



เครื่องควบคุมกำลังงานไฟฟ้าด้วยไมโครคอนโทรลเลอร์

POWER CONTROL BY MICROCONTROLLER

โดย

นาย จีราวุธ วารินทร์

นาย อัครวิทย์ ชวนะพงศ์

วัน เดือน ปี.....พ.ศ. ๒๕๖๑
เลขทะเบียน.....๐๓๔๗๒๗
เลขเรียกหนังสือ.....ท ๒๖๐๒๗.๖๖

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๖๑

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034727

ปีการศึกษา 2537

POWER CONTROL BY MICROCONTROLLER

โดย

นาย จีราวุธ วารินทร์ 34102065

นาย อัครัช ชวนะพงศ์ 34109494

อาจารย์ที่ปรึกษา

อาจารย์ เกียรติวรรณ ทรงสัตย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมกำลังงานไฟฟ้าด้วยไมโครคอนโทรลเลอร์

ผู้จัดทำ :

1. นาย จีราวุธ วารินทร์
2. นาย อัคคิธ ชวนะพงศ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมกำลังงานไฟฟ้าด้วยไมโครคอนโทรลเลอร์

นาย จีราวุธ วารินทร์ 34102065

นาย อัครัช ชวนะพงศ์ 34109494

อาจารย์ เกียรติวรรณ ทรงสตัย อาจารย์ที่ปรึกษา

บทคัดย่อ

โครงการนี้เป็นการควบคุมพลังงานไฟฟ้าที่จ่ายให้กับโหลดที่เป็นความต้านทาน บริสุทธิโดยใช้ไมโครโปรเซสเซอร์ 8031 เป็นตัวควบคุมการทำงานของวงจรทั้งหมด โดย วงจรนี้จะเป็นตัวควบคุมโวลต์เตจที่จ่ายให้กับโหลดโดยใช้หลักการควบคุมเฟส (PHASE CONTROL) การควบคุมสามารถทำงานโดยโปรแกรมที่เราตั้งไว้ล่วงหน้า หรือควบคุม ณ เวลานั้น ผลที่ออกมาจะถูกบันทึกเก็บไว้ในหน่วยความจำซึ่งสามารถเรียกออกมาดูได้ภายหลัง และสามารถแสดงสถานะต่างๆของระบบออกทางจอ LCD การส่ง สัญญาณควบคุมจะส่งเป็นความถี่ซึ่งมีความถูกต้องสูง ไม่ถูกรบกวนจากสภาวะแวดล้อมภายนอก

ABSTRACT

The purpose of this project is to create system used to control the power energy that should be send to load,the pure resistance.The system is present in from of circuit that uses Microcontroller 8031 to control all of functions.By the Phase Control Method,Voltage from the generator is controlled by the frequency signal to transmit to load.This kind of signal is high accuracy and not distrubed by environment.The control process is setting by programed that will be execute automatically or set by manual.Then the result will be recorded in memory that can be loaded and displayed after the process finished.The status of system is display on the LCD monitor

สารบัญ	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
- หลักการควบคุมพลังงานไฟฟ้า	
- ไมโครคอนโทรลเลอร์ 8031	
บทที่ 3 การสร้างและการทำงานของโครงงาน	13
- หลักการทำงานของเครื่องควบคุม	
- ANT-32 EMBEDDED CONTROL BOARD	
- การควบคุมพลังงานไฟฟ้าด้วย TCA785	
- การเปลี่ยนความถี่เป็นแรงดันไฟฟ้า	
- การเปลี่ยนแรงดันไฟฟ้าเป็นความถี่	
- การเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก	
- การเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล	
- การวัดกระแสและแรงดัน	
- การเชื่อมต่อด้วยแสง (Opto isolator)	
- การเชื่อมต่อด้วย Pulse Transformer	
- การเชื่อมต่อกับระบบไมโครคอนโทรลเลอร์	
- การแสดงผลทางจอ LCD	
- โปรแกรมควบคุม	
บทที่ 4 การทดลองและผลการทดลอง	50
บทที่ 5 บทวิจารณ์และสรุป	75
ภาพผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

สารบัญรูปภาพ	หน้า
รูปที่ 2.1 เป็นโครงสร้างใหญ่ๆของ 8051	4
รูปที่ 2.2 สถาปัตยกรรมศาสตร์ภายในของ 8051	7
รูปที่ 2.3 ไดอะแกรมขาของ 8051 แบบ DIP	8
รูปที่ 3.1 แสดงส่วนประกอบทั้งหมดของวงจร	15
รูปที่ 3.2 แสดงลักษณะภายในของ TCA 785	21
รูปที่ 3.3 แสดงลักษณะสัญญาณที่ขาต่างๆของ TCA 785	22
รูปที่ 3.4 แสดงวงจรควบคุมพลังงานด้วย TCA 785	23
รูปที่ 3.5 เปลี่ยนสัญญาณดิจิตอลเป็นอนาล็อก	25
รูปที่ 3.6 เปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล	26
รูปที่ 3.7 แสดงวงจร Voltage to Frequency Converter	27
รูปที่ 3.8 แสดงวงจร Frequency to Voltage Converter	28
รูปที่ 3.9 วงจร Precision Full - Wave rectifier	30
รูปที่ 3.10 การเชื่อมต่อด้วย ออปโตไอโซเรเตอร์	31
รูปที่ 3.11 เชื่อมต่อด้วย Pulse Transformer	32
รูปที่ 3.12 แสดงเมนูหลักของระบบ	44
รูปที่ 3.13 แสดงหน้าจอของโหมดตั้งเวลา	45
รูปที่ 3.14 แสดงหน้าจอของโหมด Manual	45
รูปที่ 3.15 แสดงหน้าจอของโหมด Program	46
รูปที่ 3.16 แสดงหน้าจอของโหมด Display Data	46
รูปที่ 3.17 รูปแบบการเก็บข้อมูลในหน่วยความจำสำรอง	47
รูปที่ 4.1 วงจรเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก ในการทดลองครั้งที่ 1	50
รูปที่ 4.2 วงจรเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก ในการทดลองครั้งที่ 2	51

สารบัญรูปภาพ	หน้า
รูปที่ 4.22 วงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 2	65
รูปที่ 4.23 สัญญาณเอาต์พุตวงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 3	66
รูปที่ 4.24 วงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 4	67
รูปที่ 4.25 สัญญาณเอาต์พุตวงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 4	67
รูปที่ 4.26 วงจรเปลี่ยนความถี่เป็นแรงดัน	68
รูปที่ 4.27 แสดงวงจรซัมมิงแอมป์ (Summing Amp)	71



สารบัญรูปภาพ	หน้า
รูปที่ 4.3 วงจรโวลต์เตจเรกูเลเตอร์ของไฟลอปที่เลี้ยงไอซี DAC0808	52
รูปที่ 4.4 วงจรเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก เมื่อใส่วงจรอินเวอร์ตเตอร์ที่เอาต์พุต	53
รูปที่ 4.5 วงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล	54
รูปที่ 4.6 วงจรออสซิลเลเตอร์ โพลีทีฟฟีดแบค (OSC POSITIVE FEEDBACK)	55
รูปที่ 4.7 ลักษณะสัญญาณออกจากวงจร ออสซิลเลเตอร์ โพลีทีฟฟีดแบค	55
รูปที่ 4.8 วงจรพรีซีชันฟูลเวฟเรคตีไฟลด์	56
รูปที่ 4.9 รายละเอียดของ LM 1485	56
รูปที่ 4.10 รูปเปรียบเทียบสัญญาณอินพุตและเอาต์พุตจาก วงจรพรีซีชันฟูลเวฟเรคตีไฟลด์	57
รูปที่ 4.11 รูปเปรียบเทียบสัญญาณอินพุตและเอาต์พุตจาก วงจรพรีซีชันฟูลเวฟเรคตีไฟลด์	58
รูปที่ 4.12 วงจรพรีซีชันฟูลเวฟเรคตีไฟลด์ที่สมบูรณ์	59
รูปที่ 4.13 สัญญาณเอาต์พุตจากวงจรพรีซีชันฟูลเวฟเรคตีไฟลด์	59
รูปที่ 4.14 วงจรควบคุมพลังงานด้วย TCA785	60
รูปที่ 4.15 สัญญาณเอาต์พุตจากวงจรควบคุมพลังงาน	61
รูปที่ 4.16 วงจรเชื่อมต่อดัวยพัลส์ทราานฟอร์เมอร์	62
รูปที่ 4.17 สัญญาณเอาต์พุตที่พัลส์ทราานฟอร์เมอร์	62
รูปที่ 4.18 วงจรเปลี่ยนแรงดันเป็นความถี่	63
รูปที่ 4.19 สัญญาณเอาต์พุตวงจรเปลี่ยนแรงดันเป็นความถี่	64
รูปที่ 4.20 วงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 1	65
รูปที่ 4.21 สัญญาณเอาต์พุตวงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 1	65

สารบัญตาราง	หน้า
ตารางที่ 3.1 ค่า CONTROL CODE ของ 8255	18
ตารางที่ 3.2 แสดง COMMAND BYTE ในการเขียนและอ่านค่า CLOCK และ RAM	20
ตารางที่ 4.1 แสดงผลการทดลองของวงจรเปลี่ยนสัญญาณ ดิจิตอลเป็นสัญญาณอนาล็อก	52
ตารางที่ 4.2 แสดงผลการทดลองส่งค่าควบคุมจากไมโครคอนโทรลเลอร์	70
ตารางที่ 4.3 แสดงผลการทดลองส่งค่าควบคุมจากไมโครคอนโทรลเลอร์ เมื่อมีวงจรขั้วมิงแอมป์	72
ตารางที่ 4.4 แสดงผลการทดลองการวัดแรงดัน	73
ตารางที่ 4.5 แสดงผลการทดลองการวัดกระแส	73

บทที่ 1

บทนำ

ในโรงงานอุตสาหกรรมต่างๆ โดยทั่วไป จะประกอบด้วยเครื่องมือเครื่องจักร รวมทั้งระบบต่างๆที่ใช้ในกระบวนการผลิต การควบคุมระบบต่างๆ ให้เป็นไปตามต้องการเป็นสิ่งสำคัญมากเพื่อให้ได้ผลผลิตที่ดีที่สุด รวดเร็วที่สุดและต้นทุนต่ำที่สุด กำลังงานไฟฟ้าที่จ่ายให้แก่โหลดเพื่อทำงานต่าง ๆ เช่น เครื่องให้ความร้อน(Heater) หรือ มอเตอร์(Motor) เป็นสิ่งหนึ่งที่ต้องการการควบคุม พลังงานให้เป็นไปตามต้องการ

โครงการนี้เป็น การควบคุมพลังงานไฟฟ้าโดยใช้ไมโครโปรเซสเซอร์ 8031 เป็นตัวควบคุมการทำงานของวงจรมีวัตถุประสงค์เพื่อใช้ควบคุมพลังงานไฟฟ้าในระบบที่เป็นความต้านทานบริสุทธิ์ โดยวงจรนี้จะเป็นตัวควบคุมโวลต์เดจที่จ่ายให้กับโหลดซึ่งในที่นี้ ให้การควบคุมสามารถทำงานโดยโปรแกรมที่เราตั้งไว้ล่วงหน้า หรือควบคุม ณ เวลานั้น ผลที่ออกมาจะถูกบันทึกเก็บไว้ภายในหน่วยความจำซึ่งสามารถเรียกออกมาดูได้ภายหลัง และสามารถแสดงสถานะต่างๆเช่น กระแส,แรงดันไฟฟ้า,กำลังงานที่ให้กับระบบ และ เวลาปัจจุบันของระบบออกทางจอ LCD

การรับส่งสัญญาณระหว่างวงจรควบคุมและวงจรไมโครคอนโทรลเลอร์จะ ใช้ความถี่เป็นตัวส่งซึ่งจะมีความถูกต้องสูงและไม่โดนรบกวนจากสภาวะแวดล้อมภายนอก

บทที่ 2

ทฤษฎีและหลักการ

2.1 หลักการควบคุมพลังงานไฟฟ้า

พลังงานไฟฟ้าเป็นพลังงานรูปแบบหนึ่งที่นิยมใช้ในปัจจุบันเราสามารถที่จะควบคุมพลังงานไฟฟ้าที่ส่งให้แก่โหลดได้โดยการจำกัดหรือควบคุมตัวแปรตัวใดตัวหนึ่งจากกฎของโอห์ม

$$V = IR \quad \dots(1)$$

$$P = IV \quad \dots(2)$$

$$P = I^2 R \quad \dots(3)$$

เมื่อให้ V คือค่าความต่างศักย์

I คือกระแส

R คือค่าความต้านทานของโหลด

P คือกำลังงานไฟฟ้า

เมื่อโหลดเป็นความต้านทานบริสุทธิ์คือไม่มีค่าความเหนี่ยวนำ (L) และตัวเก็บประจุ (C) จะพบว่าค่าพลังงานขึ้นกับตัวแปรสองตัวคือกระแสที่ให้กับโหลดและความต่างศักย์ที่ตกคร่อมโหลด แต่การที่เราจะควบคุมกระแสที่ให้กับโหลดนั้นทำได้ยากและซับซ้อนต้องใช้อุปกรณ์ที่มีขนาดใหญ่

เราจึงหันมาสนใจการควบคุมความต่างศักย์แทนซึ่งใช้อุปกรณ์น้อยและประหยัดกว่ามาก วิธีที่นิยมใช้มากที่สุดคือ การควบคุมเฟส (Phase control) ซึ่งจะกล่าวถึงในบทต่อไป

2.2 ไมโครคอนโทรลเลอร์ในตระกูล MCS-51

เป็นไมโครคอนโทรลเลอร์แบบชิพเดียว (Single Chip Microcontroller) คือไมโครคอมพิวเตอร์ที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชิพเดียวเหมาะสำหรับการควบคุมระบบอัตโนมัติเป็นอย่างมาก เพราะผู้ใช้งานสามารถเขียน

โปรแกรมควบคุมการทำงานได้อย่างสะดวกตามต้องการ และไม่ใครคอนโทรลเลอร์ในตระกูล MCS-51 นี้ยังมีโครงสร้างและชุดคำสั่งที่แตกต่างกันเล็กน้อย ทำให้สามารถเลือกเบอร์ที่เหมาะสมกับการใช้งานได้อย่างดีที่สุด

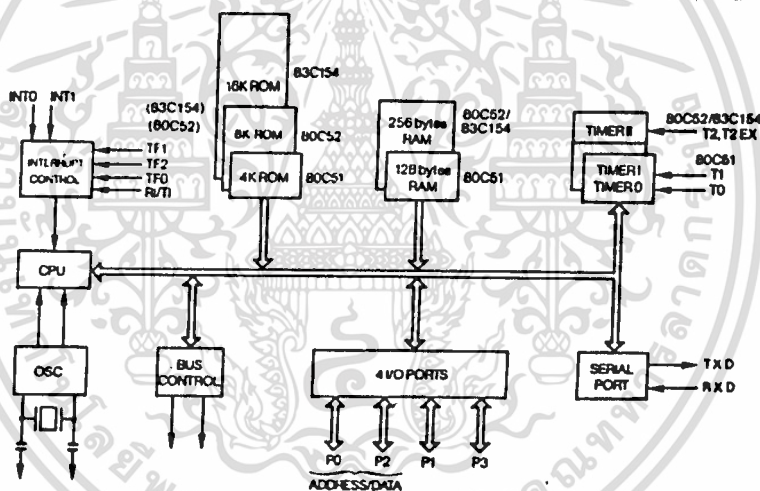
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีข้อดีดังนี้

- สามารถนำข้อมูลมา AND, OR หรือทำ Complement ได้ทั้งแบบทีละ 8 บิต และ 1 บิต
- สามารถใช้กับหน่วยความจำสำหรับโปรแกรม (Program Memory) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับเก็บชุดคำสั่งที่จะให้ MCS-51 ทำงานได้สูงสุด 64 กิโลไบต์ ทำให้เขียนโปรแกรมควบคุมการทำงานได้มาก
- สามารถต่อกับหน่วยความจำสำหรับข้อมูล (Data Memory) ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการทำงานของโปรแกรมได้สูงสุด 64 กิโลไบต์
- มีพอร์ทแบบขนาน (Parallel Port) สำหรับข้อมูลเข้าและออกจำนวน 32 บิต โดยที่ข้อมูลแต่ละบิตเป็นอิสระต่อกัน
- มีวงจร Timer/Counter ขนาด 16 บิต 2 ชุด (8052 มี 3 ชุด) ที่ทำงานในโหมดต่างๆ ได้ถึง 4 โหมด
- มี Universal Asynchronous Receiver Transmitter (UART) สำหรับรับ-ส่งข้อมูลอนุกรม (Serial) แบบ Full Duplex ที่สามารถเลือกรูปแบบการรับ-ส่งข้อมูลได้ 4 แบบ
- มีแหล่งกำเนิดสัญญาณขอขัดจังหวะการทำงานของโปรแกรม (Interrupt Request Signal) 6 แหล่ง ที่สามารถทำการกระโดดไปทำงานตอบสนองการขัดจังหวะ (Interrupt Service Routine) ได้ต่างๆ กันถึง 5 ตำแหน่ง
- สามารถเลือกการทำงานให้อยู่ในโหมดของ Idle และ Power Down ซึ่งจะประหยัดการใช้กำลังไฟในการทำงาน

ซึ่งจะเห็นว่าจากคุณสมบัติต่างๆของMCS-51 ทำให้นิยมนำมาใช้ในระบบควบคุมอัตโนมัติเป็นอย่างมาก ต่อไปนี้จะขอลำดับถึง MCS-51 โดยเรียกว่า 8051 เพื่อความสะดวกในการอธิบาย

2.2.1 โครงสร้างของ 8051

ภายใน 8051 ประกอบด้วย GATE ต่างๆ เช่น AND, OR, NOT ซึ่ง GATE เหล่านี้จะถูกนำเอามาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง (Instruction Decoder), วงจร สร้างสัญญาณนาฬิกา (Clock Signal Generator) โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยๆ ดังในรูปที่ 2.1



รูปที่ 2.1 เป็นโครงสร้างใหญ่ๆ ของ 8051

เนื่องจากลักษณะของ 8051 เป็นคอมพิวเตอร์จึงประกอบด้วย 3 ส่วนหลักๆ คือ ส่วนที่ 1 คือ หน่วยประมวลผลกลาง (Central Processing Unit) หรือตัวประมวลผล ส่วนนี้จะมีวงจรที่ทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่นๆ เรียกว่า วงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุมได้แก่สัญญาณสำหรับการติดต่อกับหน่วยความจำ, อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออกจากตัว 8051 ซึ่งส่วนควบคุมการขัดจังหวะ (Interrupt Control) และส่วนควบคุมบัส (Bus Control) ก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณควบคุมจากส่วน หน่วยประมวลผลกลาง นี้จะทำการสร้างสัญญาณโดยการถอดรหัสคำสั่ง (Instruction) ตามที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์เพื่อให้ทุกๆ ส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง

ใน หน่วยประมวลผลกลาง นี้ยังประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่าส่วนประมวลผล (Arithmetic Logic Unit) ส่วนนี้จะทำหน้าที่ประมวลผลข้อมูลเช่น การบวก, ลบ, คูณ หรือหารข้อมูล แล้วนำผลลัพธ์ไปเก็บไว้ในรีจิสเตอร์ หรือหน่วยความจำที่ต้องการ

ส่วนที่ 2 คือ หน่วยความจำ(Memory) มีไว้สำหรับจดจำข้อมูล การเอาข้อมูลไปเก็บในหน่วยความจำเรียกว่าการเขียน (Write) ข้อมูล และการเอาข้อมูลออกจากหน่วยความจำจะเรียกว่าการอ่าน (Read) ข้อมูล ซึ่งไม่ว่าจะเขียนหรืออ่านต้องรู้ตำแหน่งของหน่วยความจำหรือแอดเดรส (Address) ก่อน แต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลได้ค่าเดียวเท่านั้น ในไมโครโปรเซสเซอร์ทั่วไปรวมทั้ง 8051 ข้อมูลในแต่ละตำแหน่งของหน่วยความจำจะมีค่าได้เพียง 8 หลักของเลขฐานสอง (8 บิตเท่ากับ 1 ไบท์) ดังนั้นแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลมีค่าได้ระหว่าง 0 ถึง 255 แต่จำนวนตำแหน่งที่จะเก็บข้อมูลได้ขึ้นกับไมโครโปรเซสเซอร์แต่ละเบอร์ ส่วนที่ 3 อุปกรณ์อินพุตและเอาต์พุต (Input/Output Device) เป็นส่วนที่จะใช้ส่งข้อมูลเข้าหรือออกจาก 8051 ทำให้ 8051 ติดต่อกับภายนอกได้ จากรูปที่ 2 อุปกรณ์อินพุตและเอาต์พุตได้แก่ 4 I/O Port, Timer 0, Timer 1, Serial Port การทำงานของแต่ละส่วนมีดังนี้

1. 4 I/O Port คือที่ใช้รับ-ส่งข้อมูลซึ่งเป็นสัญญาณดิจิทัลเข้าหรือออกจากตัว 8051 พอร์ตมีทั้งหมด 4 พอร์ต โดยแต่ละพอร์ตจะรับ-ส่งข้อมูลได้ 8 บิต มีพอร์ต P0, P1, P2 และ P3 บางพอร์ตจะใช้ทำงานได้มากกว่า 1 หน้าที่ เช่น พอร์ต P0 และ P2 จะใช้สำหรับการส่งค่าตำแหน่ง (Address) ของหน่วยความจำที่ต้องการติดต่อ และพอร์ต P0 จะใช้รับ-ส่งข้อมูลเมื่อติดต่อกับหน่วยความจำได้ด้วย แต่สิ่งเหล่านี้ไม่ได้เกิดในเวลาเดียวกัน จะใช้วิธีทำงานตามลำดับ โดยควบคุมจากสัญญาณควบคุม (Control) ที่ถอดรหัสมาจากแต่ละคำสั่งที่ให้คอมพิวเตอร์ทำงานนั่นเอง และสัญญาณทั้งหมดจะอ้างอิงกับสัญญาณนาฬิกา

2. Timer 0 และ Timer 1 เป็นวงจรรับที่สามารถกำหนดให้ทำการนับจำนวน ไชเคิลของสัญญาณที่ต่อจากภายนอก 8051 หรือจำนวนไชเคิลของสัญญาณนาฬิกา ภายใน 8051 ก็ได้ ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดย หน่วยประมวลผลกลาง

3. Serial Port หรือพอร์ตอนุกรม หน่วยประมวลผลกลาง จะอ่านและเขียน ข้อมูลกับ Serial Port เป็นแบบ 8 บิต แต่ข้อมูลจะถูกส่งออกจาก 8051 เรียงไปทีละบิต ออกจากขา TXD และในการรับข้อมูลเข้าก็จะรับเข้ามาทีละบิตทางขา RXD แล้วจัดเรียง ใหม่เป็น 8 บิต เพื่อให้ หน่วยประมวลผลกลาง อ่านไปใช้งานต่อไป

2.2.2 การจัดการหน่วยความจำของ 8051

หน่วยความจำของ 8051 แบ่งออกได้เป็น 2 แบบตามลักษณะการใช้งานคือ

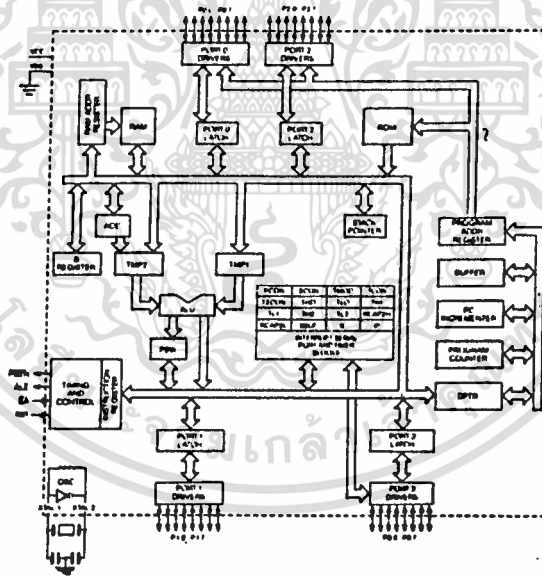
1) Program Memory เป็นหน่วยความจำที่ใช้เก็บคำสั่ง เมื่อ 8051 ทำงานก็จะอ่าน ข้อมูลที่เก็บในหน่วยความจำประเภทนี้เข้าไปถอดรหัสแล้วสร้างสัญญาณควบคุมส่วน อื่นๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำประเภทนี้ต้องเป็นแบบ Read Only Memory (ROM) และผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็น รหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ

8031 จะไม่มี ROM อยู่ภายใน ผู้ใช้จะต้องเลือกการใช้งาน Program Memory ที่ อยู่ภายนอกวงจรรวมทั้งหมด 64 กิโลไบต์

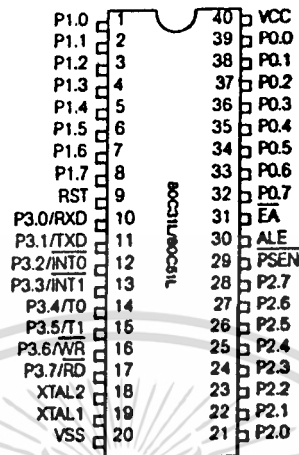
2) Data Memory เป็นหน่วยความจำที่ 8051 ใช้สำหรับพัก, เก็บข้อมูล แล้วเรียกมาใช้ใหม่ในระหว่างการทำงานของ 8051 หน่วยความจำประเภทนี้เป็นประเภท Random Access Memory (RAM) ถ้าปิดเครื่องหรือไม่จ่ายไฟให้แก่ RAM แล้วข้อมูลที่อยู่ภายในจะสูญหายไป

2.2.3 สถาปัตยกรรมของ 8051

สถาปัตยกรรมภายในของ 8051 แสดงดังรูปที่ 2.2 ซึ่งจะอธิบายส่วนย่อยๆ ของภายใน 8051 เพียงชีพเดียว และสัญญาณจากภายในจะต่อสู่ภายนอกทางขา (Pin) ของ 8051 ที่มีอยู่ 40 ขา ดังรูปที่ 2.3



รูปที่ 2.2 สถาปัตยกรรมภายในของ 8051



รูปที่ 2.3 ไดอะแกรมขาของ 8051 แบบ DIP

8051 ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 8051 มีขาอยู่ข้างละ 20 ขา รวมทั้งหมด 40 ขา จะใช้งานต่างๆ กันดังนี้คือ

Vcc - ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลต์เข้าไปเพื่อให้วงจรรวมทำงานได้ระดับโวลต์เตจของลอจิก 0 และ 1 ของ 8051 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง

Vss - ขา 20 เป็นขาที่ต้องต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน

Port 0 - เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับแต่ละขาจะเขียนว่า P0.0, P0.1,..., P0.7 โดยที่ P0.7 หมายถึงบิต 7 ของพอร์ท 0 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด (Most Significant) และ P0.0 คือ บิต 0 ของพอร์ท 0 เป็นบิตที่มีนัยสำคัญต่ำสุด (Least significant) พอร์ท 0 นี้ใช้ได้ทั้งการรับส่งตำแหน่งและข้อมูลกับหน่วยความจำ หรือใช้เป็นพอร์ทรับ-ส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกจากพอร์ท 0 จะถูกแลตช์ไว้ที่ขาของพอร์ท



Port 1 - เป็นพอร์ทขนานขนาด 8 บิต คือขา P1.0 ถึง P1.7 (ขา 1-8) P1.0 หมายถึง บิต 0 ของพอร์ท 1 ซึ่งเป็นบิตที่มีนัยสำคัญต่ำสุด และบิต P1.7 หมายถึงบิตที่ 7 ของพอร์ท 1 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด พอร์ท 1 นี้ จะใช้ทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกมาทางพอร์ท 1 จะถูกแลทช์ไว้แล้วส่งออกไปทางแต่ละขาของพอร์ท

Port 2 - เป็นพอร์ทขนานขนาด 8 บิต คือขา P2.0 ถึง P2.7 (บิต 0 ถึง บิต 7 ของพอร์ท 2) พอร์ท 2 จะใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ซึ่งเป็นค่า 8 บิตบนของตำแหน่ง และใช้เป็นพอร์ทรับและส่งข้อมูลกับภายนอก

Port 3 - คือขา P3.0 ถึง P3.7 หรือขา 10-17 ตามลำดับ ซึ่งพอร์ท 3 นั้นนอกจากจะรับ-ส่งข้อมูลแล้วยังสามารถใช้ในการทำงานอื่นได้อีก ซึ่งแต่ละบิตของพอร์ท 3 จะมีฟังก์ชันอื่นดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/INT0 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/INT1 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/T0 (Timer/Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร

Timer/Counter 0 ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ T0 หรือสัญญาณนาฬิกาก็ได้

P3.5/T1 (Timer/Counter 1 External Input) ขารับสัญญาณเข้าไปยัง

Timer/Counter 1 ซึ่งมีการทำงานเหมือนกับ T0

P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลจากภายนอก 8051

P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

RST - ขารีสเตทนี้จะใช้ทำการรีเซ็ตการทำงานของ 8051 ให้มีการเริ่มต้นทำงานใหม่

ALE - เป็นขาที่ใช้ส่งสัญญาณออกไปภายนอกเพื่อควบคุมการแลทช์ค่าแอดเดรสไบต์ต่ำจากพอร์ท 0 ในระหว่างติดต่อหน่วยความจำสำหรับโปรแกรมหรือข้อมูลภายนอก ปกติถ้าไม่มีการติดต่อกับหน่วยความจำ ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์ ดังนั้นเราจึงสามารถใช้ความถี่ที่ได้จากขานี้ไปใช้งานอย่างอื่นได้ แต่ความถี่ที่ขานี้จะลดลงครึ่งหนึ่ง ในระหว่างการติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลที่อยู่ภายนอกชิพ

PSEN - เป็นขาที่ 29 ใช้ส่งสัญญาณสโตรบเมื่อต้องการอ่านคำสั่งที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 ในกรณีที่อ่านคำสั่งซึ่งเก็บในหน่วยความจำสำหรับโปรแกรมภายใน จะไม่มีการส่งสัญญาณออกมาจากขานี้

EA - เป็นขาสำหรับเลือกให้ 8051 ทำงานจาก โปรแกรมที่อยู่ภายในหรือภายนอก ชิพ ถ้าที่ขานี้มีลอจิกเป็น 0 หมายถึงใช้โปรแกรมจากหน่วยความจำที่เก็บโปรแกรมภายนอก ถ้าลอจิกเป็น 1 ให้ 8051 ใช้โปรแกรมจากหน่วยความจำที่เก็บโปรแกรมภายในชิพ สำหรับเบอร์ 8031 ไม่มีหน่วยความจำภายใน จึงให้ต่อ ขานี้ลงกราวด์เสมอ

XTAL1 - ใช้ต่อคริสตัลภายนอก โดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์

XTAL2 - ใช้ต่อคริสตัลภายนอก โดยเป็นเอาต์พุตออกจากวงจรออสซิลเลเตอร์

2.4 รีจิสเตอร์ฟังก์ชันพิเศษ

รีจิสเตอร์ในกลุ่มรีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register; SFR) เป็นรีจิสเตอร์สำหรับใช้งานเฉพาะ คือข้อมูลที่ถูกนำไปเก็บในรีจิสเตอร์เหล่านี้จะมีความหมายเฉพาะตัวของรีจิสเตอร์ แต่การอ้างอิงจะถือเสมือนว่าเป็นหน่วยความจำตำแหน่งหนึ่ง รีจิสเตอร์เหล่านี้ มีดังนี้

ACC (Accumulator)

เป็นรีจิสเตอร์ขนาด 8 บิตทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยงานภายในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การทำงานของรีจิสเตอร์นี้ มีลักษณะเช่นเดียวกับตัวแอดคิวมูเลเตอร์ของโปรเซสเซอร์ทั่วไป

B Register

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้สำหรับคำสั่งการคูณและหารตัวเลข ในกรณีที่ไม่ได้ใช้ทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้เช่นเดียวกับรีจิสเตอร์ทั่วไปได้

PSW (Program Status Word)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ทำหน้าที่บอกถึงแฟล็กสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงค์ (Bank) ของรีจิสเตอร์ที่ใช้งานด้วย

SP (Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งแอดเดรสหน่วยความจำของบริเวณ ที่ใช้งานเป็นสแต็ก (stack) สำหรับเก็บข้อมูลแอดคิวมูเลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม

DPTR (Data Pointer Register)

เป็นรีจิสเตอร์ขนาด 16 บิต และสามารถใช้งานแยกเป็นรีจิสเตอร์ขนาด 8 บิต จำนวน 2 ตัว คือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าแอดเดรสของหน่วยความจำหรืออุปกรณ์ที่จะต้องใช้งานภายในโปรแกรม การใช้รีจิสเตอร์ DPTR นั้นนับว่ามีประโยชน์ เนื่องจากทำให้หน่วยประมวลผลกลางสามารถใช้เทคนิคของการแอดเดรสแบบทางอ้อมได้

SBUF (Serial Data Buffer)

รีจิสเตอร์นี้มีขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์นี้มีอยู่ด้วยกันสองชุด และแยกจากกันอย่างชัดเจน สำหรับการส่งและการรับ โดย หน่วยประมวลผลกลาง จะทำการจัดการเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

SCON (Serial Port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต ใช้สำหรับควบคุมการส่งและรับข้อมูลผ่านทางพอร์ตอนุกรม โดยสามารถเลือกได้ถึง 4 โหมด ซึ่งแต่ละบิตจะมีความหมายในการควบคุมการทำงานต่างๆ กัน

TMOD (Timer/Counter Mode Register)

เป็นรีจิสเตอร์ขนาด 8 บิตที่มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer 1 โดยการเซตค่าที่แต่ละบิตของรีจิสเตอร์นี้ โดยสามารถเลือกให้ทำงานเป็น Timer หรือ Counter และยังสามารถเลือกโหมดการทำงานได้ถึง 3 โหมด

TCON (Timer Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้ควบคุมการทำงานและบอกสถานะของ Timer 0 และ Timer 1 ซึ่งทำโดยการเซตค่าที่บิตต่างๆ ของรีจิสเตอร์

IE (Interrupt Enable Register)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้ควบคุมการตอบสนองการอินเทอร์รัปต์ โดยสามารถกำหนดการให้ตอบสนองหรือไม่ตอบสนองการอินเทอร์รัปต์ต่อสัญญาณอินเทอร์รัปต์ของพอร์ตสื่อสารอนุกรม, ไทม์เมอร์ 1 (เมื่อเกิด overflow), ไทม์เมอร์ 0 (เมื่อเกิด overflow) และการอินเทอร์รัปต์จากภายนอก

IP (Interrupt Priority Register)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่แต่ละบิตนั้นใช้สำหรับกำหนดความสำคัญของการอินเทอร์รัปต์

PCON (Power Control Register)

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมหน้าที่การทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรเซสเซอร์ การกำหนดอัตราการทำงานของอัตราเร็วในการสื่อสารข้อมูลอนุกรม และแฟล็กสถานะสำหรับการใช้งานทั่วไป

บทที่ 3

การสร้างและการทำงานของโครงงาน

3.1 หลักการทำงานของเครื่องควบคุม

วงจรนี้จะทำการควบคุมโวลต์เตจที่จ่ายให้แก่โหลดโดยการเปลี่ยนแปลงมุมที่ ทริก เอสซีอาร์ (SCR) เมื่อมุมที่ทริก เอสซีอาร์ (SCR) เปลี่ยนแปลง พลังงานที่จ่ายให้กับ โหลดก็เปลี่ยนแปลงตามด้วย มุมทริกที่ เอสซีอาร์ (SCR) นี้จะควบคุมโดยไมโครคอนโทร ลเลอร์ ซึ่งจะส่งข้อมูลขนาด 8 บิต ผ่าน วงจรแปลงสัญญาณดิจิตอลเป็น สัญญาณ อนาล็อก เป็นแรงดันไฟฟ้า 0-10 โวลต์ แล้วแปลงเป็นความถี่ด้วยวงจร แปลงแรงดัน ไฟฟ้าเป็นความถี่ ความถี่นี้จะถูกส่งผ่านสายนำสัญญาณไปที่ ส่วนควบคุม ซึ่งมี วงจร แปลงความถี่เป็นแรงดันไฟฟ้า แปลงความถี่กลับเป็นโวลต์เตจนำไปป้อนให้แก่ IC Phase Control TCA 785 IC ตัวนี้จะเป็นตัวกำเนิดสัญญาณที่จะไปทริก เอสซีอาร์ (SCR) การเปลี่ยนแปลงทั้งหมดจึงขึ้นกับข้อมูลที่ส่งจาก ไมโครคอนโทรลเลอร์ เพียงอย่างเดียว

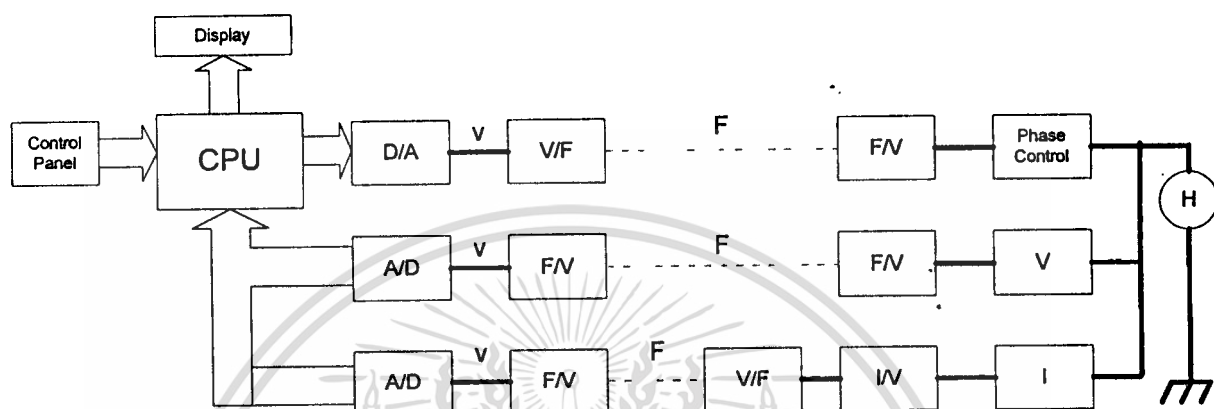
เราจึงสามารถควบคุมพลังงานทั้งหมดโดยโปรแกรมที่เขียนขึ้นสั่งงาน ไมโคร คอนโทรลเลอร์ ผลของการควบคุมจะแสดงออกมาที่จอ LCD ในรูปของ โวลต์เตจ และ กระแส ซึ่งส่งมาจาก ส่วนควบคุม ในรูปความถี่เช่นกัน ที่ต้องส่งในรูปความถี่นี้ เนื่อง จากไม่มีสัญญาณรบกวนและความถี่จะไม่ขึ้นกับระยะทางระหว่าง ตัวคอนโทรล กับ ส่วนควบคุม ทำให้มีความถูกต้องในการรับส่งสัญญาณสูง

คุณสมบัติของโครงการนี้คือ

1. ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์
2. แสดงค่า โวลต์เตจ กระแส Setpoint Menu ทั้งหมดผ่านทางจอ LCD
3. สามารถปรับโวลต์เตจ ที่ป้อนให้แก่โหลดอัตโนมัติตามโปรแกรมที่ตั้งไว้ หรือปรับโวลต์เตจตามที่ต้องการทันที
4. บันทึกค่า โวลต์เตจ และ กระแส ณ เวลาที่โปรแกรมไว้เก็บไว้ในหน่วย ความจำแล้วสามารถแสดงออกมาเมื่อต้องการ
5. ใช้ระบบ Real Time Clock เป็นฐานเวลา

วงจรทั้งหมดสามารถแบ่งออกเป็นส่วนๆดังนี้

- ANT-32 EMBEDDED CONTROL BOARD
- การควบคุมพลังงานไฟฟ้าด้วย TCA785
- การเปลี่ยนความถี่เป็นแรงดันไฟฟ้า
- การเปลี่ยนแรงดันไฟฟ้าเป็นความถี่
- การเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล
- การเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก
- การวัดกระแสและแรงดัน
- การเชื่อมต่อด้วยแสง (Opto isolator)
- การเชื่อมต่อด้วย Pluse Transformer
- การเชื่อมต่อกับระบบไมโครคอนโทรลเลอร์
- การแสดงผลทางจอ LCD
- โปรแกรมควบคุม



รูปที่ 3.1 แสดงส่วนประกอบทั้งหมดของวงจร

3.2 ANT-32 EMBEDDED CONTROL BOARD

ANT-32 เป็นบอร์ดไมโครคอนโทรลเลอร์นำไปใช้งานในลักษณะ EMBEDDED CONTROLLER กล่าวคือ เป็นบอร์ดที่ถูกออกแบบมาเพื่องานควบคุมโดยเฉพาะ โดยถูกติดตั้งอยู่ในเครื่องมือเครื่องจักรกล เครื่องใช้ไฟฟ้า รวมทั้งระบบอัตโนมัติต่างๆ บอร์ดนี้สามารถใช้กับ หน่วยประมวลผลกลาง ตระกูล MCS-51 ได้ทั้งหมด นอกจากนี้ ANT-32 ยังประกอบไปด้วยวงจรในส่วน Watchdog Timer, Battery Backup และ Power Fail Detector ใช้ชิพ MAX691 วงจร Real Time Clock ใช้ชิพ DS1202

คุณสมบัติของบอร์ด ANT-32

- เป็นบอร์ดคอนโทรลใช้กับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ใช้กับหน่วยประมวลผลกลาง เบอร์ 8031 ทำงานที่ความถี่สัญญาณนาฬิกา 11.0592 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้งานหน่วยความจำบอร์ดได้ 3 ตำแหน่งด้วยกันคือ

U2 เป็นหน่วยความจำโปรแกรม (PROGRAM MEMORY) ใช้กับ EPROM ขนาด 8-32KByte เบอร์ 2764,27128 หรือ 27256

U3 เป็นหน่วยความจำข้อมูล (DATA MEMORY) ใช้กับ RAM ขนาด 8KByte เบอร์ 6264 หรือ 32KByte เบอร์ 62256 สามารถแบคอัพข้อมูลได้โดยใช้แบตเตอรี่ลิเทียม

U4 เป็นหน่วยความจำโปรแกรมและข้อมูล (PROGRAM AND DATA MEMORY) ใช้กับ EPROM, RAM หรือ EEPROM ขนาด 8-32 KByte โดยใช้ EPROM เบอร์ 2764,27256 ใช้ RAM เบอร์ 6264,62256 หรือ EEPROM เบอร์ 2867(A) ,28256(A)

- มีพอร์ต I/O เบอร์ 8255 จำนวน 2 ตัว(48 บิต) สำหรับต่อไปใช้งานภายนอก
- มีวงจร SERIAL INTERFACE DRIVER RS232 ด้วยชิพเบอร์ MAX232 สำหรับการต่อเข้าเครื่องไมโครคอมพิวเตอร์
- มีวงจร Watchdog Timer, Powerup/down Reset ด้วยชิพเบอร์ MAX691
- มีวงจร RTC (real time clock) ใช้ชิพเบอร์ DS1202
- มีคอนเน็คเตอร์สำหรับ PORT 1 ของไมโครคอนโทรลเลอร์โดยเฉพาะ
- มีคอนเน็คเตอร์สำหรับ SYSTEM BUS ทำให้ขยายระบบได้ง่าย และสามารถใช้บอร์ดยาต่าง ๆ ที่มีขึ้นในอนาคต
- สามารถเลือกเบอร์และชนิดหน่วยความจำ หรือกำหนดคุณสมบัติต่างๆ ของบอร์ดได้ด้วยจัมป์เปอร์

การปรับจัมป์เปอร์เลือกหน่วยความจำ

ไมโครคอนโทรลเลอร์ตระกูล 8031(32) สามารถต่อกับหน่วยความจำภายนอกได้ถึง 128 KByte โดยแบ่งออกเป็น 2 ส่วนคือ 64 KByte เป็นหน่วยความจำโปรแกรม (PROGRAM MEMORY) และอีก 64 KByte เป็นหน่วยความจำข้อมูล (DATA MEMORY) ซึ่งหน่วยความจำทั้งสองส่วนนี้มีตำแหน่งแอดเดรสที่ 0000H-FFFFH เหมือนกัน แต่จะถูกแยกออกจากกันด้วยสัญญาณควบคุมที่ต่างกัน โดยสัญญาณ PSEN ใช้ควบคุมในการอ่านหน่วยความจำโปรแกรม (EPROM) สัญญาณ RD และ WR ใช้ควบคุมการอ่านหน่วยความจำโปรแกรมและข้อมูล (PROGRAM AND DATA MEMORY) ใช้สัญญาณ

GET ซึ่งสัญญาณนี้ได้จากการ AND สัญญาณ PSEN และ RD หน่วยความจำส่วนนี้สามารถใช้กับ EPROM หรือ RAM

สำหรับบอร์ด ANT-32 ได้จัดหน่วยความจำออกเป็น 3 ส่วนด้วยกันคือ

U2 เป็นหน่วยความจำโปรแกรม (PROGRAM MEMORY) แอดเดรส 0000H-7FFFH

U3 เป็นหน่วยความจำข้อมูล (DATA MEMORY) แอดเดรส 0000H-7FFFH

U4 เป็นหน่วยความจำโปรแกรมและข้อมูล (PROGRAM AND DATA MEMORY) แอดเดรส 8000H-F7FFFH

ส่วนแอดเดรส F800H-FFFFH ใช้เป็นตำแหน่งของพอร์ทอินพุต/เอาต์พุต หน่วยความจำ U2 , U3 และ U4 สามารถเลือกขนาด (SIZE) และชนิด (TYPE) ได้ด้วยจัมป์เปอร์

อินพุตเอาต์พุตพอร์ท (8255)

8255 Programmable Peripheral Interface (PPI) เป็นพอร์ทแบบขนานที่เป็นที่นิยมใช้งานกันมากมาย สำหรับบอร์ด ANT-32 ใช้พอร์ท 8255 จำนวน 2 ตัวทำหน้าที่เป็นพอร์ททำให้มีพอร์ทอินพุตเอาต์พุตถึง $24 \times 2 = 48$ บิต โดยแบ่งเป็น USER PORT 1 และ 2 มีตำแหน่งแอดเดรสดังนี้

USER PORT 1 (U10) แอดเดรส $F800H + 8255 \text{ offset addr} = \text{actual addr}$

port A ตำแหน่งแอดเดรส $F800H + 00H = F800H$

port B ตำแหน่งแอดเดรส $F800H + 01H = F801H$

port C ตำแหน่งแอดเดรส $F800H + 02H = F802H$

Mode port ตำแหน่งแอดเดรส $F800H + 03H = F803H$

USER PORT 2 (U11) แอดเดรส $FC00H + 8255 \text{ offset addr} = \text{actual addr}$

port A ตำแหน่งแอดเดรส $FC00H + 00H = FC00H$

port B ตำแหน่งแอดเดรส $FC00H + 01H = FC01H$

port C ตำแหน่งแอดเดรส $FC00H + 02H = FC02H$

Mode port ตำแหน่งแอดเดรส $FC00H + 03H = FC03H$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะใช้งานพอร์ท 8255 ผู้ใช้ต้องทำการกำหนดโหมดการทำงาน (configuration) ของพอร์ท A , B , C ให้เป็นพอร์ทอินพุตหรือเอาต์พุต โดยทำการเขียนค่า control code ไปที่ Mode Port ซึ่ง Mode Port นี้สามารถเขียนได้เท่านั้นไม่สามารถอ่านได้ในที่นี้จะกล่าวเฉพาะการทำงานในโหมด 0 ซึ่งเป็นโหมดที่ใช้งานได้สะดวกและง่ายต่อการทำความเข้าใจ ดังแสดงค่า control code ดังตารางที่ 1

ตารางที่ 3.1 ค่า CONTROL CODE ของ 8255

Port A (PA0-PA7)	Port C บน (PC4-PC7)	Port B (PB0-PB7)	Port C ล่าง (PC0-PC3)	Control code (HEX)
OUTPUT	OUTPUT	OUTPUT	OUTPUT	80H
OUTPUT	OUTPUT	OUTPUT	INPUT	81H
OUTPUT	OUTPUT	INPUT	OUTPUT	82H
OUTPUT	OUTPUT	INPUT	INPUT	83H
OUTPUT	INPUT	OUTPUT	OUTPUT	88H
OUTPUT	INPUT	OUTPUT	INPUT	89H
OUTPUT	INPUT	INPUT	OUTPUT	8AH
OUTPUT	INPUT	INPUT	INPUT	8BH
INPUT	OUTPUT	OUTPUT	OUTPUT	90H
INPUT	OUTPUT	OUTPUT	INPUT	91H
INPUT	OUTPUT	INPUT	OUTPUT	92H
INPUT	OUTPUT	INPUT	INPUT	93H
INPUT	INPUT	OUTPUT	OUTPUT	98H
INPUT	INPUT	OUTPUT	INPUT	99H
INPUT	INPUT	INPUT	OUTPUT	9AH
INPUT	INPUT	INPUT	INPUT	9BH

REAL TIME CLOCK & CALENDAR

สำหรับการใช้งานระบบไมโครคอนโทรลเลอร์ที่มีเวลาเข้ามาเกี่ยวข้องด้วยจำเป็นต้องมีวงจรในส่วนที่ทำหน้าที่เป็น RTC (Real Time Clock) คือนาฬิกาเวลาจริง ซึ่งบอร์ด ANT-32 ใช้ชิพ RTC เบอร์ DS1202 Serial Timekeeper Chip ของ DALLAS SEMICONDUCTOR โดยต่อร่วมกับอุปกรณ์ภายนอกเพียงเล็กน้อยและที่สำคัญคือ ไม่ต้องการการปรับแต่ง ซึ่งเมื่อจะใช้ RTC นี้ต้องมีชิพ DS1202 และ MAX691 รวมทั้งคริสตัล 32.768 KHz และแบตเตอรี่ลิเธียม ANT-32 ด้วย

DS1202 (U7) ประกอบไปด้วย Real Time Clock / Calendar และ Static RAM ขนาด 24 ไบท์ ทำการอินเตอร์เฟสกับ หน่วยประมวลผลกลาง ในแบบอนุกรม โดยใช้สายเพียง 3 เส้นคือ (1) ขา RST(reset) , (2) ขา I(Data line) และ (3)ขา SCLK (Serial clock) ขาสัญญาณทั้งสามนี้จะต่อเข้ากับขา P1.6,P1.4 และ P1.5 ของ หน่วยประมวลผลกลาง ตามลำดับ เมื่อต้องการทราบค่าเวลา หน่วยประมวลผลกลาง ต้องทำการอ่านเวลาจาก RTC เพราะว่า DS1202 ไม่มีขาสำหรับไปอินเตอร์รัพท์ หน่วยประมวลผลกลาง หน่วยประมวลผลกลาง สามารถเขียนหรืออ่านข้อมูลของ CLOCK หรือ RAM ได้ 2 วิธีคือ Single-byte และ Multiple-byte โดยทั้งสองวิธี หน่วยประมวลผลกลาง ต้องส่ง command byte (8บิต) ให้ DS1202 เพื่อบอกให้ DS1202 ทราบว่าจะทำการเขียนหรืออ่าน CLOCK หรือ RAM พร้อมตำแหน่ง (address) และตามด้วยข้อมูล ในขณะที่กำลังติดต่อกับ DS1202 สัญญาณที่ขา RSTต้องเป็นลอจิก "1" ขา SCLK จะเป็นสัญญาณ Serial Clock เพื่อทำการเขียนหรืออ่านข้อมูล โดยใช้สัญญาณ Clock 1 ลูก สำหรับข้อมูล 1 บิต ส่วนขา I/O เป็นข้อมูลอนุกรม โดยจะเป็นอินพุตเมื่อทำการเขียน และเป็นเอาต์พุตเมื่อทำการอ่าน โดยข้อมูลที่จะเขียนหรืออ่านนี้จะเริ่มจากบิต 0 และจบด้วยบิต 7 ค่าของ command byte ในการเขียนและอ่าน CLOCK และ RAM แสดงไว้ดังตารางที่ 2

ตารางที่ 3.2 แสดง COMMAND BYTE ในการเขียนและอ่านค่า CLOCK และRAM-

REGISTER	FUNCTION	COMMAND ADDRESS (HEX)	WRITE=W READ=R	RANGE DATA (BCD)	REGISTER DEFINITION								
					7	6	5	4	3	2	1	0	
0	SECONDS	80	W	00-59	CH	10 SEC				SEC			
		81	R										
1	MINUTES	82	W	00-59	0	10 MIN				MIN			
		83	R										
2	12HRS	84	W	01-12	12	0	AP	HR	HOUR				
	24 HRS	85	R	00-23	24	0	10	HR					
3	DATE	86	W	01-31	0	0	10DATE		DATE				
		87	R										
4	MONTH	88	W	01-12	0	0	0	10M	MONTH				
		89	R										
5	DAY	8A	W	01-07	0	0	0	0	DAY				
		8B	R										
6	YEAR	8C	W	00-99	10 YEAR				YEAR				
		8D	R										
7	WRITE	8E	W	00-80	WP	ALWAYS ZERO							
	PROTECT	8F	R										

31	CLOCK BURST	BE	W
		BF	R
0	RAM 0	C0	W
		C1	R
•	•	•	•
		•	•
23	RAM 23	EE	W
31	RAM BURST	FE	W
		FF	R

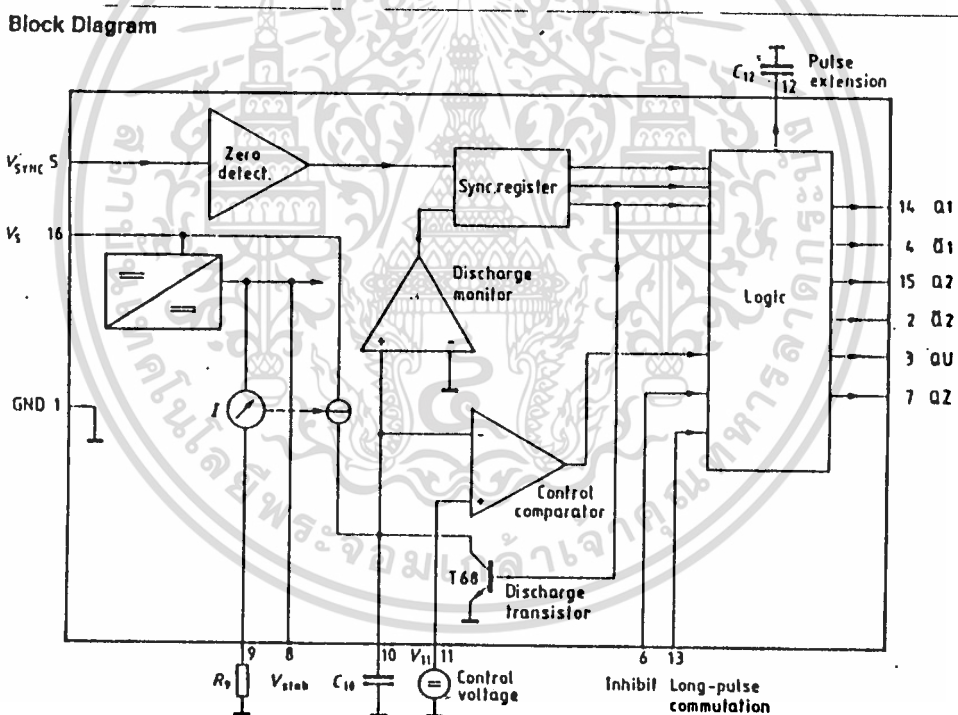
3.3 การควบคุมพลังงานไฟฟ้าด้วย TCA785

ในวงจรส่วนควบคุมพลังงานนี้ เราใช้ IC TCA 785 เป็นตัวกำเนิดสัญญาณทริก เอสซีอาร์ (SCR) เพื่อควบคุม Phase ในการทริก เอสซีอาร์ การทำงานของ TCA 785 เริ่มต้นด้วยการนำสัญญาณซิงค์จากแหล่งจ่ายสัญญาณความถี่เดียวกันมาผ่านการกำจัด กระแสให้พอเหมาะพอดีก่อนเข้าขา 5 จากนั้นจะผ่านวงจรตรวจจับแรงดันศูนย์ (zero

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

voltage detector) ไปยังส่วนของวงจรเชิงโคโรไนซ์รีจิสเตอร์ (SYNC.register) ซึ่งควบคุมสัญญาณแรมป์ที่เกิดจากการเก็บประจุของ C_{10} ด้วยกระแสที่คงที่ ซึ่งกำหนดความชันของสัญญาณแรมป์ด้วย R_9 ถ้าสัญญาณแรมป์ที่ 10 มีระดับ แรงดันต่างไปจากแรงดันไฟตรงควบคุมที่ขา 11

ตัวควบคุมการเปรียบเทียบ (control comparator) จะผลิตสัญญาณเชิงตรรกะไปยังวงจรลอจิกต่อไป ซึ่งสัญญาณลอจิกนี้สามารถควบคุมให้เลื่อนไปมาจาก 0-180 องศาได้ในทุก ๆ ครั้งคลื่นจะปรากฏพัลส์ต่าง ๆ ที่ เอาท์พุท ดังรูป 3.3



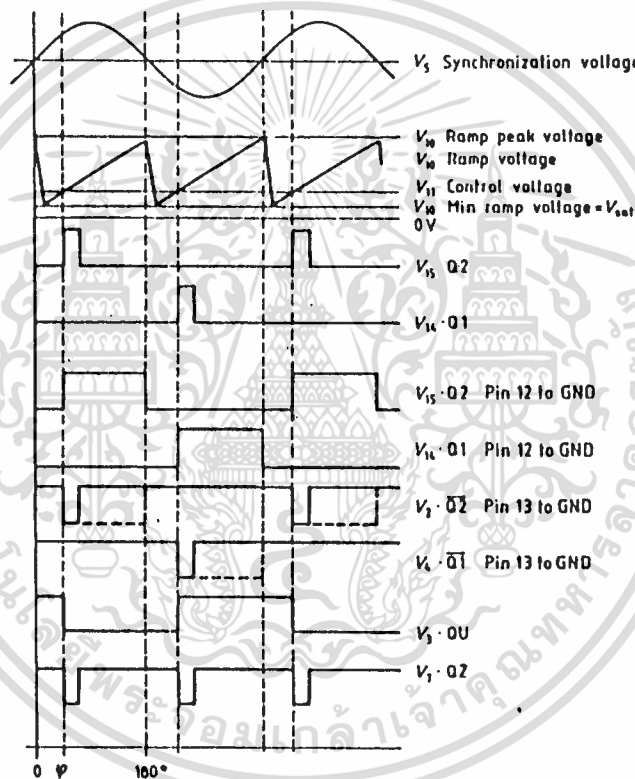
รูปที่ 3.2 แสดงลักษณะภายในของ TCA785

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

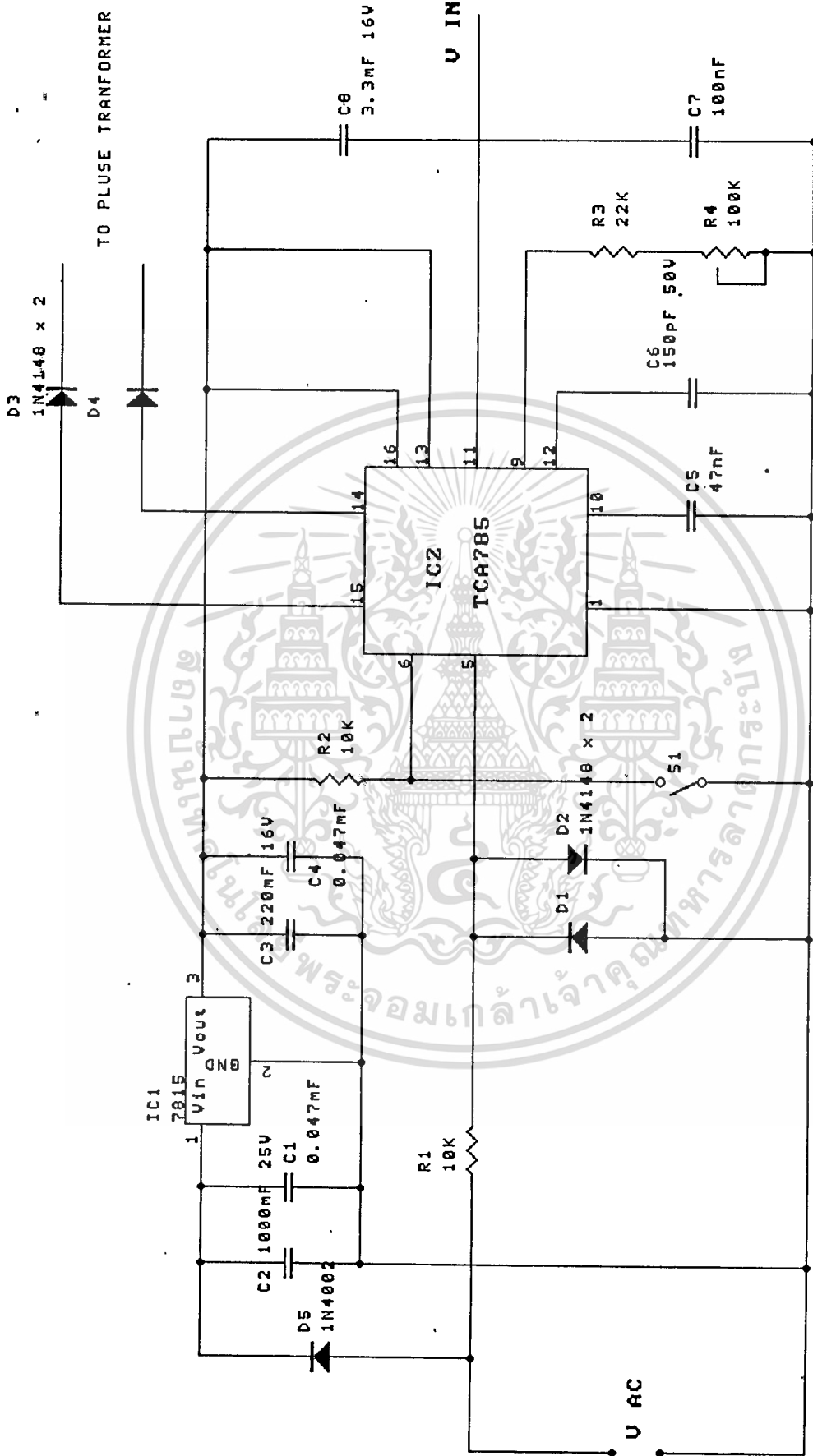
พัลส์ต่าง ๆ เหล่านี้ปกติจะมีความกว้างพัลส์ประมาณ 30 ไมโครวินาที สามารถขยายความกว้างพัลส์ต่าง ๆ ออกไปอีกถึง 180 องศา โดยการเปลี่ยนค่า C_{12} หรือจะลัด C_{12} ลงกราวด์ก็จะได้ความกว้างพัลส์ ตั้งแต่มุมทริกไปถึง 180 องศา

สำหรับขา 6 จะเป็นตัวปิดเปิดพัลส์ทั้งหมด หากขา 6 มีลอจิกเป็น "1" ก็จะมีพัลส์ออกตามปกติถ้าขา 6 มีลอจิกเป็น "0" ก็จะไม่มีการพัลส์ใด ๆ ออกมา

Pulse Diagram



รูปที่ 3.3 แสดงลักษณะสัญญาณที่ขาต่างๆของ TCA 785



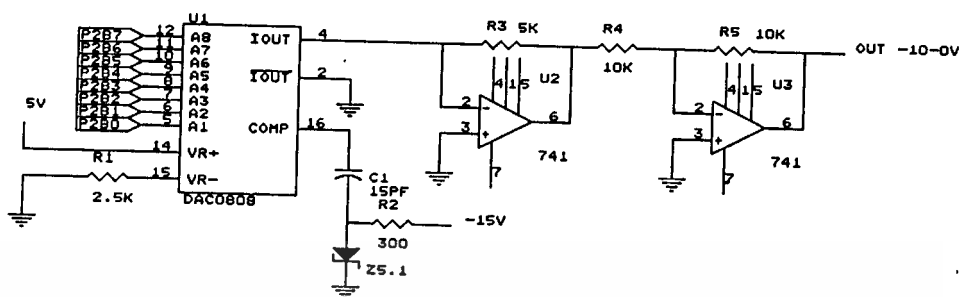
รูปที่ 3.4 แสดงวงจรควบคุมพัลส์งานด้วย TCA 785

วงจรนี้เป็นส่วนควบคุมมุมทริกที่จะทริก เอสซีอาร์ (SCR) ในส่วน กระแสสลับ ให้ทำงานหรือ ON โดยจะให้ เอาท์พุทเป็นพัลส์ มีทั้งซีกบวกและซีกลบของไฟ กระแสสลับ สำหรับแรงดันไฟฟ้าที่นำมาควบคุมควบคุมมุมทริก มาจากวงจรเปลี่ยนสัญญาณดิจิทัลเป็นอนาล็อก กับ Ramp peak voltage V_{10} ของ TCA 785 โดยค่า control voltage $= V_{10} - V_{D/A}$ สำหรับ V_{10} ปรับค่าได้โดยการปรับค่าความต้านทานที่ ขา 9 ถ้าค่า $V_{D/A}$ น้อย ค่า control voltage จะมากทำให้ได้ พลังงานจ่ายโหลดน้อยเนื่องจากมุมทริกมาก ในทางกลับกันถ้า $V_{D/A}$ มากจะได้พลังงานมากด้วย

3.4 การเปลี่ยนสัญญาณดิจิทัลเป็นอนาล็อก

วงจร แปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (Digital to Analog Converter) เป็นวงจรรับค่าที่ส่งออกมาจาก หน่วยประมวลผลกลาง ซึ่งเป็นค่า ไบนารี 8 บิต แล้วเปลี่ยนเป็นค่าโวลต์เดจที่ส่งต่อไปยังส่วน phase controlโดยผ่านวงจร Summing ซึ่งทำให้เมื่อส่งค่าจาก หน่วยประมวลผลกลาง ออกมาน้อยพลังงาน ที่โหลด ได้รับก็น้อยด้วย เมื่อส่งค่ามาก พลังงานก็มากตามด้วย

สำหรับวงจรที่ใช้ IC เบอร์ DAC0808 ซึ่งจะได้ค่า เอาท์พุท เป็นกระแสจะมากหรือน้อย ขึ้นกับ อินพุทโดยค่าที่มากที่สุดจะเกิดเมื่อส่ง อินพุท เป็น FFH ค่าเอาท์พุทสูงสุดเท่ากับกระแสอ้างอิงที่ขา 14 ซึ่ง ตามที่วงจรมีค่า 2 mA เนื่องจากเอาท์พุท ที่ขา4 จากIC เป็นกระแส ดังนั้นจึงต่อ ออปแอมป์เพื่อเปลี่ยนกระแสเป็นแรงดันให้ได้ค่ามากที่สุด 10 V ค่าแรงดัน เอาท์พุท

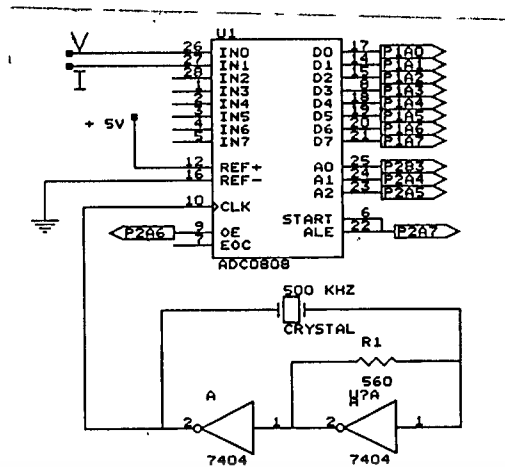


รูปที่ 3.5 เปลี่ยนสัญญาณดิจิทัลเป็นอนาล็อก

3.5 การเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

ส่วนวงจร วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (Analog to Digital Converter) เป็นวงจรเปลี่ยนค่าสัญญาณที่เป็นอนาล็อกให้เป็นสัญญาณทางดิจิทัล ค่าที่ได้จาก แปลงความถี่เป็นแรงดันไฟฟ้า (Frequency to Voltage converter) จะมีโวลต์ เตจเป็นลบ ดังนั้นจึงต้องผ่านวงจรอินเวอร์ตติ่งก่อนเข้า วงจรแปลงสัญญาณสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล โดยใช้ช้อปแอมป์ สำหรับวงจรใช้ IC เบอร์ ADC0808 ซึ่งจะให้ค่า binary 8 bit ตามอินพุตซึ่งช่วง Span เท่ากับสองเท่าของ $V_{ref}/2$ ขา 9 ในวงจรนี้ให้ $V_{ref}/2 = 1.5$ ดังนั้นปรับค่าได้ในช่วง 3 V โดยการปรับ ค่า zero ได้ที่ขา 7

การเปลี่ยนค่าต้องให้สัญญาณ ALE และ START ก่อน ถึงจะรับค่าอินพุตขา 26 และขา 27 เข้ามาแล้วเปลี่ยนค่า หลังจากนั้นจึงให้สัญญาณ OE แล้ว IC จะส่งค่าไบนารีออกทางเอาต์พุตทั้ง 8 ขา



รูปที่ 3.6 เปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล

3.6 การเปลี่ยนแรงดันไฟฟ้าเป็นความถี่

วงจร แปลงแรงดันไฟฟ้าเป็นความถี่ (Voltage to Frequency Converter) ใช้ IC เบอร์ LM331 เป็นตัวเปลี่ยนค่า DC voltage เป็นความถี่แปรตาม อินพุต โดยได้ค่าความถี่เท่ากับ

$$f_{out} = -V_{in} * R_s / (2.09 V * R_{in} * R_t * C_t)$$

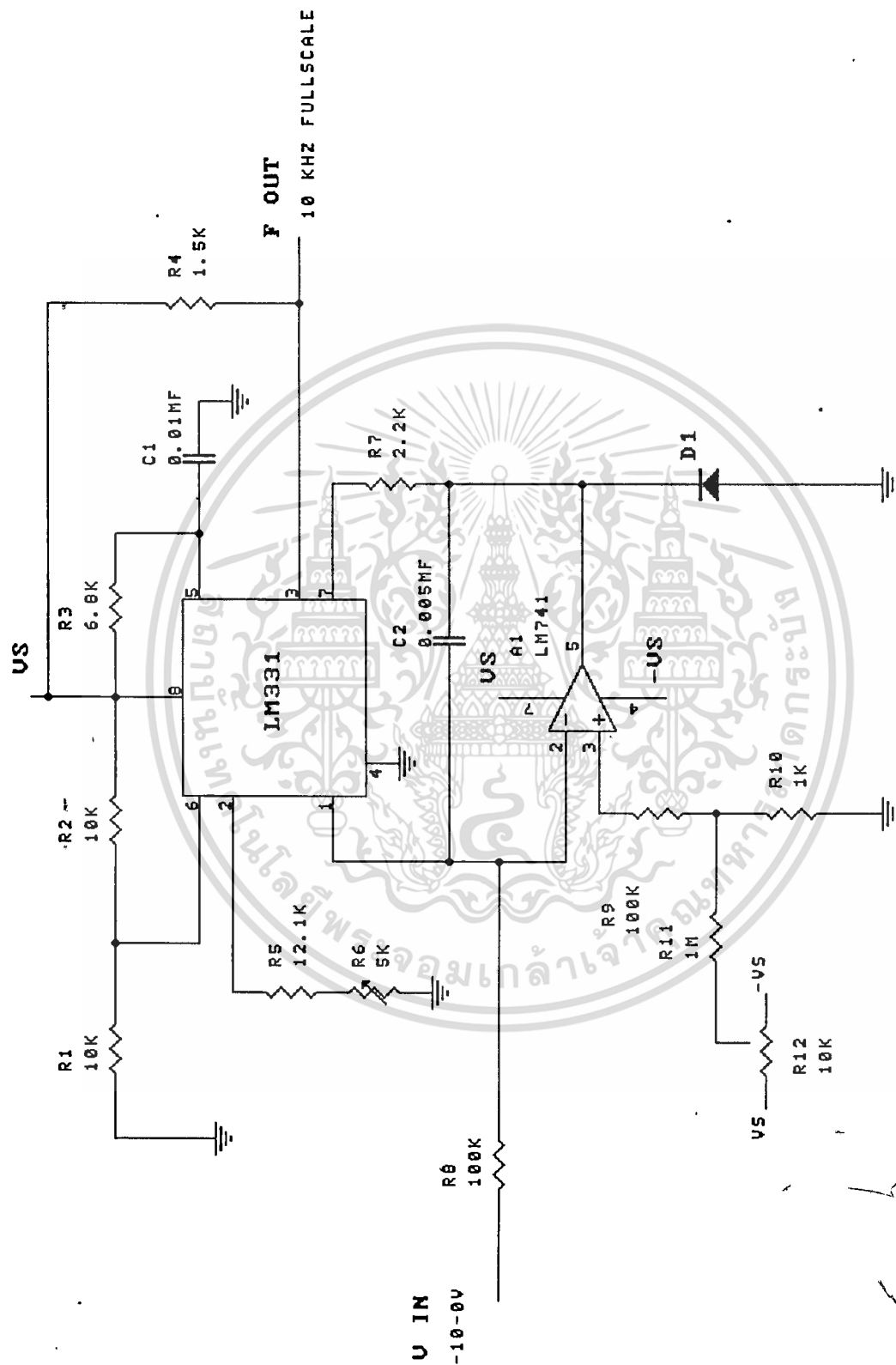
การปรับ Range สามารถปรับได้ที่ค่า R_s สำหรับวงจรนี้ อินพุต full scale = -10 Voutput full scale = 10 kHz

3.7 การเปลี่ยนความถี่เป็นแรงดันไฟฟ้า

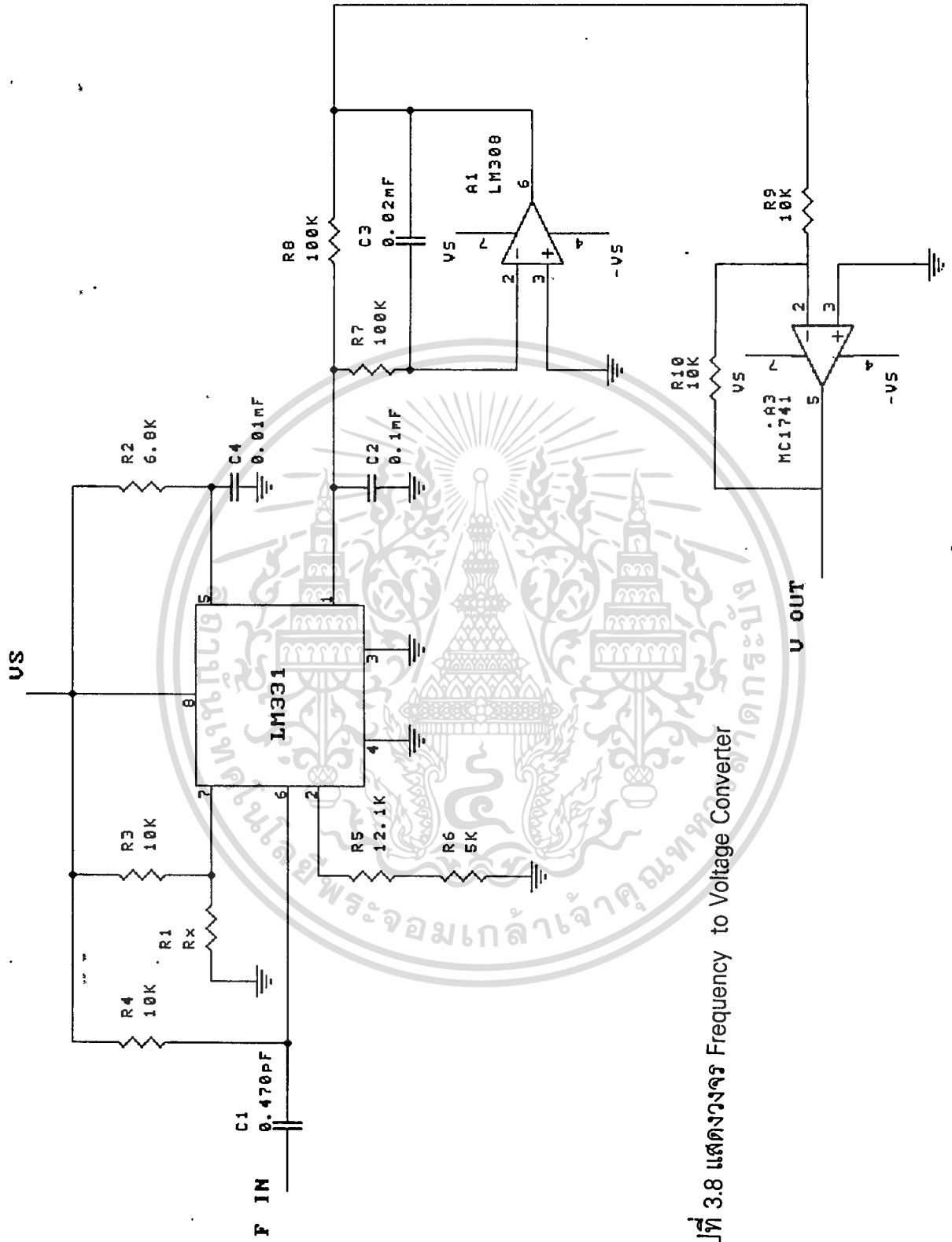
ส่วนวงจร แปลงความถี่เป็นแรงดันไฟฟ้า (Frequency to Voltage Converter) เป็นวงจรเปลี่ยนค่าความถี่เป็นค่าโวลต์เฉจ วงจรนี้ใช้ LM331 เช่นเดียวกับ วงจรแปลงแรงดันไฟฟ้าเป็นความถี่ โดยแปลงความถี่ที่ได้จากออปโตไอโซเลต เป็น DC ค่าลบ ค่าที่ได้ที่อัตราส่วนกลับกับวงจร แปลงแรงดันไฟฟ้าเป็นความถี่ คือ

$$V_{out} = -f_{in} * 2.09 * R_f * (R_t * C_t) / R_s$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงวงจร Voltage to Frequency Converter



รูปที่ 3.8 แสดงวงจร Frequency to Voltage Converter

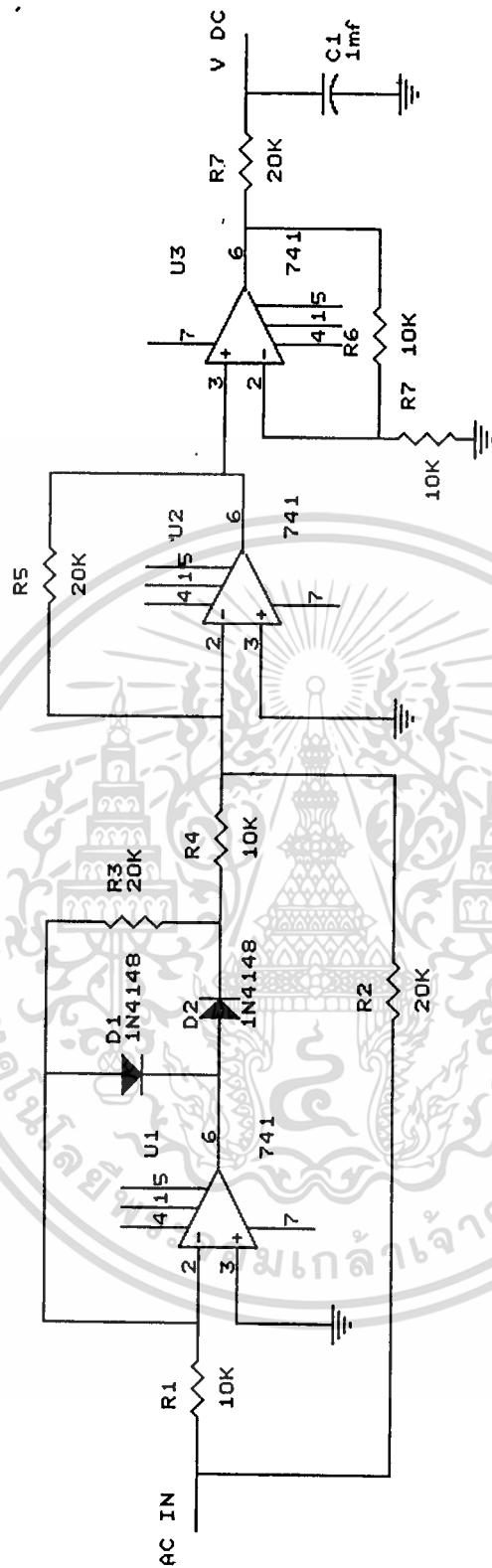
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การวัดแรงดันและกระแส

การวัดโวลต์เตจนี้ใช้ หลักการแบ่งแรงดัน โดยใช้อปแอมป์ร่วมกับไดโอด เป็น Precision full - wave rectifier ได้เอาท์พุทเป็นไฟ DC ค่าลบ แปรค่าตามไฟ กระแสสลับ ซึ่งเป็นอินพุทที่ได้จากการแบ่งแรงดัน

การวัดกระแสนั้นจะใช้ หม้อแปลงกระแส แปลงกระแสจากสาย กระแสสลับ ที่เข้าสู่โหลดมาเป็นกระแสนาณน้อยๆ ผ่านเข้าความต้านทาน แล้วนำโวลต์เตจที่คร่อมความต้านทานนั้น มาแปลงส่งให้ หน่วยประมวลผลกลาง ต่อไป





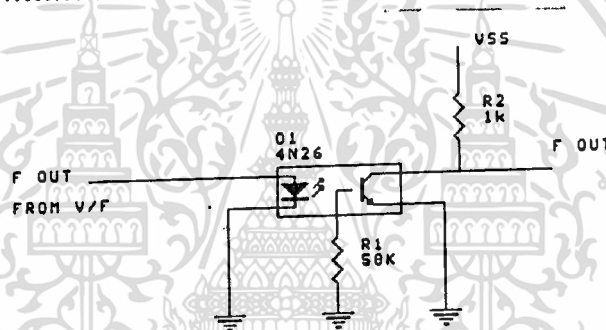
รูปที่ 3.9 วงจร Precision Full - wave rectifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 การเชื่อมต่อด้วยแสง(Opto isolator)

วงจรทางด้านกระแสลับมักจะสร้างสัญญาณรบกวนทางด้านกระแสตรงเสมอ ในกรณีที่ต่อกราวด์ร่วมกัน ดังนั้นจึงต้องแยกกราวด์จากกัน ในการแยกกราวด์ออกจากกันนั้นในมืออยู่หลายวิธี วิธีหนึ่งคือใช้ ออปโต ไอโซเลเตอร์ (Opto Isolator) โดยแบ่งออกเป็น 2 ส่วนคือด้านอินพุตและเอาต์พุต

ด้านอินพุตเป็นไดโอดเปล่งแสง ส่วนทางด้านเอาต์พุตเป็นโฟโตทรานซิสเตอร์ เมื่อมีกระแสไหลผ่านอินพุตไดโอดจะเปล่งแสงไปยังเบสของโฟโตทรานซิสเตอร์ทางด้าน เอาต์พุต ทำให้มีทรานซิสเตอร์อยู่ในสภาวะปิด มีกระแสไหลจากขาคอเล็กเตอร์ไปยัง อิมิเตอร์ ซึ่งเมื่อต่อวงจรดัง รูป 3.10 จะทำให้อินพุตและเอาต์พุตมีลักษณะตรงข้ามกัน แต่จะมีความถี่เท่ากันเสมอ

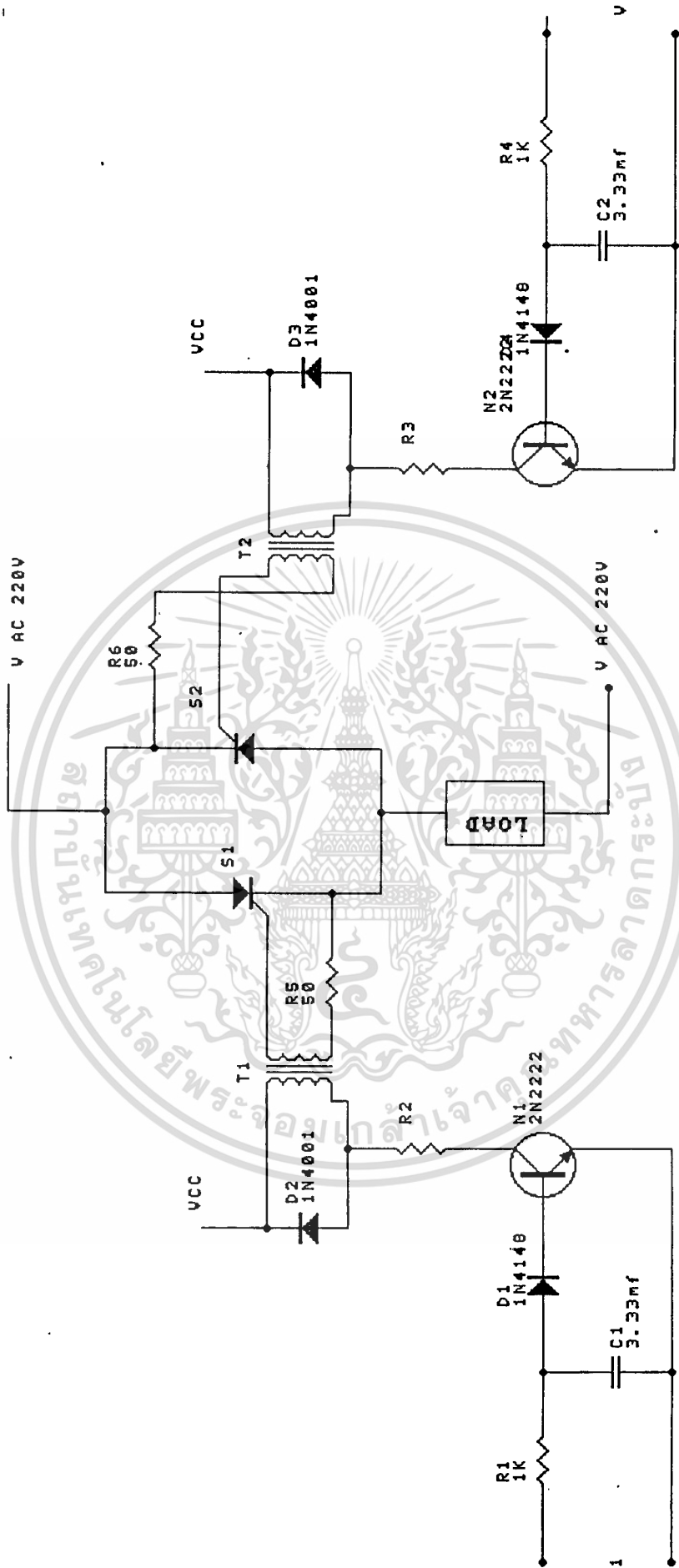


รูปที่ 3.10 การเชื่อมต่อด้วย ออปโตไอโซเรเตอร์

3.10 การเชื่อมต่อด้วย Pulse Transformer

การใช้พัลส์ทรานส์ฟอร์มเมอร์ เป็นการแยกกราวด์เช่นเดียวกับการใช้ออปโตไอโซเรเตอร์ ซึ่งประกอบด้วยขดลวด 2 ขด คือขดปฐมภูมิและขดทุติยภูมิ เมื่อมีกระแสผ่านขดปฐมภูมิจะเกิดฟลักซ์แม่เหล็กไปคล้องขดทุติยภูมิ ทำให้เกิดกระแสไหลในขดทุติยภูมิ ซึ่งกระแสที่ผ่านขดทุติยภูมินี้จะแปรค่าตามกระแสที่ไหลผ่านขดปฐมภูมิ

ในโครงการนี้ ใช้พัลส์ทรานส์ฟอร์มเมอร์เป็นตัวรับกระแสจาก TCA 785 เข้าไปยังด้านปฐมภูมิ ส่วนกระแสทางด้านทุติยภูมิจะไปทริก เอสซีอาร์(SCR) ทั้งสองตัว ซึ่ง เอสซีอาร์ (SCR) ต่อกลับข้ามกัน ทำให้ไฟบวกและไฟลบที่จ่ายให้แก่โหลดจะมากหรือน้อยขึ้นอยู่กับมุมทริกของ เอสซีอาร์ (SCR) มี วงจรลักษณะดังรูป 3.11



รูปที่ 3.11 เชื่อมต่อด้วย Pulse Transformer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 การเชื่อมต่อกับระบบไมโครคอนโทรลเลอร์

การเชื่อมต่อกับระบบไมโครคอนโทรลเลอร์นั้นเนื่องจากวงจรส่วนใหญ่จะมีสัญญาณเป็นอนาล็อกก่อนที่จะเข้าสู่ระบบไมโครคอนโทรลเลอร์นั้นจะต้องถูกแปลงสัญญาณจากวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลก่อนจากนั้นไมโครคอนโทรลเลอร์จึงจะนำข้อมูลเข้าสู่ขบวนการได้ ส่วนเมื่อไมโครคอนโทรลเลอร์ต้องการส่งข้อมูลเข้าสู่วงจรก็จะต้องผ่านวงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อกเสียก่อนที่จะนำสัญญาณไปควบคุมต่อไป

การติดต่อระหว่างไมโครคอนโทรลเลอร์กับวงจรภายนอกจะติดต่อผ่านทางพอร์ท 8255 ซึ่งมีทั้งหมดสองตัว ตัวละ 3 ไบต์ ซึ่งมีการเชื่อมต่อดังนี้

USERPORT1

- PORTA เชื่อมกับพอร์ทข้อมูลจอ LCD
- PORTB เชื่อมกับสายสัญญาณควบคุม LCD และสายสัญญาณควบคุมวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล
- PORTC เชื่อมกับสวิทช์ ควบคุม

USERPORT2

- PORTA เชื่อมกับพอร์ทข้อมูลวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล
- PORTB เชื่อมกับพอร์ทข้อมูลวงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อก
- PORTC ว่าง

3.12 การแสดงผลทางจอ LCD

ปัจจุบัน LCD เป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องมือเครื่องใช้ต่าง ๆ ทั้งนี้เนื่องจากมีความเหมาะสมด้วยประการทั้งปวง ทั้งในด้านการกินกระแสดำ สามารถแสดงผลเป็นตัวอักษรและตัวเลข หรือแสดงเป็นกราฟฟิคได้ (เฉพาะรุ่น) จะติดปัญหาก็คือในด้านวงจร ซึ่งมีระบบการทำงานที่ซับซ้อน และหาอุปกรณ์ได้ค่อนข้าง

ยาก แต่ขณะนี้ผู้ผลิต LCD จะทำรุ่นที่เป็น LCD MODULE ออกมา คือเป็น MODULE ที่มีตัว LCD และวงจรควบคุมมาให้พร้อม (เรียกว่า LCM) ซึ่งทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ง่าย และสะดวกสำหรับการเขียนโปรแกรม รวมทั้งมีจำหน่ายกันอย่างกว้างขวาง และมีราคาที่เหมาะสม

LCD MODULE มีอยู่มากมายหลายรุ่น และมีคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลักคือ แบบ DOT MATRIX และ GRAPHIC โดยแบบ DOT MATRIX จะแสดงผลเป็นตัวอักษรขนาด 5x8 DOT และมีจำนวนอักษรและบรรทัดแตกต่างกันไปในแต่ละรุ่น ส่วนแบบ GRAPHIC จะสามารถแสดงผลในรูปแบบ BIT-MAP คือจะสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ แนวทางในการใช้งานทั้ง 2 แบบ จะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบ DOT MATRIX มากกว่า เนื่องจากมีราคาถูก และเพียงพอต่องานส่วนใหญ่ คุณสมบัติของ DOT MATRIX LCD MODULE สามารถสรุปเป็นข้อ ๆ ได้ดังนี้

1. มีให้เลือกหลายรุ่นตามการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วย DOT MATRIX ขนาด 5x8 DOT
3. สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะ คือแบบ MEMORY MAP (20-PIN LCD BUS) และแบบผ่าน 8255 PORT (26-PIN 8255 BUS) โดยกรณี 26-PIN 8255 BUS จะใช้แผ่น PCB (DMCAD) เป็นตัว ADAPTER ทำให้เป็น 8255 BUS อีกที
4. การใช้งานง่ายและสะดวก ระบบไมโครคอนโทรลเลอร์เพียงแค่ส่งข้อมูลให้กับ LCD MODULE เท่านั้น ข้อความก็จะปรากฏบนแผงแสดง และจะคงค้างไว้ตลอด ทำให้ไม่ต้องเสียเวลาหลักของระบบไมโครคอนโทรลเลอร์
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น CLEAR DISPLAY, HOME CURSOR, ON-OFF CURSOR, BLINK CHARACTER และอื่น ๆ อีก
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว

ยาก แต่ขณะนี้ผู้ผลิต LCD จะทำรุ่นที่เป็น LCD MODULE ออกมา คือเป็น MODULE ที่มีตัว LCD และวงจรควบคุมมาให้พร้อม (เรียกว่า LCM) ซึ่งทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ง่าย และสะดวกสำหรับการเขียนโปรแกรม รวมทั้งมีจำหน่ายกันอย่างกว้างขวาง และมีราคาที่เหมาะสม

LCD MODULE มีอยู่มากมายหลายรุ่น และมีคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลักคือ แบบ DOT MATRIX และ GRAPHIC โดยแบบ DOT MATRIX จะแสดงผลเป็นตัวอักษรขนาด 5x8 DOT และมีจำนวนอักษรและบรรทัดแตกต่างกันไปในแต่ละรุ่น ส่วนแบบ GRAPHIC จะสามารถแสดงผลในแบบ BIT-MAP คือจะสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ แนวทางในการใช้งานทั้ง 2 แบบ จะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบ DOT MATRIX มากกว่า เนื่องจากมีราคาถูก และเพียงพอต่องานส่วนใหญ่ คุณสมบัติของ DOT MATRIX LCD MODULE สามารถสรุปเป็นข้อ ๆ ได้ดังนี้

1. มีให้เลือกหลายรุ่นตามการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วย DOT MATRIX ขนาด 5x8 DOT
3. สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะ คือแบบ MEMORY MAP (20-PIN LCD BUS) และแบบผ่าน 8255 PORT (26-PIN 8255 BUS) โดยกรณี 26-PIN 8255 BUS จะใช้แผ่น PCB (DMCAD) เป็นตัว ADAPTER ทำให้เป็น 8255 BUS อีกที
4. การใช้งานง่ายและสะดวก ระบบไมโครคอนโทรลเลอร์เพียงแค่ส่งข้อมูลให้กับ LCD MODULE เท่านั้น ข้อความก็จะปรากฏบนแผงแสดง และจะคงค้างไว้ตลอด ทำให้ไม่ต้องเสียเวลาหลักของระบบไมโครคอนโทรลเลอร์
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น CLEAR DISPLAY, HOME CUSOR, ON-OFF CURSOR, BLINK CHARACTER และอื่น ๆ อีก
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว

7. กินกระแสต่ำ และมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 V.
เท่านั้น

ขาสัญญาณของ LCD MODULE

PIN	SYMBOL	LEVEL	FUNCTION
	L		
1	Vss	—	0 V. GND
2	Vcc	—	+5 V. POWER SUPPLY
3	Vee	—	+V FOR LIQUID CRYSTAL DRIVE
4	RS	H/L	REGISTER SELECT H: DATA INPUT L: INSTRUCTION INPUT
5	RW	H/L	H: DATA READ L: DATA WRITE
6	E	H	ENABLE SIGNAL (L -> H)
7	DB0	H/L	DATA BUS BIT 0
8	DB1	H/L	DATA BUS BIT 1
9	DB2	H/L	DATA BUS BIT 2
10	DB3	H/L	DATA BUS BIT 3
11	DB4	H/L	DATA BUS BIT 4
12	DB5	H/L	DATA BUS BIT 5
13	DB6	H/L	DATA BUS BIT 6
14	DB7	H/L	DATA BUS BIT 7

ภาพแสดงผังเวลาของขาสัญญาณ

Item	Symbol	Measuring Condition	Standard Value			Unit
			min.	typ.	max	
Enable Cycle Time	TcycE	Figs.1, 2	100 0	-	-	nS
Enable Pulse Width, High Level	PWEH	Figs.1, 2	450	-	-	nS
Enable Rise and Decay Time	tEr,tEf	Figs.1, 2	-	-	25	nS
Address Setup Time, RS, R/W-E	tAS	Figs.1, 2	140	-	-	nS
Data Delay Time	tDDR	Fig.2	-	-	320	nS
Data Setup Time	tDSW	Fig.1	195	-	-	nS
Data Hold Time	tH	Fig.1	10	-	-	nS
Data Hold Time	tDHR	Fig.2	20	-	-	nS
Address Hold Time	tAH	Figs.1, 2	10	-	-	nS

*Vcc = 5.0 V. $\pm$ 5 %, Ta = 25°C

การต่อเข้ากับระบบไมโครโปรเซสเซอร์

LCD MODULE จะต่อเข้ากับระบบไมโครโปรเซสเซอร์ได้ 2 ลักษณะ คือแบบ MEMORY MAP โดยผ่าน LCD BUS ขนาด 20 PIN และแบบ I/O PORT โดยผ่าน 8255 BUS ขนาด 26 PIN ซึ่งทั้งสองแบบนี้จะมีข้อดีและข้อด้อยแตกต่างกันไป โดยแต่ละแบบจะมีหลักการดังนี้

การต่อแบบ MEMORY MAP

1. สามารถต่อเข้ากับ CHIP เบอร์ทั่ว ๆ ไปได้ เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโครคอนโทรลเลอร์มองเห็น LCD MODULE ในลักษณะของ MEMORY ได้ทันที
2. ผู้ใช้สามารถเขียนและอ่านข้อมูลจาก LCD MODULE ได้ ทำให้มองเสมือนว่าเป็น MEMORY BUFFER ไปในตัว
3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบ FLAG ความพร้อมในขณะที่ LCD MODULE กำลังทำงานได้
4. ใช้ได้กับบอร์ดที่มี LCD BUS มาให้พร้อมเท่านั้น
5. ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่งและต้องมีการ DECODE ละเอียดพอสมควร
6. การจัดขาสัญญาณจะต้องเป็นไปตามแบบของ CHIP แต่ละเบอร์ด้วย

การต่อแบบ I/O PORT

1. สามารถต่อเข้ากับ I/O PORT ใด ๆ ก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมาให้ตรงกับข้อกำหนดของ LCD MODULE
2. ผู้ใช้จะเขียนข้อมูลให้ LCD MODULE ได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนด MEMORY ส่วนหนึ่งให้เป็นเสมือน BUFFER ให้กับ LCD MODULE อีกที
3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบไมโครคอนโทรลเลอร์เองเพื่อรอให้ LCD MODULE กระทำขบวนการต่าง ๆ
4. ใช้ได้กับบอร์ดทั่ว ๆ ไปที่มี PORT
5. ไม่เปลืองส่วนของ MEMORY ในการใช้งาน
6. การจัดขาสัญญาณกระทำได้อย่างอิสระ
ซึ่งในโครงการนี้เลือกใช้แบบ I/O MAP

ชุดคำสั่งควบคุมและการแสดงข้อความ

การเขียนหรืออ่านข้อมูลกับ LCD MODULE ก็คือการกำหนดคุณสมบัติต่าง ๆ ในการใช้งานของ LCD ตามชุดคำสั่งควบคุม และรวมถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดตามตารางต่อไปนี้

ความเข้าใจพื้นฐาน

1. การเขียนข้อมูลให้กับ LCD MODULE จะแบ่งเป็น 2 ลักษณะคือ INSTRUCTION และ DATA โดยจะกำหนดด้วยขาสัญญาณ RS คือถ้า $RS = 0$ จะหมายถึงส่งสัญญาณควบคุม (INSTRUCTION) หรืออ่านค่า FLAG สภาพการทำงานของ LCD MODULE และถ้า $RS = 1$ จะหมายถึงการเขียนหรืออ่าน DATA กับ LCD MODULE

2. หลักการในการเขียนข้อมูลให้ LCD MODULE นี้ คือเมื่อมีการเขียนข้อมูลไปแล้ว ตัว LCD MODULE จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง (ตามค่า EXECUTE TIME ในตาราง) ซึ่งระบบไมโครคอนโทรลเลอร์สามารถตรวจสอบได้จาก BUSY FLAG (BF) และถ้าเรียบร้อยแล้ว จึงจะสามารถเขียนข้อมูลต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบ I/O PORT คือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครคอนโทรลเลอร์ก็ต้องใช้วิธีการหน่วงเวลาแทน

3. การเขียนข้อมูลให้กับ LCD MODULE นี้ สามารถทำได้ทั้งแบบ 8 BIT และ 4 BIT โดยกรณี 4 BIT จะใช้สายสัญญาณ DATA เพียง 4 เส้น คือ DB4 - DB7 (ใช้สำหรับระบบไมโครคอนโทรลเลอร์แบบ 4 BIT หรือเพื่อการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 BIT เพียงแต่ให้เขียน 2 ครั้ง คือ DB4 - DB7 ก่อน แล้วตามด้วย DB0 - DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในคำสั่ง FUNCTION SET ด้วย

4. DDRAM (DISPLAY DATA RAM) คือหน่วยความจำภายในตัว LCD MODULE สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง (8 ตัว) โดยจะอ้าง ADDRESS ได้ทั้งหมด 64 BYTE คือ 8 ตัวอักษร คูณกับ 8 ROW

รายละเอียดของแต่ละคำสั่ง

1. CLEAR DISPLAY

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

สำหรับการ CLEAR DISPLAY โดยจะทำการเขียนตัวอักษร SPACE ลงใน DDRAM ทั้งหมด และกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะกลับไปที่ตำแหน่งซ้ายบนสุดของจอภาพ

2. CURSOR AT HOME

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

สำหรับกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

3. ENTRY MODE SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

- I/D = 0 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ DECREMENT
- I/D = 1 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ INCREMENT
- S = 0 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะถูกเลื่อนไปทิศทางตามค่า I/D

$S = 1$ เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะอยู่กับที่ และตัวอักษรจะถูกดันไปทิศทางตามค่า I/D

การกำหนด I/D และ S นี้ ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้ว จะต้องไม่ใช่คำสั่ง CLEAR DISPLAY อีก

4. DISPLAY ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

$D = 0$ กำหนดให้ OFF DISPLAY

$D = 1$ กำหนดให้ ON DISPLAY

$C = 0$ กำหนดให้ OFF CURSOR

$C = 1$ กำหนดให้ ON CURSOR โดย CURSOR จะเป็นเส้นชี้ได้ตัวอักษร

$B = 0$ กำหนดให้ไม่มีการกระพริบที่ตำแหน่ง CURSOR

$B = 1$ กำหนดให้มีการกระพริบที่ตำแหน่ง CURSOR (กระพริบเป็นรูปสี่เหลี่ยม)

5. DISPLAY SHIFT

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

$S/C = 0$ กำหนดให้เลื่อน CURSOR ตามทิศทาง R/L ไป 1 ตำแหน่ง

$S/C = 1$ กำหนดให้เลื่อนข้อความบนแผงแสดงตามทิศทาง R/L ไป 1 COLUMN (เลื่อนทุกบรรทัด)

$R/L = 0$ กำหนดให้มีทิศทางไปทางซ้าย

8. SET DDRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

สำหรับการกำหนด ADDRESS ของ DDRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที ตำแหน่งของ ADDRESS ในแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนอักขรต่อบรรทัดไม่เท่ากัน ซึ่งแสดงดังภาพต่อไปนี้

รุ่น DMC164

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F

9. BUSY FLAG AND ADDRESS READ

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	ADDRESS						

สำหรับการอ่านค่า BF (BUSY FLAG) ซึ่งบอกถึงความพร้อมของ LCD MODULE ในการรับข้อมูล ถ้า BF = 0 หมายถึงว่าพร้อมที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายถึงว่ายังไม่พร้อม นอกจากนี้ ยังเป็นการอ่านค่า ADDRESS ของ CGRAM หรือ DDRAM ด้วย

การอ่านและเขียนข้อมูลกับ DDRAM / CGRAM

1. WRITE DATA TO DDRAM OR CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	DATA							

สำหรับการเขียนข้อมูลลงหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

2. READ DATA FROM DDRAM OR CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	DATA							

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่ม หรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

การโหลดอักขรที่ออกแบบเอง

LCD MODULE จะสามารถโหลดตัวอักษรที่ออกแบบเองได้ โดยกระทำได้ 8 ตัวอักษรซึ่งหลักการก็คือ ให้ทำการโหลดรูปแบบที่กำหนดไว้แล้วลงในหน่วยความจำ CGRAM นั่นเอง ADDRESS ที่อ้างถึง CGRAM จะใช้เพียง 6 BIT โดยอ้างได้ทั้งหมด 64 BYTE (คือ 8 ตัวอักษร คูณกับ 8 ROW) และการเรียกใช้ตัวอักษรที่ออกแบบเองนี้ จะใช้รหัส 00-07 (หรือ 08-0F ก็ได้) ขั้นตอนในการโหลดตัวอักษรจะเป็นดังนี้

R/L = 1 กำหนดให้มีทิศทางไปทางขวา

6. FUNCTION SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้การติดต่อกับ LCD MODULE เป็นแบบ 4 BIT

DL = 1 กำหนดให้การติดต่อกับ LCD MODULE เป็นแบบ 8 BIT

จะสังเกตว่า การกำหนดค่า DL นี้ สามารถกระทำได้ที่ DB4 - DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 BIT ตั้งแต่ครั้งแรก หลังจากจ่ายไฟเลี้ยง ก็จะทำให้ LCD MODULE มีการรับข้อมูลแบบ 4 BIT ทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 DUTY และ 1/11 DUTY

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 DUTY

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5x7 DOTS

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5x10 DOTS (กรณีที่ LCD MODULE เป็นแบบ 5x7 อยู่แล้ว ก็จะไม่เกิดอะไร)

7. SET CGRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

สำหรับการกำหนด ADDRESS ของ CGRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่าน และเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที

1. กำหนด ADDRESS ของ CGRAM ด้วยคำสั่ง SET CGRAM ทั้งนี้ถ้าโหลดอักขรตัวแรกก็จะใช้ ADDRESS ที่ตำแหน่ง 0

2. เขียนข้อมูลที่เป็นรูปแบบของตัวอักษร ด้วยคำสั่ง WRITE DATA โดยรูปแบบนี้จะใช้ 8 BYTE ต่อหนึ่งตัวอักษร คือเรียง ROW จากแถวบนลงล่าง และใช้ BIT ที่ 0-4 เท่านั้น การโหลดนี้จะทำได้ต่อเนื่องไปเรื่อย ๆ โดยไม่ต้องกำหนด ADDRESS ใหม่

3. ตัวอักษรที่โหลดลงไปแล้ว สามารถเรียกใช้ได้ตลอดไปในขณะที่ยังมีไฟเลี้ยงอยู่ ซึ่งการโหลดนี้ให้กระทำที่ส่วนต้นของโปรแกรมได้เลย รูปแบบของตัวอักษรมีตัวอย่างดังนี้

	ADDRESS	DATA
	00	04
	01	0E
	02	1F
	03	1F
	04	0E
	05	04
	06	04
	07	1F

3.13 โปรแกรมควบคุมการทำงาน

โปรแกรมควบคุมการทำงานจะมีหน้าที่ควบคุมการทำงานของ TCA785 และติดต่อสื่อสารกับผู้ใช้ระบบผ่านทางจอ LCD ซึ่งจะมีเมนูหลักดังรูป 3.12

VOLT V >..... V		
POW.....KW >..... KW		
CURRENT A		
12:00	PG/MAN	MENU

รูปที่ 3.12 แสดงเมนูหลักของระบบ

โหมดการทำงานของโปรแกรมนี้มีอยู่ 4 โหมด ด้วยกัน คือ

1. Set Time Mode เป็นโหมดที่ใช้ในการตั้งเวลาปัจจุบันให้กับระบบดังรูป

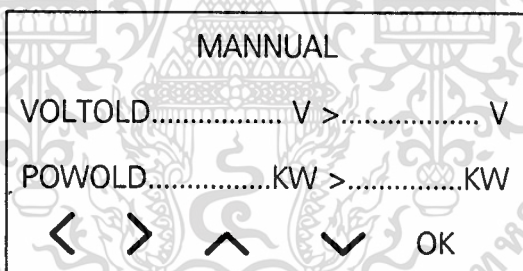
รูป 3.13



รูปที่ 3.13 แสดงหน้าจอของโหมดตั้งเวลา

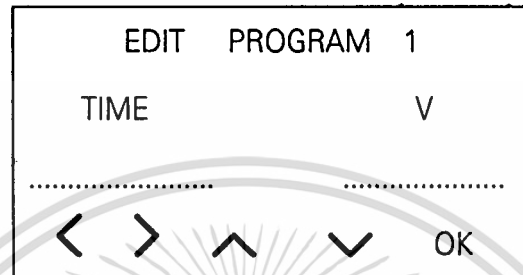
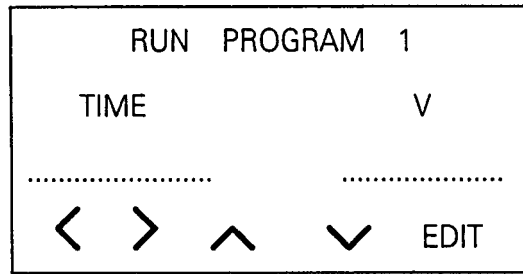
2. Manual Control Mode เป็นโหมดที่ใช้ป้อนค่าแรงดันที่ต้องการให้กับระบบดัง

รูป 3.14



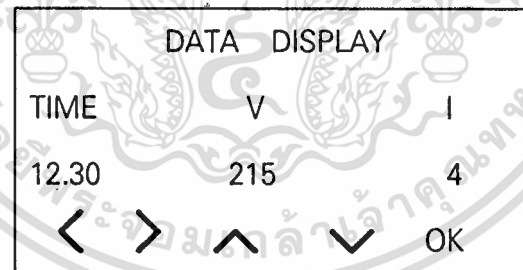
รูปที่ 3.14 แสดงหน้าจอของโหมด Manual

3. Program Control Mode เป็นโหมดที่ใช้เลือกโปรแกรมการทำงานของระบบซึ่งมีให้ เลือก 3 แบบ และยังสามารถแก้ไขโปรแกรมทั้ง 3 โดยการตั้งเวลาและค่าแรงดันที่ต้องการ ณ เวลานั้นได้ทันที ดังรูป 3.15



รูปที่ 3.15 แสดงหน้าจอของโหมด Program

4. Display Data Mode เป็นโหมดที่แสดงข้อมูลที่เก็บไว้ของวันนั้นๆออกมาทางจอ LCD ดังรูป 3.16



รูปที่ 3.16 แสดงหน้าจอของโหมด Display Data

การแสดงผลสถานะของระบบ

1. แรงดันไฟฟ้าที่วัดได้ และที่ตั้งไว้
2. กระแสที่วัดได้ และที่ตั้งไว้
3. ค่าพลังงานที่คำนวณได้
4. เวลาปัจจุบัน

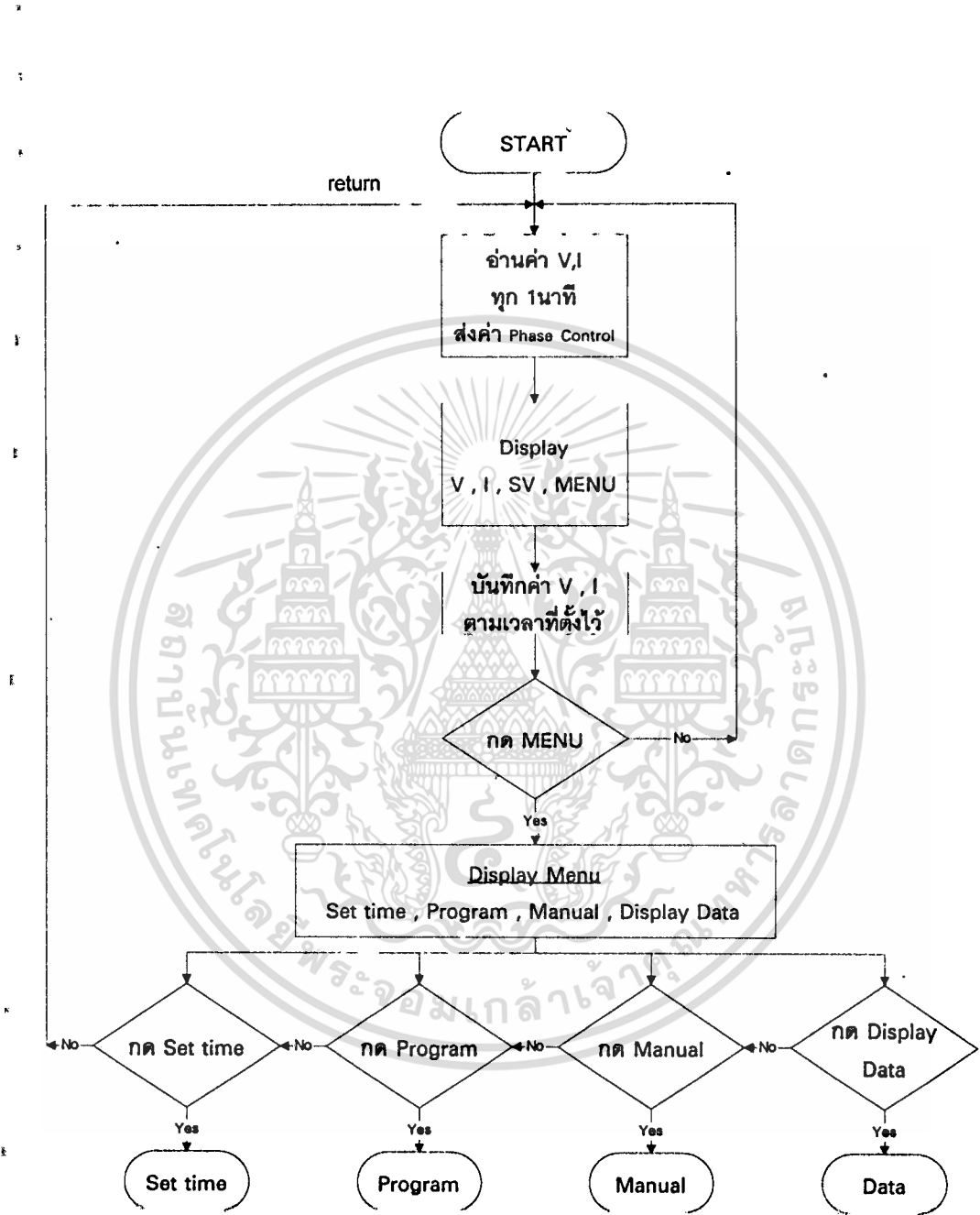
การเก็บข้อมูล

เนื่องจากการเก็บข้อมูลจำนวนมากเพื่อนำมาศึกษาสภาพการต่างๆของระบบ โดยผู้ใช้งาน ข้อมูลต่างๆจะถูกบันทึกในหน่วยความจำชั่วคราว (RAM) และแสดงผลออกทางหน้าจอ LCD ซึ่งข้อมูลนั้นจะมีรายละเอียดของเวลา แรงดันไฟฟ้าที่วัดได้ และกระแสไฟฟ้าที่วัดได้แสดงให้แก่ผู้ใช้โดยมีรูปแบบการเก็บข้อมูลเป็นดังรูป 3.17

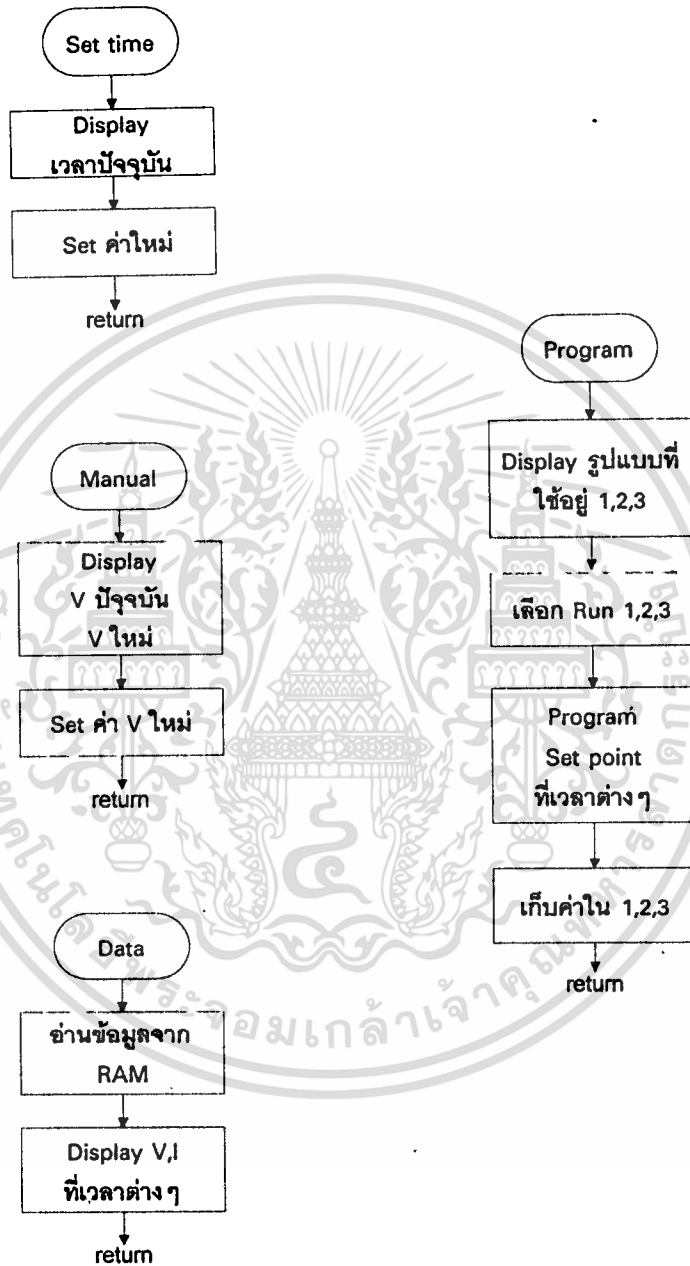
DI2a	F800H<DPTR
DI1a	
DV2a	
DV1a	
THa	
TMa	เวลานาที
:	:
:	:
DI21	ค่ากระแสไบต์ต่ำ
DI11	ค่ากระแสไบต์สูง
DV2 ₁	ค่าแรงดันไบต์ต่ำ
DV1 ₁	ค่าแรงดันไบต์สูง
TDH ₁	เวลาชั่วโมง
TDM ₁	02FFH

รูปที่ 3.17 รูปแบบการเก็บข้อมูลในหน่วยความจำสำรอง

โปรแกรมทั้งหมดสามารถแสดงการทำงานด้วยโฟลชาร์ตดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

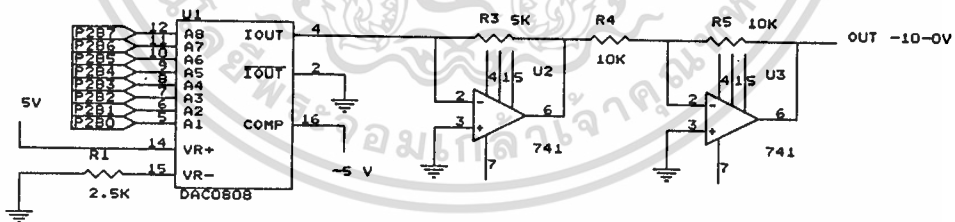


บทที่ 4

การทดลองและผลการทดลอง

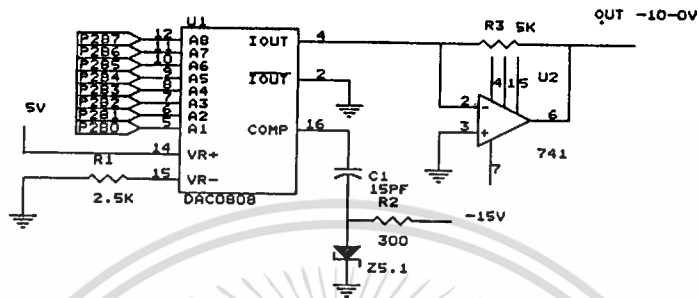
การทดลองวงจรเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก การทดลองครั้งที่ 1

กระแส I_{out} ที่ขา 4 ขึ้นอยู่กับกระแสอ้างอิงที่ขา 14 ปรับกระแสอ้างอิงเท่ากับ 2 mA จะได้ว่ากระแสเอาต์พุตที่ขา 4 สูงสุดเท่ากับ 2 mA ด้วย เนื่องจากต้องการโวลต์เต็มสูงสุดที่เอาต์พุตเท่ากับ 10 โวลต์ ดังนั้นความต้านทานที่คร่อมออปแอมป์จึงประมาณ 2.5 K จากรูปที่ 4.1 ป้อนไบนารีอินพุตที่ขา D_0 ถึง D_7 ค่าต่างๆ และวัดสัญญาณที่เอาต์พุต ปรากฏว่าเกิดออสซิลเลตไม่ว่าไบนารีที่เป็นอินพุตจะเป็นเท่าใดก็ตาม



รูปที่ 4.1 วงจรเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกในการทดลองครั้งที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 วงจรเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกในการทดลองครั้งที่ 2

การทดลองครั้งที่ 2

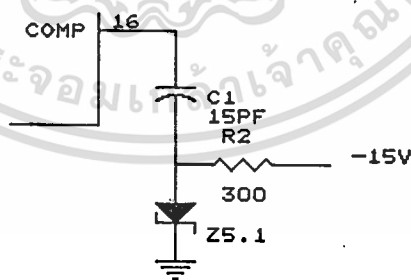
ทดลองป้อนไฟเลี้ยง -5 โวลท์ที่ขา 3 โดยมาจากซีเนอร์ไดโอดแทน -5 โวลท์เดิม ที่มาจากไอซีเรกูเรเตอร์ 7805 จากนั้นทำการทดลองป้อนสัญญาณไบนารี อินพุตค่า ต่างๆ และวัดค่าเอาต์พุต จะได้ผลการทดลองตามตารางที่ 4.1

หมายเหตุ

โวลท์เตจ + 5 โวลท์ที่มาจากขา 13 ของไอซี เรกูเรเตอร์เบอร์ 7805 ส่วนโวลท์เตจ -5 โวลท์ที่ขา 3 มาจากซีเนอร์ไดโอด 5.1 โวลท์ แสดงดังรูปที่ 4.1.

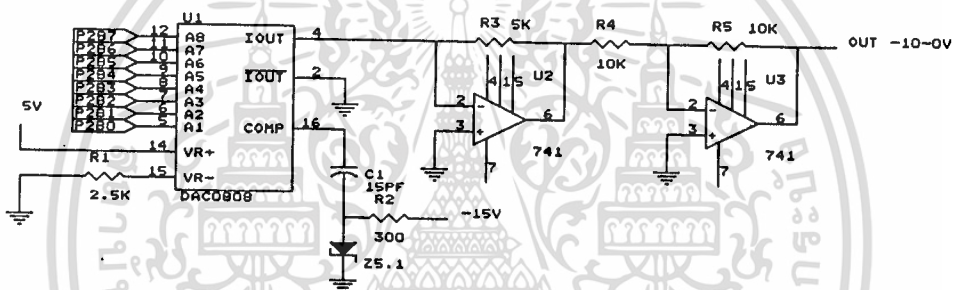
ตารางที่ 4.1 แสดงผลการทดลองของวงจรเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก

BINARY INPUT	ANALOG OUTPUT (V)
0000 0000	0.000
0000 0001	0.040
0000 0010	0.080
0000 0011	0.120
0000 0100	0.160
0000 0101	0.200
0000 0110	0.240
0000 0111	0.280
0000 1000	0.328
0111 1111	5.000
1111 1111	10.000



รูปที่ 4.3 วงจรโวลต์เตจเรกกูเรเตอร์ของไฟลอปที่เลี้ยงไอซี DAC0808

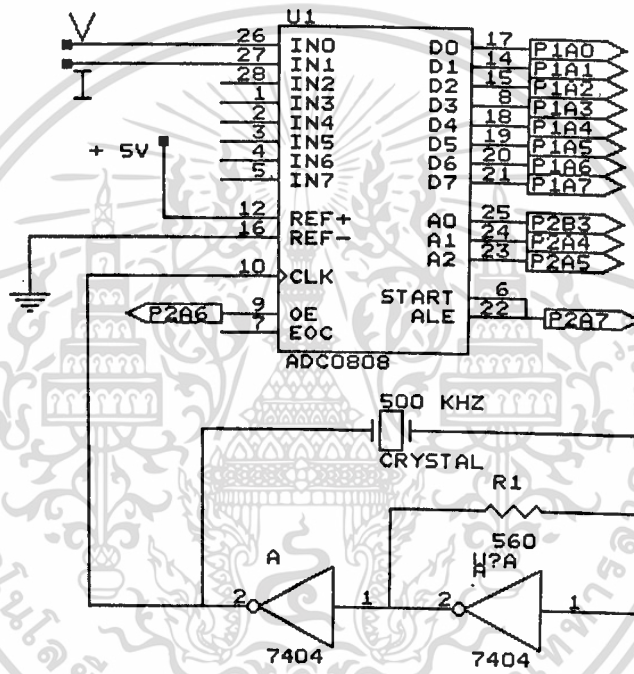
เนื่องจากเอาต์พุตของวงจรรูปที่ 2 จะได้ค่าโวลต์เตจ 0-10 โวลต์ ซึ่งเอาต์พุตที่จะนำไปใช้คือ -10 - 0 โวลต์ ดังนั้นจึงต้องใส่อินเวอร์เตอร์ที่เอาต์พุตดังนั้นวงจรเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกที่ใช้คือ รูปที่ 4.4



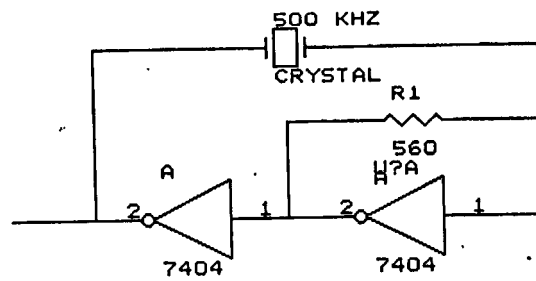
รูปที่ 4.4 วงจรเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกเมื่อใส่วงจรอินเวอร์เตอร์ที่เอาต์พุต

การทดลองเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล

ใช้ไอซีเบอร์ ADC0809 โดยมีวงจรดังรูปที่ 4.5 ซึ่งต้องใช้สัญญาณนาฬิกาที่มีความถี่ 500 KHz ทดลองต่อวงจรสร้างสัญญาณนาฬิกาโดยใช้วงจรออสซิลเลเตอร์ โพสิทีฟฟีดแบค (OSC POSITIVE FEEDBACK) ดังรูปที่ 4.6 จากนั้นวัดสัญญาณที่เอาต์พุต

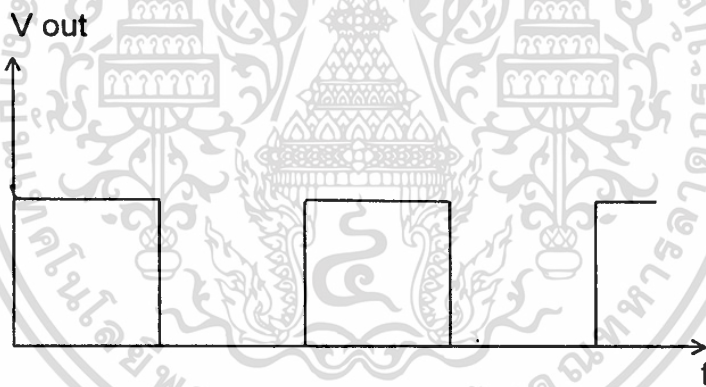


รูปที่ 4.5 วงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล



รูปที่ 4.6 วงจรออสซิลเลเตอร์ โพลีทีฟฟิแบค (OSC POSITIVE FEEDBACK)

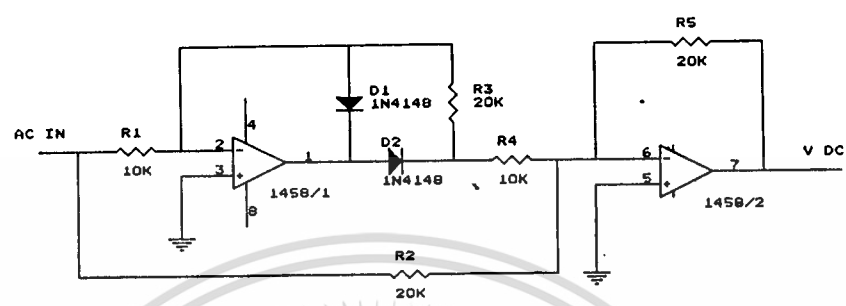
ผลการทดลองได้สัญญาณสี่เหลี่ยม ที่มีความถี่ 500 KHz พอดีแต่เกิดโอเวอร์ชูตเล็กน้อย ดังรูปที่ 4.7



รูปที่ 4.7 ลักษณะสัญญาณออกจากวงจรออสซิลเลเตอร์ โพลีทีฟฟิแบค

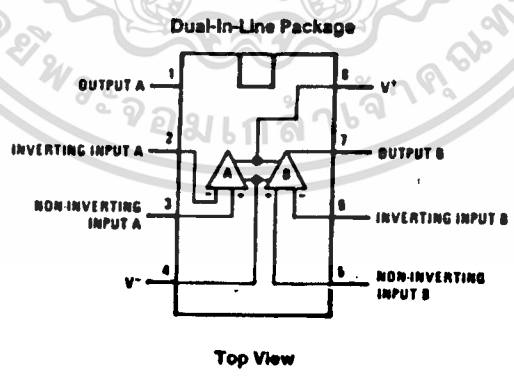
การทดลองเปลี่ยนแรงดันกระแสกลับไปเป็นกระแสตรง

เนื่องจากวัดแรงดันจะแบ่งแรงดันมาจากโหลด ดังนั้นต้องมีการเปลี่ยนแรงดันจากกระแสกลับให้เป็นกระแสตรงเพื่อใช้ในการวัดและการควบคุม โดยการทดลองต่อวงจรปริซึ้นฟูลเวฟเรกติไฟล์ (PRECISION FULLWAVE RECTIFIER) ดังรูปที่ 4.8



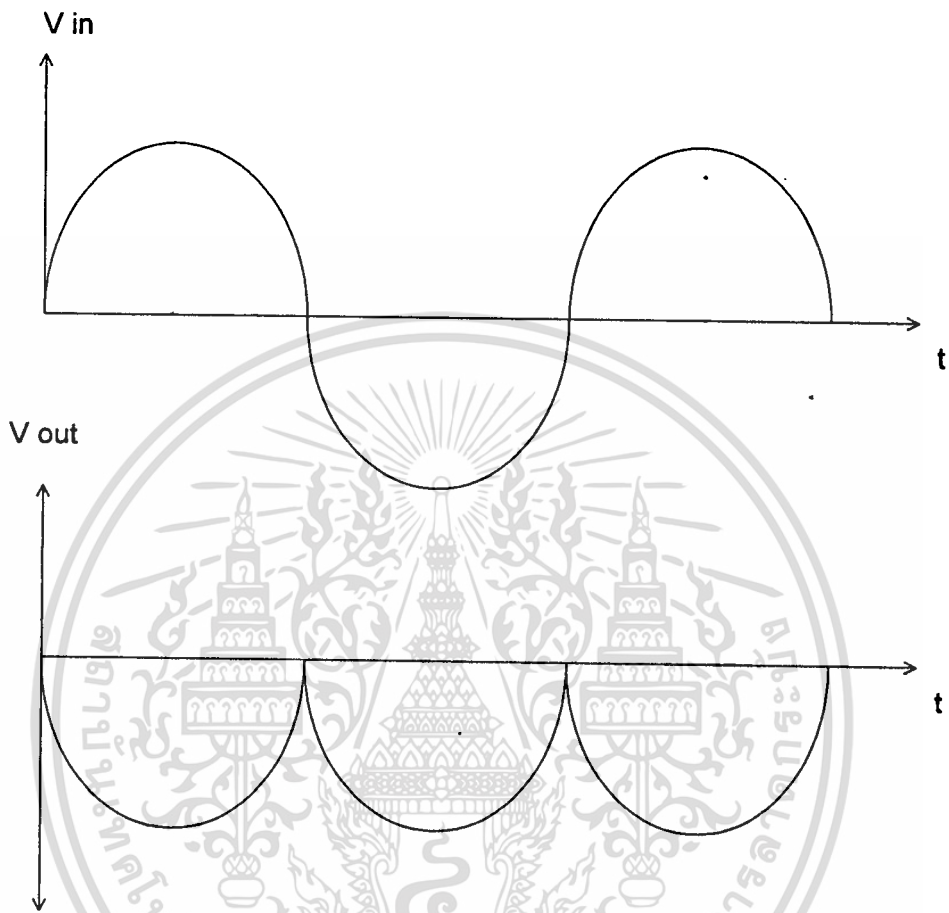
รูปที่ 4.8 วงจรพรีซีชันฟูลเวฟเรกติไฟเออร์

จากวงจรรูปที่ 8 ใช้ไอซีเบอร์ LM1458 เป็นออฟแอมป์ ซึ่งมีรายละเอียดของขา ดังรูปที่ 4.9



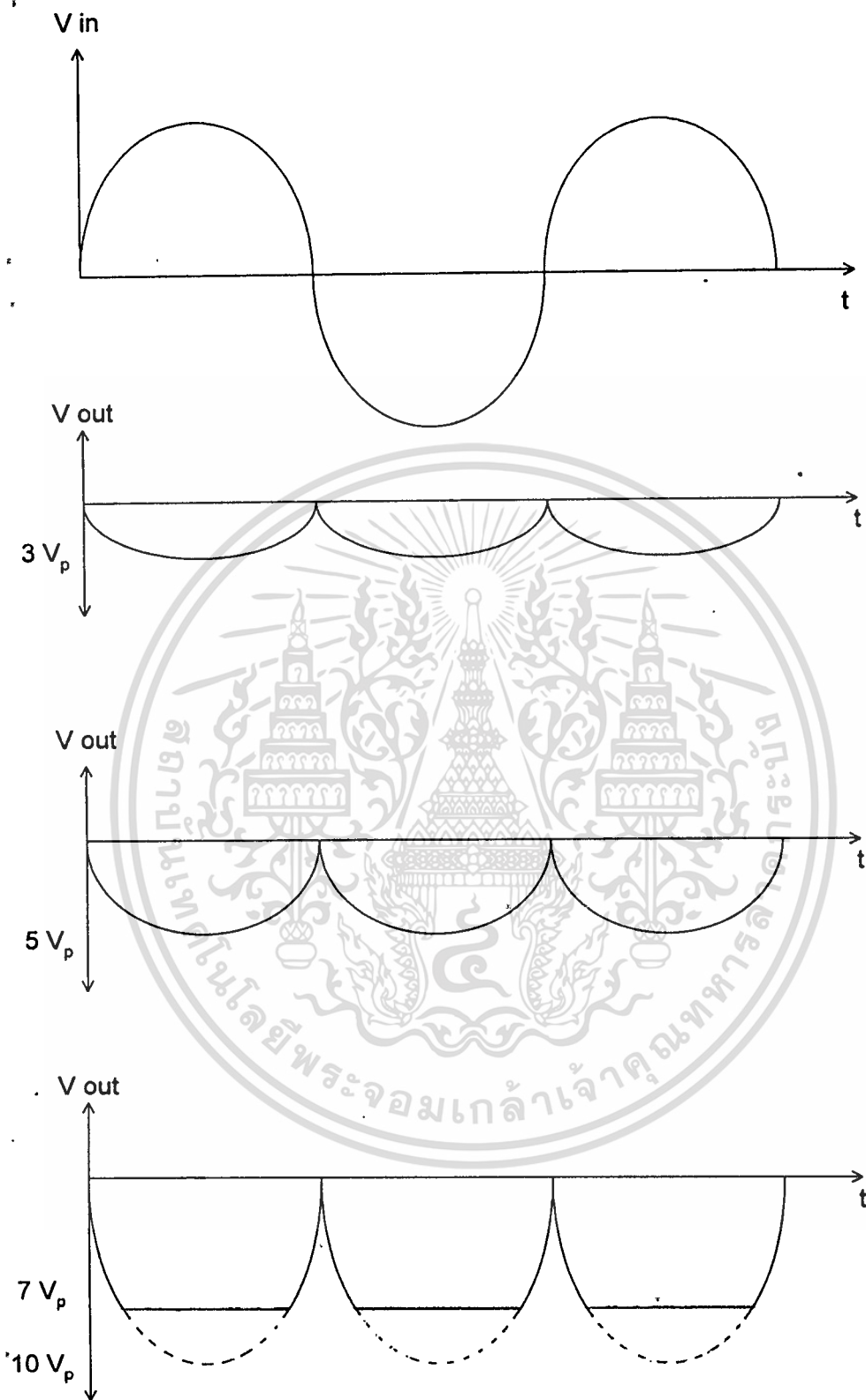
รูปที่ 4.9 รายละเอียดของ LM 1485

จากทดลองวัดแรงเอาต์พุตที่ขา 7 ของ LM1458 จะได้กราฟแรงดัน ซึ่งเปรียบเทียบกับสัญญาณอินพุตดังรูปที่ 4.10



รูปที่ 4.10 รูปเปรียบเทียบสัญญาณอินพุตและเอาต์พุตจากวงจรพรีซีชันฟูลเวฟเรกติไฟ์

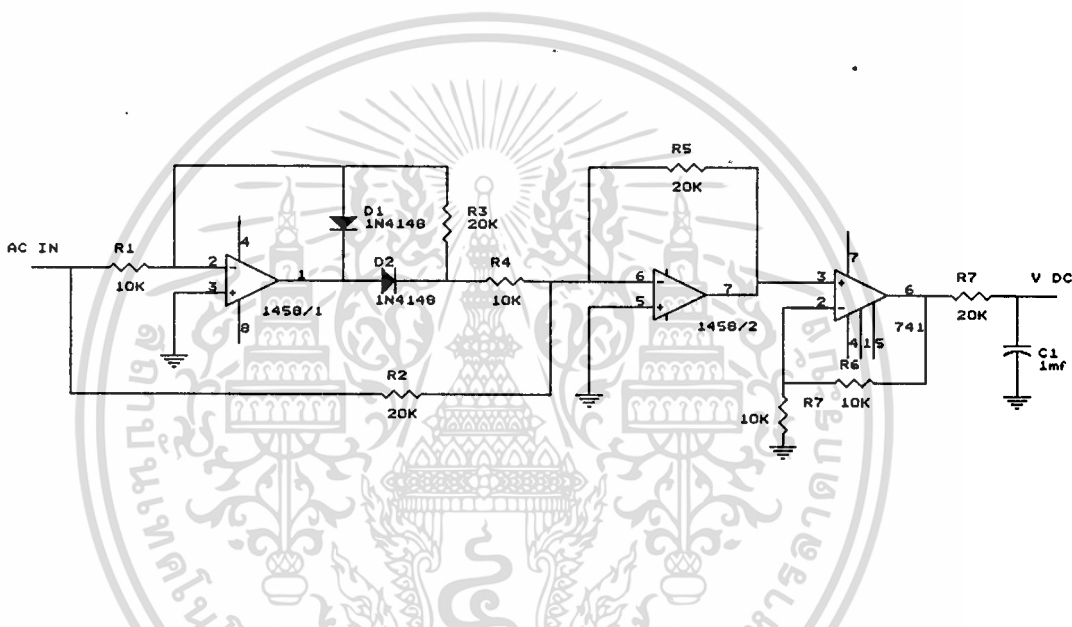
และจากการทดลองป้อนอินพุตที่เป็นแรงดันกระแสลับตั้งแต่ 0 โวลท์พีค ไปจนถึง 10 โวลท์พีค ปรากฏว่าได้แรงดันเอาต์พุตที่ขา 7 ของ LM1458 เป็นฮาร์ฟเวฟ (HALF WAVE) ดังรูปที่ 4.11



รูปที่ 4.11 รูปเปรียบเทียบสัญญาณอินพุตและเอาต์พุตจากวงจรฟูลเวฟเรกติไฟเลอร์

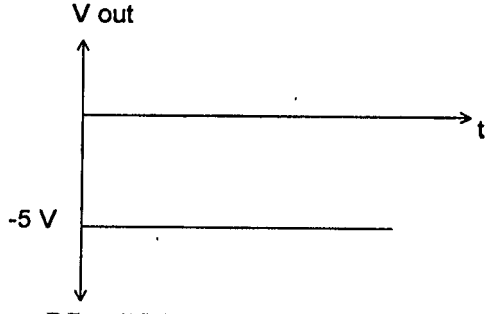
ซึ่งจากการทดลองจะได้ว่าอินพุตแรงดันกระแสกลับมีค่าได้สูงสุดประมาณ 7 โวลท์พีคที่ทำให้เอาต์พุตที่ขา 7 ของ LM1458 เป็นฮาร์ฟเวฟ

เนื่องจากเอาต์พุตที่จะนำไปใช้นี้มีค่าแรงดันต่ำเกินไปจึงต้องใช้วงจรขยาย นอนอินเวอร์ตติง (NON-INVERTING) ที่มีอัตราขยาย 2 เท่าต่อเข้ากับเอาต์พุตที่ขา 7 ของ LM1458 และเอาต์พุตที่ต้องนำไปใช้ต้องเป็นแรงดันกระแสตรงที่มีค่าติดลบดังนั้นต้องต่อฟิลเตอร์ (FILTER) ดังรูปที่ 4.12



รูปที่ 4.12 วงจรพรีซีชันฟูลเวฟเรกติไฟเลอร์ที่สมบูรณ์

ทดลองป้อนแรงดันกระแสกลับที่อินพุตได้สัญญาณเอาต์พุตที่เป็นแรงดันกระแสตรงดังรูปที่ 4.13

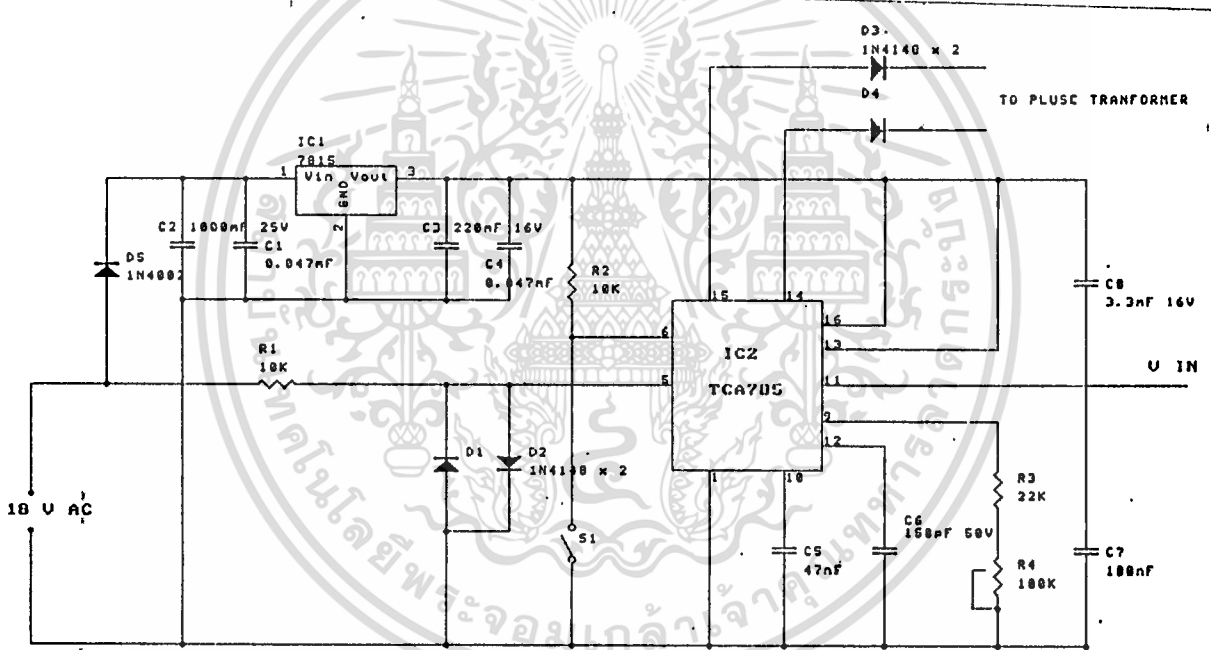


รูปที่ 4.13 สัญญาณเอาต์พุตจากวงจรพรีซีชันฟูลเวฟเรกติไฟเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

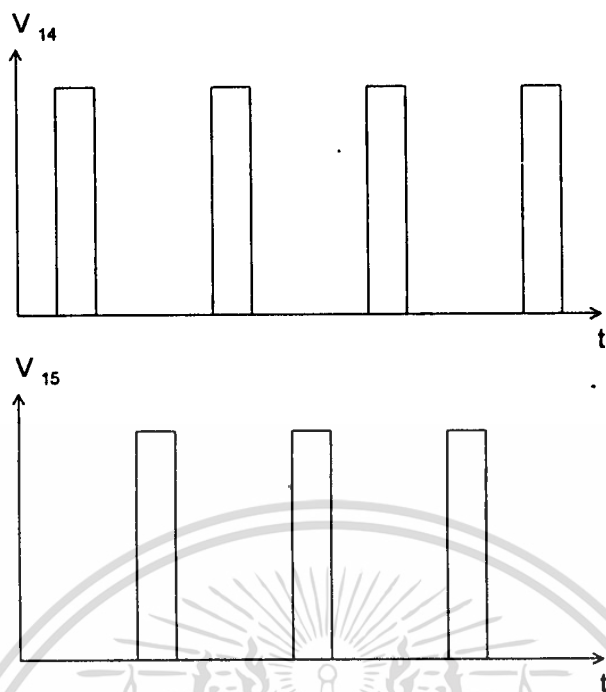
การทดลองวงจรควบคุมพลังงานไฟฟ้าด้วย TCA785

ทำการทดลองต่อวงจรควบคุมต่อพลังงานไฟฟ้าดังรูปที่ 4.14 โดยป้อนสัญญาณอินพุตเป็นแรงดันกระแสตรงค่าต่างๆ ที่ขา 11 ของ TCA785 จะได้สัญญาณเอาต์พุตที่ขา 14 และขา 15 ดังรูปที่ 4.15



รูปที่ 4.14 วงจรควบคุมพลังงานด้วย TCA785

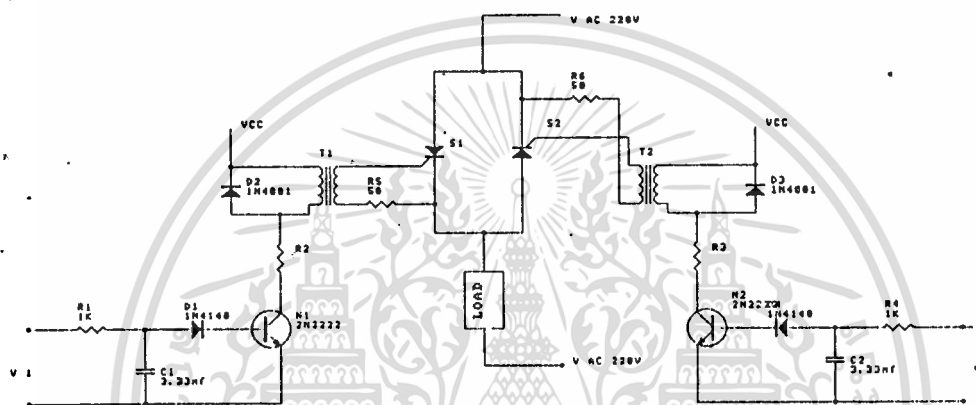
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



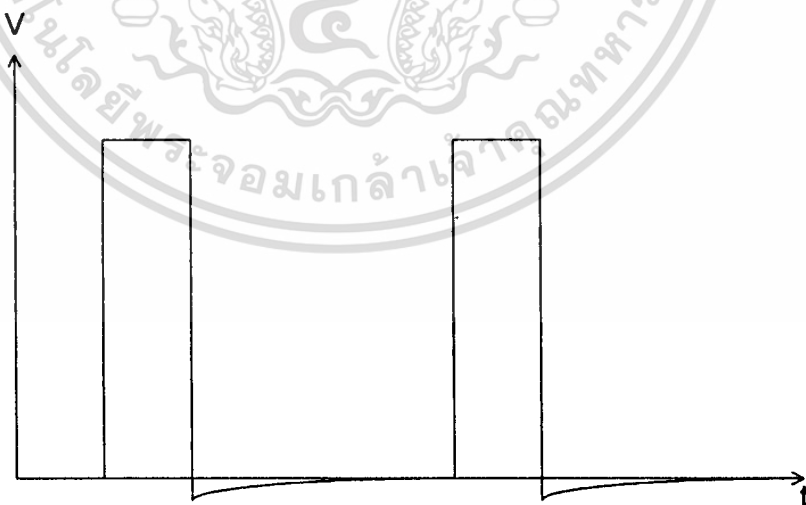
รูปที่ 4.15 สัญญาณเอาต์พุตจากวงจรควบคุมพลังงาน

การทดลองวงจรเชื่อมต่อกันด้วยพัลส์ทรานฟอเมอร์

เนื่องจากการควบคุมพลังงานไฟฟ้าโดยการนำพัลส์ไปทริกที่ไทรสเตอร์ (THYRISTER) อาจมีสัญญาณรบกวนเนื่องจากสัญญาณเอชี่ ดังนั้นต้องมีการแยกกราวด์ของส่วนควบคุมและส่วนที่จะไปทริกไทรสเตอร์ วิธีการเชื่อมต่อโดยการแยกกราวด์โดยใช้ พัลส์ทรานฟอเมอร์ จากการทดลองต่อวงจรดังรูปที่ 4.16 โดยอินพุตเป็นพัลส์ที่มาจาก TCA785 จะได้เอาต์พุตที่ พัลส์ทรานฟอเมอร์ ดังรูปที่ 4.17



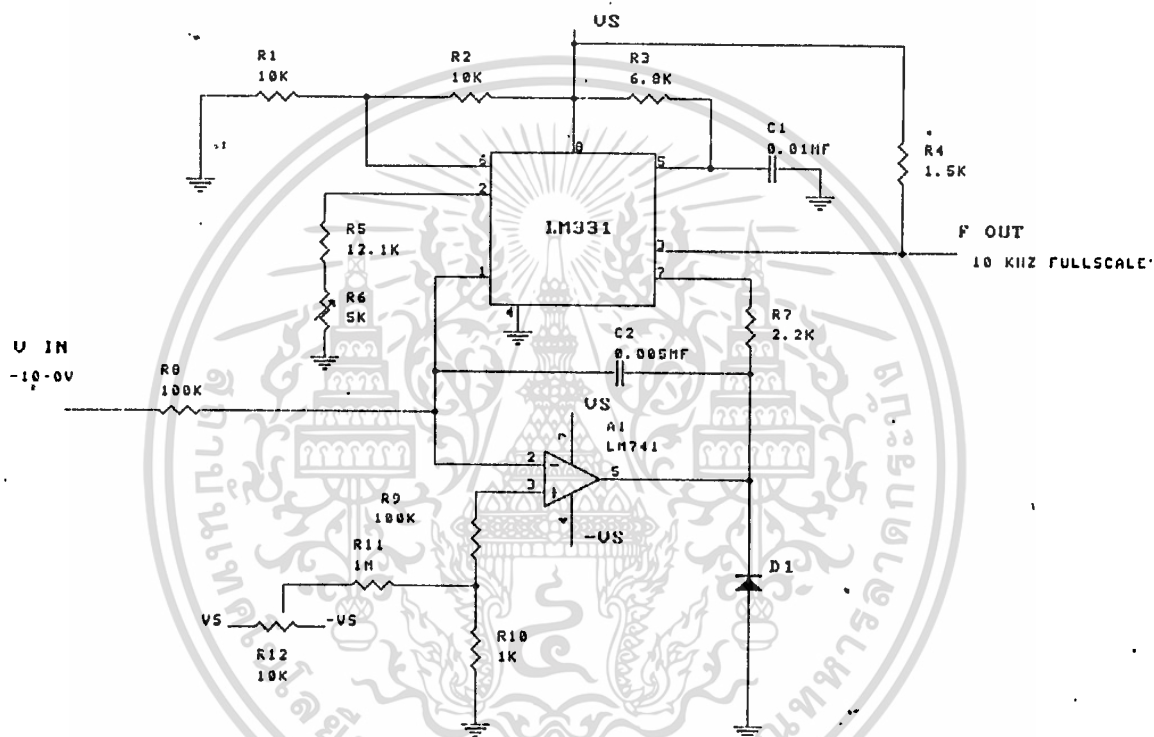
รูปที่ 4.16 วงจรเชื่อมต่อดัวยพัลส์ทรานฟอเมอร์



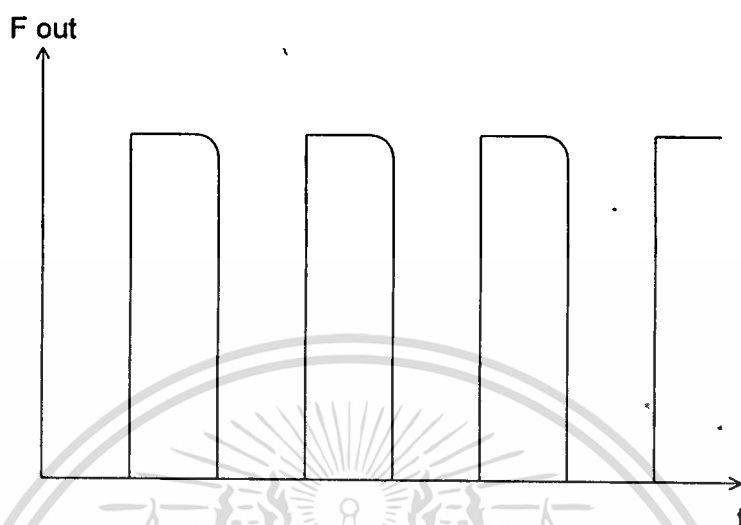
รูปที่ 4.17 สัญญาณเอาต์พุตที่พัลส์ทรานฟอเมอร์

การทดลองวงจรเปลี่ยนแรงดันเป็นความถี่

ทดลองต่อวงจรดังรูปที่ 4.18 โดยอินพุตเป็นแรงดันตั้งแต่ -10 ถึง 0 โวลต์ ซึ่งมาจากวงจร ฟรียูเอนซีเรคตีไฟร์ ซึ่งสัญญาณเอาต์พุตที่ขา 3 ของ LM331 เป็นสัญญาณสี่เหลี่ยม แสดงรูปที่ 4.19



รูปที่ 4.18 วงจรเปลี่ยนแรงดันเป็นความถี่



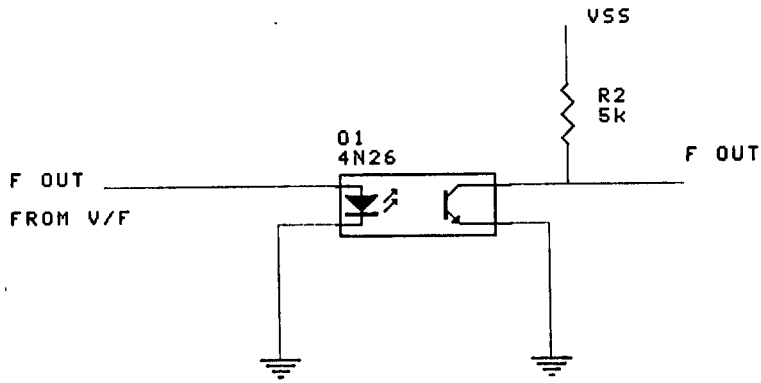
รูปที่ 4.19 สัญญาณเอาต์พุตวงจรเปลี่ยนแรงดันเป็นความถี่

การทดลองวงจรเชื่อมต่อด้วยแสง

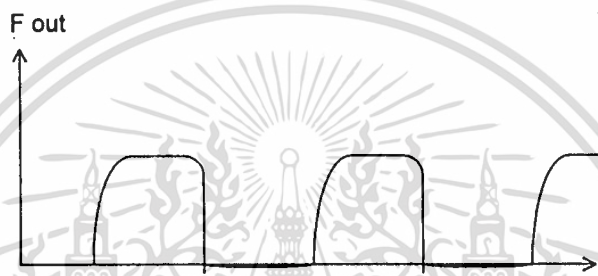
ในส่วนของตัวส่งและตัวรับความถี่ ถ้าต่อกราวน์ร่วมกันอาจเกิดสัญญาณรบกวนต่อกันได้ อันเป็นผลมาจากสายส่ง ดังนั้นจึงต้องมีการแยกกราวน์ออกจากกัน ก็คือวิธีการเชื่อมต่อโดยการแยกกราวน์โดยใช้อปโตไอโซเลเตอร์ (OPTO ISOLATOR)

การทดลองครั้งที่ 1

ทดลองต่อวงจรดังรูปที่ 4.20 โดยการป้อนอินพุตเป็นพัลส์สี่เหลี่ยมจากฟังก์ชันเจนเนอเรเตอร์ เข้าที่ขา 1 ของ 4N26 ได้สัญญาณเอาต์พุตที่ขา 5 ของ 4N26 ดังรูปที่ 4.21



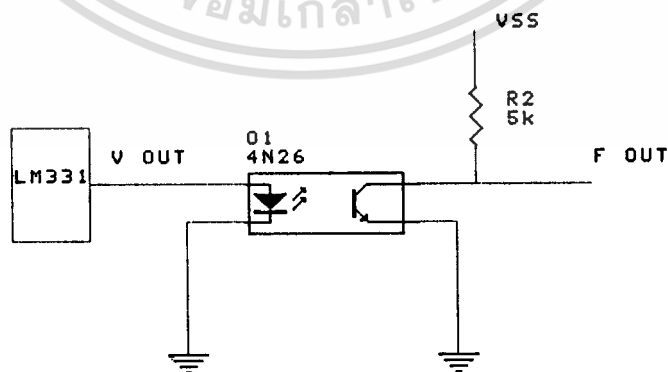
รูปที่ 4.20 วงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 1



รูปที่ 4.21 สัญญาณเอาต์พุตวงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 1

การทดลองครั้งที่ 2

นำเอาต์พุตที่มาจากวงจรเปลี่ยนโวลต์เตจเป็นความถี่ที่มาจากขา 3 ของ LM331 ไปเชื่อมต่อกับวงจร ออปโตไอโซเลเตอร์ ดังรูปที่ 4.22

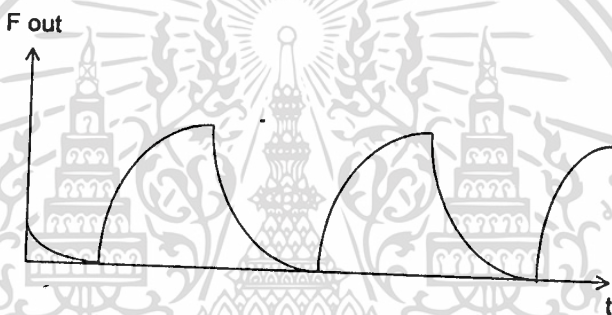


รูปที่ 4.22 วงจรเชื่อมต่อดัวยแสงในการทดลองครั้งที่ 2

จากนั้นป้อนแรงดันที่เป็นลบเข้าอินพุตของวงจรแล้ววัดสัญญาณเอาพุทที่ขา 5 ของ 4N26 ปรากฏว่าไม่มีสัญญาณความถี่ที่เอาท์พุทขา 5 ของ 4N26

การทดลองครั้งที่ 3

ทดลองปรับความต้านทานที่ขา 5 ของ 4N26 เป็น 1 K และทดลองปรับความต้านทานที่ขา 3 ของ LM331 ที่ต่ออยู่กับ V_{cc} โดยใช้ความต้านทานปรับค่าได้แล้ววัดสัญญาณเอาท์พุทที่ขา 5 ของ 4N26 ได้สัญญาณดังรูปที่ 4.23



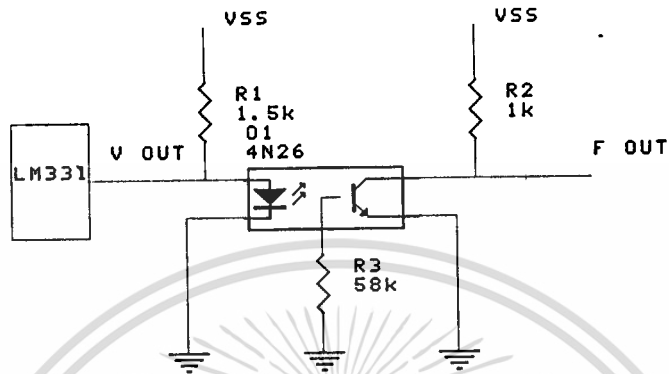
รูปที่ 4.23 สัญญาณเอาท์พุทวงจรเชื่อมต่อด้วยแสงในการทดลองครั้งที่ 3

จากการทดลองไม่ว่าจะปรับความต้านทานเท่าใดก็ไม่สามารถได้สัญญาณสี่เหลี่ยม

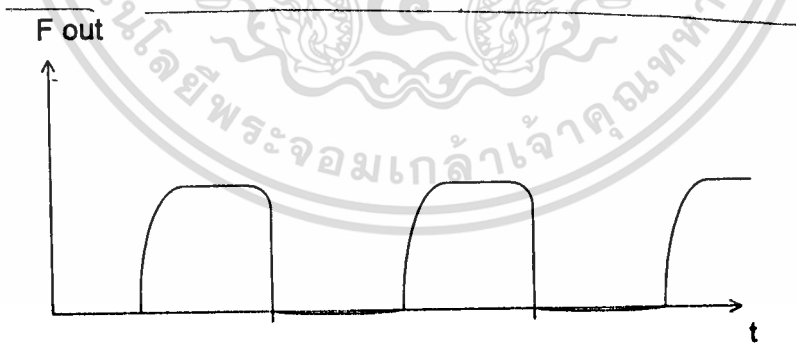
การทดลองครั้งที่ 4

ทดลองใส่ความต้านทานที่ขา 6 ของ 4N26 ต่อดึงกราวน์ โดยใช้ความต้านทานปรับค่าได้ปรับความต้านทานที่ขา 3 ของ LM331 , ขา 5 ของ 4N26 และ ที่ขา 6 ของ 4N26 แล้ววัดสัญญาณเอาท์พุทจนกว่าจะได้สัญญาณสี่เหลี่ยมที่ขา 5 ของ 4N26

จากการทดลองได้ความต้านทานดังรูปที่ 4.24 และสัญญาณสี่เหลี่ยมที่ได้ที่ขา 5 ของ 4N26 แสดงดังรูปที่ 4.25



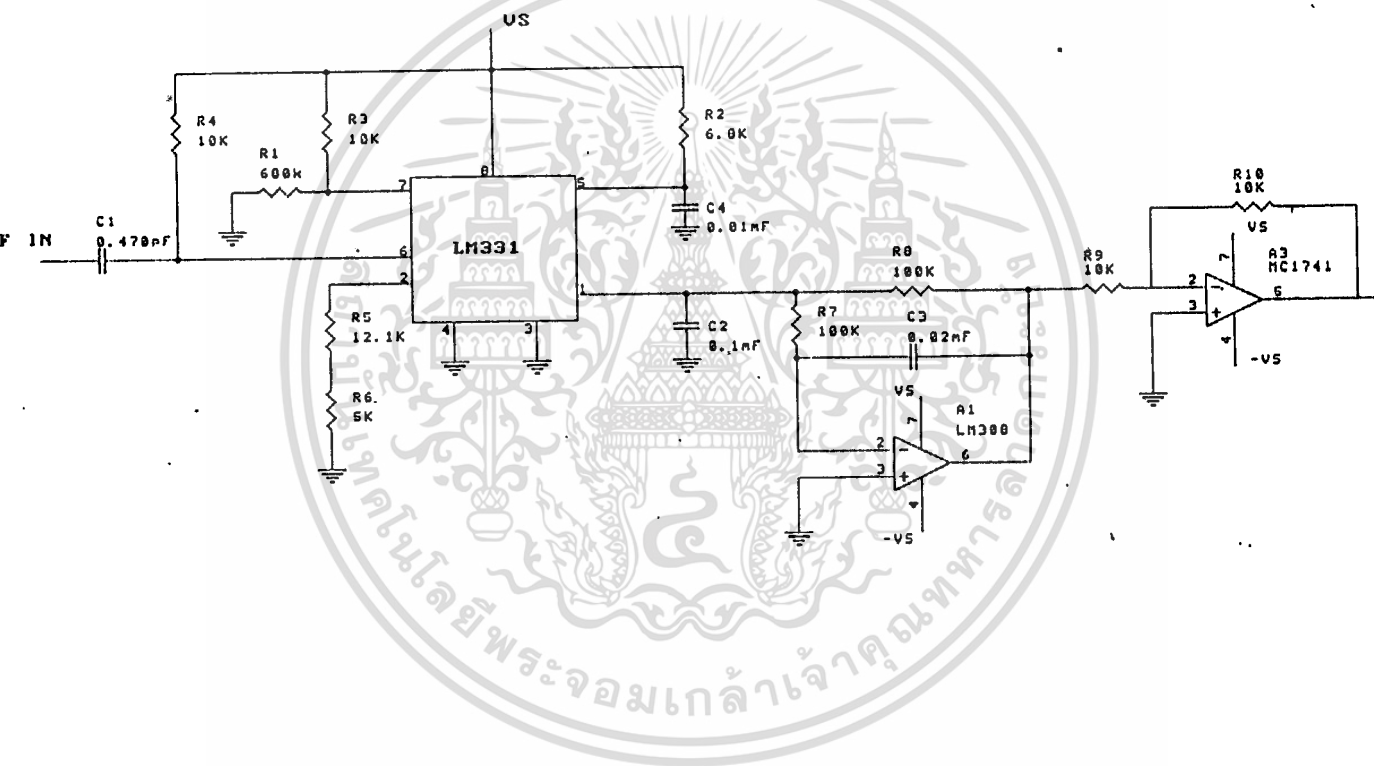
รูปที่ 4.24 วงจรเชื่อมต่อดำรงแสงในการทดลองครั้งที่ 4



รูปที่ 4.25 สัญญาณเอาต์พุตวงจรเชื่อมต่อดำรงแสงในการทดลองครั้งที่ 4

การทดลองวงจรเปลี่ยนความถี่เป็นแรงดัน

ทดลองต่อวงจรดังรูปที่ 4.26 โดยป้อนความถี่ที่เป็นสัญญาณสี่เหลี่ยมจาก ฟังก์ชันเจนเนอเรเตอร์ เข้าที่อินพุต แล้ววัดสัญญาณเอาต์พุต



รูปที่ 4.26 วงจรเปลี่ยนความถี่เป็นแรงดัน

จากการทดลอง โวลต์เตจที่ได้จะแปรค่าตามความถี่ที่เข้ามาโดยมีความถี่ที่ทำให้เอาต์พุตมีค่าสูงสุด 10 โวลต์ คือที่ 10 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองต่อวงจรรวม

วงจรรวมสามารถแบ่งออกเป็นส่วนใหญ่ๆได้สองส่วนคือส่วนที่แทนควบคุมและส่วนควบคุมอุปกรณ์

ส่วนที่แทนควบคุมประกอบด้วยวงจรต่างๆดังนี้

-บอร์ดคอนโทรลเลอร์ ANT-32

ใช้ในการประมวลผลและส่งสัญญาณควบคุมการทำงาน

-จอแสดงผล LCD

ใช้แสดงสถานะของระบบที่วัดได้และค่าที่ควบคุม

-วงจรเปลี่ยนสัญญาณดิจิตอลเป็นอนาล็อก

ใช้เปลี่ยนสัญญาณควบคุมจากไมโครคอนโทรลเลอร์ซึ่งเป็นดิจิตอลเป็นค่าแรงดันเพื่อส่งให้วงจรเปลี่ยนแรงดันเป็นความถี่ต่อไป

-วงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล

ใช้เปลี่ยนค่าแรงดันที่ได้จากวงจรเปลี่ยนความถี่เป็นแรงดันเป็นค่าทางดิจิตอลเพื่อนำไปประมวลผลต่อไปค่าที่ได้เป็นค่าวัดแรงดันและกระแสที่ส่งมาจากส่วนควบคุมอุปกรณ์

-วงจรเปลี่ยนแรงดันเป็นความถี่

ใช้ในการสื่อสารระหว่างส่วนควบคุมทั้งสองส่วนสัญญาณต่างๆที่ส่งระหว่างสองส่วนจะถูกเปลี่ยนเป็นความถี่ก่อนที่จะส่งถึงกันและกันในที่นี้จะใช้ส่งสัญญาณควบคุมแรงดันไปยังอีกส่วนของวงจร

-วงจรเปลี่ยนความถี่เป็นแรงดัน

เมื่อสัญญาณจากอีกส่วนหนึ่งถูกส่งมาแล้วส่วนที่รับจะใช้วงนี้ในการเปลี่ยนสัญญาณที่เป็นความถี่เป็นแรงดันเพื่อนำไปประมวลผลต่อไป สัญญาณที่ส่งมานี้คือสัญญาณข้อมูลแรงดันและกระแสที่วัดได้จากระบบ

-วงจรเชื่อมต่อดำวยแสง

ใช้ในการเชื่อมสัญญาณความถี่ของวงจรสองส่วนเพื่อแยกระบบกราวด์ของวงจรสองส่วนออกจากกัน

ส่วนควบคุมอุปกรณ์ประกอบด้วยวงจรดังต่อไปนี้

-วงจรเปลี่ยนความถี่เป็นแรงดัน

เพื่อใช้ในการเปลี่ยนสัญญาณควบคุมแรงดัน

-วงจรเปลี่ยนแรงดันความถี่

ใช้ในการส่งค่าแรงดันและกระแสที่วัดได้ของระบบ

-วงจรควบคุมพลังงานไฟฟ้าด้วย TCA785

นำค่าข้อมูลที่ได้จากวงจรเปลี่ยนสัญญาณความถี่เป็นแรงดันมาควบคุมการเปิดปิดของเอสซีอาร์ ผ่านวงจรเชื่อมต่อด้วยพัลลัททรานฟอเมอร์เพื่อนำไปควบคุมแรงดันต่อไป

-วงจรวัดกระแสและแรงดัน

วงจรที่ใช้วัดกระแสและแรงดันประกอบด้วยวงจรปริซึ้นเรกติไฟลเฮอร์ ที่ทำการจัดแรงดันไฟฟ้าที่ได้จากการแบ่งแรงดันโดยใช้หม้อแปลงลดแรงดัน (Stepdown Transformer) เพื่อวัดแรงดันไฟฟ้าของระบบและแรงดันไฟฟ้าที่ได้จากหม้อแปลงกระแส (Curent Transformer) เป็นกระแสตรงปรับค่าให้เหมาะสมแล้วส่งเข้าสู่วงจรแปลงเป็นความถี่ เพื่อส่งไปยังตัวควบคุมต่อไป

เมื่อนำวงจรทั้งหมดต่อเข้าด้วยกันทดลองส่งค่าจากไมโครคอนโทรลเลอร์ สามารถวัดแรงดันที่โหลดซึ่งเป็นหลอดไฟฟ้าชนิดไส้ขนาด 100 W ได้ดังตารางที่ 4.2

ตารางที่ 4.2 แสดงผลการทดลองส่งค่าควบคุมจากไมโครคอนโทรลเลอร์

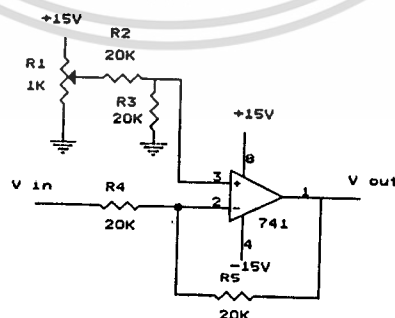
ค่าควบคุมส่งจากไมโครคอนโทรลเลอร์ (HEX)	ค่าแรงดันที่ได้จากวงจรเปลี่ยนความถี่เป็นแรงดันไฟฟ้า (Volt)	ค่าแรงดันควบคุมที่คร่อมโหลด (Volt)
FFH	0.2	หลอดกระพริบ
EBH	10	0
D7H	9.4	0.3
C3H	8.4	15
AFH	7	34
9BH	6.8	58
87H	6	88

ค่าควบคุมส่งจาก ไมโครคอนโทรเลอร์(HEX)	ค่าแรงดันที่ได้จากวงจรเปลี่ยน ความถี่เป็นแรงดันไฟฟ้า (Volt)	ค่าแรงดันควบคุมที่ คร่อมโหลด (Volt)
73H	5.2	115
5FH	4.5	140
4BH	3.8	168
37H	3.0	190
23H	2.0	205
0FH	1.2	216
05H	0.6	218
00H	0.2	หลอดกระพริบ

จากตารางพบว่าค่าควบคุมที่ส่งออกจากไมโครคอนโทรเลอร์มีค่าเพิ่มขึ้นคงที่ และค่าแรงดันที่ได้จากวงจรเปลี่ยนความถี่เป็นแรงดันก็มีการเพิ่มขึ้นในลักษณะใกล้เคียงกันแต่ค่าแรงดันควบคุมที่คร่อมโหลดมีการเปลี่ยนแปลงลักษณะไม่คงที่

นอกจากนี้เมื่อส่งค่า 00H และ FFH ออกไปแรงดันไฟฟ้าที่คร่อมโหลดมีการเปลี่ยนแปลงตลอดเวลาไม่คงที่ทำให้หลอดไฟกระพริบเนื่องจากช่วงที่เปิดเอสซีอาร์เป็นช่วงที่อยู่ระหว่างรูปคลื่นกำลังเปลี่ยนแปลงพอดี ทำให้บางครั้งเปิดเอสซีอาร์ไม่ได้บางครั้งก็ได้

เราแก้ปัญหานี้โดยการเพิ่มวงจรซัมมิงแอมป์ (Summing Amp) เพื่อช่วยในการปรับแต่งสัญญาณควบคุมก่อนเข้าสู่วงจรควบคุมพลังงานให้เหมาะสมยิ่งขึ้นซึ่งมีวงจรดังรูป 4.27



รูปที่ 4.27 แสดงวงจรซัมมิงแอมป์ (Summing Amp)

หลังจากที่เพิ่มวงจรเข้าไปแล้วสามารถแก้ไขปัญหาดังกล่าวได้และค่าที่ส่งไปให้
ผลในลักษณะกลับกันผลการทดลองสามารถแสดงได้ดังตารางที่ 4.3

ตารางที่ 4.3 แสดงผลการทดลองส่งค่าควบคุมจากไมโครคอนโทรลเลอร์เมื่อมี
วงจรขั้วมิงแอมป์ (แรงดันของระบบ 218 โวลท์)

ค่าควบคุมส่งจาก ไมโครคอนโทรลเลอร์(HEX)	ค่าแรงดันที่ได้จากวงจรเปลี่ยน ความถี่เป็นแรงดันไฟฟ้า (Volt)	ค่าแรงดันควบคุมที่ คร่อมโหลด (Volt)
00H	11.5	0
14H	10.5	0
28H	9.6	1
3CH	9.0	11
50H	8.0	30
64H	7.0	54
78H	6.2	88
8CH	5.5	113
A0H	4.8	142
B4H	3.7	168
C8H	2.7	190
DCH	1.7	208
F0H	0.8	216
FAH	0.6	218
FFH	0.4	218

จากผลการทดลองเห็นได้ว่าค่าที่ส่งออกไปไม่เป็นอัตราส่วนที่คงที่กับแรงดันควบคุม การควบคุมจึงต้องใช้วิธีการส่งค่าควบคุมจากตารางที่สร้างขึ้นเพื่อให้ได้ค่าแรงดันควบคุมตามต้องการ แต่หากว่าแรงดันไฟฟ้าของระบบเปลี่ยนแปลงไปก็ต้องเปลี่ยนค่าในตารางด้วย

ส่วนการวัดค่ากลับมาของวงจรวัดกระแสและแรงดันนั้นได้ผลการทดลองดัง
ตารางที่ 4.4 และ 4.5

ตารางที่ 4.4 แสดงผลการทดลองการวัดแรงดัน

ค่าจริงที่วัดได้จากระบบ โดยใช้ดิจิตอล มัลติมิเตอร์ (Volt)	ค่าที่ได้จากวงจรเปลี่ยนสัญญาณ อนาล็อกเป็นดิจิตอล (0-255 Decimal)
160	168-170
170	170-174
180	176-178
190	182-183
200	184-186
210	187-188
220	188-189

ตารางที่ 4.5 แสดงผลการทดลองการวัดกระแส

ค่าจริงที่วัดได้จากระบบ โดยใช้ดิจิตอล มัลติมิเตอร์ (Amp) (Volt)	ค่าที่ได้จากวงจรเปลี่ยนสัญญาณ อนาล็อกเป็นดิจิตอล (0-255 Decimal)
0.029 (160)	45-49
0.031 (170)	46-52
0.033 (180)	46-50
0.035 (190)	46-50
0.037 (200)	46-50
0.039 (210)	42-50
0.041 (220)	42-50

จะเห็นได้ว่าค่าที่ได้ออกมาจากการวัดแรงดันของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลนั้นมีการเปลี่ยนแปลงค่อนข้างที่จะไม่เป็นเส้นตรงและยังพบอีกว่าการทำงานแต่ละครั้งของวงจรให้ค่าออกมาไม่เท่ากันเมื่อวัดแรงดันที่เท่ากัน จะต้องมีการปรับแต่งทุกครั้งที่เปิดเครื่องใหม่ บางครั้งก็ไม่สามารถปรับแต่งได้

ส่วนจากวงจรวัดกระแสที่ได้ออกมาจากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลก็ไม่ค่อยเปลี่ยนแปลงตามกระแสที่วัดมากนักและยังประสบปัญหาเหมือนการวัดแรงดันเช่นกันคือผลที่ออกมาแต่ละครั้งไม่เท่ากันและบางครั้งก็ปรับแต่งไม่ค่อยได้



บทที่ 5

บทวิจารณ์และสรุป

จากผลการทดลองพบว่าทุกวงจรสามารถทำงานได้จริงแตเมื่อนำมารวมกันแล้วพบว่าบางส่วนยังไม่สามารถทำงานได้อาจเป็นเนื่องจากการประกอบวงจรที่ผิดพลาดนั่นเองซึ่งสามารถตรวจสอบและแก้ไขได้

ในส่วนของโปรแกรมควบคุมนั้นยังไม่ได้เน้นไปทางระบบป้อนกลับ แต่จะเน้นไปทางการควบคุมตามที่ใช้ต้องการ แนวทางในการพัฒนาคือสร้างระบบควบคุมแบบป้อนกลับที่สมบูรณ์และวงจรที่มีความเสถียรสูงที่สามารถนำไปใช้งานจริงนอกจากนี้ยังพัฒนาไปเป็นการควบคุมไฟฟ้าสามเฟสได้อีกด้วย

ส่วนการควบคุมพลังงานให้เป็นไปได้ดังโปรแกรมนั้นยังไม่สามารถทำได้อย่างถูกต้องมากนักเนื่องจากการเปลี่ยนมุมทริกเอสซีอาร์นั้นถึงแม้ว่าสามารถควบคุมให้เปลี่ยนแปลงแบบเส้นตรงได้แต่พลังงานที่ออกมานั้นไม่ได้มีการเปลี่ยนแปลงเป็นแบบเส้นตรงตามสัญญาณที่ป้อนให้ ทำให้การเขียนโปรแกรมควบคุมพลังงานให้ได้ดังต้องการต้องใช้วิธีอ่านค่าควบคุมจากตารางซึ่งสร้างขึ้นแล้วส่งออกไปซึ่งเป็นเรื่องที่ค่อนข้างยุ่งยากและผิดพลาดได้ง่าย ค่าที่ส่งออกไปยังขึ้นกับระดับแรงดันไฟฟ้าที่จ่ายให้กับระบบทั้งหมดอีกด้วย เนื่องจากการเปิดปิดเอสซีอาร์ จะไปควบคุมไฟฟ้าจากจุดนี้ เมื่อแรงดันไฟฟ้าของระบบเปลี่ยนแปลงไปค่าที่ต้องส่งก็ต้องเปลี่ยนแปลงด้วยเพื่อให้ได้ค่าแรงดันควบคุมที่ถูกต้อง ซึ่งหากใช้วิธีการเปิดตารางแล้วจะยุ่งยากมาก แนวทางการแก้ไขอาจใช้วิธีหาความสัมพันธ์ระหว่างแรงดันของระบบ, แรงดันควบคุมและค่าควบคุมที่ส่งจากไมโครคอนโทรลเลอร์แล้วนำมาคำนวณค่าควบคุมที่จะส่งออกไป

สำหรับการวัดค่ากลับจะเห็นได้ว่าค่าที่วัดได้จะไม่คงที่เนื่องมาจากความผิดพลาดของวงจรในส่วนวัดค่ากลับให้ค่าออกมาไม่คงที่ในขณะที่ค่าของระบบไม่ว่าจะเป็นแรงดันและกระแสไฟฟ้าคงที่ บางครั้งค่าที่ได้จะลดลงเรื่อยๆจนคงที่ ซึ่งค่าที่คงที่นี้จะไม่เท่ากันในแต่ละครั้งทั้งวงจรวัดแรงดันและวัดกระแส โดยเฉพาะวงจรวัดกระแสมีการเปลี่ยนแปลงที่น้อยมากจนไม่สามารถนำค่ามาวิเคราะห์ได้

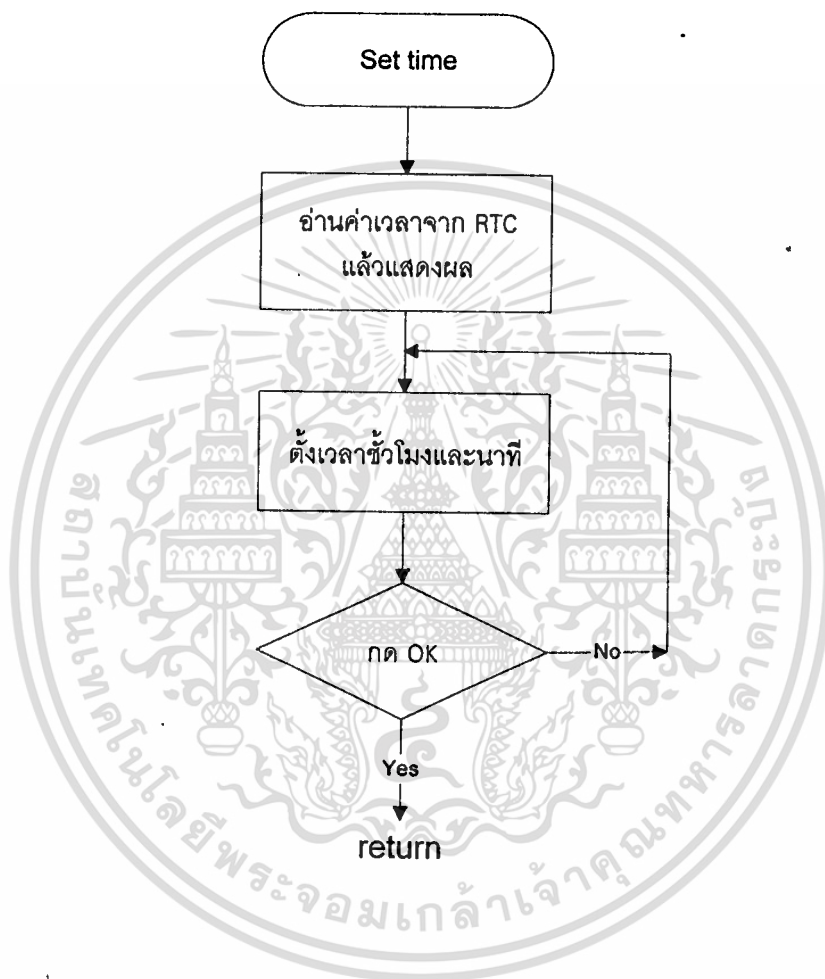
ความผิดพลาดของวงจรวัดค่ากลับนี้อาจเป็นเพราะความผิดพลาดของอุปกรณ์ต่างๆของวงจรเช่น ตัวต้านทานซึ่งใช้แบบความผิดพลาด 5 % และออปแอมป์คุณภาพต่ำ ความผิดพลาดของวงจรแสดงจากการเพิ่มหรือลดอัตราขยายของวงจรตลอดเวลา ทำให้ค่าที่ส่งเข้าวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลไม่คงที่แม้ค่าในระบบจะคงที่ก็ตาม ค่าที่นำไปคำนวณด้วยไมโครคอนโทรลเลอร์ก็จะไม่คงที่และไม่สามารถบอกได้ว่าค่าที่วัดมาเป็นค่าเท่าใด

ปัญหานี้อาจแก้ไขได้โดยการใช้อุปกรณ์ต่างที่มีคุณภาพดีมีเปอร์เซ็นต์ความผิดพลาดต่ำหรือออกแบบวิธีการวัดค่าแรงดันไฟฟ้าและกระแสใหม่

จากข้อผิดพลาดที่กล่าวมาทำให้โครงงานนี้ไม่สามารถทำงานได้สมบูรณ์ในการวัดค่ากลับมาจากระบบที่ควบคุม แต่ก็โครงงานนี้ก็สามารถทำงานในลักษณะส่งค่าออกไปควบคุมเพียงอย่างเดียวทั้งการควบคุมด้วยคนหรือการควบคุมโดยโปรแกรมอัตโนมัติ ส่วนการบันทึกข้อมูลนั้นสามารถทำได้แต่ได้ค่าที่ไม่ถูกต้องตามที่วงจรวัดค่าจากระบบวัดได้ ซึ่งหากวงจรส่วนนี้สามารถทำงานได้ก็จะทำให้โครงงานนี้เสร็จสมบูรณ์

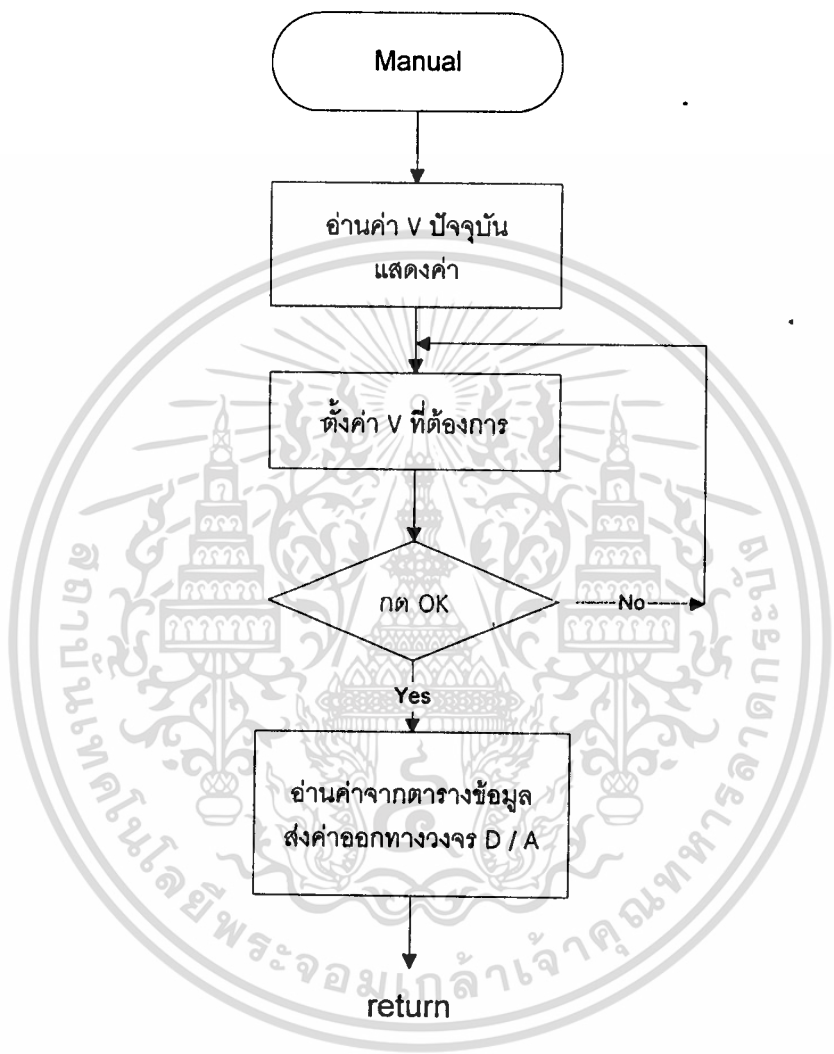
ภาพผนวก

โฟลชาร์ตแสดงการทำงานของโปรแกรมในส่วนโปรแกรมย่อย
ส่วน SET TIME

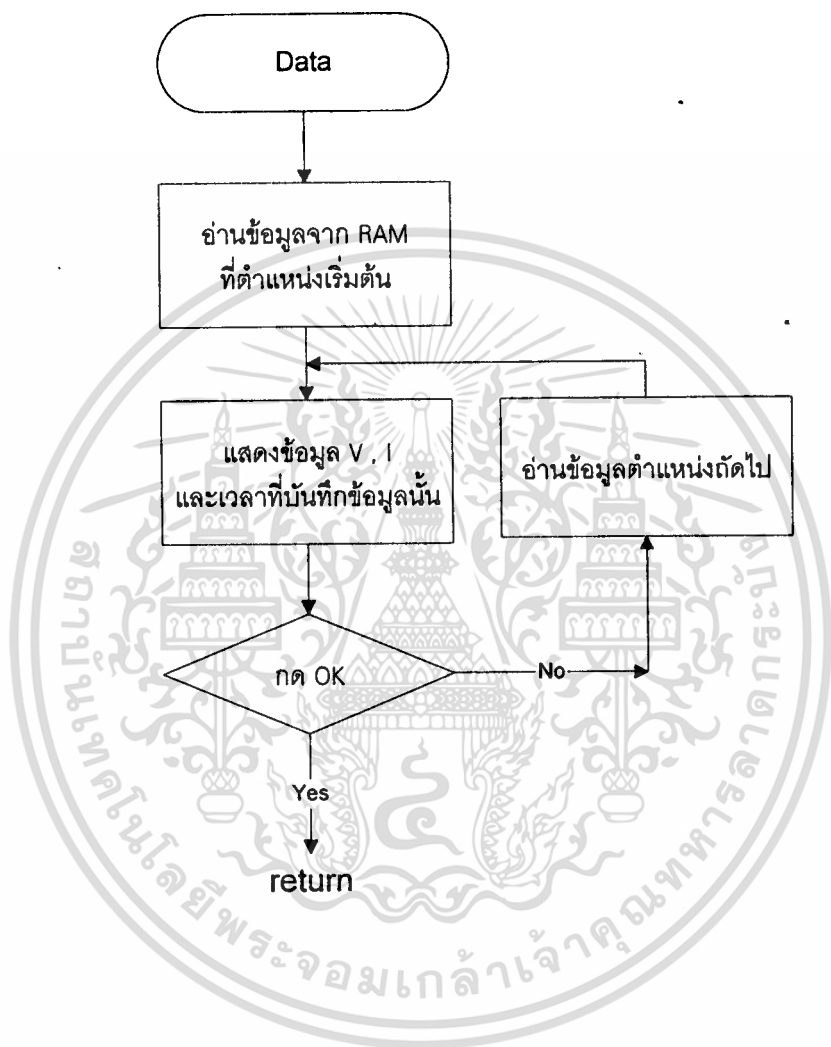


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

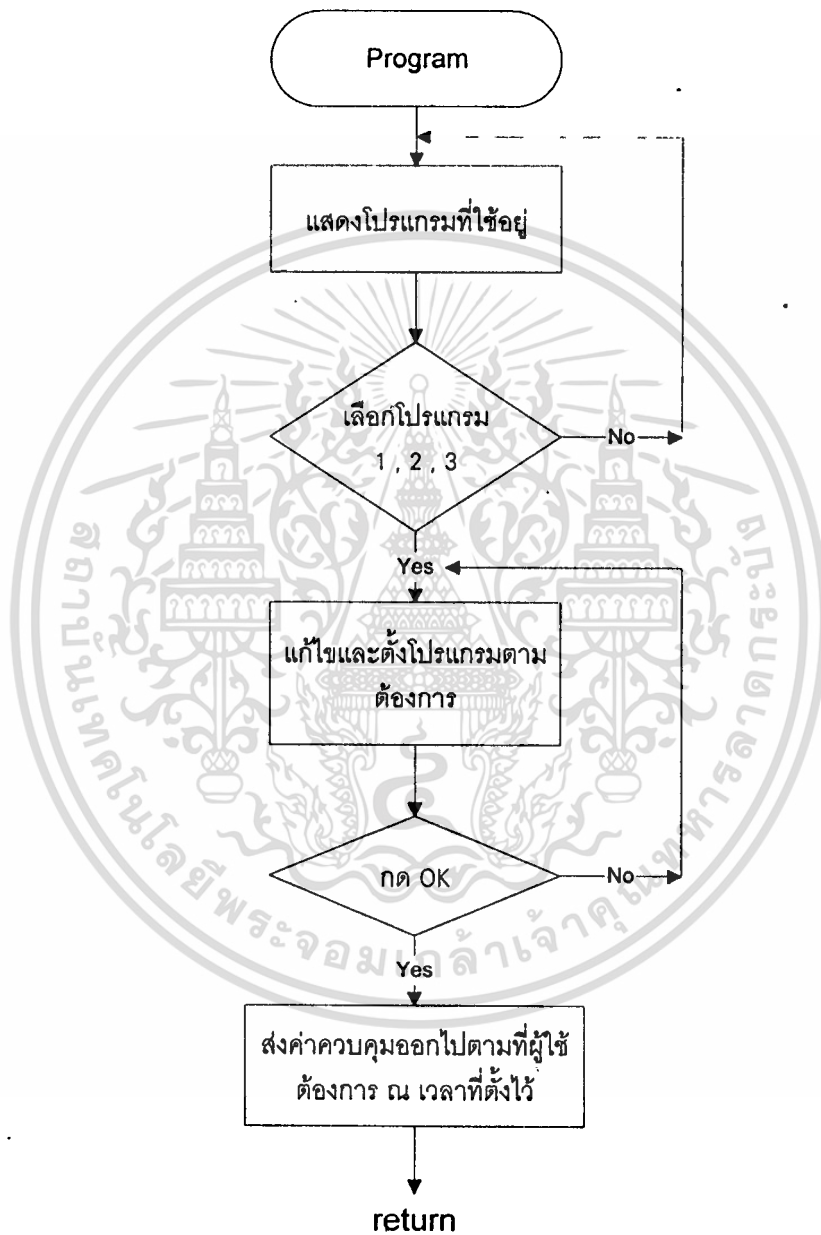
ส่วน MANUAL



ส่วน DATA



ส่วน PROGRAM



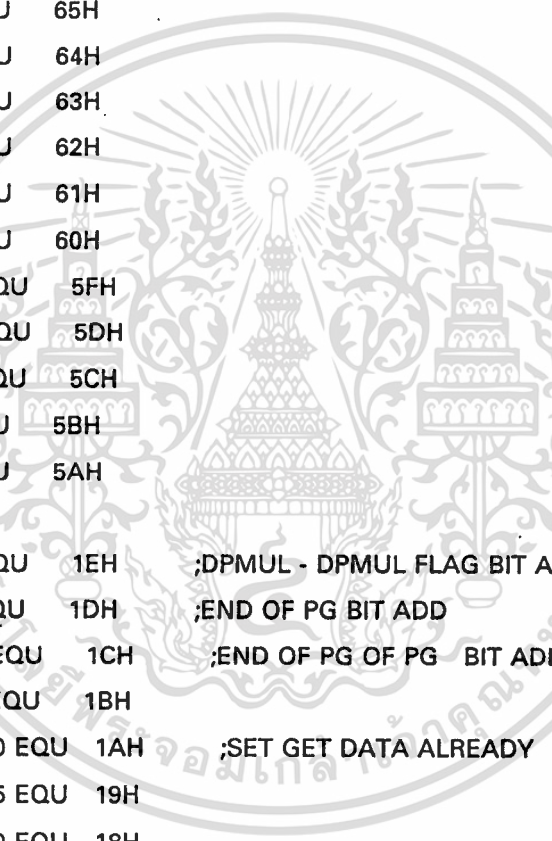
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงาน

;PROJECT: POWER CONTROL
;BY : GERAUT VARIN 34102065
; AKATAT CHAWANAPONG 34109494
; CONTROL ENGINEERING KMIT'L
;DATE : 14/04/95

ORG	EQU	0000H
DP1A	EQU	0F800H
DP1B	EQU	0F801H
DP1C	EQU	0F802H
DP1CON	EQU	0F803H
DP2A	EQU	0FC00H
DP2B	EQU	0FC01H
DP2C	EQU	0FC02H
DP2CON	EQU	0FC03H
STAPG1	EQU	8001H
STAPG2	EQU	80C4H
STAPG3	EQU	8187H
STADATA	EQU	82FFH
ENDDH	EQU	82FEH
ENDDL	EQU	82FDH
TH	EQU	7FH
TM	EQU	7EH
V1	EQU	7DH
V2	EQU	7CH
V3	EQU	7BH
PO1	EQU	7AH
PO2	EQU	79H
PO3	EQU	78H
I1	EQU	7H
I2	EQU	76H
I3	EQU	75H
VS1	EQU	74H
VS2	EQU	73H
VS3	EQU	72H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

D1      EQU    71H
D2      EQU    70H
D3      EQU    6FH
DV1     EQU    6CH
DV2     EQU    6BH
DI1     EQU    6AH
DI2     EQU    69H
TDH     EQU    68H
TDM     EQU    67H
HEXH    EQU    66H
HEXL    EQU    65H
BCD1    EQU    64H
BCD2    EQU    63H
BCD3    EQU    62H
BCD4    EQU    61H
BCD5    EQU    60H
PGRUN   EQU    5FH
BUFDPH  EQU    5DH
BUFDPL  EQU    5CH
STA     EQU    5BH
TBUF    EQU    5AH

MULFG   EQU    1EH      ;DPMUL - DPMUL FLAG BIT ADD
PGEND   EQU    1DH      ;END OF PG BIT ADD
PGENDPG EQU    1CH      ;END OF PG OF PG BIT ADD
TOPDATA EQU    1BH
DATAGET00 EQU 1AH      ;SET GET DATA ALREADY
DATAGET15 EQU    19H
DATAGET30 EQU    18H
DATAGET45 EQU    17H
RUNPG   EQU    16H      ;SET RUN PG

HIGHV1  EQU    02H      ;MAX V
HIGHV2  EQU    05H
HIGHV3  EQU    06H
LOWV1   EQU    02H      ;LOW V
LOWV2   EQU    01H

```

```

LOWV3 EQU 00H
DIVVH EQU 00H ; DIV V BY 1 SEND DATA CONTROL
DIVVL EQU 01H
MULVH EQU 00H ;MUL V BY 1 RECIVE VOLT
MULVL EQU 01H
MULIH EQU 00H ;MUL I BY 1 RECIVE CURRENT
MULIL EQU 01H

```

```

CLR MULFG
CLR PGEND
CLR PGENDPG
CLR TOPDATA
CLR DATAGET00
CLR DATAGET15
CLR DATAGET30
CLR DATAGET45
CLR RUNPG
MOV TH,#00H
MOV TM,#00H
MOV V1,#02H
MOV V2,#02H
MOV V3,#00H
MOV I1,#00H
MOV I2,#00H
MOV I3,#05H
MOV PO1,#00H
MOV PO2,#00H
MOV PO3,#00H
MOV VS1,#02H
MOV VS2,#02H
MOV VS3,#00H
MOV HEXH,#00H
MOV HEXL,#00H
MOV PGRUN,#31H
MOV BUFDPH,#00H

```

MOV BUFDPL,#00H

MOV TBUF,#0FFH

;CONTROL 8255

MOV DPTR,#DP1CON

MOV A,#89H ;CONTROLWORD

MOVX @DPTR,A

MOV R6,#0A0H

LCALL DELAY1MSEC

MOV DPTR,#DP2CON

MOV A,#90H ;CONTROLWORD

MOVX @DPTR,A

MOV R6,#0A0H

LCALL DELAY1MSEC

;SET INIT

MOV DPTR,#ENDDH

MOV A,#82H

MOVX @DPTR,A

MOV DPTR,#ENDDL

MOV A,#0FFH

MOVX @DPTR,A

MOV DPTR,#8000H ;SETSTART PG1

MOV A,#0FFH

MOVX @DPTR,A

MOV DPTR,#80C3H ;SETEND PG1

MOV A,#0FFH

MOVX @DPTR,A

MOV DPTR,#8186H ;SETEND PG2

MOV A,#0FFH

MOVX @DPTR,A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
MOV    DPTR,#8249H      ;SETEND PG3
MOV     A,#0FFH
MOVX    @DPTR,A
```

```
MOV     DPTR,#STAPG1
MOV     A,#33H
MOVX    @DPTR,A
```

```
INC     DPTR
```

```
MOV     A,#12H
MOVX    @DPTR,A
```

```
INC     DPTR
```

```
MOV     A,#00H
MOVX    @DPTR,A
```

```
INC     DPTR
```

```
MOV     A,#0DCH
MOVX    @DPTR,A
```

```
INC     DPTR
```

```
MOV     A,#35H
MOVX    @DPTR,A
```

```
INC     DPTR
```

```
MOV     A,#13H
MOVX    @DPTR,A
```

```
INC     DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    A,#00H
MOVX   @DPTR,A
```

```
INC    DPTR
```

```
MOV    A,#0D2H
MOVX   @DPTR,A
```

```
INC    DPTR
```

```
MOV    A,#12H
MOVX   @DPTR,A
```

```
INC    DPTR
```

```
MOV    A,#23H
MOVX   @DPTR,A
```

```
INC    DPTR
```

```
MOV    A,#00H
MOVX   @DPTR,A
```

```
INC    DPTR
```

```
MOV    A,#0FAH
MOVX   @DPTR,A
```

```
INC    DPTR
```

```
MOV    A,#0FFH
MOVX   @DPTR,A
```

```
MOV    DPTR,#STAPG2
MOV    A,#12H
MOVX   @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INC DPTR

MOV A,#14H

MOVX @DPTR,A

INC DPTR

MOV A,#00H

MOVX @DPTR,A

INC DPTR

MOV A,#0DCH

MOVX @DPTR,A

INC DPTR

MOV A,#0FFH

MOVX @DPTR,A

MOV DPTR,#STAPG3

MOV A,#00H

MOVX @DPTR,A

INC DPTR

MOV A,#15H

MOVX @DPTR,A

INC DPTR

MOV A,#00H

MOVX @DPTR,A

INC DPTR

MOV A,#0D0H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOVX @DPTR,A

INC DPTR

MOV A,#0FFH

MOVX @DPTR,A

SETLCD:

MOV A,#00111000B ;FUCNTIONSET

LCALL LCDWI

MOV A,#00001100B ;DISPLAY ON

LCALL LCDWI

MOV A,#00000001B ;CLEAR

LCALL LCDWI

MOV R6,#0AH

LCALL DELAY1MSEC

SETTI:

CLR P1.6 ;RST=0

SETB P1.5 ;SCLK=1

SETCHARMENU:

MOV A,#01000000B

LCALL LCDWI

MOV A,#00H ;CHAR 1 LEFT AROW

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

MOV A,#08H

LCALL LCDWD

MOV A,#1FH

LCALL LCDWD

MOV A,#08H

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

```

MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD

MOV    A,#00H ;CHAR 2 TAIL OF LEFT AROW
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#1CH
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H ;CHAR 3 TAIL OF RIGHT AROW
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#07H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD
MOV    A,#00H
LCALL  LCDWD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCALL LCDWD

MOV A,#00H ;CHAR 4 RIGHT AROW

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

MOV A,#02H

LCALL LCDWD

MOV A,#1FH

LCALL LCDWD

MOV A,#02H

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

MOV A,#00H

LCALL LCDWD

MOV A,#00H

LCALL LCDWD

MOV A,#04H ;CHAR 5 UP AROW

LCALL LCDWD

MOV A,#0EH

LCALL LCDWD

MOV A,#15H

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

MOV A,#04H

LCALL LCDWD

MOV A,#04H ;CHAR 6 DOWN AROW


```

LCALL LCDWD
MOV A,#04H
LCALL LCDWD
MOV A,#04H
LCALL LCDWD
MOV A,#04H
LCALL LCDWD
MOV A,#04H
LCALL LCDWD
MOV A,#15H
LCALL LCDWD
MOV A,#0EH
LCALL LCDWD
MOV A,#04H
LCALL LCDWD

LJMP MAIN

```

*****LCD WRITE INSTRUCTION (A,DPTR)*****

```

LCDWI:  MOV DPTR,#DP1A
        MOVX @DPTR,A
        MOV DPTR,#DP1B
        MOVX A,@DPTR
        CLR ACC.0 ;RS=0
        CLR ACC.1 ;R/W=0
        CLR ACC.2 ;ENABLE=0
        MOVX @DPTR,A
        SETB ACC.2 ;ENABLE=1
        MOVX @DPTR,A
        CLR ACC.2 ;ENABLE=0
        MOVX @DPTR,A
        MOV R6,#0AH
        LCALL DELAY1MSEC
        RET

```

*****LCD WRITE DATA (A,DPTR)*****

```

LCDWD:

```

```

MOV    DPTR,#DP1A
MOVX   @DPTR,A
MOV    DPTR,#DP1B
MOVX   A,@DPTR
SETB   ACC.0 ;RS=1
CLR    ACC.1 ;R/W=0
CLR    ACC.2 ;ENABLE=0
MOVX   @DPTR,A
SETB   ACC.2 ;ENABLE=1
MOVX   @DPTR,A
CLR    ACC.2 ;ENABLE=0
MOVX   @DPTR,A
MOV    R6,#0AH
LCALL  DELAY1MSEC
RET

```

***** TIME BYTE WRITE (R6 COMMAND,R7 DATA) *****

; P1.4 = I/O DATA

; P1.6 = RST\

; P1.5 = SCLK

```

TBYTEWR:  CLR    P1.4 ;COOMAND WRITE
          LCALL  DELAY
          SETB   P1.6 ;RST=1
          LCALL  DELAY
          MOV    B,#8D ;SEND COMMAND
          CLR    C

```

```

TBYTEWR1: MOV    A,R6
          RRC    A
          MOV    R6,A
          MOV    P1.4,C

          LCALL  SCLKRW
          DJNZ   B,TBYTEWR1

```

```

MOV    B,#8D ;SEND DATA

```

```

CLR    C
TBYTEWR2:  MOV    A,R7
           RRC    A
           MOV    R7,A
           MOV    P1.4,C
           LCALL  SCLKRW
           DJNZ   B,TBYTEWR2
           CLR    P1.6 ;RST=0
           LCALL  DELAY
           RET

;***** TIME BYTE READ (R6 COMMAND,R7 DATA) *****

TBYTERD:   CLR    P1.4 ;COMAND READ
           LCALL  DELAY
           SETB   P1.6 ;RST=1
           LCALL  DELAY
           MOV    B,#8D ;SEND COMMAND
           CLR    C
TBYTERD1:  MOV    A,R6
           RRC    A
           MOV    R6,A
           MOV    P1.4,C
           LCALL  SCLKCOM
           DJNZ   B,TBYTERD1

           MOV    B,#8D ;RECIVE DATA
           MOV    R7,#0D

TBYTERD2:  LCALL  SCLKRW
           MOV    A,R7
           MOV    C,P1.4 ;READ SERIAL DATA
           RRC    A
           MOV    R7,A
           DJNZ   B,TBYTERD2
           CLR    P1.6 ;RST=0
           LCALL  DELAY
           RET

;***** SCLKCOM F>R *****

```

```

SCLKCOM:   CLR   P1.5
           LCALL DELAY
           SETB  P1.5
           LCALL DELAY
           RET

;***** SCLKRW R>F *****
SCLKRW:    SETB  P1.5
           LCALL DELAY
           CLR   P1.5
           LCALL DELAY
           RET

;***** DELAY FOR RW DATA *****
DELAY:     MOV   R1,#5D
           DJNZ  R1,$
           RET

;***** TIME DISPLAY (R5,R4 ADD,TH,TM)*****
TDISP:     MOV   R6,#85H ;READ HOUR
           LCALL TBYTERD
           MOV   TH,R7
           LCALL DISHOUR

           MOV   A,#3AH
           LCALL LCDWD   ;DISPLAY COLON

           INC   R4
           INC   R4
           INC   R4
           MOV   R6,#83H ;READ MIN
           LCALL TBYTERD
           MOV   TM,R7
           LCALL DISMIN
           RET

;*****DISHOUR TH,R4 ADD,R7,R5 *****
DISHOUR:    MOV   R7,TH
           MOV   A,R7 ;R7=00(10 HR)(HR)
           ANL   A,#00111111B ;CLEAR BIT1

```

```

SWAP  A  ;
MOV   R1,#05 ;SWAP R5
MOV   R5,#00110000B
XCHD  A,@R1 ;R5= NUMBER DISPLAY
MOV   A,R4
LCALL LCDWI
MOV   A,R5
LCALL LCDWD ;DISPLAY HOUR

```

```

MOV   R7,TH
MOV   A,R7 ;R7=00(10 HR)(HR)
ANL   A,#00111111B ;CLEAR BIT1
MOV   R1,#05 ;SWAP R5
MOV   R5,#00110000B
XCHD  A,@R1 ;R5= NUMBER DISPLAY
MOV   A,R5
LCALL LCDWD ;DISPLAY HOUR

RET

```

***** DISMIN R4 ADD,TM,R7,R5*****

```

DISMIN:  MOV   R7,TM
MOV   A,R7 ;R7=0(10 MIN)(MIN)
ANL   A,#01111111B ;CLEAR BIT1
SWAP  A  ;
MOV   R1,#05 ;SWAP R5
MOV   R5,#00110000B
XCHD  A,@R1 ;R5= NUMBER DISPLAY
MOV   A,R4
LCALL LCDWI
MOV   A,R5
LCALL LCDWD ;DISPLAY MIN

MOV   R7,TM
MOV   A,R7 ;R7=0(10 MIN)(MIN)
ANL   A,#01111111B ;CLEAR BIT1
MOV   R1,#05 ;SWAP R5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    R5,#00110000B
XCHD   A,@R1 ;R5= NUMBER DISPLAY
MOV    A,R5
LCALL  LCDWD ;DISPLAY MIN
RET

```

***** MENU DISPLAY *****

```

BOTDIS:  MOV    A,#0D0H
          LCALL  LCDWI
          MOV    DPTR,#BOTTOM
          MOV    R2,#16

```

```

BOTDIS1: CLR    A
          MOVC   A,@A+DPTR
          PUSH   DPH
          PUSH   DPL
          LCALL  LCDWD
          POP    DPL
          POP    DPH
          INC    DPTR
          DJNZ   R2,BOTDIS1
          RET

```

```

MENU1:   MOV    A,#80H
          LCALL  LCDWI
          MOV    DPTR,#MENU1R1
          MOV    R2,#16

```

```

MENU11:  CLR    A
          MOVC   A,@A+DPTR
          PUSH   DPH
          PUSH   DPL
          LCALL  LCDWD
          POP    DPL
          POP    DPH
          INC    DPTR
          DJNZ   R2,MENU11

```

```

MOV    A,#0C0H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

LCALL LCDWI
MOV DPTR,#MENU1R2
MOV R2,#16
MENU12: CLR A
        MOV A,@A+DPTR
        PUSH DPH
        PUSH DPL
        LCALL LCDWD
        POP DPL
        POP DPH
        INC DPTR
        DJNZ R2,MENU12

```

```

MOV A,#90H
LCALL LCDWI
MOV DPTR,#MENU1R3
MOV R2,#16
MENU13: CLR A
        MOV A,@A+DPTR
        PUSH DPH
        PUSH DPL
        LCALL LCDWD
        POP DPL
        POP DPH
        INC DPTR
        DJNZ R2,MENU13

```

```

MOV A,#0DCH
LCALL LCDWI
MOV DPTR,#MENU1R4
MOV R2,#4

```

```

MENU14: CLR A
        MOV A,@A+DPTR
        PUSH DPH
        PUSH DPL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCALL LCDWD
POP DPL
POP DPH
INC DPTR
DJNZ R2,MENU14
RET
```

```
MENU2: MOV A,#0C2H
        LCALL LCDWI
        MOV DPTR,#MENU21
        MOV R2,#12
```

```
MEU21: CLR A
        MOVC A,@A+DPTR
        PUSH DPH
        PUSH DPL
        LCALL LCDWD
        POP DPL
        POP DPH
        INC DPTR
        DJNZ R2,MEU21

        MOV A,#0D0H
        LCALL LCDWI
        MOV DPTR,#MENU22
        MOV R2,#16
```

```
MEU22: CLR A
        MOVC A,@A+DPTR
        PUSH DPH
        PUSH DPL
        LCALL LCDWD
        POP DPL
        POP DPH
        INC DPTR
        DJNZ R2,MEU22
        RET
```

```
MENU2SET: MOV A,#084H
```

```

        LCALL LCDWI
        MOV  DPTR,#MENU2SETT
        MOV  R2,#8
MENU2S1:  CLR  A
        MOVC A,@A+DPTR
        PUSH DPH
        PUSH DPL
        LCALL LCDWD
        POP  DPL
        POP  DPH
        INC  DPTR
        DJNZ R2,MENU2S1
        RET

MENU2PRO: MOV  A,#0C4H
        LCALL LCDWI
        MOV  DPTR,#MENU2PROG
        MOV  R2,#7
MENU2P1:  CLR  A
        MOVC A,@A+DPTR
        PUSH DPH
        PUSH DPL
        LCALL LCDWD
        POP  DPL
        POP  DPH
        INC  DPTR
        DJNZ R2,MENU2P1
        RET

MENU2MA:  MOV  A,#0C5H
        LCALL LCDWI
        MOV  DPTR,#MENU2MAN
        MOV  R2,#6
MENU2M1:  CLR  A
        MOVC A,@A+DPTR
        PUSH DPH
        PUSH DPL

```

```

        LCALL LCDWD
        POP  DPL
        POP  DPH
        INC  DPTR
        DJNZ R2,MENU2M1
        RET
MENU2DA: MOV  A,#0C2H
        LCALL LCDWI
        MOV  DPTR,#MENU2DAT
        MOV  R2,#12
MENU2D1: CLR  A
        MOVC A,@A+DPTR
        PUSH DPH
        PUSH DPL
        LCALL LCDWD
        POP  DPL
        POP  DPH
        INC  DPTR
        DJNZ R2,MENU2D1
        RET
MENU3:  LCALL MENU2SET
        MOV  R4,#095H
        LCALL TDISP
        LCALL TDISP
        RET
MENU4:  MOV  A,#085H
        LCALL LCDWI
        MOV  DPTR,#MENU2MAN
        MOV  R2,#6
MENU41: CLR  A
        MOVC A,@A+DPTR
        PUSH DPH
        PUSH DPL
        LCALL LCDWD
        POP  DPL
        POP  DPH
        INC  DPTR

```

DJNZ R2,MENU41

MOV A,#0C0H

LCALL LCDWI

MOV DPTR,#MENU4R2

MOV R2,#16

MENU42: CLR A

MOVC A,@A+DPTR

PUSH DPH

PUSH DPL

LCALL LCDWD

POP DPL

POP DPH

INC DPTR

DJNZ R2,MENU42

MOV A,#90H

LCALL LCDWI

MOV DPTR,#MENU4R3

MOV R2,#16

MENU43: CLR A

MOVC A,@A+DPTR

PUSH DPH

PUSH DPL

LCALL LCDWD

POP DPL

POP DPH

INC DPTR

DJNZ R2,MENU43

RET

MENU6: MOV A,#81H

LCALL LCDWI

MOV DPTR,#RUN

MOV R2,#3

MENU61: CLR A

```
MOVC A,@A+DPTR
PUSH DPH
PUSH DPL
LCALL LCDWD
POP DPL
POP DPH
INC DPTR
DJNZ R2,MENU61
```

```
MOV A,#84H
LCALL LCDWI
MOV DPTR,#MENU6R1
MOV R2,#8
MENU62: CLR A
MOVC A,@A+DPTR
PUSH DPH
PUSH DPL
LCALL LCDWD
POP DPL
POP DPH
INC DPTR
DJNZ R2,MENU62
```

```
MOV A,#0C0H
LCALL LCDWI
MOV DPTR,#MENU6R2
MOV R2,#13
MENU63: CLR A
MOVC A,@A+DPTR
PUSH DPH
PUSH DPL
LCALL LCDWD
POP DPL
POP DPH
INC DPTR
DJNZ R2,MENU63
```



```

MOV    A,#0D0H
LCALL  LCDWI
MOV    DPTR,#MENU6R3
MOV    R2,#16
MENU64: CLR    A
        MOVC   A,@A+DPTR
        PUSH   DPH
        PUSH   DPL
        LCALL  LCDWD
        POP    DPL
        POP    DPH
        INC    DPTR
        DJNZ   R2,MENU64
        RET
MENU7:  MOV    A,#81H
        LCALL  LCDWI
        MOV    DPTR,#MENU7R1
        MOV    R2,#13D
MENU71: CLR    A
        MOVC   A,@A+DPTR
        PUSH   DPH
        PUSH   DPL
        LCALL  LCDWD
        POP    DPL
        POP    DPH
        INC    DPTR
        DJNZ   R2,MENU71

        MOV    A,#0C1H
        LCALL  LCDWI
        MOV    DPTR,#MENU7R2
        MOV    R2,#14D
MENU72: CLR    A
        MOVC   A,@A+DPTR

```

```

PUSH  DPH
PUSH  DPL
LCALL LCDWD
POP   DPL
POP   DPH
INC   DPTR
DJNZ  R2,MENU72

```

```
RET
```

```

CLS:    MOV  A,#00000001B ;CLEAR
        LCALL LCDWI
        MOV  R6,#0AH
        LCALL DELAY1MSEC
        RET

```

```
***** DELAY LOOP 1msec(R6) *****
```

```

DELAY1MSEC: MOV  R0,#02H
DL1:      MOV  R1,#0E5H
DL2:      DJNZ  R1,DL2
          DJNZ  R0,DL1
          DJNZ  R6,DELAY1MSEC
          RET

```

```
*****SACN KEY B(KEY)*****
```

```

SCANKEY: MOV  B,#00H
          MOV  DPTR,#DP1C
          MOVX A,@DPTR          ;A=XXXX KKKK
          XRL  A,#0FFH
          ANL  A,#1FH           ;A =000K KKKK
          MOV  R7,A
          JNZ  CHECKKEY        ;IF PRESS JUMP

```

```

RTN:     MOV  B,#00H
          RET          ;NOT PRESS

```

```

CHECKKEY:  MOV  R1,#08H
           MOV  R6,#01H

LOOP:      RLC  A           ;ROTATE TO CARRYFLAG
           JNC  INLOOP      ;C =0 JUMP DJNZ
           DEC  R6          ;C =1 DEC R6

INLOOP:    DJNZ R1,LOOP
           MOV  A,R6
           JZ   CHECKAGAIN  ;R6=0 JUMP
           LJMP RTN         ;R6<>0 RET

```

```

CHECKAGAIN: MOV  R6,#0AH      ;DELAY=18msec
            LCALL DELAY1MSEC

```

```

MOV  DPTR,#DP1C
MOVX A,@DPTR      ;A=XXXX KKKK
XRL  A,#0FFH
ANL  A,#1FH       ;A =000K KKKK
CJNE A,07H,RTN
MOV  B,A

MOV  R6,#0AH      ;DELAY=18msec
LCALL DELAY1MSEC

```

```

PRESS:     MOV  DPTR,#DP1C
            MOVX A,@DPTR      ;A=XXXX KKKK
            XRL  A,#0FFH
            ANL  A,#1FH       ;A =000K KKKK
            JZ   KEYEND       ;IF NOT PRESS JUMP
            LJMP PRESS

```

```

KEYEND:    MOV  B,R7
            RET

```

*****DECDPTR*****

```

DDEC:      MOV  A,DPL
            CJNE A,#00H,DDEC1
            DEC  DPH

```

```
DDEC1:      DEC   DPL
            RET
```

```
;*****BCDINC(A)*****
```

```
BCDINC:      INC   A
            MOV   R0,A
            ANL   A,#0FH
            CJNE  A,#0AH,BCDINC1
            MOV   A,R0
            ADD   A,#06H
            RET
```

```
BCDINC1:     MOV   A,R0
            RET
```

```
;*****BCDDEC(A)*****
```

```
BCDDEC:      DEC   A
            MOV   R0,A
            ANL   A,#0FH
            CJNE  A,#0FH,BCDDEC1
            MOV   A,R0
            SUBB  A,#06H
            RET
```

```
BCDDEC1:     MOV   A,R0
            RET
```

```
;*****BCDTOHEX D1 D2 D3 HEXH HEXL*****
```

```
BCDTOHEX:    MOV   DPTR,#0064H    ;HUNDREAD
            MOV   R2,#00H
            MOV   R3,D1
            LCALL DPMUL
            MOV   HEXH,DPH
            MOV   HEXL,DPL
```

```
            MOV   DPTR,#000AH    ;TEN
            MOV   R2,#00H
            MOV   R3,D2
            LCALL DPMUL
            MOV   R2,HEXH
            MOV   R3,HEXL
            LCALL DPADD
```

```

MOV    HEXH,DPH
MOV    HEXL,DPL

MOV    DPTR,#0001H    ;UNIT
MOV    R2,#00H
MOV    R3,D3
LCALL  DPMUL
MOV    R2,HEXH
MOV    R3,HEXL
LCALL  DPADD
MOV    HEXH,DPH
MOV    HEXL,DPL
RET

```

;*****HEXTOBCD DPH,DPL BCD1 BCD2 BCD3 BCD4 BCD5*****

HEXTOBCD:

```

MOV    R2,#27H
MOV    R3,#10H    ;DIV 10000
LCALL  DPDIV
MOV    BCD1,DPL

MOV    DPH,R4
MOV    DPL,R5
MOV    R2,#03H    ;DIV 1000
MOV    R3,#0E8H
LCALL  DPDIV
MOV    BCD2,DPL

```

```

MOV    DPH,R4
MOV    DPL,R5
MOV    R2,#00H
MOV    R3,#64H    ;DIV 100
LCALL  DPDIV
MOV    BCD3,DPL

```

```

MOV    DPH,R4
MOV    DPL,R5
MOV    R2,#00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV R3,#0AH ;DIV 10
LCALL DPDIV
MOV BCD4,DPL

```

```

MOV DPH,R4
MOV DPL,R5
MOV R2,#00H
MOV R3,#01H ;DIV 1
LCALL DPDIV
MOV BCD5,DPL

```

```

RET

```

```

;***** DPMUL SUB ***** 12
; IN = DPTR MULTIPLIER
; = R2,R3 MULTIPLICAND
; OUT = DPTR RESULT
; = CARRY FLAG SET WHEN RESULT OVERFLOW
; REG = A,R1,R2,R3,R4,R5,DPTR

```

```

DPMUL: MOV R4,#0 ;CLEAR RESULT
MOV R5,#0
CLR MULFG
MOV R1,#16

```

```

DPMUL1: MOV A,R5 ;*2
ADD A,R5
MOV R5,A
MOV A,R4
ADDC A,R4
MOV R4,A
JNC DPMUL2 ;NO CARRY
SETB MULFG ;OVERFLOW

```

```

DPMUL2: MOV A,R3
RLC A
MOV R3,A
MOV A,R2

```



```

RLC A
MOV R2,A
JNC DPMUL3
MOV A,R5          ;R4R5=R4R5+DPTR
ADD A,DPL
MOV R5,A
MOV A,R4
ADDC A,DPH
MOV R4,A
JNC DPMUL3        ;NO CARRY
SETB MULFG        ;OVERFLOW
DPMUL3: DJNZ R1,DPMUL1
MOV DPH,R4
MOV DPL,R5
MOV C,MULFG        ;LOAD MULFG TO CARRY
RET
; ***** DPADD SUB ***** 10
; DPTR = DPTR + R2,R3
; IN = DPTR,R2,R3
; OUT = DPTR
; REG = A,DPTR

DPADD: MOV A,DPL
ADD A,R3
MOV DPL,A
MOV A,DPH
ADDC A,R2
MOV DPH,A
RET

```

```

; ***** DPDIV SUB ***** 13
; IN = DPTR DIVIDEND
; = R2,R3 DIVISOR
; OUT = DPTR RESULT
; = R4,R5 REMAINDER
; = CARRY FLAG SET WHEN HAS REMAINDER

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; REG = A,R1,R2,R3,R4,R5,DPTR

DPDIV: CLR C
MOV R4,#0 ;CLEAR RESULT

MOV R5,#0
MOV R1,#16

DPDIV1: MOV A,DPL
RLC A
MOV DPL,A
MOV A,DPH
RLC A
MOV DPH,A

MOV A,R5 ;*2 (WITH CARRY)

ADDC A,R5

MOV R5,A

MOV A,R4

ADDC A,R4

MOV R4,A

MOV A,R5 ;R4R5=R4R5-R2R3 (WITH CARRY)

SUBB A,R3

MOV R5,A

MOV A,R4

SUBB A,R2

MOV R4,A

JNC DPDIV2

MOV A,R5 ;R4R5=R4R5+R2R3

ADD A,R3

MOV R5,A

MOV A,R4

ADDC A,R2

MOV R4,A

DPDIV2: CPL C
DJNZ R1,DPDIV1

```

MOV A,DPL          ;*2 (WITH CARRY)
ADDC A,DPL
MOV DPL,A
MOV A,DPH
ADDC A,DPH
MOV DPH,A

MOV A,R4
ORL A,R5
JZ DPDIV3
SETB C              ;HAS REMAINDER->CARRY SET
RET
DPDIV3: CLR C        ;NO REMAINDER->CARRY CLEAR
RET
; ***** DPSUB SUB ***** 11
; DPTR = DPTR - R2,R3
; IN = DPTR,R2,R3
; OUT = DPTR
; REG = A,DPTR

DPSUB: CLR C
MOV A,DPL
SUBB A,R3
MOV DPL,A
MOV A,DPH
SUBB A,R2
MOV DPH,A
RET

;*****COMP16 DPTR R0,R1,C *****
DPCOM: CLR C
MOV A,DPL
CJNE A,00H,DPCOM1
MOV A,DPH
CJNE A,01H,DPCOM1
SETB C              ;SET IF EQUAL 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
DPCOM1: CLR C ;CLR IF NOT EQUAL 0
RET
;*****SELCTKEYTIME*****
SELECTTIME: LJMP RIGHT
SELECTKEYTIME: LCALL SCANKEY
MOV A,B
CJNE A,#00H,S2
LJMP SELECTKEYTIME
S2: CJNE A,#01H,S3
LJMP REFT
S3: CJNE A,#02H,S4
LJMP RIGHT
S4: CJNE A,#04H,S5
LJMP UP
S5: CJNE A,#08H,S6
LJMP DOWN
S6: CJNE A,#10H,SELECTKEYTIME
LJMP OK
LOOP2: LJMP SELECTKEYTIME
REFT: MOV A,#0C6H ;SET HOUR
LCALL LCDWI
MOV A,#0001110B
LCALL LCDWI
MOV R3,#00H
LJMP SELECTKEYTIME

RIGHT: MOV A,#0C9H ;SET MIN
LCALL LCDWI
MOV A,#00001110B
LCALL LCDWI
MOV R3,#0FFH
LJMP SELECTKEYTIME

UP: CJNE R3,#00H,SETUMIN

```

```

MOV    A,TH
CJNE   A,#23H,UH      ;TH BCD
MOV    TH,#00H
MOV    R4,#0C5H
LCALL  DISHOUR
MOV    A,#0C6H
LCALL  LCDWI
LJMP   SELECTKEYTIME

```

```

UH:     MOV    A,TH
        LCALL  BCDINC
        MOV    TH,A
        MOV    R4,#0C5H
        LCALL  DISHOUR
        MOV    A,#0C6H
        LCALL  LCDWI
        LJMP   SELECTKEYTIME

```

```

SETUMIN: CJNE   R3,#0FFH,LOOP2
          MOV    A,TM
          CJNE   A,#59H,UM
          MOV    TM,#00H
          MOV    R4,#0C8H
          LCALL  DISMIN
          MOV    A,#0C9H
          LCALL  LCDWI
          LJMP   SELECTKEYTIME

```

```

UM:     MOV    A,TM
        LCALL  BCDINC
        MOV    TM,A
        MOV    R4,#0C8H
        LCALL  DISMIN
        MOV    A,#0C9H
        LCALL  LCDWI
        LJMP   SELECTKEYTIME

```

DOWN: CJNE R3,#00H,SETDMIN
 MOV A,TH
 CJNE A,#00H,DH
 MOV TH,#23H
 MOV R4,#0C5H
 LCALL DISHOUR
 MOV A,#0C6H
 LCALL LCDWI
 LJMP SELECTKEYTIME

DH: MOV A,TH
 LCALL BCDDEC
 MOV TH,A
 MOV R4,#0C5H
 LCALL DISHOUR
 MOV A,#0C6H
 LCALL LCDWI
 LJMP SELECTKEYTIME

SETDMIN: CJNE R3,#0FFH,LOOP3
 MOV A,TM
 CJNE A,#00H,DM
 MOV TM,#59H
 MOV R4,#0C8H
 LCALL DISMIN
 MOV A,#0C9H
 LCALL LCDWI
 LJMP SELECTKEYTIME

DM: MOV A,TM
 LCALL BCDDEC
 MOV TM,A
 MOV R4,#0C8H
 LCALL DISMIN
 MOV A,#0C9H
 LCALL LCDWI


```

                                LJMP  SELECTKEYTIME
LOOP3:                        LJMP  SELECTKEYTIME

OK:                            MOV   A,#00001100B
                                LCALL LCDWI
                                LCALL SETTIME1
                                LJMP  MAIN
;***** SETTIME *****

```

SETTIME:

```

                                MOV   R4,#0C5H      ;START ADDRESS
                                LCALL  TDISP
                                LCALL  BOTDIS
                                LCALL  MENU2SET
                                LCALL  SELECTTIME

SETTIME1:                      MOV   R6,#8EH ;WRITE PROTECT COMMAND
                                MOV   R7,#00H
                                LCALL  TBYTEWR

                                MOV   R6,#80H ;WRITE SECOND
                                MOV   R7,#00H ;SEC=0
                                LCALL  TBYTEWR

                                MOV   R6,#82H ;WRITE MIN
                                MOV   R7,TM ;TM SET MIN
                                LCALL  TBYTEWR

                                MOV   R6,#84H ;WRITE HOUR
                                MOV   R7,TH ;TH SET HOUR
                                LCALL  TBYTEWR

                                MOV   R6,#8EH ;WRITE PROTECT ACTIVE
                                MOV   R7,#80H
                                LCALL  TBYTEWR
                                RET

```

*****MAN*****

```
MAN:      LCALL MENU4
          LCALL BOTDIS

          MOV D1,V1
          MOV D2,V2
          MOV D3,V3
          MOV A,#0C5H
          LCALL VDISP
          MOV A,#0CCH
          LCALL VDISP          ;VOLT DISPLAY

          LCALL CALPOW          ;V*I

          MOV A,#95H
          LCALL PDISP          ;POWER DISPLAY
          MOV A,#9BH
          LCALL PDISP

KV:        MOV A,#0CEH
          LCALL LCDWI
          MOV A,#000001110B
          LCALL LCDWI

SELECMAN:  LCALL SCANKEY
          MOV A,B
          CJNE A,#00H,SM7
          LJMP SELECMAN

SM7:       CJNE A,#01H,SM3
          LJMP KV

SM3:       CJNE A,#02H,SM4
          LJMP KV

SM4:       CJNE A,#04H,SM5
          LJMP UPM

SM5:       CJNE A,#08H,SM6
          LJMP DOWNM

SM6:       CJNE A,#10H,SELECMAN
```

LJMP OKM

UPM: MOV R1,V1
 MOV R2,V2
 MOV R3,V3
 CJNE R3,#HIGHV3,UPM1
 CJNE R2,#HIGHV2,UPM1
 CJNE R1,#HIGHV1,UPM1
 MOV R1,#LOWV1
 MOV R2,#LOWV2
 MOV R3,#LOWV3
 LJMP UPM2

UPM1: LCALL UP3DIG

UPM2: MOV V1,R1
 MOV V2,R2
 MOV V3,R3
 MOV A,#0CCH
 MOV D1,V1
 MOV D2,V2
 MOV D3,V3
 LCALL VDISP
 LCALL CALPOW
 MOV A,#9BH
 LCALL PDISP
 LJMP KV

DOWNM:

 MOV R1,V1
 MOV R2,V2
 MOV R3,V3
 CJNE R3,#LOWV3,DOWNM1
 CJNE R2,#LOWV2,DOWNM1
 CJNE R1,#LOWV1,DOWNM1

```
MOV R1,#HIGHV1
MOV R2,#HIGHV2
MOV R3,#HIGHV3
LJMP DOWNM2
```

```
DOWNM1: LCALL DOWN3DIG
```

```
DOWNM2: MOV V1,R1
MOV V2,R2
MOV V3,R3
MOV A,#0CCH
MOV D1,V1
MOV D2,V2
MOV D3,V3
LCALL VDISP
LCALL CALPOW
MOV A,#9BH
LCALL PDISP
LJMP KV
```

```
OKM: MOV A,#00001100B
LCALL LCDWI
LCALL OUTVOLT
MOV VS1,V1
MOV VS2,V2
MOV VS3,V3
LCALL CLS

MOV A,#0D7H
LCALL LCDWI
MOV DPTR,#MANMENU1
MOV R2,#3
```

```
MANMENU: CLR A
MOVC A,@A+DPTR
PUSH DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH  DPL
LCALL LCDWD
POP   DPL
POP   DPH
INC   DPTR
DJNZ  R2,MANMENU
CLR   RUNPG
LJMP  MAIN1

```

***** UP3DIG R1,R2,R3 *****

```

UP3DIG:  MOV  A,R3
         CJNE A,#09H,UP3DIG1
         MOV  R3,#00H
         MOV  A,R2
         CJNE A,#09H,UP3DIG2
         MOV  R2,#00H
         MOV  A,R1
         CJNE A,#09H,UP3DIG3
         MOV  R1,#00H
         RET

```

```

UP3DIG1: MOV  A,R3
         LCALL BCDINC
         MOV  R3,A
         RET

```

```

UP3DIG2: MOV  A,R2
         LCALL BCDINC
         MOV  R2,A
         RET

```

```

UP3DIG3: MOV  A,R1
         LCALL BCDINC
         MOV  R1,A
         RET

```

***** DOWN3DIG R1,R2,R3 *****

```
DOWN3DIG:  MOV    A,R3
           CJNE   A,#00H,DOWN3DIG1
           MOV    R3,#09H
           MOV    A,R2
           CJNE   A,#00H,DOWN3DIG2
           MOV    R2,#09H
           MOV    A,R1
           CJNE   A,#00H,DOWN3DIG3
           MOV    R1,#09H
           RET
```

```
DOWN3DIG1: MOV    A,R3
           LCALL  BCDDEC
           MOV    R3,A
           RET
```

```
DOWN3DIG2: MOV    A,R2
           LCALL  BCDDEC
           MOV    R2,A
           RET
```

```
DOWN3DIG3: MOV    A,R1
           LCALL  BCDDEC
           MOV    R1,A
           RET
```

***** CALPOW D1,D2,D3,I1,I2,I3,PO1,PO2,PO3 *****

```
CALPOW:   LCALL  BCDTOHEX
           MOV    R6,HEXH
           MOV    R7,HEXL

           MOV    D1,I1
           MOV    D2,I2
           MOV    D3,I3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCALL BCDTOHEX

```
MOV R2,HEXH
MOV R3,HEXL
MOV DPH,R6
MOV DPL,R7
LCALL DPMUL
LCALL HEXTOBCD
MOV PO1,BCD2
MOV PO2,BCD3
MOV PO3,BCD4
RET
```

*****VDISP A ADD D1 D2 D3 *****

VDISP: LCALL LCDWI
MOV A,D1
ANL A,#0FH
ORL A,#30H
LCALL LCDWD
MOV A,D2
ANL A,#0FH
ORL A,#30H
LCALL LCDWD
MOV A,D3
ANL A,#0FH
ORL A,#30H
LCALL LCDWD
RET

*****PDISP PO1 PO2 PO3 *****

PDISP: LCALL LCDWI
MOV A,PO1
ANL A,#0FH
ORL A,#30H
LCALL LCDWD

MOV A,#2EH

LCALL LCDWD

MOV A,PO2

ANL A,#0FH

ORL A,#30H

LCALL LCDWD

RET

*****OUTVOLT V1,V2,V3 *****

OUTVOLT: MOV D1,V1
MOV D2,V2
MOV D3,V3
LCALL BCDTOHEX
MOV DPH,HEXH
MOV DPL,HEXL
MOV R2,#DIVVH
MOV R3,#DIVVL
LCALL DPDIV
MOV A,DPL
MOV DPTR,#DP2B
MOVX @DPTR,A
MOV R6,#010H
LCALL DELAY1MSEC
RET

*****IN CURRENT*****

INCURRENT: MOV DPTR,#DP1B
MOVX A,@DPTR
ANL A,#00000111B
MOVX @DPTR,A :SEND ADD IN

MOV R6,#05H
LCALL DELAY1MSEC

ORL A,#01000000B
MOVX @DPTR,A :SEND ALE/START

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    R6,#0AH
LCALL  DELAY1MSEC

ANL    A,#10111111B
MOVX   @DPTR,A

MOV    R6,#10H
LCALL  DELAY1MSEC

ANL    A,#01111111B
ORL    A,#10000000B
MOVX   @DPTR,A      :SEND OE

MOV    R6,#05H
LCALL  DELAY1MSEC

ANL    A,#01111111B
MOVX   @DPTR,A

MOV    R6,#05H
LCALL  DELAY1MSEC

MOV    DPTR,#DP2A
MOVX   A,@DPTR      :GET I

MOV    DPL,A
MOV    DPH,#00H
MOV    R2,#MULIH
MOV    R3,#MULIL
LCALL  DPMUL
LCALL  HEXTOBCD
MOV    I1,BCD3
MOV    I2,BCD4
MOV    I3,BCD5
RET

```

*****IN VOLT*****

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INVOLT:    MOV    DPTR,#DP1B
            MOVX   A,@DPTR
            ANL    A,#00000111B      :SEND ADD IN
            ANL    A,#00001000B
            MOVX   @DPTR,A
            MOV    R6,#05H
            LCALL  DELAY1MSEC
            ORL    A,#01000000B
            MOVX   @DPTR,A          :SEND ALE/START
            MOV    R6,#0AH
            LCALL  DELAY1MSEC
            ANL    A,#10111111B
            MOVX   @DPTR,A
            MOV    R6,#10H
            LCALL  DELAY1MSEC
            ANL    A,#01111111B
            ORL    A,#10000000B
            MOVX   @DPTR,A          :SEND OE
            MOV    R6,#05H
            LCALL  DELAY1MSEC
            ANL    A,#01111111B
            MOVX   @DPTR,A
            MOV    R6,#05H
            LCALL  DELAY1MSEC
            MOV    DPTR,#DP2A
            MOVX   A,@DPTR          :GET V
            MOV    DPL,A
            MOV    DPH,#00H
            MOV    R2,#MULVH
            MOV    R3,#MULVL
            LCALL  DPMUL
            LCALL  HEXTOBCD
            MOV    V1,BCD3
            MOV    V2,BCD4
            MOV    V3,BCD5
            RET

```

*****PROGRAM*****

```
PROG:      MOV    A,#0001100
           LCALL  LCDWI
           CLR    PGEND
           CLR    PGENDPG
           CLR    TOPDATA
           LCALL  MENU6
           MOV    A,#8EH
           LCALL  LCDWI
           MOV    A,PGRUN
           LCALL  LCDWD
           LCALL  PGSTA
           LCALL  DISPG

PGSCAN:    LCALL  SCANKEY
           MOV    A,B
           CJNE   A,#00H,PRG1
           LJMP   PGSCAN

PRG1:      CJNE   A,#01H,PRG2
           LJMP   DOPG

PRG2:      CJNE   A,#02H,PRG3
           LJMP   UPPG

PRG3:      CJNE   A,#04H,PRG4
           LCALL  DOPGD
           LJMP   PGSCAN

PRG4:      CJNE   A,#08H,PRG5
           LCALL  UPPGD
           LJMP   PGSCAN

PRG5:      CJNE   A,#10H,PGSCAN
           LJMP   EDITPG

DOPG:      MOV    A,PGRUN
```

```

CJNE  A,#31H,DOPG1
MOV   PGRUN,#33H
CLR   PGENDPG
CLR   PGEND
MOV   A,#90H           ;CLEAR DISPLAY
LCALL CLSDATA
LCALL PGSTA
LCALL DISPG
LJMP  PGSCAN

```

```

DOPG1:  DEC   PGRUN
        CLR   PGENDPG
        CLR   PGEND
        MOV   A,#90H           ;CLEAR DISPLAY
        LCALL CLSDATA
        LCALL PGSTA
        LCALL DISPG
        LJMP  PGSCAN

```

```

UPPG:   MOV   A,PGRUN
        CJNE  A,#33H,UPPG1
        MOV   PGRUN,#31H
        CLR   PGENDPG
        CLR   PGEND
        MOV   A,#90H           ;CLEAR DISPLAY
        LCALL CLSDATA
        LCALL PGSTA
        LCALL DISPG
        LJMP  PGSCAN

```

```

UPPG1:  INC   PGRUN
        CLR   PGENDPG
        CLR   PGEND
        MOV   A,#90H           ;CLEAR DISPLAY
        LCALL CLSDATA
        LCALL PGSTA

```



```
LCALL DISPG
LJMP PGSCAN
```

```
UPPGD: CLR TOPDATA
JB PGEND,UPPGD1
MOV DPH,BUFDPH
MOV DPL,BUFDPL
INC DPTR
MOV BUFDPH,DPH
MOV BUFDPL,DPL
MOV A,#90H ;CLEAR DISPLAY
LCALL CLSDATA
LCALL DISPG
UPPGD1: RET
```

```
DOPGD: JB TOPDATA,UPPGD1
JB PGEND,DOPGD2
MOV DPH,BUFDPH
MOV DPL,BUFDPL
LCALL DDEC
LCALL DDEC
LCALL DDEC
```

```
LCALL DDEC
```

DOPGD1:

```
LCALL DDEC
LCALL DDEC
LCALL DDEC
```

```
MOV BUFDPH,DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV BUFDPL,DPL

MOV R0,#01H

MOV R1,#80H

LCALL DPCOM

JNC DOPGD4

SETB TOPDATA

LJMP DOPGD3

DOPGD4: MOV R0,#0C4H

MOV R1,#80H

LCALL DPCOM

JNC DOPGD5

SETB TOPDATA

LJMP DOPGD3

DOPGD5: MOV R0,#87H

MOV R1,#81H

LCALL DPCOM

JNC DOPGD3

SETB TOPDATA

DOPGD3: MOV A,#90H ;CLEAR DISPLAY

LCALL CLSDATA

LCALL DISPG

RET

DOPGD2:

MOV DPH,BUFDPH

MOV DPL,BUFDPL

LCALL DDEC

MOV BUFDPH,DPH

MOV BUFDPL,DPL

CLR PGEND

LJMP DOPGD1

EDITPG: CLR PGEND
CLR PGENDPG
CLR TOPDATA
LCALL BOTDIS

MOV A,#080H
LCALL LCDWI
MOV DPTR,#EDIT
MOV R2,#4D

EDITPG1: CLR A
MOVC A,@A+DPTR
PUSH DPH
PUSH DPL
LCALL LCDWD
POP DPL
POP DPH
INC DPTR
DJNZ R2,EDITPG1

LCALL PGSTA
MOV A,#90H ;CLEAR DISPLAY
LCALL CLSDATA
LCALL DISPG
MOV STA,#00H
LCALL SEEDPG

EDPGSCAN: LCALL SCANKEY
MOV A,B
CJNE A,#00H,EDPG1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LJMP EDPGSCAN

EDPG1: CJNE A,#01H,EDPG2

LJMP DSEDPG

EDPG2: CJNE A,#02H,EDPG3

LJMP USEDPG

EDPG3: CJNE A,#04H,EDPG4

LJMP UPEDPG

EDPG4: CJNE A,#08H,EDPG5

LJMP DOEDPG

EDPG5: CJNE A,#10H,EDPGSCAN

LJMP EDOK

EDOK: MOV A,#00001100H

LCALL LCDWI

LCALL CLS

MOV A,#0D8H

LCALL LCDWI

MOV DPTR,#PGMENU1

MOV R2,#2

PGMENU: CLR A

MOVC A,@A+DPTR

PUSH DPH

PUSH DPL

LCALL LCDWD

POP DPL

POP DPH

INC DPTR

DJNZ R2,PGMENU

SETB RUNPG

LJMP MAIN1

```

DSEDPG:    MOV    A,STA                ;DOWN SELECT
            CJNE   A,#00H,DSEDPG1
            JB     TOPDATA,DSEDPG3
            LCALL  DOPGD
            CLR    PGEND
            LJMP   DSEDPG2

DSEDPG3:    MOV    STA,#00H
            LCALL  SEEDPG
            LJMP   EDPGSCAN

DSEDPG2:    MOV    STA,#02H
            LCALL  SEEDPG
            LJMP   EDPGSCAN

DSEDPG1:    DEC    STA
            LCALL  SEEDPG
            LJMP   EDPGSCAN

USEDPG:     MOV    A,STA                ;UP SELECT
            CJNE   A,#02H,USEDPG1
            LCALL  UPPGD
            CLR    TOPDATA
            JB     PGEND,USEDPG2
            MOV    STA,#00H
            LCALL  SEEDPG
            LJMP   EDPGSCAN

USEDPG2:    MOV    STA,#00H
            MOV    A,#00001100B
            LCALL  LCDWI
            LJMP   EDPGSCAN

```

USEDPG1: JB PGEND,USEDPG2
INC STA
LCALL SEEDPG
LJMP EDPGSCAN

UPEDPG: JB PGEND,NEWPG
MOV A,STA
CJNE A,#00H,UPEDPG1
LJMP UPPGH

UPEDPG1: CJNE A,#01H,UPEDPG2
LJMP UPPGM

UPEDPG2: CJNE A,#02H,UPPGH3
LJMP UPPGV

NEWPG: MOV A,#90H
LCALL CLSDATA
CLR PGEND
MOV DPH,BUFDPH
MOV DPL,BUFDPL
MOV A,TM
MOVX @DPTR,A
INC DPTR
MOV A,TH
MOVX @DPTR,A
INC DPTR
MOV A,DV1
MOVX @DPTR,A
INC DPTR
MOV A,DV2
MOVX @DPTR,A
INC DPTR
MOV A,#0FFH
MOVX @DPTR,A
LCALL DDEC


```
LCALL DISPG
MOV A,#92H
LCALL LCDWI
MOV A,#00001110B
LCALL LCDWI
LJMP EDPGSCAN
```

```
UPPGH:  MOV DPL,BUFDPL
        MOV DPH,BUFDPH
        LCALL DDEC
        LCALL DDEC
        MOV BUFDPL,DPL
        MOV BUFDPH,DPH
        MOVX A,@DPTR
        MOV TH,A
        CJNE A,#23H,UPPGH1
        MOV TH,#00H
        LJMP UPPGH2
```

```
UPPGH1: MOV A,TH
        LCALL BCDINC
        MOV TH,A
```

```
UPPGH2: MOV DPL,BUFDPL
        MOV DPH,BUFDPH
        MOV A,TH
        MOVX @DPTR,A
        INC DPTR
        INC DPTR
        MOV BUFDPL,DPL
        MOV BUFDPH,DPH

        MOV R4,#091H
        LCALL DISHOUR
```

```

MOV    A,#092H
LCALL  LCDWI
UPPGH3: LJMP  EDPGSCAN

UPPGM:  MOV    DPL,BUFDPL
        MOV    DPH,BUFDPH
        LCALL  DDEC
        LCALL  DDEC
        LCALL  DDEC
        MOV    BUFDPL,DPL
        MOV    BUFDPH,DPH
        MOVX   A,@DPTR
        MOV    TM,A
        CJNE   A,#59H,UPPGM1
        MOV    TM,#00H
        LJMP   UPPGM2

UPPGM1: MOV    A,TM
        LCALL  BCDINC
        MOV    TM,A

UPPGM2: MOV    DPL,BUFDPL
        MOV    DPH,BUFDPH
        MOV    A,TM
        MOVX   @DPTR,A
        INC    DPTR
        INC    DPTR
        INC    DPTR
        MOV    BUFDPL,DPL
        MOV    BUFDPH,DPH
        MOV    R4,#094H
        LCALL  DISMIN
        MOV    A,#095H
        LCALL  LCDWI
UPPGM3: LJMP  EDPGSCAN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UPPGV: MOV DPL,BUFDPL
 MOV DPH,BUFDPH
 MOVX A,@DPTR
 MOV DV2,A
 LCALL DDEC
 MOVX A,@DPTR
 MOV DV1,A
 MOV BUFDPL,DPL
 MOV BUFDPH,DPH

 MOV DPH,DV1 ; DISDATA
 MOV DPL,DV2
 LCALL HEXTOBCD
 MOV R1,BCD3
 MOV R2,BCD4
 MOV R3,BCD5
 CJNE R3,#HIGHV3,UPPGV1
 CJNE R2,#HIGHV2,UPPGV1
 CJNE R1,#HIGHV1,UPPGV1
 MOV R1,#LOWV1
 MOV R2,#LOWV2
 MOV R3,#LOWV3
 LJMP UPPGV2

UPPGV1: LCALL UP3DIG

UPPGV2: MOV D1,R1
 MOV D2,R2
 MOV D3,R3
 MOV A,#09BH
 LCALL VDISP
 LCALL BCDTOHEX

 MOV DV1,HEXH
 MOV DV2,HEXL

MOV DPL,BUFDPL
MOV DPH,BUFDPH

MOV A,DV1
MOVX @DPTR,A
INC DPTR
MOV A,DV2
MOVX @DPTR,A

MOV BUFDPL,DPL
MOV BUFDPH,DPH
MOV A,#09DH
LCALL LCDWI
LJMP EDPGSCAN

DOEDPG: JB PGEND,DOEDPG3
MOV A,STA
CJNE A,#00H,DOEDPG1
LJMP DOPGH

DOEDPG1: CJNE A,#01H,DOEDPG2
LJMP DOPGM

DOEDPG2: CJNE A,#02H,DOPGH3
LJMP DOPGV

DOEDPG3: LJMP NEWPG

DOPGH: MOV DPL,BUFDPL
MOV DPH,BUFDPH
LCALL DDEC
LCALL DDEC
MOV BUFDPL,DPL
MOV BUFDPH,DPH

```

MOVX  A,@DPTR
MOV   TH,A
CJNE  A,#00H,DOPGH1
MOV   TH,#23H
LJMP  DOPGH2

DOPGH1:  MOV   A,TH
         LCALL BCDDEC
         MOV   TH,A

DOPGH2:  MOV   DPL,BUFDPL
         MOV   DPH,BUFDPH
         MOV   A,TH
         MOVX  @DPTR,A
         INC   DPTR
         INC   DPTR
         MOV   BUFDPL,DPL
         MOV   BUFDPH,DPH
         MOV   R4,#091H
         LCALL DISHOUR

         MOV   A,#092H
         LCALL LCDWI

DOPGH3:  LJMP  EDPGSCAN

DOPGM:   MOV   DPL,BUFDPL
         MOV   DPH,BUFDPH
         LCALL DDEC
         LCALL DDEC
         LCALL DDEC
         MOV   BUFDPL,DPL
         MOV   BUFDPH,DPH
         MOVX  A,@DPTR
         MOV   TM,A
         CJNE  A,#00H,DOPGM1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    TM,#59H

LJMP   DOPGM2

DOPGM1:  MOV    A,TM
         LCALL  BCDDEC
         MOV    TM,A

DOPGM2:  MOV    DPL,BUFDPL
         MOV    DPH,BUFDPH
         MOV    A,TM
         MOVX   @DPTR,A
         INC    DPTR
         INC    DPTR
         INC    DPTR
         MOV    BUFDPL,DPL
         MOV    BUFDPH,DPH
         MOV    R4,#094H
         LCALL  DISMIN
         MOV    A,#095H
         LCALL  LCDWI
DOPGM3:  LJMP   EDPGSCAN

DOPGV:   MOV    DPL,BUFDPL
         MOV    DPH,BUFDPH
         MOVX   A,@DPTR
         MOV    DV2,A
         LCALL  DDEC
         MOVX   A,@DPTR
         MOV    DV1,A
         MOV    BUFDPL,DPL
         MOV    BUFDPH,DPH

         MOV    DPH,DV1      ; DISDATA
         MOV    DPL,DV2
         LCALL  HEXTOBCD
         MOV    R1,BCD3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MOV    R2,BCD4
MOV    R3,BCD5

CJNE   R3,#LOWV3,DOPGV1
CJNE   R2,#LOWV2,DOPGV1
CJNE   R1,#LOWV1,DOPGV1
MOV     R1,#HIGHV1
MOV     R2,#HIGHV2
MOV     R3,#HIGHV3
LJMP    DOPGV2

```

```
DOPGV1:  LCALL  DOWN3DIG
```

```

DOPGV2:  MOV     D1,R1
          MOV     D2,R2
          MOV     D3,R3
          MOV     A,#09BH
          LCALL   VDISP
          LCALL   BCDTOHEX

          MOV     DV1,HEXH
          MOV     DV2,HEXL

          MOV     DPL,BUFDPL
          MOV     DPH,BUFDPH

          MOV     A,DV1
          MOVX    @DPTR,A
          INC     DPTR
          MOV     A,DV2
          MOVX    @DPTR,A

          MOV     BUFDPL,DPL
          MOV     BUFDPH,DPH
          MOV     A,#09DH
          LCALL   LCDWI
          LJMP    EDPGSCAN

```

SEEDPG: MOV A,STA
 CJNE A,#00H,SEEDPG1
 MOV A,#92H
 LCALL LCDWI
 MOV A,#000001110B
 LCALL LCDWI
 RET

SEEDPG1: CJNE A,#01H,SEEDPG2
 MOV A,#95H
 LCALL LCDWI
 MOV A,#000001110B
 LCALL LCDWI
 RET

SEEDPG2: CJNE A,#02H,SEEDPG
 MOV A,#9DH
 LCALL LCDWI
 MOV A,#000001110B
 LCALL LCDWI
 RET

PGSTA: MOV A,PGRUN
 CJNE A,#31H,PG1
 MOV DPTR,#STAPG1
 MOV BUFDPH,DPH
 MOV BUFDPL,DPL
 SETB TOPDATA
 RET

PG1: CJNE A,#32H,PG2
 MOV DPTR,#STAPG2
 MOV BUFDPH,DPH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    BUFDPL,DPL
SETB   TOPDATA
RET
PG2:   CJNE   A,#33H,PGSTA
MOV     DPTR,#STAPG3
MOV     BUFDPH,DPH
MOV     BUFDPL,DPL
SETB    TOPDATA
RET

```

```

DISPG:  MOV    A,#8EH
        LCALL  LCDWI
        MOV    A,PGRUN
        LCALL  LCDWD
DATAPGDIS: MOV    DPH,BUFDPH
        MOV    DPL,BUFDPL
        MOVX   A,@DPTR
        CJNE   A,#0FFH,DATAPGDIS1
        LCALL  PGDEND
        RET

```

```

DATAPGDIS1: MOVX  A,@DPTR
             MOV   TM,A

```

```

INC     DPTR

```

```

MOVX    A,@DPTR
MOV     TH,A

```

```

INC     DPTR

```

```

MOVX    A,@DPTR
MOV     DV1,A

```

INC DPTR

MOVX A,@DPTR

MOV DV2,A

MOV BUFDPH,DPH

MOV BUFDPL,DPL

MOV R4,#91H

LCALL DISHOUR

MOV A,#3AH

LCALL LCDWD ;DISPLAY COLON

MOV R4,#94H

LCALL DISMIN

MOV DPH,DV1 ;DISDATA

MOV DPL,DV2

LCALL HEXTOBCD

MOV D1,BCD3

MOV D2,BCD4

MOV D3,BCD5

MOV A,#9BH

LCALL VDISP

RET

PGDEND: SETB PGEND

MOV DPH,BUFDPH

MOV DPL,BUFDPL

INC DPTR

MOVX A,@DPTR

CJNE A,#0FFH,PGEND1 ;CHECK END PG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        SETB  PGENDPG          ;END
PGEND1:  MOV   A,#093H
        LCALL LCDWI
        MOV   DPTR,#PGENDMENU
        MOV   R2,#11D

```

```

PGENDMENU1: CLR  A
          MOVC  A,@A+DPTR
          PUSH  DPH
          PUSH  DPL
          LCALL LCDWD
          POP   DPL
          POP   DPH
          INC   DPTR
          DJNZ  R2,PGENDMENU1
          MOV   A,#00001100B
          LCALL LCDWI
          RET

```

*****CLSDATA A ADD *****

```

CLSDATA:  LCALL LCDWI
          MOV   R2,#16D

CLSDATA1: MOV   A,#0FEH
          LCALL LCDWD
          DJNZ  R2,CLSDATA1

          RET

```

*****DISPLAY*****

```

DISPLAY:  LCALL MENU7
          LCALL BOTDIS
          MOV   DPTR,#STADATA
          MOV   BUFDPH,DPH
          MOV   BUFDPL,DPL
          MOV   A,#00001100B
          LCALL LCDWI

```

```

MOV DPTR,#ENDDH
MOVX A,@DPTR
CJNE A,BUFDPH,DIS6
MOV DPTR,#ENDDL
MOVX A,@DPTR
CJNE A,BUFDPL,DIS6
MOV A,#90H
LCALL CLSDATA
LCALL PGEND1
LJMP DISKSCAN

DIS6: MOV A,#90H
LCALL CLSDATA
LCALL DISDATA

DISKSCAN: LCALL SCANKEY
MOV A,B
CJNE A,#00H,DIS1
LJMP DISKSCAN

DIS1: CJNE A,#01H,DIS2
LJMP DISKSCAN

DIS2: CJNE A,#02H,DIS3
LJMP DISKSCAN

DIS3: CJNE A,#04H,DIS4
LJMP DWDATA

DIS4: CJNE A,#08H,DIS5
LJMP UPDATA

DIS5: CJNE A,#10H,DISKSCAN
LJMP DISOK

DISOK: LJMP MAIN

DWDATA: MOV A,BUFDPL

```



```
CJNE A,#0FFH,DWDATA1
MOV A,BUFDPH
CJNE A,#82H,DWDATA1
LJMP DISKSCAN
```

```
DWDATA1: MOV DPH,BUFDPH
MOV DPL,BUFDPL
MOV R2,#00H
MOV R3,#06H
LCALL DPSUB
MOV BUFDPH,DPH
MOV BUFDPL,DPL
MOV A,#90H
LCALL CLSDATA
LCALL DISDATA
LJMP DISKSCAN
```

```
UPDATA: MOV DPTR,#ENDDH
MOVX A,@DPTR
CJNE A,BUFDPH,UPDATA2
MOV DPTR,#ENDDL
MOVX A,@DPTR
CJNE A,BUFDPL,UPDATA2
LJMP DISKSCAN
```

```
UPDATA2: MOV DPH,BUFDPH
MOV DPL,BUFDPL
MOV R2,#00H
MOV R3,#06H
LCALL DPADD
MOV BUFDPH,DPH
MOV BUFDPL,DPL

MOV DPTR,#ENDDH
MOVX A,@DPTR
CJNE A,BUFDPH,UPDATA1
```

```

MOV    R4,#91H
LCALL  DISHOUR

MOV    A,#3AH
LCALL  LCDWD  ;DISPLAY COLON

MOV    R4,#94H
LCALL  DISMIN

```

```

MOV    DPH,DV1      ; DISDATA
MOV    DPL,DV2
LCALL  HEXTOBCD
MOV    D1,BCD3
MOV    D2,BCD4
MOV    D3,BCD5
MOV    A,#98H
LCALL  VDISP

MOV    DPH,DI1      ; DISDATA
MOV    DPL,DI2
LCALL  HEXTOBCD
MOV    D1,BCD3
MOV    D2,BCD4
MOV    D3,BCD5
MOV    A,#9CH
LCALL  IDISP
RET

```

***** IDISP DPH,DPL,D1,D2,D3,A ADD *****

```

IDISP:      LCALL  LCDWI

MOV    A,D1
CJNE   A,#00H,IDISP1
MOV    A,#20H
LCALL  LCDWD

```

```

MOV DPTR,#ENDDL
MOVX A,@DPTR
CJNE A,BUFDPL,UPDATA1
MOV A,#90H
LCALL CLSDATA
LCALL PGEND1

LJMP DISKSCAN

```

```

UPDATA1: MOV A,#90H
LCALL CLSDATA
LCALL DISDATA
LJMP DISKSCAN

```

*****DISDATA*****

```

DISDATA: MOV DPH,BUFDPH
MOV DPL,BUFDPL
MOVX A,@DPTR ;READ TM
MOV TM,A
INC DPTR
MOVX A,@DPTR ;READ TH
MOV TH,A
INC DPTR
MOVX A,@DPTR ;READ DV1
MOV DV1,A
INC DPTR
MOVX A,@DPTR ;READ DV2
MOV DV2,A
INC DPTR
MOVX A,@DPTR ;READ DI1
MOV DI1,A
INC DPTR
MOVX A,@DPTR ;READ DI2
MOV DI2,A

```

```

LJMP IDISP2

IDISP1: ANL A,#0FH
        ORL A,#30H
        LCALL LCDWD

IDISP2: MOV A,D2
        CJNE A,#00H,IDISP3
        MOV A,#20H
        LCALL LCDWD
        LJMP IDISP4

IDISP3: ANL A,#0FH
        ORL A,#30H
        LCALL LCDWD
        ; MOV A,2EH ; DISPLAY DOT
        ; LCALL LCDWD

IDISP4: MOV A,D3
        ANL A,#0FH
        ORL A,#30H
        LCALL LCDWD
        RET

;*****GET DATA *****

GETDATA: MOV A,TM
        CJNE A,#01H,GETDATA1 ;TIME GET DATA
        CLR DATAGET45
        JB DATAGET00,GETDATA5
        SETB DATAGET00
        LJMP GETDATA4

GETDATA1: CJNE A,#02H,GETDATA2 ;TIME GET DATA

```

```
CLR    DATAGET00
JB     DATAGET15,GETDATA5
SETB   DATAGET15
LJMP   GETDATA4
```

```
GETDATA2:  CJNE   A,#03H,GETDATA3      ;TIME GETDATA
CLR       DATAGET15
JB       DATAGET30,GETDATA5
SETB     DATAGET30
LJMP     GETDATA4
```

```
GETDATA3:  CJNE   A,#04H,GETDATA5      ;TIME GETDATA
CLR       DATAGET30
JB       DATAGET45,GETDATA5
SETB     DATAGET45
LJMP     GETDATA4
```

```
GETDATA5:  RET
```

```
GETDATA4:  LCALL  INCURRENT
LCALL  INVOLT
MOV     DPTR,#ENDDH
MOVX    A,@DPTR
MOV     R0,A
MOV     DPTR,#ENDDL
MOVX    A,@DPTR
MOV     BUFDPH,A
MOV     BUFDPH,R0      ;START WRITEDATA
MOV     DPL,BUFDPH
MOV     DPH,BUFDPH
```

```
MOV     A,TH           ;KEEP TIME
MOVX    @DPTR,A
INC     DPTR
```

```
MOV     A,TH           ;KEEP HOUR
MOVX    @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC    DPTR

MOV    BUFDPH,DPH
MOV    BUFDPL,DPL
MOV    D1,V1
MOV    D2,V2
MOV    D3,V3
LCALL  BCDTOHEX
MOV    DV1,HEXH
MOV    DV2,HEXL
MOV    DPH,BUFDPH
MOV    DPL,BUFDPL
MOV    A,DV1
MOVX   @DPTR,A           ;KEEP DV1
INC    DPTR
MOV    A,DV2
MOVX   @DPTR,A           ;KEEP DV2
INC    DPTR

MOV    BUFDPH,DPH
MOV    BUFDPL,DPL
MOV    D1,I1
MOV    D2,I2
MOV    D3,I3
LCALL  BCDTOHEX
MOV    DI1,HEXH
MOV    DI2,HEXL
MOV    DPH,BUFDPH
MOV    DPL,BUFDPL
MOV    A,DI1
MOVX   @DPTR,A           ;KEEP DI1
INC    DPTR
MOV    A,DI2
MOVX   @DPTR,A           ;KEEP DI2

INC    DPTR

```



```
MOV    BUFDPH,DPH
MOV    BUFDPL,DPL
MOV    DPTR,#ENDDH
MOV    A,BUFDPH
MOVSX  @DPTR,A
```

```
MOV    DPTR,#ENDDL
MOV    A,BUFDPL
MOVSX  @DPTR,A
```

```
RET
```

```
*****RUN PROGRAM*****
```

```
PGCHECK:  MOV    A,PGRUN
           LCALL  PGSTA
```

```
PGCHECK3: MOV    DPH,BUFDPH
           MOV    DPL,BUFDPL
```

```
PGCHECK5: MOVSX  A,@DPTR
           CJNE   A,#0FFH,PGCHECK1
           RET
```

```
PGCHECK1: MOV    D1,A
           INC    DPTR
           MOVSX  A,@DPTR
           MOV    D2,A
           CJNE   A,TH,PGCHECK4
           MOV    A,D1
           CJNE   A,TM,PGCHECK4
```

```
INC    DPTR
MOVSX  A,@DPTR
MOV    DV1,A
INC    DPTR
MOVSX  A,@DPTR
MOV    DV2,A
```

```

MOV    BUFDPH,DPH
MOV    BUFDPL,DPL

MOV    DPH,DV1
MOV    DPL,DV2
LCALL  HEXTOBCD
MOV    V1,BCD3
MOV    V2,BCD4
MOV    V3,BCD5
LCALL  OUTVOLT      ;SEND PG VOLT

```

```

MOV    DPH,BUFDPH
MOV    DPL,BUFDPL
INC    DPTR
MOVX   A,@DPTR
CJNE   A,#0FFH,PGCHECK2
MOV    VS1,V1
MOV    VS2,V2
MOV    VS3,V3
RET

```

PGCHECK2:

```

INC    DPTR
INC    DPTR

MOVX   A,@DPTR
MOV    DV1,A
INC    DPTR
MOVX   A,@DPTR
MOV    DV2,A
MOV    DPH,DV1
MOV    DPL,DV2
LCALL  HEXTOBCD
MOV    VS1,BCD3
MOV    VS2,BCD4
MOV    VS3,BCD5

```

RET

```
PGCHECK4:  ; MOV  BUFDPL,DPH
            ; MOV  BUFDPL,DPL
            INC  DPTR
            INC  DPTR
            INC  DPTR
            LJMP PGCHECK5
```

***** INVI *****

```
INVI:      MOV  A,TM
            CJNE A,TBUF,INVI1
            RET
INVI1:     LCALL INVOLT
            LCALL INCURRENT
            MOV  TBUF,TM
            RET
```

***** MAIN *****

```
MAIN:      MOV  A,#00001100B
            LCALL LCDWI
            LCALL CLS
```

```
MAIN1:     LCALL MENU1
```

```
MAIN2:     MOV  R4,#0D1H
            LCALL TDISP
            LCALL INVI
            LCALL GETDATA
```

```
JNB  RUNPG,GO6      ;RUN PROGRAM
LCALL PGCHECK
```

```
GO6:      MOV  D1,V1
            MOV  D2,V2
            MOV  D3,V3
            MOV  A,#085H
```

```

LCALL VDISP

MOV D1,VS1
MOV D2,VS2
MOV D3,VS3
MOV A,#08CH
LCALL VDISP ;VOLT DISPLAY

```

```

LCALL CALPOW ;V*I

```

```

MOV A,#0CBH
LCALL PDISP ;SET POINT POWER

```

```

MOV D1,V1 ;REAL POWER
MOV D2,V2
MOV D3,V3
LCALL CALPOW
MOV A,#0C4H
LCALL PDISP

```

```

MOV D1,I1
MOV D2,I2
MOV D3,I3
MOV A,#9AH
LCALL IDISP

```

```

LCALL SCANKEY
MOV A,B
CJNE A,#00H,GO
LJMP MAIN2

```

```

GO:      LCALL CLS
GOM:     LCALL MENU2
         LCALL SCANKEY
         MOV A,B

```

```

CJNE  A,#00H,G01
LJMP  GOM

GO1:   CJNE  A,#01H,G02
      LCALL  CLS
      LJMP  SETTIME

GO2:   CJNE  A,#02H,G03
      LCALL  CLS
      LJMP  PROG

GO3:   CJNE  A,#04H,G04
      LCALL  CLS
      LJMP  MAN

GO4:   CJNE  A,#08H,G05
      LCALL  CLS
      LJMP  DISPLAY

GO5:   CJNE  A,#10H,G0
      LCALL  CLS
      LJMP  MAIN

```

*****DATA TABLE*****

```

BOTTOM:  DB    0FEH,00H,01H,0FEH,02H,03H,0FEH,04H,0FEH,0FEH,05H,
          0FEH,4FH,4BH,2EH

RUN:      DB    52H,55H,4EH

EDIT:     DB    45H,44H,49H,54H

PGENDMENU: DB    2AH,2AH,2AH,0FEH,45H,4EH,44H,0FEH,2AH,2AH,2AH

AUTOMENU1: DB    41H,55H,54H,4FH

MANMENU1:  DB    4DH,41H,4EH

PGMENU1:   DB    50H,47H

MENU1R1:   DB    56H,6FH,6CH,74H,0FEH,0FEH,0FEH,0FEH,56H,7EH,0FEH,
          0FEH,0FEH,0FEH,0FEH,56H

MENU1R2:   DB    50H,6FH,77H,0FEH,0FEH,0FEH,0FEH,4BH,57H,7EH,0FEH,

```

0FEH,0FEH,0FEH,4BH,57H

MENU1R3: DB 43H,75H,72H,72H,65H,6EH,74H,0FEH,0FEH,0FEH,0FEH,
0FEH,0FEH,0FEH,0FEH,41H

MENU1R4: DB 4DH,45H,4EH,55H

MENU21: DB 44H,69H,73H,70H,6CH,61H,79H,0FEH,4DH,65H,6EH,75H

MENU22: DB 53H,45H,54H,0FEH,50H,52H,4FH,0FEH,4DH,41H,4EH,
0FEH,44H,49H,53H,0FEH,44H

MENU2SETT: DB 53H,45H,54H,0FEH,54H,49H,4DH,45H

MENU2PROG: DB 50H,52H,4FH,47H,52H,41H,4DH

MENU2MAN: DB 4DH,41H,4EH,55H,41H,4CH

MENU2DAT: DB 44H,41H,54H,41H,0FEH,44H,49H,53H,50H,4CH,41H,59H

MENU4R2: DB 56H,6FH,6CH,64H,0FEH,0FEH,0FEH,0FEH,0FEH,56H,
7EH,0FEH,0FEH,0FEH,0FEH,56H

MENU4R3: DB 50H,6FH,6CH,64H,0FEH,0FEH,0FEH,0FEH,4BH,57H,7EH,
0FEH,0FEH,0FEH,4BH,57H

MENU6R1: DB 0FEH,50H,52H,4FH,47H,52H,41H,4DH

MENU6R2: DB 0FEH,0FEH,54H,49H,4DH,45H,0FEH,0FEH,0FEH,0FEH,0FEH,0FEH,56H

MENU6R3: DB 0FEH,00H,01H,0FEH,02H,03H,0FEH,04H,0FEH,0FEH,
05H,0FEH,45H,44H,49H,54H

MENU7R1: DB 0FEH,44H,41H,54H,41H,0FEH,44H,49H,53H,50H,4CH,41H,59H

MENU7R2: DB 0FEH,54H,49H,4DH,45H,0FEH,0FEH,0FEH,56H,0FEH,
0FEH,0FEH,49H,0FEH

END

กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ เกียรติวรรณ ทรงสัจย์ ที่ให้คำแนะนำสั่งสอนและช่วยเหลือเรามาโดยตลอด อาจารย์ วันชัย ธีรวัชรา ที่คอยตักเตือนและให้ความกระจ่างในสิ่งที่สงสัยในยามวิกาลจนทำให้โครงงานนี้สำเร็จลงได้

ขอขอบพระคุณอาจารย์ทุกท่านที่อบรมสั่งสอนให้ความรู้พวกเรามาตลอด

ขอขอบคุณเพื่อนๆทุกท่านที่โดยเฉพาะอย่างยิ่ง แจน เจี๊บบ กุ้งดำ) เกียรติ ที่คอยช่วยเหลือให้กำลังใจและตักเตือนจนทำให้เราสามารถทำงานนี้สำเร็จได้

ขอขอบคุณน้องๆทุกคนที่เอาใจช่วยเสมอมา

และสุดท้ายขอขอบพระคุณคุณพ่อคุณแม่ที่เลี้ยงดูและส่งเสริมให้เรียนมาจนถึงปัจจุบัน



เอกสารอ้างอิง

1. บริษัท ศิลาเรีเสิร์ช จำกัด, “ANT-32 V2.0 USER ‘S MANUAL”,บริษัท ศิลาเรีเสิร์ช, จำกัด ,39 หน้า
2. บริษัท ศิลาเรีเสิร์ช จำกัด, “DMC XXX USER’S MANUAL”,บริษัท ศิลาเรีเสิร์ช จำกัด ,25 หน้า
3. บัณฑิต จามรติ , “คู่มือการใช้โปรแกรม OrCAD/SDT & PROTEL”, บริษัท ซีเ็ด ยูเคชั่น จำกัด ,190 หน้า,2536
4. สุเจตน์ จันทรังษ์, “ไมโครคอนโทรลเลอร์ชิพเดี่ยว 8051 ”, วิทยาลัยมหานคร, 178 หน้า,2535
5. INTEL , “MCS[®]- 51 PROGRAMMER’S GUIDE AND INSTRUCTION SET”, Intel Corporation , 1980