในลักษณะเดิมอยู่ โดยเป็นเพียงการนำเอาเอกสารต่าง ๆ ในแต่ละแฟ้มเอกสารมาจัดเก็บในรูปของแฟ้มข้อมูลแทน และด้วยเหตุนี้จึงจำเป็นที่จะต้องอาศัยผู้ที่มีความชำนาญ เช่น โปรแกรมเมอร์หรือนักวิเคราะห์ระบบ เข้ามาช่วยกำหนดโครงสร้างของแฟ้มข้อมูลเพื่อที่จะสามารถนำแฟ้มข้อมูลนั้นไปจัดเก็บข้อมูลและนำไปประมวลผลได้ตามความต้องการ

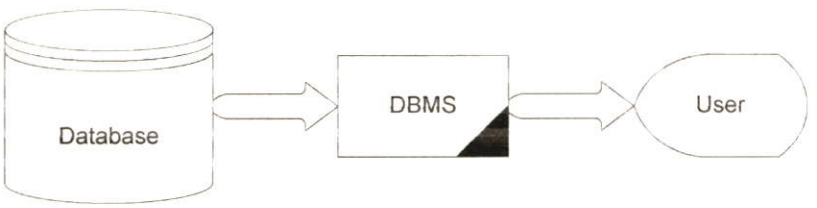


รูปที่ 3.3 โครงสร้างระบบแฟ้มข้อมูล

3.4.2 ความหมายของระบบฐานข้อมูล (Database System)

จากปัญหาต่าง ๆ ที่เกิดขึ้นในระบบแฟ้มข้อมูล ได้ก่อให้เกิดการจัดเก็บข้อมูลในรูปแบบใหม่ขึ้นที่เรียกว่า "ฐานข้อมูล (Database)" การจัดเก็บข้อมูลในฐานข้อมูลนี้จะแตกต่างจากการจัดเก็บข้อมูลแบบแฟ้มข้อมูล เนื่องจากฐานข้อมูลเป็นการนำเอาข้อมูลต่างๆ ที่มีความสัมพันธ์กันซึ่งแต่เดิมจัดเก็บอยู่ในแต่ละแฟ้มข้อมูลมาจัดเก็บไว้ในที่เดียวกัน เช่น ข้อมูลพนักงาน สินค้าคงคลัง พนักงานขาย และลูกค้า ซึ่งแต่เดิมถูกจัดเก็บอยู่ในรูปของแฟ้มข้อมูลของฝ่ายต่างๆ ได้ถูกนำมาจัดเก็บรวมกันไว้ภายในฐานข้อมูลเดียว ซึ่งเป็นฐานข้อมูลรวมของบริษัท ส่งผลให้แต่ละฝ่ายสามารถใช้ข้อมูลร่วมกัน และสามารถแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นในระบบแฟ้มข้อมูลได้

ข้อมูลต่าง ๆ ที่ถูกจัดเก็บเป็นฐานข้อมูล นอกจากจะต้องเป็นข้อมูลที่ความสัมพันธ์กันแล้ว ยังจะต้องเป็นข้อมูลที่ใช้สนับสนุนการดำเนินงานอย่างน้อยอย่างใดอย่างหนึ่งขององค์กร ดังนั้นอาจกล่าวได้ว่าแต่ละฐานข้อมูลจะเทียบเท่ากับระบบแฟ้มข้อมูล 1 ระบบ และจะเรียกฐานข้อมูลที่จัดทำขึ้นเพื่อสนับสนุนการดำเนินงานอย่างใดอย่างหนึ่งนั้นว่า "ระบบฐานข้อมูล" (Database System) เช่น ระบบฐานข้อมูลเงินเดือน ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่างๆ ที่สนับสนุนการคำนวณเงินเดือน หรือระบบฐานข้อมูลค่าทางไฟฟ้า ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่างๆ ที่สนับสนุนการแสดงผล วิเคราะห์ระบบไฟฟ้า เป็นต้น



รูปที่ 3.4 โครงสร้างระบบฐานข้อมูล

ระบบฐานข้อมูลจำเป็นต้องมีตัวกลางในการเข้าถึงข้อมูลเพื่อติดต่อแลกเปลี่ยนหรือค้นหาเพื่อความรวดเร็ว รูปที่ 3.4 ผู้ใช้งานสามารถเข้าถึงข้อมูลได้โดยผ่านตัวกลาง DBMS โปรแกรม DBMS นี้จะทำหน้าที่ช่วยในการจัดการไฟล์ในฐานข้อมูล เช่น การสืบค้นข้อมูล, การเรียงลำดับข้อมูล, การจัดารอินเด็กซ์ (Index) และจัดระเบียบข้อมูล ดังนั้นตัวโปรแกรมที่เรียกใช้ฐานข้อมูลผ่าน DBMS จึงมีขนาดเล็กลงเนื่องจากระบบจัดการข้อมูลส่วนใหญ่ถูกส่งให้โปรแกรม DBMS ทำงานแทน

3.4.3 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลโดยทั่วไป จะเกี่ยวข้องกับ 4 ส่วนหลัก ๆ ดังนี้

1. ข้อมูล (Data)

2. ฮาร์ดแวร์ (Hardware) อุปกรณ์ทางคอมพิวเตอร์ที่มีส่วนเกี่ยวข้องกับระบบฐานข้อมูล จะประกอบด้วย 3 ส่วนคือหน่วยความจำสำรอง (Secondary Storage), หน่วยประมวลผล และหน่วยความจำหลัก

3. ซอฟท์แวร์ (Software)

4. ผู้ใช้ระบบฐานข้อมูล (User) ผู้ที่เรียกใช้ข้อมูลจากระบบฐานข้อมูลมาใช้งาน สามารถแบ่งออกเป็น 3 กลุ่มได้ดังนี้

- Application Programmer
- End User แบ่งออกเป็น 2 กลุ่มดังนี้
  - Naive User
  - Sophisticated User
- Database Administrator (DBA)

3.4.4 ความเป็นอิสระของข้อมูล (Data Independence)

ข้อมูลจะต้องเป็นอิสระจากโปรแกรมที่เรียกได้ ในการกำหนดให้ข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ จะแบ่งออกเป็น 2 ระดับดังนี้

1. ระดับ Physical เป็นระดับที่โครงสร้างทางกายภายของข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ คือตัวโปรแกรมที่เรียกใช้ฐานข้อมูลจะไม่มีความเกี่ยวข้องกับตัวฐานข้อมูลเลย แต่สามารถเรียกใช้ข้อมูลในฐานข้อมูลได้โดยผ่านตัวกลาง หรือระบบสืบค้น ในระดับนี้การเข้าถึงข้อมูลในฐานข้อมูลสามารถใช้โปรแกรมใดก็ได้ที่สนับสนุนฐานข้อมูล และมีการอ้างถึงฐานข้อมูลได้ถูกต้อง ในวิทยานิพนธ์ฉบับนี้ใช้การเข้าถึงฐานข้อมูลโดยการอ้างถึงตัวฐานข้อมูลผ่าน Alias ซึ่งเป็นไฟล์ที่จะอ้างไปถึงตำแหน่งจริงของข้อมูลบนหน่วยความจำ

2. ระดับ Logical เป็นระดับที่ความสัมพันธ์ระหว่างข้อมูลในส่วนต่างๆ ภายในฐานข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ ซึ่งในระดับ Logical จะทำได้ยากกว่าในระดับ Physical เนื่องจากความสัมพันธ์ระหว่างข้อมูลในส่วนต่างๆ ภายในแฟ้มข้อมูลเดียวกันจะมีความเกี่ยวพันกับโปรแกรมมากกว่าโครงสร้างทางกายภาพของข้อมูล

3.4.5 โปรแกรมตัวกลางฐานข้อมูล (Database Management System)

เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ภายในฐานข้อมูล ซึ่งต่างจากระบบแฟ้มข้อมูลที่หน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อกับข้อมูลในฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มคำสั่ง DML หรือ DDL หรือจะด้วยโปรแกรมต่าง ๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูลจะถูกโปรแกรม DBMS นำมาแปล (Compile) เป็นการกระทำ (Operation) ต่าง ๆ ภายใต้คำสั่งนั้น ๆ เพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไป สำหรับส่วนการทำงานต่าง ๆ ภายในโปรแกรม DBMS ที่ทำหน้าที่ในการแปลคำสั่งไปเป็นการกระทำต่าง ๆ ที่จะกระทำกับข้อมูลนั้น ประกอบด้วยส่วนการทำงานต่าง ๆ ดังนี้

1. Database Manager
2. Query Processor
3. Data Manipulation Language Precompiled
4. Data Definition Language Precompiled
5. Application Programs Object Code

3.4.6 หน้าที่ของโปรแกรม DBMS

สำหรับหน้าที่ของโปรแกรม DBMS มีดังนี้

1. ทำหน้าที่แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูล ให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจ

2. ทำหน้าที่ในการนำคำสั่งต่าง ๆ ซึ่งได้รับการแปลแล้ว ไปสั่งให้ฐานข้อมูลทำงาน เช่น การเรียกใช้ข้อมูล (Retrieve) การจัดเก็บข้อมูล (Update) การลบข้อมูล (Delete) การเพิ่มข้อมูล (Add) เป็นต้น

3. ทำหน้าที่ป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะคอยตรวจสอบว่าคำสั่งใดที่สามารถทำงานได้ และคำสั่งใดที่ไม่สามารถทำงานได้

4. ทำหน้าที่รักษาความสัมพันธ์ของข้อมูลภายในฐานให้มีความถูกต้องอยู่เสมอ เช่นเมื่อมีการเปลี่ยนแปลงข้อมูลในที่หนึ่ง ก็จะไปเปลี่ยนแปลงข้อมูลในที่ที่มีความสัมพันธ์กัน หรืออ้างถึงกันอยู่ให้ถูกต้อง

5. ทำหน้าที่เก็บรายละเอียดต่างๆที่เกี่ยวข้องกับข้อมูลภายในฐานข้อมูลไว้ใน Data Dictionary ซึ่งรายละเอียดเหล่านี้จึงมักจะถูกเรียกว่า “ข้อมูลของข้อมูล” (Metadata)

6. ทำหน้าที่ควบคุมให้ฐานข้อมูลทำงานได้อย่างถูกต้อง และมีประสิทธิภาพ

3.4.7 โครงสร้าง และการจัดการไฟล์ข้อมูล Data Dictionary และ File Manager

ทุกฐานข้อมูลจะต้องมีส่วนที่ใช้เก็บข้อมูลในลักษณะ Metadata ซึ่งเป็นข้อมูลที่บอกถึงรายละเอียดของตัวข้อมูลที่เก็บอยู่ในฐานข้อมูล เช่น โครงสร้างของข้อมูล โครงสร้างของ Table โครงสร้างของ Index กฎที่ใช้ควบคุมความถูกต้องของข้อมูล (Integrity Rule) กฎที่ใช้ในการรักษาความปลอดภัยให้กับข้อมูล (Security Rule) ฯลฯ ข้อมูลเหล่านี้จัดเป็นข้อมูลที่มีความจำเป็นต่อโปรแกรม DBMS ในการตัดสินใจที่จะดำเนินการใด ๆ กับฐานข้อมูล เช่น ข้อมูลที่เกี่ยวข้องกับกฎที่ใช้ในการรักษาความปลอดภัยให้กับข้อมูล จะถูกนำมาใช้ในการพิจารณาให้สิทธิแก่ผู้ใช้ในการใช้งานฐานข้อมูล เป็นต้น สำหรับส่วนที่ใช้จัดเก็บข้อมูลในลักษณะของ Metadata นี้ ได้แก่ Data Dictionary หรือ Catalog

สำหรับ File Manager เป็นส่วนที่ทำหน้าที่บริหารและจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลระดับกายภาพ (Physical Level)

3.4.8 ประโยชน์ของฐานข้อมูล

การจัดนำข้อมูลที่มีความสัมพันธ์กันมาใช้ร่วมกันเป็นฐานข้อมูลนั้นจะก่อให้เกิดประโยชน์ดังนี้

1. สามารถลดความซ้ำซ้อนของข้อมูล (Data Redundancy) โดยไม่จำเป็นต้องจัดเก็บข้อมูลที่ซ้ำซ้อนกันไว้ในระบบแฟ้มข้อมูลของแต่ละหน่วยงานเหมือนเช่นเดิม แต่สามารถนำข้อมูลมาใช้ร่วมกันในคุณลักษณะ Integrated แทน

2. สามารถหลีกเลี่ยงความขัดแย้งของข้อมูล (Data Inconsistency) เนื่องจากไม่ต้องจัดเก็บข้อมูลที่ซ้ำซ้อนกันหลายแฟ้มข้อมูล ดังนั้นการแก้ไขข้อมูลในแต่ละชุดจะไม่ก่อให้เกิดค่าที่แตกต่างกันได้

3. แต่ละหน่วยงานในองค์กรสามารถใช้ข้อมูลร่วมกันได้

4. สามารถกำหนดให้ข้อมูลมีรูปแบบที่เป็นมาตรฐานเดียวกันได้ เพื่อให้ผู้ใช้ข้อมูลในฐานข้อมูลชุดเดียวกัน สามารถเข้าใจและสื่อสารถึงความหมายเดียวกัน

5. สามารถกำหนดระบบความปลอดภัยให้กับข้อมูลได้โดยกำหนดระดับความสามารถในการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนให้แตกต่างกันตามความรับผิดชอบ

6. สามารถรักษาความถูกต้องของข้อมูลได้โดยระบุกฎเกณฑ์ในการควบคุมความผิดพลาดที่อาจเกิดขึ้นจากการป้อนข้อมูลผิด

7. สามารถตอบสนองต่อความต้องการใช้ข้อมูลในหลายรูปแบบ

8. ทำให้ข้อมูลเป็นอิสระจากโปรแกรมที่ใช้งานข้อมูลนั้น (Data Independence) ซึ่งส่งผลให้ผู้พัฒนาโปรแกรมสามารถแก้ไขโครงสร้างของข้อมูลโดยไม่กระทบต่อโปรแกรมที่เรียกใช้งานข้อมูลนั้น เช่น ในกรณีที่ต้องการเปลี่ยนขนาดของ Field สำหรับระบบแฟ้มข้อมูลจะกระทำได้ยาก

เนื่องจากต้องเปลี่ยนแปลงตัวโปรแกรมที่อ้างถึง Field นั้นทั้งหมด ซึ่งต่างจากการใช้ระบบฐานข้อมูลที่การอ้างถึงข้อมูลจะไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูลจึงไม่ส่งผลให้ต้องแก้ไขโปรแกรมที่เรียกใช้ข้อมูลนั้นมากนัก

3.5 หลักการ RS232C

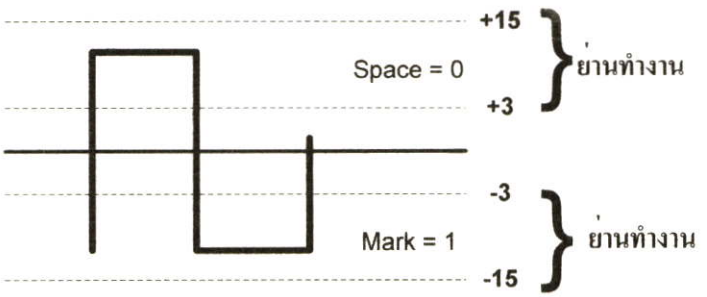
การสื่อสารข้อมูลแบบอะซิงโครนัสบน IBM/PC จะทำงานผ่านการ์ดอะแดปเตอร์ ซึ่งถูกจัดอยู่ในพื้นที่ของอุปกรณ์อินพุท/เอาท์พุท ตำแหน่ง 3F8H – 3FEH (COM1) และ 2F8 – 2FEH (COM2) โดยสามารถโปรแกรมโครงสร้างของการสื่อสารข้อมูลในลักษณะต่างๆ เช่น อัตราการรับ-ส่งข้อมูล จำนวนบิตหยุด การเลือกพาริตี้ เป็นต้น

การ์ดอะแดปเตอร์นี้จะมีไอซี 8250 เป็นส่วนสำคัญในการควบคุมทำงานการสื่อสารข้อมูล โดยมีคุณสมบัติเพิ่มเติม เช่น

- มีดับเบิ้ลบัฟเฟอร์สำหรับการส่งข้อมูล มีขาอินพุทเพื่อใช้เป็นสัญญาณนาฬิกาในการรับข้อมูลที่อิสระ
- มีฟังก์ชันการควบคุมโมเด็มเช่น CTS (Clear To Send), RTS (Request To Send), DSR (Data Set Ready), DTR (Data Terminal Ready), RI (Ring Indicator), และการตรวจสอบสัญญาณ Carrier
- การตรวจสอบการเริ่มต้นผิดพลาด
- การสร้างสัญญาณ Break และการตรวจสอบสัญญาณ Break

การสื่อสารข้อมูลสามารถสร้างสัญญาณเพื่อให้เกิดการอินเทอรัพท์ไปที่ระบบได้ โดยถ้าเป็นการทำงานของพอร์ท COM 1 จะเกิดอินเทอรัพท์ที่ IRQ4 และถ้าเป็นการทำงานของพอร์ท COM2 จะเกิดอินเทอรัพท์ที่ IRQ3 ถ้าหากมีการเพิ่มพอร์ทอนุกรมเข้ามาอีก ระบบปฏิบัติการ Windows จะสร้างตำแหน่งอินเทอรัพท์ให้โดยอัตโนมัติ การเขียนโปรมแกรมรับข้อมูลอนุกรมจะต้องตรวจจับสัญญาณนี้แล้วเขียนโปรมแกรมรับข้อมูลเข้ามาประมวลผลต่อไป

3.5.1 สัญญาณอนุกรม RS232C



รูปที่ 3.5 ระดับสัญญาณไฟฟ้าในการสื่อสารข้อมูลบน IBM PC

การสื่อสารข้อมูลบนการ์ดนี้จะใช้มาตรฐาน RS232C ซึ่งจะแสดงความหมาย และระดับแรงดัน ตามตารางที่ 3.1 สัญญาณไฟฟ้าจากรูปที่ 3.5 เป็นสัญญาณในสายรับ (Rx) และส่งข้อมูล (Tx) เทียบกับสายสัญญาณกราวด์ มีย่านการทำงานอยู่ในช่วง -3 V. ถึง -15 V. และ +3 V. ถึง +15 V.

ตารางที่ 3.1 ระดับสัญญาณแรงดันอนุกรมมาตรฐาน RS 232C

ลอจิก	ระดับแรงดัน	สภาวะสัญญาณ
0	+3 V. ถึง +15 V.	Space
1	-3 V. ถึง –15 V.	Mark

ข้อมูลที่มีระดับเป็นลอจิก 0 จะเรียกว่าสภาวะ “Spacing” ซึ่งแรงดันที่ส่งออกไปภายนอกจะเป็นแรงดันบวกตั้งแต่ +3 V. ถึง +15 V. ส่วนข้อมูลระดับเป็นลอจิก 1 จะเรียกว่าสภาวะ “Marking” ซึ่งแรงดัน ที่ส่งออกไปภายนอกจะเป็นแรงดันลบตั้งแต่ –3 V. ถึง –15 V. ถ้าระดับแรงดันอยู่ในช่วงมากกว่า –3 V. จนถึงน้อยกว่า +3 V. และมากกว่า +15 V. และน้อยกว่า –15 V. จะเป็นสภาวะที่ไม่สามารถบอกได้ว่าข้อมูลที่ได้รับเป็นลอจิกใด

เป็นการสื่อสารข้อมูลสภาวะ “Marking” ลอจิก 1 จะถูกใช้เป็นสภาวะ OFF (ขณะเริ่มต้นที่ไม่มีการส่งข้อมูล) จนกว่าจะมีสภาวะ “Spacing” ลอจิก 0 ซึ่งถือว่าเป็นบิตเริ่มต้น

ขาสัญญาณและรายละเอียดขาสัญญาณต่างของ Connector

Connector ที่ใช้สำหรับพอร์ทอนุกรมจะมี 2 ลักษณะ คือแบบ 9 ขา (DB9) และแบบ 25 ขา (DB25) ซึ่งทั้ง 2 แบบมีรายละเอียดดังนี้

ตารางที่ 3.2 รายละเอียดของ Connector DB9

IBM 9-Pin Serial connector		
Pin	**Function**	**Mnemonic**
1	Carrier defect	CD
2	Receive data	RXD
3	Transmit data	TXD
4	Data terminal ready	DTR
5	Signal Ground	GND
6	Data set ready	DSR
7	Request to send	RTS
8	Clear to send	CTS
9	Ring indicator	RI

3.5.2 รีจีสเตอร์ต่าง ๆ สำหรับควบคุมพอร์ทอนุกรม

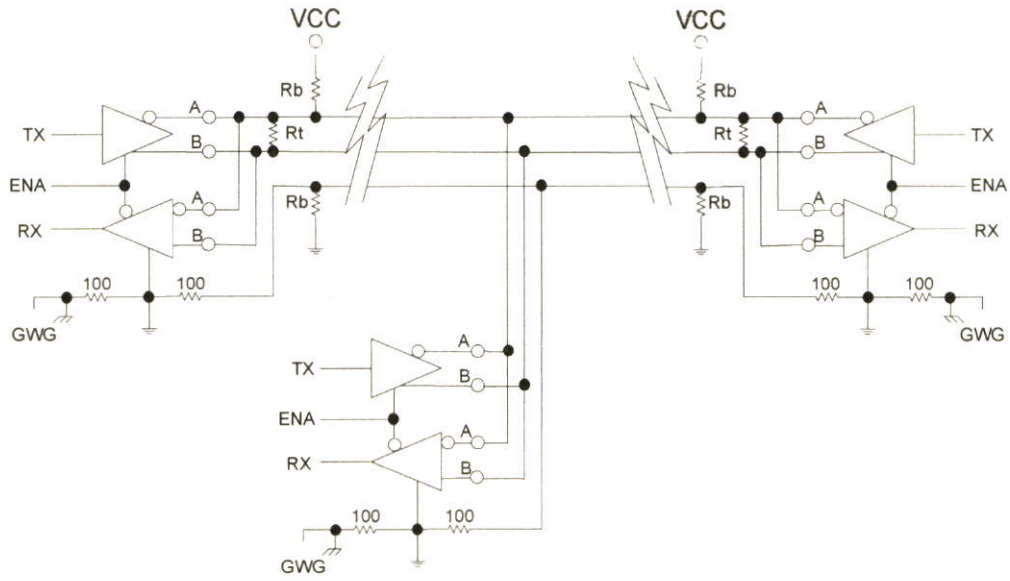
ในการโปรแกรมเพื่อสื่อสารข้อมูลอนุกรม นั้นจะต้องพิจารณาโครงสร้างของรีจีสเตอร์ต่างๆ ที่ ทำหน้าที่ควบคุมพอร์ท ซึ่งรีจีสเตอร์เหล่านี้จะมีตำแหน่งที่แน่นอนบน IBM PC ซึ่งทำหน้าที่กำหนดโครงสร้างพื้นฐานต่างๆ เช่น จำนวนบิตข้อมูล จำนวนบิตหยุด อัตราบอด และการควบคุมโมเด็ม เป็นต้น โดยมีตำแหน่งดังตารางที่ 3.3 ตำแหน่งรีจีสเตอร์เหล่านี้จะใช้อ้างถึงในโปรแกรมเพื่อใช้ควบคุมฮาร์ดแวร์ต่อไป ในกรณีที่รู้ตำแหน่งพอร์ทที่แน่นอน เช่น COM1 หรือ COM2 ก็สามารถอ้างถึงตำแหน่งรีจีสเตอร์โดยตรงได้เลย แต่ถ้าหากเป็นพอร์ทที่เพิ่มเข้ามาในภายหลังเช่น พอร์ท USB จะไม่สามารถรู้ตำแหน่งที่แน่นอนของพอร์ทได้ขึ้นอยู่กับว่าตอนนั้นมีพอร์ทไหนว่างอยู่ ในกรณีนี้จะต้องมีการเขียนโปรแกรมให้มีการตรวจเช็คพอร์ท และให้ผู้ใช้งานสามารถเลือกพอร์ทที่มีอยู่ได้อย่างอิสระ

ตารางที่ 3.3 ตำแหน่งของรีจีสเตอร์พอร์ทอนุกรมบน IBM PC

ตำแหน่ง I/O		รีจีสเตอร์ของ 8250	สภาวะ
COM1	COM2		DLAB
3F8	2F8	Tx Buffer	DLAB = 0 Write
3F8	2F8	Rx Buffer	DLAB = 0 Read
3F8	2F8	Divisor Latch LSB	DLAB = 1
3F9	2F9	Divisor Latch MSB	DLAB = 1
3F9	2F9	Interrupt Enable Reg.	
3FA	2FA	Interrupt Identification Reg.	
3FB	2FB	Line Control Reg.	
3FC	2FC	Modem Control Reg.	
3FD	2FD	Line Status Reg.	
3FE	2FE	Modem Status Reg.	

3.6 หลักการ RS485

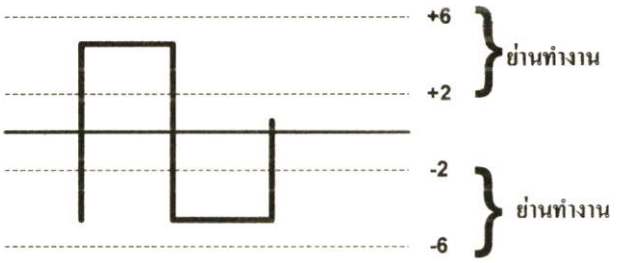
ถ้าเอ่ยถึงการสื่อสารอนุกรมส่วนใหญ่จะนึกถึงมาตรฐาน RS232C เพราะเป็นมาตรฐานที่ใช้กันอย่างแพร่หลาย และมีพอร์ท RS232C ติดอยู่กับคอมพิวเตอร์เกือบทุกเครื่อง รวมไปถึงอุปกรณ์ต่อพ่วงอื่นๆอีกมากมาย ถึงแม้ว่าRS232C จะใช้งานได้สะดวกแต่ก็มีข้อเสียตรงที่ไม่สามารถใช้สื่อสารระยะไกล และไม่สามารถต่ออุปกรณ์หลายๆตัวบนบัสเดียวกันได้ เนื่องจากสายสัญญาณต้องเทียบศักย์ไฟฟ้ากับสายกราวด์ ทำให้เกิดสัญญาณรบกวนชนิดคอมมอนโหมดขึ้น การใช้งานบางชนิดต้องการสื่อสารข้อมูลที่มีระยะทางไกลออกไป หรือต้องการสื่อสารกับอุปกรณ์หลายตัวบนบัสเดียวกัน มาตรฐาน RS485 และ RS422 เป็นทางเลือกที่ดีเพราะสามารถส่งข้อมูลได้ไกลสูงสุด 1.2 กิโลเมตร และต่ออุปกรณ์ร่วมบนบัสเดียวกันได้สูงสุด 32 ตัว ด้วยไอซีมาตรฐาน เช่น SN75176 หรือสูงสุด 255 ตัว ด้วยไอซีพิเศษ



รูปที่ 3.6 ระบบสื่อสารอนุกรม RS485

การที่ RS485 สามารถส่งข้อมูลออกไปได้ไกลก็เพราะว่าใช้หลักการเปรียบเทียบความต่างศักย์ไฟฟ้าระหว่างสายสัญญาณ 2 เส้น (A กับ B) ไม่ได้เปรียบเทียบกับกราวด์เหมือน RS232C จึงขจัดปัญหาเรื่องสัญญาณกราวด์ระหว่างตัวส่งกับตัวรับไม่เท่ากันเนื่องจากแรงดันตกคร่อมในสายส่งข้อมูลจะมีสถานะเป็นลอจิกสูง หรือ "1" เมื่อสาย A มีความต่างศักย์มากกว่าสาย B เป็นปริมาณ 200 mV. และข้อมูลเป็นลอจิกต่ำ หรือ "0" เมื่อสาย B มีความต่างศักย์มากกว่าสาย A เป็นปริมาณ 200 mV. ดังรูปที่ 3.7

จากรูปที่ 3.6 แสดงการต่ออุปกรณ์สื่อสารบนบัสอนุกรม RS485 ขั้วสัญญาณ A กับ B เป็นสายข้อมูล การสื่อสารที่ดีจำเป็นต้องมีการปรับค่าอิมพีแดนซ์ (Impedance) ของสายให้เหมาะสม เพื่อป้องกันข้อมูลสูญหายอันเนื่องมาจากระดับสัญญาณไม่ถูกต้องโดยใส่ตัวต้านทานTerminator (R_t)ขนานกับขั้ว A B ค่า R_t จะต้องมีค่ามากกว่า 90 Ω ขึ้นไปโดยทั่วไปใช้ 120 Ω ใส่ไว้ที่ 2 จุดคือจุดที่อยู่ต้นบัส กับจุดปลายบัส ส่วนจุดต่ออื่นๆไม่ต้องใส่เพราะจะทำให้ค่าอิมพีแดนซ์ของสายต่ำเกิน ในกรณีที่ตัว Converter ใช้แหล่งจ่ายจากคอมพิวเตอร์ไม่ควรต่อ R_t ใกล้กับตัว Converter เพราะจะใช้กระแสมากเกินทำให้แรงดันตกลง ถ้าจำเป็นต้องใส่ให้ใส่ตัวเก็บประจุอนุกรมกับ R_t เพื่อลดการไหลของกระแสลง ในสภาวะไม่มีการส่งสัญญาณจะไม่สามารถรู้ได้ว่าขณะนี้สายสัญญาณมีสถานะลอจิกอะไรจึงจำเป็นต้องใส่ตัวต้านทาน Bias (R_b) พูลอัพขั้ว A และพูลดาวน์ ขั้ว B ไว้เพื่อให้ขณะที่ไม่มีการส่งสัญญาณ สายส่งมีสถานะลอจิก 1 เสมอ ในกรณีที่มีการต่ออุปกรณ์หลายตัวบนบัสต้องคำนวณค่าอิมพีแดนซ์ของสายส่งและชดเชยให้มีกระแสไหลในวงจรไม่ต่ำกว่า 3.5 mA.



รูปที่ 3.7 ระดับสัญญาณไฟฟ้าระหว่างขั้ว A กับ B ของ RS485

3.6.1 การคำนวณเพื่อชดเชยค่าอิมพีแดนซ์ในระบบ RS485

ก่อนที่จะติดตั้งระบบ RS485 จะต้องมีการคำนวณเพื่อชดเชยค่าอิมพีแดนซ์ (Impedance) เสียก่อนเพื่อให้การสื่อสารข้อมูลในระบบเป็นไปอย่างถูกต้อง ถ้าหากไม่มีการชดเชยแล้วระดับสัญญาณในสายข้อมูลจะไม่ได้มาตรฐานตามข้อกำหนด ส่งผลให้ข้อมูลที่ได้รับผิดไปจากความเป็นจริง แสดงตัวอย่างการคำนวณดังต่อไปนี้

บนบัสมีอุปกรณ์สื่อสารจำนวน 10 ชุดแต่ละชุดใช้ IC 75176 ซึ่งมีค่าอิมพีแดนซ์ภายใน12 kΩ เป็นตัวขับRS485

ที่ระดับแรงดัน 5 V. ค่าความต้านทานสูงสุดของระบบหาจาก

$$R_{max} = \frac{5\,V.}{3.5mA.}$$

$$R_{max} = 1428\,\Omega$$

ค่าความต้านทานรวมในระบบหาจากการขนานกันของความต้านทานภายใน

$$R_{in} = 12k.//12k.//12k.//12k.//12k.//12k.//12k.//12k.//12k.//12k.$$

$$R_{in} = 1.2k.$$

ถ้ามีการใส่ความต้านทาน Terminator 2 ตัว ตัวละ 120 Ω ที่หัวและท้ายบัส

$$R_t = 120//120$$

$$R_t = 60\ \Omega$$

รวมค่าความต้านทานทั้งระบบ

$$R_{Total} = 1200//60$$

$$R_{Total} = 57\ \Omega$$

หาค่า R_b

$$R_b = (\frac{R_{max} - R_t}{2})$$

$$R_b = (\frac{1428 - 57}{2})$$

$$R_b = 685\,\Omega$$

3.6.2 การคำนวณเพื่อหาค่าความเร็วในการรับ-ส่ง สูงสุด

การออกแบบระบบสื่อสารจำเป็นต้องคำนึงถึง สภาวะแวดล้อม ชนิดสายส่ง ความยาวสาย และอัตราเร็วในการรับส่ง จากกฎที่ว่าเวลาหน่วง (Delay Time) จะต้องน้อยกว่าเวลาที่ให้ในการรับ-ส่ง ต่อ 1 บิต ค่าเวลาหน่วงหาได้จากเวลาที่ใช้ในการรับหรือส่ง 1 รอบ ตามสมการ

$$t = r(\frac{L}{KC}) \qquad (4.1)$$

เมื่อ

r คือ จำนวนรอบการวิ่งของข้อมูล (ปกติใช้ r = 3 เท่าของความยาวสายส่ง)

L คือ ความยาวสานส่ง (เมตร)

K คือ ค่าคงที่ของสายส่ง (66%-75%)

C คือ ความเร็วแสง (3×10^8 m/s)

ตัวอย่าง การคำนวณเพื่อหาค่าความเร็วสูงสุดที่สามารถใช้ได้

สายส่งยาว 4000 ฟุต ค่าคงที่ของสายส่ง = 66%

$$t = 3 \times \frac{(4000 / 3)}{0.66 \times 3 \times 10^{8}}$$

$$t = 20\,uS.$$

ถ้าใช้อัตรารับส่ง (Baud Rate) = 9600 bps ต้องใช้เวลา

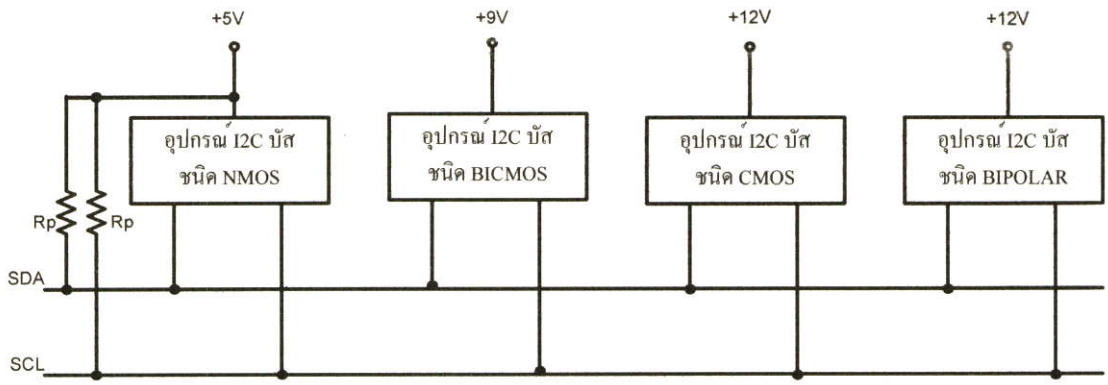
$$\frac{1}{9600} = 104\,uS.$$

เวลาที่ใช้ 104 uS. มากกว่า 20 uS. แสดงว่าสามารถใช้ความเร็ว 9600 b/s ที่สายส่ง 4000 ฟุต ได้

ข้อแตกต่างระหว่าง RS485 และ RS422 ก็คือ RS485 เป็นชนิดสื่อสารทางเดียว (Half Duplex) ส่วน RS422 สามารถสื่อสารได้ 2 ทิศทางในเวลาเดียวกัน (Full Duplex) ทั้งสองชนิดมีระยะทางในการสื่อสารสูงสุด 1.2 กิโลเมตรเท่ากัน แต่ RS485 ใช้สายสัญญาณเพียง 2 เส้นในขณะที่ RS422 ต้องใช้สายสัญญาณถึง 4 เส้น เพราะแยกกันระหว่างส่งกับรับ

3.7 หลักการ I^2C

I^2C ย่อมาจาก Inter-IC Communication หมายถึงการติดต่อสื่อสารระหว่างไอซีโดยบัส I^2C ได้รับการพัฒนาขึ้นจากบริษัทฟิลลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีสามารถสื่อสารกันได้ด้วยสายสัญญาณเพียง 2 เส้น สายเส้นหนึ่งเป็นสายข้อมูล อีกเส้นหนึ่งเป็นสัญญาณนาฬิกา กำหนดจังหวะการทำงาน บนบัส I^2C สามารถมีอุปกรณ์ได้มากกว่า 2 ตัว สามารถอ้างถึงกันได้ด้วยแอดเดรส หรือหมายเลขตำแหน่งของอุปกรณ์ สายข้อมูลบนบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock Line)



รูปที่ 3.8 การต่ออุปกรณ์บนบัสข้อมูล I^2C

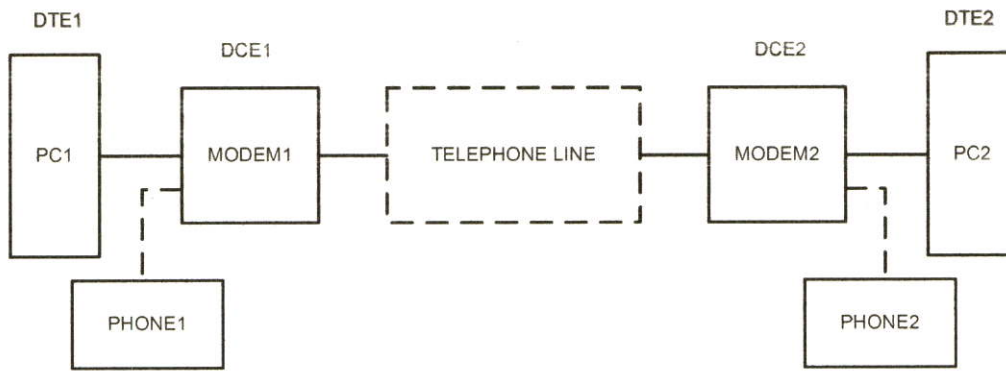
สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (Bi-Directional Line) ต้องมีการต่อตัวต้านทานพูลอัปกับแรงดัน +5 V. ไว้ตลอดเวลาเพื่อให้สายสัญญาณมีสภาวะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยป้องกันสัญญาณรบกวนที่อาจเข้ามาในสายสัญญาณทั้งสอง วงจรเอาท์พุทของอุปกรณ์ที่ต่ออยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (Open-Drain) หรือคอลเล็คเตอร์เปิด (Open-Collector) อัตราการถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (Standard Mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (Fast Mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้าที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF. การเข้าถึงอุปกรณ์บนบัส I^2C ใช้แอดเดรสขนาด 7 บิต หรือ 10 บิต

ข้อดีอีกประการหนึ่งของบัส I^2C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ เช่น อุปกรณ์บนบัส I^2C ตัวหนึ่งใช้ไฟเลี้ยง 5 V. ส่วนอีกตัวใช้ไฟเลี้ยง 12 V. การต่อร่วมกันบนบัส I^2C ต้องต่อสายสัญญาณ SDA และ SCL เข้าด้วยกัน แต่ต้องต่อความต้านทานพูลอัปไว้ที่แรงดัน +5 V. เสมอ

การโปรแกรมการอ่าน-เขียนข้อมูลจะมีรูปแบบเฉพาะ ซึ่งจะต้องเป็นไปตามข้อกำหนดมาตรฐานของ I^2C ชุดข้อมูลประกอบไปด้วยส่วนของข้อมูลดังนี้ Start, Address, Data, Ackknow และ Stop ภายในชุดข้อมูลจะมีการบอกจุดเริ่มต้นเพื่อเริ่มการติดต่อ ตามด้วยตำแหน่งของอุปกรณ์ที่ต้องการติดต่อด้วย แล้วก็ข้อมูลที่ต้องการส่ง เมื่อมีการส่งข้อมูลเรียบร้อยแล้วจะต้องตรวจสอบว่าผู้รับได้รับข้อมูลเรียบร้อยหรือไม่โดยการตรวจสอบสัญญาณ Ackknow จากผู้รับ แล้วตามด้วยคำสั่งหยุด ในกรณีที่ไม่มีสัญญาณ Ackknow จากผู้รับจะถือว่าผู้รับไม่ได้รับข้อมูลนั้น อาจมีการส่งข้อมูลใหม่ซ้ำก็ได้

3.8 หลักการโมเด็ม

โมเด็มเป็นอุปกรณ์แปลงข้อมูลดิจิตอลเป็นข้อมูลอนาล็อคเพื่อส่งข้อมูลผ่านสายส่ง โดยทั่วไปใช้สายโทรศัพท์เป็นตัวกลางในการสื่อสาร ในที่นี้ขอกล่าวถึงโมเด็มที่มีขั้วต่ออนุกรม RS232C และมีโปรแกรมควบคุมตั้งแต่ V9.0 ขึ้นไปเท่านั้นเพราะง่ายต่อการควบคุม ราคาถูก หาซื้อง่าย และใช้กันอย่างแพร่หลาย หลักการของโมเด็มชนิดนี้คือแปลงสัญญาณดิจิตอลให้เป็นสัญญาณ Phase Shift Keying ที่มีการตรวจสอบขนาดสัญญาณด้วยเพื่อเพิ่มความเร็วในการสื่อสาร



รูปที่ 3.9 ไดอะแกรมการส่งข้อมูลผ่านระบบโทรศัพท์

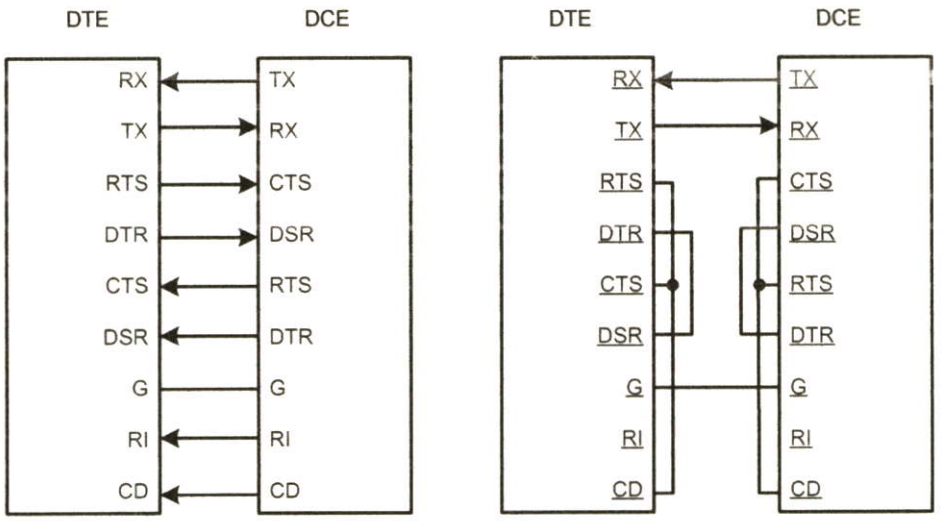
จากรูปที่ 3.9 แสดงการสื่อสารผ่านระบบโทรศัพท์ระหว่างคอมพิวเตอร์ 2 ตัวผ่านโมเด็ม คอมพิวเตอร์ทั้ง สองฝ่ายจะต้องติดตั้งโมเด็มเพื่อแปลงสัญญาณดิจิตอลเป็นสัญญาณอะนาล็อคที่ถูกเข้ารหัสส่งผ่านระบบโทรศัพท์ซึ่งในระบบโทรศัพท์จะประกอบด้วย อุปกรณ์สลับคู่สาย และตัวสร้างสัญญาณพาห์ อาจใช้สัญญาณโทรศัพท์จากองค์การ หรือสัญญาณจากตู้ PABX ก็ได้

3.8.1 การควบคุมโมเด็ม

การควบคุมโมเด็มสามารถเลือกควบคุมได้ 2 ชนิดคือ

1. ควบคุมทาง Hardware (Hardware Flow Control) คือควบคุมด้วยขาช่วยของ DB9 เช่น (RTS, CTS, DTR, DSR)

2. ควบคุมด้วย AT Command (Software Flow Control) คือ ใช้คำสั่งควบคุม



(a) Hardware Flow Control (b) Software Flow Control

รูปที่ 3.10 (a-b) การต่อวงจรควบคุมโมเด็ม

ก่อนใช้งานโมเด็มจะต้องเลือกวิธีควบคุมเสียก่อน แล้วต่อวงจรควบคุมตามรูปที่ 3.10 การเลือกวิธีควบคุมสามารถทำได้โดยการป้อนคำสั่ง AT Command รหัสคำสั่งสามารถดูได้จากคู่มือการใช้งานของโมเด็ม ถ้าเลือก Hardware Flow Control การควบคุมการทำงานเช่น การรับสาย การวางสาย การโทรออก จะต้องใช้ขาสัญญาณช่วยเป็นตัวกำหนดให้โมเด็มทำงาน ถ้าหากเลือก Software Flow Control จะต้องส่งคำสั่งควบคุมมาควบคุมโมเด็ม

3.8.2 หน้าที่ของขาโมเด็มชนิด RS232C

RX = รับข้อมูล

TX = ส่งข้อมูล

RST = ใช้บอกอุปกรณ์ที่ต่ออยู่ว่าตัวมันพร้อมที่จะ รับ-ส่งข้อมูล ในโหมดการควบคุมด้วย Hardware ขานี้จะใช้บอกว่าหน่อยความจำสำรอง (Buffer Memories) เต็มให้หยุดการรับ-ส่งชั่วคราว

CTS = ต่อกับขา RTS เพื่อตรวจสอบสัญญาณเพื่อส่งต่อหรือหยุดส่งข้อมูล

DTR = ใช้แสดงว่าช่องสัญญาณว่างหรือไม่ เช่นสามารถติดต่อกับปลายทางได้หรือโทรติดแล้ว หรือใช้วางสายจากการติดต่อ โดยปกติถ้าตั้งให้โมเด็มรับสายอัตโนมัติจะต้องเซ็ตให้ขานี้เป็น "1" เพื่ออนุญาตให้รับสายอัตโนมัติได้

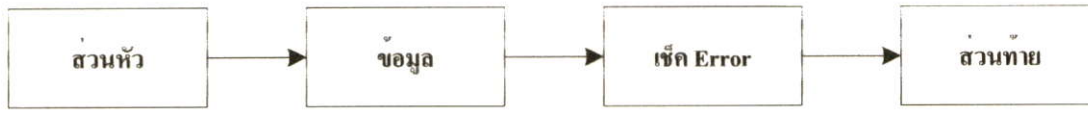
DSR = ใช้ตรวจสอบขา DTR ว่าขณะนี้โมเด็มติดต่อปลายทางอยู่หรือไม่

CD = ใช้แสดงความสมบูรณ์ของการต่อสายสำเร็จ และบอกว่าขณะนี้ได้รับสัญญาณพาหะ (Carrier Signal)

ก่อนการใช้งานโมเด็มจะต้องโปรแกรมให้โมเด็มทำงานในโหมดต่างๆตามที่เรากำหนดก่อน การโปรแกรมนี้จะใช้คำสั่ง AT Command ซึ่งแต่ละยี่ห้อก็จะมีคำสั่งที่แตกต่างกันไป แต่ยังคงคำสั่งมาตรฐานที่เหมือนกัน แสดงตัวอย่างคำสั่ง AT Command ในภาคผนวก ข.

3.9 การกำหนดโปรโตคอล (protocol)

การกำหนดข้อตกลงในการสื่อสารข้อมูลเป็นสิ่งสำคัญที่จะบอกว่าข้อมูลส่วนไหนหมายถึงอะไรเพื่อการสื่อสารที่มีประสิทธิภาพ รวมไปถึงการป้องกัน Error การกำหนดโปโตคอลสามารถกำหนดได้เองอย่างอิสระให้เหมาะสมกับระบบที่ใช้งานเพื่อความกระชับของข้อมูล ส่งผลให้การส่งข้อมูลมีความถูกต้อง และมีความเร็วสูงสุด แต่วิธีนี้โปรโตคอลที่กำหนดขึ้นจะต้องนำมาใช้กับทั้งตัวส่ง และตัวรับ คือ ก่อนส่งข้อมูลตัวส่งจะต้องนำข้อมูลไปเข้ารหัสตามโปโตคอลที่กำหนดขึ้น แล้วส่งข้อมูลนั้นไปยังตัวรับเมื่อตัวรับได้รับข้อมูลก็นำข้อมูลที่ได้รับมาถอดรหัสข้อมูลตามรหัสที่ตัวส่งสร้างขึ้นเพื่อถอดเอาข้อมูลที่จำเป็นมาใช้งาน ปัจจุบันมีโปโตคอลมาตรฐานที่ใช้งานทั่วไปเพื่อความสะดวกในการใช้งาน เช่น Modbus, Profibus ฯลฯ



รูปที่ 3.11 รูปแบบมาตรฐานของชุดข้อมูล

โดยทั่วไปแล้วการเข้ารหัสข้อมูลจะแบ่งข้อมูลออกเป็น 5 ส่วนดังรูปที่ 3.11 แต่ก็ขึ้นอยู่กับองค์ประกอบอื่นด้วย สามารถเพิ่ม หรือลดบางส่วนได้เช่น ถ้าเป็นการสื่อสารข้อมูลภายในบอร์ดเดียวกันโอกาสที่จะเกิดสัญญาณรบกวนจากภายนอกมีน้อยก็สามารถตัดส่วนเช็ค Error ออกได้เพื่อเพิ่มความเร็วในการส่งข้อมูล ในทางกลับกันถ้าข้อมูลถูกส่งออกไปภายนอกมีโอกาสที่จะเกิดสัญญาณรบกวนได้สูงอาจเพิ่มส่วนแก้ไขข้อมูลเข้าไปเพื่อให้ระบบสามารถแก้ไขข้อผิดพลาดของข้อมูลได้หากเกิดข้อผิดพลาดของข้อมูลขึ้น แต่ละส่วนของข้อมูลในรูปที่ 3.11 มีประโยชน์ดังนี้

1. ส่วนหัว (Head) คือส่วนแรกที่ต้องส่งส่วนใหญ่จะใช้เก็บข้อมูลเกี่ยวกับผู้รับ และคุณลักษณะของข้อมูล เช่น แอดเดรสของผู้รับ ความยาวข้อมูล ชนิดของข้อมูล ฯลฯ

2. ส่วนข้อมูล (Data) เป็นตัวข้อมูลที่ต้องการส่งให้ผู้รับ

3. เช็ค Error (Error Detection) เกิดจากการเข้ารหัสข้อมูลทั้งหมดด้วยการกระทำทางคณิตศาสตร์เฉพาะ ให้ด้านรับถอดรหัสแบบเดียวกันเพื่อตรวจสอบว่าข้อมูลที่รับมาถูกต้องหรือไม่ ใช้ตรวจสอบ error ซึ่งจะกล่าวในหัวข้อถัดไป

4. ส่วนปิดท้าย (End) ใช้บอกผู้รับให้รู้ว่าการส่งข้อมูลได้สิ้นสุดลงแล้ว หรือในบางกรณีก็ใช้เป็นตัวทำให้เกิด Interrupt ขึ้นทางด้านรับเมื่อชุดข้อมูลส่วนปิดท้ายเป็นไปตามรูปแบบที่กำหนดไว้

3.10 การป้องกันไม่ให้เกิด Error (Error Prevention)

การสื่อสารข้อมูล ความถูกต้องของข้อมูลเป็นหัวใจสำคัญ หากข้อมูลที่ได้มานั้นไม่มีความน่าเชื่อถือหรือความน่าเชื่อถือต่ำจะส่งผลให้การควบคุมผิดพลาดตามไปด้วย ในบางครั้งเราไม่สามารถควบคุมไม่ให้เกิดความผิดพลาดได้ เช่น ขณะสื่อสารข้อมูล เกิดไฟฟ้าดับ หรือ เกิดสัญญาณรบกวนจากภายนอก ทำให้ได้รับข้อมูลที่ผิดไปจากความเป็นจริง ปกติแล้วจะต้องเกิดความผิดพลาดอยู่เสมอ ดังนั้นต้องมีกระบวนการที่ช่วยตรวจสอบข้อมูลที่ได้รับมาว่าถูกต้องหรือไม่ก่อนที่จะนำข้อมูลนั้นไปใช้งาน

แหล่งกำเนิด Error สามารถจำแนกได้ดังนี้

- White Noise หรือ Thermal Noise คือสัญญาณรบกวน เกิดจากอุณหภูมิซึ่งมีผลต่อการเคลื่อนที่ของอิเล็คตรอน

- Impulse Noise เป็นสัญญาณที่มีลักษณะยอดแหลม (Spikes) ในช่วงเวลาสั้นๆ เกิดจากฟ้าผ่า, รอยต่อหลวม หรือการกระชากของกระแสอย่างรุนแรง

- Cross Talk เกิดจากการรบกวนข้ามกันของสัญญาณในสายใกล้เคียงกันเหนี่ยวนำเข้ามาเป็นสัดส่วนโดยตรงกับระยะทาง, Bandwidth, ความแรงของสัญญาณ, ความใกล้ชิดกับช่องส่งสัญญาณอื่น และจะเป็นสัดส่วนผกผันกับการ Shield และการแยกความห่างระหว่างช่องสัญญาณ

- สัญญาณสะท้อน (Echo) คือการสะท้อนหรือการกลับทิศทางของสัญญาณที่ส่งออกไป มักเกิดตรงส่วนปลายสุดของสัญญาณ สามารถแก้ไขได้โดยติดตั้ง Terminator ตรงส่วนปลายสุดของบัสสัญญาณ

- การสูญเสียพลังงาน (Attenuation) คือการที่สัญญาณอ่อนแรงลงเนื่องจากระยะทาง และคุณสมบัติของสายส่ง แก้ไขโดยใช้ขนาดสายส่งให้เหมาะสมกับระยะทางและความเร็วในการส่งข้อมูล ถ้าต้องการส่งให้ไกลออกไปก็สามารถใช้อุปกรณ์ทวนสัญญาณ (Repeater) เข้าช่วย

การกำจัดสัญญาณรบกวนให้หมดไปจากระบบนั้นเป็นไปได้ยาก ดังนั้นการป้องกันไม่ให้เกิด Error นั้นจะต้องใช้เทคนิคของการป้องกัน Error เข้าช่วยลดความเสียหายของข้อมูลมีเทคนิคดังนี้

- การลดความเร็วในการส่งข้อมูล
- เพิ่มการ Shield ให้สายส่ง
- ติดตั้งอุปกรณ์เพิ่มระยะทาง (Line Driver) หรือตัวทวนสัญญาณ (Repeater)

- ใช้อุปกรณ์รับ-ส่งที่มีประสิทธิภาพสูง

3.10.1 การตรวจจับ Error (Error Detection)

ถึงแม้ว่าจะมีการป้องกันการเกิด Error เท่าใดก็ตามก็ยังมีโอกาสเกิด Error ได้เนื่องจากสัญญาณรบกวนชั่วขณะที่ไม่สามารถคาดเดาได้ หรือจากสภาพแวดล้อมที่เปลี่ยนแปลงไปจากเดิม จึงต้องใช้เทคนิคการตรวจจับ Error เข้าช่วยดักจับ Error เพื่อการแก้ไขต่อไป มีหลายวิธีดังนี้

3.10.2 วิธีตรวจสอบพาริตี้ (Parity Check)

วิธีนี้เป็นที่นิยมมากที่สุดเพราะง่ายต่อการสร้างสัญญาณส่ง และการตรวจจับ วิธีการคือสร้างบิตสุดท้ายเพิ่มขึ้นอีก 1 บิตเพื่อเอาไว้เช็คว่าผลรวมของข้อมูลที่มีค่าเป็น 1 ที่ส่งมาก่อนพาริตี้บิตนี้มีค่าเป็นเลขคู่ (Even)หรือเลขคี่ (Odd) ตรงนี้สามารถเลือกกำหนดได้ว่าจะให้เป็นเลขคู่หรือเลขคี่ เช่น กำหนดเป็นเลขคี่ถ้าผลรวมข้อมูลบิตที่เป็น 1 เป็นเลขคู่แล้ว พาริตี้จะเป็น 1 เพราะจะต้องทำให้ผลรวมบิตที่เป็น 1 ทั้งหมด (ข้อมูล+พาริตี้) เป็นเลขคี่ ในกรณี ตั้งให้เป็นเลขคู่ก็เหมือนกันจะต้องทำให้ให้ผลรวมบิตที่เป็น 1 ทั้งหมด (ข้อมูล+พาริตี้) เป็นเลขคู่ ทั้งตัวรับและตัวส่งจะต้องตั้งค่าพาริตี้เหมือนกัน เมื่อตัวรับได้รับข้อมูลก็จะต้องเช็คว่าพาริตี้บิตถูกต้องตามข้อกำหนดหรือเปล่า ถ้าถูกต้องก็ดำเนินการจัดเก็บข้อมูล ถ้าหากผิดก็ให้ทำการแก้ไขต่อไป การตรวจสอบพาริตี้นี้สามารถนำไปใช้ร่วมกับการตรวจสอบวิธีอื่นได้เพื่อเพิ่มความถูกต้อง

3.10.3 การตรวจจับ Error แบบ Longitudinal Redundancy Check (LRC)

วิธีนี้คล้ายกับการสร้างพาริตี้แต่จะใช้กับการส่งข้อมูลที่เป็นบล็อคหรือเป็นกลุ่มของ ไบต์ข้อมูลโดยจะสร้างรหัสตรวจจับไว้ท้ายบล็อคข้อมูลเรียกว่า Block Check Character (BCC) รหัสที่ว่าเกิดจากการสร้างพาริตี้จากข้อมูลทั้งหมดชนิดแยกบิต คือมีทั้งพาริตี้คู่ และคี่ แต่พาริตี้จะสร้างจากบิตที่มีหลักตรงกันเท่านั้น เช่น บิตที่ 1 ของ BCC เกิดจากการนำค่าที่เป็น 1 ของบิตแรกของข้อมูลทั้งหมดมาคิดพาริตี้ ส่วนบิตถัดไปก็ใช้วิธีเดียวกันคือ คิดแบบบิตต่อบิต วิธีนี้สามารถใช้ร่วมกับการตรวจจับพาริตี้ได้

3.10.4 การสร้างรหัสตรวจสอบ Cyclic Redundancy Check (CRC)

สร้างจากการนำข้อมูลทั้งบล็อคมาเข้ารหัสทางคณิตศาสตร์เพื่อสร้าง Character ตรวจสอบส่งตามไป ส่วนผู้รับก็นำข้อมูลที่รับมาทั้งหมดมาเข้ารหัสเดียวกับตัวส่งแล้วนำค่าที่ได้มาเปรียบเทียบกับรหัสตรวจสอบที่ตัวส่ง ส่งมาหรือไม่ถ้าเหมือนกันแสดงว่าข้อมูลที่ได้รับมาถูกต้อง การสร้าง CRC นั้นเกิดจากการสร้าง Polynomial ขึ้นมาการกำหนด Degree ของ Polynomial นั้นขึ้นอยู่กับจำนวนของข้อมูลที่ต้องการส่งในแต่ละบล็อค ปัจจุบันมีมาตรฐานดังนี้ CRC-12, CRC-16, CRC-32 และ CRC-CCITT

การส่งข้อมูลประมาณมากจำเป็นต้องแบ่งข้อมูลออกเป็นส่วนย่อยๆ ส่งออกเป็นชุดๆ (Packet) ชุดข้อมูลเหล่านี้อาจถูกส่งไปตามเส้นทางที่ต่างกันเมื่อถึงปลายทางแล้วผู้รับจะนำชุดข้อมูลเหล่านั้นมารวมกันกลับคืนเป็นข้อมูลเดิม การที่ผู้รับสามารถเรียงข้อมูลกลับคืนได้ถูกต้องนั้นภายในข้อมูลแต่ละชุดต้องมีตัวบอกลำดับข้อมูลติดไปด้วย เพื่อตรวจการเรียงลำดับ (Sequence Check)

รหัสแก้ไข Error (Error Correction Codes) เป็นวิธีที่นอกจากใช้ตรวจสอบ Error แล้วยังใช้แก้ไข Error ได้อีกด้วยรหัสแก้ไขนี้เรียกว่า (Forward Error Correcting) จะเพิ่มบิตเพื่อใช้แก้ไขเข้ามาเหมาะสำหรับแก้ไข Error ที่ไม่เกิน 1 บิต เช่นการแก้ไขข้อมูลในหน่วยความจำ หากมีข้อมูล Error มากกว่า 1 บิต จะต้องเพิ่มบิตแก้ไขทำให้เกิดความรุงรังเนื่องจากประสิทธิภาพจะลดลง

เมื่อเกิด Error ขึ้นจะต้องมีการแก้ไขถ้าเราใช้รหัสแก้ไข (Error Correction Code) ผู้รับก็สามารถแก้ไข Error นั้นได้ด้วยตัวเองแต่เนื่องจากมีข้อจำกัดหากจำนวนบิตที่ Error มีมากกว่า 1 บิต ก็ต้องเพิ่มบิตแก้ไขหรืออาจแก้ไขไม่ได้เลย วิธีที่ดีที่สุดก็คือการขอข้อมูลนั้นซ้ำอีกครั้ง ขั้นตอนการแก้ไขแบบขอข้อมูลใหม่คือ ทุกครั้งที่ได้รับข้อมูลครบ 1 packet ผู้รับจะต้องส่งข้อความตอบรับ (Message Acknowledgment) กลับมายังผู้ส่งเพื่อบอกว่าได้รับข้อมูลชุดนั้นสมบูรณ์เรียบร้อยแล้ว (Positive Acknowledgment) แต่ถ้าผู้รับเช็คข้อมูลที่รับมาแล้วเกิด Error ก็จะส่งสัญญาณขอทวนข้อมูลนั้นซ้ำ (Negative Acknowledgment) การขอข้อมูลซ้ำจะมีจำนวนครั้งที่จำกัด (Retry Limit) โดยปกติแล้วจะอยู่ในช่วง 3-100 ครั้งเพื่อป้องกันการวนขอข้อมูลไม่มีที่สิ้นสุดหากการติดต่อสื่อสารนั้นล้มเหลว เมื่อพยายามขอซ้ำจนครบจำนวนที่กำหนดแล้ว ก็จะเลิกการติดต่อพร้อมทิ้งฟ้องความผิดปกติในการติดต่อสื่อสาร และรอจนกว่าจะมีการแก้ไขจึงสามารถเริ่มการสื่อสารอีกครั้ง

บทที่ 4
การออกแบบและสร้างระบบควบคุมความต้องการพลังไฟฟ้า

4.1 บทนำ

การออกแบบระบบจะแยกออกแบบเป็นส่วนๆ ของโมดูลย่อยแต่ละโมดูลสามารถรับคำสั่งการทำงานได้ แล้วนำโมดูลมาประกอบกันเป็นระบบตามความเหมาะสมของระบบที่ต้องการควบคุม ดังรูปที่ 4.1 จะเห็นว่าการควบคุมเริ่มจาก ตัวควบคุม อ่านค่า ความต้องการพลังไฟฟ้าจากมิเตอร์แต่ละตัวมารวมกันแล้วนำมาเปรียบเทียบกับเวลา และค่าความต้องการพลังไฟฟ้าที่จะต้องควบคุมถ้าหากอยู่ในช่วงเวลาที่ต้องการควบคุม และ มีค่าความต้องการพลังไฟฟ้าต่ำ หรือ สูงเกินกว่า ค่าที่กำหนดไว้ จะทำการประมวลผลหา รีเลย์ เพื่อทำการ ตัด-ต่อ ตามโปรแกรมที่ตั้งไว้ ถ้าหากต้องการควบคุมเพียงอย่างเดียว มีเพียง 3 ส่วนก็เพียงพอ แต่ถ้าต้องการบันทึก, วิเคราะห์ และแสดงผลข้อมูลก็สามารถเพิ่ม ตัวตรวจเช็คบันทึกสถานะกับคอมพิวเตอร์เข้ามาเพื่อเก็บข้อมูล ระบบยังรองรับการสื่อสารผ่านสารโทรศัพท์ ด้วยการใช้โมเด็มเป็นตัวส่งผ่านข้อมูลในกรณีต้องการเข้าถึงข้อมูลจากระยะไกล การสื่อสารข้อมูลในแต่ละส่วนจะมีรูปแบบ (Protocol) เฉพาะประกอบไปด้วย 1.หมายเลขตำแหน่ง 2.จำนวนความยาวข้อมูล 3 ตัวข้อมูล 4.เช็คข้อมูลผิดพลาด

4.2 ความสามารถของระบบ

เครื่องต้นแบบเพื่อนำไปใช้งานจริงมีความสามารถดังต่อไปนี้

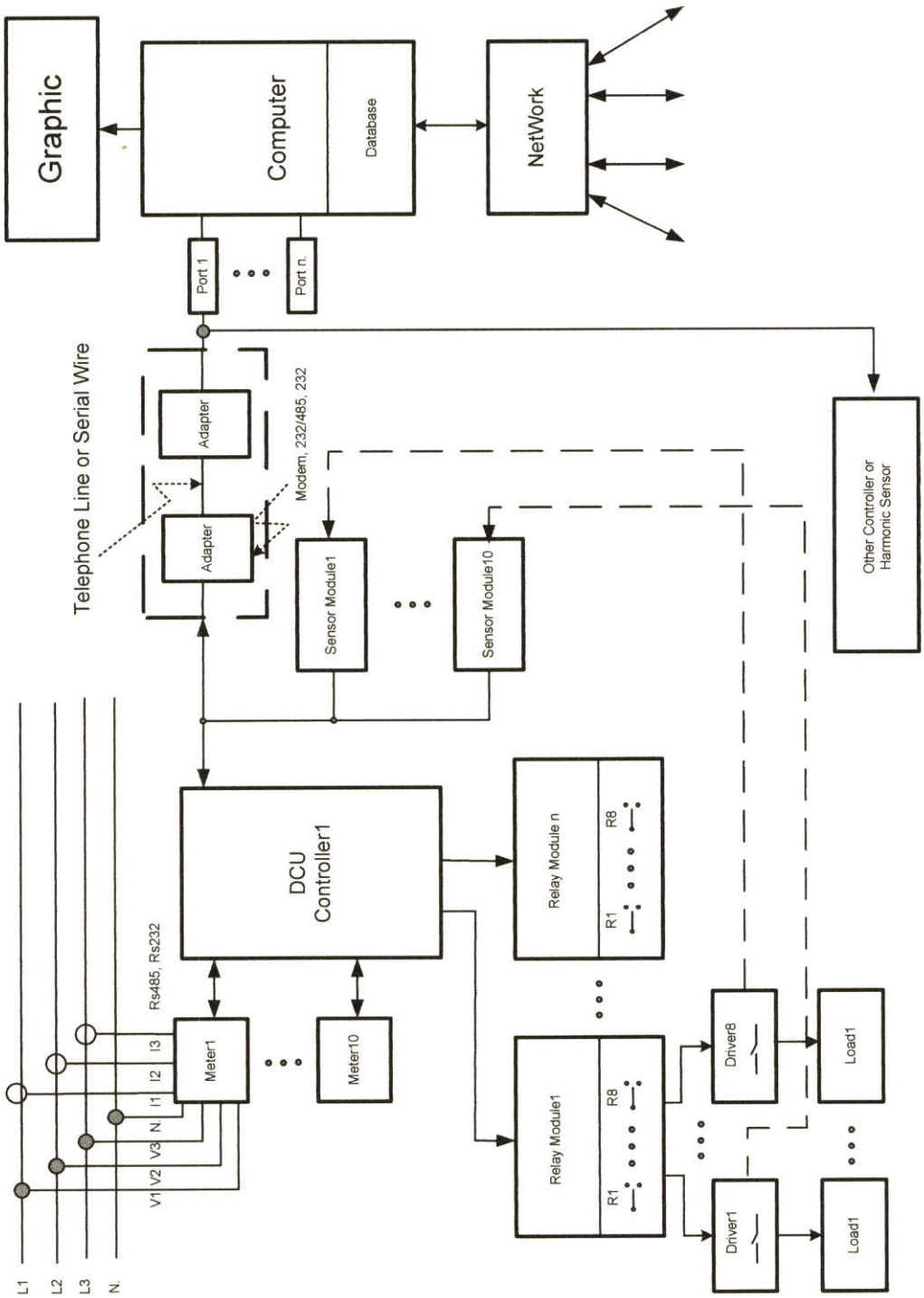
- Comport เดียวสามารถต่อชุดควบคุมได้หลายชุด (ไม่เกิน10 ชุด)
- อ่านค่าความต้องการพลังไฟฟ้าจากมิเตอร์มารวมกันแล้วควบคุมโหลดตามฟังก์ชั่นที่กำหนด
- อ่านค่าทางไฟฟ้าต่างๆจากมิเตอร์เพื่อเก็บเข้า หน่วยความจำ หรือ ส่งให้คอมพิวเตอร์ โดยผ่านพอร์ท อนุกรม หรือ โมเด็ม
- โหลดข้อมูลทางไฟฟ้า เช่น แรงดัน, กระแส, กำลัง ฯลฯ แบบ เวลาจริง (Real Time) ได้
- ควบคุมการปิด-เปิดโหลด ตามขนาด และ วัน เวลาที่กำหนด
- ใช้เป็น ตัวบันทึกข้อมูลทางไฟฟ้า (Data Logger) ได้ ทั้งในกรณีที่มี หรือ ไม่มีชุดรีเลย์ควบคุมโดยบันทึกข้อมูลลงหน่วยความจำ
- ใช้เก็บข้อมูลทางไฟฟ้า และ แสดงผลฮาร์มอนิค
- มีการแสดงผลที่คอมพิวเตอร์ ทั้งแบบตัวหนังสือ และกราฟ

- มีรายงานข้อมูล
- สามารถ Import และ Export ข้อมูลได้ (Text, Word, Excel)
- บันทึกข้อมูลลงดาต้าเบส
- มีระบบป้องกันการใช้งาน (Password)
- สามารถบันทึกประวัติการใช้งานระบบได้
- ข้อมูล คำสั่ง และพารามิเตอร์ต่างๆไม่สูญหายหากไฟดับ
- กำหนดอัตราเร็ว และรูปแบบการรับส่งข้อมูลได้
- สามารถเชื่อมต่อมิเตอร์ได้หลายตัว (ไม่เกิน 10 ตัว) บนบัส อนุกรม RS 485
- สามารถส่งข้อมูลผ่านโมเด็มให้คอมพิวเตอร์ที่อยู่ไกลออกไปได้
- มีระบบป้องกันข้อมูลผิดพลาด และการหยุดทำงานของระบบ
- มีระบบป้องกันแรงดันเกิน, แรงดันต่ำ และฟิลเตอร์
- สามารถเลือกจัดกลุ่มการควบคุมโหลดในโหมดต่างๆได้
- มีการคำนวณค่าไฟฟ้าทั้งชนิด TOU, TOD และวิเคราะห์ระบบ
- สามารถตั้งค่าการทำงานได้จากคอมพิวเตอร์
- ค่าการทำงานสามารถบันทึกไว้ใช้ต่อไปได้
- สามารถรีเซ็ตตัวเองได้หากเกิดการหยุดของระบบ
- โปรแกรมคอมพิวเตอร์ สามารถกลับมาทำงานต่อ ณ. จุดเดิมได้หากเริ่มเครื่องใหม่
- สามารถตั้งเวลาหน่วง เปิด-ปิด รีเลย์ ได้
- แต่ละโมดูลสามารถนำไปใช้แยกทำงานอย่างอื่นได้อย่างอิสระภายใต้การควบคุมเฉพาะ เช่น นำตัวควบคุมไปใช้บันทึกข้อมูลหรือนำชุดรีเลย์ ไปใช้ควบคุมด้วยตัวสั่งงานอื่น เป็นต้น
- ระบบสามารถขยายหรือลดได้อย่างอิสระ

4.3 การทำงานของระบบควบคุมความต้องการพลังไฟฟ้า

ระบบควบคุมความต้องการพลังไฟฟ้าประกอบด้วยมิเตอร์วัดค่าความต้องการพลังไฟฟ้า (Power Demand Meter) ตรวจจับแรงดัน และกระแสไฟฟ้า 3 เฟสเข้ามาคำนวณแล้วเก็บค่าต่างๆที่คำนวณไว้ในหน่วยความจำเพื่อแสดงผล และรอการเรียกขอจากภายนอกผ่านพอร์ทอนุกรม การควบคุมเกิดจากตัวควบคุมความต้องการพลังไฟฟ้า (DCU Controller) ดังรูปที่ 4.1 ส่งคำสั่ง (Command) ผ่านบัสอนุกรม RS232C หรือ RS485 เพื่อร้องขอข้อมูลในหน่วยความจำของมิเตอร์ โดยอ้างถึงแอดเดรสที่ใช้เก็บข้อมูลค่าความต้องการพลังไฟฟ้าปัจจุบันภายในมิเตอร์ และแอดเดรสของหน่วยความจำที่ต้องการ หลังจากมิเตอร์ได้รับคำสั่งขอข้อมูลที่ถูกต้องแล้วจะส่งข้อมูล ณ ตำแหน่งหน่วยความจำที่ถูกอ้างถึงออกมา ตัวควบคุมความต้องการพลังไฟฟ้าจะนำค่าความต้องการ

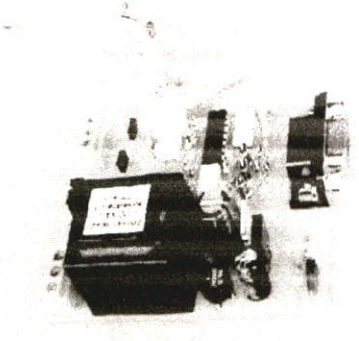
พลังไฟฟ้าที่ได้มาพร้อมทั้งอ่านค่าเวลาปัจจุบันจากเรียลไทม์คล็อค (RTC) ภายในมาประมวลผลเปรียบเทียบกับโปรแกรมที่ถูกป้อนให้ทำงานจากคอมพิวเตอร์ก่อนหน้านี้ เพื่อคำนวณหาค่าควบคุมส่งไปยัง โมดูลรีเลย์เพื่อควบคุมการใช้หลังไฟฟ้า ส่วนโมดูลอินพุททำหน้าที่เป็นดาต้าล็อคเกอร์ (Data Logger) เก็บข้อมูลการตัดต่อโหลดเพื่อใช้ในการวิเคราะห์ต่อไปโดยจะเก็บข้อมูลไว้ในหน่วยความจำภายในเพื่อรอการเรียกขอข้อมูลจากภายนอกผ่านพอร์ทอนุกรม RS232C หรือ RS485 ส่วนคอมพิวเตอร์ทำหน้าที่ป้อนข้อมูลการควบคุม, เรียกเก็บ และบันทึกข้อมูล, ควบคุมโหลดโดยตรง, คำนวณ,ติดต่อแลกเปลี่ยนข้อมูลกับระบบอื่นผ่านเน็ตเวอร์ค และแสดงผลข้อมูล ในระบบควบคุมไม่จำเป็นต้องมีคอมพิวเตอร์ก็ได้ หลังจากที่ป้อนคำสั่งควบคุมเสร็จก็สามารถปลดคอมพิวเตอร์ออกจากระบบได้ และจะต่อใช้งานเมื่อต้องการเก็บข้อมูล,ควบคุม หรือป้อนคำสั่งควบคุมใหม่ การที่สามารถปลดคอมพิวเตอร์ออกจากระบบแล้วยังสามารถทำงานต่อได้มีข้อดีที่ระบบไม่ขึ้นอยู่กับคอมพิวเตอร์ ถ้าคอมพิวเตอร์ล่มหรือเกิดหยุดการทำงานระบบยังสามารถควบคุมค่าความต้องการพลังไฟฟ้าต่อได้เป็นปกติ



รูปที่ 4.1 ระบบควบคุมความต้องการพลังไฟฟ้า

4.4 การออกแบบโมดูลแปลง RS232C เป็น RS 485 ชนิดแยกกราวด์ (RS232C to RS 485 Converter Isolate Module)

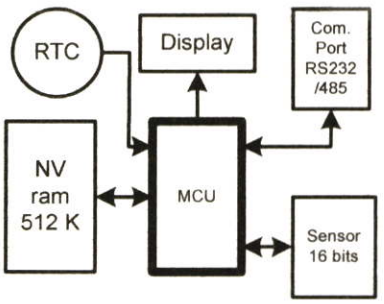
เป็นวงจรแปลง มาตรฐานการสื่อสารอนุกรม RS232C เป็น RS 485 ชนิดแยกกราวด์ เพื่อลดสัญญาณรบกวน และ สามารถส่งข้อมูลได้ไกลขึ้นสูงสุดถึง1.2 กิโลเมตร สามารถต่ออุปกรณ์ได้ 32 จุด บนบัสเดียวกันด้วยไอซีเบอร์ 75176 และ 256 จุด หากใช้ไอซีพิเศษบางเบอร์ มีไฟแสดงผลสถานะข้อมูลขาเข้าและขาออก ควบคุมทิศทางข้อมูลได้ 2 โหมดคือ เลือกทิศทางอัตโนมัติ กับใช้สัญญาณทริกเลือกทิศทาง ในโหมดเลือกทิศทางอัตโนมัติจะใช้สัญญาณ Rx ป้อนกลับเข้ามากำหนดทิศทางถ้าหากมีข้อมูลเข้ามาก็เปลี่ยนเป็นตัวรับข้อมูล ถ้าไม่มีข้อมูลเข้ามาก็เป็นตัวส่งข้อมูล การทำงานในโหมดนี้ไม่สามารถรับ-ส่งข้อมูลความเร็วสูงได้เนื่องจากต้องเสียเวลารอเปลี่ยนทิศทางของข้อมูล ปกติจะตั้งให้ทำงานที่ไม่เกิน 19200 bps ส่วนในโหมดใช้สัญญาณทริกต้องใช้สัญญาณ “0” เพื่อรับข้อมูล และใช้สัญญาณ “1” เพื่อส่งข้อมูล ตัวควบคุมจะต้องรู้ว่าขณะนี้ต้องการรับหรือส่งเพื่อสร้างสัญญาณเลือกทิศทาง ในขณะที่ไม่มีการส่งข้อมูลให้เคลียร์สัญญาณขาทริกเป็น “0” เพื่อรอรับข้อมูล การต่อใช้งานกับคอมพิวเตอร์จะใช้ขาสัญญาณ RTS ของ RS232C เป็นตัวกำหนดทิศทางการไหลของข้อมูลโดยการเขียนโปรแกรมเปลี่ยนสถานะก่อนที่จะมีการรับ-ส่ง ถ้าหากต้องการสื่อสารข้อมูลให้ไกลกว่านั้นก็สามารถใช้อุปกรณ์ทวนสัญญาณ หรือโมเด็มแทนได้ การแยกกราวด์ใช้ Opto เบอร์ 6N137 เป็น Fast Speed Opto



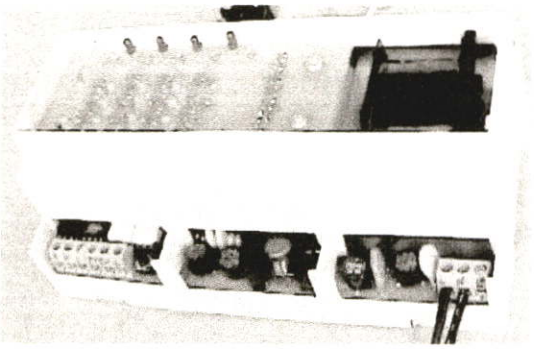
รูปที่ 4.2 โมดูลแปลง RS232C เป็น RS 485 ชนิดแยกกราวด์ที่ออกแบบสร้าง และพัฒนาขึ้น

4.5 การออกแบบโมดูลตรวจเช็คบันทึกสถานะ (Input Module)

โมดูลตรวจเช็คบันทึกสถานะทำหน้าที่ตรวจเช็ค และเก็บบันทึกข้อมูลสถานะของโหลด เพื่อนำไปใช้วิเคราะห์ระบบต่อไป ใช้ไมโครคอนโทรลเลอร์ MCS-51 เป็นหน่วยประมวลผล มีไฟแสดงสถานะของอินพุท การเข้าออกของข้อมูล และแสดง Error ของหน่วยความจำ และฐานเวลาจริง เนื่องจาก โมดูลตรวจเช็คบันทึกสถานะนี้จะใช้เก็บบันทึกข้อมูลสถานะอินพุทตามวันเวลาจริง ถ้าหากหน่วยความจำ หรือฐานเวลาจริงผิดพลาดก็จะทำให้ได้ข้อมูลที่ผิดพลาดทั้งหมด รูปที่ 4.3 แสดงบล็อคโครงสร้างภายในของโมดูลตรวจเช็คบันทึกสถานะ



รูปที่ 4.3 บล็อกไดอะแกรมของโมดูลตัวตรวจเช็คบันทึกสถานะ



รูปที่ 4.4 โมดูลตัวตรวจเช็คบันทึกสถานะที่ออกแบบสร้าง และพัฒนาขึ้น

โมดูลตรวจเช็คบันทึกสถานะในรูปที่ 4.4 ใช้บันทึกสถานะของโหลดตามความเป็นจริง เพื่อใช้ในการวิเคราะห์การทำงานของระบบควบคุมความต้องการพลังไฟฟ้าว่ามีการควบคุมโหลดที่จุดไหน เวลาไหนบ้าง โดยจะบันทึกข้อมูลสถานะอินพุท 16 บิต เมื่อเกิดการเปลี่ยนแปลงของอินพุท บิตใดบิตหนึ่ง ก็จะบันทึกสถานะขณะนั้นพร้อมกับเวลาปัจจุบันทันที อินพุทออกแบบให้แยกกราวด์เพื่อป้องกันความเสียหายกับวงจรหากเกิดสัญญาณรบกวน หรือ แรงดันเกิน อินพุทใช้

ระดับแรงดัน 0 Vdc. และ12 Vdc. บอกระดับลอจิก “0” และ “1” การเก็บข้อมูลในหน่วยความจำเป็นลักษณะวนเก็บซ้ำไปเรื่อยๆ บนหน่วยความจำ NVRAM ขนาด 131 KB ข้อมูลที่บันทึกประกอบด้วย data1 + data2 + year + month + date + hour + minute + second รวมทั้งหมด 8 ไบต์ เวลาในการเปลี่ยนแปลงสถานะจะต้องไม่ต่ำกว่า 1 วินาที จึงจะสามารถบันทึกได้เพื่อป้องกันการเปลี่ยนของสถานะที่เวลาเดียวกัน หรือการสั่นของสัญญาณอินพุท หากข้อมูลเต็มก็จะลบข้อมูลเก่าสุดทิ้งแล้วแทนที่ด้วยข้อมูลใหม่ มีพอร์ทสื่อสารข้อมูล RS232C และ RS485 ชนิดแยกกราวด์เพื่อป้องกันสัญญาณรบกวน เลือกใช้อย่างใดอย่างหนึ่ง

4.5.1 คุณสมบัติของโมดูลตัวตรวจเช็คบันทึกสถานะ

- ใช้ MCU ตระกูล MCS-51 Crystal clock at 11.0592 MHz. ที่ Speed ×2
- บันทึก Parameter คือ address, parity และ baud rate ในหน่วยความจำ EEPROM
- Switching Power Supply
- Software Error Protection
- Noise Filter
- มี Input ชนิด Isolate
- มีการติดต่อสื่อสารอนุกรม RS485 หรือ RS232C ชนิด Isolate
- หน่วยความจำ NVRAM ขนาด 131 Kbytes
- RTC with Battery Backup
- มีโหมด Default Setup
- 16 Bits Input Detection
- ตั้งค่า Baud Rate ได้ 2 ค่าคือ 9600 bps และ 19,200 bps
- ตั้งค่า Parity ได้ 5 ค่า คือ none, odd, even, space และmark

4.5.2 การป้องกันโมดูลตัวตรวจเช็คบันทึกสถานะ

การออกแบบเพื่อใช้งานจริงจำเป็นต้องมีการป้องกันระบบอันเนื่องมาจากสภาวะแวดล้อมหรือจากการใช้งานเพราะจะช่วยลด และป้องกันไม่ให้เกิดความเสียหายในระบบการป้องกันแบ่งเป็น 2 ประเภทใหญ่ๆ คือ การป้องกันฮาร์ดแวร์ ตามตารางที 4.1 และการป้องกันซอฟแวร์ ตามตารางที่ 4.2

ตารางที่ 4.1 การป้องกันฮาร์ดแวร์ของโมดูลตรวจเช็คบันทึกสถานะ

ลำดับที่	ชนิด	ฮาร์ดแวร์
1	สัญญาณรบกวนจากแหล่งจ่ายไฟ	วงจร L-C Fiter
2	กระแสเกิน	ฟิวส์
3	สัญญาณ Surge จากแหล่งจ่ายไฟ	วงจร Varistor 275 V.
4	สัญญาณ Surge จากวงจรสื่อสาร	วงจร varistor 7.5 V.
5	แรงดันต่ำ	วงจร Switching Power Supply
6	สัญญาณรบกวนจาก Input	วงจร Opto Isolate
7	สัญญาณรบกวนจากวงจรสื่อสาร	วงจร Opto Isolate
8	กระแสเกินในวงจรสื่อสาร	ฟิวส์

ตารางที่ 4.2 การป้องกันซอฟแวร์ของโมดูลตรวจเช็คบันทึกสถานะ

ลำดับที่	ชนิด	คอมพิวเตอร์	การแก้ไข	โมดูล อินพุท	การแก้ไข
1	Watch Dog	ไม่มี	ไม่มี	มี	รีเซ็ต
2	Framing Error	ไม่มี	ไม่มี	มี	ไม่นำคำสั่งมาใช้
3	CRC Error	มี	ส่งซ้ำ	มี	รีเซ็ตการรับข้อมูล
4	Byte Out Of Time	ไม่มี	ส่งซ้ำ	มี	รีเซ็ตการรับข้อมูล
5	Total Out Of Time	มี	ส่งซ้ำ	ไม่มี	ไม่มี
6	Address Recognition	ไม่มี	ไม่มี	มี	ไม่ Interrupt
7	Check Nvram Status	ไม่มี	ไม่มี	มี	LED แสดงผล
8	Check RTC Status	ไม่มี	ไม่มี	มี	LED แสดงผล
9	Reduce EMI	ไม่มี	ไม่มี	มี	ไม่ส่งพัลส์ที่ขา ALE

4.5.3 การออกแบบการติดต่อสื่อสาร

การติดต่อสื่อสารกับอุปกรณ์ภายนอก จะต้องมีรูปแบบข้อกำหนดเฉพาะเพื่อเป็นการบอกให้ทั้งสองฝ่ายเข้าใจตรงกันถึงข้อมูลที่ได้รับ ในการติดต่อกับ Master แบ่งออกเป็นสองประเภทหลักๆคือ

1. Master ขอดูข้อมูล
2. Master ขอแก้ไขข้อมูล

การส่งข้อมูลมีรูปแบบทั่วไปดังรูปที่ 4.5

รูปที่ 4.5 รูปแบบการติดต่อสื่อสารกับโมดูลตัวตรวจเช็คบันทึกสถานะ

รูปแบบการติดต่อสื่อสารในรูปที่ 4.5 เป็นรูปแบบการสื่อสารที่ใช้กับโมดูลอื่นด้วย เริ่มจากการส่งคำสั่งร้องขอข้อมูลออกไปยังอุปกรณ์ตำแหน่งที่ต้องการ เมื่ออุปกรณ์ในระบบสื่อสารที่มีตำแหน่งตรงกับ ตำแหน่งที่ร้องขอมาก็จะตรวจสอบคำสั่ง แล้วส่งข้อมูลย้อนกลับไป

ตารางที่ 4.3 ฟังก์ชั่นที่ใช้ควบคุมโมดูลตรวจเช็คบันทึกสถานะ

หมายเลขฟังก์ชั่น	ความหมาย
01	ตั้งค่าเริ่มต้น
02	ขอข้อมูลในหน่วยความจำ
03	ตั้งค่าแอดเดรส
04	ขอค่าแอดเดรส
05	ตั้งค่าบอดเรท
06	ลบข้อมูลในหน่วยความจำ
07	ตั้งเวลา
08	ตั้งค่าพาริตี้
09	ตั้งโหมดการส่งข้อมูล

4.5.4 การออกแบบซอฟแวร์

การติดต่อเพื่อตั้งค่าหรือขอข้อมูลถูกแบ่งเป็น 9 ชนิดตามตารางที่ 4.3 หมายเลขฟังก์ชั่นเป็นรหัสใช้บอกว่าคำสั่งนั้นๆให้ทำอะไร การสื่อสารต้องกำหนดรูปแบบเฉพาะ (Protocol) เพื่อให้ฝ่ายส่งและฝ่ายรับมีความเข้าในตรงกัน การกำหนดรูปแบบเฉพาะจะทำให้การสื่อสารข้อมูลมีความกระชับ ส่งผลให้การรับส่งข้อมูลมีประสิทธิภาพสูงขึ้น ซอฟแวร์ที่ใช้ได้กำหนดรูปแบบดังนี้

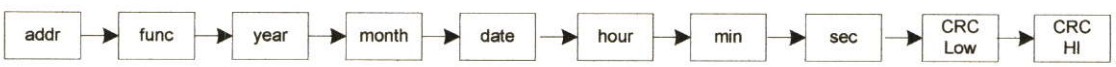
- การรับข้อมูล ใช้ Interrupt Serial
- การส่งข้อมูล ไม่ใช้ Interrupt Serial ใช้การวนส่งแทน
- เมื่อรับข้อมูลเข้ามา 1 ไบต์แรก ก็จะเริ่มตั้งเวลาคอย เป็นเวลา 10 mS. เพื่อคอย การรีเซ็ตข้อมูลอนุกรม ถ้าหากไม่มีข้อมูลไบต์ถัดไปเข้ามา
- ตัวโปรแกรมหลักจะคอยตรวจสอบ Input ตลอดเวลาหากมีสถานะเปลี่ยนแปลง และคงค่านั้นไว้เกิน 1 วินาที จะทำการบันทึกค่าสถานะ Input ทั้ง 16 บิต พร้อมทั้งเวลาปัจจุบันลงในหน่วยความจำ

รูปแบบการรับส่งข้อมูลจะแบ่งเป็น 2 ส่วนหลักๆ คือส่วนของผู้ส่งคำสั่ง (Master) และผู้ตอบกลับคำสั่ง (Slave) ส่วนใหญ่แล้วอุปกรณ์ที่ใช้ส่งคำสั่งจะเป็นคอมพิวเตอร์เพราะว่าคอมพิวเตอร์มีหน่วยความจำขนาดใหญ่ และง่ายต่อการโปรแกรม เมื่อโมดูลตรวจเช็คบันทึกสถานะได้รับคำสั่งที่มีรูปแบบถูกต้องจะตอบสนองต่อคำสั่งนั้น และส่งข้อมูลบางอย่างตอบกลับไปให้กับผู้ส่งเพื่อบอกว่าได้รับข้อมูลเรียบร้อยแล้ว

จากตารางที่ 4.3 คำสั่งทั้ง 9 ฟังก์ชั่นสามารถแบ่งกลุ่มได้ 2 กลุ่มคือ กลุ่มคำสั่งที่ใช้ข้อมูลเพียงไบต์เดียว และกลุ่มคำสั่งที่ใช้ข้อมูล 6 ไบต์

กลุ่มคำสั่งที่ใช้ข้อมูลเพียงไบต์เดียวคือ ฟังก์ชั่น 1-6 และ 8-9 ข้อมูลที่ส่งมาจะนำไปใช้ประกอบกับฟังก์ชั่น เช่นฟังก์ชั่น 05 หมายถึงการตั้งค่าบอดเรท ค่าข้อมูลที่ส่งแนบมาด้วยจะนำไปใช้ตั้งค่าบอดเรท ค่า 96D หมายถึงตั้งบอดเรท 9600 bps ค่า 192D หมายถึงตั้งบอดเรท 19200 bps เป็นต้น

รูปที่ 4.6 บล็อคคำสั่งที่มีข้อมูลไบต์เดียว

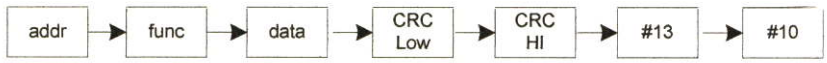


รูปที่ 4.7 บล็อกคำสั่งที่มีข้อมูล 6 ไบต์

นอกเหนือจากกลุ่มคำสั่งที่มีข้อมูลไบต์เดียวเป็นคำสั่งขอตั้งวันเวลาดังรูปที่ 4.7 ค่าพารามิเตอร์ที่ส่งตามมาอีก 6 ไบต์จะเป็นค่าวันเวลาปัจจุบันที่ส่งมาตั้งให้กับฐานเวลาจริง

ข้อมูลคำสั่งทั้งหมดประกอบด้วย 4 ส่วนหลักคือ หมายเลขตำแหน่ง (Address), ฟังก์ชั่น (Function), ข้อมูล (Data) และ เช็คข้อผิดพลาด (CRC Error) ผู้รับจะไม่รับข้อมูลไบต์ถัดไปถ้าหากข้อมูลไบต์แรกไม่ใช้หมายเลขตำแหน่งของตน ถ้าข้อมูลไบต์แรกตรงกับหมายเลขตำแหน่งแล้วจะเปิดรอรับข้อมูลไบต์ถัดไปตามเวลาที่กำหนดประมาณ 10 mS. ถ้าไม่มีข้อมูลเข้ามาก็จะรีเซ็ตการรับข้อมูลนั้นทันทีเพื่อรอรับข้อมูลไบต์ใหม่ที่มีค่าตรงกับหมายเลขตำแหน่ง หรือถ้ามีข้อมูลไบต์ที่ 2 เข้ามาแต่ไม่อยู่ในย่าน 0-9 ก็จะรีเซ็ตการรับข้อมูลเช่นกัน หลังจากรับค่าฟังก์ชั่นที่ถูกต้องเข้ามาแล้วก็คำนวณหาจำนวนข้อมูลที่เหลือเพื่อตั้งตัวนับจำนวนข้อมูลให้ตรงกับผู้ส่งหลังจากรับข้อมูลครบแล้วก็นำข้อมูลที่รับได้ทั้งหมดมาเข้ารหัส CRC-16 เพื่อหาค่า CRC Low และ CRC Hi นำมาเปรียบเทียบกับค่าที่ส่งมาถ้าตรงกันก็ประมวลผลคำสั่งนั้น แล้วส่งข้อความตอบกลับไปบอกผู้ส่งว่าได้รับคำสั่งถูกต้องเรียบร้อยแล้ว แต่ถ้า CRC ไม่ตรงกันก็ไม่นำคำสั่งนั้นไปประมวลผล แล้วรีเซ็ตการรับข้อมูลและไม่ส่งข้อความตอบกลับ เพราะข้อมูลที่รับเข้ามามีข้อผิดพลาดเกิดขึ้น

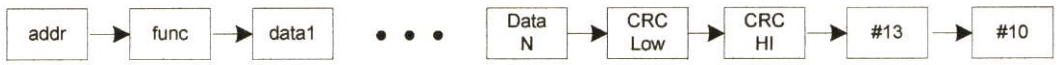
เมื่อได้รับข้อมูลคำสั่งถูกต้องครบถ้วนแล้วจะต้องมีการส่งข้อความตอบกลับในลักษณะทวนคำสั่ง หรือส่งข้อมูลตามคำสั่งที่ร้องขอ คำสั่งตั้งค่าเมื่อประมวลผลเสร็จจะส่งคำสั่งนั้นย้อนกลับไปเพื่อบอกว่าได้รับและประมวลผลคำสั่งแล้ว ข้อความที่ส่งย้อมกลับจะเพิ่มตัวปิดท้ายข้อมูลจำนวน 2 ไบต์เพื่อใช้บอกผู้รับว่าสิ้นสุดการส่งข้อมูลแล้ว และยังใช้ตัวปิดท้ายนี้ช่วยอินเทอร์รัพ (Interrupt) การรับข้อมูลด้วย ในกรณีใช้คอมพิวเตอร์เป็นตัวรับข้อมูลสามารถตั้งค่าให้ข้อมูลที่เข้ามาถูกพักไว้ในบัฟเฟอร์ชั่วคราวก่อนเพื่อรอให้ได้ข้อมูลครบก่อนจึงเข้าสู่การประมวลผล ตัวปิดท้ายนี้จะเป็นตัวบอกว่าเมื่อไหร่จึงจะดึงข้อมูลในบัฟเฟอร์มาประมวลผลได้



รูปที่ 4.8 บล็อคข้อมูลทวนคำสั่งกลับ

เมื่อได้รับคำสั่งประเภทขอข้อมูลข้อความที่ตอบกลับจะเป็นข้อมูลในหน่วยความจำตามที่ผู้ขอกำหนดมากับพารามิเตอร์ การขอข้อมูลใช้ฟังก์ชั่น 02 ตามด้วยรูปแบบข้อมูลที่ต้องการแบ่งเป็น 2 ชนิดคือ ข้อมูลเป็นบล็อคละ 8 ไบต์ ดังรูปที 4.9 และข้อมูลเป็นบล็อคที่มีความยาวของข้อมูลเป็นจำนวนตามที่กำหนดมาดังรูปที่ 4.10

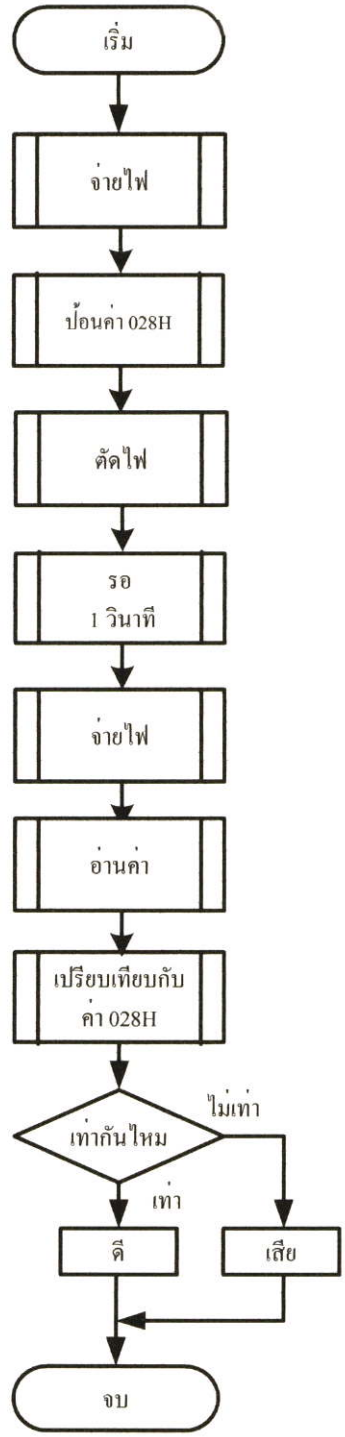
รูปที่ 4.9 บล็อกข้อมูลตอบกลับที่มีความยาวข้อมูล 8 ไบต์



รูปที่ 4.10 บล็อกข้อมูลตอบกลับที่มีความยาวข้อมูล N ไบต์

4.5.5 การเช็คหน่วยความจำ NVRAM

สถานะหน่วยความจำมีความสำคัญมากถ้าหากแบตเตอรี่แบคอัพเสื่อมทำให้ไม่สามารถเก็บรักษาข้อมูลไว้ได้ข้อมูลที่เก็บมาก็นำไปใช้ไม่ได้ เนื่องจากโมดูลตรวจเช็คบันทึกสถานะมีการบันทึกข้อมูลบ่อยครั้ง และต้องการเก็บรักษาข้อมูลไว้ได้ในขณะที่ไม่มีไฟเลี้ยงจึงเลือกหน่วยความจำชนิด NVRAM ที่สามารถบันทึกข้อมูลได้ไม่จำกัดจำนวนครั้งและข้อมูลไม่สูญหายหากไฟดับ แต่มีข้อเสียตรงที่แบตเตอรี่แบคอัพมีอายุการใช้งานประมาณ 5-10 ปี ขึ้นอยู่กับรุ่น และยี่ห้อ จึงต้องมีการตรวจเช็คสถานะของหน่วยความจำทุกครั้งก่อนใช้งาน เพื่อให้มั่นใจว่าข้อมูลที่บันทึกจะไม่สูญหาย ขั้นตอนการเช็คสถานะหน่วยความจำได้แสดงในรูปที่ 4.11



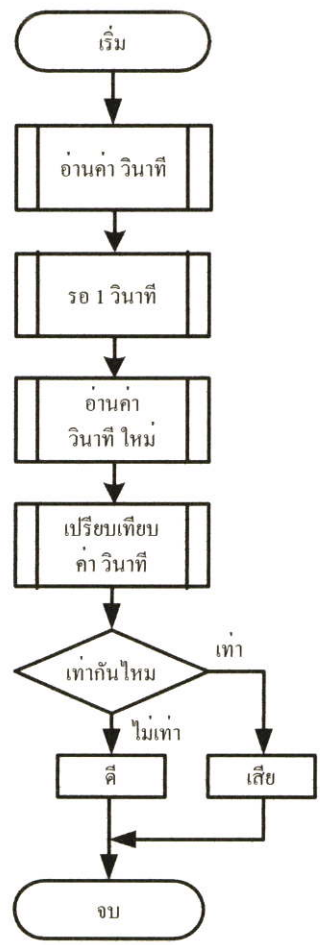
รูปที่ 4.11 ขั้นตอนการเช็คสถานะหน่วยความจำ NVRAM

การเช็คสถานะหน่วยความจำ NVRAM จะต้องป้อนค่าใดค่าหนึ่งเข้าไปเก็บไว้ในหน่วยความจำก่อนแล้วตัดแหล่งจ่ายไฟเลี้ยงออกแล้วรอเวลาคืนสภาพ หลังจากนั้นก็จ่ายไฟคืนแล้วอ่านค่าข้อมูล ณ ตำแหน่งหน่วยความจำที่ป้อนข้อมูลไว้ นำมาเปรียบเทียบกับข้อมูลที่ป้อนให้ตอนแรกว่าตรงกันหรือไม่ ถ้าตรงกันแสดงว่าหน่วยความจำยังสามารถเก็บข้อมูลไว้ได้แม้ไม่มี

แหล่งจ่ายไฟ แต่ถ้าข้อมูลที่ได้ไม่ตรงกันแสดงว่าแบตเตอรี่แบ็คอัพข้อมูลเสื่อมต้องเปลี่ยนหน่วยความจำก่อนจึงจะใช้งานได้

4.5.6 การเช็คฐานเวลาจริง

ข้อมูลสถานะจะถูกบันทึกพร้อมกับวันเวลาจริงในปัจจุบันถ้าหากวันเวลาไม่ตรงก็ทำให้ข้อมูลที่บันทึกผิดพลาดจากความเป็นจริง หรือว่านาฬิกาฐานเวลาไม่เดิน ก็ไม่สามารถบอกได้ว่าข้อมูลที่บันทึกนั้นเกิดขึ้นที่เวลาใด ก่อนเริ่มเครื่องจะต้องตรวจเช็คฐานเวลาจริงก่อนว่ายังทำงานอยู่หรือไม่ มีขั้นตอนการตรวจเช็คตามรูปที่ 4.12 การตรวจเช็คเริ่มจากการอ่านค่าวินาทีเข้ามาเก็บไว้ก่อน แล้วรอให้เวลาผ่านไป 1 วินาทีค่อยอ่านค่าวินาทีเข้ามาอีกที นำมาเปรียบเทียบกับค่าเก่าถ้าหากว่าได้ค่าเท่ากันแสดงว่าวงจรนาฬิกาไม่เดิน แต่ถ้าไม่เท่ากันหรือค่าใหม่ต่างจากค่าเก่า 1 วินาที แสดงว่าวงจรนาฬิกาทำงาน



รูปที่ 4.12 ขั้นตอนการเช็คสถานะฐานเวลาจริง

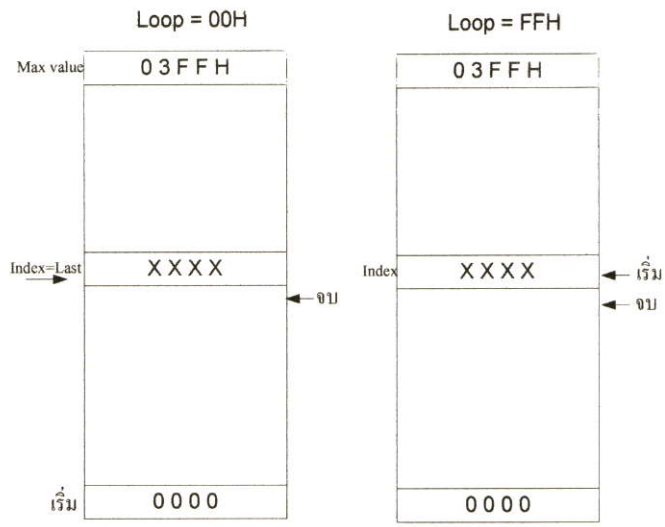
4.5.7 การบันทึกข้อมูลลงหน่วยความจำ

ข้อมูลสถานะอินพุทจะถูกบันทึกก็ต่อเมื่อสถานะมีการเปลี่ยนไปจากสถานะเดิม และคงสถานะนั้นค้างไว้เป็นเวลาไม่ต่ำกว่า 1 วินาทีเพื่อป้องกันการสั่นของอินพุท ทำให้มีการบันทึกสถานะ 2 สถานะในเวลาเดียวกัน การบันทึกข้อมูลลงหน่วยความจำนั้นจะบันทึกเป็นชุดของข้อมูลซ้อนกันไป และจะวนกลับมาบันทึกทับแทนข้อมูลตัวท้ายสุดเมื่อหน่วยความจำเต็ม โดยใช้พื้นที่ส่วนหัวของหน่วยความจำเก็บค่าลำดับการบันทึกปัจจุบันไว้เพื่อจะได้รู้ว่าขณะนี้ได้ใช้หน่วยความจำไปเท่าไหร่แล้ว และจะเริ่มบันทึกค่าต่อไปที่ตำแหน่งไหน รูปที่ 4.13 แสดงข้อมูลส่วนหัว และตัวอย่างข้อมูลชุดแรกในหน่วยความจำตำแหน่ง 00000 ถูกจองไว้ใช้สำหรับตรวจสอบสถานะของหน่วยความจำ NVRAM ตามหัวข้อที่ รูปที่ 4.13 ในตำแหน่งที่ 00001 ใช้บอกว่าข้อมูลที่บันทึกอยู่มีการวนบันทึกซ้ำแล้วหรือยัง ข้อมูลในตำแหน่ง 00002-00003 เป็นตำแหน่งที่ใช้บันทึกข้อมูลล่าสุด ตั้งแต่ตำแหน่งที่ 00004-0000B เป็นพื้นที่ใช้เก็บข้อมูลชุดแรก ข้อมูลชุดถัดไปจะถูกจัดเก็บต่อจากตำแหน่ง 0000B เป็นต้นไปจนถึงตำแหน่งสุดท้ายที่ 1FFFF

0000B	Second
0000A	Minute
00009	Hour
00008	Date
00007	Month
00006	Year
00005	Data Low
00004	Data Hi
00003	Index Low
00002	Index Hi
00001	Loop < FF=Loop 00= No Loop
00000	Test NVRam

รูปที่ 4.13 การบันทึกส่วนหัวของข้อมูล

ข้อมูลที่บันลงหน่วยความจำมี 2 ลักษณะคือ ช่วงที่ยังไม่มีการบันทึกซ้ำ และมีการบันทึกข้อมูลซ้ำแล้วรูปแบบการบันทึกจะแตกต่างกันดังรูปที่ 4.14



(a) ยังไม่วนบันทึก (b)มีการวนบันทึก

รูปที่ 4.14 การจัดลำดับของหน่วยความจำในขณะที่ (a) ยังไม่วนบันทึก (b) มีการวนบันทึก

เมื่อต้องการดึงข้อมูลจากหน่วยความจำส่งออกไปภายนอกหลังจากโหลดค่าออกมาจากหน่วยความจำแล้วจะต้องนำข้อมูลมาเพิ่มส่วนต่างๆ เพื่อรอการส่งออก ดังรูปที่ 4.15

11	CRC HI
10	CRC Low
9	Sec
8	Min
7	Hour
6	Date
5	Month
4	Year
3	Data Low
2	Data Hi
1	Func
0	Addr

รูปที่ 4.15 การจัดเรียงข้อมูลเพื่อส่งออก

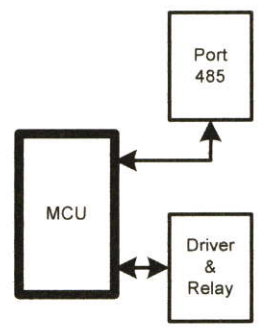
ยังมีพารามิเตอร์ที่ใช้ควบคุมบางส่วนถูกเก็บไว้เพื่อใช้งานเกี่ยวกับลักษณะการทำงานเท่านั้นจึงถูกจัดเก็บไว้ในหน่วยความจำ EEPROM แยกไว้ต่างหากดังรูปที่ 4.16

00003	Parity 00=None 01=Odd 02=Even | 03=Mark 04=Space
00002	Baud Rate 192D=19200 96D=9600
00001	Address

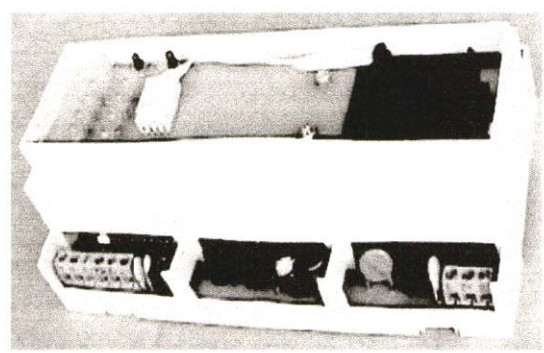
รูปที่ 4.16 พารามิเตอร์ที่จัดเก็บในหน่วยความจำ EEPROM

4.6 การออกแบบโมดูลรีเลย์ควบคุม (Relay Module)

ทำหน้าที่รับคำสั่งจากภายนอกเพื่อควบคุมรีเลย์ 8 ตัว คำสั่งที่ป้อนเข้ามาจะต้องมีรูปแบบ (Protocol) ที่ถูกต้องจึงจะสามารถทำงานได้มีพอร์ทสื่อสารข้อมูล RS232C และ RS485 ชนิดแยกกราวด์เพื่อป้องกันสัญญาณรบกวน เลือกใช้อย่างใดอย่างหนึ่ง



รูปที่ 4.17 บล็อกไดอะแกรมของชุดรีเลย์โมดูล



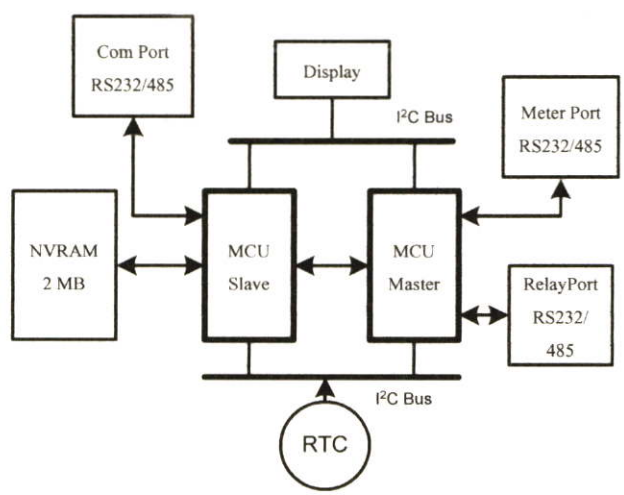
รูปที่ 4.18 ชุดรีเลย์โมดูลที่ออกแบบสร้าง และพัฒนาขึ้น

การออกแบบชุดคำสั่งมีลักษณะคล้ายกับโมดูลตรวจเช็คบันทึกสถานะ ต่างกันตรงที่ชุดรีเลย์โมดูลไม่มีหน่วยความจำสำหรับเก็บข้อมูลจึงเพิ่มหน่วยความจำ EEPROM เข้ามาเพื่อเก็บ

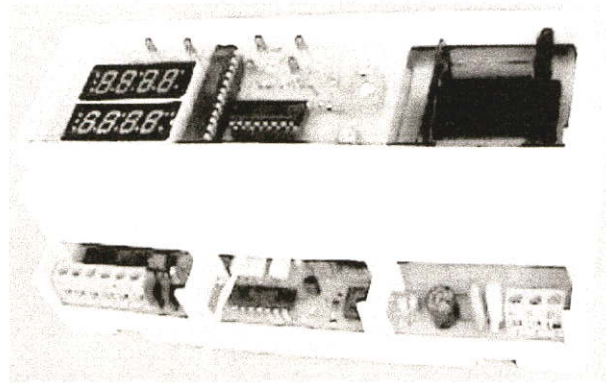
ค่าพารามิเตอร์ต่างๆ ที่ใช้ ในส่วนแสดงผลมี LED แสดงผลสถานะของรีเลย์ และสถานะของข้อมูลเข้าออก

4.7 การออกแบบโมดูลควบคุม และบันทึกความต้องการพลังไฟฟ้า (Demand Controller Module)

ตัวควบคุมมี 2 หน่วยประมวลผล (Microcontroller) ร่วมกันประมวลผล แบ่งเป็น ตัวหลัก (Master) กับ ตัวรอง (Slave) แยกกันทำงานคนละฝั่ง ตัวหลักจะทำงานติดต่อสื่อสารกับคอมพิวเตอร์ และอ่านข้อมูลจากตัวรองมาเก็บในหน่วยความจำ NVRAM ขนาด 2 MB ซึ่งเป็นหน่วยความจำที่ข้อมูลไม่สูญหายหากไฟดับ รอการเรียกขอจากคอมพิวเตอร์ ส่วนตัวรองจะทำหน้าที่อ่านค่าความต้องการพลังงานไฟฟ้าจาก มิเตอร์ ภายนอกมาประมวลผล แล้วส่งสัญญาณไปควบคุม ชุดรีเลย์ ตามโปรแกรมที่ตั้งไว้ ดังรูปที่ 4.19 จะสังเกตเห็นว่าทั้งตัวหลัก และตัวรองใช้ส่วนแสดงผล (Display) กับฐานเวลาจริง (Real Time Clock) ร่วมกันอยู่ ดังนั้นจึงต้องมีขาสัญญาณเป็นตัวบอกว่าฐานเวลาจริงกำลังถูกเรียกใช้อยู่หรือไม่ถ้าถูกใช้อยู่ก็ให้รอก่อน มีพอร์ทสื่อสารข้อมูล RS232C และ RS485 ชนิดแยกกราวด์เพื่อป้องกันสัญญาณรบกวน เลือกใช้อย่างใดอย่างหนึ่ง การออกแบบการป้องกัน และการติดต่อสื่อสารมีลักษณะคล้ายกับโมดูลตัวตรวจเช็คบันทึกสถานะจึงไม่ขอกล่าวถึง

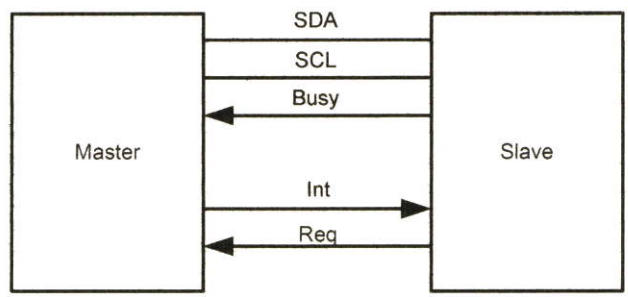


รูปที่ 4.19 บล็อคไดอะแกรมของโมดูลตัวควบคุมความต้องการพลังไฟฟ้า



รูปที่ 4.20 โมดูลตัวควบคุมความต้องการพลังไฟฟ้าที่ออกแบบสร้าง และพัฒนาขึ้น

เนื่องจากใช้ไมโครคอนโทรลเลอร์ 2 ตัวจึงต้องมีการออกแบบในส่วนการติดต่อสื่อสารกันระหว่างไมโครคอนโทรลเลอร์โดยใช้การสื่อสารแบบอนุกรม เพื่อลดสายสัญญาณรูปแบบการสื่อสารที่ออกแบบมีลักษณะคล้าย I^2C ต่างกันตรงที่เพิ่มสัญญาณควบคุมเข้ามาเพื่อกำหนดจังหวะและเช็คสถานะของกันและกันก่อนจะมีการสื่อสาร ดังรูปที่ 4.21

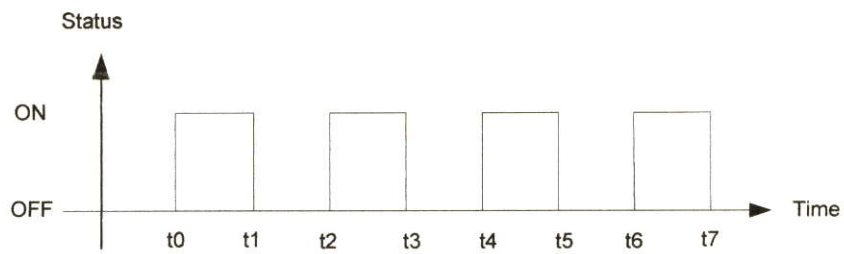


รูปที่ 4.21 สัญญาณสื่อสารระหว่างไมโครคอนโทรลเลอร์

การควบคุมแบ่งเป็น 2 ลักษณะคือการควบคุมในโหมดเวลา และการควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้าทั้ง2 โหมดสามารถแยกตั้งค่าให้รีเลย์ทำงานได้อย่างอิสระโดยการเข้าไปตั้งโปรแกรมเลือกให้รีเลย์มีสถานการณ์ทำงานในโหมดที่ต้องการ พร้อมทั้งตั้งค่าพารามิเตอร์ที่ใช้ควบคุมด้วย

4.7.1 การควบคุมในโหมดเวลา

รีเลย์ที่ทำงานในโหมดนี้จะ ON–OFF ในช่วงเวลาที่โปรแกรมไว้โดยไม่สนใจค่าพลังไฟฟ้า การทำงานในโหมดนี้ใช้ควบคุมโหลดที่รู้ช่วงเวลาการทำงานที่แน่นอนเพื่ออนุรักษ์พลังงานในเวลาที่ไม่ได้ใช้งานก็ปิดเองอัตโนมัติ แต่ถ้าต้องการให้ทำงานในกรณีพิเศษก็สามารถควบคุมได้โดยตรง การควบคุมในโหมดเวลานี้มีรูปแบบการควบคุมดังรูปที่ 4.22

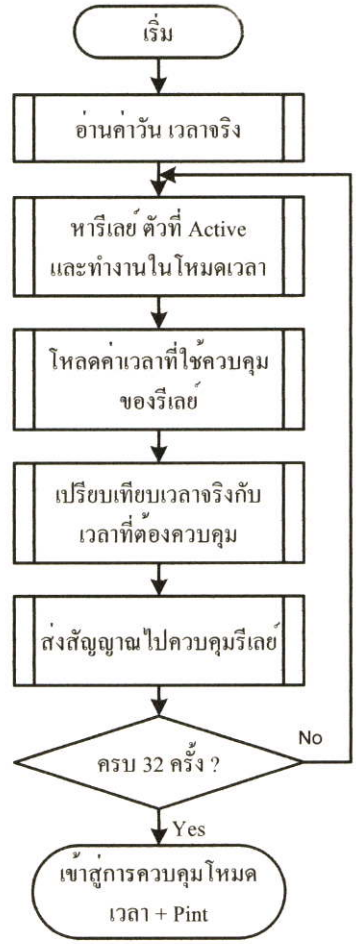


รูปที่ 4.22 การควบคุมในโหมดเวลา

ข้อมูลที่ใช้ควบคุมในโหมดนี้จึงเป็นข้อมูลของเวลาที่รีเลย์ทำงานใน 4 ช่วงของวัน ถูกจัดเก็บแยกส่วนไว้แต่ละรีเลย์เป็นค่าของชั่วโมง กับนาทีดังรูปที่ 4.23

t7 นาที.	16
t7 ชม.	15
t6 นาที.	14
t6 ชม.	13
t5 นาที.	12
t5 ชม.	11
t4 นาที.	10
t4 ชม.	9
t3 นาที.	8
t3 ชม.	7
t2 นาที.	6
t2 ชม.	5
t1 นาที.	4
t1 ชม.	3
t0 นาที.	2
t0 ชม.	1

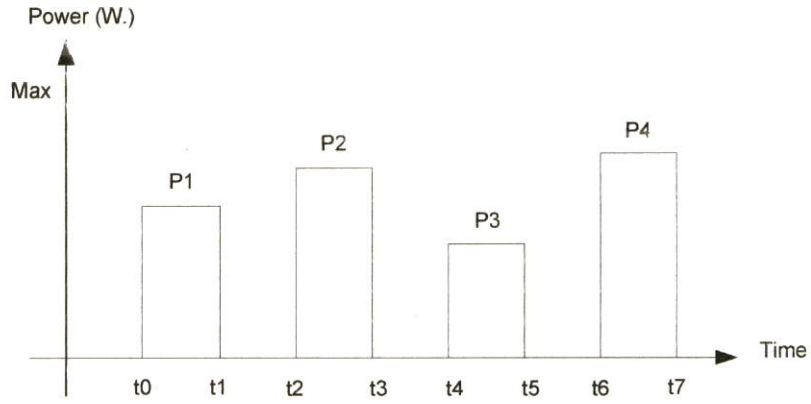
รูปที่ 4.23 ข้อมูลที่ใช้ควบคุมในโหมดเวลา



รูปที่ 4.24 โฟวล์ชาร์ทการควบคุมในโหมดเวลา

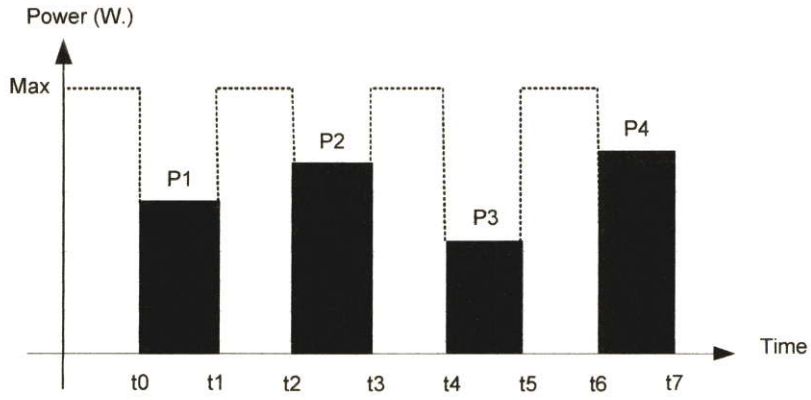
4.7.2 การควบคุมในโหมดเวลาร่วมกับค่าความต้องการพลังไฟฟ้า

ในโหมดนี้การควบคุมจะพิจารณาถึงค่าความต้องการพลังไฟฟ้าร่วมกับเวลาโดยจะพิจารณา เวลาที่ตั้งไว้ก่อน ถ้าหากไม่อยู่ในช่วงเวลาที่ต้องการควบคุมก็จะไม่มีการควบคุมเกิดขึ้น แต่ถ้าอยู่ในช่วงเวลาที่ต้องการควบคุมแล้ว จะมีการอ่านค่าพลังไฟฟ้าปัจจุบันจากมิเตอร์ที่ต่ออยู่แต่ละตัวมารวมกัน แล้วนำมาเปรียบเทียบกับค่าที่ตั้งไว้เพื่อทำการควบคุมโหลด ดังรูปที่ 4.25 แสดงตัวอย่างการควบคุมโหลดในโหมดเวลาร่วมกับพลังไฟฟ้า



รูปที่ 4.25 การควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้า

ในการควบคุมจริงในช่วงที่ไม่มีการควบคุมจะต้องป้อนค่าพลังไฟฟ้าที่ต้องการควบคุมเป็นค่าสูงสดเพื่อให้มีการสั่งงานรีเลย์คืนสภาพกลับแบบเรียงลำดับทีละตัว วิธีนี้เพื่อป้องกันการคืนสภาพของรีเลย์พร้อมกันทีเดียวดังนั้นในการควบคุมจริงจึงมีค่าการควบคุมดังรูปที่ 4.26



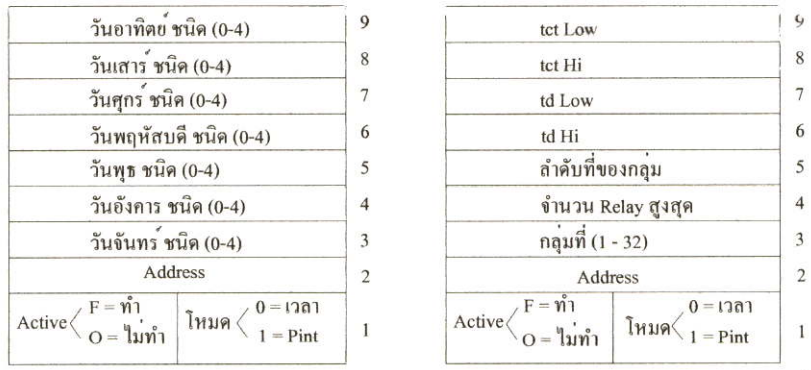
รูปที่ 4.26 ค่าที่ใช้ควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้า

ค่าพารามิเตอร์ที่จัดเก็บไว้ควบคุมจะเป็นค่าที่ใช้ร่วมกันทุกรีเลย์ที่ทำงานในโหมดนี้ การทำงานของรีเลย์จะทำงานเรียงลำดับกันไป เมื่อค่าพลังไฟฟ้าต่ำหรือสูงกว่าค่าที่ตั้งไว้ การควบคุมสามารถตั้งได้สูงสุด 4 ช่วงต่อวันดังนั้นค่าพารามิเตอร์ที่ใช้ควบคุมจะประกอบไปด้วยค่าพลังไฟฟ้าและค่าเวลาทั้ง 4 ช่วงดังรูปที่ 4.27

p4 Dimention	17
p4 Low	16
p4 Hi	15
p3 Dimention	14
p3 Low	13
p3 Hi	12
p2 Dimention	11
p2 Low	10
p2 Hi	20
p1 Dimention	19
p1 Low	18
p1 Hi	17
t7 นาที.	16
t7 ชม.	15
t6 นาที.	14
t6 ชม.	13
t5 นาที.	12
t5 ชม.	11
t4 นาที.	10
t4 ชม.	9
t3 นาที.	8
t3 ชม.	7
t2 นาที.	6
t2 ชม.	5
t1 นาที.	4
t1 ชม.	3
t0 นาที.	2
t0 ชม.	1

รูปที่ 4.27 ค่าพารามิเตอร์ที่ใช้ควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้า

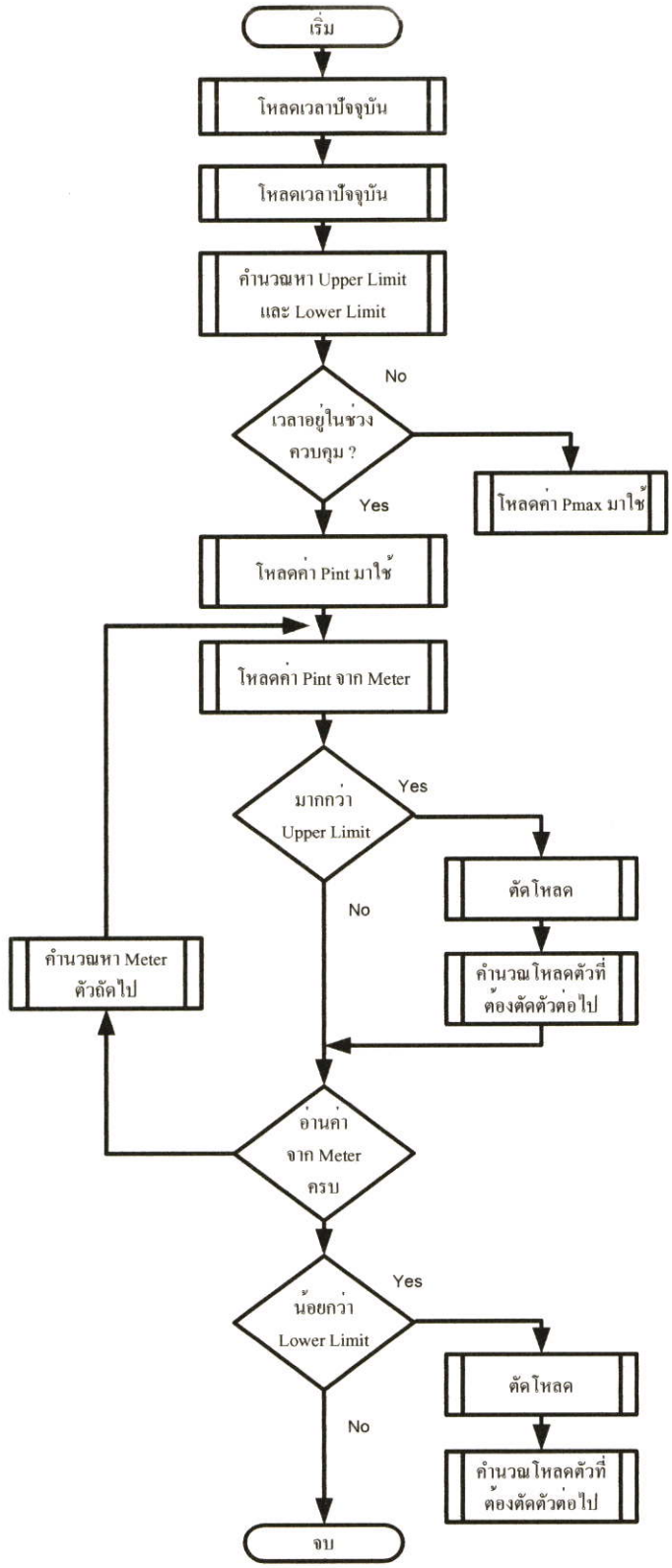
รีเลย์แต่ละตัวสามารถเลือกตั้งค่าการควบคุมได้ว่าจะทำงานในโหมดไหนเมื่อเลือกแล้ว ค่าพารามิเตอร์ของรีเลย์ตัวนั้นจะมีค่าดังรูปที่ 4.28 ในรูปที่ 4.28 (a) เป็นค่าพารามิเตอร์ของรีเลย์ที่ทำงานในโหมดเวลาไบต์แรกจะถูกแบ่งเป็น 2 ส่วน ส่วนแรกจะเก็บข้อมูลการ Active ของรีเลย์ตัวนั้น ๆ ในส่วนหลังใช้เก็บโหมดการทำงาน ส่วนไบต์ที่ 2 จะเก็บค่าตำแหน่งของรีเลย์นั้น ๆ ตั้งแต่ไบต์ที่ 3 – 9 จะเก็บค่าหมายเลขชนิดในโหมดเวลาแต่ละวันของสัปดาห์เป็นจำนวน 7 วัน ซึ่งจะนำค่านี้ไปเปิดตารางข้อมูลในหน่วยความจำเพื่อนำมาควบคุมต่อไป ในรูปที่ 4.28 (b) เป็นค่าพารามิเตอร์ของรีเลย์ที่ทำงานในโหมดเวลาร่วมกับพลังไฟฟ้าในไบต์แรกกับไบต์ที่ 2 จะเก็บค่าเช่นเดียวกับโหมดเวลา ไบต์ที่ 3 จะเก็บหมายเลขกลุ่มที่จะใช้จัดลำดับในการควบคุม ไบต์ที่ 4 เก็บค่ารีเลย์สูงสุดที่มีอยู่ในระบบ ไบต์ที่ 5 เก็บลำดับที่ของกลุ่มที่รีเลย์อยู่ไบต์ที่ 6 กับ 7 เก็บค่าเวลาหน่วงก่อนจะมีการทำงานของรีเลย์ มีหน่วยเป็นนาที ไบต์ที่ 8 กับ 9 เก็บค่าเวลาหน่วงก่อนจะมีการคืนการทำงานของรีเลย์ มีหน่วยเป็นนาทีเช่นกัน



(a) โหมดเวลา (b) โหมดเวลาร่วมกับพลังไฟฟ้า

รูปที่ 4.28 ค่าพารามิเตอร์ที่ใช้ควบคุม (a) โหมดเวลา (b) โหมดเวลาร่วมกับพลังไฟฟ้า

โปรแกรมควบคุมจะอยู่ในส่วนโปรแกรมหลักของ Slave หลังจากอ่านค่าพลังไฟฟ้าเข้ามาแล้วจะอ่านค่าเวลาปัจจุบันจาก RTC ตามเข้ามาเพื่อควบคุมทั้งในโหมดเวลา และโหมดเวลาร่วมกับพลังไฟฟ้า โดยเริ่มจากโหมดเวลาก่อน ด้วยการเช็คค่าพารามิเตอร์ของรีเลย์แต่ละตัวว่ามีการตั้งให้ทำงานไหม และทำงานในโหมดไหน ถ้าเซ็ตให้ทำงานจะมีการโหลดค่าพารามิเตอร์ที่ตั้งไว้ในหน่วยความจำมาใช้ควบคุมตามรูปที่ 4.29



รูปที่ 4.29 โฟวล์ชาร์ทการควบคุมในโหมดเวลาร่วมกับพลังไฟฟ้า

4.7.3 การตัด-ต่อรีเลย์ และคำนวณหารีเลย์ที่ต้องทำงานตัวต่อไป

การคำนวณจะต้องมีการตั้งรีจีสเตอร์ไว้เป็นที่พักข้อมูลในที่นี้จะมีการอ้างชื่อรีจีสเตอร์ต่างๆ ซึ่งหมายถึงรีจีสเตอร์ที่สร้างขึ้นเพื่อเอาไว้พักข้อมูลนั่นเอง มีขั้นตอนการคำนวณดังนี้

1. เริ่มตัดต่อ Master จะต้องอินเทอร์รัพ Slave ก่อนด้วยขอบขาลง และตั้งให้ Slave ตอบสนองอินเทอร์รัพ จากภายนอก เมื่อ Master ถูกอินเทอร์รัพ ภายใน Loop อินเทอร์รัพของ Slave จะเข้าไป SetB Int_Slave เพื่อบอกให้ Main Loop รู้ว่าจะต้องเข้าไปติดต่อกับ Master แล้ว ก่อนที่ Master จะส่งสัญญาณอินเทอร์รัพจะต้องเช็คดูว่าตอนนั้น Slave ว่างอยู่หรือเปล่า โดยเช็คจากขา Busy หาก = 0 แสดงว่าไม่ว่างต้องรอให้ = 1 ก่อนจึงจะอินเทอร์รัพได้

2. เมื่อ Bit Int_Slave ถูก Set ใน Main Loop จะมองเห็นแล้วจะกระโดดเข้าไปเตรียมตัดต่อกับ Master โดยจะ Clr Slave_Reg เพื่อบอกให้ Master รู้ว่าตอนนี้พร้อมที่จะตัดต่อแล้ว

3. เมื่อ Master รู้ว่า Slave พร้อมแล้วก็จะส่ง Start เข้ามา → ข้อมูล → รอรับ Ack จาก Slave → Stop

Note ในการรับของ Slave แต่ละบิตจะมีการเช็คว่าถ้าหาข้อมูล SDA = 0 แล้วในช่วงที่ SCL = 1 จะต้องไม่มีการเปลี่ยนสถานะจาก '0' ไปเป็น '1' ถ้าเปลี่ยนจะถือว่าเป็น Stop ทันที

4. เมื่อ Slave รับเสร็จจะต้อง SetB Slave_Reg เพื่อบอก Master ว่ารับเสร็จแล้วและบอกว่ายังไม่พร้อมจะส่ง Master จะเช็ค Bit นี้เพื่อดูว่า Slave พร้อมที่จะส่งข้อมูลกลับหรือยัง

5. Slave จะProcess คำสั่งจาก Master แล้วเอาข้อมูลไปเก็บใน Buffer เตรียมส่งกลับ

6. ถ้าหาก Process นั้นมีการส่งข้อมูลกลับคืน Master ตอนนี้ Master จะรู้อยู่แล้วว่า Slave จะต้องส่งข้อมูลมา Master จะต้องรอจนกว่า Slave จะ Process เสร็จโดยดูจากขา Slave_Reg ถ้า Slave พร้อมส่งจะต้อง Clr Slave_Reg ก่อน

4.7.4 ขั้นตอนการตรวจสอบการตัดและหา Relay ตัวที่ต้องตัด

ตอนเริ่มเปิดเครื่องให้ (Next_On, Group_On, Next_Off, Group_Off) = 1 (ตัวแรกของกลุ่ม 1)

1. เมื่อ Pint มากกว่า Upper Limit ดูว่า $t_d = 0$ หรือไม่ ถ้า = 0 แล้วตัดตัว Next_On ได้เลย แล้วไปคำนวณหาตัว ตัด-ต่อ ต่อไป ถ้ายังไม่ $t_d \neq 0$ แล้วให้เคลียร์รีจีสเตอร์ ที่เอาไว้นับเวลารอรีจีสเตอร์td_wait_Hi และ td_wait_Low พร้อม SetB Over_Pint ไว้ และบันทึกเวลาเก่า (Old_Time) ไว้ ในขั้นตอนนี้จะต้องรีเซ็ต การรอต่อ t_{ct} ด้วยเพราะค่าพลังไฟฟ้า เพิ่มขึ้นจนถึง Upper Limit แล้ว

2. ถ้าพลังไฟฟ้ามากกว่า Over Limit ซึ่งจะบอกว่าเคยตั้งเวลารอไว้ให้เอา (เวลาปัจจุบัน-เวลาเก่า)=เวลาที่เพิ่มขึ้น (วินาทีเท่านั้น) นำไปบวกกับเวลารอ (td_wait_Hi กับ td_wait_Low) แล้วนำมาเปรียบเทียบกับค่าเวลา t_d จริงว่าเท่ากับ t_d แล้วจึงทำการตัด ถ้ายังไม่เท่าก็ลบออกไป

3. ถ้า t _wait = t_d แล้วก็ตัด Next_On แล้วมาคำนวณหาตัวที่ต้องตัด-ต่อตัวต่อไป

4. การหาตัวตัด-ต่อ ตัวต่อไปต้องดูก่อนว่าต้องควบคุมชนิดไหน FIFO หรือ LIFO

เมื่อเกิดการตัดแล้วให้เพิ่ม Next_On ขึ้นไป 1 ถ้าหากมากกว่าค่าสูงสุดของจำนวนรีเลย์ในกลุ่มแล้วให้เปลี่ยนไปใช้กลุ่มถัดไป หรือ Group_On+1 ส่วนตัวที่ต่อตัวถัดไปให้ดูว่าเป็น FIFO หรือ LIFO

-ในโหมด FIFO ให้ตัวที่จะต่อเป็นตัวแรกของกลุ่มแรกเสมอ

-ในโหมด LIFO ให้ตัวที่จะต่อตัวถัดไปเป็นตัวที่จะตัดตัวถัดไปลบ 1 ถ้า = 0 แล้วให้ทำ (Group_On)-1 จะเป็นตัวสุดท้ายของกลุ่มก่อนหน้านั้น

4.7.5 ขั้นตอนการตรวจสอบการต่อและการหา Relay ตัวที่ต้องต่อ

1. เมื่อค่าพลังไฟฟ้า ต่ำกว่า Lower Limit แล้วต้องดูว่า $t_{ct} = 0$ หรือไม่ถ้า = 0 แล้วตัดตัว Next_Off แล้วไปคำนวณหาตัวตัด-ต่อตัวต่อไป ถ้า $t_{ct} = 0$ แล้วให้ตั้งรีจีสเตอร์ไว้นับเวลารอ td_wait_Hi, td_wait_Low ใช้ตัวเดียวกันกับตัวตัดเลย พร้อม SetB Under_Pint ไว้ และบันทึกค่าเวลาเก่า (Old_time) ไว้ในขั้นตอนนี้จะต้องรีเซ็ตการรอตัดค่านี้ เพราะค่าพลังไฟฟ้าได้ลดลงต่ำกว่า Lower Limit แล้ว

2. ถ้าค่าพลังไฟฟ้ายังต่ำกว่า Lower Limit ในรอบถัดไปก็ให้เอาเวลา (เวลาปัจจุบัน – เวลาเก่า) = เวลาที่เพิ่มขึ้น (วินาที) นำไปบวกกับเวลา (td_wait_Hi, td_wait_Low) แล้วนำมาเปรียบเทียบกับค่าเวลา t_{ct} จริงถ้ามากกว่าหรือเท่ากับ t_{ct} แล้วจึงทำงานต่อถ้ายังไม่เท่าให้ลบออกไป

3. ถ้า t _wait = t_{ct} แล้วก็ต่อตัว Next_Off แล้วมาคำนวณหาตัวที่ต้องตัด-ต่อตัวต่อไป

4. การคำนวณค่าลำดับรีเลย์ใหม่หลังจากการต่อ

- ในโหมด FIFO

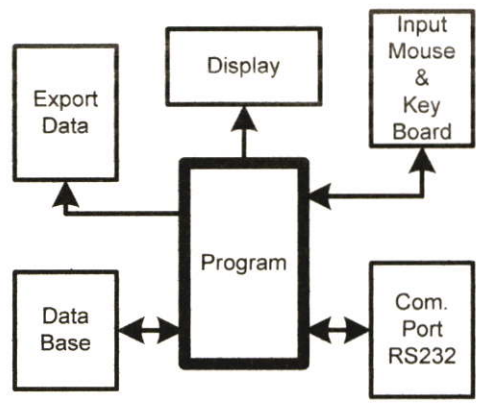
ครั้งแรกที่เริ่ม หรือหลังจากการ ตัด-ต่อ จนครบและวน Loop Group แล้วต้องให้ Next_Off = 1 ของกลุ่มสม่ำเสมอ แล้วถ้ามีการต่อแล้วก็ค่อยๆ เพิ่มขึ้นเรื่อยๆ เหมือนการคำนวณ FIFO ของการตัด-ต่อ

- ในโหมด LIFO

เหมือนกับ LIFO ของแบบการตัด

4.8 การออกแบบโปรแกรมคอมพิวเตอร์

ในวิทยานิพนธ์ฉบับนี้ใช้โปรแกรม Delphi7 เขียนติดต่อกับระบบภายนอก แสดงผลข้อมูล และติดต่อแลกเปลี่ยนข้อมูลกับดาต้าเบส (Database) ชนิดพาราด็อกซ์7 (Paradox7) ซึ่งเป็นฐานข้อมูลชนิดหนึ่งที่มีมากับโปรแกรม Delphi โครงสร้างโปรแกรมดังรูปที่ 4.30



รูปที่ 4.30 บล็อคไดอะแกรม ของโปรแกรมคอมพิวเตอร์

4.8.1 โปรแกรมติดต่อสื่อสารกับโมเด็ม

บางครั้งระบบต้องการสื่อสารข้อมูลระยะไกลเกินกว่าที่การสื่อสารข้อมูลภายในไม่สามารถรองรับได้จำเป็นต้องใช้ระบบโทรศัพท์เข้ามาช่วยเป็นตัวกลางสื่อสารข้อมูล เพื่อกำจัดข้อกำหนดของระยะทางในการสื่อสารข้อมูล ขั้นตอนการใช้งานโมเด็มก็ไม่ได้ยุ่งยาก มีขั้นตอนดังนี้

1. โปรแกรมโมเด็มเพื่อกำหนดให้โมเด็มมีการทำงานตามความต้องการ การโปรแกรมโมเด็มจะกระทำเพียงครั้งเดียวหลังจากนั้นโมเด็มจะเก็บค่าที่โปรแกรมไว้ในหน่วยความจำเพื่อใช้งานต่อไป

2. สั่งให้โมเด็มโทรออกไปยังหมายเลขปลายทาง

3. รอสัญญาณตอบรับ (Carrier) จากปลายทางเพื่อบอกว่ามีการรับสายจากหมายเลขที่ติดต่อไป ถ้าไม่มีการตอบรับโมเด็มจะส่งสัญญาณคู่สายไม่ว่าง (Busy)

4. ถ้ามีการรับสายจากหมายเลขปลายทางก็ทำการสื่อสารข้อมูล แต่ถ้าคู่สายไม่ว่างหรือไม่รับสายก็ให้ต่อใหม่เป็นจำนวนครั้งที่กำหนด หรือวางสายแล้วแต่ผู้ใช้กำหนด

4.8.2 โปรแกรมติดต่อสื่อสารกับ โมดูลควบคุม และบันทึกความต้องการพลังไฟฟ้า (Demand Controller Module)

โปรแกรมสามารถสื่อสารกับ DCU Module ได้ 3 ทาง คือ RS232C, RS485 และผ่านโมเด็ม ถ้าสื่อสารผ่านอนุกรม RS232C ไม่ต้องมีอุปกรณ์แปลงสัญญาณเพราะที่คอมพิวเตอร์มีพอร์ท RS232C อยู่แล้วสามารถต่อออกไปใช้งานได้เลย แต่ถ้าสื่อสารผ่านอนุกรม RS485 จำเป็นต้องใช้ตัวแปลงอนุกรม RS232C ให้เป็น RS485 ก่อน ปกติ RS232C to RS485 Converter จะมีฟังก์การควบคุมการกำหนดทิศทางการไหลของข้อมูลได้ 2 ชนิดคือ แบบอัตโนมัติ และใช้สัญญาณภายนอกกำหนด การกำหนดด้วยการใช้สัญญาณภายนอกจะทำให้สามารถสื่อสารด้วยอัตราเร็วมากกว่า แบบอัตโนมัติเพราะไม่ต้องเสียเวลาตรวจสัญญาณป้อนกลับเพื่อเช็คว่าขณะนี้สัญญาณมีทิศทางใด เพื่อให้ได้ความเร็วสูงโปรแกรมที่ต้องการสื่อสารผ่านอนุกรม RS485 จึงใช้ขาสัญญาณ RST เป็นสัญญาณกำหนดทิศทางการไหลของข้อมูล ส่วนโมเด็มก็ใช้ขาสัญญาณช่วยต่างๆในการเช็คสถานะและควบคุมการโทรออก-วางสาย

4.8.3 โปรแกรมติดต่อสื่อสารกับโมดูลตรวจเช็คบันทึกสถานะ (Input Module)

ตัวโมดูลตัวตรวจเช็คบันทึกสถานะสามารถติดต่อสื่อสารได้ 3 ทาง คือ RS232C, RS485 และ ผ่านโมเด็ม แตกต่างกันตรงโปโตคอลการติดต่อเนื่องจาก โมดูลตัวตรวจเช็คบันทึกสถานะต้องการคำสั่งที่แตกต่างจากโมดูลอื่น ลักษณะคำสั่งสามารถดูได้จากการออกแบบ Input Module

4.8.4 โปรแกรมติดต่อสื่อสารกับโมดูลรีเลย์ควบคุม (Relay Module)

ตัวโมดูลรีเลย์ควบคุมก็สามารถติดต่อสื่อสารได้ 3 ทาง คือ RS232C, RS485 และ ผ่านโมเด็ม เหมือนโมดูลอื่นๆ แต่โปโตคอลจะเป็นลักษณะเกี่ยวกับการควบคุมเกือบทั้งหมด ดังนั้นข้อมูลที่ได้รับจากโมดูลรีเลย์ควบคุมจะเป็นข้อมูลตอบกลับเพื่อยืนยันคำสั่ง การควบคุมรีเลย์โมดูลโดยตรงนั้นจะกระทำได้นอกเหนือจากการควบคุมของตัวควบคุมค่าความต้องการพลังไฟฟ้าในกรณีที่ต้องการควบคุมพิเศษ

4.8.5 โปรแกรมฐานข้อมูล (Database)

โปรแกรมฐานที่ใช้เป็นฐานข้อมูลระดับ Desktop ใช้ฐานข้อมูลชนิด พาราด็อกซ์7 (Paradox7) ตัวโปรแกรมติดต่อกับฐานข้อมูลผ่านเครื่องมือช่วย BDE (Borland Database Engine) ซึ่งตัว BDE ทำหน้าที่ติดต่อระหว่างโปรแกรม และจัดการกับฐานข้อมูล

4.8.6 โปรแกรมแสดงผล (Monitor)

มีส่วนติดต่อกับผู้ใช้งาน การแสดงผลมี 2 ชนิด คือ ตัวหนังสือ (Text) และ กราฟ (Graphic)

4.8.7 โปรแกรมคำนวณ และวิเคราะห์ข้อมูล

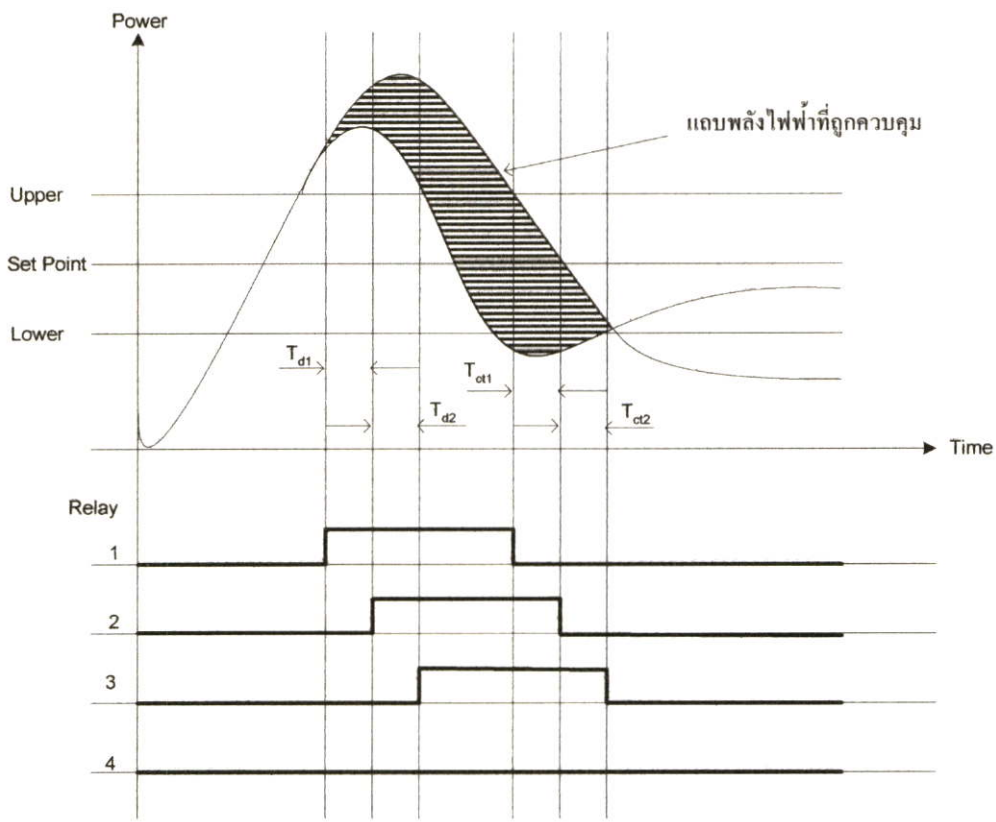
ส่วนการคำนวณเป็นส่วนช่วยเหลือผู้ใช้งานให้สะดวกสบายในการขอข้อมูล แบ่งเป็น

1. การคำนวณ ค่าไฟฟ้าในอัตรา TOU และ TOD
2. การคำนวณค่าทางไฟฟ้าจากข้อมูลที่ส่งมาไปสู่ค่าอื่นๆ
3. การคำนวณเพื่อหาข้อมูลที่ใช้ในการวิเคราะห์ระบบ
4. การทำนายโหลด

บทที่ 5

การทดลอง

การทดลองทั้งหมดมีหลายส่วน ตั้งแต่ทดลองติดต่อสื่อสารกันด้วยรูปแบบ (Protocol) ต่างๆ การทดลองในขณะออกแบบวงจร การทดลองควบคุมโหลดตามค่าความต้องการพลังงานไฟฟ้าที่กำหนด ในช่วงเริ่มวัดค่าความต้องการพลังงานไฟฟ้า และควบคุมต่อเนื่องจากการวัดก่อนหน้านั้น โดยมีรูปแบบการควบคุมดังนี้



รูปที่ 5.1 แสดงการควบคุมความต้องการพลังงานไฟฟ้าที่มีการควบคุมชนิด FIFO

จากรูปที่ 5.1 มีการตั้งค่าควบคุมความต้องการพลังงานไฟฟ้าไว้ ณ. จุดตั้งค่าควบคุม (Set Point) แต่การควบคุมรีเลย์นั้นจะควบคุมด้วยแถบฮีสเตอร์รีซีส (Hysteresis Band) คือ เริ่มเมื่อค่าความต้องการพลังงานไฟฟ้าสูง หรือ ต่ำกว่าแถบควบคุม (Upper และ Lower) บวกด้วยค่าเวลาหน่วงเพื่อป้องกันไม่ให้มีการทำงานกลับไปมาในช่วงเวลาใกล้กัน T_d และT_{ct} เมื่อค่าความต้องการพลังงานไฟฟ้าสูงกว่า Upper Limit รีเลย์จะยังไม่ทำงาน จนกระทั่งเวลาผ่านไปเป็นเวลา T_{d1} แล้วรีเลย์ตัวแรกจะทำงานเพื่อลดโหลดลง แต่ค่าความต้องการพลังงานไฟฟ้ายังพุ่งขึ้นสูงกว่า Upper

Limit อีก จนกระทั้งเวลาผ่านไปเป็นเวลา T_{d2} รีเลย์ตัวที่ 2 จึงทำงานเมื่อเวลาผ่านไปเป็น T_{d3} ค่าความต้องการพลังงานไฟฟ้ายังสูงกว่า Upper Limit อีก รีเลย์ตัวที่ 3 จึงทำงาน เมื่อโหลดลดลงต่ำกว่า Lower Limit จะยังไม่มีการคืนโหลดในทันที จะต้องรอเวลาเป็นเวลา T_{ct1} ก่อน จึงจะเริ่มต่อโหลดที่ 1 คืนถ้าค่าความต้องการพลังงานไฟฟ้ายังต่ำกว่าค่า Lower Limit อีก และเวลาผ่านไปเป็นเวลา T_{ct2} รีเลย์ตัวที่ 2 ก็จะคืนโหลด ถ้ายังต่ำกว่า Lower Limit อีกเป็นเวลา T_{ct3} รีเลย์ตัวที่ 3 จึงคืนโหลด

5.1 ทดลองการติดต่อสื่อสาร

การสื่อสารข้อมูลที่ถูกต้องส่งผลให้ควบคุมได้ถูกต้อง ถ้าหากการสื่อสารล้มเหลว หรือผิดพลาดข้อมูลที่ใช้ในการควบคุมก็เป็นข้อมูลที่ผิด ทำให้ควบคุมผิดพลาดตามไปด้วย จึงจำเป็นต้องทบสอบระบบสื่อสารซ้ำๆกันหลายรอบเพื่อให้แน่ใจว่าเมื่อนำไปใช้งานจริงจะไม่เกิดข้อผิดพลาดขึ้น หรือถ้าเกิดผิดพลาดก็ต้องมีระบบป้องกันรองรับเพื่อให้ผู้ใช้งานมีความมั่นใจในระบบ การทดลองจะใช้การส่งข้อมูลที่มีความยาวต่างกัน 3 ชนิด คือ 5 ไบต์, 10 ไบต์ และ 50 ไบต์ จำนวน 50 ครั้ง สายข้อมูลยาว 2 เมตร และตรวจสอบดูว่าเกิด Error ขึ้นเป็นกี่เปอร์เซ็นต์

5.1.1 ทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS232C

การทดลองนี้เป็นการทดสอบความถูกต้องในการรับ-ส่งข้อมูลผ่านอนุกรม RS232C เพื่อหาค่าที่ดีที่สุดนำไปใช้ โดยทดลองส่งข้อมูลระหว่างคอมพิวเตอร์กับ Input Module

ตารางที่ 5.1 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS232C จำนวน 50 ครั้ง

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	1	10
50	7	14

ตารางที่ 5.2 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS232C จำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	0	0
50	0	0

จากตารางที่ 5.1 ข้อมูลที่รับได้มีค่า Error สูงขึ้นเมื่อมีจำนวนข้อมูลที่ใช้ส่งเพิ่มขึ้น สามารถแก้ไขได้โดยการขอทวนสัญญาณอีก ถ้าหากเกิด Error ขึ้น จะช่วยให้ข้อมูลที่ได้มีความถูกต้อง 100 % ตามตารางที่ 5.2 อย่างไรก็ตามค่าความถูกต้องก็ขึ้นอยู่กับสายข้อมูล และสภาพแวดล้อมด้วย

5.1.2 ทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS458

เนื่องจากอนุกรม RS485 เป็นการสื่อสารชนิดสื่อสารทางเดียว (Half Duplex) ค่าเวลาหน่วงก่อนการกลับทิศทางข้อมูลมีผลต่อความถูกต้องของข้อมูล การทดลองจะใช้การส่งข้อมูลที่มีความยาวต่างกัน 3 ชนิด คือ 5 ไบต์, 10 ไบต์ และ 50 ไบต์ จำนวน 50 ครั้ง สายข้อมูลยาว 10 เมตร เวลาหน่วง = 0 mS. และ 0 mS . แล้วตรวจสอบดูว่าเกิด Error ขึ้นเป็นกี่เปอร์เซ็นต์

ตารางที่ 5.3 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS485 จำนวน 50 ครั้ง เวลาหน่วง 0 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	50	100
10	50	100
50	50	100

ตารางที่ 5.4 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS485 จำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง เวลาหน่วง 0 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	50	100
10	50	100
50	50	100

ตารางที่ 5.5 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS485 จำนวน 50 ครั้ง เวลาหน่วง 10 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	5	10
50	8	16

ตารางที่ 5.6 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม RS485 จำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง เวลาหน่วง 10 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	0	0
50	0	0

จากตารางที่ 5.3 และตารางที่ 5.4 ถ้าเวลาหน่วง = 0 mS. จะทำให้ตัวส่งเปลี่ยนสถานะตัวเองจากผู้ส่งไปเป็นผู้รับไม่ทันเมื่อมีข้อมูลตอบรับกลับมาก็ไม่สามารถรับได้ถูกต้อง ถึงแม้ว่าจะมีการทวนการร้องขอแล้วก็ตาม ส่วนตารางที่ 5.5 มีการหน่วงเวลา = 10 mS. ทำให้มีเวลาพอที่จะสามารถเปลี่ยนสภาวะจากผู้ส่งเป็นผู้รับได้ทัน จึงสามารถรับข้อมูลได้แต่ก็ไม่ถูกต้องทั้งหมด จึงต้องเพิ่มการส่งซ้ำถ้าข้อมูล Error จำนวน 3 ครั้งเข้าไปดังตารางที่ 5.6 ทำให้สามารถรับข้อมูลได้ถูกต้อง 100 %

5.1.3 การทดลองรับ-ส่งข้อมูลผ่านโมเด็ม

การทดลองสื่อสารผ่านโมเด็มใช้ ตู้ชุมสายโทรศัพท์ภายใน (PABX) ในการทดลองเพื่อสร้างสัญญาณโทรศัพท์ให้กับโมเด็ม ผลการทดลองที่ได้เป็นผลการทดลองรับข้อมูล

ตารางที่ 5.7 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็มจำนวน 50 ครั้ง เวลาหน่วง 0 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	2	4
10	9	18
50	32	64

ตารางที่ 5.8 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็มจำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง เวลาหน่วง 0 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	1	2
50	13	26

ตารางที่ 5.9 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็ม จำนวน 50 ครั้ง เวลาหน่วง 10 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	0	0
50	2	4

ตารางที่ 5.10 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็ม จำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง เวลาหน่วง 10 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	1	2
50	3	6

ตารางที่ 5.11 ผลการทดลองรับ-ส่งข้อมูลผ่านโมเด็มจำนวน 50 ครั้งมีการส่งซ้ำ 3 ครั้ง เวลาหน่วง 20 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	0	0
50	0	0

จากตารางที่ 5.7 และตารางที่ 5.8 ยังมีข้อมูลที่รับได้ถูกต้องผ่านโมเด็มเพราะว่าข้อมูลที่ส่งเข้าโมเด็มจะต้องหน่วงเวลาออกไปเนื่องจากกระบวนการแปลงของโมเด็ม ถึงแม้ว่าโปรแกรมไม่มีเวลาหน่วง ก็ยังสามารถรับข้อมูลได้เนื่องจากเวลาหน่วงเกิดขึ้นที่โมเด็ม แต่ค่าความถูกต้องยังไม่เป็นที่น่าพอใจ เพราะในการใช้งานจริงต้องไม่มีการสื่อสารที่ผิดพลาดเลย จากตารางที่ 5.9 และ เมื่อเพิ่มเวลาหน่วงก็ทำให้ค่าความถูกต้องเพิ่มขึ้น แต่ก็ยังมีผิดพลาดอยู่บ้าง ตารางที่ 5.11 ปรับเพิ่มเวลาหน่วงเป็น 20 mS. และมีการทวนสัญญาณ 3 ครั้งหากผิดพลาดทำให้ได้รับข้อมูลที่ไม่ผิดพลาดเลย

5.1.4 ทดลองรับ-ส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์ผ่านอนุกรม I^2C

ทดสอบรับ-ส่งข้อมูลบนบอร์ด PCB ที่มีระยะไม่เกิน 1 ฟุต ป้อนข้อมูลเหมือนกับการทดสอบอนุกรม RS 232C แต่ปรับค่าเวลาหน่วงลง

ตารางที่ 5.12 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม I^2C จำนวน 50 ครั้ง เวลาหน่วง 20 uS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	0	0
50	0	0

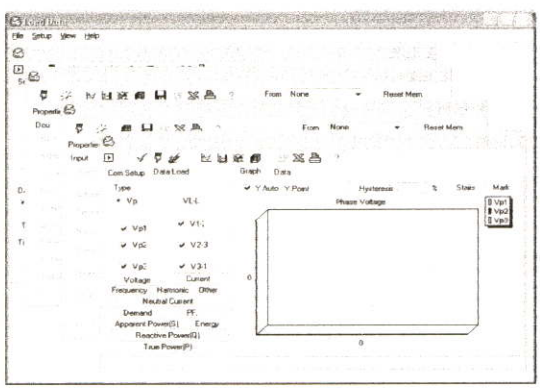
ตารางที่ 5.13 ผลการทดลองรับ-ส่งข้อมูลผ่านอนุกรม I^2C จำนวน 50 ครั้ง เวลาหน่วง 1 mS.

จำนวนไบต์ข้อมูล	ผิดพลาด (ครั้ง)	Error (%)
5	0	0
10	0	0
50	0	0

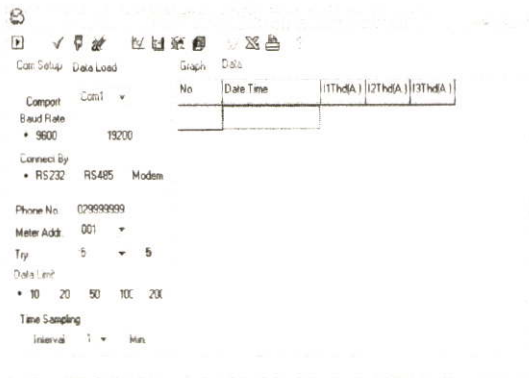
จากตารางที่ 5.12 และ ตารางที่ 5.13 แสดงว่าการสื่อสารข้อมูลผ่านอนุกรม I^2C มีค่าความถูกต้องสูงเนื่องจาก การส่งข้อมูลมีลักษณะซิงโครไนซ์ (Synchronize) ตามขาสัญญาณเวลา

5.2 การอ่านค่าต่าง ๆ ทางไฟฟ้ามาจัดเก็บและแสดงผล

โปรแกรมสามารถอ่านข้อมูลจากโมดูลต่างๆมาเพื่อจัดเก็บหรือ แสดงผลได้โดยตรง รูปที่ 5.2 และ รูปที่ 5.3 เป็นการดึงข้อมูลจากโมดูลต่างๆขึ้นมาแสดงผลโดยตรง

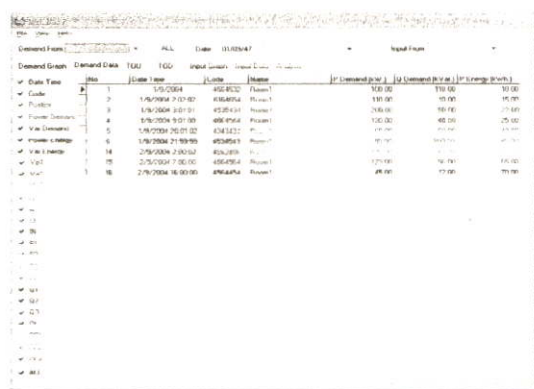


รูปที่ 5.2 ฟอร์มขอโหลดค่าจากโมดูลภายนอก

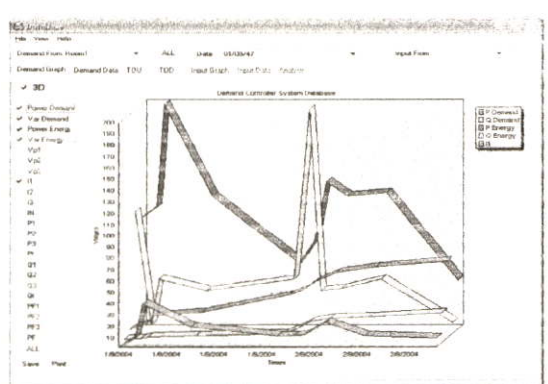


รูปที่ 5.3 ฟอร์มขอโหลดค่าจากเพาเวอร์มิเตอร์ A2000

รูปที่ 5.4 และ รูปที่ 5.5 เป็นการดึงข้อมูลจากโมดูลควบคุมและบันทึกความต้องการพลังไฟฟ้ามาเก็บไว้ที่ ดาต้าเบส แล้วเรียกข้อมูลจากดาต้าเบสที่จัดเก็บไว้มาแสดงผลในรูปแบบของมูล และกราฟ ตามวันเวลาที่กำหนดให้แสดงข้อมูล

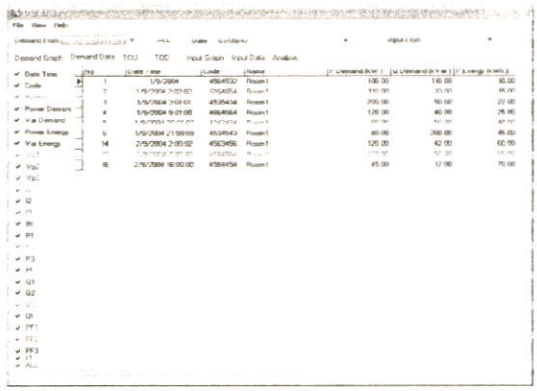


รูปที่ 5.4 ฟอร์มแสดงข้อมูลทางไฟฟ้าที่เก็บไว้ในดาต้าเบส

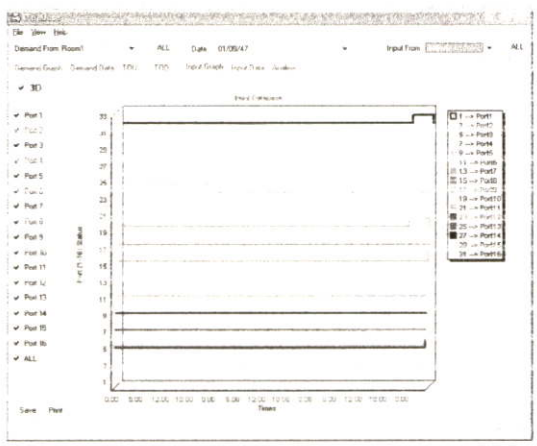


รูปที่ 5.5 ฟอร์มแสดงกราฟข้อมูลทางไฟฟ้าที่เก็บไว้ในดาต้าเบส

รูปที่ 5.6 และ รูปที่ 5.7 เป็นการดึงข้อมูลจากโมดูลอินพุทมาเก็บไว้ที่ดาต้าเบส แล้วเรียกข้อมูลจากดาต้าเบสที่จัดเก็บไว้มาแสดงผลในรูปแบบของข้อมูล และกราฟตามวันเวลาที่กำหนดให้แสดงข้อมูล

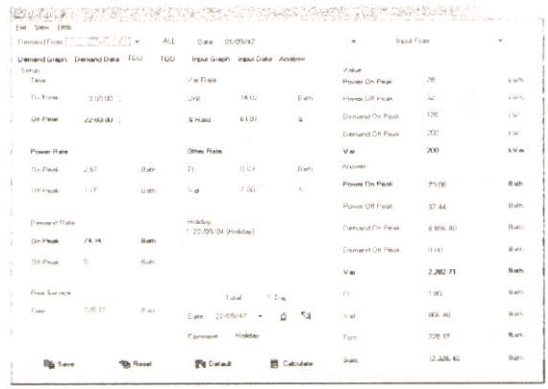


รูปที่ 5.6 ฟอร์มแสดงข้อมูลจากโมดูลตัวตรวจเช็คบันทึกสถานะที่เก็บไว้ในดาต้าเบส

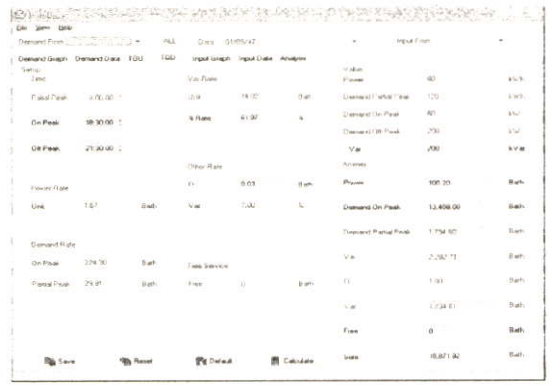


รูปที่ 5.7 ฟอร์มแสดงกราฟข้อมูลจากโมดูลตัวตรวจเช็คบันทึกสถานะที่เก็บไว้ในดาต้าเบส

รูปที่ 5.8 และรูปที่ 5.9 เป็นการดึงข้อมูลจากดาต้าเบสมาคำนวณหาค่าไฟฟ้าในอัตรา TOU และ TOD ตามวันเวลาที่กำหนดให้แสดงข้อมูล



รูปที่ 5.8 ฟอร์มคำนวณค่าไฟฟ้าอัตรา TOU



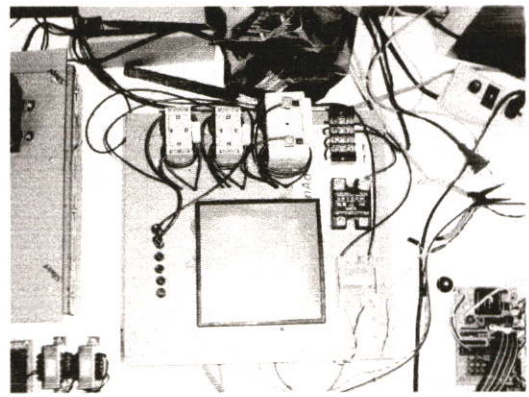
รูปที่ 5.9 ฟอร์มคำนวณค่าไฟฟ้าอัตรา TOD

5.3 การควบคุมความต้องการพลังไฟฟ้า

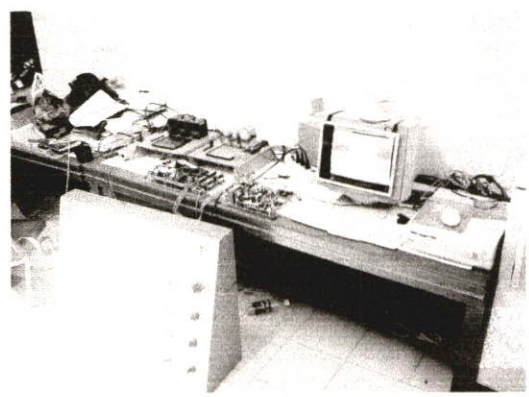
การทดลองควบคุมค่าความต้องการพลังไฟฟ้าแบ่งทดสอบเป็น 2 ช่วง ในช่วงแรกจะทดสอบโดยการจำลองโหลด และควบคุมแผงหลอดไฟ ในช่วงที่ 2 จะทดลองกับโหลดจริงที่ห้องทดลองของภาควิชาวิศวกรรมไฟฟ้า ห้อง ECC405 โดยมีการเพิ่มโหลดที่สามารถควบคุมได้เข้าไปในระบบเพื่อทำการควบคุม

5.3.1 การทดลองเบื้องต้นกับโหลดหลอดไฟ

จากรูปที่ 5.1 มีการตั้งค่าควบคุมความต้องการพลังไฟฟ้าไว้ ณ จุดตั้งค่าควบคุม (Set Point) แต่การควบคุมรีเลย์ นั้นจะควบคุมด้วยแถบฮีสเตอร์รีซีส (Hysteresis Band) คือ เริ่มทำงานเมื่อค่าความต้องการพลังไฟฟ้าสูงหรือต่ำกว่าแถบควบคุม Upper และ Lower บวกด้วยค่าเวลาหน่วงเพื่อป้องกันไม่ให้มีการทำงานกลับไปมาในช่วงเวลาใกล้กัน T_d และT_{ct}



รูปที่ 5.10 เพาเวอร์มิเตอร์ รุ่น A2000 ที่ใช้ทดสอบ



รูปที่ 5.11 ทดสอบควบคุมค่าความต้องการพลังไฟฟ้า

ตารางที่ 5.14 แสดงการทดลองควบคุมความต้องการพลังงานที่ค่าความต้องการพลังไฟฟ้าเริ่มต้น 0 W.

Time	Demand	Relay1	Relay2	Relay3	Relay4
0	0	ON	ON	ON	ON
5	134	ON	ON	ON	ON
7	172	OFF	ON	ON	ON
8	190	OFF	OFF	ON	ON
9	210	OFF	OFF	OFF	OFF
15	210	OFF	OFF	OFF	OFF

จากตารางที่ 5.14 ทดลองโหลดเริ่มต้น 400 W. รีเลย์แต่ละตัวควบคุมโหลด 100 W. ตั้งค่าความต้องการพลังไฟฟ้า ณ จุดควบคุมไว้ที่ 150 W. ค่าฮีสเตอร์รีซีส 10% ค่าเวลาหน่วง 1 นาทีทั้งตัด

และต่อ รีเลย์จะเริ่มตัดโหลดหลังจากค่าความต้องการพลังไฟฟ้าสูงกว่า 165 W. ไปแล้ว 1 นาที ทำให้ค่าความต้องการพลังไฟฟ้าสูงเกินแถบควบคุมไปแล้ว 1 นาที ก่อนเริ่มตัด และจะเริ่มต่อโหลดหลังจากค่าความต้องการพลังไฟฟ้าต่ำกว่า 135 W. และเกินเวลา 1 นาที หลังจากการตัดหรือต่อครั้งสุดท้ายไปแล้ว จะเริ่มต่อตัวถัดไปหลังจากนั้นอีก 1 นาที จากตารางที่ 5.14 ค่าความต้องการพลังไฟฟ้าจะยังคงเพิ่มขึ้นเรื่อยๆในลักษณะบวกสะสม เมื่อมีการตัดโหลดอัตราเร็วในการเพิ่มขึ้นของค่าความต้องการพลังไฟฟ้าสะสมสูงสุดก็ลดลง ในเวลาที่ 9 นาที มีการตัดโหลดจนหมดแต่ค่าความต้องการพลังไฟฟ้าสูงสุดไม่ลดลงเพราะเป็นผลบวกสะสมในช่วงเริ่มต้นต้องบวกให้ครบ 15 นาที ถึงแม้ว่าจะไม่มีการใช้พลังงานก็ตาม ดังนั้นการตั้งค่าควบคุมจะต้องคำนึงถึงจุดนี้ด้วย เพราะอาจจะทำให้เกิดการตัดของรีเลย์ทุกตัว

ตารางที่ 5.15 ผลการทดลองควบคุมความต้องการพลังงานที่ค่าความต้องการพลังไฟฟ้าเริ่มต้น 200 W.

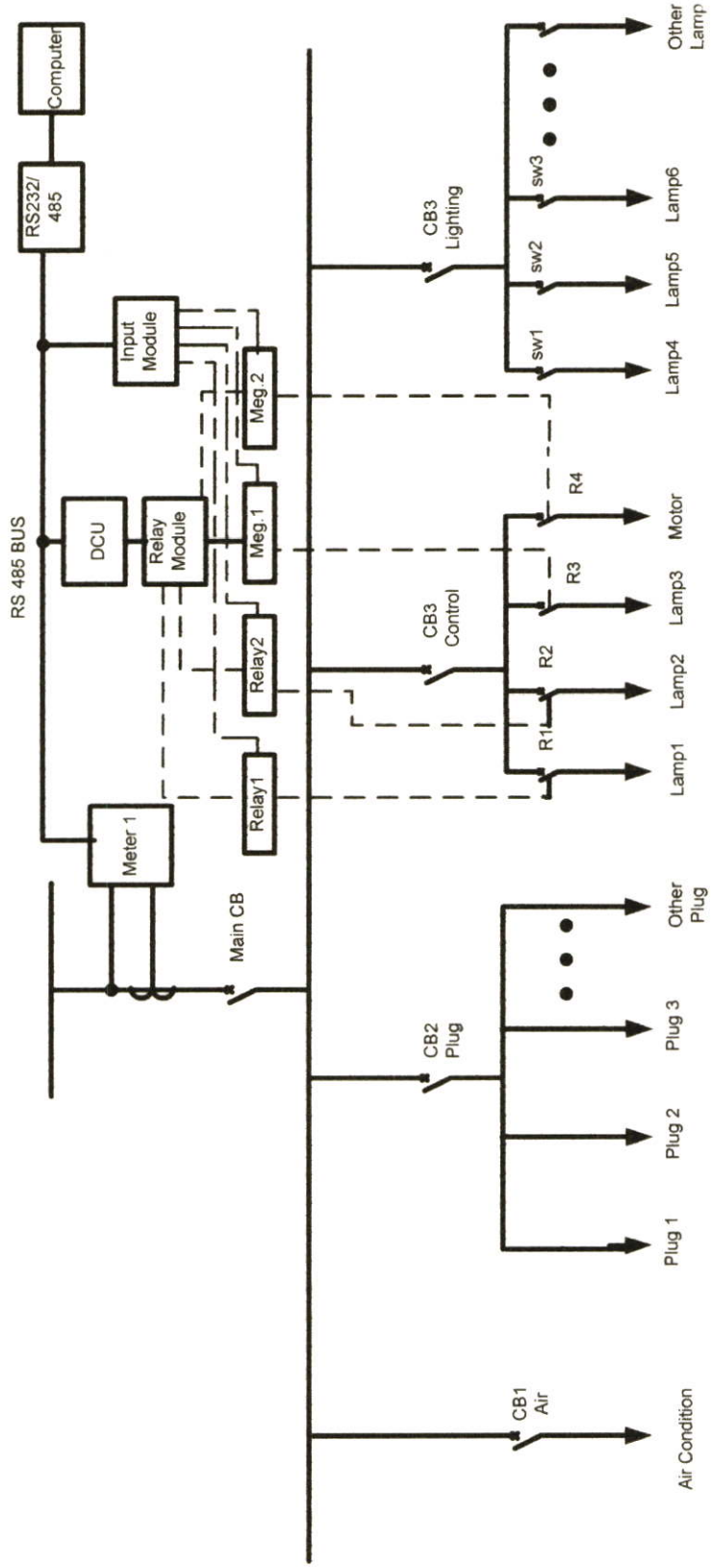
Time	Demand	Relay1	Relay2	Relay3	Relay4	
X	200	ON	ON	ON	ON	
X+5	261	OFF	ON	ON	ON	←
X+6	261	OFF	OFF	ON	ON	
X+7	254	OFF	OFF	OFF	ON	
X+8	240	OFF	OFF	OFF	OFF	
X+12	193	ON	OFF	OFF	OFF	←
X+13	193	ON	ON	OFF	OFF	
X+14	200	ON	ON	ON	OFF	
X+15	213	ON	ON	ON	ON	←

จากตารางที่ 5.15 โหลดเริ่มต้น 400 W. รีเลย์แต่ละตัวควบคุมโหลด 100 W. ตั้งค่าความต้องการพลังไฟฟ้า ณ จุดควบคุมไว้ที่ 230 W. ค่าฮีสเตอร์รีซีส 10% ค่าเวลาหน่วง 1 นาที ทั้งตัดและต่อ กรณีนี้เป็นกรณีที่มีการเปิดใช้งานมาแล้วมากกว่า 15 นาที ทำให้มีค่าความต้องการพลังไฟฟ้าสูงสุดก่อนหน้านั้นคงอยู่เท่ากับ 200 W. ค่าความต้องการพลังไฟฟ้าต่อไปนี้จะคิดจากการบวกค่าพลังไฟฟ้าใหม่ พร้อมทั้งลบค่าเก่าสุดออก แล้วหารด้วยเวลา 15 นาที รีเลย์จะเริ่มตัดโหลดหลังจากค่าความต้องการพลังไฟฟ้าสูงกว่า 253 W. ไปแล้ว 1 นาที (ที่เวลา X+5) และจะเริ่มต่อโหลดหลังจากค่าความต้องการพลังไฟฟ้าต่ำกว่า 207 W. หลังจากมีการตัดหรือต่อเกิน 1 นาทีไปแล้ว (ที่เวลา X+12) จะเห็นว่าตัวควบคุมพยายามรักษาค่าความต้องการพลังไฟฟ้าให้อยู่ภายในแถบฮีสเตอร์รีซีส การเลือกขนาดโหลดให้รีเลย์ควบคุมมีผลต่อการเข้าสู่สภาวะคงตัวของการควบคุม (ที่เวลา

X+12) หากไม่เหมาะสมแล้วจะเกิดการแกว่ง เช่น ตัดและต่อในเวลาต่อมาซ้ำไปเรื่อยๆ ซึ่งวิธีการเลือกขนาด และการจัดการกับโหลดนี้อยู่ในเรื่องการวิเคราะห์โหลด [2] ผู้เขียนจะพัฒนาต่อไป

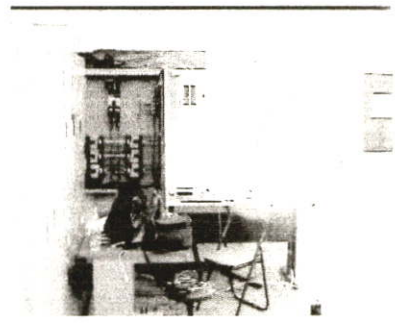
5.3.2 การทดลองกับโหลดจริง

ในการทดลองกับโหลดจริงนี้ได้ทำการทดลองกับห้องทดลองของภาควิชาวิศวกรรมไฟฟ้า ECC405 เริ่มจากการตรวจสอบโหลดจัดทำรายการอุปกรณ์ไฟฟ้าที่มีอยู่ทั้งหมดเพื่อหาโหลดที่ต่ออยู่ทั้งหมด (Connect Load) จัดแบ่งโหลดออกเป็น 4 หมวดใหญ่ๆคือ เครื่องปรับอากาศ, ปลั๊ก, แสงสว่าง และโหลดที่สามารถควบคุมได้ โดยโหลดที่สามารถตัดได้นั้นได้จัดเพิ่มเข้ามาในระบบ จำนวน 4 ชุด เป็นโหลดหลอดไฟ ขนาด 1000 W. จำนวน 3 ชุด และมอเตอร์ 7.5 kW. ที่ทำงานแบบไม่มีโหลดอีก 1 ชุด เพื่อใช้ทดสอบการควบคุม ใช้มิเตอร์ 1 ตัว วัดค่าพลังไฟฟ้าปัจจุบันที่จุดจ่ายไฟหลัก ส่งข้อมูลให้กับตัวควบคุม DCU เพื่อคำนวณควบคุมโหลด โดยส่งข้อมูลควบคุมไปยังชุดรีเลย์โมดูล เพื่อควบคุมโหลดทั้ง 4 ชุด ใช้อุปกรณ์ตัดต่อเป็นโซลิดสเตทรีเลย์ 2 ตัว และแมกเนติก อีก 2 ตัว จุดที่ควบคุมมีการตรวจจับสัญญาณป้อนให้กับอุปกรณ์บันทึกสถานะอินพุท เพื่อเก็บข้อมูลไว้ตรวจเช็คว่ามีการตัด-ต่อที่โหลดตัวไหนเวลาใดบ้าง ดังในรูปที่ 5.12 ก่อนการโปรแกรมการควบคุมต้องพิจารณาว่าควรจัดให้อุปกรณ์ตัวไหนตัดก่อนถ้าค่าความต้องการพลังไฟฟ้าสูงกว่าค่าที่กำหนด เมื่อได้รายละเอียดโหลดมาแล้วก็ติดตั้งอุปกรณ์วัด และควบคุม



รูปที่ 5.12 One Line Diagram ของการทดลองควบคุมความสว่างของการหลอดไฟฟ้า

(a) ชุดโหลดที่เพิ่มเข้ามา



(b) การวัดค่าพลังไฟฟ้า

รูปที่ 5.13 (a-b) การทดลองควบคุมค่าความต้องการพลังไฟฟ้า

ตารางที่ 5.16 รายการสำรวจโหลดที่ต่ออยู่ในระบบทั้งหมด

ลำดับที่	รายการ	จำนวน	กำลังไฟฟ้า (W.)	รวมกำลังไฟฟ้า (W.)
1	หลอดฟลูออเรสเซ็นต์	36	70	2520
2	คอมพิวเตอร์	500	9	4500
3	ตู้เย็น	160	1	160
4	เครื่องปรับอากาศ	3520	4	14080
5	พรินเตอร์	2	20	40
6	อุกรณ์ทดลอง	10	500	1000
7	โอสซิลโลสโคป	5	300	1500
8	โหลดหลอดไฟที่เพิ่มเข้ามา	3	1000	3000
9	โหลดมอเตอร์ที่เพิ่มเข้ามา	1	2000	2000
	รวม			28800

หลังจากติดตั้งแล้วจะต้องตรวจสอบกำลังไฟฟ้าที่ใช้จริงของระบบก่อน เพื่อนำไปคำนวณหาค่าความต้องการพลังไฟฟ้าที่เหมาะสมที่จะนำไปใช้ควบคุม การตรวจสอบจะใช้มิเตอร์วัดกำลังงานไฟฟ้ามาวัด ณ จุดจ่ายไฟฟ้าเข้าระบบที่ต้องการควบคุม แล้วอ่านค่ากำลังงานไฟฟ้าที่ใช้ไปใน 1 วัน แล้วนำค่าที่ได้ไปคำนวณ ตามสมการที่ 2.4

การควบคุมความต้องการพลังไฟฟ้าสูงสุดจะต้องคำนวณหาค่าความต้องการพลังไฟฟ้าสูงสุดที่เหมาะสมเสียก่อน ค่าความต้องการพลังไฟฟ้าสูงสุดที่เหมาะสมหาได้จากค่า LF ที่เหมาะสม เมื่อห้องทดลองเริ่มใช้ไฟฟ้าตั้งแต่เวลา 6.00 น. ถึง เวลา 1.00 น. รวมเวลาใช้ไฟฟ้าทั้งสิ้น 19 ชม. ดังนั้นเลือกค่า LF ที่เหมาะสมคิดจาก 80 % ของช่วงเวลาใช้ไฟฟ้าคำนวณได้ดังนี้

$$LF = \frac{19}{24} \times 80\%$$

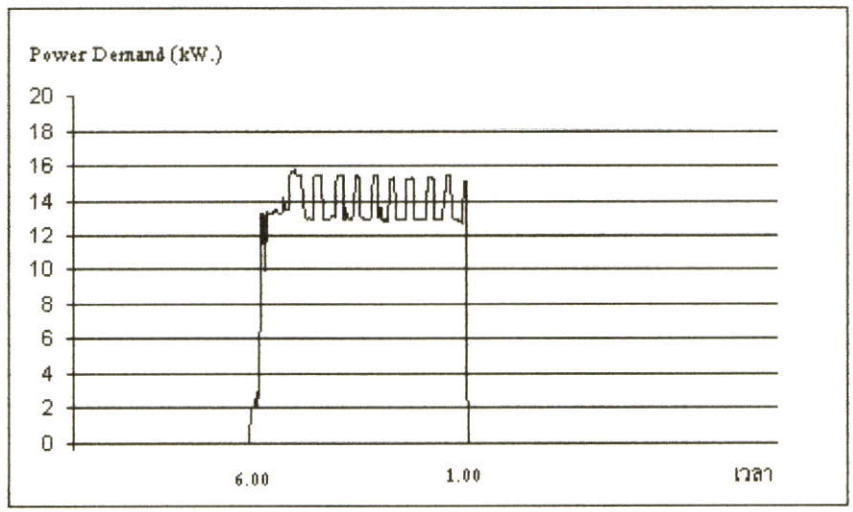
$$LF = 63.33\ \%$$

นำค่า LF ที่ได้มาคำนวณหาค่าความต้องการพลังไฟฟ้าสูงสุดที่เหมาะสมจากสมการที่ 2.4

ในการทดลองนี้ได้ใช้ตรวจวัดค่ากำลังงานไฟฟ้าในรอบ 1 วัน แทน 1 เดือน เพื่อความสะดวก รวดเร็ว และไม่ส่งผลกับระบบมาก เนื่องจากระบบที่ควบคุมเป็นห้องทดลองมีการใช้โหลดไม่สม่ำเสมอเหมือนกันทุกวัน จึงจำลองการควบคุมแค่วันเดียว ค่าที่นำมาใช้คำนวณเกิดจากการวัดใน 1 วัน จากการวัดด้วย Power Meter รุ่น A2000 สามารถวัดค่ากำลังไฟฟ้าที่ใช้ไปได้ 266 kWh. นำไปคำนวณหาค่าความต้องการพลังไฟฟ้าที่เหมาะสมจากสมการที่ 2.4 ได้

$$D_m^* = \frac{266\,\text{kWh.} \times 100\%}{63.33 \times 24} = 17.5\text{kW.}$$

เนื่องจากโหลดที่เพิ่มเข้ามามีปริมาณ 5 kW. เป็นโหลดที่สามารถตัดออกได้จากระบบจึงตั้งค่าควบคุมพลังไฟฟ้าไว้ที่ 17.5-5 = 12.5 kW. ใช้ค่าฮีสเตอรีซีส = 10% ได้ค่า Upper Limit = 13.75 kW. และค่า Lower Limit = 11.25 kW.



รูปที่ 5.14 กราฟการใช้กำลังไฟฟ้า (Load Curve) ใน 1 วัน

จากรูปที่ 5.14 สามารถหาค่าความต้องการพลังไฟฟ้าสูงสุดได้เท่ากับ 15.8 kW. และค่าความต้องการพลังไฟฟ้าต่ำสุดได้ เท่ากับ 0 kW.

จากรูปที่ 5.14 สามารถหาค่าความต้องการพลังไฟฟ้าเฉลี่ยได้จาก

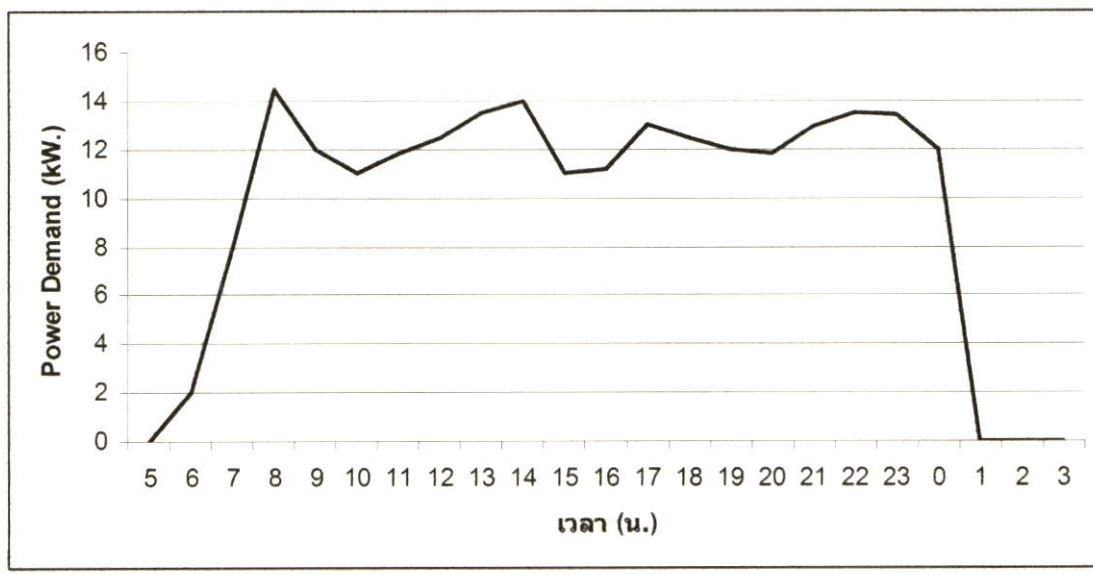
$$\text{Average Demand} = \frac{\sum(\text{kW} \times \text{t})}{\text{t}_{\text{total}}}$$

$$\text{Average Demand} = \frac{(14 \times 19) + (0 \times 5)}{24} = 11.083 \text{ kW.}$$

หาค่าดีมานด์แฟคเตอร์ (Demand Factor) จากสมการ

$$\text{Demand Factor} = \frac{Maximum \quad Demand}{Connect \quad Load} \times 100\%$$

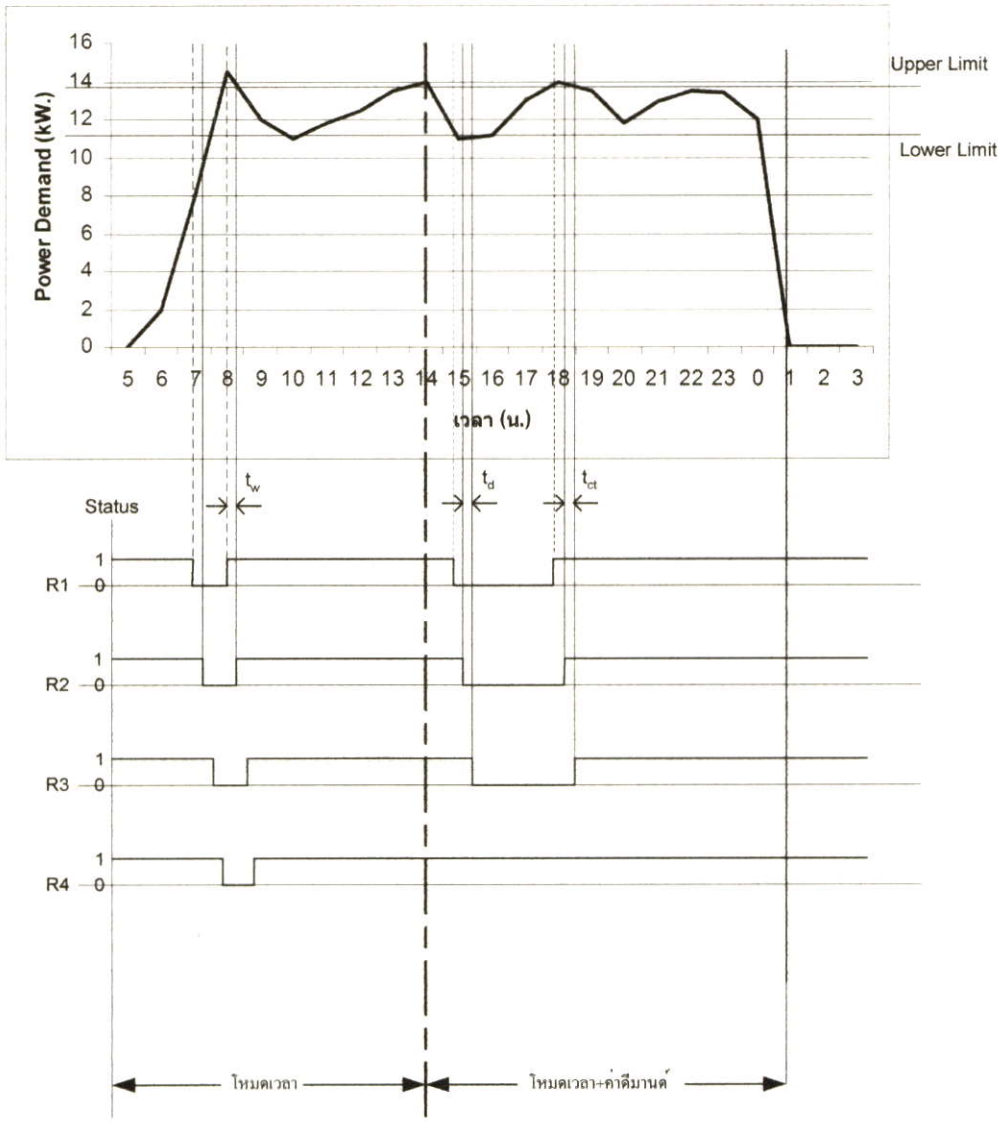
$$\text{Demand Factor} = \frac{15.8}{28.80} \times 100 = 54.8\%$$



รูปที่ 5.15 กราฟความต้องการพลังไฟฟ้า (Demand) ที่ถูกควบคุมใน 1 วัน

เมื่อมีการควบคุมค่าความต้องการพลังไฟฟ้าไว้ในย่าน Upper Limit = 13.75 และค่า Lower Limit = 11.25 kW. จากรูปที่ 5.15 เป็นกราฟค่าความต้องการพลังไฟฟ้า แบ่งการควบคุมเป็น 2 ช่วงเวลา คือ ตั้งแต่เวลา 6.00 น. ถึง 14.00 น. ใช้การควบคุมในโหมดเวลามีการตั้งค่าควบคุมดังแสดงในรูป 5.16 ช่วงเวลาที่กำหนดตั้งแต่เวลา 14.00 น. ถึง 01.00 น. ใช้การควบคุมในโหมดเวลา

ร่วมกับค่าความต้องการพลังไฟฟ้าโดยตั้งค่า Upper Limit เท่ากับ 13.75 kW. และค่า Lower Limit เท่ากับ 11.25 kW. ตั้งค่า t_d เท่ากับ 15 นาทีและ t_{ct} เท่ากับ 15 นาที



รูปที่ 5.16 กราฟการทำงานของรีเลย์เทียบกับค่าความต้องการพลังไฟฟ้า

ในช่วงเวลาตั้งแต่ 6.00 น. ถึง 16.00 น. เป็นการทำงานในโหมดเวลามีการตั้งค่าให้รีเลย์แต่ละตัวทำงานในช่วงเวลาต่างกัน 15 นาที (t_w) หลังจาก 14.00 น. แล้วมีการเปลี่ยนโหมดการควบคุมเป็นควบคุมด้วยเวลาร่วมกับค่าความต้องการพลังไฟฟ้า แต่ในขณะนั้นรีเลย์ทั้ง 4 ตัวมีสถานะทำงานอยู่ คือ โหลดที่ต่อร่วมทั้ง 4 ถูกตัดออกจนกระทั่งเวลา 14.45 น. ค่าความต้องการพลังงานไฟฟ้าลดต่ำลงกว่าค่า Lower Limit ทำให้รีเลย์ตัวที่ 1 (R1) ทำงานต่อโหลดคืนเข้าสู่ระบบแล้วรอจนครบค่าเวลา t_d ก็กลับมาตรวจพบว่าโหลดยังต่ำกว่าค่า Lower Limit อยู่ทำให้รีเลย์ 2 (R2) ทำงานต่อโหลด

คืนเข้าสู่ระบบ หลังจากนั้นค่าความต้องการพลังงานไฟฟ้าก็เริ่มกลับเข้าสู่แบนฮีสเตอรีซีสทำให้ไม่มีการทำงานของรีเลย์จนกระทั่งเวลา 17.50 น. ค่าความต้องการพลังไฟฟ้าพุ่งสูงขึ้นเกินกว่าค่า Upper Limit รีเลย์ก็เริ่มทำงานตัดโหลดอีกครั้ง หลังจากตัดโหลดไปแล้ว 3 ตัวค่าความต้องการพลังไฟฟ้ากลับเข้าสู่แบนฮีสเตอรีซิสหลังจากนั้นไปจึงไม่มีการทำงานของรีเลย์

5.3.3 สรุปขั้นตอนการติดตั้งระบบควบคุมค่าความต้องการพลังไฟฟ้า

1. สำรวจโหลด และช่วงเวลาการใช้โหลดโดยใช้ใบสำรวจจดบันทึก
2. แบ่งโหลดที่จะควบคุม
3. ต่อวงจรวัด และควบคุม
4. วัดปริมาณพลังงานไฟฟ้าที่ใช้ใน 1 วัน
5. คำนวณหาค่าที่ใช้ควบคุม
6. เริ่มควบคุม
7. สรุปและบันทึกผล

บทที่ 6

สรุปและวิจารณ์ผลการทดลอง

6.1 สรุปผลการวิจัย

การควบคุมค่าความต้องการพลังไฟฟ้าให้เหมาะสมตามช่วงเวลาต่างๆ จะทำให้เกิดประสิทธิผลสูงสุด ช่วยลดค่าไฟฟ้าได้สูงสุด และ ส่งผลกระทบกับ การใช้ไฟฟ้าน้อยที่สุด เนื่องจากจะช่วยลดทั้งค่าความต้องการพลังไฟฟ้า และ ค่าพลังงานไฟฟ้า แต่การควบคุมนี้จะต้องเกิดจาก การวัด, การสื่อสารข้อมูล และการควบคุม ข้อมูลที่ได้จากการวัดนำไปใช้ประมวลผลควบคุมการใช้พลังไฟฟ้าโดยการส่งถ่ายข้อมูล (Data Transfer) โอกาสผิดพลาดของข้อมูลระหว่างการสื่อสารมีค่อนข้างสูง มาตรการแก้ไขข้อผิดพลาดจากการรับ-ส่งข้อมูลเป็นปัจจัยสำคัญที่ช่วยให้ได้ข้อมูลที่ถูกต้อง อีกส่วนก็คือ การตั้งค่าควบคุม ตัวควบคุมที่มีข้อมูลในการควบคุมที่เหมาะสมจึงสามารถจัดการกับโหลดให้ถูกต้องได้ในช่วงเวลาต่างๆ หากข้อมูลการควบคุมผิดพลาด ก็จะส่งผลเสียกับระบบงานที่ต้องใช้ไฟฟ้าได้ ข้อมูลที่ว่านี้ได้มาจากการวัด และการจัดเรียงลำดับความสำคัญของโหลดแต่ละจุดในแต่ละช่วงเวลา จากข้อมูลของผู้เชี่ยวชาญ ว่าจุดไหนตัดได้ ในช่วงเวลาไหน นำมารวบรวมเป็นฐานข้อมูล คำนวณหาค่าที่เหมาะสมเพื่อใช้ควบคุมต่อไป ระบบควบคุมความต้องการพลังไฟฟ้าชาญฉลาดนี้ มีความสามารถที่จะ บันทึก วิเคราะห์ และป้อนโปรแกรมเพื่อควบคุมให้เกิดประโยชน์สูงสุด

6.2 ข้อเสนอแนะในการพัฒนาต่อไป

ระบบที่จัดทำขึ้นยังขาดอุปกรณ์วัดค่าความต้องการพลังไฟฟ้า (Power Demand Meter) ที่ยังต้องใช้จากต่างประเทศอยู่ เนื่องจากกระบวนการผลิตต้องใช้เวลานานการ วิจัย ทดสอบ และสอบเทียบเครื่องมือวัด จนมีความเที่ยงตรงสูง เป็นเรื่องที่อยู่นอกเหนือจากวิทยานิพนธ์ฉบับนี้ แต่จะจัดเป็นแผนงานวิจัยในขั้นต่อไปเพื่อจัดสร้างระบบควบคุมความต้องการพลังไฟฟ้าที่สมบูรณ์ ปราศจากการนำเข้า พัฒนาให้สามารถรองรับความต้องการใช้งานได้หลากหลายรูปแบบ ทางด้านโปรแกรม (Software) ก็ต้องพัฒนาให้มีความสามารถเพิ่มขึ้น จัดแต่งให้มาตรฐานเพิ่มฟังก์ชั่นการทางาน เหเหมาะสมสอดคลองกบการ เชงานจรง และทดสอบ เนสภาวะแวดลอมตางๆเพอสรางความมั่นใจในเสถียรภาพของระบบ

เอกสารอ้างอิง

[1] Ming Yuan Cho, Shih Wei Gau, Cha Win Huang, "**Development of Microprocessor Based Demand Control System for Industrial and Commercial Customer**," IEEE 2001.

[2] A.J. Hoffman,"**Peak demand control in commercial buildings with target peak adjustment base on load forecasting**," Proceedings of the 1998 IEEE International Conference on Control Application Trieste, Italy 1-4 September 1998.

[3] Richard E. Stephens, "**Load Control Demand Reduction Estimation**," IEEE Transaction on Power Syatem, Vol. 3, No. 1, February 1988.

[4] Dr W C Beattie and Dr P McCafferty, "**The Use of a forecasting algorithm to provide improved maximum Demand Control,** "

[5] A. Martins, H. Jorge, J. Mota, R. Parracho and A. Gomes, "**A PC-Based Simulation Package for Supporting End-User Demand Side Management** ," IEEE Transitions on Power System, Vol. 6 No. 3, August 1991.

[6] Industrial Technology Center. 2547. **Electric & Energy**. [Online]. Available : http://www.9engineer.com/

[7] การไฟฟ้านครหลวง. 2547. **ค่าไฟฟ้า.** [Online]. Available : http://www.mea.or.th/

[8] การไฟฟ้าส่วนภูมิภาค. 2547. **วิธีคิดค่าไฟฟ้าอัตรา TOU และ TOD**. [Online]. Available : http://www.pea.co.th/

[9] GOSSEN METRAWATT. 2547. **A2000 Multifunctional Power Meter Modbus Communications Protocol-Mod 1.** [Online]. Available : http://www.gmc-instruments.com/english/produkte/a2000.htm

[10] B&B Electronic. 2547. **RS422 And RS485 Application Note**. [Online]. Available : http://www.bb-elec.com/

[11] ธีรวัฒน์ ประกอบผล. 2541. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์**. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)

ภาคผนวก

ภาคผนวก ก.

อัตราค่าไฟฟ้าของผู้ใช้ไฟฟ้าประเภทต่างๆ

อัตราค่าไฟฟ้า

ประเภทที่ 1 บ้านอยู่อาศัย

สำหรับการใช้ไฟฟ้าภายในบ้านเรือนที่อยู่อาศัย ตลอดจนบริเวณที่เกี่ยวข้อง รวมทั้งวัด สำนักสงฆ์ และสถานประกอบศาสนกิจของทุกศาสนา โดยต่อผ่านเครื่องวัดไฟฟ้าเครื่องเดียว

1.1 ปริมาณการใช้พลังงานไฟฟ้า ไม่เกิน 150 หน่วยต่อเดือน

อัตรารายเดือน ค่าพลังงานงานไฟฟ้า (Energy Charge)

5 หน่วยแรก	(หน่วยที่ 1 – 5)	เป็นเงิน	4.96	บาท
10 หน่วยต่อไป	(หน่วยที่ 6 – 15)	หน่วยละ	0.7124	บาท
10 หน่วยต่อไป	(หน่วยที่ 16 – 25)	หน่วยละ	0.8993	บาท
65 หน่วยต่อไป	(หน่วยที่ 26 – 35)	หน่วยละ	1.5348	บาท
50 หน่วยต่อไป	(หน่วยที่ 101 – 150)	หน่วยละ	1.6282	บาท
250 หน่วยต่อไป	(หน่วยที่ 151 – 400)	หน่วยละ	2.1329	บาท
เกิน 400 หน่วยขึ้นไป	(หน่วยที่ 401 เป็นต้นไป)	หน่วยละ	2.4226	บาท

ค่าไฟฟ้าต่ำสุด : เดือนละ 4.67 บาท

1.2 ปริมาณการใช้พลังงานไฟฟ้า เกิน 150 หน่วยต่อเดือน

อัตรารายเดือน ค่าพลังงานไฟฟ้า (Energy Charge)

35 หน่วยแรก	(หน่วยที่ 1 – 35)	เป็นเงิน	85.21	บาท
115 หน่วยต่อไป	(หน่วยที่ 36 – 150)	หน่วยละ	1.1236	บาท
250 หน่วยต่อไป	(หน่วยที่ 151 – 400)	หน่วยละ	2.1329	บาท
เกิน 400 หน่วยขึ้นไป	(หน่วยที่ 401 เป็นต้นไป)	หน่วยละ	2.4226	บาท

ค่าไฟฟ้าต่ำสุด : เดือนละ 83.18 บาท

หมายเหตุ 1. ผู้ใช้ไฟฟ้าที่ติดตั้งเครื่องวัดไม่เกิน 5 แอมป์ 220 โวลต์ 1 เฟส 2 สาย จะจัดเข้า ประเภทที่ 1.1 แต่หากมีการใช้ไฟฟ้าเกิน 150 หน่วยติดต่อกัน 3 เดือน ในเดือนถัดไปจะจัดเข้าประเภทที่ 1.2และเมื่อใดที่การใช้ไฟฟ้าไม่เกิน 150 หน่วย ติดต่อกัน 3 เดือน ในเดือนถัดไปจะจัดเข้าประเภทที่ 1.1

2. ผู้ใช้ไฟฟ้าที่ติดตั้งเครื่องวัดเกิน 5 แอมป์ 220 โวลต์ 1 เฟส 2 สาย ให้ใช้อัตราประเภทที่ 1.2

ประเภทที่ 2 กิจการขนาดเล็ก

สำหรับการใช้ไฟฟ้าเพื่อประกอบธุรกิจธุรกิจรวมกับบ้านอยู่อาศัยอุตสาหกรรม ส่วนราชการที่มีลักษณะเป็นอุตสาหกรรม รัฐวิสาหกิจ หรืออื่น ๆ ตลอดจนบริเวณที่เกี่ยวข้อง ซึ่งมีความต้องการพลังงานไฟฟ้าไม่ถึง 30 กิโลวัตต์ โดยต่อผ่านเครื่องวัดไฟฟ้าเครื่องเดียว

อัตรารายเดือน ค่าพลังงานไฟฟ้า (Energy Charge)

35 หน่วยแรก	(หน่วยที่ 1 – 35)หน่วยละ	89.89	บาท
115 หน่วยต่อไป	(หน่วยที่ 36 – 150) หน่วยละ	1.1236	บาท
250 หน่วยต่อไป	(หน่วยที่ 151 - 400)หน่วยละ	2.1329	บาท
เกิน 400 หน่วยขึ้นไป	(หน่วยที่ 401 เป็นต้นไป)หน่วยละ	2.4226	บาท

ค่าไฟฟ้าต่ำสุด : เดือนละ 87.85 บาท

หมายเหตุ หากเดือนใดมีความต้องการการพลังงานไฟฟ้าตั้งแต่ 30 กิโลวัตต์ขึ้นไป จะจัดเข้าอยู่ในประเภทที่ 3-5 แล้วแต่กรณี

ประเภทที่ 3 กิจการขนาดกลาง

สำหรับการใช้ไฟฟ้าเพื่อประกอบธุรกิจ อุตสาหกรรม ส่วนราชการที่มีลักษณะเป็นอุตสาหกรรมรัฐวิสาหกิจ ตลอดจนบริเวณที่เกี่ยวข้อง ซึ่งมีความต้องการพลังงานไฟฟ้าเฉลี่ยใน 15 นาทีสูงสุด ตั้งแต่ 30 กิโลวัตต์ แต่ไม่ถึง 2.000 กิโลวัตต์ และมีปริมาณการใช้พลังงานไฟฟ้าเฉลี่ย 3 เดือนไม่เกิน 355,000 หน่วยต่อเดือน รวมถึงส่วนราชการ (ตามหลักเกณฑ์ประเภทที่ 6) ที่มีปริมาณการใช้พลังงานไฟฟ้าเฉลี่ย 3 เดือนเกิน 250,000 หน่วย แต่ไม่เกิน

3.1 อัตราปกติ

อัตรารายเดือน	**ค่าความต้องการพลังไฟฟ้า (Demand Charge) บาท/กิโลวัตต์**	**ค่าพลังงานไฟฟ้า (Energy Charge) บาท/หน่วย**
3.1.1 แรงดันตั้งแต่ 39 กิโลโวลต์ขึ้นไป	175.70	1.0208
3.1.2 แรงดัน 22 – 33 กิโลโวลต์	196.26	1.0582
3.1.3 แรงดันต่ำกว่า 22 กิโลโวลต์	221.50	1.0862

3.2 คิดตามช่วงเวลาของการใช้ (Time Of Use Rate)

อัตรารายเดือน	ค่าความต้องการพลังไฟฟ้า (Demand Charge) บาท/กิโลวัตต์	ค่าพลังงานไฟฟ้า (Energy Charge) บาท/หน่วย			ค่าบริการ (Service Charge) บาท
	1*	1*	2*	3*	
3.2.1 แรงดัน 115 กิโลโวลต์ขึ้นไป	102.80	1.5349	0.6671	0.6062	400.00
3.2.2 แรงดัน 69 กิโลโวลต์	158.88	1.6292	0.6769	0.6153	400.00
3.2.3 แรงดัน 22-33 กิโลโวลต์	200.93	1.7736	0.6861	0.6236	850.00
3.2.4 แรงดันต่ำกว่า 22 กิโลโวลต์	214.95	1.8891	0.7283	0.6616	850.00

1* วันจันทร์-เสาร์ เวลา 9.00 – 22.00 น.(On Peak)

2* วันจันทร์-เสาร์ เวลา 22.00 – 09.00 น.(Off Peak)

3* วันอาทิตย์ เวลา 00.00 – 24.00 น.(Off Peak)

อัตราขั้นต่ำ : ค่าไฟฟ้าต่ำสุดต้องไม่ต่ำกว่าร้อยละ 70 ของค่าความต้องการพลังงานไฟฟ้าที่สูงสุดในรอบ 12 เดือนที่ผ่านมา สิ้นสุดในเดือนปัจจุบัน

หมายเหตุ 1. กรณีการติดตั้งเครื่องวัดไฟฟ้ากระทำทางด้านแรงต่ำของหม้อแปลงซึ่งเป็นสมบัติของผู้ใช้ไฟฟ้า ให้คำนวณกิโลวัตต์และหน่วยคิดเงินเพิ่มขึ้นอีกร้อยละ 2 เพื่อให้ได้ผลคลุมไปถึงการสูญเสียในหม้อแปลงไฟฟ้า ซึ่งมิได้วัดรวมไว้ด้วย

2. เดือนใดความต้องการพลังงานไฟฟ้าสูงสุดไม่ถึง 30 กิโลวัตต์ ค่าไปป้ายังคงคำนวณ ตามอัตราดังกล่าวแต่ทั้งนี้ต้องไม่ต่ำกว่าอัตราขั้นต่ำ หากความต้องการพลังงานไฟฟ้าไม่ถึง 30 กิโลวัตต์ ติดต่อกันเป็นเวลา 12 เดือน และในเดือนถัดไปก็ยังไม่ถึง 30 กิโลวัตต์อีก ให้เปลี่ยนประเภทผู้ใช้ไฟฟ้าเป็นประเภทที่ 2 หรือ 6 แล้วแต่กรณี

3. อัตราประเภทที่ 3.2 เป็นอัตราเลือกสำหรับผู้ที่มีการใช้ไฟฟ้าเฉลี่ย 3 เดือน เกิน 250,000 หน่วยต่อเดือน

4. อัตราประเภทที่ 3.2 เป็นอัตราบังคับสำหรับส่วนราชการ (ตามหลักเกณฑ์ประเภทที่ 6) ตั้งแต่ค่าไฟฟ้าประจำเดือนตุลาคม 2540 เป็นต้นไป

5. หากใช้อัตราประเภทที่ 3.2 แล้ว จะกลับไปใช้อัตราประเภทที่ 3.1 ไม่ได้

ประเภทที่ 4 กิจการขนาดใหญ่

สำหรับการใช้ไฟฟ้าประกอบธุรกิจ อุตสาหกรรม ส่วนราชการ รัฐวิสาหกิจ สถานที่บริเวณที่เกี่ยวข้อง ซึ่งมีความต้องการพลังไฟฟ้าเฉลี่ยใน 15 นาทีสูงสุด ตั้งแต่ 2,000 กิโลวัตต์ขึ้นไป หรือมีปริมาณการใช้พลังงานไฟฟ้าเฉลี่ย 3 เดือนเกิน 355,000 หน่วยต่อเดือน โดยต่อเครื่องวัดไฟฟ้าเครื่องเดียว

4.1 คิดตามช่วงเวลาของวัน (Time of Day Rate)

อัตรารายเดือน	ค่าความต้องการพลังไฟฟ้า (Demand Charge) บาท/กิโลวัตต์			ค่าพลังงานไฟฟ้า (Energy Charge) บาท/หน่วย
	1*	2*	3*	
4.1.1 แรงดัน 69 กิโลโวลต์	224.30	29.91	0	1.0208
4.1.2 แรงดัน 22 – 33 กิโลโวลต์	285.05	58.88	0	1.0582
4.1.3 แรงดันต่ำกว่า 22 กิโลโวลต์	332.71	68.22	0	1.0862

1* เวลา 18.30 – 21.30 น. (On Peak)

2* เวลา 08.00 – 18.30 น. (Partial Peak) คิดเฉพาะส่วนที่เกินช่วงที่ Peak

3* เวลา 21.30 – 08.00 น.ของทุกวัน (Off Peak)

4.2 คิดตามช่วงเวลาของการใช้ (Time Of Use Rate)

อัตรารายเดือน	ค่าความต้องการพลังไฟฟ้า (Demand Charge) บาท/กิโลวัตต์	ค่าพลังงานไฟฟ้า (Energy Charge) บาท/หน่วย			ค่าบริการ (Service Charge) บาท
	1*	1*	2*	3*	
4.2.1 แรงดัน115 กิโลโวลต์ขึ้นไป	102.80	1.5349	0.6671	0.6062	400.00
4.2.2 แรงดัน 69 กิโลโวลต์	158.88	1.6292	0.6769	0.6153	400.00
4.2.3 แรงดัน 22-33 กิโลโวลต์	200.93	1.7736	0.6861	0.6236	850.00
4.2.4 แรงดันต่ำกว่า 22 กิโลโวลต์	214.95	1.8891	0.7283	0.6616	850.00

1* วันจันทร์-เสาร์ เวลา 9.00 – 22.00 น.(On Peak)

2* วันจันทร์-เสาร์ เวลา 22.00 – 09.00 น.(Off Peak)

3* วันอาทิตย์ เวลา 00.00 – 24.00 น.(Off Peak)

อัตราขั้นต่ำ : ค่าไฟฟ้าต่ำสุดต้องไม่ต่ำกว่าร้อยละ 70 ของค่าความต้องการพลังงานไฟฟ้าสูงสุดในรอบ 12 เดือนที่ผ่านมา สิ้นสุดในเดือนปัจจุบัน

หมายเหตุ

1. เดือนใดความต้องการพลังงานไฟฟ้าไม่ถึง 2,000 กิโลวัตต์ หรือการใช้ไฟฟ้าเฉลี่ย 3 เดือนไม่เกิน 355.000 หน่วยต่อเดือน ค่าไฟฟ้ายังคงคำนวณตามอัตราดังกล่าว แต่ทั้งนี้ ต้องไม่ต่ำกว่าอัตราขั้นต่ำ หากความต้องการพลังงานไฟฟ้าไม่ถึง 30 กิโลวัตต์ ติดต่อกันเป็นเวลา 12 เดือน และในเดือนถัดไปก็ยังไม่ถึง 30 กิโลวัตต์อีก ให้เปลี่ยนประเภทผู้ใช้ไฟฟ้าเป็นประเภทที่ 2 หรือ 6 แล้วแต่กรณี
2. อัตราประเภทที่ 4.2 เป็นอัตราเลือกสำหรับผู้ใช้ไฟฟ้าประเภทที่ 4 หรือเข้าหลักเกณฑ์ค่าไฟฟ้าคิดตามช่วงเวลาของวัน
3. อัตราประเภทที่ 4.2 เป็นอัตราบังคับสำหรับผู้ใช้ไฟฟ้าประเภทที่ 4 เป็นครั้งแรกตั้งแต่ค่าไฟฟ้าประจำเดือนตุลาคม 2540 เป็นต้นไป
4. หากใช้อัตราประเภทที่ 4.2 แล้ว จะกลับไปใช้อัตราประเภทที่ 4.1 ไม่ได้

ประเภทที่ 5 กิจการเฉพาะอย่าง

สำหรับการใช้ไฟฟ้าเพื่อประกอบกิจการโรงแรมและบริการที่พักอาศัยให้เช่า ตลอดจนบริเวณที่เกี่ยวข้องซึ่งมีความต้องการพลังงานไฟฟ้าเฉลี่ยใน 15 นาทีสูงสุด ตั้งแต่ 30 กิโลวัตต์ขึ้นไป โดยต่อผ่านเครื่องวัดไฟฟ้าเครื่องเดียว

5.1 อัตราปกติ

อัตรารายเดือน	ค่าความต้องการพลังไฟฟ้า (Demand Charge) บาท/กิโลวัตต์	ค่าพลังงานไฟฟ้า (Energy Charge) บาท/หน่วย
3.1.1 แรงดันตั้งแต่ 39 กิโลโวลต์ขึ้นไป	220.56	1.0208
3.1.2 แรงดัน 22 – 33 กิโลโวลต์	256.07	1.0582
3.1.3 แรงดันต่ำกว่า 22 กิโลโวลต์	276.64	1.0862

5.2 คิดตามช่วงเวลาของการใช้ (Time Of Use Rate)

อัตรารายเดือน	ค่าความต้องการพลังไฟฟ้า (Demand Charge) บาท/กิโลวัตต์	ค่าพลังงานไฟฟ้า (Energy Charge) บาท/หน่วย			ค่าบริการ (Service Charge) บาท
	1*	1*	2*	3*	
4.2.1 แรงดัน115 กิโลโวลต์ขึ้นไป	102.80	1.5349	0.6671	0.6062	400.00
4.2.2 แรงดัน 69 กิโลโวลต์	158.88	1.6292	0.6769	0.6153	400.00
4.2.3 แรงดัน 22-33 กิโลโวลต์	200.93	1.7736	0.6861	0.6236	850.00
4.2.4 แรงดันต่ำกว่า 22 กิโลโวลต์	214.95	1.8891	0.7283	0.6616	850.00

1* วันจันทร์-เสาร์ เวลา 9.00 – 22.00 น.(On Peak)

2* วันจันทร์-เสาร์ เวลา 22.00 – 09.00 น.(Off Peak)

3* วันอาทิตย์ เวลา 00.00 – 24.00 น.(Off Peak)

อัตราขั้นต่ำ : ค่าไฟฟ้าต่ำสุดต้องไม่ต่ำกว่าร้อยละ 70 ของค่าความต้องการพลังงานไฟฟ้าสูงสุดในรอบ 12 เดือนที่ผ่านมา สิ้นสุดในเดือนปัจจุบัน

หมายเหตุ

1. กรณีการติดตั้งเครื่องวัดไฟฟ้ากระทำทางด้านแรงต่ำของหม้อแปลงซึ่งเป็นสมบัติของผู้ใช้ไฟฟ้า ให้
   ให้คำนวณกิโลวัตต์และหน่วยคิดเงินเพิ่มขึ้นอีกร้อยละ 2 เพื่อให้ได้ผลคลุมไปถึงการสูญเสียในหม้อแปลงไฟฟ้า ซึ่งมิได้วัดรวมไว้ด้วย
2. เดือนใดความต้องการพลังงานไฟฟ้าสูงสุดไม่ถึง 30 กิโลวัตต์ ค่าไฟฟ้ายังคงคำนวณตามอัตราดังกล่าว แต่ทั้งนี้ต้องไม่ต่ำกว่าอัตราขั้นต่ำ หากความต้องกาลพลังงานไฟฟ้าไม่ถึง 30 กิโลวัตต์ ติดต่อกันเป็นเวลา 12 เดือน และในเดือนถัดไปก็ยังไม่ถึง 30 กิโลวัตต์อีก ให้เปลี่ยนประเภทผู้ใช้ไฟฟ้าเป็นประเภทที่ 2
3. อัตราประเภทที่ 5.2 เป็นอัตราเลือกสำหรับผู้ที่มีการใช้ไฟฟ้าเฉลี่ย 3 เดือน เกิน 250,000 หน่วยต่อเดือน
4. อัตราประเภทที่ 5.2 เป็นอัตราบังคับสำหรับผู้ที่มีการใช้ไฟฟ้าเฉลี่ย 3 เดือนเกิน 355,000 หน่วยต่อเดือน เป็นครั้งแรกตั้งแต่ค่าไฟฟ้าประจำเดือนตุลาคม 2540 เป็นต้นไป
5. หากใช้อัตราประเภทที่ 5.2 แล้ว จะกลับไปใช้อัตราประเภทที่ 5.1 ไม่ได้

ประเภทที่ 6 ส่วนราชการและองค์การที่ไม่แสวงหากำไร

สำหรับการใช้ไฟฟ้าของหน่วยราชการ สำนักงานหรือหน่วยงานอื่นใดของรัฐ หน่วยงานตามกฎหมายว่าด้วยระเบียบบริหารราชการส่วนท้องถิ่น ซึ่งมีปริมาณการใช้พลังงานไฟฟ้าเฉลี่ย 3 เดือนไม่เกิน 250,000 หน่วยต่อเดือน รวมถึงองค์การที่ไม่ใช่ส่วนราชการแต่มีวัตถุประสงค์ในการให้บริการโดยไม่คิดค่าตอบแทน แต่ไม่รวมหน่วยงานของรัฐวิสาหกิจ สถานทูต สถานที่ทำการ

ของหน่วยงานราชการต่างประเทศ และสถานที่ทำการขององค์การระหว่างประเทศ โดยต่อผ่านเครื่องวัดไฟฟ้าเครื่องเดียว

อัตรารายเดือน ค่าพลังงานไฟฟ้า (Energy Charge)

6.1 แรงดันตั้งแต่ 69 กิโลวัตต์ขึ้นไป

10,000 หน่วยแรก (หน่วยที่ 1 – 10,000)	เป็นเงิน	14,413.31	บาท
เกิน 10,000 หน่วยขึ้นไป (หน่วยที่ 10,001 ขึ้นไป)	หน่วยละ	1.4413	บาท

ค่าไฟฟ้าต่ำสุด : เดือนละ 13,831.78 บาท

6.2 แรงดัน 22-33 กิโลโวลต์

300 หน่วยแรก (หน่วยที่ 1 – 300)	ปืนเงิน	480.06	บาท
เกิน 300 หน่วยขึ้นไป (หน่วยที่ 301 ขึ้นไป)	หน่วยละ	1.6002	บาท

ค่าไฟฟ้าต่ำสุด : เดือนละ 462.62 บาท

6.3 แรงดันต่ำกว่า 22 กิโลโวลต์

10 หน่วยแรก (หน่วยที่ 1 – 10)	เป็นเงิน	18.06	บาท
เกิน 10 หน่วยขึ้นไป (หน่วยที่ 11 ขึ้นไป)	หน่วยละ	1.8058	บาท

ค่าไฟฟ้าต่ำสุด : เดือนละ 17.48 บาท

หมายเหตุ

1. วัด สถานประกอบศาสนกิจของศาสนา ตลอดจนบริเวณที่เกี่ยวข้องที่คิดอัตราประเภทบ้านอยู่อาศัย หากมีการใช้ไฟฟ้าเกิน 325 หน่วยติดต่อกัน 3 เดือน ในเดือนถัดไปจะจัดเข้าประเภทที่ 6 และเมื่อใดที่การใช้ไฟฟ้าไม่เกิน 325 หน่วยติดต่อกัน 3 เดือน ในเดือนถัดไปจะจัดเข้าประเภทที่ 1
2. กรณีการติดตั้งเครื่องวัดไฟฟ้ากระทำทางด้านแรงต่ำของหม้อแปลงไฟฟ้า ซึ่งเป็นสมบัติของผู้ใช้ไฟฟ้า ให้คำนวณหน่วยคิดเงินเพิ่มขึ้นอีกร้อยละ 2 เพื่อให้ได้ผลคลุมไปถึงการสูญเสียในหม้อแปลงไฟฟ้า ซึ่งมิได้วัดรวมไปด้วย

ประเภทที่ 7 สูบน้ำเพื่อการเกษตร

สำหรับการใช้ไฟฟ้ากับเครื่องสูบน้ำเพื่อการเกษตรของหน่วยงานราชการ กลุ่มเกษตรกรที่จดทะเบียนจัดตั้งกลุ่มเกษตร หรือสหกรณ์เพื่อการเกษตร โดยต่อผ่านเครื่องวัดไฟฟ้าเครื่องเดียว

อัตรารายเดือน ค่าพลังงานไฟฟ้า (Energy Charge)

100 หน่วยแรก (หน่วยที่ 1 – 100)	เป็นเงิน	115.16	บาท
เกิน 100 หน่วยขึ้นไป (หน่วยที่ 101 เป็นต้นไป)	หน่วยละ	1.1516	บาท

ค่าไฟฟ้าต่ำสุด : เดือนละ 109.35 บาท

หมายเหตุ กรณีการติดตั้งเครื่องวัดไฟฟ้ากระทำทางด้านแรงต่ำของหม้อแปลงไฟฟ้า ซึ่งเป็นสมบัติของผู้ใช้ไฟฟ้า หรือหม้อแปลงของการไฟฟ้าส่วนภูมิภาค (เฉพาะที่ติดตั้งเครื่องวัดไฟฟ้ากระทำการด้านแรงต่ำประกอบ ซีที) ให้คำนวณหน่วยคิดเงินเพิ่มขึ้นอีกร้อยละ 2 เพื่อให้ได้ผลคลุมไปถึงการสูญเสียในหม้อแปลงไฟฟ้าซึ่งมิได้วัดรวมไว้ด้วย

ข้อกำหนดเกี่ยวกับอัตราค่าไฟฟ้า

1.ค่าไฟฟ้าที่เรียกเก็บตามอัตราข้างต้น ยังไม่รวมภาษีมูลค่าเพิ่ม

2.หากค่าใช้จ่ายที่ไม่อยู่ในความควบคุมของการไฟฟ้าเปลี่ยนแปลงจะมีการปรับค่าพลังงานไฟฟ้าต่อหน่วยโดยอัตราการปรับจะแสดงอยู่ในใบเสร็จรับเงินค่าไฟฟ้า

เริ่มใช้ตั้งแต่ ค่าไฟฟ้าประจำเดือน มกราคม 2540 เป็นต้นไป

ภาคผนวก ข.

คำสั่ง AT COMMAND SET SUMMARY

A.1. BASIC AT COMMANDS

Command	Function
A/	Re-execute command.
A	Go off-hook and attempt to answer a call.
B0	Select V.22 connection at 1200 bps.
B1	Select Bell 212A connection at 1200 bps.
C1	Return OK message.
Dn	Dial modifier.
E0	Turn off command echo.
E1	Turn on command echo.
F0	Select auto-detect mode (equivalent to N1).
F1	Select V.21 or Bell 103.
F2	Reserved.
F3	Select V.23 line modulation.
F4	Select V.22 or Bell 212A 1200 bps line speed.
F5	Select V.22 bis line modulation.
F6	Select V.32 bis or V.32 4800 line modulation.
F7	Select V.32 bis 7200 line modulation.
F8	Select V.32 bis or V.32 9600 line modulation.
F9	Select V.32 bis 12000 line modulation.
F10	Select V.32 bis 14400 line modulation.
H0	Initiate a hang-up sequence.
H1	If on-hook, go off-hook and enter command mode.
I0	Report product code.
I1	Report pre-computed checksum. I2 Report OK.
I3	Report firmware revision, model, and interface type.
I4	Report response programmed by an OEM.
I5	Report the country code parameter.
I6	Report modem data pump model and code revision.
I7	Reports the DAA code (W-class models only).
L0	Set low speaker volume.
L1	Set low speaker volume.
L2	Set medium speaker volume.
L3	Set high speaker volume.
M0	Turn speaker off.
M1	Turn speaker on during handshaking and turn speaker off while receiving carrier.
M2	Turn speaker on during handshaking and while receiving carrier.
M3	Turn speaker off during dialing and receiving carrier and turn speaker on during answering.
N0	Turn off automode detection.
N1	Turn on automode detection.
O0	Go on-line.
O1	Go on-line and initiate a retrain sequence. PForce pulse dialing.
Q0	Allow result codes to DTE.
Q1	Inhibit result codes to DTE.
Sn	Select S-Register as default.

* Serial interface operation only

Command	Function
Sn?	Return the value of S-Register n.
=v	Set default S-Register to value v.
?	Return the value of default S-Register.
T	Force DTMF dialing.
V0	Report short form (terse) result codes.
V1	Report long form (verbose) result codes.
W0	Report DTE speed in EC mode.
W1	Report line speed, EC protocol and DTE speed.
W2	Report DCE speed in EC mode.
X0	Report basic call progress result codes, i.e., OK, CONNECT, RING, NO CARRIER (also, for busy, if enabled, and dial tone not detected), NO ANSWER and ERROR.
X1	Report basic call progress result codes and connections speeds (OK, CONNECT, RING, NO CARRIER (also, for busy, if enabled, and dial tone not detected), NO ANSWER, CONNECT XXXX, and ERROR.
X2	Report basic call progress result codes and connections speeds, i.e., OK, CONNECT, RING, NO CARRIER (also, for busy, if enabled, and dial tone not detected), NO ANSWER, CONNECT XXXX, and ERROR.
X3	Report basic call progress result codes and connection rate, i.e., OK, CONNECT, RING, NO CARRIER, NO ANSWER, CONNECT XXXX, BUSY, and ERROR.
X4	Report all call progress result codes and connection rate, i.e., OK, CONNECT, RING, NO CARRIER, NO ANSWER, CONNECT XXXX, BUSY, NO DIAL TONE and ERROR.
Y0	Disable long space disconnect before on-hook.
Y1	Enable long space disconnect before on-hook.
Z0	Restore stored profile 0 after warm reset.
Z1	Restore stored profile 1 after warm reset.
&C0	Force RLSD active regardless of the carrier state.
&C1	Allow RLSD to follow the carrier state.
&D0	Interpret DTR ON-to-OFF transition per &Qn:
&Q0, &Q5, &Q6	The modem ignores DTR.
&Q1, &Q4	The modem hangs up.
&Q2, &Q3	The modem hangs up.
&D1	Interpret DTR ON-to-OFF transition per &Qn:
&Q0, &Q1, &Q4,.Q5, &Q6	Asynchronous escape.
&Q2, &Q3	The modem hangs up.
&D2	Interpret DTR ON-to-OFF transition per &Qn:
&Q0 through &Q6	The modem hangs up.
&D3	Interpret DTR ON-to-OFF transition per &Qn:.
&Q0, &Q1, &Q4,.	
&Q5, &Q6	The modem performs soft reset.
&Q2, &Q3	The modem hangs up.
&F0	Restore factory configuration 0.
&F1	Restore factory configuration 1.
&G0	Disable guard tone.
&G1	Disable guard tone.
&G2	Enable 1800 Hz guard tone.
&J0	Set S-Register response only for compatibility.

* Serial interface operation only

Command	Function
&J1	Set S-Register response only for compatibility.
&K0	Disable DTE/DCE flow control.
&K3	Enable RTS/CTS DTE/DCE flow control.
&K4	Enable XON/XOFF DTE/DCE flow control.
&K5	Enable transparent XON/XOFF flow control.
&K6	Enable both RTS/CTS and XON/XOFF flow control.
&L0	Select dial up line operation.
&L1	Select leased line operation.
&M0	Select direct asynchronous mode.
&M1	Select sync connect with async off-line command mode. *
&M2	Select sync connect with async off-line command mode and enable DTR dialing of directory zero. *
&M3	Select sync connect with async off-line command mode and enable DTR to act as
&P0	Set 10 pps pulse dial with 39%/61% make/break.
&P1	Set 10 pps pulse dial with 33%/67% make/break.
&P2	Set 20 pps pulse dial with 39%/61% make/break.
&P3	Set 20 pps pulse dial with 33%/67% make/break.
&Q0	Select direct asynchronous mode.
&Q1	Select sync connect with async off-line command mode. *
&Q2	Select sync connect with async off-line command mode and enable DTR dialing of directory zero. *
&Q3	Select sync connect with async off-line command mode and enable DTR to act asTalk/Data switch. *
&Q4	Select Hayes AutoSync mode.
&Q5	Modem negotiates an error corrected link.
&Q6	Select asynchronous operation in normal mode.
&R0	CTS tracks RTS (async) or acts per V.25 (sync).
&R1	CTS is always active.
&S0	DSR is always active.
&S1	DSR acts per V.25.
&T0	Terminate any test in progress.
&T1	Initiate local analog loopback.
&T2	Returns ERROR result code.
&T3	Initiate local digital loopback.
&T4	Allow remote digital loopback.
&T5	Disallow remote digital loopback request.
&T6	Request an RDL without self-test.
&T7	Request an RDL with self-test.
&T8	Initiate local analog loop with self-test.
&V	Display current configurations.
&W0	Store the active profile in NVRAM profile 0.
&W1	Store the active profile in NVRAM profile 1.
&X0	Select internal timing for the transmit clock.
&X1	Select external timing for the transmit clock.
&X2	Select slave receive timing for the transmit clock.
&Y0	Recall stored profile 0 upon power up.
&Y1	Recall stored profile 1 upon power up.
&Zn=x	Store dial string x (to 35) to location n (0 to 3).
%E0	Disable line quality monitor and auto retrain.
%E1	Enable line quality monitor and auto retrain.
%E2	Enable line quality monitor and fallback/fall forward.
%L	Return received line signal level.
%Q	Report the line signal quality.

* Serial interface operation only

Command	Function
%TTn	PTT certification test signals.
\Kn	Controls break handling during three states: When modem receives a break from the DTE:
\K0,2,4	Enter on-line command mode, no break sent to the remote modem.
\K1	Clear buffers and send break to remote modem.
\K3	Send break to remote modem immediately.
\K5	Send break to remote modem in sequence with transmitted data. When modem receives \B in on-line command state:
\K0,1	Clear buffers and send break to remote modem.
\K2,3	Send break to remote modem immediately.
\K4,5	Send break to remote modem in sequence with transmitted data. When modem receives break from the remote modem:
\K0,1	Clear data buffers and send break to DTE.
\K2,3	Send a break immediately to DTE.
\K4,5	Send a break with received data to the DTE.
\N0	Select normal speed buffered mode.
\N1	Select direct mode.
\N2	Select reliable link mode.
\N3	Select auto reliable mode.
\N4	Force LAPM mode.
\N5	Force MNP mode.
+H0	Disable RPI.
+H1	Enable RPI and set DTE speed to 19200 bps.
+H2	Enable RPI and set DTE speed to 38400 bps.
+H3	Enable RPI and set DTE speed to 57600 bps.
**0	Download to flash memory at last sensed speed.
**1	Download to flash memory at 38.4 kbps.
**2	Download to flash memory at 57.6 kbps.
-SDR=0	Disable Distinctive Ring.
-SDR=1	Enable Distinctive Ring Type 1.
-SDR=2	Enable Distinctive Ring Type 2.
-SDR=3	Enable Distinctive Ring Type 1 and 2.
-SDR=4	Enable Distinctive Ring Type 3.
-SDR=5	Enable Distinctive Ring Type 1 and 3.
-SDR=6	Enable Distinctive Ring Type 2 and 3.
-SDR=7	Enable Distinctive Ring Type 1, 2, and 3.

* Serial interface operation only

ภาคผนวก ค.

MCS-51 Instruction Set

Mnemonic		Description	Byte	Cycle
Arithmetic Operations				
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A, @Ri	Add indirect RAM to accumulator	1	1
ADD	A,#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry flag	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A, @Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A, #data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1	1
SUBB	A,#data	Subtract immediate data from A with borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Multiply A and B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1

Mnemonic		Description	Byte	Cycle
Logic Operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive OR register to accumulator	1	1
XRL	A direct	Exclusive OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive OR indirect RAM to accumulator	1	1
XRL	A,#data	Exclusive OR immediate data to accumulator	2	1
XRL	direct,A	Exclusive OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive OR immediate data to direct byte	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate accumulator left through carry	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate accumulator right through carry	1	1
SWAP	A	Swap nibbles within the accumulator	1	1

Mnemonic		Description	Byte	Cycle
Data Transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct *)	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri, #data	Move immediate data to indirect RAM	2	1
MOV	DPTR, #data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to accumulator	1	2
MOVC	A,@A + PC	Move code byte relative to PC to accumulator	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr.) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr.) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr.)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr.)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with accumulator	1	1
XCHD	A,@Ri	Exchange low-order nibble indir. RAM with A	1	1

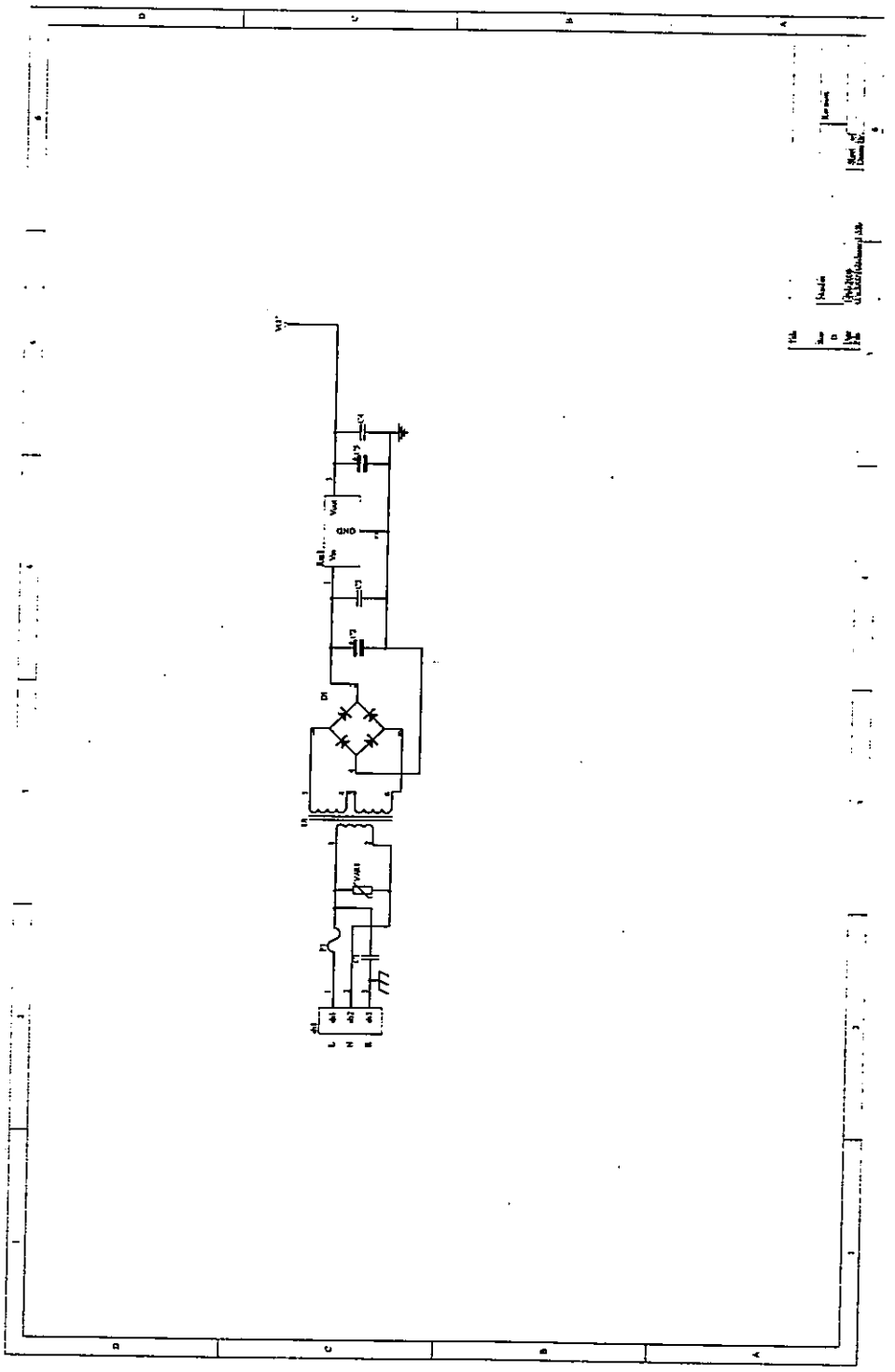
(*) MOV A,ACC is not a valid instruction

Mnemonic		Description	Byte	Cycle
Boolean Variable Manipulation				
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2
Program and Machine Control				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long iump	3	2
SJMP	rel	Short jump (relative addr.)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if carry flag is not set	2	2
JB	bit,rel	Jump if direct bit is set	3	2
JNB	bit,rel	Jump if direct bit is not set	3	2
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2
CJNE	A,direct,rel	Compare direct byte to A and jump if not equal	3	2

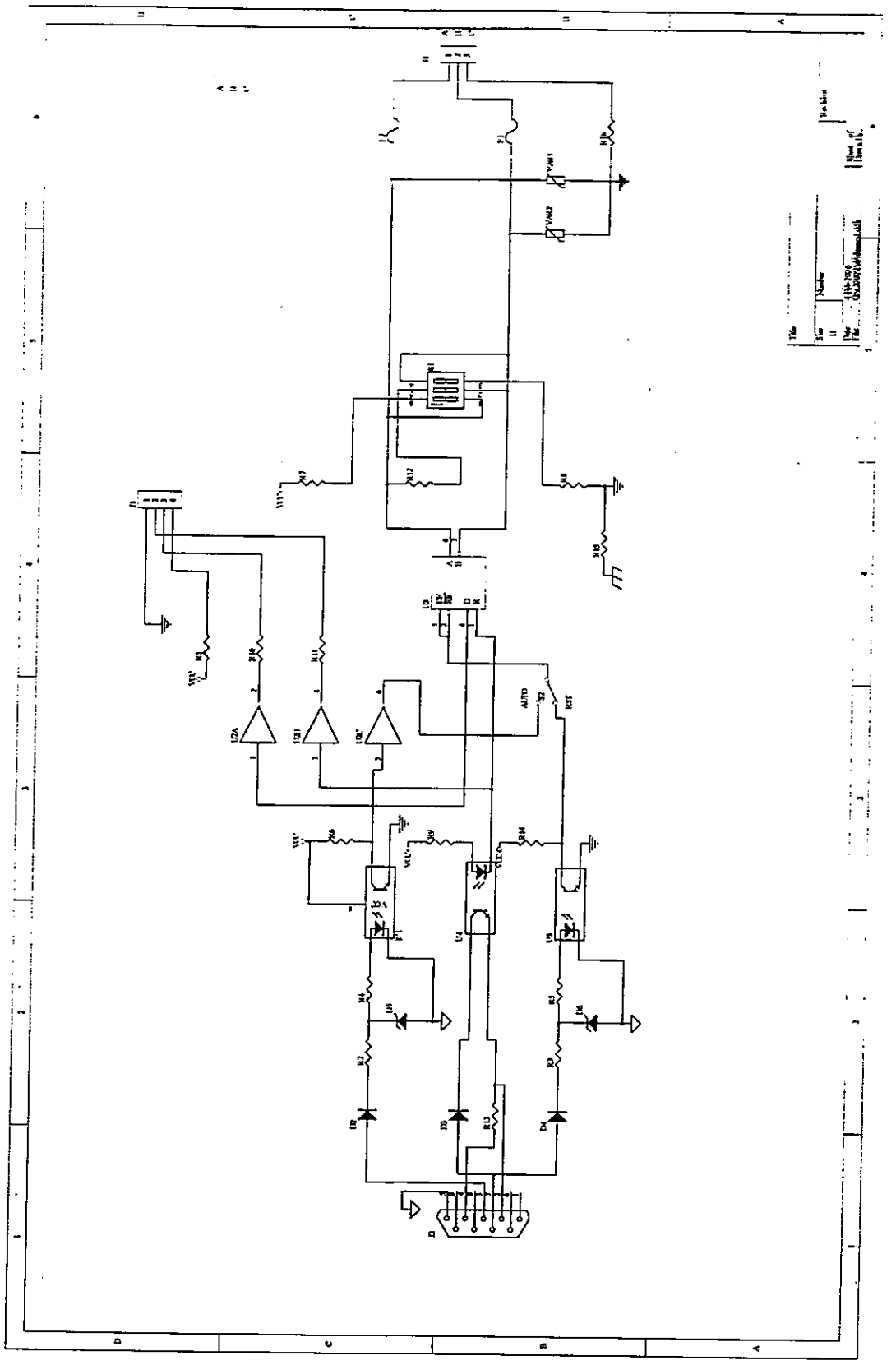
Mnemonic		Description	Byte	Cycle
Program and Machine Control (cont'd)				
CJNE	A,=data,rel	Compare immediate to A and jump if not equal	3	2
CJNE	Rn,#data rel	Compare immed. to reg. and jump if not equal	3	2
CJNE	@Ri,#data,rel	Compare immed. to ind. and jump if not equal	3	2
DJNZ	Rn,rel	Decrement register and jump if not zero	2	2
DJNZ	direct,rel	Decrement direct byte and jump if not zero	3	2
NOP		No operation	1	1

ภาคผนวก ง.

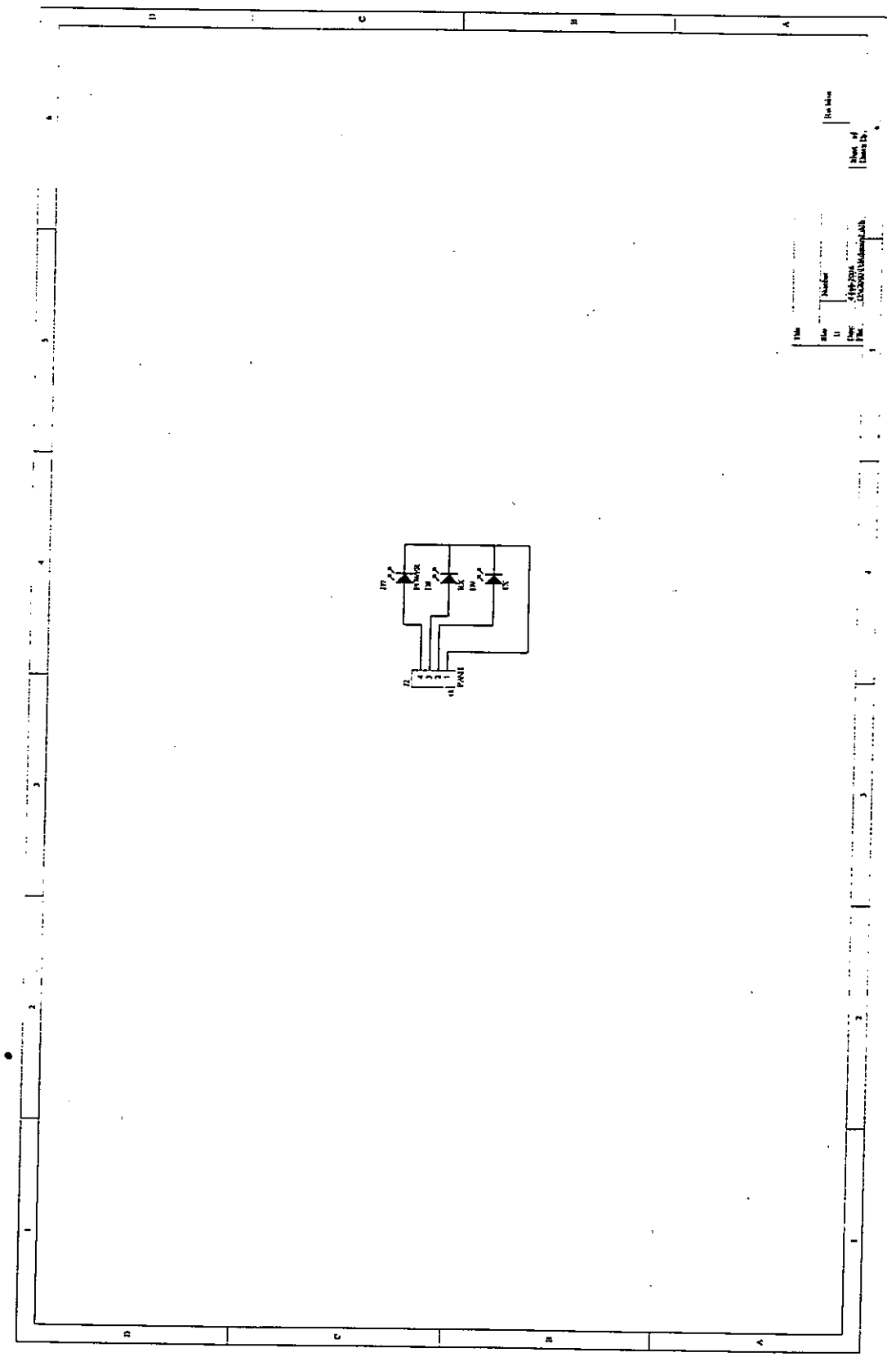
วงจรที่ใช้ในงานวิจัย



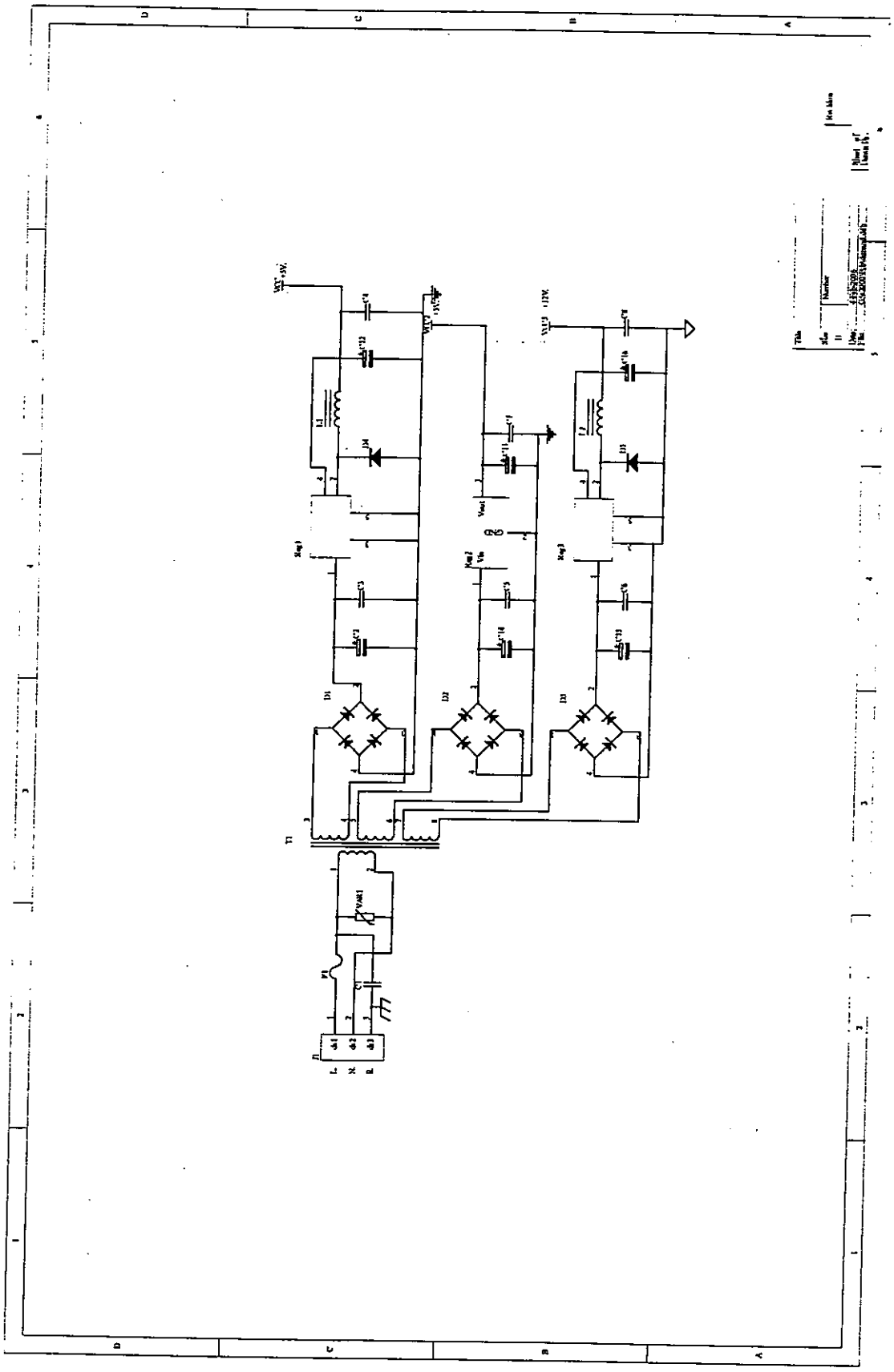
วงจรแหล่งจ่ายไฟของโมดูลแปลง RS232C เป็น RS485 ชนิดแยกกราวด์



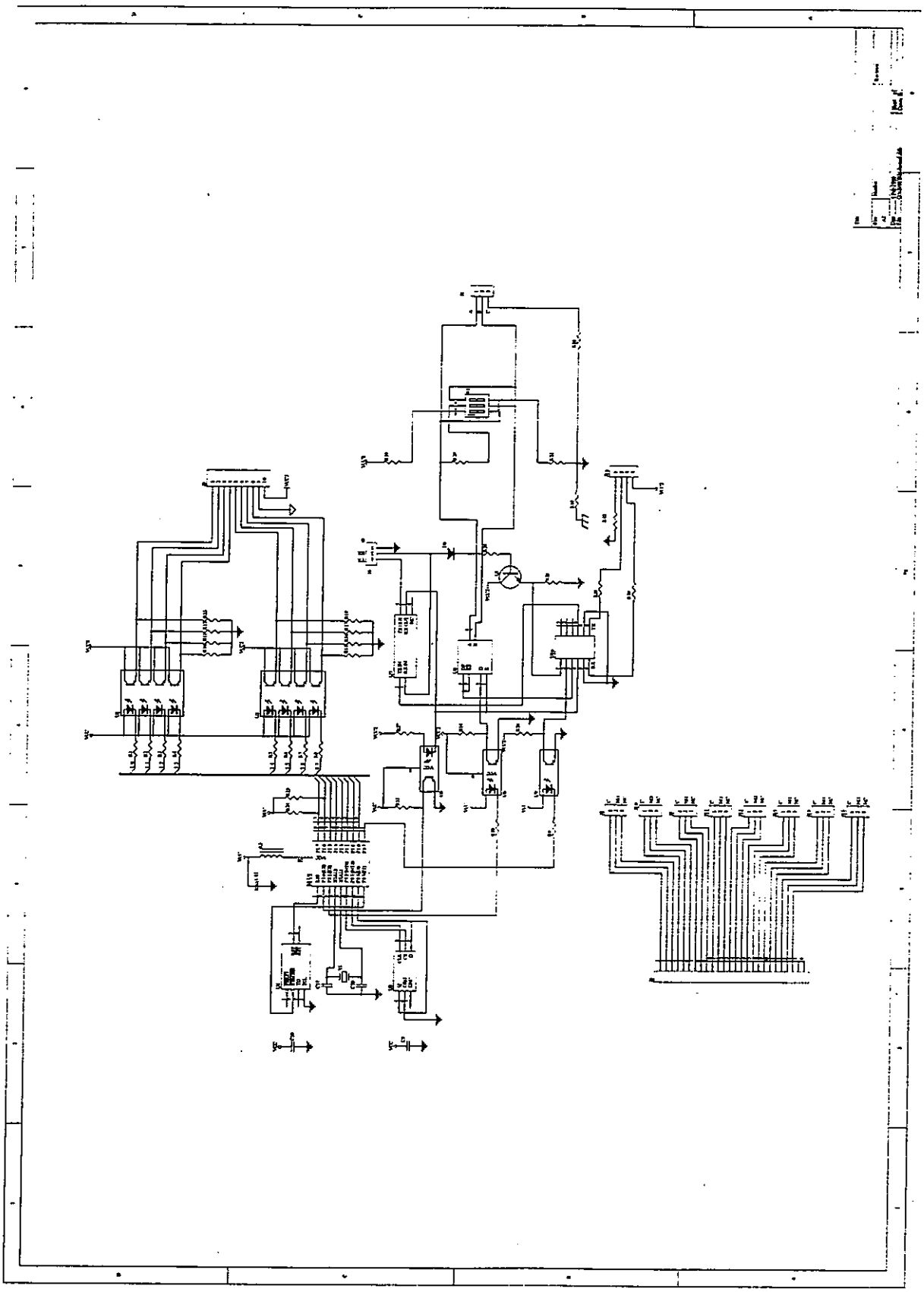
วงจรควบคุมของโมดูลแปลง RS232C เป็น RS485 ชนิดแยกกราวด์



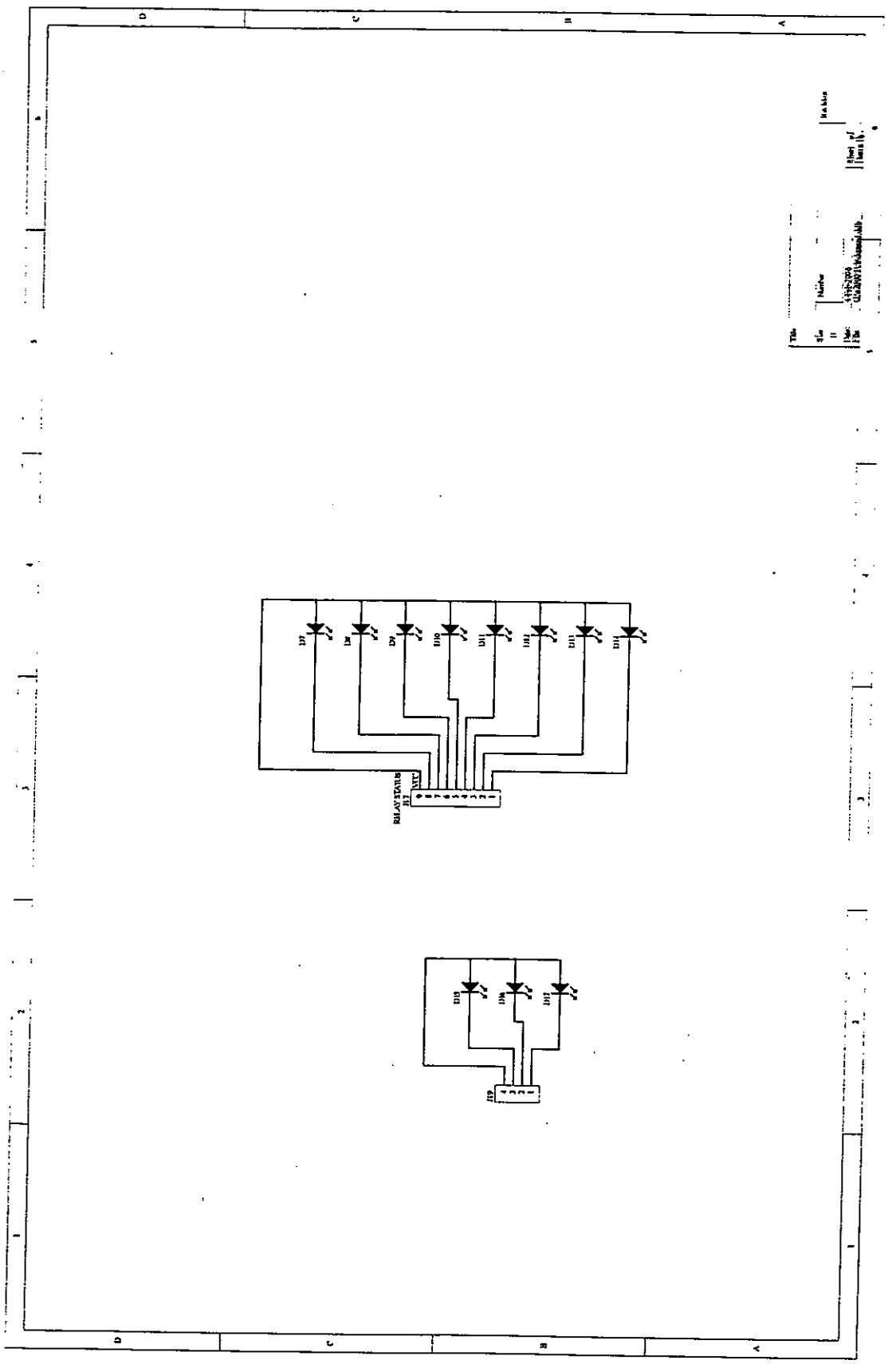
วงจรแสดงผลของโมดูลแปลง RS232C เป็น RS485 ชนิดแยกกราวด์



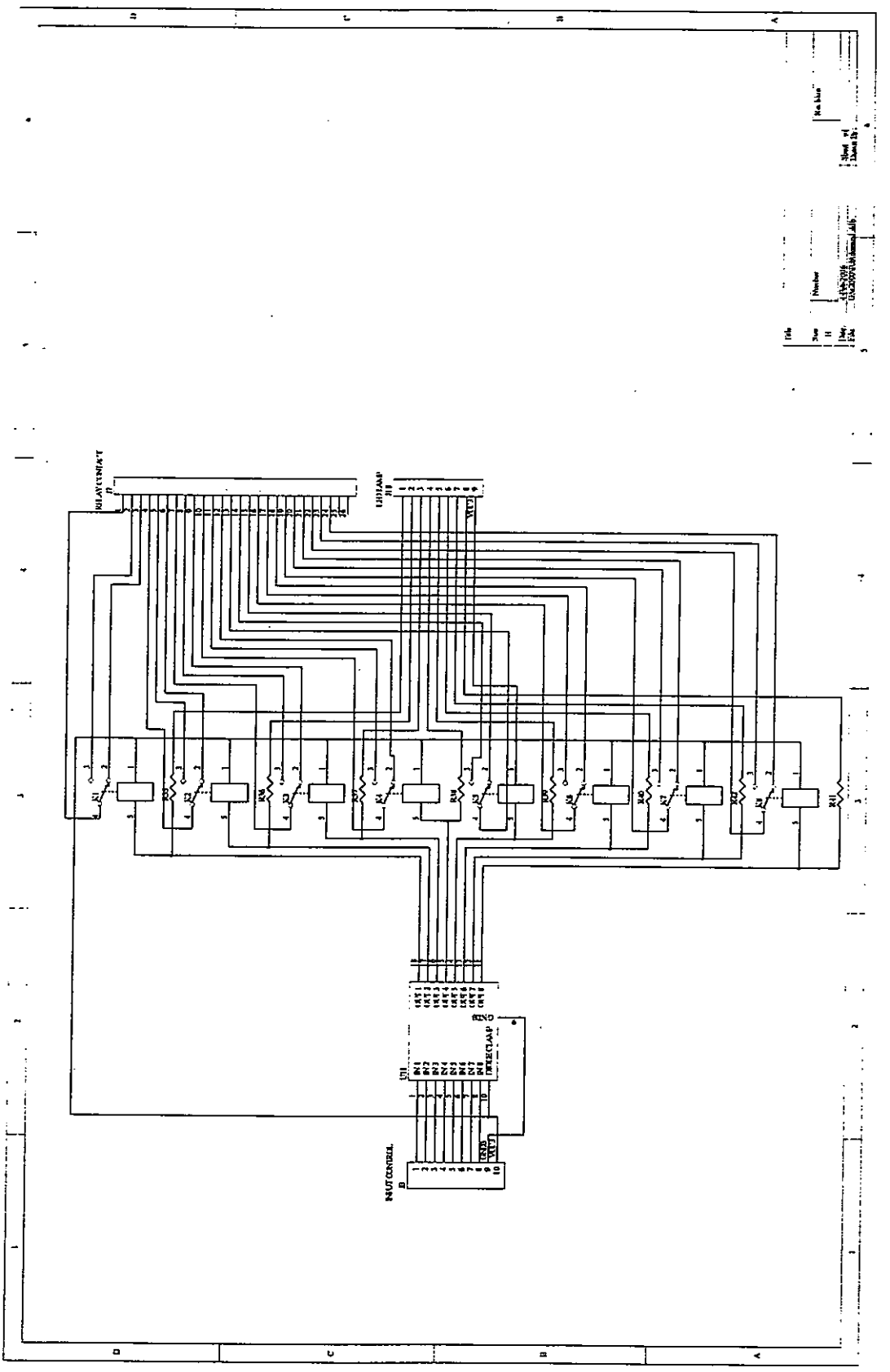
วงจรแหล่งจ่ายไฟของโมดูลรีเลย์ควบคุม



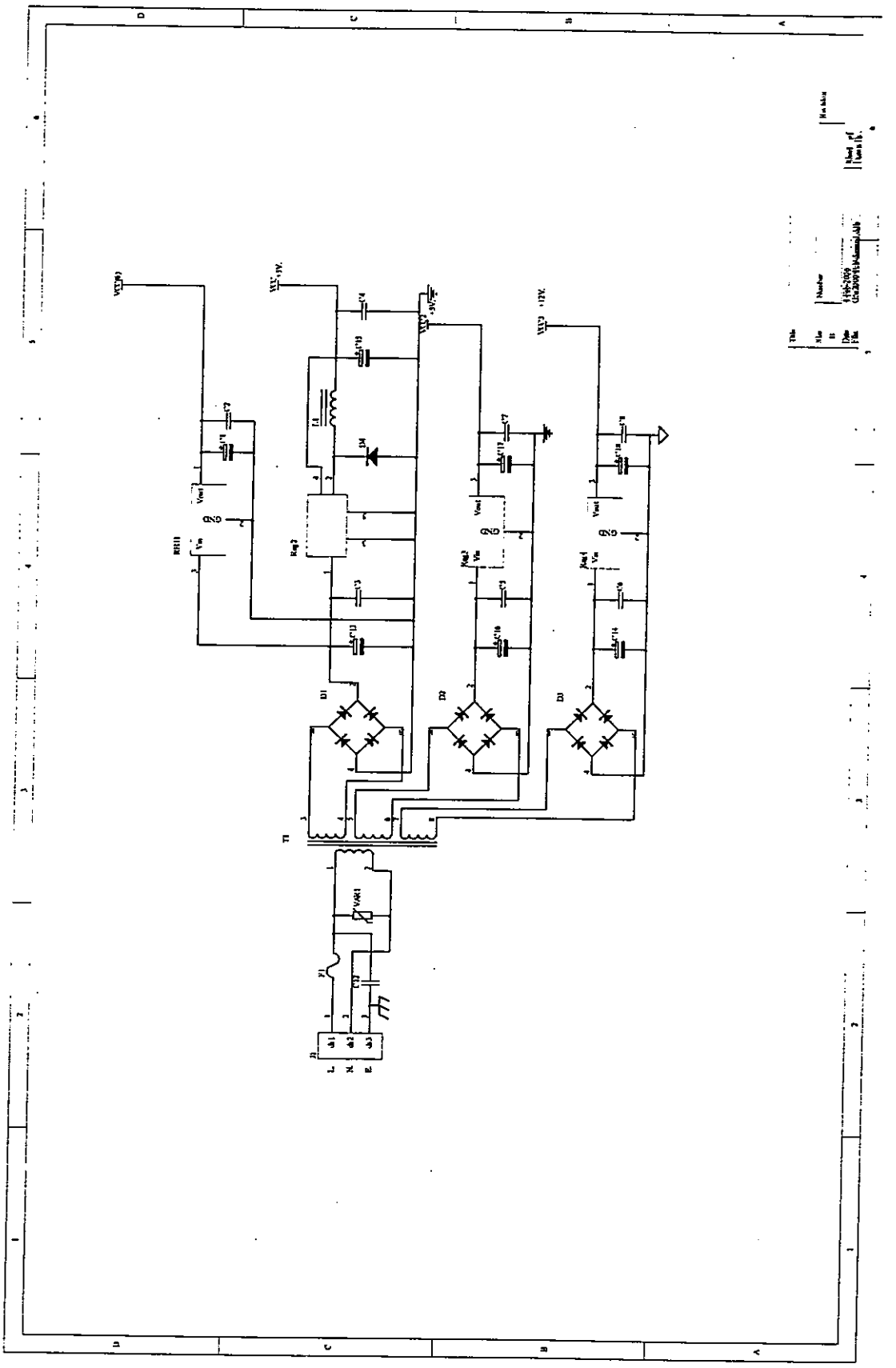
วงจรควบคุมของโมดูลรีเลย์ควบคุม



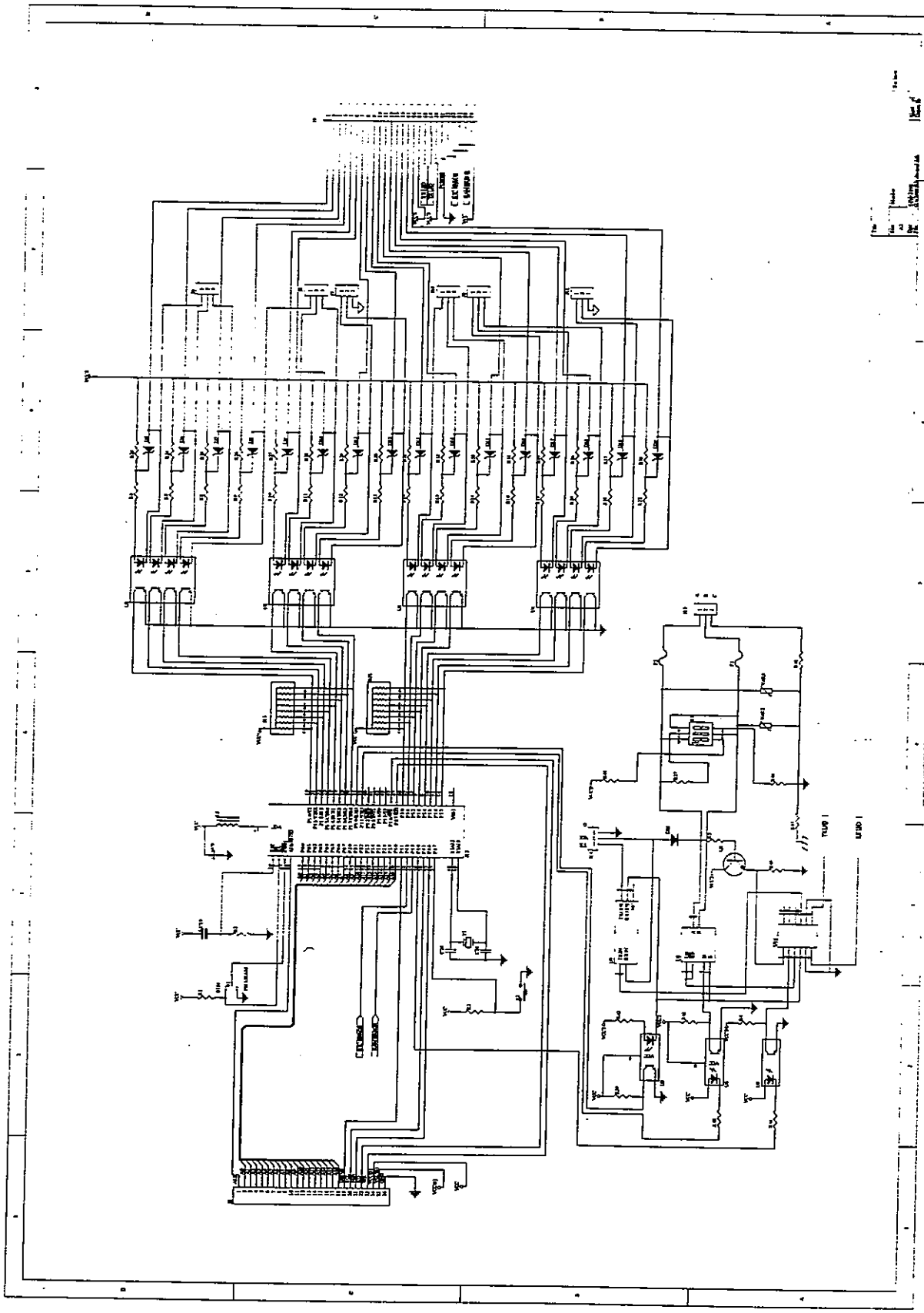
วงจรแสดงผลของโมดูลรีเลย์ควบคุม



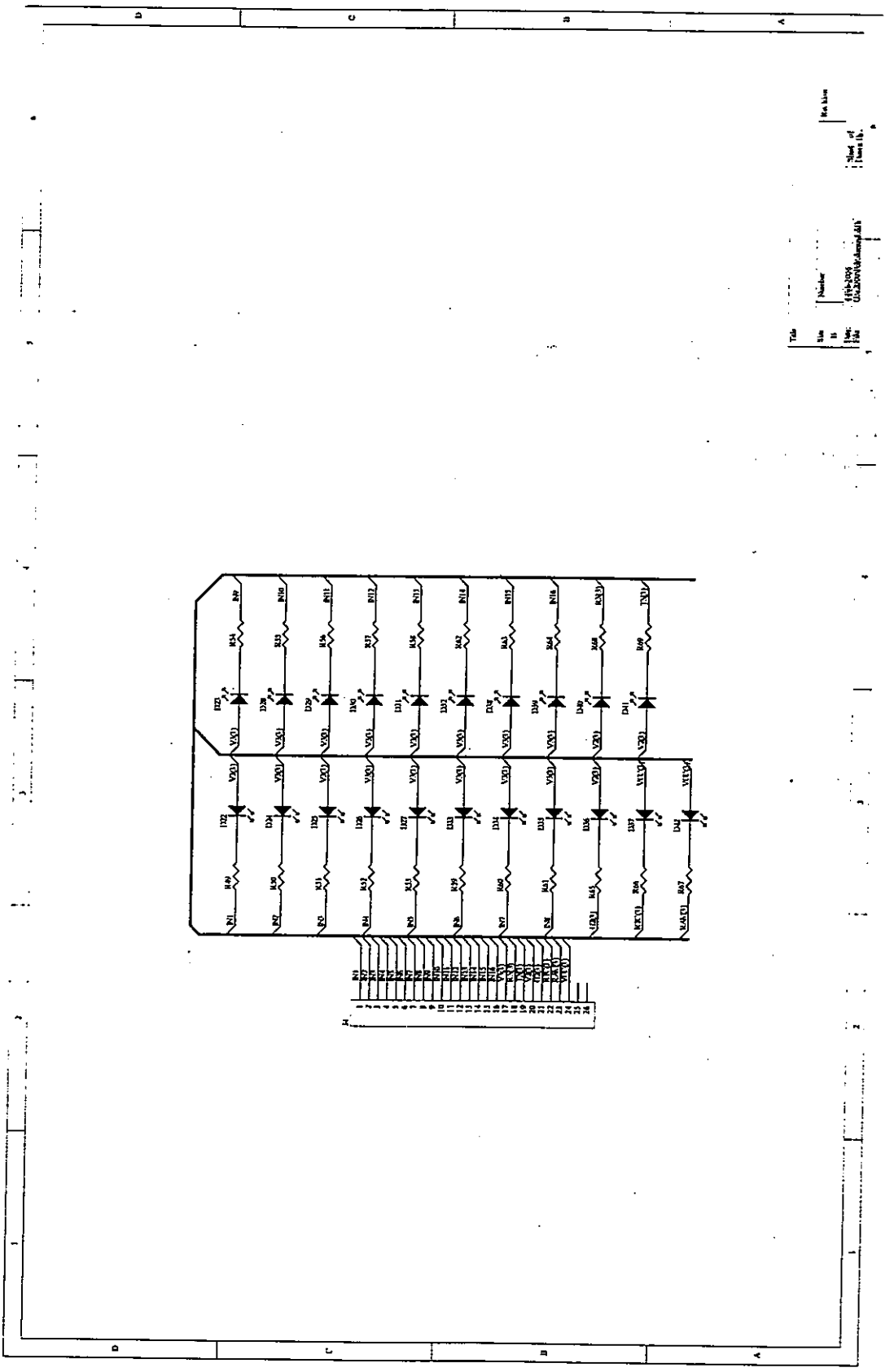
วงจรขับของโมดูลรีเลย์ควบคุม



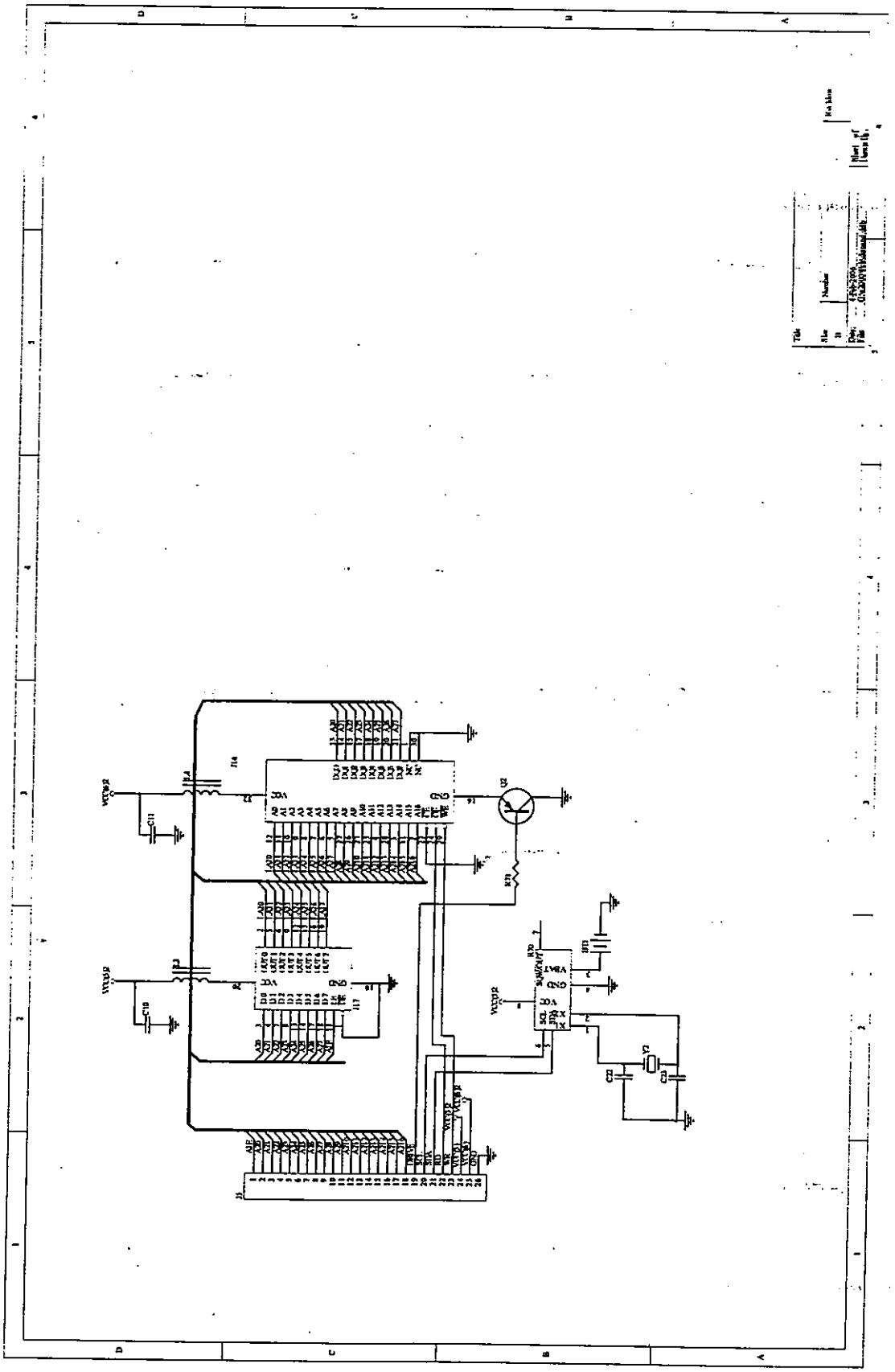
วงจรแหล่งจ่ายไฟของโมดูลตรวจเช็คบันทึกสถานะ



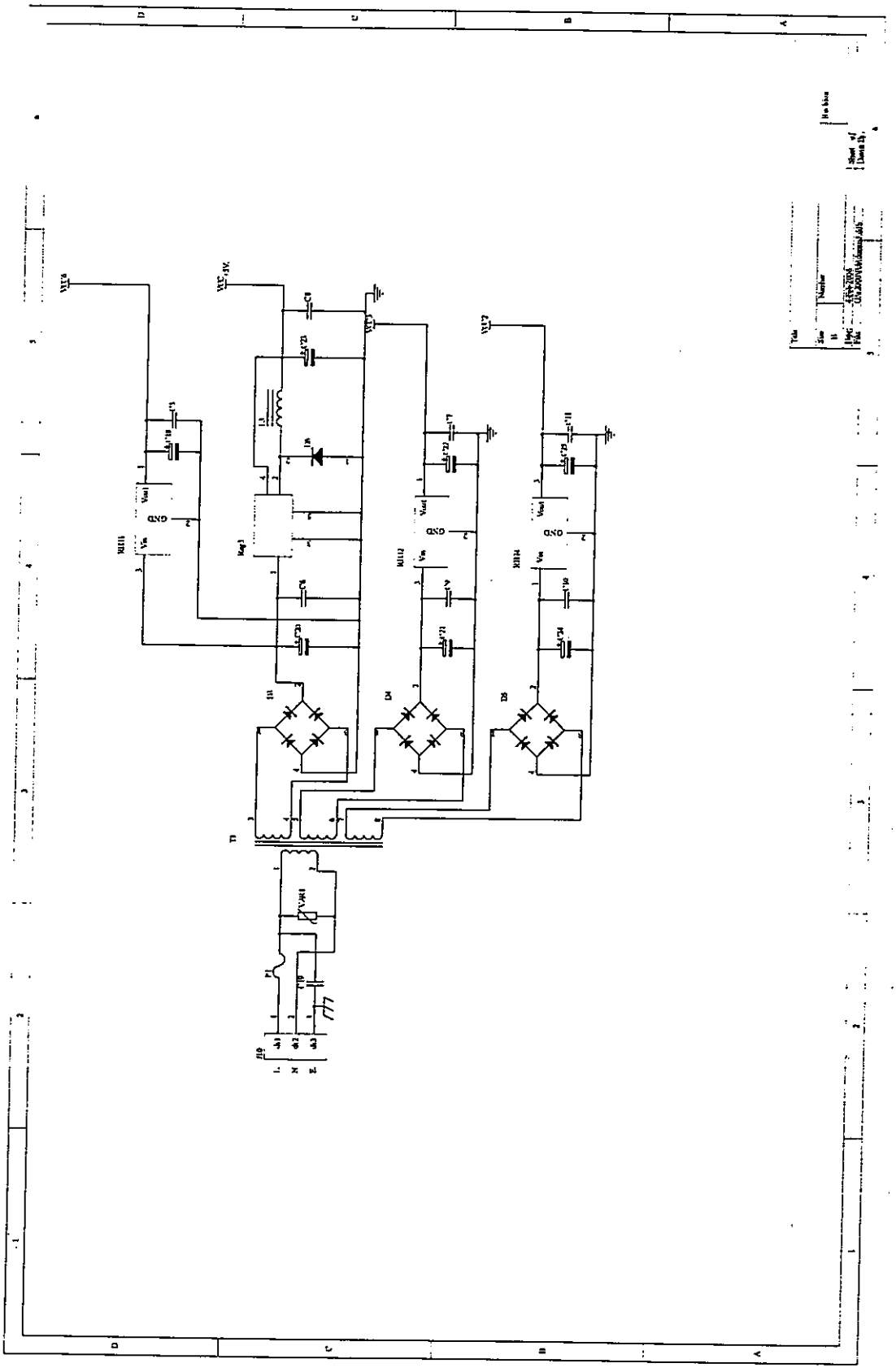
วงจรควบคุมของโมดูลตรวจเช็คบันทึกสถานะ



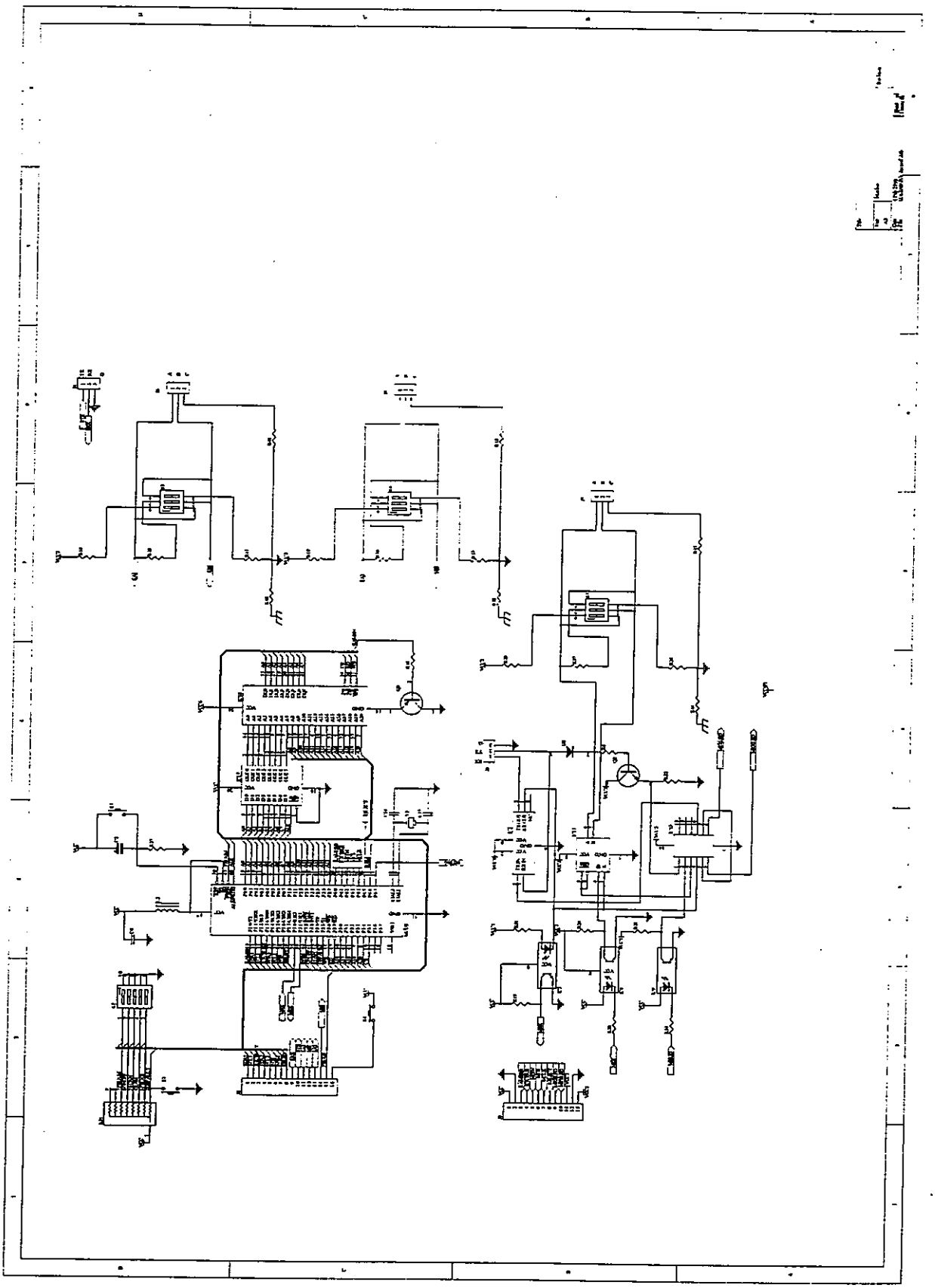
วงจรแสดงผลของโมดูลตรวจเช็คบันทึกสถานะ



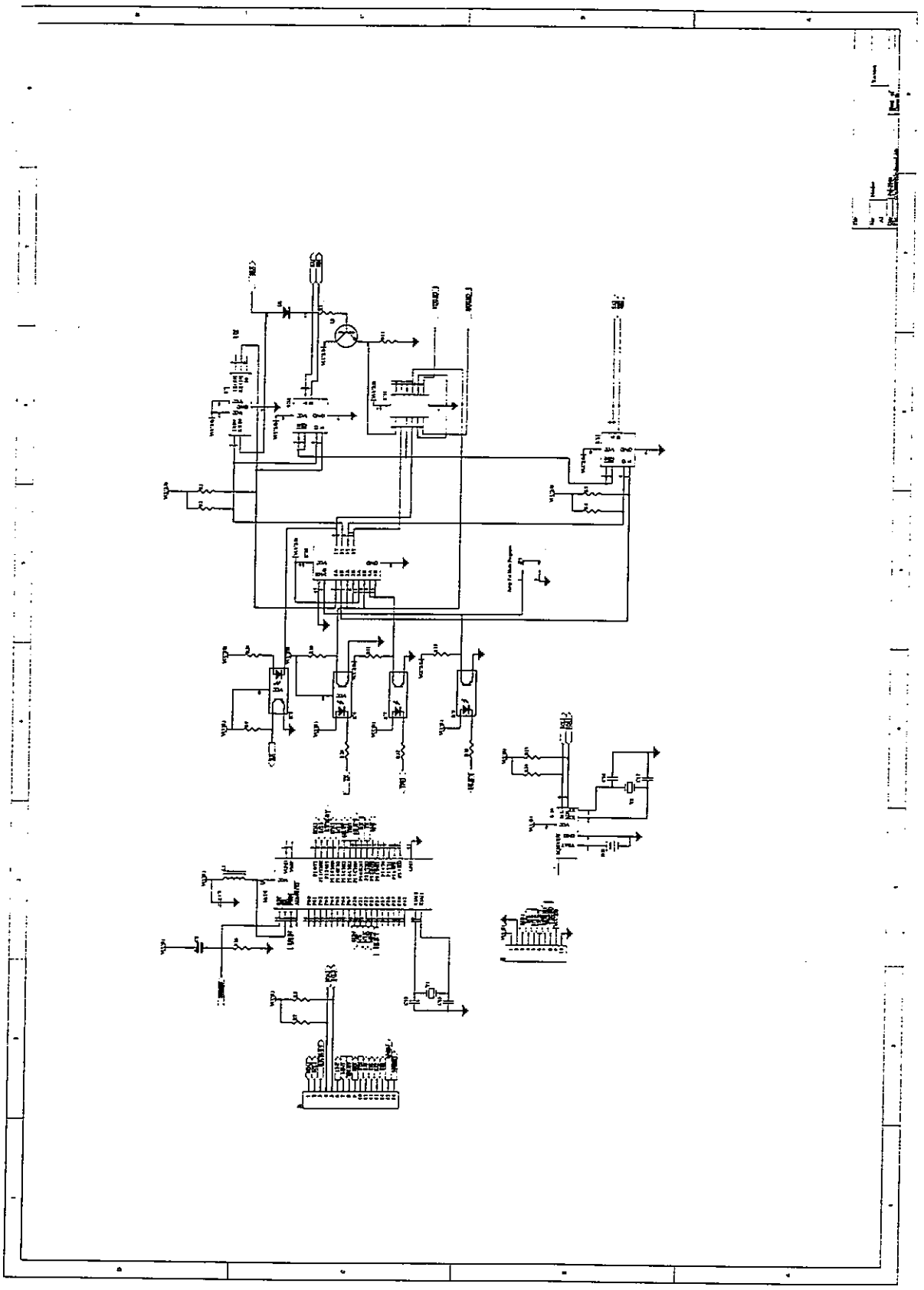
วงจรควบคุมหน่วยความจำ NVRAM ของโมดูลตรวจเช็คบันทึกสถานะ



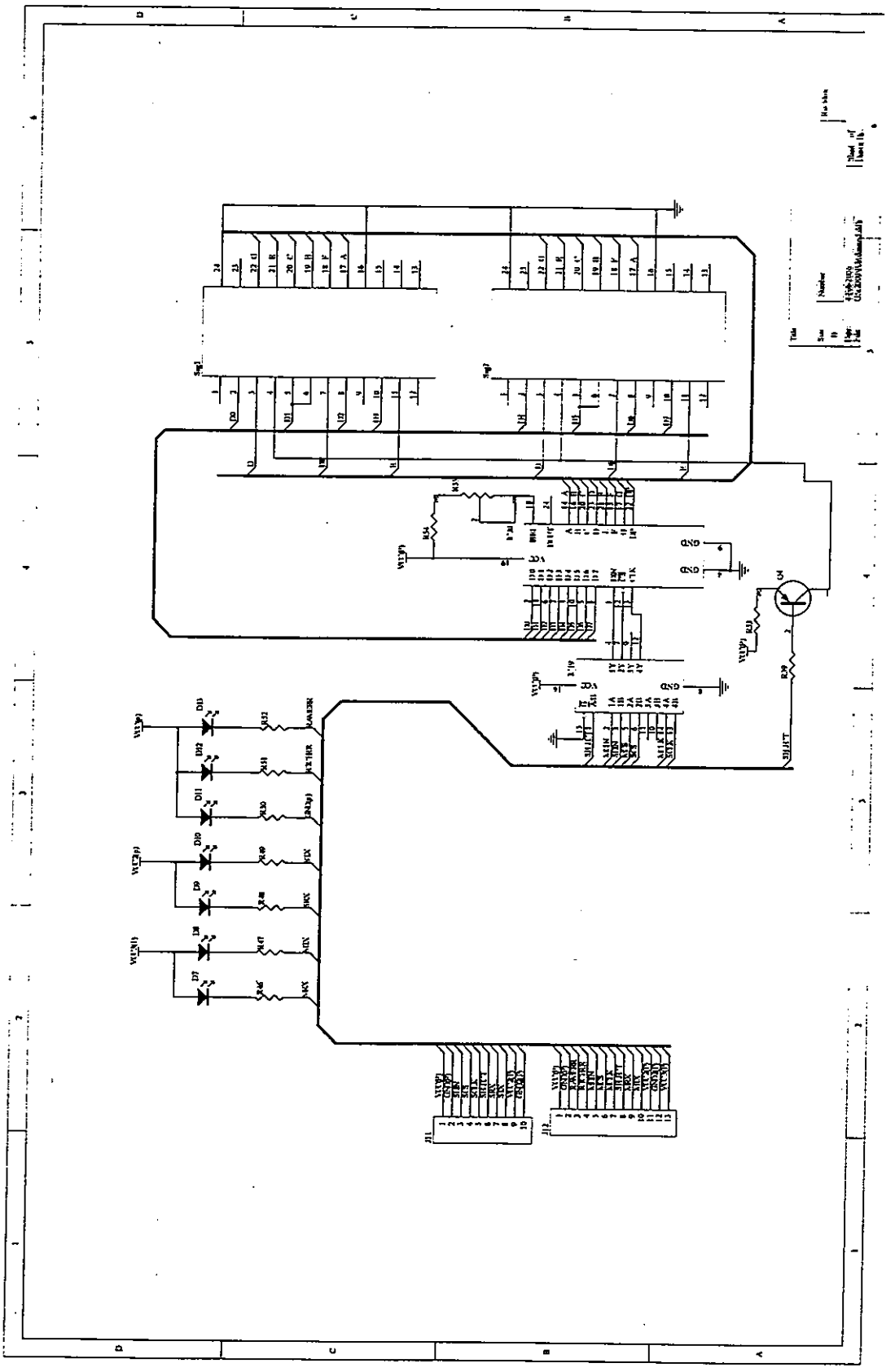
วงจรแหล่งจ่ายไฟของโมดูลควบคุม และบันทึกความต้องการพลังไฟฟ้า



วงจรควบคุม Master ของโมดูลควบคุม และบันทึกความต้องการพลังไฟฟ้า



วงจร Slave ของโมดูลควบคุม และบันทึกความต้องการพลังไฟฟ้า

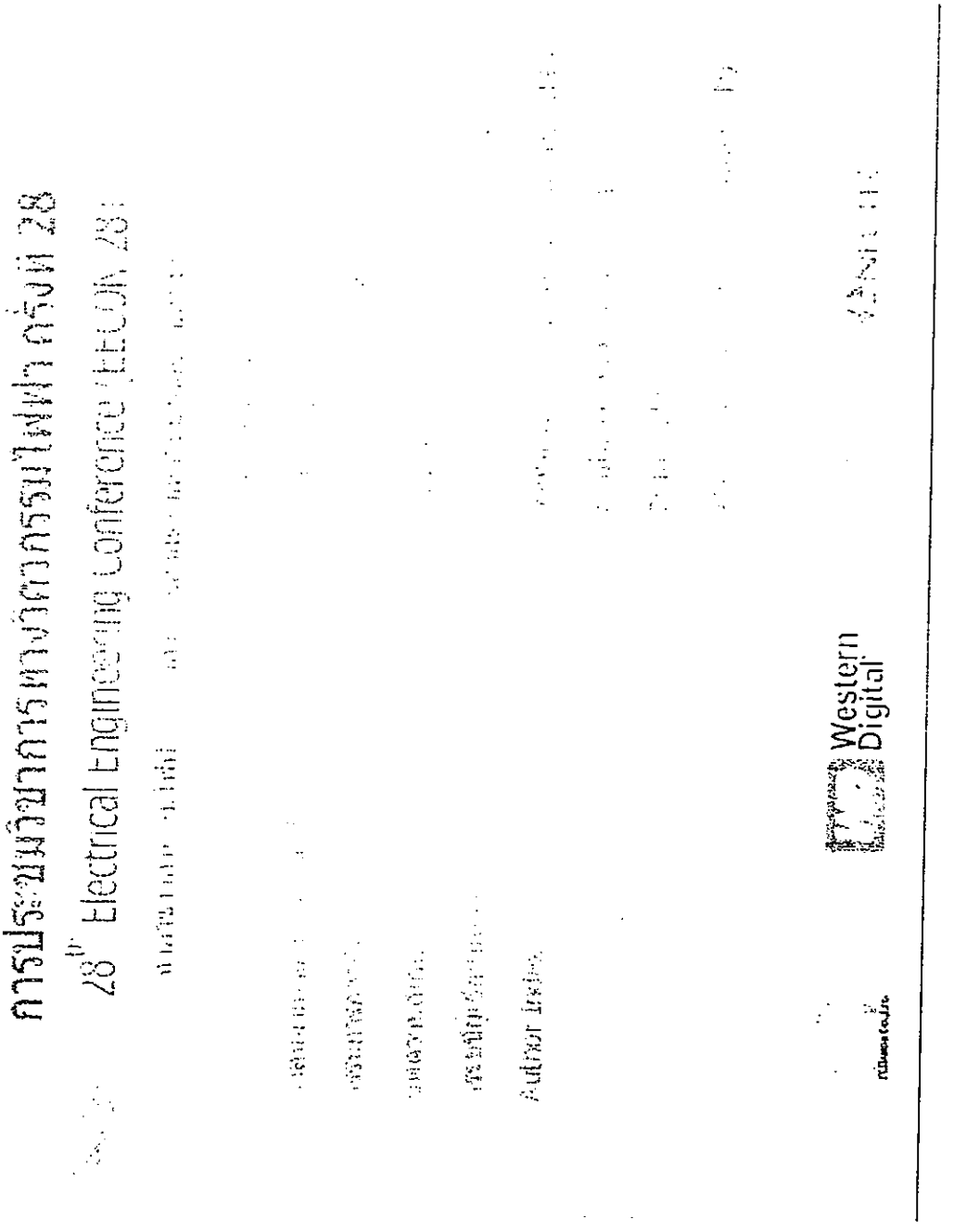


วงจรแสดงผลของโมดูลควบคุม และบันทึกความต้องการพลังไฟฟ้า

ภาคผนวก จ.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. สุพจน์ แก้วกรณ์ และ วิจิตร กิณเรศ. "ระบบควบคุมความต้องการพลังงานไฟฟ้าชาญฉลาด" การประชุมวิชาการทางไฟฟ้าครั้งที่ 28 (EECON-28) 20-21, มหาวิทยาลัยธรรมศาสตร์, 20-21 ตุลาคม 2548, หน้า 641-644



ระบบควบคุมความต้องการพลังไฟฟ้าชาญฉลาด

A Smart Power Demand Controller System

สุพจน์ แก้วกรณ์ วิจิตร กิจเรศ

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เขตลาดกระบัง กรุงเทพฯ 10520

โทร 06-5903258 E-mail: kmin_5@yahoo.com

บทคัดย่อ

บทความนี้ได้นำเสนอ การพัฒนาระบบควบคุมความต้องการพลังไฟฟ้าชาญฉลาด สำหรับการควบคุมความต้องการพลังไฟฟ้าให้เหมาะสม ระบบประกอบด้วย มิเตอร์วัดความต้องการพลังไฟฟ้า, ตัวควบคุม, ชุดรีเลย์, ตัวตรวจเช็คบันทึกสถานะ, ตัวแปลงการสื่อสารอนุกรม และ คอมพิวเตอร์ ระบบจะควบคุมโดยอ่านค่าความต้องการพลังไฟฟ้า จากมิเตอร์ผ่านพอร์ทอนุกรม RS485 มาประมวลผล และวิเคราะห์ จากนั้นควบคุมโหลด ชนิด เปิด-ปิด ตามโปรแกรมที่กำหนด ส่วนคอมพิวเตอร์นั้นจะใช้เก็บบันทึก วิเคราะห์ข้อมูล และ ป้อนคำสั่งควบคุมไปยังตัวควบคุม ตามความต้องการของผู้ใช้

คำสำคัญ: ความต้องการพลังไฟฟ้า, ดาต้าเบส, ตัวประกอบโหลด, แถบฮีสเตอร์รีซีส

Abstract

This paper proposes a development of a smart power demand controller system for optimum power demand control. The system consists of power demand meters, demand controller module, relay module, input sensor module, communication adaptor and PC. The controller reads demand power via power meters using RS-485 port. Then the data is processed and analyzed. The load is controlled based on ON-OFF technique for relay set according to the given program. The PC is used for data record, data analysis, command input according to user requirement.

Keyword: Power demand, Database, Load factor, Hysteresis band, TOU, TOD

บทนำ

ในปัจจุบันเชื้อเพลิงมีราคาสูงขึ้นเป็นผลให้ค่าไฟฟ้ามีราคาสูงขึ้นตามไปด้วย การควบคุมความต้องการพลังไฟฟ้าเป็นอีกหนึ่งหนทางที่จะสามารถลดค่าใช้จ่าย อีกทั้งยังช่วยชาติประหยัดต้นทุนการผลิตไฟฟ้า เนื่องจากพฤติกรรมการใช้ไฟฟ้าในแต่ละช่วงเวลาไม่เหมือนกัน หากค่าความต้องการพลังไฟฟ้าสูงสุดในแต่ละวันมีค่ามากกว่าค่าความต้องการพลังไฟฟ้าเฉลี่ยมาก การไฟฟ้าจะต้องสร้างโรงไฟฟ้าให้เพียงพอกับความต้องการพลังไฟฟ้าสูงสุด แต่จะทำงานเฉพาะช่วงที่มีความต้องการพลังไฟฟ้าสูงสุดเท่านั้น อีกทั้งยังใช้เชื้อเพลิงที่มีราคาสูงกว่าปกติ ทำให้เกิดต้นทุนในการผลิตเพิ่มขึ้น ดังนั้นจึงมีการคิดค่าความต้องการพลังไฟฟ้าสูงสุดกับผู้ใช้ไฟฟ้าด้วย เพื่อชักจูงให้ผู้ใช้ไฟฟ้าปรับเปลี่ยนพฤติกรรมการใช้ไฟฟ้าให้เหมาะสม บทความนี้จึงได้นำเสนอ ระบบควบคุมความต้องการพลังไฟฟ้าชาญฉลาด ที่สามารถควบคุมความต้องการพลังไฟฟ้า, บันทึกข้อมูลทางไฟฟ้า, วิเคราะห์พฤติกรรมของโหลด, คำนวณค่าไฟฟ้าในระบบ TOU และ TOD แสดงผลในรูปกราฟฟิก ฯลฯ เพื่อช่วยควบคุมความต้องการพลังไฟฟ้า ให้เหมาะสม และ ยังสามารถช่วยลดค่าพลังไฟฟ้าได้อีกด้วย ภายในระบบประกอบด้วยโมดูลย่อยๆหลายโมดูล ซึ่งแต่ละโมดูลทำงานอิสระต่อกัน แต่สามารถรับคำสั่งจากภายนอกได้ ทำให้ระบบมีความอ่อนตัวสูง ง่ายต่อการปรับเปลี่ยนจำนวนโมดูลตามความเหมาะสม จึงมีการพัฒนาวิจัยเครื่องต้นแบบเพื่อนำไปใช้งานจริง

2. ทฤษฎี

2.1 การคิดค่าพลังไฟฟ้าสูงสุด

$$D = \frac{\sum_{t=0}^{t=15} P_t \times \Delta T}{15} \qquad (1)$$

การวัดค่าพลังไฟฟ้าสูงสุดจะวัดค่าพลังไฟฟ้าเฉลี่ยของทุกๆช่วงเวลา 15 นาทีอย่างต่อเนื่องตลอดทั้งเดือน ในหนึ่งเดือนจะวัดค่าพลังไฟฟ้าเฉลี่ยของช่วงเวลา 15 นาที ได้ ประมาณ 2,880 ครั้ง แล้วนำค่าที่สูงที่สุดมาคิด

2.2 การควบคุมค่าความต้องการกำลังไฟฟ้าสูงสุดให้เหมาะสม

พิจารณาได้จากตัวประกอบโหลด (Load factor)

$$LF(\%) = \frac{E \times 100\%}{Dm \times h} \qquad (2)$$

ค่าความต้องการกำลังไฟฟ้าสูงสุดที่เหมาะสมหาได้จากค่า LF ที่เหมาะสม

$$Dm^{*} = \frac{E \times 100^{\circ}_{0}}{LF^{*} \times h} \qquad (3)$$

โดยที่

D คือ ค่าพลังไฟฟ้าเฉลี่ยของช่วงเวลา 15 นาที

P_r คือ ค่าพลังไฟฟ้าชั่วขณะ

ΔT คือ ช่วงเวลาที่ใช้ไฟฟ้า

LF คือ ตัวประกอบโหลด

LF* คือ ตัวประกอบโหลดที่เหมาะสม

E คือ จำนวนหน่วยพลังงานที่ใช้ในรอบ 1 เดือน (กิโลวัตต์-ชั่วโมง)

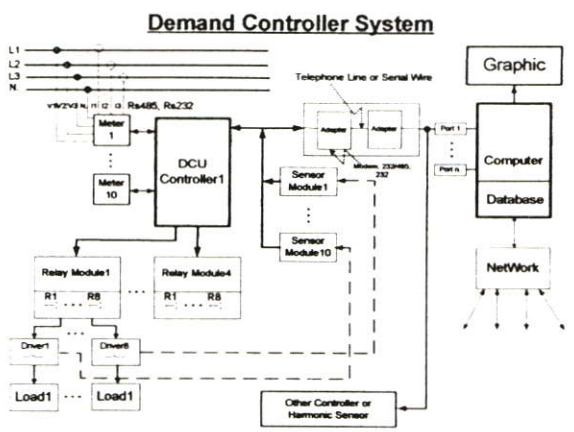
Dm คือ ความต้องการพลังไฟฟ้าสูงสุด

Dm* คือ ความต้องการพลังไฟฟ้าที่เหมาะสม

h คือ จำนวนชั่วโมงที่ทำงานในเดือนนั้น

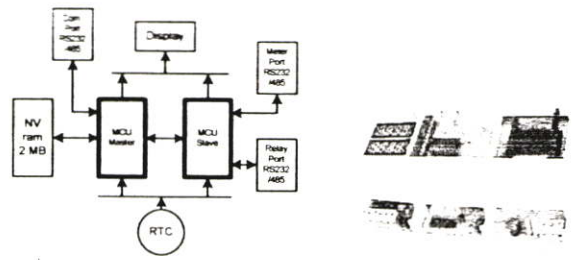
3. การออกแบบ

การออกแบบระบบจะออกแบบเป็นส่วนๆ ของโมดูลย่อย แต่ละโมดูลสามารถรับคำสั่งการทำงานได้ แล้วนำโมดูลมาประกอบกันเป็นระบบตามความเหมาะสมของระบบที่ต้องการควบคุม ดังรูปที่ 1 จะเห็นว่าการควบคุมเริ่มจาก ตัวควบคุม(DCU) อ่านค่าความต้องการพลังไฟฟ้าจากมิเตอร์แต่ละตัวมารวมกันแล้วนำมาเปรียบเทียบกับเวลา และ ค่าความต้องการพลังไฟฟ้าที่จะต้องควบคุมถ้าหากอยู่ในช่วงเวลาที่ต้องการควบคุม และ มีค่าความต้องการพลังไฟฟ้าต่ำหรือสูงเกินกว่าค่าที่กำหนดไว้จะทำการประมวลผลหารีเลย์เพื่อทำการตัด-ต่อ ตามโปรแกรมที่ตั้งไว้ ถ้าหากต้องการควบคุมเพียงอย่างเดียวมีเพียง 3 ส่วนก็เพียงพอ แต่ถ้า ต้องการบันทึก วิเคราะห์ และแสดงผลข้อมูลก็สามารถเพิ่ม ตัวตรวจเช็คบันทึกสถานะร่วมกับคอมพิวเตอร์ เข้ามาเพื่อเก็บข้อมูล ระบบยังรองรับการสื่อสารผ่านสารโทรศัพท์ด้วยการใช้โมเด็มเป็นตัวส่งผ่านข้อมูลในกรณีต้องการเข้าถึงข้อมูลจากระยะไกล การสื่อสารข้อมูลในแต่ละส่วนจะมีรูปแบบ (Protocol) เฉพาะ



รูปที่ 1 ระบบควบคุมความต้องการพลังไฟฟ้า

3.1 ตัวควบคุมความต้องการพลังไฟฟ้า



รูปที่ 2 บล็อกไดอะแกรม และ โมดูลควบคุมความต้องการพลังไฟฟ้า

ตัวควบคุม มี 2 หน่วยประมวลผล(Microcontroller) ร่วมกันประมวลผล แบ่งเป็น ตัวหลัก(Master) และ ตัวรอง(Slave) แยกกันทำงานคนละฝั่ง ตัวหลักจะทำงานติดต่อสื่อสารกับ คอมพิวเตอร์ และอ่านข้อมูลจากตัวรองมาเก็บในหน่วยความจำ NV Ram รอการเรียกขอจากคอมพิวเตอร์ ส่วนตัวรองจะทำหน้าที่อ่านค่าความต้องการพลังไฟฟ้าจากมิเตอร์ภายนอกมาประมวลผลแล้วส่งสัญญาณไปควบคุมชุดรีเลย์ตามโปรแกรมที่ตั้งไว้

3.2 ชุดรีเลย์

ทำหน้าที่รับคำสั่งจากภายนอกเพื่อควบคุมรีเลย์ 8 ตัว คำสั่งที่ป้อนเข้ามาจะต้องมีรูปแบบ(Protocol) ที่ถูกต้องจึงจะสามารถทำงานได้

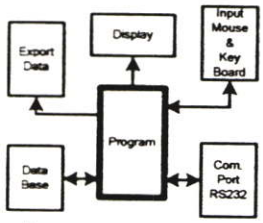
3.3 ตัวตรวจเช็คบันทึกสถานะ

ตัวตรวจเช็คบันทึกสถานะจะใช้บันทึกสถานะของโหลดตามความเป็นจริง โดยจะบันทึกได้ 16 บิต เมื่อเกิดการเปลี่ยนแปลงของ อินพุท บิตใดบิตหนึ่ง ก็จะบันทึกสถานะขณะนั้นกับเวลาปัจจุบันทันที การเก็บข้อมูลในหน่วยความจำเป็นลักษณะวนเก็บซ้ำไปเรื่อยๆ หากข้อมูลเต็มก็จะลบข้อมูลเก่าสุดทิ้งแล้วแทนที่ด้วยข้อมูลใหม่

3.4 ตัวแปลงการสื่อสารอนุกรม RS232 เป็น RS 485

เป็นวงจรแปลง มาตรฐานการสื่อสารอนุกรม RS232 เป็น RS 485 ชนิด แยกกราวด์ เพื่อลดสัญญาณรบกวน และ สามารถส่งข้อมูลได้ไกลขึ้นสูงสุดถึง1.2 กิโลเมตร สามารถต่ออุปกรณ์ได้ 32 จุด บนบัสเดียวกัน หรือ 256 จุด หากใช้ IC พิเศษบางเบอร์ ถ้าหากต้องการสื่อสารข้อมูลให้ไกลกว่านั้นก็สามารถใช้โมเด็มแทนได้

3.5 โปรแกรมคอมพิวเตอร์

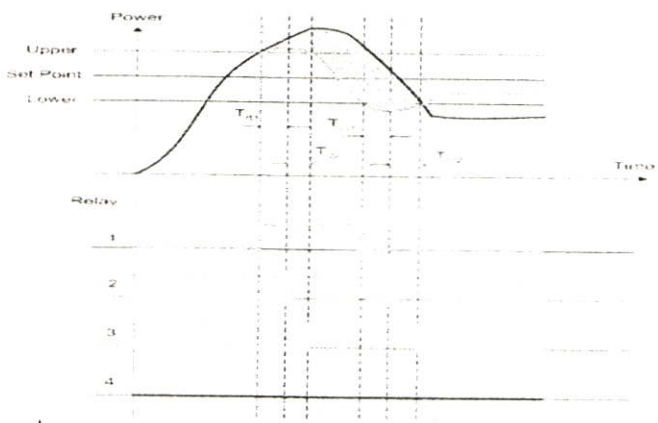


รูปที่ 3 บล็อกไดอะแกรม ของโปรแกรมคอมพิวเตอร์

คอมพิวเตอร์ จะทำหน้าที่สร้าง และ ส่งข้อมูลควบคุมไปเก็บไว้ในหน่วยความจำของตัวตัวควบคุม เพื่อให้ทำงานตามที่กำหนด พร้อมทั้งอ่านข้อมูลมาบันทึก วิเคราะห์ และแสดงผล การเก็บข้อมูลจะจัดเก็บไว้ในฐานข้อมูลแยกเก็บไว้เป็นรายเดือนเพื่อให้ง่ายต่อการใช้งาน

4 การทดลองและผลการทดลอง

การทดลองทั้งหมดมีหลายส่วน ตั้งแต่ทดลองติดต่อสื่อสารกันด้วยรูปแบบ(Protocol) ต่างๆ การทดลองต่างๆในขณะออกแบบวงจร ฯลฯ แต่เนื่องจากข้อจำกัดของบทความ ในบทความนี้จึงขอกล่าวถึงเฉพาะการทดลองควบคุมโหลดตามค่าความต้องการพลังไฟฟ้าที่กำหนดในช่วงเริ่มวัดค่าความต้องการพลังไฟฟ้า และ ควบคุมต่อเนื่องจากการวัดก่อนหน้านั้น โดยมีรูปแบบการควบคุมดังนี้



รูปที่ 4 แสดงการควบคุมความต้องการพลังไฟฟ้าที่มีการควบคุมชนิด FIFO

จากรูปที่ 4 มีการตั้งค่าควบคุมความต้องการพลังไฟฟ้าไว้ ณ จุดตั้งค่าควบคุม(Set Point) แต่การควบคุมรีเลย์ นั้นจะควบคุมด้วยแถบฮีสเตอร์รีซีส(Hysteresis Band) คือ เริ่มทำงานเมื่อค่าความต้องการพลังไฟฟ้าสูงหรือต่ำกว่าแถบควบคุม Upper และ Lower บวกด้วยค่าเวลาหน่วงเพื่อป้องกันไม่ให้มีการทำงานกลับไปมาในช่วงเวลาใกล้กัน T_d และ T_{ct}

ตารางที่1 แสดงการทดลองควบคุมความต้องการพลังงานที่ค่าความต้องการพลังไฟฟ้าเริ่มต้น 0 W.

Time	Demand	Relay1	Relay2	Relay3	Relay4
0	0	ON	ON	ON	ON
5	134	ON	ON	ON	ON
7	172	OFF	ON	ON	ON
8	190	OFF	OFF	ON	ON
9	210	OFF	OFF	OFF	OFF
15	210	OFF	OFF	OFF	OFF

จากตารางที่1 โหลดเริ่มต้น 400 W. รีเลย์แต่ละตัวควบคุมโหลด 100 W. ตั้งค่าความต้องการพลังไฟฟ้า ณ จุดควบคุมไว้ที่ 150 W. ค่าฮีสเตอร์รีซีส 10% ค่าเวลาหน่วง 1 นาทีทั้งตัดและต่อ รีเลย์จะเริ่มตัดโหลดหลังจากค่าความต้องการพลังไฟฟ้าสูงกว่า 165 W. ไปแล้ว 1 นาที ทำให้ค่าความต้องการพลังไฟฟ้าสูงเกินแถบควบคุมไปแล้ว 1 นาที ก่อนเริ่มตัด และ จะเริ่มต่อโหลดหลังจากค่าความต้องการพลังไฟฟ้าต่ำกว่า 135 W. และ เกินเวลา 1 นาที หลังจากการตัดหรือต่อครั้งสุดท้ายไปแล้ว จะเริ่มต่อตัวถัดไปหลังจากนั้นอีก 1 นาที จากตารางค่าความต้องการพลังไฟฟ้าจะยังคงเพิ่มขึ้นเรื่อยๆในลักษณะบวกสะสม เมื่อมีการตัดโหลดอัตราเร็วในการเพิ่มขึ้นของค่าความต้องการพลังไฟฟ้าสะสมสูงสุดก็ลดลง ในเวลาที่ 9 นาที มีการตัดโหลดจนหมดแต่ค่าความต้องการพลังไฟฟ้าสูงสุดไม่ลดลงเพราะเป็นผลบวกสะสมในช่วงเริ่มต้นต้องบวกให้ครบ 15 นาที ถึงแม้ว่าจะไม่มีการใช้พลังงานก็ตาม ดังนั้นการตั้งค่าควบคุมจะต้องคำนึงถึงจุดนี้ด้วย เพราะอาจจะทำให้เกิดการตัดของรีเลย์ทุกตัว

ตารางที่2 ผลการทดลองควบคุมความต้องการพลังงานที่ค่าความต้องการพลังไฟฟ้าเริ่มต้น 200 W.

Time	Demand	Relay1	Relay2	Relay3	Relay4	
X	200	ON	ON	ON	ON	
X+5	261	OFF	ON	ON	ON	←
X+6	261	OFF	OFF	ON	ON	
X+7	254	OFF	OFF	OFF	ON	
X+8	240	OFF	OFF	OFF	OFF	
X+12	193	ON	OFF	OFF	OFF	←
X+13	193	ON	ON	OFF	OFF	
X+14	200	ON	ON	ON	OFF	
X+15	213	ON	ON	ON	ON	←

จากตารางที่ 2 โหลดเริ่มต้น 400 W. รีเลย์แต่ละตัวควบคุมโหลด100 W. ตั้งค่าความต้องการพลังไฟฟ้า ณ จุดควบคุมไว้ที่ 230 W. ค่าฮีสเตอรีซีส 10% ค่าเวลาหน่วง 1 นาที ทั้งตัดและต่อ กรณีนี้เป็นกรณีที่มีการเปิดใช้งานมาแล้วมากกว่า 15 นาที ทำให้มีค่าความต้องการพลังไฟฟ้าสูงสุดของหน่วยนี้คงอยู่เท่ากับ 200 W. ค่าความต้องการพลังไฟฟ้าต่อไปนี้จะคิดจากการบวกค่าพลังไฟฟ้าใหม่ พร้อมทั้งลบค่าเก่าสุดออก แล้วหารด้วยเวลา 15 นาที รีเลย์จะเริ่มตัดโหลดหลังจากค่าความต้องการพลังไฟฟ้าสูงกว่า 253 W. ไปแล้ว 1 นาที (ที่เวลา X+5) และ จะเริ่มต่อโหลดหลังจากค่าความต้องการพลังไฟฟ้าต่ำกว่า 207 W. หลังจากมีการตัดหรือต่อเกิน 1 นาที ไปแล้ว(ที่เวลา X+12) จะเห็นว่าตัวควบคุมพยายามรักษาค่าความต้องการพลังไฟฟ้าให้อยู่ภายในแถบฮีสเตอรีซีส การเลือกขนาดโหลดให้รีเลย์ควบคุมมีผลต่อการเข้าสู่สภาวะคงตัวของการควบคุม (ที่เวลา X+12) หากไม่เหมาะสมแล้วจะเกิดการแกว่ง เช่น ตัดและต่อในเวลาต่อมาซ้ำไปเรื่อยๆ ซึ่งวิธีการเลือกขนาด และ การจัดการกับโหลดนี้อยู่ในเรื่องการวิเคราะห์โหลด[2] ผู้เขียนจะจัดพิมพ์บทความต่อไป

5. สรุป

การควบคุมค่าความต้องการพลังไฟฟ้าให้เหมาะสมตามช่วงเวลาต่างๆจะทำให้เกิดประสิทธิภาพสูงสุด ช่วยลดค่าไฟฟ้าได้สูงสุด และส่งผลกระทบกับการใช้ไฟฟ้าน้อยที่สุด เนื่องจากจะช่วยลดทั้งค่าความต้องการพลังไฟฟ้า และ ค่าพลังไฟฟ้า แต่การควบคุมนี้จะต้องเกิดจากตัวควบคุมที่มีข้อมูลในการควบคุมที่เหมาะสมจึงจะสามารถจัดการกับโหลดให้ถูกต้องได้ในช่วงเวลาต่างๆ หากข้อมูลการควบคุมผิดพลาดก็จะส่งผลเสียกับระบบงานที่ต้องใช้ไฟฟ้าได้ ข้อมูลที่ว่านี้ได้มาจาก สถิติที่เก็บรวบรวมไว้ในฐานข้อมูล และ การจัดเรียงลำดับความสำคัญของโหลดแต่ละจุดในแต่ละช่วงเวลาว่าจุดไหนตัดได้ในช่วงเวลาไหน จากความชำนาญของผู้ดูแลระบบ ซึ่งระบบควบคุมความต้องการพลังไฟฟ้าขาญฉลาดที่ได้นำเสนอนี้ มีความสามารถที่จะบันทึก วิเคราะห์ และ ป้อนโปรแกรมเพื่อควบคุมให้เกิดประโยชน์สูงสุด

6. กิตติกรรมประกาศ

คณะผู้วิจัยขอขอบคุณ บ.แอมพ์ทรอนอินทรูเมนท์ ประเทศไทยจำกัด ที่ได้สนับสนุนทุน และ อุปกรณ์ในการทำวิจัย เว็บไซต์ต่างๆทั้งไทย และ ต่างประเทศ การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย สำหรับข้อมูล

เอกสารอ้างอิง

[1] Ming Yuan Cho, Shih Wei Gau, Cha Win Huang, "Development of Microprocessor Based Demand Control System for Industrial and Commercial Customer," IEEE 2001.

[2] A.J. Hoffman,"Peak demand control in commercial buildings with target peak adjustment base on load forecasting," Proceedings of the 1998 IEEE International Conference on Control Application Trieste, Italy 1-4 September 1998.

[3] Richard E. Stephens, "Load Control Demand Reduction Estimation," IEEE Transaction on Power Syatem, Vol. 3, No. 1, February 1988.

[4] Dr W C Beattie and Dr P McCafferty, "The Use of a forecasting algorithm to provide improved maximum Demand Control, "

[5] A. Martins, H. Jorge, J. Mota, R. Parracho and A. Gomes, "A PC-Based Simulation Package for Supporting End-User Demand Side Management ," IEEE Transitions on Power System, Vol. 6 No. 3, August 1991.

[6] http://www.9Engineer.com

[7] การไฟฟ้านครหลวง(กฟน) http://www.mea.or.th

ประวัติผู้เขียนบทความ

สุพจน์ แก้วกรณ์ สำเร็จการศึกษาวิศวกรรมศาสตร์ สาขาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยเกษมบัณฑิต ปัจจุบันกำลังศึกษาต่อระดับปริญญาโท สาขาวิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วิจิตร กิณเรศ สำเร็จการศึกษาระดับปริญญาตรี (เกียรตินิยมอันดับ 2) และปริญญาโท สาขาวิศวกรรมไฟฟ้า จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง Ph.D. จาก University of Nottingham, UK. ปัจจุบันดำรงตำแหน่งรองศาสตราจารย์ประจำภาควิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง งานวิจัยที่สนใจ Power Electronics and Electric Drives.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายสุพจน์ แก้วกรณ์
วัน เดือน ปี เกิด	28 เมษายน พ.ศ.2521 ที่จังหวัดพิษณุโลก
การศึกษา	สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษมบัณฑิต ในปีการศึกษา 2543 และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2544
ความชำนาญเฉพาะด้าน	1.) วงจรอิเล็คทรอนิคส์ 2.) ระบบไฟฟ้า 3.) ไมโครคอนโทลเลอร์ MCS-51 4.) การเขียนโปรแกรมด้วย Delphi 5.) ระบบฐานข้อมูลระดับ Desktop 6.) การสื่อสารข้อมูล