



เครื่องควบคุมความต้องการพลังงานไฟฟ้า

( DEMAND CONTROLLER )

1. นาย ชีรัช เตชานันท์ศิลป์ รหัส 34161108
2. นาย วิรัช เหล่าหงศ์ไพศาล รหัส 34161128



อาจารย์ที่ปรึกษา

อาจารย์ วิทยา หิพย์สุวรรณพร

สาขาวิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกวนนำไปใช้

ปริญญาโท ปีการศึกษา 2535

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขาวิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

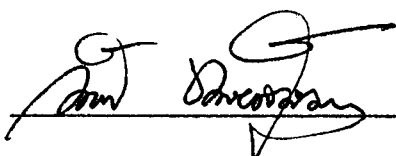
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง



1. นาย ธีรชัย เตชานันท์ศิลป์ รหัส 34161108

2. นาย วิรัช เหล่าพงศ์ไพศาล รหัส 34161128

  
อาจารย์ที่ปรึกษา  
( อาจารย์ วิทยา หิพย์สุวรรณพร )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032751

## เครื่องควบคุมความต้องการพลังงานไฟฟ้า

1. นาย ธีรชัย เตชานันทศิลป์
2. นาย วีริช เหล่าพงค์ไพศาล

อาจารย์ วิทยา ทิพย์สุวรรณพร

อาจารย์ที่ปรึกษา

บทคัดย่อ

เครื่องควบคุมความต้องการพลังงานไฟฟ้านี้เป็นส่วนหนึ่งของบริษัทอินทร์ระดับปริญญาตรี ซึ่งได้ออกแบบมาเพื่อใช้ควบคุมค่าความต้องการพลังงานไฟฟ้าสูงสุดที่มีค่าตามต้องการโดยใช้เทคนิคการคาดการณ์แบบเชิงเส้น ( LINEAR PREDICTION ) มาช่วยในการตัดสินใจในการตัดต่ออุปกรณ์ไฟฟ้าซึ่งมีอยู่ 8 ช่อง 9 ที่มีค่าความต้องการพลังงานไฟฟ้าสูงสุดอยู่ในช่วงที่ตั้งไว้

เครื่องควบคุมความต้องการพลังงานไฟฟ้าประกอบด้วยส่วนสำคัญ 2 ส่วนคือ ไมโครคอนโทรลเลอร์เบอร์ 8031 และไมโครคอมพิวเตอร์ โดยไมโครคอนโทรลเลอร์จะรับสัญญาณพัลส์จากมิเตอร์กีโรวัตต์-ชั่วโมง ซึ่งติดตั้ง ENCODER ไว้ข้างใน จากนั้นจะส่งค่าจำนวนพัลส์ต่อเวลาที่ไปให้แก่ ไมโครคอมพิวเตอร์ทุกๆ 1 นาทีทางพอร์ตอนุกรม RS-232 ซึ่งไมโครคอมพิวเตอร์จะนำค่าดังกล่าวมาคำนวณเป็นค่าพลังงานไฟฟ้าที่ใช้ไป และจะทำการคาดการณ์ค่าความต้องการพลังงานไฟฟ้าที่จะเกิดขึ้น เมื่อครบค่าเวลา 15 นาทีจะส่งสัญญาณไปที่ไมโครคอนโทรลเลอร์ เพื่อทำการตัดต่ออุปกรณ์ไฟฟ้าโดยอัตโนมัติ เมื่อค่าที่คาดการณ์ไว้มีค่าเกินหรือต่ำกว่าค่าความต้องการพลังงานไฟฟ้าที่ตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DEMAND CONTROLLER

MR. THEERACHAI DAECHANANTASIN

MR. VIRARACH LAUPONGPAISAN

MR. VITTAYA TIPSUVANPORN

ADVISOR

### ABSTRACT

This "Demand Controller" is a thesis submitted in partial of the requirements for the degree of bachelor of engineering. It is designed for controlling maximum energy consumption demand that can control the desired vaule by using "Lineer Prediction Technics" for deciding automatic loads switching which has 8 channels.

The demand controller consists of microcontroller 8031 and microcomputer. The microcontroller will recieved electric energy varied pulse signal which sent from encoder packed in kilowatt-hour meter then send number of pulse per minute to microcomputer every minute by serial port RS-232. The microputer will predict demand of energy consumption every minute. If demand vaule is more or less than the desired value, the microcomputer will feedback signal to microcontroller for automatic loads switching.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

บทคัดย่อ	
บทนำ	1
องค์ประกอบทาง Hardware	10
MCS-51 ไมโครคอนโทรลเลอร์	14
V-Microcontroller	37
Software 'Demand Controller' (MCS-51)	48
คู่มือการใช้งาน	57
Software 'Demand Controller' (QuickBASIC)	65
บรรณานุกรม	
กิตติกรรมประกาศ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

วัตถุประสงค์ของโครงการ

ไฟฟ้าใหม่ เป็นปัจจัยสำคัญในการดำรงชีวิตประจำวัน เนื่องจากไฟฟ้าเป็นพลังงานที่หาได้ง่าย สามารถเปลี่ยนเป็นพลังงานรูปอื่น เช่นพลังงานกล พลังงานความร้อน พลังงานแสงสว่างได้ง่าย ดังนั้นจึงหาข้อัอัตราความต้องการปริมาณไฟฟ้าเพิ่มสูงขึ้นเรื่อยๆ แน่นอนเมื่อมีความต้องการพลังงานไฟฟ้าเพิ่มขึ้นก็จำเป็นต้องมีหน่วยงานที่รับผิดชอบคือการผลิตไฟฟ้า ฝ่ายผลิตจะต้องคอยตอบสนองอย่างเพียงพอ แต่เนื่องจากการลงทุนทางด้านการผลิตกระแสไฟฟ้าจะต้องใช้เงินเป็นจำนวนมาก อีกทั้งใช้เวลานานการก่อสร้างโรงงานไฟฟ้าแต่ละโรงงานนานมาก ถ้าเราใช้ไฟฟ้าอย่างมีประสิทธิภาพไม่เพียงพอจะหาให้มีพลังงานสูญเสียเปล่าจำนวนมาก หรืออีกนัยหนึ่งคือการใช้ไฟฟ้าไม่สม่ำเสมอจะหาให้การผลิตไฟฟ้าให้พอใช้งานช่วงเวลาที่มีการใช้ไฟฟ้าพร้อมกัน ตัวอย่างเช่น ฝนช่วงหัวค่ำของแต่ละวันจะมีปริมาณการใช้ไฟฟ้าสูงสุด แต่ในช่วงเวลาอื่นจะมีปริมาณการใช้ไฟฟ้าที่ต่ำกว่า จึงตกเป็นภาระของการไฟฟ้าฝ่ายผลิตที่จะต้องสร้างโรงไฟฟ้าที่มีขนาดใหญ่เพียงพอที่จะรับกับความต้องการพลังงานไฟฟ้าสูงสุดจนแต่ละวัน จึงเป็นสาเหตุให้การไฟฟ้าฝ่ายผลิตต้องเรียกเก็บเงินเพิ่มจากผู้ซื้อพลังงานไฟฟ้าไม่เท่ากัน ผลิตโรงงานหรือธุรกิจขนาดใหญ่ที่มีการใช้พลังงานไฟฟ้ามากตั้งแต่ 30 KWขึ้นไป นอกจากจะต้องเสียค่าพลังงานไฟฟ้าแล้ว ยังจะต้องเสียค่าแพนเจอร์แพดเดอร์และค่าธรรมเนียมความต้องการพลังงานไฟฟ้า ( DEMAND CHARGE) อีกด้วย ค่าความต้องการพลังงานไฟฟ้านี้จะคิดจากพลังงานไฟฟ้าช่วงที่ใช้งาน เป็นกิโลวัตต์เฉลี่ยจน 15 นาทีที่สูงสุดในรอบเดือน

สาเหตุที่หาให้ค่าความต้องการพลังงานไฟฟ้ามีค่าสูงกว่าที่ควรจะเป็น คือ การใช้งานไฟฟ้าไม่สม่ำเสมอหรือการใช้งานไฟฟ้าสูงผิดปกติเพียง 15 นาที ก็จะทำให้ค่าพลังงานไฟฟ้าเดือนนั้นสูงขึ้นแล้ว ดังนั้นการควบคุมค่าความต้องการพลังงานไฟฟ้าให้สม่ำเสมอ ไม่มีค่าสูงผิดปกติชั่วระยะเวลาสั้นๆก็จะช่วยลดค่าความต้องการพลังงานไฟฟ้า ( DEMAND CHARGE ) และค่าไฟฟ้าได้มาก

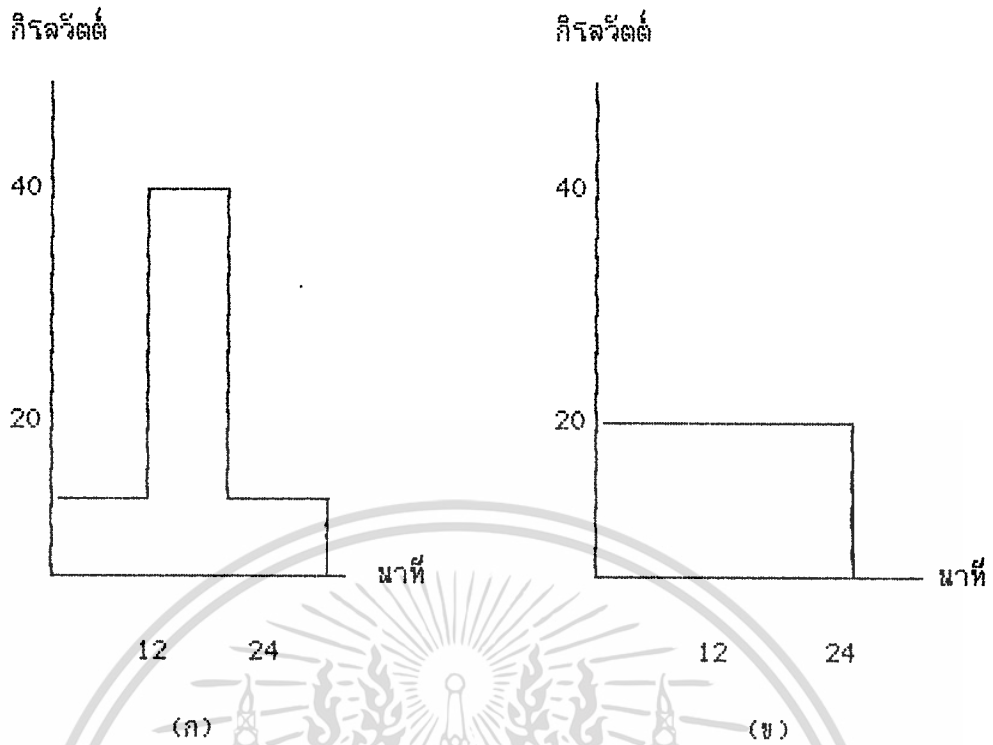
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ระวังกรณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

## เส้นกราฟของโหลด ( LOAD CURVE )

คือ เส้นกราฟที่แสดงค่าของโหลดที่เวลาต่างกัน มีแกนอนเป็นแกนเวลา และแกนตั้งเป็นค่าเฉลี่ยของปริมาณกำลังไฟฟ้าที่ใช้งานช่วงเวลา 15 นาที มีหน่วยเป็นกิโลวัตต์ เส้นกราฟของโหลดอาจได้มาโดยการนำมาตรวัดกำลังไฟแบบบันทึกค่าได้

ความหมายของการปรับปรุงค่าตัวประกอบการจ่ายไฟฟ้า

การปรับปรุงค่าตัวประกอบการจ่ายไฟฟ้า คือการปรับปรุงเส้นกราฟของโหลด หรือการควบคุมค่าความต้องการพลังงานไฟฟ้าสูงสุดนั่นเอง ซึ่งอาจจะอธิบายให้เข้าใจได้โดยดูเส้นกราฟของโหลด ในรูป 1. (ก) เป็นเส้นกราฟของโหลดในลักษณะการจ่ายไฟฟ้าที่แย่ที่สุด เพราะเส้นกราฟนี้มีค่าความต้องการพลังงานไฟฟ้าสูงสุดมีค่าสูงถึง 50 กิโลวัตต์ ในการจ่ายไฟฟ้าให้กับโหลดชนิดนี้จะต้องติดตั้งอุปกรณ์รับและจ่ายไฟ ( SWITCHGEAR ) ที่มีขีดจำกัดการจ่ายสูงถึง 50 กิโลวัตต์หรือสูงกว่า เส้นกราฟของรูป 1. (ข) เป็นเส้นกราฟของโหลดในอุดมคติ ซึ่งมีค่าความต้องการพลังงานไฟฟ้าสูงสุดเพียง 20 กิโลวัตต์เท่านั้น ในกรณีนี้อุปกรณ์รับและจ่ายไฟที่ติดตั้งอาจมีขีดเพียง 20 กิโลวัตต์ ถ้าโหลดเป็นเช่นนี้ตลอดไป การไฟฟ้าฝ่ายผลิตจะคิดค่าไฟฟ้าสำหรับกรณีนี้ถูกกว่ามาก



รูปที่ 1. แสดงให้เห็นกราฟของโหลดจนทั้ง 2 กรณี

การควบคุมค่าความต้องการพลังงานไฟฟ้าสูงสุดตามเป้าหมายเพื่อให้ได้ค่า  
 โหลดแพเตอร์ที่สูงสุดเท่าที่จะเป็นไปได้นั้นมีข้อสังเกตดังนี้

ก. การวัดค่ากำลังไฟฟ้าสูงสุดนั้น กรมสถานประกอบการธุรกิจและอุตสาหกรรมแต่ละแห่งจะมีการวัดค่าความต้องการพลังงานไฟฟ้าสูงสุด (Max Demand Power Meter) ซึ่งการไฟฟ้าติดตั้งสำหรับอ่านค่ากำลังไฟฟ้าสูงสุดเป็นรายเดือน และจะปรากฏจนจบเสร็จรับเงิน แต่ค่าที่อ่านได้จากมาตรวัดนี้ เราไม่รู้ว่าเกิดขึ้นในวันใด เวลาใด บางครั้งอาจจำเป็นต้องติดตั้งมาตรวัดกำลังไฟฟ้าแบบบันทึกค่าได้ ซึ่งนอกจากจะช่วย  
 ให้เห็นทราบค่ากำลังไฟฟ้าที่ใช้งานตลอดเวลา หรือ เห็นกราฟของโหลดด้วย การควบคุมค่ากำลังไฟฟ้าสูงสุดที่มีค่าต่ำสุดหาได้โดยอาศัยเห็นกราฟนี้

ข. ครอบคลุมแล้วค่ากำลังไฟฟ้าสูงสุดจะมีความสัมพันธ์โดยตรงกับปริมาณผล

ผลิต ดังนั้นการตั้งเป้าหมายควบคุมค่ากำลังไฟฟ้าสูงสุดจึงจำเป็นต้องมีสถิติ การรใช้ไฟฟ้า

จากจนบเสร็จรับเงินค่าไฟฟ้า หรือ จากเห็นกราฟของโหลดกับสถิติข้อมูล การผลิตรายวัน และเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ราวรายเดือนประกอบการพิจารณา  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

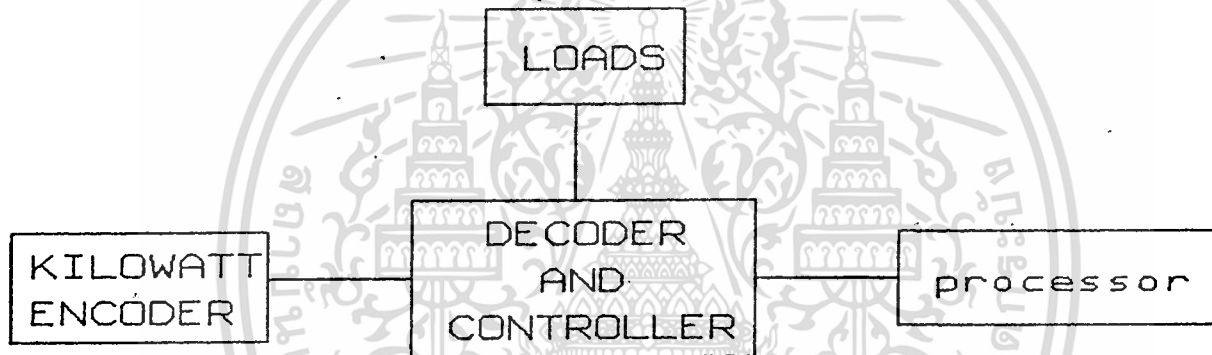


FIG 2.1 BLOCK DIAGRAM OF DEMAND CONTROLLER

Size	Document Number	REV
A	SIRICHAJ DILOGRATPIJIT	
Date:	January 1, 1980	Sheet of

อัตราค่าอนข้างสูง ดังนั้นการจัดการเกี่ยวกับการควบคุมให้ค่าความต้องการพลังงานไฟฟ้าสูงสุดในรอบเดือนมีค่าน้อยที่สุด ก็เป็นวิธีหนึ่งที่จะช่วยลดต้นทุนค่าไฟฟ้าสุทธิรายเดือนได้มากที่สุด นี่ที่จะใช้วัดประสิทธิผลของการควบคุมค่าความต้องการพลังงานไฟฟ้าสูงสุดรายเดือนได้ก็คือ ค่าตัวประกอบภาระใช้ไฟฟ้ารายเดือน นั่นเอง

**ความหมายของเทอมต่างๆที่ควรทราบ :**

**ค่าความต้องการพลังงานไฟฟ้าสูงสุด ( DEMAND MAXIMUM )**

คือ พลังงานไฟฟ้าเฉลี่ยที่ผู้ใช้ไปจนครบเวลาหนึ่งที่สูงที่สุด ในระยะเวลาคิดเงินรายคาบเวลาที่ซึ่งการวัดค่าความต้องการพลังงานไฟฟ้าสูงสุดที่การไฟฟ้านครหลวงและการไฟฟ้าฝ่ายผลิตฯคือ 15 นาที และระยะช่วงเวลาคิดเงินเป็น 1 เดือน

**ตัวประกอบภาระใช้ไฟฟ้า ( LOAD FACTOR )**

โหลดแพคเตอร์ เป็นดัชนีบ่งชี้ความสม่ำเสมอของการใช้ไฟฟ้า ซึ่งมีความว่าคืออัตราส่วนของค่าความต้องการพลังงานไฟฟ้าเฉลี่ยต่อค่าความต้องการพลังงานไฟฟ้าสูงสุดที่คิดเงินช่วงเวลา 1 วัน , 1 เดือน , 1 ปี และมีชื่อเรียกว่า โหลดแพคเตอร์รายวัน, โหลดแพคเตอร์รายเดือน, โหลดแพคเตอร์รายปี ตามลำดับ โดยค่าของโหลดแพคเตอร์จะมีค่าต่ำกว่า 100% เสมอ โหลดแพคเตอร์ที่ต่ำเกินไปคือ โหลดแพคเตอร์รายเดือน ซึ่งคำนวณได้ตามสมการดังนี้

โหลดแพคเตอร์รายเดือน

$$= \frac{\text{พลังงานไฟฟ้า(จำนวนหน่วย)} * 100}{\text{จำนวนชั่วโมง 1 เดือน} * \text{ความต้องการพลังงานไฟฟ้าสูงสุด}}$$

จำนวนชั่วโมง 1 เดือน \* ความต้องการพลังงานไฟฟ้าสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค. การควบคุมค่ากำลังไฟฟ้าสูงสุดนั้น ผู้ทำงานจำเป็นต้องเข้าใจระบบของการผลิต และต้องทราบข้อมูลโดยประมาณของอุปกรณ์ไฟฟ้า เช่น แบบ ขนาด ลักษณะ การใช้งาน ( เช่น เดินเครื่องเป็นบางเวลา , เดินเครื่องเป็นระยะ , เดินเครื่องตลอดเวลา ) การควบคุมจะเป็นไปได้ก็ต่อเมื่อเราสามารถแยกวงจรย่อย และ แบ่งประเภทของโหลดออกเป็น 3 ประเภทคือ

1. โหลดที่ต้องทำงานต่อเนื่องโดยไม่สามารถตัดเอาโหลดออกได้เลย ( Essential Loads )
2. โหลดที่สามารถตัดออกได้ แต่ต้องมีการเตรียมการล่วงหน้า เช่น การจัดวิธีการทำงานใหม่ หรือ การตัดโหลดดังกล่าวออกไปได้โดยไม่กระทบกระเทือนการผลิตมากนักหรือเพียงลดความสะดวกลงบ้าง ( Curtial Loads )
3. โหลดที่สามารถจัดช่วงเวลาทำงานไปจนเวลาใดก็ได้ ( Defera ble Loads )

ง. การควบคุมค่ากำลังไฟฟ้าสูงสุดอาจทำได้ โดยการติดตั้ง เครื่องควบคุมความต้องการพลังงานไฟฟ้า ( DEMAND CONTROLLER )

**หลักการคาดการณ์เชิงเส้น ( LINEAR PREDICTION )**

ค่าความต้องการพลังงานไฟฟ้า ( DEMAND ) จะถูกคาดการณ์ทุกๆ 1 นาที (ว่าจะมีค่าเท่าไรเมื่อครบ 15 ) เพื่อจะนำใบแจ้งผลการตัดสินค้าตัด-ต่อโหลดหรือไม่ โดยทุกๆ 1 นาที เครื่องจะนำค่าพลังงานไฟฟ้าที่เข้าใบแจ้งสินค้า มารวมกับค่าพลังงานไฟฟ้า ณปัจจุบันและค่าพลังงานไฟฟ้าที่คาดการณ์ว่าจะเกิดขึ้นในอนาคต ซึ่งก็คือค่าปัจจุบันที่ขยายออกไปจนครบ 15 นาที (เพราะเราไม่มีทางรู้ว่าอนาคตเป็นอย่างไร จึงสมมุติให้เหตุการณ์เป็นเช่นปัจจุบันไปเรื่อยๆ) จากนั้นนำมาหาค่าเฉลี่ยจนช่วง 15 นาที ก็จะได้ค่า DEMAND ที่คาดการณ์ว่าจะเกิดขึ้น (เมื่อครบ 15 นาที) จากนั้นจึงนำใบแจ้งสินค้าตัด-ต่อโหลดหรือไม่อีกที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 เครื่องควบคุมความต้องการพลังงานไฟฟ้า โดยทั่วไปจะมีวิธีการตัด-ต่อ  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหลดอยู่ 2 วิธี คือแบบ PRIORITY ORDER และ CYCLIC ORDER รถยนต์ประชิดหน้ามีพนัก  
จะจัดการตัด-ต่อโหลดแบบ PRIORITY ORDER หรือตัด-ต่อโหลดตามความสำคัญนั่นเอง  
โดยมีหลักการดังนี้ คือ

ก) เครื่องจะนำค่า DEMAND ที่คาดการณ์ในแต่ละนาที่มาเปรียบเทียบกับขีดจำกัดสูงสุดและขีดจำกัดต่ำสุดของค่า DEMAND ( UPPER DEMAND LIMIT AND LOWER DEMAND LIMIT ) ซึ่งผู้ใช้งานทราบดี

ข) ถ้าค่าที่ได้มีค่ามากกว่าขีดจำกัดสูงสุด เครื่องก็จะทำการตัดโหลดที่มีความสำคัญต่ำสุด ซึ่งยังทำงาน (ON) อยู่ก่อน

ค) ถ้าค่า DEMAND ที่คาดการณ์มีค่าต่ำกว่าขีดจำกัดต่ำสุด เครื่องก็จะทำการต่อโหลดที่มีความสำคัญสูงสุด ซึ่งยัง OFF ก่อน

โดยเมื่อจะทำการตัดโหลดจะต้องดูว่าถึงเวลาต่ำสุดของการต่อ ( MINIMUM ON TIME ) หรือยัง ถ้ายังก็ยังไม่สามารถตัดโหลดนั้นออกได้ และ เมื่อจะทำการต่อโหลดจะต้องตรวจสอบก่อนว่าถึงค่าเวลาต่ำสุดของการถูกตัด ( MINIMUM OFF TIME ) หรือยัง ถ้ายังก็ไม่สามารถต่อโหลดนั้นได้ จึงได้ตรวจสอบโหลดที่มีความสำคัญสูงกว่าถัดไป ( เพราะโหลดบางอย่าง เช่น เครื่องปรับอากาศ ไม่สามารถปิดเปิดจนเวลาใกล้เคียงกันได้ เพราะจะหาอายุการใช้งานสั้นลง นอกจากนี้ถ้าโหลดถูกตัดออกนานเกินไป ก็อาจเป็นผลเสียต่อระบบการผลิตหรือการใช้งานได้ จึงได้จัดให้เครื่องมีการตรวจสอบเวลาการถูกตัดสูงสุด ( MAXIMUM OFF TIME ) โดยถ้าโหลดถูกตัดจนถึงเวลาการถูกตัดสูงสุด ก็จะทำให้เครื่องทำการต่อโหลดเข้าก็ระบบโดยอัตโนมัติ )

จากหลักการดังกล่าว หากให้เราสามารถสรุปเป็นสูตรสำหรับหาค่า DEMAND ที่คาดการณ์ไว้ ( PREDICTION DEMAND ) ดังนี้

$$\text{PREDICTION DEM} = [ \text{SUM} + \text{WATT} ( 15 - T ) ]$$

โดย SUM คือ ผลรวมของค่าพลังงานไฟฟ้าที่โหลดใช้ไปในอดีต โดยจะมี  
ค่าอยู่ณนาทีที่ 1 ถึง 15

WATT คือ ค่าพลังงานไฟฟ้าที่โหลดใช้ไปในขณะช่วง 1 นาทีปัจจุบัน

T คือ ค่าเวลาที่คิด มีค่าตั้งแต่ 1 ถึง 15 นาที

จากหลักการดังกล่าวมาแล้ว หากที่เราสามารถเขียน BLOCK DIAGRAM  
ของเครื่องควบคุมความต้องการพลังงานไฟฟ้า ( DEMAND CONTROLLER) ได้ดังรูปที่ 2.1  
โดยมีการทำงานดังนี้

KILOWATT ENDECODER จะทำหน้าที่สร้างพัลส์ที่มีความถี่แปรเปลี่ยน  
ไปตามค่าพลังงานไฟฟ้าที่โหลดใช้ไป

DECODER AND CONTROLLER จะทำหน้าที่แปลงสัญญาณพัลส์ที่  
ส่งมาจาก KILOWATT ENCODER มาเป็นค่าพลังงานไฟฟ้าที่โหลดใช้ไปในทุกๆ 1 นาที ( หน่วย เป็น กิโลวัตต์-นาที) เพื่อส่งไปประมวลผลที่ภาค PROCESSOR นอกจากนี้ยังหา  
การควบคุมโหลด ( ที่สามารถตัดออกได้โดยไม่กระทบกระเทือนมากนัก) โดยการสั่งจาก  
ภาค PROCESSOR

PROCESSOR จะนำค่าค่าพลังงานไฟฟ้าที่โหลดใช้ไปในแต่ละนาทีมาหา  
การคำนวณค่าความต้องการพลังงานไฟฟ้าใน 15 นาที ถ้ามีค่ามากกว่าขีดจำกัดสูงสุด  
(UPPER LIMIT) ที่ผู้ใช้งานทราไว้ ก็จะทำให้การส่งสัญญาณมาสั่งให้ภาค DECODER AND  
CONTROLLER ทำการตัดโหลดที่มีความสำคัญน้อยที่สุดออก แต่ถ้ามีค่าน้อยกว่าขีดจำกัด  
(LOWER LIMIT) ที่ผู้ใช้งานทราไว้ ก็ทำให้การส่งสัญญาณมาสั่งให้ภาค DECODER AND  
CONTROLLER ทำการต่อโหลดที่มีความสำคัญสูงสุด (ที่ยังไม่ได้ต่อ) ก่อน และถ้ามีค่าอยู่ระ  
หว่างขีดจำกัดสูงสุดและขีดจำกัดต่ำสุดจะเป็นสภาวะปกติ จะไม่ทำการส่งสัญญาณไปตัดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
หรือเผยแพร่ทางอื่น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

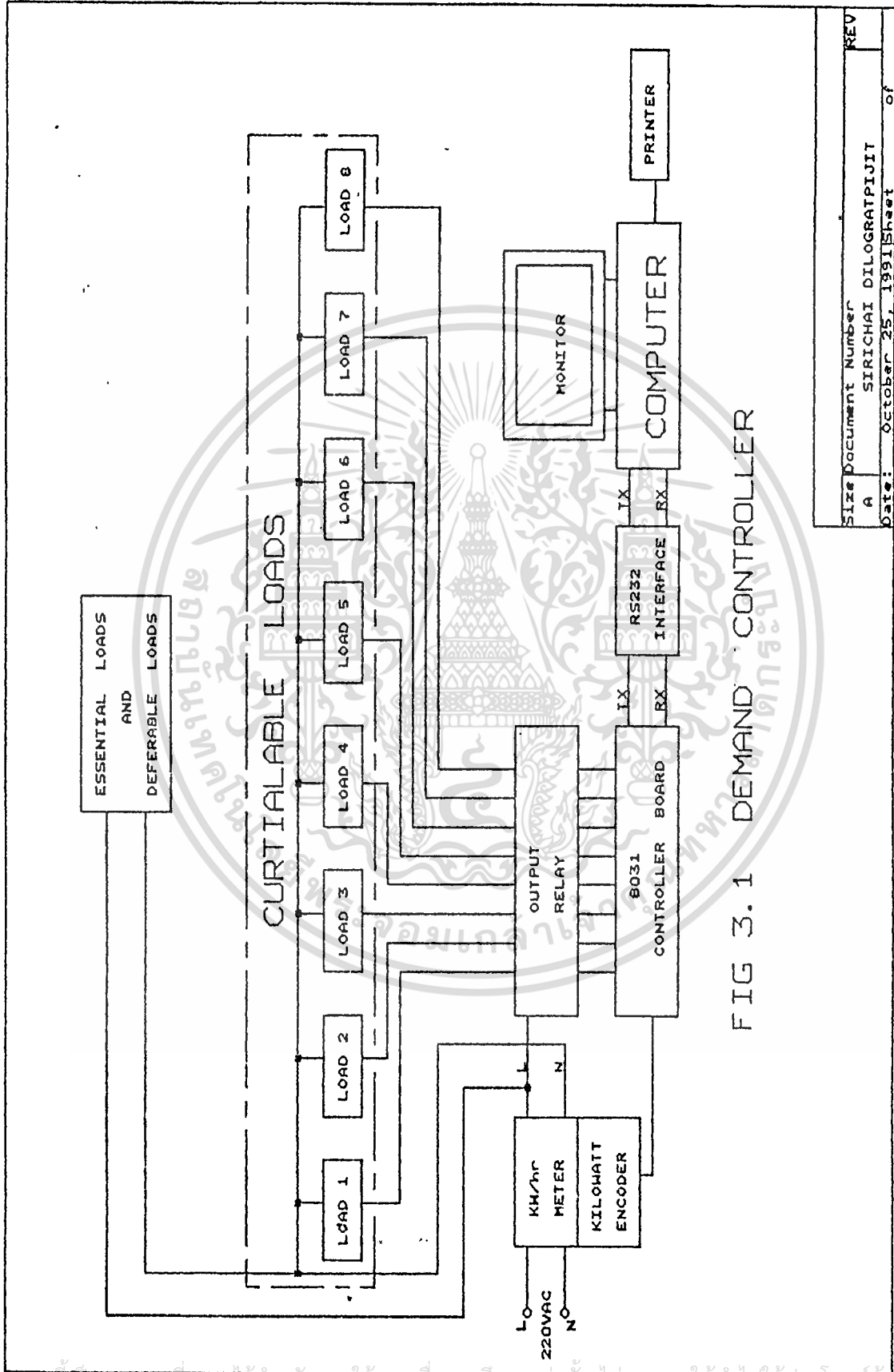


FIG 3.1 DEMAND CONTROLLER

Size Document Number	REV
A	SIRICHAI DILOGRATPIJIT
Date:	October 25, 1991 Sheet of

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## องค์ประกอบด้านฮาร์ดแวร์

จากหลักการและทฤษฎีที่ตั้งกล่าวมาแล้ว เราจึงออกแบบงานฮาร์ดแวร์ได้ดัง

นี้

### การออกแบบ KILOWATT ENCODER

โดยปกติแล้ว KW-HR METER แบบธรรมดาจะไม่สามารถสร้างสัญญาณพัลส์ที่เปลี่ยนแปลงไปตามค่าพลังงานไฟฟ้าที่จ่าย (ยกเว้นแบบพิเศษซึ่งมีราคาแพง) ดังนั้นจึงจำเป็นต้องสร้างวงจร ENCODER เพื่อสร้างสัญญาณพัลส์ที่มีความถี่เปลี่ยนแปลงไปตามค่าพลังงานไฟฟ้าที่จ่าย โดยการใช้ KW-HR METER ที่มีจานหมุน และใช้หลักการของ "OPTOCOUPLE" ซึ่งใช้อุปกรณ์ IRED (Infrared Emitting Diode) กับ PHOTO TRANSISTOR เป็นตัวหลักในการสร้างสัญญาณพัลส์โดยทำการติดตั้งลงในมิเตอร์และอยู่ที่อยู่ในแนวเดียวกับรูของจานหมุน ซึ่งลักษณะการติดตั้งและวงจรที่จ่ายแสดงดังรูป

จากรูป เมื่อรูของจานหมุนอยู่จานหมุนอยู่ตำแหน่งเดียวกับ IRED และ PHOTO TRANSISTOR ก็จะทำให้รังสีอินฟราเรดจาก IRED ส่งผ่านรูของจานหมุนเป็นพัลส์ให้ PHOTO TRANSISTOR นำกระแสได้ ซึ่งเป็นผลทำให้  $Q_2$  และ  $Q_3$  ทำงานด้วย ดังนั้นเอาต์พุตที่ถูกส่งออกไปจะมีระดับเป็น HIGH (+5v) ในทางตรงกันข้ามถ้ารูของจานหมุนไม่อยู่ตำแหน่งเดียวกับ IRED และ PHOTO TRANSISTOR  $Q_1$  ไม่ได้รับพัลส์จาก IRED ทำให้  $Q_2$  และ  $Q_3$  cutoff ดังนั้นจึงมีระดับแรงดันที่เอาต์พุตเป็น LOW จากการทำงานในลักษณะนี้ เมื่อจานหมุนหมุนไปเนื่องจากจ่ายพลังงานไฟฟ้าของโหลดก็จะทำให้เกิดสัญญาณพัลส์ที่มีความถี่เปลี่ยนแปลงไปตามพลังงานไฟฟ้าที่ไหลตบจ่ายได้

สัญญาณพัลส์ที่ได้จะถูกส่งไปเข้า COUNTER 0 ที่ขา  $T_0$  ของ 8031 CONTROLLER BOARD โดยจากคุณสมบัติของ KW-HR METER จานหมุนจะหมุนเป็นจำนวน 1200 รอบเมื่อมีการจ่ายพลังงานไฟฟ้าไป 1 กิโลวัตต์-ชั่วโมง แต่ที่จานหมุนมีรูอยู่ 2

รู ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
1 กิโลวัตต์-ชั่วโมง จะมีสัญญาณพัลส์ =  $1200 \times 2 = 2400$  พัลส์  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุผลเบื้องหลังนี้ และต้องอ้างอิงถึงเว็บไซต์นี้ทุกครั้งที่มีการนำไปใช้

จะได้ 1 กิโลวัตต์-นาทิจะมีสัญญาณพัลส์ =  $2400/60 = 40$  พัลส์

ฉะนั้น สัญญาณพัลส์ 1 พัลส์ จะมีค่าเป็น  $1/40$  กิโลวัตต์-นาทิจะมีค่าเป็น  $0.025$

กิโลวัตต์-นาทิจะมีค่าเป็น

ค่าพลังงานไฟฟ้าที่ใช้งาน 1 นาที = จำนวนพัลส์ที่ใช้งาน 1 นาที \*  $0.025$

กิโลวัตต์-นาทิจะมีค่าเป็น

### การออกแบบ DECODER & CONTROLLER BOARD

จากหน้าที่หลักของภาคนี้จะเห็นได้ว่า MCS-51 มีความเหมาะสมมากที่สุดสำหรับงานนี้ โดยจะใช้ไอซีเบอร์ 8031 เป็นตัวหลักในการทำงาน โดยจะหาหน้าที่นับจำนวนพัลส์ที่รับมาจากภาค KILOWATT ENCODER โดยจะใช้ COUNTER 0 ซึ่งอยู่ภายในตัว IC 8031 และจะหาการนับจำนวนพัลส์เป็นเวลา 1 นาที ดังนั้น IC 8031 จึงต้องการสร้างฐานเวลา 1 นาที แล้วหาการส่งค่าจำนวนพัลส์ไปยังภาค PROCESSOR โดยผ่านทางพอร์ตอนุกรม (RS-232)

หน้าที่หลักอีกอย่างของภาคนี้ คือ จะหาการรับข้อมูลจากไมโครคอมพิวเตอร์ แล้วนำมาถอดรหัสเพื่อหาการตัดต่อโหลด โดยผ่านทางเอาต์พุตที่เลือกตาม code ที่ภาค PROCESSOR ส่งมา

จากที่กล่าวมาแล้วนี้ หากให้เราสามารถออกแบบวงจรได้ ดังแสดงในรูปแบบที่ 3.3, 3.4 และ 3.5 โดยแยกเป็น 8031 CONTROLLER BOARD, RS-232 INTERFACE และ OUTPUT RELAY ตามลำดับ ส่วนการที่จะให้มันทำงานได้ตามที่เราต้องการ ก็ต้องอาศัยโปรแกรมที่ได้ออกแบบไว้

### การออกแบบ PROCESSOR

เนื่องจากหน้าที่หลักของภาคนี้ จะหาการประมวลผลค่าพลังงานไฟฟ้าที่ใช้งานในรูปแบบของค่า DEMAND และตัดสัญญาณการตัดต่อโหลด เพื่อให้ได้ค่า DEMAND อยู่ในช่วงจำกัด ( UPPER LIMIT & LOWER LIMIT ) ซึ่งกำหนดได้โดยผู้ใช้งาน ดังนั้นจึงได้ใช้ไมโครคอมพิวเตอร์มาทำงานในส่วนนี้ และหาให้เราสามารถกำหนดการทำงานได้โดยผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การพัฒนาทางซอฟต์แวร์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

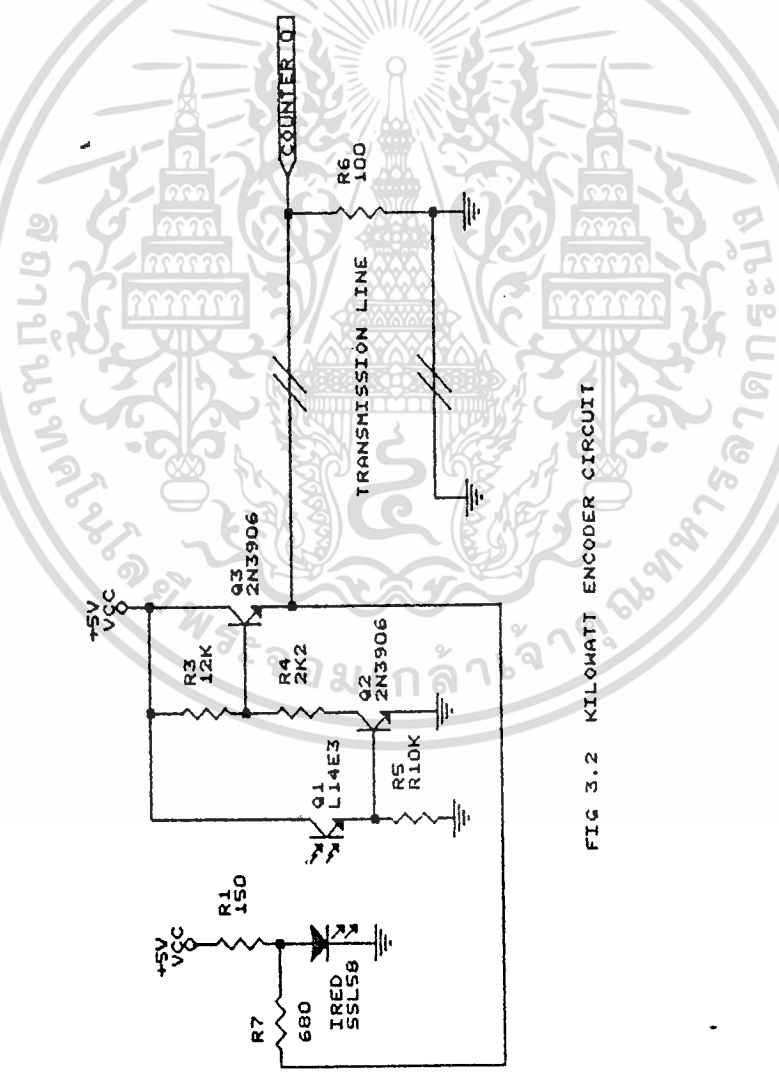
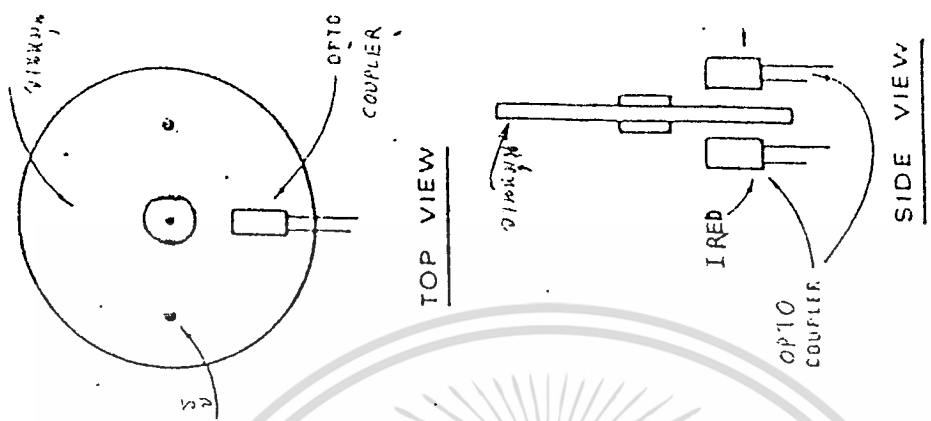


FIG 3.2 KILOWATT ENCODER CIRCUIT

Size	Document Number	REV
A	SIRICHAI DILOGRATPIJIT	
Date:	October 25, 1991	Sheet of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## mcs-51 ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว

จนถึงปัจจุบันมีไมโครคอนโทรลเลอร์ mcs-51 ซึ่งเป็นไมโครคอมพิวเตอร์แบบชิพเดี่ยว (ไม่ต้องต่ออุปกรณ์ภายนอกก็สามารถทำงานได้) มีความสะดวกในการใช้งาน และเขียนโปรแกรมควบคุมด้วยภาษาเบสิกได้ โดยไม่ต้องศึกษาการทำงานของวงจรเหมือนภาษาแอสเซมบลี หรือถ้าใช้แอสเซมบลีก็สามารถใช้งานได้เช่นกัน

mcs-51 เป็นอุปกรณ์ที่ออกแบบมาสนองความต้องการของผู้ใช้ คือมีสายอินพุทและเอาต์พุทภายในตัวเอง พอร์ตของอินพุทและเอาต์พุทมีเพอร์อินเตอร์เฟส และสายควบคุมอื่นๆที่ใช้สำหรับแลกเปลี่ยนข้อมูลกับแอดเดรส และยังมีชุดคำสั่งเพิ่มขึ้นเป็นพิเศษเพื่อจัดการข้อมูล แกรมทำด้วยวงจรรีจเวลาทั้งวงจรรีจกับด้วย (ปกติวงจรรีจจะสามารถทำงานเป็นวงจรรีจเวลาได้ด้วย จึงเรียกควบคู่กันไปคือ วงจรรีจเวลา/วงจรรีจ)

ตารางที่ 1 เป็นรายการของตระกูล mcs-51 ซึ่งแสดงถึง จำนวนหน่วยความจำ วงจรรีจเวลา/วงจรรีจ และระดับของการซิงโครไนซ์

### mcs-51 ไมโครคอนโทรลเลอร์ประกอบด้วยคุณสมบัติ

- ชิพ 8 บิต ที่ควบคุมได้ง่าย
- เพิ่มการทำงานลอจิกครั้งละ 1 บิตได้
- สายอินพุทเอาต์พุทมีจำนวน 32 เส้น
- มีแรมบัสภายในขนาด 128 ไบต์หรือ 256 ไบต์
- วงจรรีจเวลา/วงจรรีจมีขนาด 2, 3 หรือ 16 บิต
- กำหนดเป็น UART ที่สามารถส่งข้อมูลอนุกรมได้สองทิศทาง
- อินเตอร์รัพต์ แบ่งออกเป็น 2 ระดับจาก 5, 6 แหล่ง
- มีสัญญาณนาฬิกาอยู่ภายในตัว
- มีหน่วยความจำสำหรับเก็บข้อมูลภายใน ขนาด 4, 6 กิโลไบต์ (อีพ롬 8751 and 8752)

- มีแอดเดรสของหน่วยความจำสำหรับเก็บโปรแกรมจำนวนทั้งหมด 64 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของหน่วยความจำสำหรับเก็บข้อมูล จำนวนทั้งหมด 64 กิโลไบต์  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

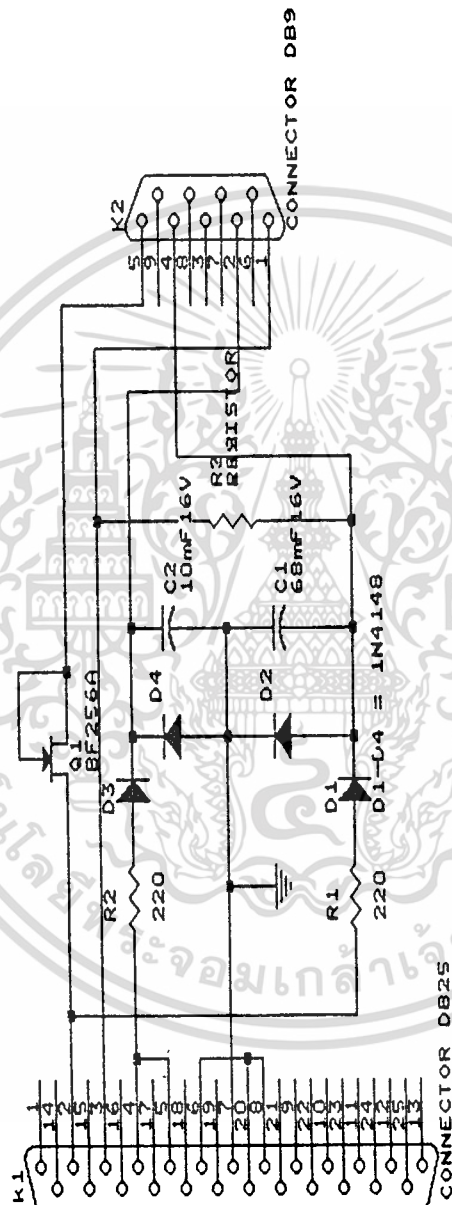
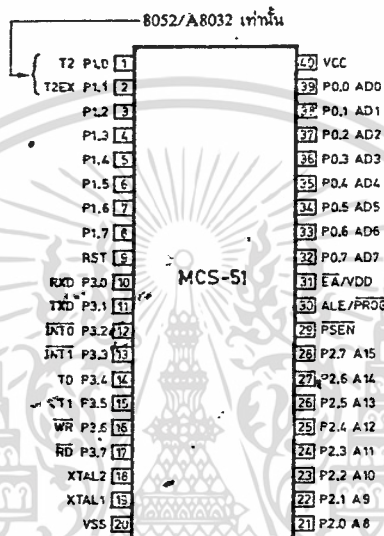


FIG 3.4 RS-232 INTERFACE CURRENT LOOP ADAPTER.

Size	Document Number	REV
A	SIRICHAI DILOGRAIPIJIT	
Date:	October 23, 1991	Sheet of

- คำสั่งทั้งหมดมี 111 คำสั่ง (64รอบ)
- ทำงานด้วยเลขฐาน 10 and 16
- ตัวแปรภาษาเบสิกมีขนาดอีกทีไรบต์(8052 AH-BASIC)
- มีคำสั่งรอยเฉพาะของภาษาเบสิกที่งใช้สำหรับอินพุตและเอาต์พุต วงจรนับและอินเตอร์เฟสแบบอนุกรม(8052 AH-BASIC)



รูปที่ 1 แสดงการจัดขาต่างๆของ MCS-51

จากรูปที่ 1 และ 2 นั้นแสดงจำนวนขา และหน้าที่ของ 8051 ที่หมายถึงตระกูล MCS-51 ทั้งหมด

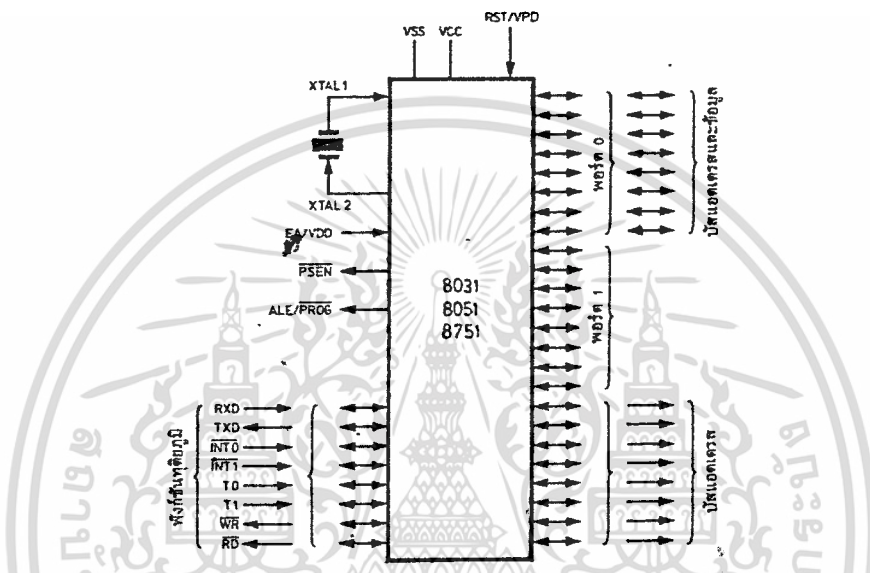
8051, 8051AH AND 8052AH เป็นอุปกรณ์ที่ทำงานโดยการควบคุมจากโปรแกรมในหน่วยความจำ(รอม) โดยการปรับปรแกรมด้วยต้นแบบ เพื่อให้เกิดข้อมูลรอมครั้งหนึ่งเป็นจำนวนมากจึงเหมาะกับการผลิตเพื่อใช้งานควบคุมจำนวนมากๆ (8051 นั้นมีหน่วยความจำรอมจำนวน 4 กิโลไบต์ส่วน 8052AH จะมีหน่วยความจำเพิ่มขึ้น โดยสร้างจาก HMOS 11)

มีอีกสองเบอร์ที่ทำงานคล้ายกันคือ 8031AH and 8032AH ที่ใช้แทน 8051 and 8052AH ได้ตามลำดับ โดยไม่ต้องส่งให้โรงงานปรับปรแกรมที่เพราะเราจะเขียนและ

เอกสารนี้ทดสอบปรแกรมด้วยหน่วยความจำภายนอกแทน (ไม่มีรอมภายใน) ไม่ว่าจะนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการการป้องกันจากภายนอก ก็สามารถเลือก 8751 and 8752 ที่มี  
อีพროมภายในที่จัดค่าเวลาในการอ่านข้อมูล เพื่อป้องกันการลอกเลียนโปรแกรม

เบอร์ที่นำเสนอมามากที่สุดสำหรับการที่ไม่ต้องศึกษา ภาษาแอสเซมบลีคือ  
8052AH BASIC แตกต่างจากเบอร์อื่นาคือ ใช้ตัวแปรภาษาเบสิกซึ่งเป็นรอมภายใน ที่เรา  
จะมาศึกษาการใช้งานของโปรแกรมต่อไป



รูปที่ 2 แสดงหน้าที่ของพอร์ตเมื่อคอนโทรลเลอร์ทำงานกับหน่วยความจำภายนอก  
หน่วยความจำภายใน

จากรูปที่ 3 เป็นหน่วยความจำภายในตัว MCS-51 หน่วยความจำนี้แบ่งได้ 2  
กลุ่ม คือ หน่วยความจำสำหรับเก็บโปรแกรม และหน่วยความจำสำหรับเก็บข้อมูล หน่วย  
ความจำแรกมีแอดเดรสที่ต่ำกว่า 4,6 กิโลไบต์ บรรจุอยู่รวม ส่วน MCS-51 ที่ใหม่  
มีรอมอยู่ภายใน จะใช้หน่วยความจำภายนอก ซึ่งอาจเป็นรอม, แรม, อีพროมแทนก็ได้

เบอร์	หน่วยความจำภายใน		คัมเวลา/บิต	อินเตอร์รัพต์ หมายเลข
	โปรแกรม	ข้อมูล		
8052AH	8K×8 ROM	256×8 RAM	3×16-Bit	6
8051AH	4K×8 ROM	128×8 RAM	2×16-Bit	5
8051	4K×8 ROM	128×8 RAM	2×16-Bit	5
8032AH	ไม่มี	256×8 RAM	3×16-Bit	6
-8031AH	ไม่มี	128×8 RAM	2×16-Bit	5
-8031	ไม่มี	128×8 RAM	2×16-Bit	5
8751H	4K×8 EPROM	128×8 RAM	2×16-Bit	5
8751H-12	4K×8 EPROM	128×8 RAM	2×16-Bit	5

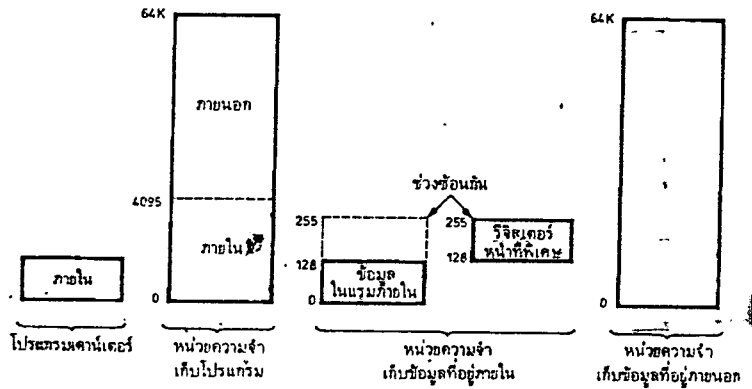
### ตารางที่ 1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

MCS จะอ่านหน่วยความจำสำหรับเก็บโปรแกรมเข้ามาเป็นภาษาเครื่องตามลำดับ ส่วนหน่วยความจำสำหรับเก็บข้อมูลจะใช้เป็นที่เก็บตัวแปร, การคำนวณทางพลวัตทันที, การจัดการกับข้อมูลที่มีขนาด 16 บิต (word), ตารางที่ใช้ค้นหาค่าต่างๆ และหน้าที่อื่นที่คล้ายกัน

#### รีจิสเตอร์ภายใน MCS

MCS มีรีจิสเตอร์ที่อ่านหน่วยความจำและการใช้งานตามคำสั่งต่างๆ ประกอบด้วยแอดเดรสยูนิเตอร์ รีจิสเตอร์ B ที่ใช้ในการคูณและหาร รีจิสเตอร์สถานะและดีทอยน์เตอร์ ตาต้าพอยน์เตอร์ (2×16 บิต 1×16 บิต) พอร์ตหมายเลขศูนย์ถึงพอร์ตหมายเลขสาม รีจิสเตอร์แบบคู่ ซึ่งใช้ในการส่งและรับข้อมูลแบบอนุกรม รีจิสเตอร์ 16 บิต ที่เป็นวงจรตั้งเวลาและวงจรถับ รีจิสเตอร์ซึ่งจองไว้สำหรับใช้สำหรับนับตัวที่ 3 รีจิสเตอร์คำสั่งสำหรับหาหน้าที่พิเศษ (เช่น การอินเตอร์รัพต์ RTC: Real Time Clock) และอินพุต, เอาต์พุตแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- INTERNAL → หน่วยความจำภายใน
- EXTERNAL → หน่วยความจำภายนอก
- PROGRAM COUNTER → วงจรนับลำดับโปรแกรม
- PROGRAM MEMORY → หน่วยความจำสำหรับเก็บโปรแกรม
- INTERNAL DATA RAM → แรมภายในที่เก็บข้อมูล
- SPECIAL FUNCTION REGISTERS → รีจิสเตอร์หน้าที่พิเศษ
- INTERNAL DATA MEMORY → หน่วยความจำภายในสำหรับเก็บข้อมูล
- EXTERNAL DATA MEMORY → หน่วยความจำภายนอกสำหรับเก็บข้อมูล

รูปที่ 3 โครงสร้างภายในหน่วยความจำของ MCS-51

สายต่างๆของบัสและพอร์ต

จะเห็นว่าโครงสร้างภายในของ MCS-51 ในรูปที่ 4 จะสมมุติว่า มีบัสสองทิศทาง 4 เส้น และพอร์ตขนาด 8 บิต ตามทฤษฎีแต่ที่จริงจะใช้ก็ต่อเมื่อมีการใช้หน่วยความจำภายในตัว (รอมหรือแรม)

เมื่อไม่ใช้หน่วยความจำภายในพอร์ต 0 และ 2 จะถูกใช้ เป็นบัสของข้อมูลและแอดเดรส (ขอใช้ชื่อกลับใบดูรูปที่ 2) ดังนั้นพอร์ต 2 พอร์ตยังคงใช้งานเป็นอินพุตและเอาต์พุต พอร์ 2 ห้าหน้าที่เป็นสายสี่สายแอดเดรส A<sub>15</sub>...A<sub>8</sub> ส่วนพอร์ต 0 ห้าหน้าที่เป็นสายสี่สายแอดเดรส A<sub>7</sub>...A<sub>0</sub> ออกจากบิตข้อมูล D<sub>7</sub>...D<sub>0</sub>

เอาต์พุตของ RD and WR มาจากสายเอาต์พุตของพอร์ต 3 โดยโปรแกรมภายในใช้สี่สาย RD and WR เพื่อการเขียนและอ่านข้อมูล กับข้อมูลของหน่วยความจำภายนอก

วงจรมีหรือวงจรตั้งเวลา

จากตารางที่ 1 จะเห็นว่า 8052 มีวงจรมีและวงจรตั้งเวลาชนิด 16

เอกสารนี้เป็นมากกว่า 8051 อยู่หนึ่งตัว ถ้าต่อไปเราจะศึกษาการทำงานอย่างละเอียดของวงจรมีและไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรถัดเวลา

เมื่อทำงาน เป็นวงจรถัดเวลา

รีจิสเตอร์ที่ทำงานเป็นวงจรถัดเวลา จะเพิ่มค่าขึ้นหนึ่งของทุกการรอบคำสั่งของเครื่อง และนับด้วยอัตราสูงสุดที่ 1/12 ของความเร็วสัญญาณนาฬิกาของโปรเซสเซอร์

เมื่อทำงาน เป็นวงจรมับ

รีจิสเตอร์ของวงจรมับจะเพิ่มขึ้นหนึ่ง เมื่อมีสัญญาณบ่อน้ำที่อินพุต T0, T1, T2 (T2 มีเฉพาะ 8052) เป็นขอบของสัญญาณขอบขาลง อัตราการนับสูงสุดคือ 1/24 ของสัญญาณนาฬิกา ของโปรเซสเซอร์

วงจรมับและวงจรถัดเวลา 0 และ 1 มีวิธีโปรแกรมงานที่ทำงานได้ต่างกันถึง 4 แบบ รวมทั้งการทำงานเป็น 8 บิต หรือ 16 บิต และการบรรจุค่าพีเชดหนึ่งค่าได้อย่างอัตโนมัติ

วงจรมับและวงจรถัดเวลาที่ 1 เลือกโปรแกรมงานที่หาหน้าที่ เป็นตัวกำเนิดสัญญาณของอัตราการส่งบิต ออกไปอย่างอนุกรม สำหรับสัญญาณอินเตอร์เฟสได้

วงจรมับและวงจรถัดเวลาที่ 2 (เฉพาะ 8052 เท่านั้น) มีการทำงานย่อยอีก 3 ชนิด

1. วงจรมับ 16 บิตที่สามารถโหลดค่ากลับคืนเองอย่างอัตโนมัติ
2. วงจรมับที่จองไว้ ชนิด 16 บิต
3. วงจรกำเนิดสัญญาณของการส่งบิต เพื่อซีอินเตอร์เฟส

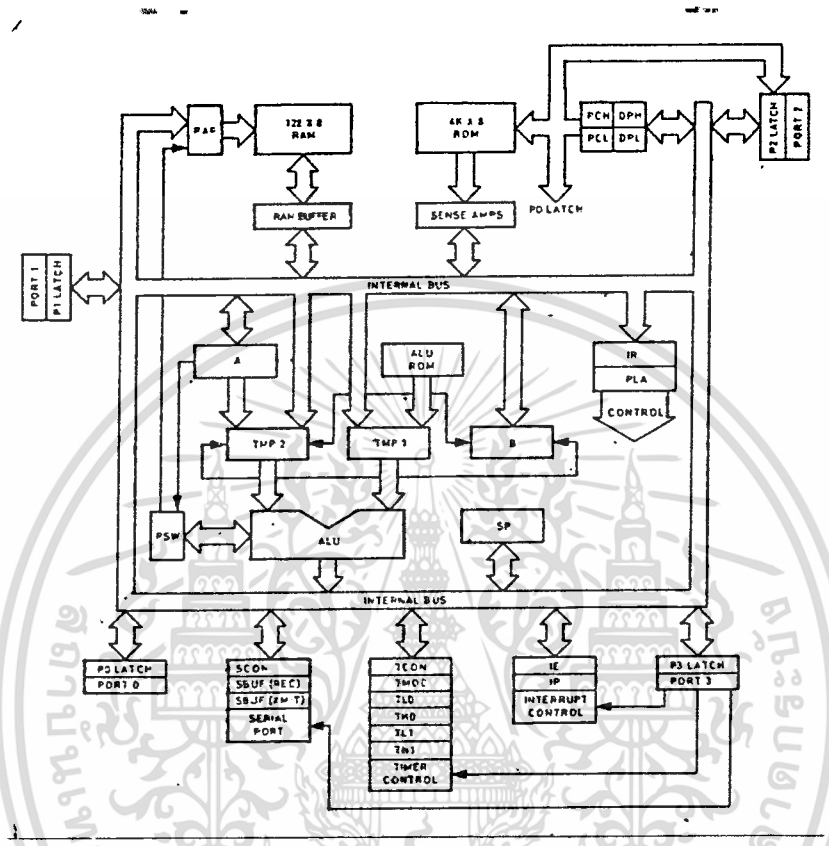
พอร์ตชนิดอนุกรมอยู่ภายใน

ไมโครคอนโทรลเลอร์ทั้งหมดในตระกูล MCS-51 เป็นชิพที่มีอินเตอร์เฟสอนุกรม ชนิดสองทิศทาง หากได้รับและส่งข้อมูลพร้อมกันได้ ตัวรับข้อมูลชนิดอะซิงโครนัส มีบัฟเฟอร์สำหรับข้อมูลเป็นพิเศษ เพื่อเพิ่มความเร็วงานการสื่อสาร

พอร์ตชนิดอนุกรมนั้น เราสามารถเลือกโปรแกรม เพื่อเลือกการทำงานแบบใดแบบหนึ่งจน 4 แบบ ด้วยการโปรแกรมควบคุมอัตราการส่งข้อมูลและรูปแบบของข้อมูล

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ในการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสัญญาณนาฬิกา 1 เมกะเฮิรตซ์ สำหรับใช้งานระบบเครือข่ายและระบบการสื่อสารของ  
 โปรเซสเซอร์หลายตัวร่วมกัน เราจะเลือกความเร็วของสัญญาณนาฬิกาด้วยวงจรนับและ  
 วงจรตั้งเวลา



รูปที่ 4 โครงสร้างของ MCS 51

**อินเตอร์รัพต์และชุดคำสั่ง**

8051 รับการอินเตอร์รัพต์ได้ 5 แห่ง ส่วน 8052 รับการอินเตอร์รัพต์  
 จากอุปกรณ์อื่น ๆ ได้ถึง 6 แห่ง คือขา INTO และขา INT1 (กำหนดให้ 9 ีระดับพ็ลล์หรือ  
 ขอบขาพ็ลล์ก็ได้) วงจรนับและวงจรตั้งเวลาที่ศูนย์และหนึ่ง (สำหรับ 8052 เพิ่มวงจรตั้ง  
 เวลา/วงจรับนับตัวที่สอง) และสุดท้ายจากพอร์ตอนุกรม

เราสามารถกำหนดระดับของความสำคัญของอินเตอร์รัพต์ได้สองระดับ โดยไม่  
 ต้องอาศัยวงจรถ่ายนอกช่วยแต่ละแห่งของอินเตอร์รัพต์ 5 หรือ 6 แห่งนั้นจะกำหนดเป็น  
 เวกเตอร์เฉพาะ (ตัวชี้แอดเดรส)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ในการค้า  
 ได้ เว้นแต่จะมีการขออนุญาตจากเจ้าของลิขสิทธิ์ก่อน มิฉะนั้นจะถือว่าผิดกฎหมาย  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของโปรแกรมที่ทำงานตามวัตถุประสงค์ของอินเทอร์พรีตนั้น หลังจากเก็บข้อมูลต่างๆ ของโปรแกรมเคอร์เนลเดอรัลงจนเสร็จแล้ว ชุดคำสั่งทั้งหมดของตัวคอนโทรลเลอร์ในตระกูล MCS แสดงในตารางที่สอง

### 8052AH - BASIC ซีพียูที่ไมโครโปรเซสเซอร์ภาษาแอสเซมบลี

ตัวนี้เป็นไมโครคอนโทรลเลอร์ที่นำเสนองานที่สุด ในตระกูล MCS 51 เนื่องจาก 9ซีพียูภาษาสำหรับโปรแกรมง่ายกว่า และมีประสิทธิภาพสูงพอที่จะใช้กับโครงการงานด้านคอมพิวเตอร์ คือภาษาเบสิก ซึ่งจะอธิบายอย่างคร่าวๆ เพื่อเป็นแนวทางในการเขียนโปรแกรมต่อไป

ตารางที่ 2 ชุดคำสั่งทั้งหมดของไมโครคอนโทรลเลอร์ 8051/8052 จากบริษัทอินเทล อินเทล ได้บรรจุรวมไว้ในชิพขนาด 8 กิโลไบต์ ครอบคลุมทั้งตัวแปลภาษาเบสิก และยังเพิ่มวงจรนับและวงจรตั้งเวลา T3 9ให้อีกตัวหนึ่ง เพื่อทำหน้าที่อินเทอร์เฟซอย่างอนุกรม ซึ่งสามารถเลือกแยกแ่งซีพียูการส่งข้อมูลได้อีกต่างหาก ชุดคำสั่งภาษาเบสิก (เวอร์ชัน 1.1) ดังรายการในตารางที่ 3 นั้นถ้าเราตั้งเกตุฯให้ละเอียดจะเห็นว่ามีความเฉพาะที่หาหน้าที่ได้แทนภาษาแอสเซมบลี หลังจากทดสอบด้วยตัวเลขและการหาค่าต่างๆ ก็สรุปได้ว่า 8052AH - BASIC ตัวนี้ เหมาะสมกับการใช้งานต่างๆ ที่ควบคุมการรับส่งข้อมูล และบันทึกข้อมูล เครื่องมือเครื่องวัด และเครื่องควบคุมงานอุตสาหกรรมได้เป็นอย่างดี

ข้อเสียของตัวแปลภาษาเบสิก คือความเร็วจะต่ำกว่าภาษาแอสเซมบลี แต่มีข้อที่เหนือกว่าคือ 9ซีพียูโปรแกรมที่คล้ายภาษาพูดจึงตรวจสอบและแก้ไขได้ง่ายแม้ผู้ใช้ไม่คุ้นเคยกับวงจรเลย ก็สามารถหาความเข้ากันได้ในการเขียนโปรแกรมควบคุมได้อย่างรวดเร็ว

### การจัดหน่วยความจำ

ในระบบของ MCS 51 จะมีการแบ่งหน่วยความจำเหมือนกับ CPU ที่ทั่วไปคือ จะแบ่งเป็นสองลักษณะตามชนิดของข้อมูลที่เก็บดังนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## -หน่วยความจำโปรแกรม

## ARITHMETIC OPERATIONS

Mnemonic	Description	Byte	Cyc
ADD	A,Rn	1	1
ADD	A,direct	2	1
ADD	A,@Ri	1	1
ADD	A,#data	2	1
ADDC	A,Rn	1	1
ADDC	A,direct	2	1
ADDC	A,@Ri	1	1
ADDC	A,#data	2	1
SUBB	A,Rn	1	1
SUBB	A,direct	2	1
SUBB	A,@Ri	1	1
SUBB	A,#data	2	1
INC	A	1	1
INC	Rn	1	1
INC	direct	2	1
INC	@Ri	1	1
DEC	A	1	1
DEC	Rn	1	1
DEC	direct	2	1
DEC	@Ri	1	1
INC	DPTR	1	2
MUL	AB	1	4
DIV	AB	1	4
DA	A	1	1

## LOGICAL OPERATIONS

Mnemonic	Destination	Byte	Cyc
ANL	A,Rn	1	1
ANL	A,direct	2	1
ANL	A,@Ri	1	1
ANL	A,#data	2	1
ANL	direct,A	3	2
ANL	direct,#data	3	2
ORL	A,Rn	1	1
ORL	A,direct	2	1
ORL	A,@Ri	1	1
ORL	A,#data	2	1
ORL	direct,A	3	2
ORL	direct,#data	3	2
XRL	A,Rn	1	1
XRL	A,direct	2	1
XRL	A,@Ri	1	1
XRL	A,#data	2	1
XRL	direct,A	3	2
XRL	direct,#data	3	2
CLR	A	1	1
CPL	A	1	1
RL	A	1	1
RLC	A	1	1
RR	A	1	1
RRC	A	1	1
SWAP	A	1	1

## DATA TRANSFER

Mnemonic	Description	Byte	Cyc
MOV	A,Rn	1	1
MOV	A,direct	2	1
MOV	A,@Ri	1	1
MOV	A,#data	2	1
MOV	Rn,A	1	1
MOV	Rn,direct	2	1
MOV	Rn,#data	2	1
MOV	direct,A	3	2
MOV	direct,Rn	3	2
MOV	direct,direct	3	2
MOV	direct,@Ri	3	2
MOV	direct,#data	3	2
MOV	@Ri,A	1	1
MOV	@Ri,direct	2	1
MOV	@Ri,#data	2	1
MOV	DPTR,#data16	3	2

## DATA TRANSFER (cont.)

Mnemonic	Description	Byte	Cyc
MOVC	A,@A+DPTR	1	2
MOVC	A,@A+PC	1	2
MOVX	A,@Ri	1	2
MOVX	A,@DPTR	1	2
MOVX	@Ri,A	1	2
MOVX	@DPTR,A	1	2
PUSH	direct	2	2
POP	direct	2	2
XCH	A,Rn	2	1
XCH	A,direct	2	1
XCH	A,@Ri	1	1
XCHD	A,@Ri	1	1

## BOOLEAN VARIABLE MANIPULATION

Mnemonic	Description	Byte	Cyc
CLR	C	1	1
CLR	bit	1	1
SETB	C	1	1
SETB	bit	1	1
CPL	C	1	1
CPL	bit	1	1
ANL	C,bit	2	2
ANL	C,#bit	2	2
ORL	C,bit	2	2
ORL	C,#bit	2	2
MOV	C,bit	2	2
MOV	bit,C	2	2

## PROGRAM AND MACHINE CONTROL

Mnemonic	Description	Byte	Cyc
ACALL	addr11	2	2
LCALL	addr16	3	2
RET		1	2
RETI		1	2
AJMP	addr11	2	2
LJMP	addr16	3	2
SJMP	rel	2	2
JMP	@A+DPTR	1	2
JZ	rel	2	2
JNZ	rel	2	2
JC	rel	2	2
JNC	rel	2	2
JB	bit,rel	3	2
JNB	bit,rel	3	2
JBC	bit,rel	3	2
CJNE	A,direct,rel	3	2
CJNE	A,#data,rel	3	2
CJNE	Rn,#data,rel	3	2
CJNE	@Ri,#data,rel	3	2
DJNZ	Rn,rel	2	2
DJNZ	direct,rel	3	2
NOP		1	1

## Notes on data addressing modes:

- Rn - Working register R0-R7
- direct - 128 internal RAM locations, any I/O port, control or status register
- @Ri - Indirect internal RAM location addressed by register R0 or R1
- #data - 8-bit constant included in instruction
- data16 - 16-bit constant included as bytes 2 & 3 of instruction
- bit - 128 software flags, any I/O pin, control or status bit

## Notes on program addressing modes:

- addr16 - Destination address for LCALL & LJMP may be anywhere within the 64-Kilobyte program memory address space.
- addr11 - Destination address for ACALL & AJMP will be within the same 2-Kilobyte page of program memory as the first byte of the following instruction.
- rel - SJMP and all conditional jumps include an 8-bit offset byte. Range is +127/-128 bytes relative to first byte of the following instruction.

All mnemonics copyrighted © Intel Corporation 1979

## หน่วยความจำโปรแกรม

งาน MCS 51 จะแบ่งหน่วยความจำโปรแกรมออกเป็นสองส่วนคือ หน่วย

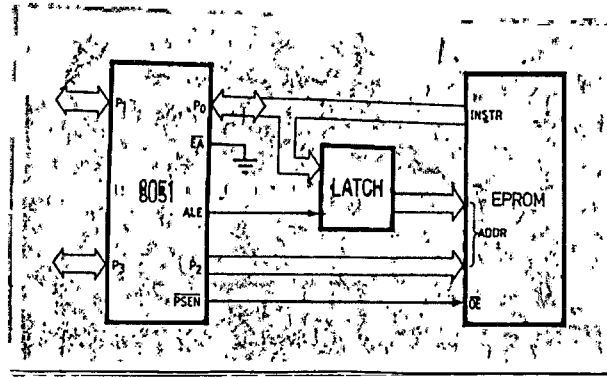
ความจำโปรแกรมภายในและหน่วยความจำโปรแกรมภายนอก หน่วยความจำโปรแกรม

ภายนอกก็คือ หน่วยความจำประเภทรอมหรืออีพรอมที่อยู่ภายในตัว MCS 51 เบอร์ 8051

หรือ 8751 ตามลำดับ ซึ่งมีขนาด 4 กิโลไบต์

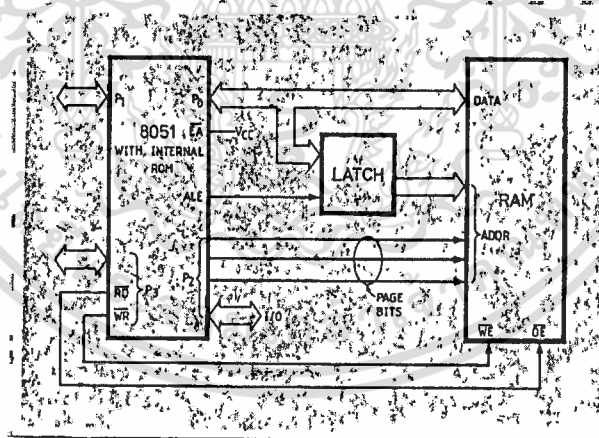
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



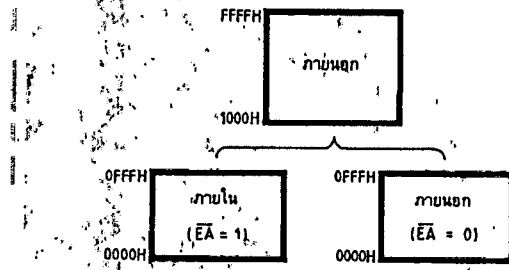
รูปที่ 1 การเชื่อมต่อกับหน่วยความจำโปรแกรม

ส่วนหน่วยความจำโปรแกรมภายนอกก็คือ หน่วยความจำที่ต่ออยู่ภายนอกตัว MCS 51 ใดๆ อาจจะเป็นการต่อเพื่อขยายหน่วยความจำเพิ่ม เนื่องจากหน่วยความจำภายในไม่พอ หรืออาจต่อเป็นหน่วยความจำโปรแกรมภายนอกทั้งหมดเลยก็ได้ 96K บิตของ 8031 เพราะภายในตัวมันไม่มีหน่วยความจำโปรแกรมเหมือนเบอร์ 8051 หรือ 8751 ใดๆ MCS 51 สามารถเลือกทั้งรับโปรแกรมหน่วยความจำโปรแกรมภายในหรือภายนอกก็ได้ โดยควบคุมที่ขา EA (ขา 31) แต่สำหรับเบอร์ 8031 จะต้องต่อขา EA ลงกราวด์เสมอ การจัดแบ่งพื้นที่หน่วยความจำโปรแกรมภายในและภายนอกแสดงในรูปที่ 3



รูปที่ 2 การเชื่อมต่อกับหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 การจัดพื้นที่หน่วยความจำโปรแกรมภายในและภายนอก

**หน่วยความจำข้อมูล**

หน่วยความจำข้อมูลของ MCS 51 ก็ถูกแบ่งออกเป็นสองส่วนเหมือนกันคือ หน่วยความจำข้อมูลภายในและภายนอก โดยหน่วยความจำข้อมูลภายนอกจะเข้าถึงได้ก็ด้วยคำสั่ง MOVX เท่านั้น สำหรับหน่วยความจำข้อมูลภายในของ 8031/8051 และ 8751 จะมีทั้งหมด 256 ไบต์ โดยแบ่งเป็น 128 ไบต์ส่วนบน ซึ่งใช้เป็นที่อยู่ของรีจิสเตอร์ฟังก์ชันพิเศษ และอีก 128 ไบต์ส่วนล่าง ซึ่งจะถูกใช้งานที่วาบ ดังรูปที่ 4 และ 5

จากรูปที่ 5 จะเห็นว่าพื้นที่หน่วยความจำข้อมูลภายในส่วน 128 ไบต์ล่าง (ตำแหน่ง 00H-7FH) จะถูกแบ่งเป็น 3 ส่วนคือ ส่วนของรีจิสเตอร์แบงค์ (00H-1FH), ส่วนพิเศษที่สามารถเข้าถึงตำแหน่งบิตได้โดยตรง (00H-2FH) และส่วนใช้งานที่วาบ (30H-7FH)

ส่วนของรีจิสเตอร์แบงค์มีทั้งหมด 4 แบงค์ แต่เราสามารถรีเซ็ตแบงค์เหล่านั้น การเลือกรีจิสเตอร์แบงค์ไหนอยู่ที่เรากำหนดค่าในรีจิสเตอร์ PSW

คำสั่งที่ใช้งานการติดต่อกับหน่วยความจำข้อมูล ซึ่งแบ่งลักษณะการกำหนดเลขที่อยู่ของหน่วยความจำข้อมูลได้ 4 รูปแบบด้วยกันคือ

**โหมดการกำหนดเลขที่อยู่รีจิสเตอร์** จะเป็นการติดต่อกับข้อมูลที่อยู่ในรีจิสเตอร์โดยตรง ตัวอย่างเช่น

```
MOV A,R0 ; ย้ายค่าใน R0 ไปไว้ใน ACC
```

```
ADD A,R3 ; บวกค่า ACC กับค่าใน R3 แล้วนำผลลัพธ์ไปไว้ใน ACC
```

**โหมดการกำหนดเลขที่อยู่โดยตรง** จะเป็นการติดต่อกับข้อมูลที่ตำแหน่งแอดเดรสที่กำหนดโดยตรง ซึ่งจะต้องอยู่ในบริเวณตำแหน่งของแรมภายใน 256 ไบต์ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สโมว A, @ R0; ; ย้ายค่าข้อมูลในหน่วยความจำที่ถูกรับไปไว้ใน ACC อันการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADD A,@ R1 ; บวกค่าข้อมูลใน ACC กับค่าข้อมูลในหน่วยความจำที่อยู่ R1  
 รวมค่าที่กำหนดเลขที่อยู่ข้อมูลในหน่วยความจำ จะเป็นการติดต่อกับข้อมูลตรงที่

อย่างเช่น

MOV A,40H ; ย้ายข้อมูลที่ตำแหน่ง 40H ไปไว้ใน ACC

ADD A,41H ; บวกค่าข้อมูลใน ACC กับข้อมูลที่ตำแหน่ง 41H

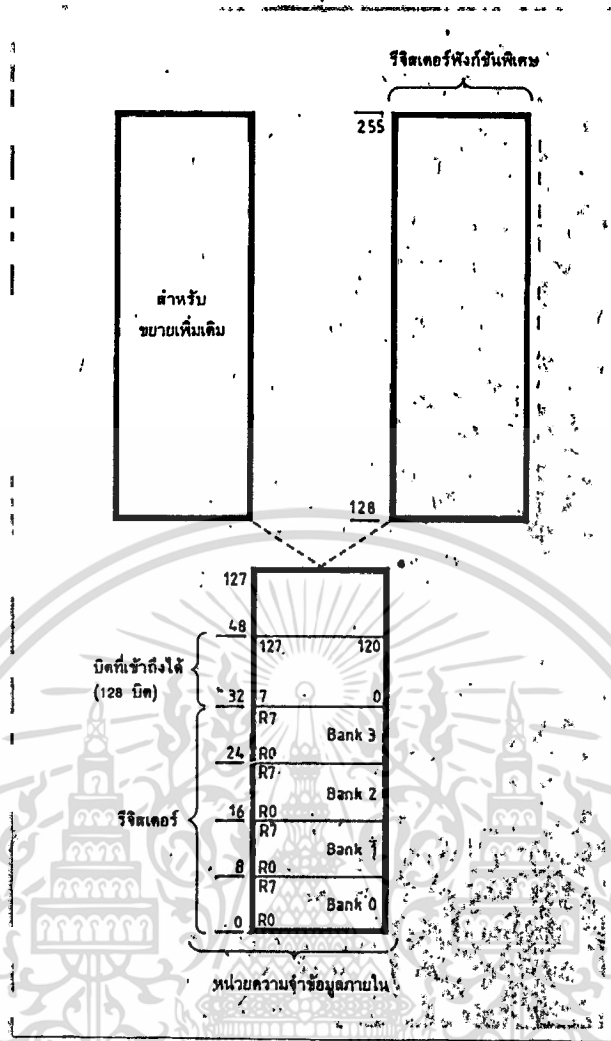
รวมค่าที่กำหนดเลขที่อยู่ข้อมูลในหน่วยความจำ จะเป็นการติดต่อกับข้อมูลที่จำค่าในตัวรีจิสเตอร์

สแต็ค R0 หรือ R1 เป็นตัวชี้ตำแหน่ง ตัวอย่างเช่น

MOV A,# 22H ; นำค่า 22H ไปไว้ใน ACC

ADD A,# 41H ; นำค่า 41H ไปบวกกับค่าใน ACC





รูปที่ 4 การจัดการหน่วยความจำข้อมูล

ตัวจับเวลา/ตัวนับ (TIMER/COUNTER)

จำนวน 8031/8051 และ 8751 จะมีตัวจับเวลา/ตัวนับขนาด 16 บิต จำนวน 2 ตัวคือ timer/counter 0 and timer/counter 1 ส่วนเบอร์ 8032/8052 จะมีเพิ่มอีก 1 ตัว โดยแต่ละตัวสามารถกำหนดการทำงานเป็นตัวจับเวลาหรือตัวนับได้ โดยการเซตหรือเคลียร์บิต C/T ที่ตัวรีจิสเตอร์ควบคุม TMOD ซึ่งอยู่ในกลุ่มฟังก์ชันพิเศษ

ตัวจับเวลา/ตัวนับ สามารถปรับปรแกรมให้มันทำงานได้ต่างกันถึง 4 รูปแบบ โดยการตั้งค่าจำนวนรีจิสเตอร์ TMOD ซึ่งจะกล่าวถึงการทำงาน ของแต่ละโหมดดังนี้

โหมด 0 รีจิสเตอร์ตัวนับจะถูกกำหนดให้มี 13 บิต ประกอบด้วยรีจิสเตอร์ TH1 8 บิต และ TL1 อีก 5 บิต อันดับต่ำ ซึ่งสามารถกำหนดให้เป็นตัวจับเวลาหรือตัวนับ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น การนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การทำซ้ำโดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะนับขึ้นครั้งละหนึ่งก็ต่อเมื่อ มีสัญญาณเข้ามา 1 ลูกและนับเป็น "1" หมดทุกบิต ก็จะกลับมาเป็น "0" หมดทุกบิตใหม่ ซึ่งจะเป็นการเกิด overflow ไปทดแฟลคอินเตอร์รัพต์ TF1 9 หักเป็น "1" ลักษณะวงจรเป็นดังรูปที่ 6

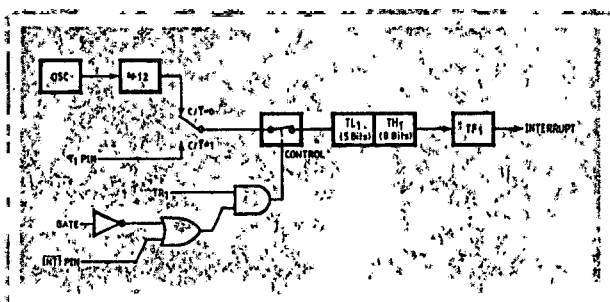
รวม 1 การทำงานจะเหมือนกับรวม 0 ทุกอย่างยกเว้นรีจิสเตอร์ตัวนับจะเป็นขนาด 16 บิต

รวม 2 จะใช้รีจิสเตอร์ TL1 เป็นตัวนับเพียงตัวเดียวและเมื่อ TL1 นับจนเป็น "1" หมดทุกบิต ก็จะมีการไหลค้ำจากรีจิสเตอร์ TH1 เข้าไปไว้จน TL1 หมดจัตนมิติ ทารัทการทดแฟลคอินเตอร์รัพต์ TF1 9 หักเป็น "1" ค่าจน TH1 นี้เราสามารถตั้งค่าได้ด้วยซอฟต์แวร์ ลักษณะวงจรแสดงจนวนรูปที่ 7

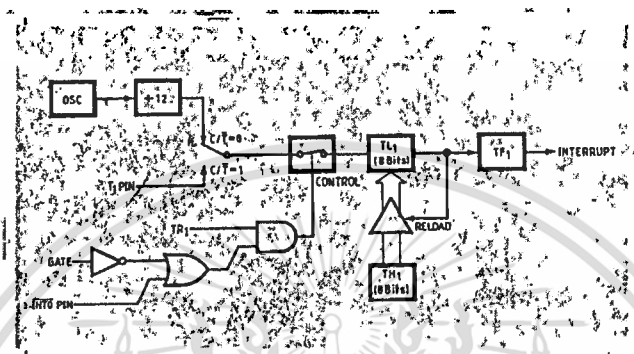
	(MSB)				(LSB)				
7FH									127
2FH	7F	7E	7D	7C	7B	7A	79	78	47
2EH	77	76	75	74	73	72	71	70	46
2DH	6F	6E	6D	6C	6B	6A	69	68	45
2CH	67	66	65	64	63	62	61	60	44
2BH	5F	5E	5D	5C	5B	5A	59	58	43
2AH	57	56	55	54	53	52	51	50	42
29H	4F	4E	4D	4C	4B	4A	49	48	41
28H	47	46	45	44	43	42	41	40	40
27H	3F	3E	3D	3C	3B	3A	39	38	39
26H	37	36	35	34	33	32	31	30	38
25H	2F	2E	2D	2C	2B	2A	29	28	37
24H	27	26	25	24	23	22	21	20	36
23H	1F	1E	1D	1C	1B	1A	19	18	35
22H	17	16	15	14	13	12	11	10	34
21H	0F	0E	0D	0C	0B	0A	09	08	33
20H	07	06	05	04	03	02	01	00	32
1FH	แบ่งที่ 3								31
18H	แบ่งที่ 2								24
17H	แบ่งที่ 1								23
10H	แบ่งที่ 0								16
0FH	แบ่งที่ 0								15
08H	แบ่งที่ 0								8
07H	แบ่งที่ 0								7
00H	แบ่งที่ 0								0

รูปที่ 5 พื้นที่การกำหนดตำแหน่งบิตของหน่วยความจำข้อมูลภายในจากรูปที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

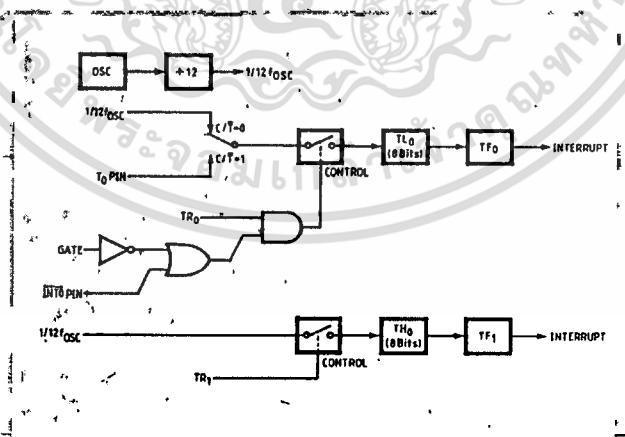


รูปที่ 6 แสดงวงจรการทำงานของตัวนับเวลา/ตัวนับที่ 1 9999999 และ 1



รูปที่ 7 แสดงวงจรการทำงานของตัวจับเวลา/ตัวนับที่ 1 9999999 และ 2

โหมด 3 เป็นการเพิ่มตัวจับเวลาขึ้นอีกหนึ่งตัว แต่จะเป็นขนาด 8 บิตทั้งคู่ ซึ่งลักษณะการทำงานอื่นๆ จะเหมือนกับโหมด 0 การทำงานแสดงดังรูปที่ 8 สำหรับรายละเอียดบทความคอมพิวเตอร์ TMO และรีจิสเตอร์ TCON แสดงในรูปที่ 9 และ 10 ตามลำดับ



รูปที่ 8 วงจรการทำงานตัวจับเวลา/ตัวนับที่ 0 9999999 และ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMER/COUNTER 1				TIMER/COUNTER 0			
GATE	C/T	M1	M0	GATE	C/T	M1	M0

GATE : เซตเป็น "1" จะเป็นกรรอนาเบิลตัวจับเวลา/ตัวนับให้ถูกควบคุมด้วยขา INTx ต้อง มีสถานะสูง และบิต TRX ใน TCON ต้องเซตเป็น "1" จึงจะเริ่มทำงาน แต่ถ้า GATE เป็น "0" ตัวจับเวลา/ตัวนับจะถูกควบคุมให้เริ่มทำงานด้วยบิต TRX เท่านั้น

C/T : เป็นบิตควบคุมในการเลือกทำงานเป็นตัวจับเวลาหรือตัวนับ ถ้าเป็น "0" จะทำงาน เป็นตัวจับเวลา ถ้าเป็น "1" ทำงานเป็นตัวนับ

M1, M0 : เป็นตัวเลือกโหมดการทำงานดังนี้

M1	M0	โหมดการทำงาน
0	0	โหมด 0
0	1	โหมด 1
1	0	โหมด 2
1	1	โหมด 3

รูปที่ 9 แสดงบิตควบคุมที่อยู่ในรีจิสเตอร์ TMOD

$TF_1$	$TR_1$	$TF_0$	$TR_0$	$IE_1$	$IT_1$	$IE_0$	$IT_0$
--------	--------	--------	--------	--------	--------	--------	--------

$TFx$  : เป็นแฟล็กอินเตอร์รีฟต์จะถูกเซตด้วยฮาร์ดแวร์เมื่อเกิดโอเวอร์โฟลวของตัวจับเวลา/ตัวนับ และจะเคลียร์ตัวเองโดยอัตโนมัติ เมื่อทำงานอินเตอร์รีฟต์นั้นเสร็จเรียบร้อยแล้ว

$TRx$  : เป็นบิตควบคุมให้ตัวจับเวลา/ตัวนับเริ่มทำงานโดยการเซตให้เป็น "1" และให้หยุดทำงานด้วยคิวเรลียร์ให้เป็น "0" โดยซอฟต์แวร์

$IEx$  : เป็นแฟล็กอินเตอร์รีฟต์จากสัญญาณภายนอก เซตด้วยฮาร์ดแวร์เมื่อมีสัญญาณขอการอินเตอร์รีฟต์ปรากฏที่ขา  $INTx$  และจะเคลียร์ตัวเองโดยอัตโนมัติ เมื่อกระโดดไปทำงานบริการอินเตอร์รีฟต์ที่ขอมาเรียบร้อยแล้ว

$ITx$  : เป็นบิตควบคุมรูปแบบสัญญาณอินเตอร์รีฟต์ภายนอก จะเซต/เคลียร์ด้วยซอฟต์แวร์ โดยถ้าเซตเป็น "1" จะถูกอินเตอร์รีฟต์ด้วยสัญญาณขอบขาสูง และถ้าเคลียร์เป็น "0" จะถูกอินเตอร์รีฟต์ด้วยสัญญาณระดับแรงดันต่ำ

หมายเหตุ x หมายถึงเลข 1 หรือเลข 0

รูปที่ 10 แสดงบิตควบคุมที่อยู่บนรีจิสเตอร์ TCON

#### พอร์ตอนุกรม

MCS 51 จะมีพอร์ตอนุกรมเป็นแบบ full duplex สามารถที่จะส่งและรับข้อมูลได้พร้อมกัน เพราะมีบัฟเฟอร์ 2 ตัว ใช้ในการรับตัวหนึ่งและส่งตัวหนึ่ง โดยโครงสร้างของรีจิสเตอร์บัฟเฟอร์ทั้ง 2 ตัวนี้ จะแยกกันแต่จะใช้การติดต่ออื่นเดียวกันคือ SBUF พอร์ตอนุกรม MCS 51 สามารถที่จะโปรแกรมให้ทำงานแตกต่างกันได้ 4 รูปแบบ

โหมด 0 ข้อมูลจะเข้าและออกทางขา RXD โดยการเปลี่ยนสัญญาณนาฬิกาออกที่ขา TXD ข้อมูลจะเป็น 8 บิต โดยจะส่งบิตน้อยสำคัญต่ำ (LSB) ก่อน อัตราบิตจะคงที่  $\frac{1}{12}$  ของความถี่ออสซิลเลเตอร์

โหมด 1 เป็นการรับ/ส่งข้อมูลขนาด 10 บิต โดยการส่งออกทางขา TXD และรับเข้าทางขา RXD รูปแบบบิตจะประกอบด้วย 1 บิตสตาร์ทเป็น "0", 8 บิตข้อมูลและ 1 บิตสตีป เป็น "1" อัตราบิตแปรผันได้ตามการตั้งตัวจับเวลาตัวที่ 1 โดยมีสูตรดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{อัตราบิด} = \frac{2^{\text{smod}}}{12 * (256 - \text{TH1})} * \text{ความถี่ออสซิลเลเตอร์}$$

32

12\*(256-TH1)

บท 2 เป็นการรับส่งข้อมูลขนาด 11 บิต เข้าทางขา RXD และ ส่งออกทางขา TXD ประกอบด้วย 1 บิตสตาร์ท มีค่า "0" , 9 บิตข้อมูลและ 1 บิตสตีป โดยการรับข้อมูลบิตที่ 9 จะนับมาเก็บที่บิต RBB ใน SCON ส่วนการส่งจะต้องส่งบิตที่ 9 ไว้ใน TBB ของ SCON ก่อน อัตราบิดสามารถเลือกได้ 2 อัตราคือ 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ขึ้นอยู่กับการเซตบิต CMOD ในรีจิสเตอร์ PCON ซึ่งรายละเอียดของรีจิสเตอร์ SCON แสดงในรูปแบบที่ 11

บท 3 จะเหมือนกับบท 2 ทุกอย่าง ยกเว้นอัตราบิดจะแปรผันตาม การตั้งตัวจับเวลา 1 ซึ่งใช้สูตรเดียวกันกับบท 1

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0, SM1 : เป็นตัวกำหนดโหมดการใช้งานของพอร์ต  
อนุกรมดังนี้

SM0	SM1	โหมด
0	0	0
0	1	1
1	0	2
1	1	3

SM2 : ควบคุมอานาเบิลการใช้โปรเซสเซอร์หลายตัวในการ  
สื่อสารซึ่งกันและกันในโหมด 2 และ 3

REN : ตัวอานาเบิลอนุกรมการรับเมื่อเซตเป็น "1" และถ้า  
เป็น "0" เป็นการดีสอเบิล

TB8 : เป็นตัวเก็บข้อมูลบิตที่ 9 ที่จะส่งในโหมด 2 และ 3

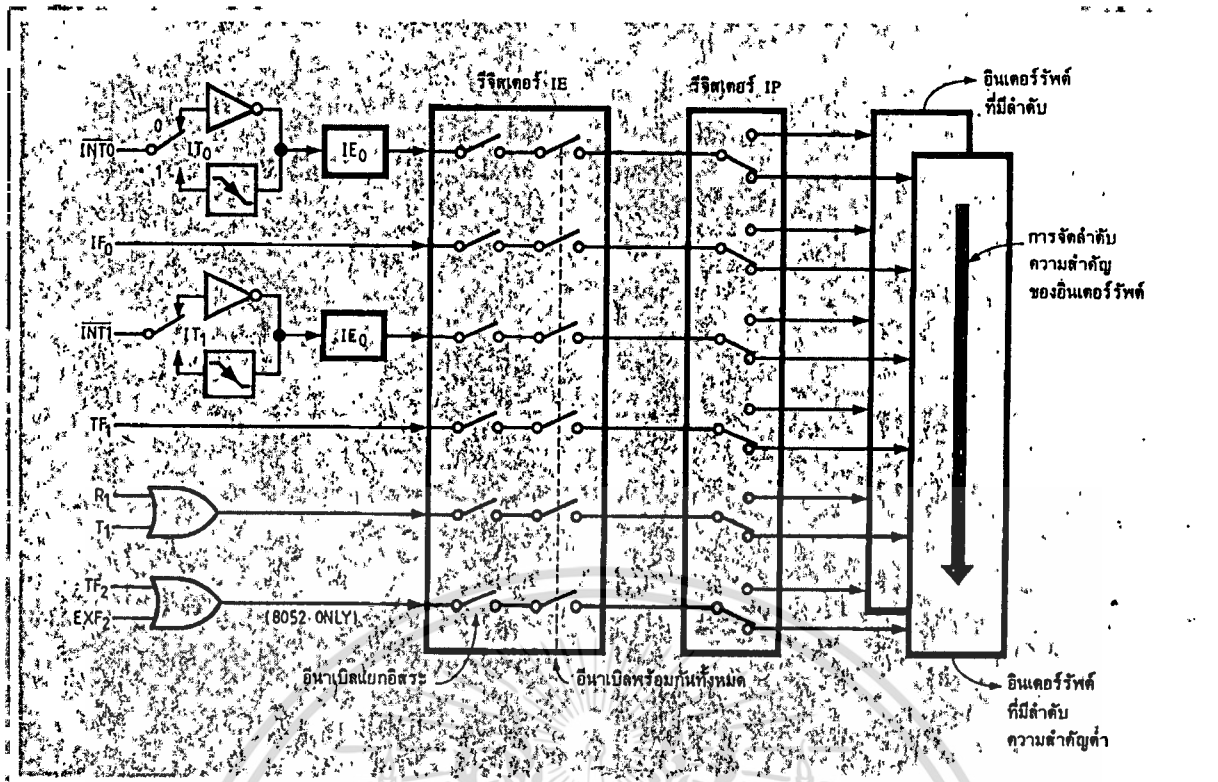
RB8 : เป็นตัวรับข้อมูลบิตที่ 9 ในโหมด 2 และ 3 ส่วนใน  
โหมด 1 จะเป็นสตัดอปบิต

TI : เป็นเฟลคอินเตอร์รัพต์การเซตด้วยฮาร์ดแวร์ที่ปลาย  
ช่วงของบิตที่ 8 ในโหมด 0 หรือที่จุดเริ่มต้นของ  
บิตสตัดอปในโหมดอื่น ในการส่งแบบอนุกรมของทุก  
โหมดจะต้องเคลียร์บิตนี้ด้วยโปรแกรมหลังการส่ง

RI : เป็นเฟลคอินเตอร์รัพต์การรับ เซตด้วยฮาร์ดแวร์  
ที่ปลายช่วงของบิตที่ 8 ในโหมด 0 หรือจุดครึ่งของ  
ช่วงบิตสตัดอปในโหมดอื่น ในการรับแบบอนุกรมจะ  
ต้องเคลียร์บิตนี้ด้วย โปรแกรมหลังการรับทุกครั้ง

รูปที่ 11 แสดงบิตควบคุมที่อยู่บนรีจิสเตอร์ SCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 แผนภูมิทางฮาร์ดแวร์ของแหล่งอินเตอร์รัพท์ชนิดต่างๆ

**การอินเตอร์รัพท์**

ในระบบของ MCS 51 จะอินเตอร์รัพท์ได้ 5 แหล่ง (ยกเว้นเบอร์ 8032 /8052 มี 6 แหล่ง) ครอบคลุมถึงระดับความสำคัญได้ 2 ระดับคือ ระดับสูงและระดับต่ำ ซึ่งจะมีแผนภูมิทางฮาร์ดแวร์ของการอินเตอร์รัพท์ ดังรูปที่ 12

การอินเตอร์รัพท์ความสำคัญต่ำสามารถที่จะถูกอินเตอร์รัพท์จากอินเตอร์รัพท์ตัวอื่นที่มีความสำคัญสูงได้ แต่ไม่สามารถที่จะถูกอินเตอร์รัพท์ จากอินเตอร์รัพท์ตัวอื่น ที่มีระดับเดียวกันได้ แต่ถ้าเป็นการอินเตอร์รัพท์จากตัวที่มีความสำคัญสูง จะไม่สามารถถูกอินเตอร์รัพท์จากตัวอื่นได้เลย ครอบคลุมถึงระดับความสำคัญของแต่ละแหล่งอินเตอร์รัพท์ได้ที่รีจิสเตอร์ IP และ ค่ากำหนดการที่มี การขอยินเตอร์รัพท์ในระดับความสำคัญเดียวกันเข้ามาพร้อมกัน ก็จะต้องมีการจัดระดับความสำคัญของแต่ละแหล่ง ดังตารางที่ 1

การถูกขัดจังหวะทั้งหมดมาทางงานบริการอินเตอร์รัพท์แต่ละแหล่งก็จะมีตำแหน่งที่จะกระโดดไปทางงานที่แน่นอน ซึ่ง MCS 51 จะกำหนดที่อยู่ตำแหน่งที่จะกระโดดไปทางงานเมื่อถูกอินเตอร์รัพท์ ดังตารางที่ 2 ส่วนรายละเอียดของรีจิสเตอร์ IE แสดงดัง

เอกสารนี้รูปที่ 13 ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แหล่งการอินเทอร์รัพต์	ลำดับความสำคัญ
อินเทอร์รัพต์ 0 จากภายนอก	1
อินเทอร์รัพต์โอเวอร์โฟลว์ของตัวจับเวลา/ตัวนับที่ 0	2
อินเทอร์รัพต์ 1 จากภายนอก	3
อินเทอร์รัพต์โอเวอร์โฟลว์ของตัวจับเวลา/ตัวนับที่ 1	4
อินเทอร์รัพต์ของพอร์ตอนุกรม	5
อินเทอร์รัพต์โอเวอร์โฟลว์ของตัวจับเวลา/ตัวนับที่ 2	6

ตารางที่ 1 ลำดับความสำคัญของแหล่งอินเทอร์รัพต์ต่างๆ

แหล่งการอินเทอร์รัพต์	ตำแหน่งที่อยู่ใน การกระโดดทำงาน
External Interrupt 0	0003H
Timer 0 Overflow	000BH
External Interrupt 1	0013H
Timer 1 Overflow	001BH
Serial Port Interrupt	0023H

ตารางที่ 2 แสดงตำแหน่งที่อยู่ของการกระโดดแต่ละแหล่งอินเทอร์รัพต์

คำสั่ง	การกระทำ	โหมดการแอดเดรส				เวลาการเอ็ก ซีกวิต (ns)
		Dir	Ind	Reg	Imm	
MOV A, <src>	A = <src>	X	X	X	X	1
MOV <dest>, A	<dest> = A	X	X	X		1
MOV <dest>, <src>	<dest> = <src>	X	X	X	X	2
MOV DPTR, #data16	DPTR = 16-bit immediate constant.				X	2
PUSH <src>	INC SP; MOV "@SP", <src>	X				2
POP <dest>	MOV <dest>, "@SP"; DEC SP	X				2
XCH A, <byte>	ACC and <byte> exchange data	X	X	X		1
XCHD A, @Ri	ACC and @Ri exchange low nibbles		X			1

ตารางที่ 3 แสดงชุดคำสั่งของอาร์กิวเมนต์การถ่ายโอนข้อมูลในแรมภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EA	-	ET2	ES	ET1	EX1	ET0	EX0
<p>EA :: ถ้า EA = 0 จะดีสแอมเบิลการอินเตอร์รัพต์ทั้งหมด ถ้า EA = 1 จะสามารถอินเตอร์รัพต์ได้ โดยแต่ละแหล่งอินเตอร์รัพต์จะมีสระในการเซต/เคลียร์ให้อินาเบิ้ลหรือดีสแอมเบิลได้</p> <p>ETx : จะเป็นบิตควบคุมการยอมรับอินเตอร์รัพต์โอเวอร์โฟลว์ของตัวจับเวลา/ตัวนับ x โดยถ้าเซตเป็น "1" จะยอมรับ แต่ถ้าเป็น "1" จะไม่ยอมรับ</p> <p>ES : จะเป็นบิตควบคุมการยอมรับอินเตอร์รัพต์ฟอร์ตฮนุกรม โดยถ้า ES = "0" จะยอมรับ แต่ถ้า ES เป็น "0" จะไม่ยอมรับ</p> <p>EXx : เป็นบิตควบคุม การยอมรับอินเตอร์รัพต์ภายนอกจากขนา INTx ถ้าเป็น "1" จะเป็นการยอมรับ แต่ถ้าเป็น "0" จะไม่ยอมรับ</p>							

รูปที่ 13 แสดงบิตควบคุมการอินเตอร์รัพต์ในรีจิสเตอร์ IE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## V-MICROCONTROLLER

ณปัจจุบันไมโครโปรเซสเซอร์มีบทบาทสำคัญเป็นอย่างมาก ถูกใช้เป็นตัวควบคุมการทำงานของเครื่องมือ เครื่องจักรกล เครื่องใช้ไฟฟ้า รวมทั้งระบบอัตโนมัติต่าง ๆ มากมาย ไมโครโปรเซสเซอร์ที่ถูกนำมาประยุกต์ใช้งานกันอย่างกว้างขวางวงเงินขนาด 8 บิตด้วยกันตัวหนึ่งก็คือ ซีพียูไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของ INTEL ซึ่งซีพียูตระกูลนี้ประกอบไปด้วย CPU เบอร์ต่าง ๆ ได้แก่ 8031, 8051, 8032, 8052, 8751, 8752 รวมทั้ง 8052AHBASIC และ DS5000 ของ DALLAS SEMICONDUCTOR

V-MICROCONTROLLER เป็นบอร์ดไมโครคอนโทรลเลอร์ขนาดเล็ก สามารถใช้ CPU เบอร์ต่าง ๆ ดังที่กล่าวมาแล้วได้ทั้งสิ้น ซึ่ง CPU แต่ละตัวในตระกูลนี้ โครงสร้างสถาปัตยกรรมส่วนใหญ่มักคล้ายคลึงกันแต่มีส่วนที่แตกต่างกันบ้างเพียงเล็กน้อย ส่วนของ INTERNAL ROM INTERNAL RAM และ TIMER/COUNTER บอร์ด V-31 ใช้สำหรับงานควบคุมขนาดเล็ก ซึ่งไม่จำเป็นต้องใช้พอร์ต I/O เพิ่มเติมก็สามารถทำการต่อขยายได้โดยผ่านทาง SYSTEM BUS 96 บิตบนบอร์ดอยู่แล้ว

### คุณสมบัติของบอร์ด V-MICROCONTROLLER

- เป็นบอร์ดคอนโทรลเลอร์ขนาดเล็กใช้กับ CPU ในตระกูล MCS-51, MCS-52 ของ INTER ได้ทุกเบอร์ ได้แก่ 8031, 8051, 8751, 8032, 8052, 8752, 8052AHBASIC รวมทั้ง DS5000 ของ DALLAS SEMICONDUCTOR บอร์ด V-MICROCONTROLLER แบ่งออกเป็น 3 แบบตามลักษณะการใช้งานดังนี้

#### V-31

ใช้ CPU เบอร์ 80C31 (CHMOS) พัฒนาโปรแกรมด้วยภาษาแอสเซมบลี

#### V-52BASIC

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวน CPU เบอร์ 8052AHBASIC ที่พัฒนาโปรแกรมด้วยภาษาเบสิก โปรดอย่าร่วมกับการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เครื่องไมโครคอมพิวเตอร์ PC

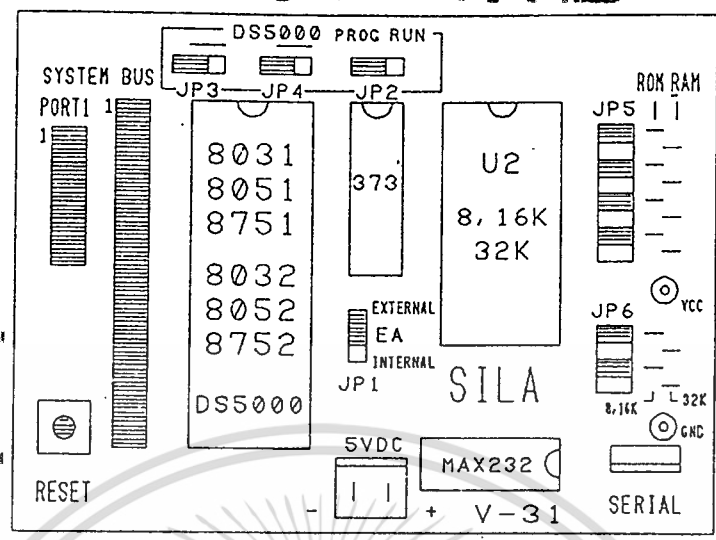
V-5000

ใช้ CPU เบอร์ DS5000 พัฒนาโปรแกรมด้วยภาษาแอสเซมบลี คุยต่อร่วมกับ  
เครื่องไมโครคอมพิวเตอร์ PC เพื่อใช้กับ CPU เบอร์ 8751

- สามารถเลือกใช้ ชนิด และ ขนาดของชิพหน่วยความจำ บนบอร์ดเป็น ROM (PROGRAMMEMORY) หรือ RAM (DATA MEMORY) ได้ทั้งขนาด 8KBYTE (2764, 6264) 16KBYTE (27128) ,32KBYTE (27256,62256)
- มีสาร์ตแวร์ในส่วนของ SERIAL INTERFACE RS-232C ใช้บนบอร์ด คุยใช้ชิพ MAX232 ของ MAXIM
- ใช้ไฟ +5VDC บอร์ดเล็กกระทัดรัด กินกระแสต่ำ
- มี HEADER STRIP 16 PIN สำหรับต่อใช้งานเป็นพอร์ต I/O ขนาด 12 บิต (PORT1 รวมทั้ง T0, T1, INTO และ INT1)
- มี HEADER STRIP 40 PIN ซึ่งต่อขนานกับขา CPU ทั้ง 40 PIN สำหรับต่อใช้งานกับอุปกรณ์ภายนอกหรือเพื่อขยายระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาพแสดงบอร์ด V-MICROCONTROLLER และตำแหน่ง JUMPER



V-MICROCONTROLLER มี JUMPER อยู่ทั้งหมด 6 จุด ซึ่งผู้ซื้อต้องปรับให้ถูกต้องตามการใช้งานของ CPU แต่ละเบอร์ มีดังนี้

- JP1 สำหรับเลือกใช้ PROGRAM MEMORY (ROM) เป็น INTERNAL (INT.) /EXTERNAL (EXT.)
- JP2 สำหรับ V-5000 ใช้เลือกโหมดการทำงานของ CPU DS5000 เป็น PROGAM /RUN
- JP3, JP4 สำหรับเลือกใช้งาน CPU เบอร์ DS5000 หรือเบอร์อื่น
- JP5 สำหรับเลือกชนิดของชิพหน่วยความจำ U2 เป็น ROM (PROGRAM MEMORY) หรือ RAM (DATA MEMORY)
- JP6 สำหรับเลือกขนาดของชิพหน่วยความจำ U2 เป็น 8,16 KBYTE หรือ 32 KBYTE

### การใช้งาน V-31

V-31 ใช้ CPU เบอร์ 80C31 (CMOS) ต่อใช้งานกับหน่วยความจำ ROM

ภายนอก (EXTERNAL PROGRAM MEMORY) ดังนั้นก่อนที่จะนำไปใช้งานจำเป็นต้องปรับค่า

เอกสารนี้เป็นเพียงเอกสารที่ผู้ซื้อต้องดู (ดูรูปหน้า 4) ดังนั้นไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP1	ปรับขาที่ตำแหน่ง EXT. (EXTERNAL)
JP2	ไม่ใช้
JP3, JP4	ปรับขาทางด้านซ้าย
JP5	ปรับขาที่ตำแหน่ง ROM เพื่อใช้เป็น PROGRAM MEMORY
JP6	เลือกขนาดของ ROM ได้ 3 เบอร์คือ 2764 , 27128 (ปรับ JP6 ขาที่ตำแหน่ง 8, 16 KBYTE) และ 27256 (ปรับ JP6 ขาที่ตำแหน่ง 32 KBYTE)

V-31 พัฒนารูปโปรแกรมด้วยภาษาแอสเซมบลีของ MCS-51 มีขั้นตอนการพัฒนาดังนี้

1. เขียน SOURCE PROGRAM ภาษาแอสเซมบลีบนเครื่องคอมพิวเตอร์ PC รดขงโปรแกรมเอ็ดิเตอร์ที่ขา และกำหนดจุดเริ่มต้นของโปรแกรม (ORIGIN ADDRESS) ที่แอดเดรส 0000H
2. นำโปรแกรมภาษาแอสเซมบลีที่เขียนเสร็จแล้วมาแปลรดขงโปรแกรมด้วยแปลภาษาแอสเซมบลี 8051 ที่ขาหรือรดขงโปรแกรม SXA51 (8051 ASSEMBLER) ก็ได้ รดขง OUTPUT FILE ที่ได้จากรดขงโปรแกรม SXA51 จะเป็นไฟล์ INTEL HEX FORMAT (.HEX)
3. ใช้งานรดขง EE-232 (EPROM EMULATOR) ช่วยในการพัฒนารดขงโปรแกรมโดยเสียบ DIP JUMPER 28 PIN ของ EE-232 แทนที่ขาตำแหน่งของ ROM บนบอร์ด V-31 แทนการรดขง EPROM แล้วทำการส่งไฟล์ (.HEX) จากเครื่องคอมพิวเตอร์ PC ขาที่ EE-232 รดขงผ่านพอร์ทอนุกรม RS-232 เพื่อรันรดขงโปรแกรมต่อไป
4. ใช้งานรดขงไม่มี EE-232 (EPROM EMULATOR) ให้นำ OUTPUT FILE ที่แปลแล้วเป็นไฟล์ .HEX นำมาแปลงเป็น OBJECT FILE ก่อน แล้วจึงนำรดขงโปรแกรมลง EPROM รดขงเครื่องรดขงโปรแกรม EPROM ที่ขา (ถ้าใช้ SEP-512 EPROM PROGRAMMER หากการรดขงโปรแกรม EPROM ก็ไม่ต้องทำการแปลง รดขงไฟล์ .HEX ได้) จากนั้นนำ EPROM ที่รดขงโปรแกรมเรียบร้อยแล้วรดขงลงบนบอร์ด V-31 ที่ตำแหน่ง ROM (U2) ขอนาฬ +5VDC 9 ขาบอร์ด V-31 เพื่อรันรดขงโปรแกรมต่อไป

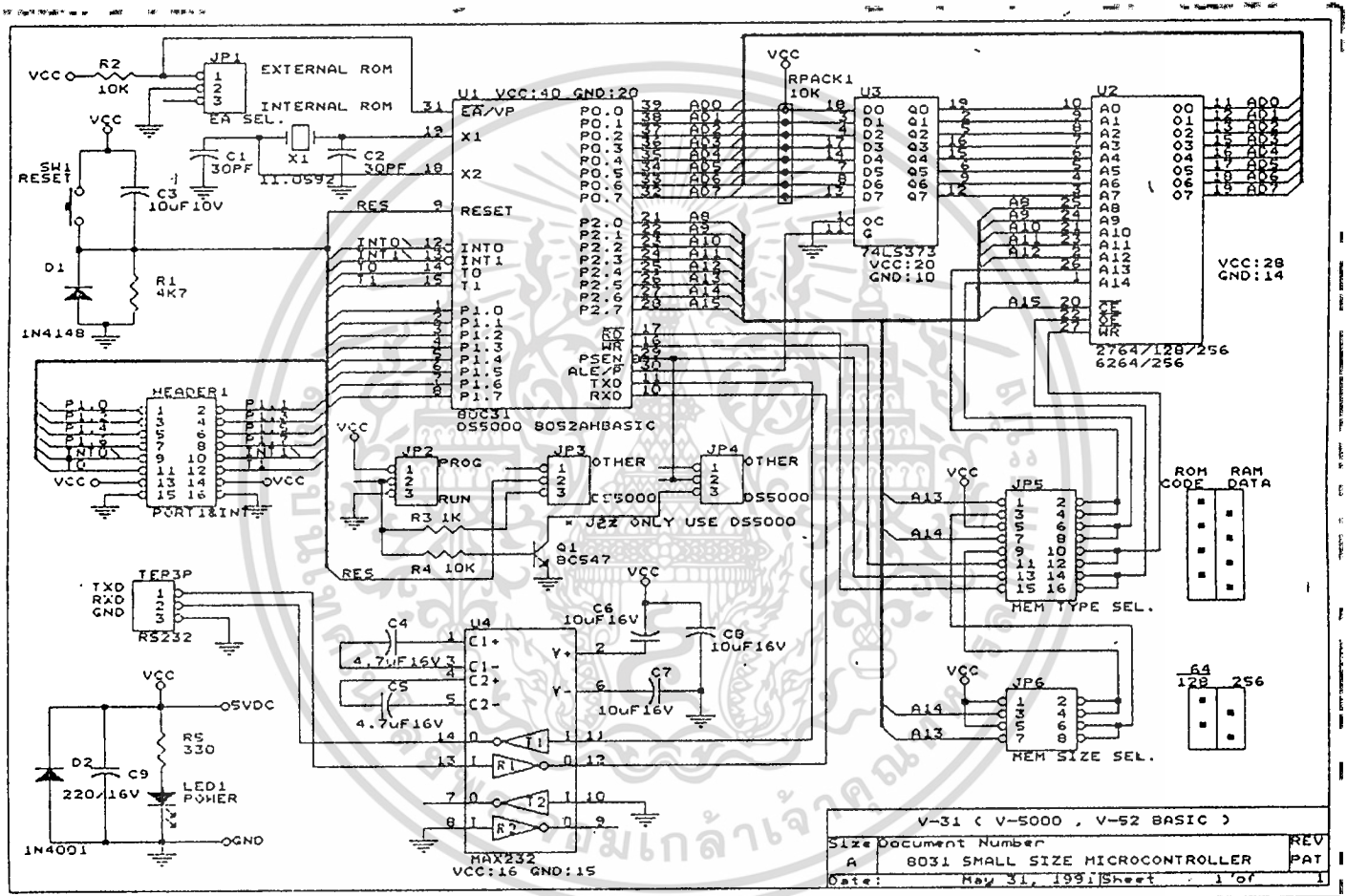
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ที่ผู้รดขงต้องการรดขงงาน SERIAL PORT ขาของ CPU บนบอร์ด V-31 ค่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถใช้งานได้โดยไม่ต้องต่อสาร์ทแวร์เพิ่มเติมแต่อย่างใด เพียงแต่เสียบชิพ MAX232 ลงบน SOCKET 16 PIN ที่วางไว้ และหากต้องการใช้งาน INTERNAL RAM เพิ่มขึ้นให้ เปลี่ยน CPU เป็นเบอร์ 8032(8052) ซึ่งจะได้ INTERNAL RAM เพิ่มขึ้นอีก 128 BYTE และได้ TIMER/COUNTER เพิ่มขึ้นอีก 1 CHANNEL

### V-31 SPECIFICATION

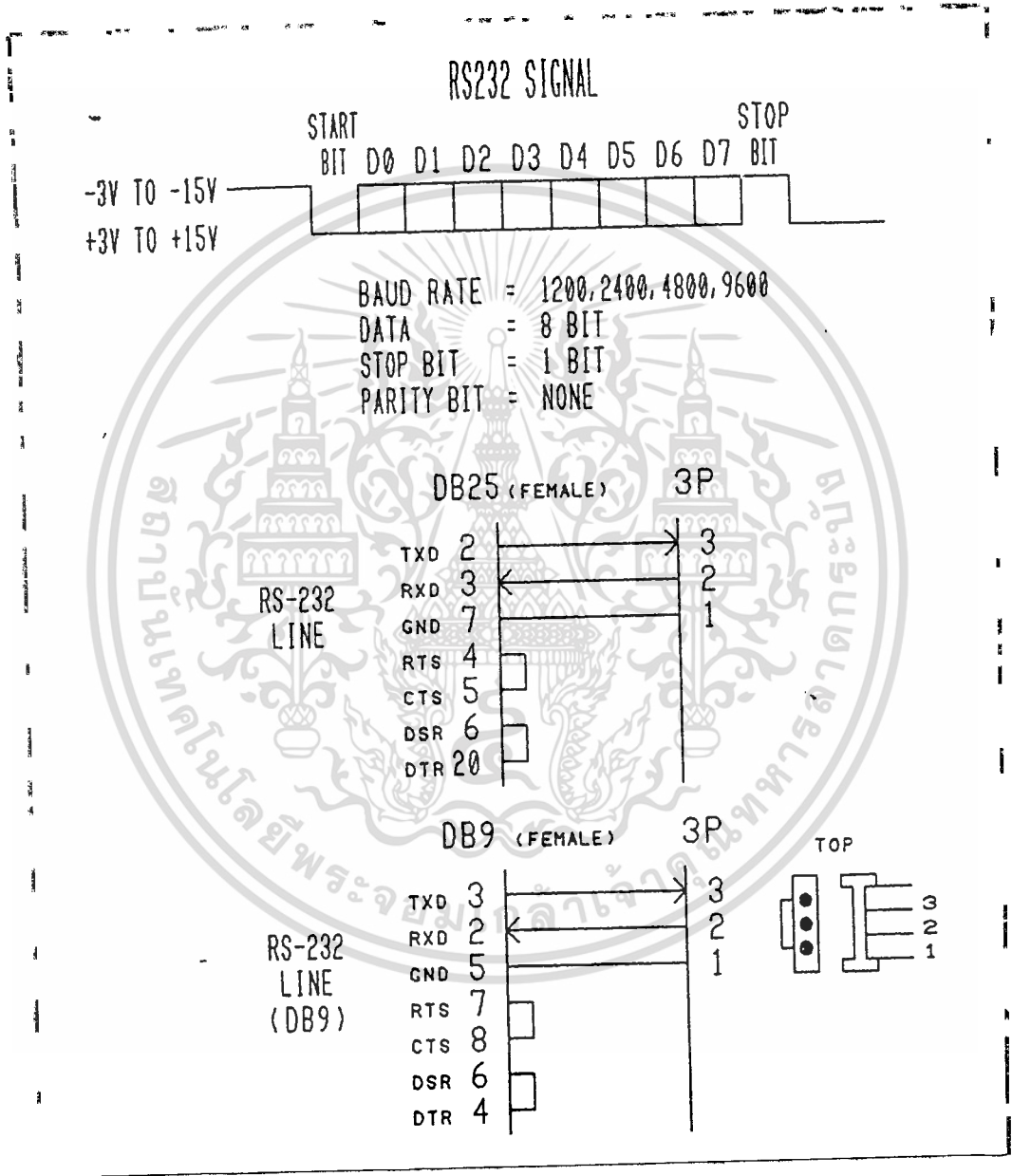
CPU	80C31 (CHMOS)
CLOCK SPEED	11.0592 MHz
INTERNAL RAM	128 BYTE
PROGRAM OR DATA MEMORY	8-32 KBYTE 2764 27128 27256 (EPROM)
	6264 62256 (RAM)
	2864 28256 (EEPROM)
	DEFAULT=28 PIN SOCKET
INTERNAL PORT	12 BIT I/O
	(PORT 1 INCLUDE TO T1 INTO INT1)
SERIAL INTERFACE	RS232C USE MAX232 CHIP
	DEFAULT=16 PIN SOCKET
CONNECTOR	1 40-PIN SYSTEM BUS (8051 COMPATIBLE)
	1 16-PIN PORT1 TO T1 INTO INT1
	1 3-PIN RS232 PORT
	1 2-PIN 5VDC POWER SUPPLY
POWER CONSUMTION	+5VDC 100mA (APPROXIMATE)
SIZE	3.50*2.85 INCHES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



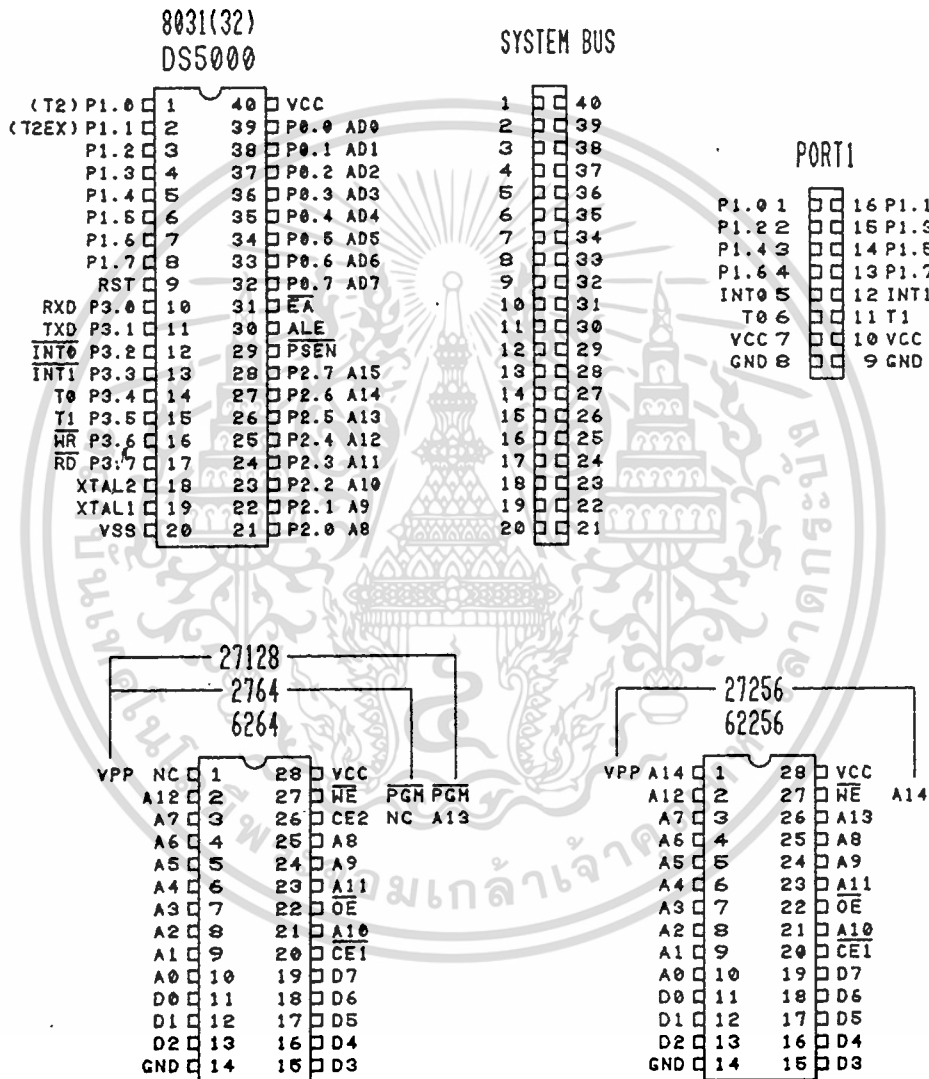
V-31 ( V-5000 , V-52 BASIC )		
Size	Document Number	REV
A	8031 SMALL SIZE MICROCONTROLLER	PAT
Date:	May 31, 1991	Sheet 1 of 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

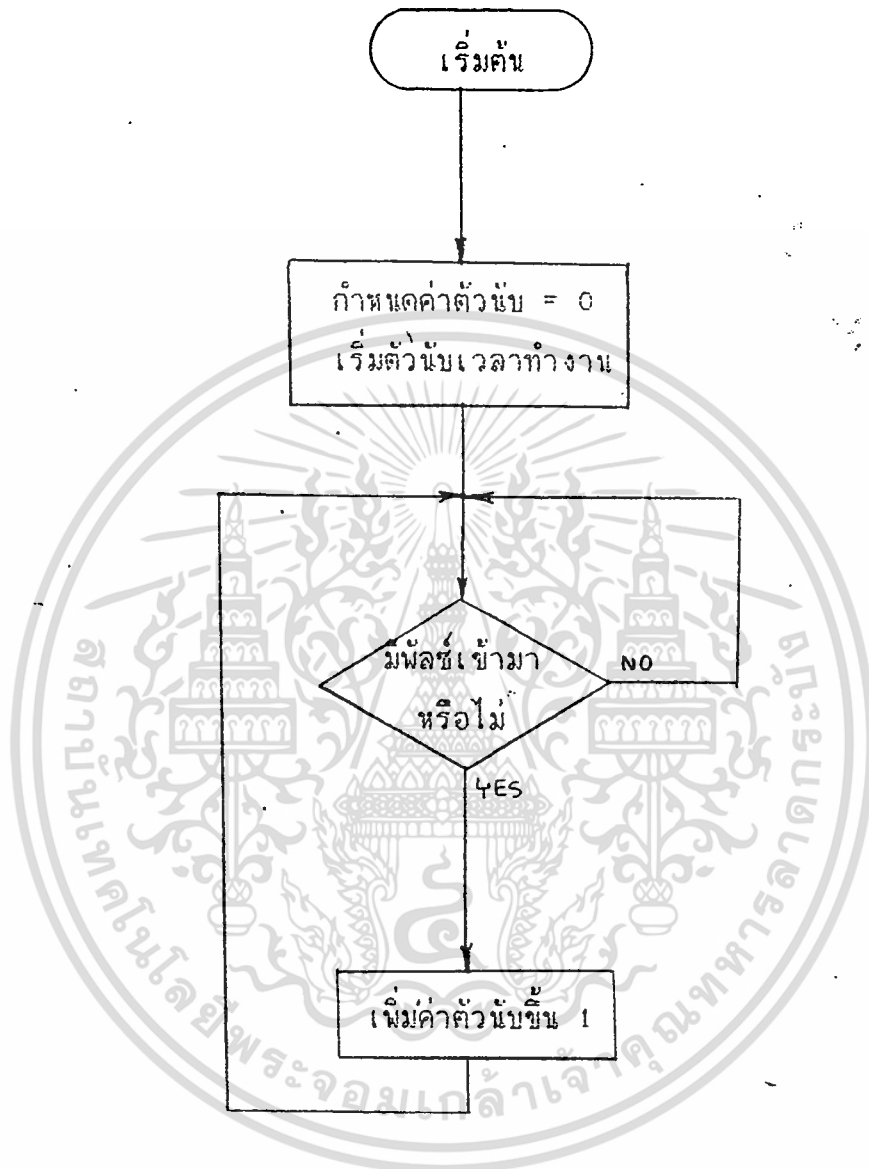


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพแสดงรายละเอียดของ CONNECTOR และ CHIP

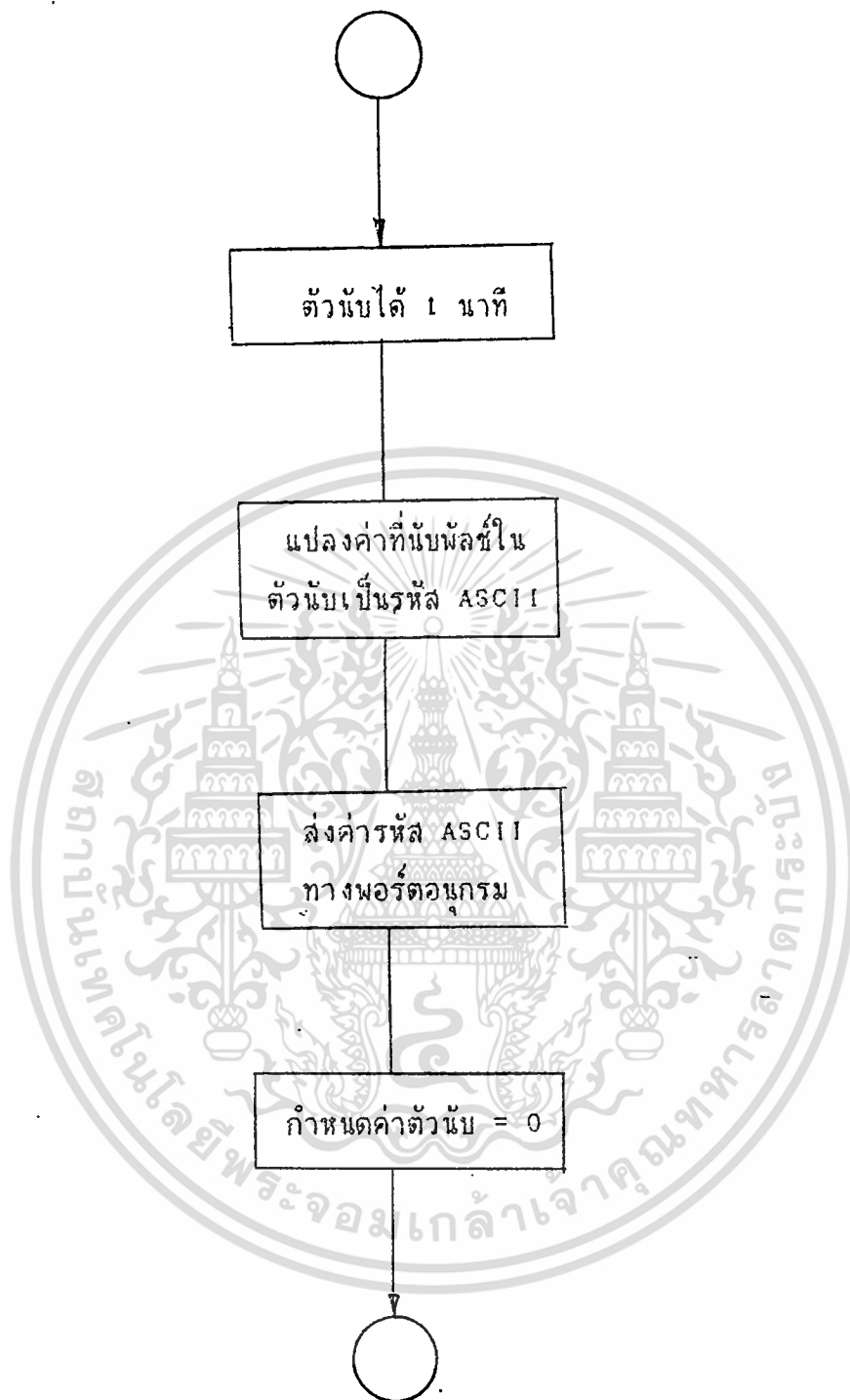


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### แผนผังโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โปรแกรมบริการอินเทอร์เน็ตเพื่อส่ง  
ค่าพัลส์ต่อนาทีให้ไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**โปรแกรมบริการอินเทอร์เน็ต  
สำหรับการตัดต่อโหลด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****

;   MAIN PROGRAM

;*****

ORG 0000H

JMP START

;*****

;   MAIN PROGRAM

;*****

ORG 040H

START: MOV IE, #90H

      MOV TMOD, #2FH

      MOV TH1, #0FDH

      MOV SCON, #52H

      MOV TH0, #00H

      MOV TLO, #00H

      SETB ET1

      SETB ET0

      SETB IT0

      SETB IT1

      SETB TR1

      SETB TR0

      CLR P3.6

      MOV R7, #0FFH

      MOV R6, #0D3H

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV R4,#00H

RBYTE:  JNB RI,$

        JMP RBYTE

;*****

; CHECK BASE TIME

;*****

TIME:   DJNZ R7,BBB

        DJNZ R6,BBB

        MOV R6,#0D3H

        DJNZ R5,BBB

        MOV 20H,TLO
        MOV TLO,#00H
        MOV A,R4
        MOV B,#64H
        DIV AB
        ADD A,#30H
        CALL TRANSMIT

        MOV A,B

        MOV B,#0AH

        DIV AB

        ADD A,#30H

        CALL TRANSMIT

        MOV A,B

        ADD A,#30H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,#0DH
CALL TRANSMIT

MOV A,20H
MOV B,#64H

DIV AB

ADD A,#30H
CALL TRANSMIT

MOV A,B
MOV B,#0AH
DIV AB
ADD A,#30H
CALL TRANSMIT
MOV A,B
ADD A,#30H
CALL TRANSMIT
MOV A,#0DH
CALL TRANSMIT

MOV R7,#0FFH

MOV R6,#0D3H

MOV R5,#04H

MOV R4,#00H

BBB: RETI

```

```

;*****

```

```

; SUBROUTINE TRANSMIT

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TRANSMIT: JNB TI,$

          CLR TI

          MOV SBUF,A

          RET

          ;*****

          ;  SUBROUTINE ON,OFF LOAD

          ;*****

OFF1:     CJNE A,#38H,OFF2

          ANL P1,#0FEH

          SETB P3.6

          CALL DELAY1

          CLR P3.6

          RET

OFF2:     CJNE A,#37H,OFF3

          ANL P1,#0FDH

          SETB P3.6

          CALL DELAY1

          CLR P3.6

          RET

OFF3:     CJNE A,#36H,OFF4

          ANL P1,#0FBH

          SETB P3.6

          CALL DELAY1

          CLR P3.6

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OFF4: CJNE A, #35H, OFF5

ANL P1, #0F7H

SETB P3.6

CALL DELAY1

CLR P3.6

RET

OFF5: CJNE A, #34H, OFF6

ANL P1, #0EFH

SETB P3.6

CALL DELAY1

CLR P3.6

RET

OFF6: CJNE A, #33H, OFF7

ANL P1, #0DFH

SETB P3.6

CALL DELAY1

CLR P3.6

RET

OFF7: CJNE A, #32H, OFF8

ANL P1, #0BFH

SETB P3.6

CALL DELAY1

CLR P3.6

RET

OFF8: CJNE A, #31H, ON1

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้สำหรับอาจารย์และบุคลากรเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANL P1,#07FH

SETB P3.6

CALL DELAY1

CLR P3.6

RET

DN1: CJNE A,#48H, DN2

ORL P1,#01H

SETB P3.6

CALL DELAY1

CLR P3.6

RET

DN2: CJNE A,#47H, DN3

ORL P1,#02H

SETB P3.6

CALL DELAY1

CLR P3.6

RET

DN3: CJNE A,#46H, DN4

ORL P1,#04H

SETB P3.6

CALL DELAY1

CLR P3.6

RET

DN4: CJNE A,#45H, DN5

ORL P1,#08H

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SETB P3.6

CALL DELAY1

CLR P3.6

RET

ON5:   CJNE A, #44H, ON6

        ORL P1, #10H

        SETB P3.6

        CALL DELAY1

        CLR P3.6

        RET

ON6:   CJNE A, #43H, ON7

        ORL P1, #20H

        SETB P3.6

        CALL DELAY1

        CLR P3.6

        RET

ON7:   CJNE A, #42H, ON8

        ORL P1, #40H

        SETB P3.6

        CALL DELAY1

        CLR P3.6

        RET

ON8:   CJNE A, #41H, ERROR

        ORL P1, #80H

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL DELAY1

CLR P3.6

ERROR:  RET

;*****

;  SUBROUTINE DELAY TIME

;*****

DELAY1:  MOV R7,#02H

        DJNZ R7,$

        RET

;*****

;  TIMER 0 INTERUPT

;*****

ORG 000BH

INC R4

RETI

;*****

;  TIMER 1 INTERUPT

;*****

ORG 001BH

JMP TIME

;*****

;  INTERUPT RECEIVEER SERIES PORT

;*****

ORG 0023H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLR RI

CALL OFF1

RETI



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คู่มือการใช้งาน

การใช้งานเครื่องควบคุมความต้องการพลังงานไฟฟ้ามีขั้นตอนดังนี้

1) เริ่มต้นต้องติดตั้งระบบในส่วนฮาร์ดแวร์ก่อนซึ่งได้แก่ การต่อจลลที่ใช้งานการควบคุมกับตัวเครื่อง ซึ่งมีทั้งหมด 8 ช่อง และเชื่อมต่อเครื่องคอมพิวเตอร์เข้ากับเครื่อง Demand Controller ทางพอร์ตอนุกรม

2) ใสแผ่นดิสก์ที่มีโปรแกรม DEMAND.EXE เข้าที่เครื่องซีดีรอม แล้วเรียกโปรแกรม DEMAND ซึ่งจะแสดงหน้าจอตั้งรูปที่ 4 ซึ่งจะมีการถามผู้ใช้งานว่าจะใช้งานว่าจะใช้งานพอร์ตอนุกรมพอร์ตาไหน 1 หรือ 2 จนหน้าจอจะมีปุ่มที่สามารถใช้งานได้ดังนี้

F1 เพื่อรายงานข้อมูลงานการใช้งาน เช่น ค่า DEMAND จนอดีตที่ผ่านมา

F2 เพื่อการตั้งค่าพารามิเตอร์ต่างๆที่ใช้งานการควบคุมพลังงาน

F3 เข้าสู่การทวงควบคุมค่า DEMAND ulyก่อนที่จะเข้าสู่การทวงงานนี้ ควรตั้งค่าพารามิเตอร์ต่างๆให้เสร็จเรียบร้อยก่อน

F10 เลิกการทวงงาน

3) เมื่อกด F2 หน้าจอจะเปลี่ยนไปเป็นรูปที่ 5 ซึ่งจะมีข้อความถามว่า "ต้องการใช้ค่าพารามิเตอร์ที่ save ใหม่ ถ้าต้องการก็กด Y แต่ถ้าไม่ต้องการก็กด N เมื่อกด N หรือ Y เรายังสามารถตั้งค่าพารามิเตอร์เองได้อีกโดย

F4 เพื่อตั้งค่าพารามิเตอร์ของ Time ซึ่งจะมีค่าที่สอดคล้องอยู่ 3 ค่า คือ Max off - Time, Min off Time และ Min on Time ของโหลดแต่ละช่อง และเราสามารถขึ้นลงตามตำแหน่งของโหลดแต่ละช่องได้โดยใช้ arrow key การเพิ่มค่าใช้ปุ่ม ENTER และการลดค่าใช้ปุ่ม SPACE BAR เมื่อได้พารามิเตอร์ต่างๆสำหรับ Time เป็นที่เรียบร้อยแล้วก็กด ESC เพื่อออกจากการตั้งค่าของ Time เมื่อต้องการตั้งค่าพารามิเตอร์ของ DEMAND กดปุ่ม F5 ซึ่งเราสามารถห้บ้อนค่าได้ตามต้องการ เมื่อได้ค่าตามต้องการแล้วกด ENTER เพื่อออกจากการตั้งค่าของ DEMAND

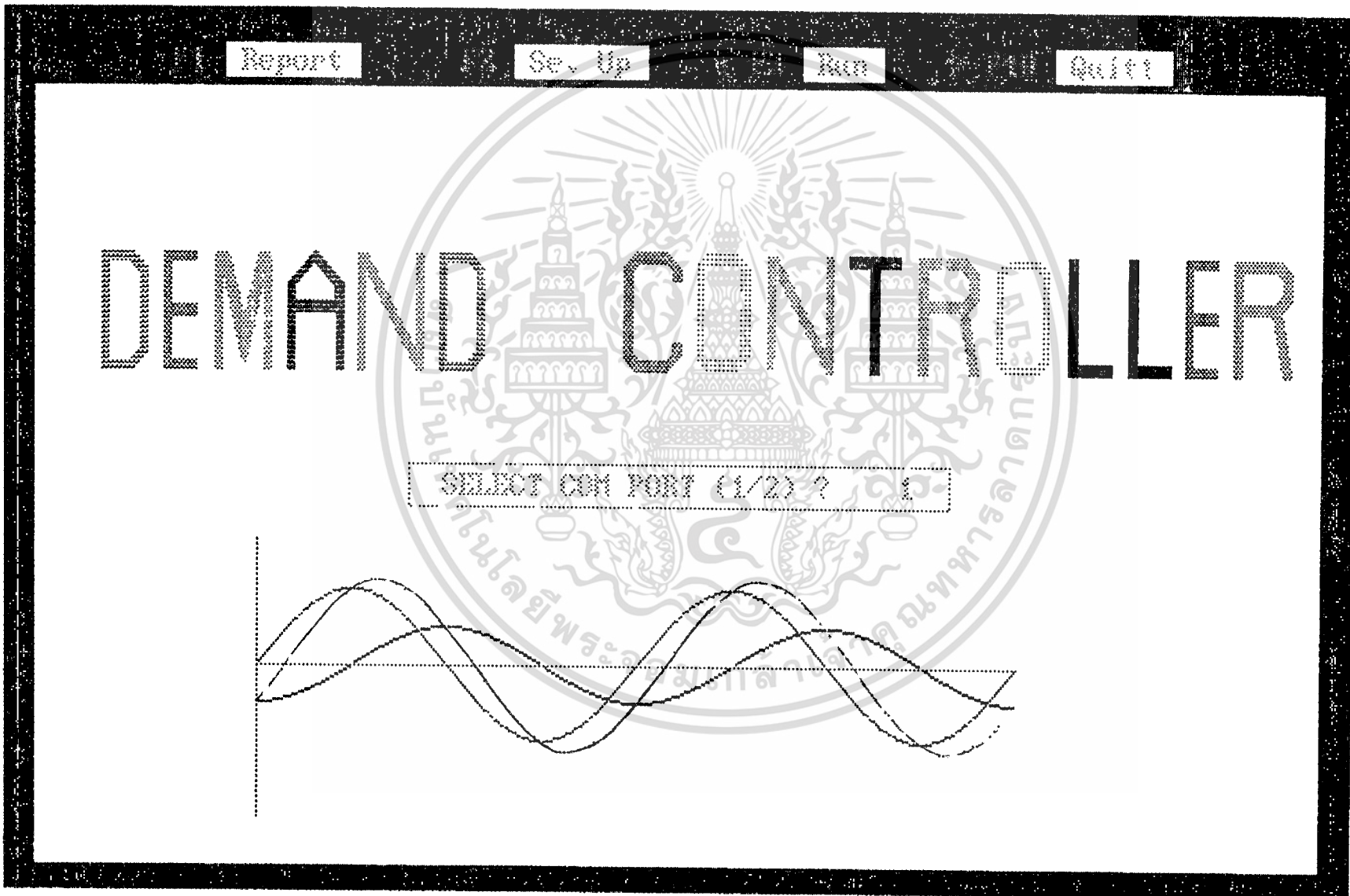
4) เมื่อกด F3 ก็จะเข้าสู่การควบคุมค่า DEMAND ( ulyก่อนที่จะเข้าสู่การทวงงานนี้ ควรปรับแต่งค่าพารามิเตอร์ต่างๆให้เสร็จเรียบร้อยก่อน ) หน้าจอจะเปลี่ยนไปเป็นดัง

รูปที่ 6 ซึ่งในหน้าจอนี้จะมีการแสดงสถานะของจอผลแต่ละช่อง พร้อมด้วยค่าพลังงานไฟฟ้าที่ผู้ใช้ (โดยจะแสดงออกมาในรูปแบบกราฟ) และในขณะที่ได้ยวกันก็จะมีการควบคุมค่า DEMAND หรือกันไปด้วย

5) เมื่อกด F1 หน้าจอจะเปลี่ยนไปเป็นดังรูปที่ 7 โดยการใช้งานจะเป็นเรื่องเกี่ยวกับข้อมูล มี OPTION ๑ ที่เลือกอยู่ 5 OPTION คือ

1. จะทำการแสดงค่า DEMAND ในช่วงเวลาที่ผ่านมาแล้ว
2. จะทำการพิมพ์ค่า DEMAND ที่ผ่านมา ออกทางเครื่องพิมพ์
3. จะทำการลบข้อมูลค่า DEMAND ในช่วงเวลาที่ไม่ต้องการออกจากดิสก์
4. จะทำการคิดเงินค่าไฟฟ้าแล้วรายงานออกมา
5. จะกลับเข้าสู่หน้าจอ RUN

ในการกดหมายเลข 1 หน้าจอจะแสดงดังรูปที่ 8 ซึ่งเราสามารถปรับเวลาที่จะแสดงค่า DEMAND ได้ โดยใช้ปุ่ม arrow key หรือ Page Up , Page Down และถ้าต้องการดูกราฟพลังงานไฟฟ้าที่ใช้ไปในแต่ละวัน กด F6 จากนั้นเลือกวันโดยใช้ปุ่ม arrow key เมื่อได้วันที่ต้องการกดปุ่ม ENTER หน้าจอก็จะแสดงกราฟพลังงานไฟฟ้าของวันนั้น และเมื่อต้องการกลับไปยังหน้าจอ RUN กดปุ่ม ESC 2 ครั้ง ก็จะเข้าสู่หน้าจอรูปที่ จากนั้นเลือกหมายเลข 5 ก็จะกลับสู่หน้าจอ RUN



**R** Report

**R** Run

**Q** Quit?

### SETUP PARAMETER

DESCRIPTION OUT PUT	PRIORITY LEVEL	Min.Off Time	Max.Off Time	Min.On Time
OUT PUT 1	1	0	0	0
OUT PUT 2	2	0	0	0
OUT PUT 3	3	0	0	0
OUT PUT 4	4	0	0	0
OUT PUT 5	5	0	0	0
OUT PUT 6	6	0	0	0
OUT PUT 7	7	0	0	0
OUT PUT 8	8	0	0	0
CONTROL MODE	AUTOMATIC			
UPPER DEMAND	0	W		
LOWER DEMAND	0	W		

DO YOU WANT TO USE DEFAULT VAULES ?  
 < Yes > or < No >

-----

24-Set Times

-----  
 -----  
 -----

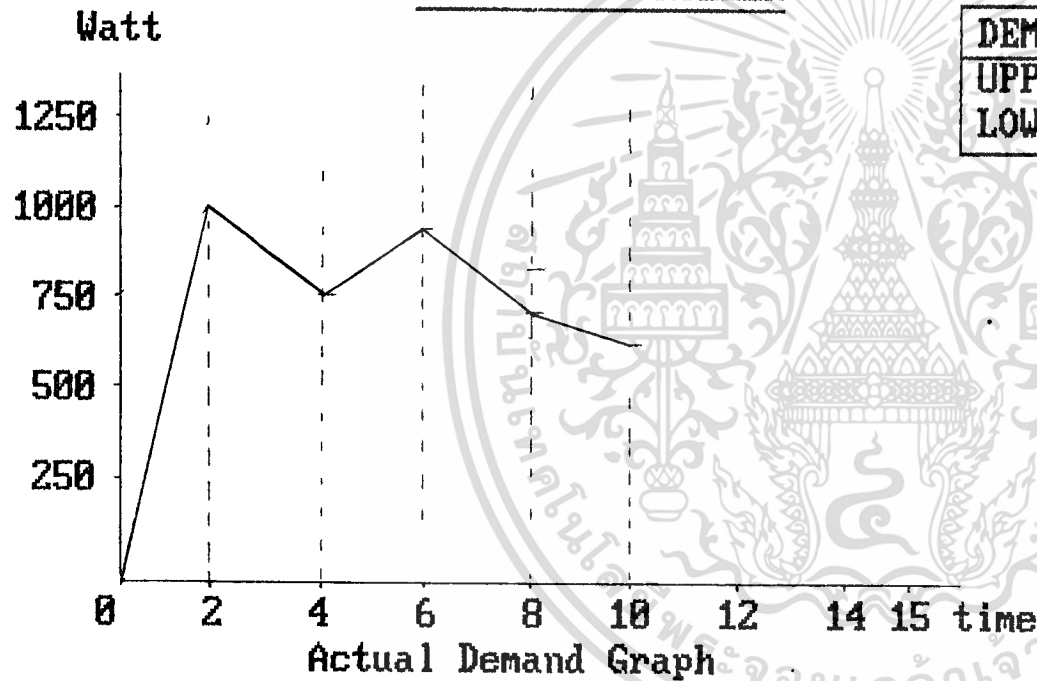
5 11/2

Report

Set Up

Quit

### DEMAND CONTROLLER



DEMANDS MAX	1000W
UPPER DEMAND	1090W
LOWER DEMAND	500 W

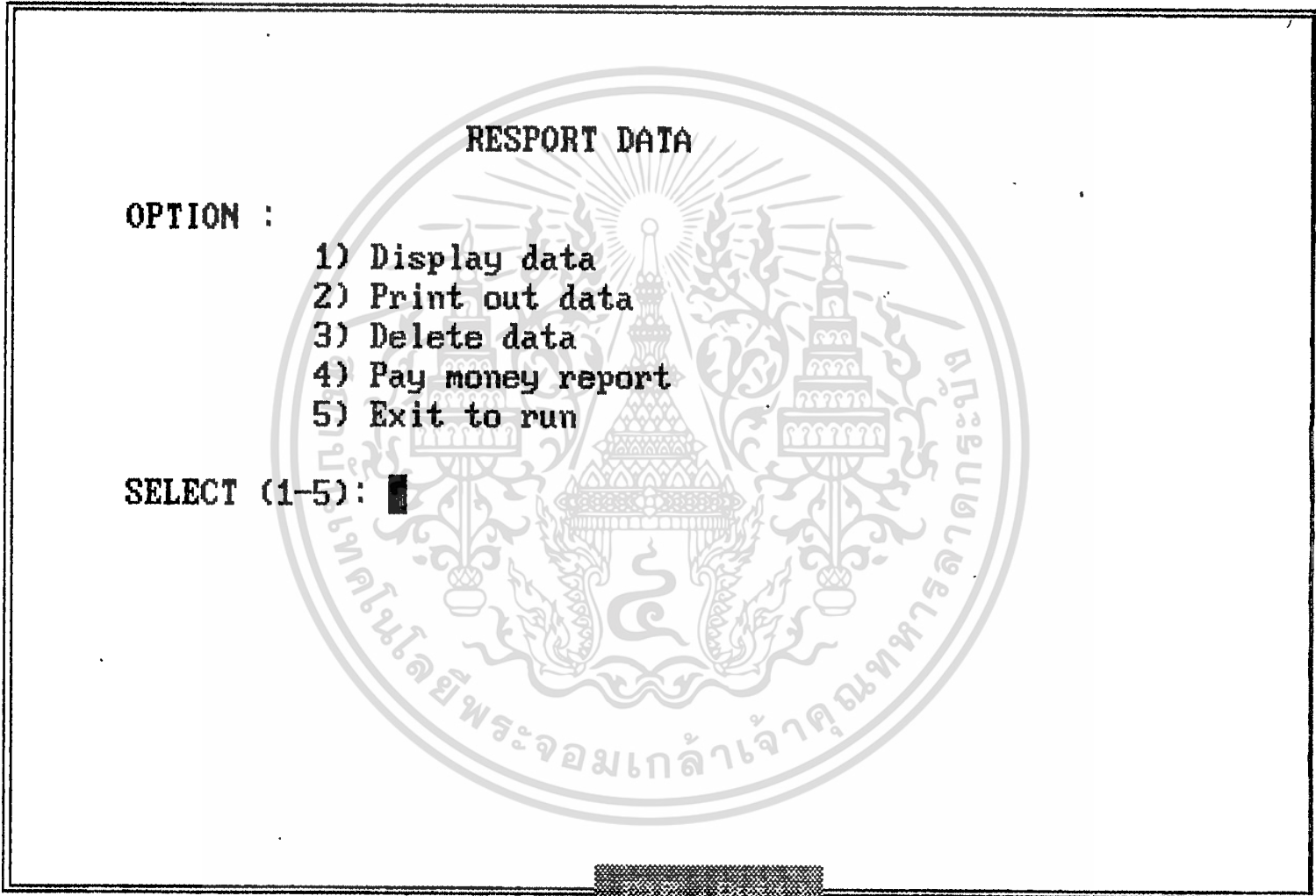
DESCRIPTION	Status
OUT PUT 1	ON
OUT PUT 2	ON
OUT PUT 3	ON
OUT PUT 4	ON
OUT PUT 5	ON
OUT PUT 6	OFF
OUT PUT 7	OFF
OUT PUT 8	OFF

DATE 03-18-1993  
 TIME 16:20:03

PREDICTION DEMAND 530WATT  
 ACCUMLATE ENERGY 378W/h

วันที่ 6

รูปที่ 7



DATE	TIME	KW/HR	0	200	400	600	800	1000	1200	1400
02-16-93	18:25	200	[REDACTED]							
02-16-93	18:26	600	[REDACTED]							
02-18-93	15:17	430	[REDACTED]							



[REDACTED]

รูปที่ 8

MARCH

DEMAND MAX : 1,455 WATT  
( DATE IS : 03-14-93 )

ACCUMULATE ENERGY 1,019 W/H

$\Sigma$  MONEY 5,139 BANTS

<ESC-Exit to Prev>

ESC Exit

รูปที่ 6

Software 'DEMAND CONTROLLER'

(QuickBASIC)

```

DECLARE SUB paymoney ()

DECLARE SUB MENU ()

DECLARE SUB DELETEDATA ()

DECLARE SUB PRINTS ()

DECLARE SUB DISPLAYS ()

DECLARE SUB MENUSETUP ()

DECLARE SUB premenu (type$)

DECLARE SUB OPENFILE (WS&)

DECLARE SUB restoretextwindow (leftcol%, topline%)

DECLARE SUB savevaule (upper%, LOWER%)

DECLARE SUB restorevaule (upper%, LOWER%)

DECLARE SUB MENURUN (LOAD%(), upper%, LOWER%, PD&, DEMANDMAX&)

DECLARE SUB MINONTIME (A%, B%, TIMENOW%, TIMEOFF%(), TIMEON%(), _
MINON%())

DECLARE SUB MINOFFTIME (c%, D%, TIMENOW%, TIMEOFF%(), TIMEON%(), _
MINOFF%())

DECLARE SUB MAXOFFTIME (TIMEOFF%(), MAXOFF%(), TIMEON%(), TIMENOW%)

DECLARE SUB INVERTTEXT (lin1%, col%, nchar%, lin2%)

DECLARE SUB colortext (lin1%, col%, nchar%, lin2%, bab%, clr%)

DECLARE SUB PLACECHARACTER (c$, x%, y%)

DECLARE SUB plotstring (lin%, col%, nchar%, s$)

DECLARE SUB savetextwindow (leftcol%, topline%, rightcol%, _
bottomline%)

```

เอกสารนี้เป็น**bottomline%**สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DECLARE SUB restoretextwindows (leftcol%, topline%)

DECLARE FUNCTION COMPUTEARRAYSIZE (x1%, y1%, x2%, y2%)

DEFINT A-Z

DIM SHARED WATT(0 TO 16)

DIM SHARED y(0 TO 16)

DIM SHARED x(0 TO 16)

DIM SHARED LOAD(1 TO 8)

DIM SHARED TIMEON%(1 TO 8)

DIM SHARED TIMEOFF%(1 TO 8)

DIM SHARED MINON%(1 TO 8)

DIM SHARED MINOFF%(1 TO 8)

DIM SHARED MAXOFF%(1 TO 8)

'#$DYNAMIC

DIM SHARED gwindow!(10)

ERASE gwindow!

'#$STATIC

CLS : T = 1: TT = 1: y(0) = 0: DEMANDMAX& = 0

CONST true = -1: CONST false = NOT true

x(0) = 81: x(1) = 82: x(2) = 100

x(3) = 119: x(4) = 138: x(5) = 157

x(6) = 176: x(7) = 195: x(8) = 214

x(9) = 233: x(10) = 252: x(11) = 271

x(12) = 290: x(13) = 309: x(14) = 328

x(15) = 347: x(16) = 366

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่โรงเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOAD(3) = true: LOAD(4) = true

LOAD(5) = true: LOAD(6) = true

LOAD(7) = true: LOAD(8) = true

DO

    dfile1% = FREEFILE

    dfile2% = FREEFILE + 1

    ON ERROR GOTO openerror

    badopen% = false%

    OPEN "DMAX.DAT" FOR INPUT AS #dfile1%

    OPEN "DMAND.INI" FOR INPUT AS #dfile2%

    ON ERROR GOTO 0

    IF badopen% THEN

        LOCATE 21, 37: COLOR 12

        PRINT "ERROR OPENING FILE"

    END IF

LOOP WHILE badopen%

DO

    IF EOF(dfile1%) THEN

        EXIT DO

    END IF

    INPUT #dfile1%, Da$, dmand1%, ac1%

LOOP

    INPUT #dfile2%, Db$, dmand2%, ac2%

    day1% = VAL(MID$(Da$, 4, 2))

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันที่ใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

day3% = VAL(MID$(DATE$, 4, 2))

IF (day3% > day1%) OR (day3% = 1) THEN

    IF (day2% < day3%) AND (day2% = day1%) THEN

        CLOSE

        datafile% = FREEFILE

        OPEN "DMAX.DAT" FOR APPEND AS #datafile%

        WRITE #datafile%, Db$, dmand2%, ac2%

        CLOSE #datafile%

    END IF

END IF

Da$ = LEFT$(DATE$, 6) + RIGHT$(DATE$, 2)

SCREEN 9: COLOR 15, 9

CALL premenu("4")

MENU

ON KEY(10) GOSUB ENDSS

KEY(10) ON

    Q = 480

DO UNTIL COUNT = 3

    IF Q >= 480 THEN

        Q = 120: ang = 0: COUNT = COUNT + 1

        amplitude1 = 30: amplitude2 = 15

        LINE (120, 260)-(480, 260), 12

        LINE (120, 210)-(120, 320), 12

    ELSE

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในสถานศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y2 = amplitude2 * COS(ang / 180 * 3.14)

PSET (Q, y1 + 260), 14: PSET (Q, y2 + 260), 13

PSET (Q, y1 + y2 + 260), 11

Q = Q + 1: ang = ang + 2

END IF

LOOP

COLOR 10: LOCATE 14, 25

PRINT " SELECT COM PORT (1/2) ? "

CALL colortext(14, 25, 32, 14, 1, 10)

DO
  i% = 54 * 8 - 14
  j% = 14 * 14 - 4
  LINE (i%, j%)-(i% + 8, j% + 1), 10, B
  DO
    n$ = INKEY$
    LOOP UNTIL n$ <> "" AND n$ >= CHR$(48) AND n$ <= CHR$(57)
    LINE (i%, j%)-(i% + 8, j% + 1), 0, B
    LOCATE 14, 54: PRINT n$
  DO
    CODE$ = INKEY$
    LOOP UNTIL CODE$ = CHR$(13)
    IF n$ <> "1" AND n$ <> "2" THEN
      BEEP: LINE (i%, j%)-(i% + 8, j% + 1), 0, B
      LOCATE 14, 54: PRINT " "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL colortext(14, 25, 32, 14, 2, 0)

EXIT DO

END IF

LOOP UNTIL n$ = "1" AND n$ = "2"

IF n$ = "1" THEN

    OPEN "COM1:9600,N,8,1,ASC" FOR RANDOM AS #1 LEN = 3

ELSEIF n$ = "2" THEN

    OPEN "COM2:9600,N,8,1,ASC" FOR RANDOM AS #1 LEN = 3

END IF

DO

    CODE$ = INKEY$

    SELECT CASE LEN(CODE$)

CASE 2

    SELECT CASE ASC(MID$(CODE$, 2, 1))

        CASE 59: GOTO DISPLAY

        CASE 60: GOTO Setup

        CASE 61: GOTO RUNIN

    END SELECT

END SELECT

END SELECT

LOOP

ENDSS: Da$ = LEFT$(DATE$, 6) + RIGHT$(DATE$, 2)

CLOSE

DO

    datafile% = FREEFILE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานวิจัยหรือการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

badopen% = false%

OPEN "DMAND.INI" FOR OUTPUT AS #datafile%

ON ERROR GOTO 0

IF badopen% THEN

    LOCATE 21, 37: COLOR 12

    PRINT "ERROR OPENING FILE"

END IF

LOOP WHILE badopen%

WRITE #datafile%, Da$, DEMANDMAX, acc&

CLOSE #datafile%

END

DISPLAY: TIMER OFF: CLS

CALL premenu("1")

LOCATE 5, 28: PRINT "RESPORT DATA"

LOCATE 7, 10: PRINT "OPTION : "

LOCATE 8, 20: PRINT "1) Display data"

LOCATE 9, 20: PRINT "2) Print out data"

LOCATE 10, 20: PRINT "3) Delete data"

LOCATE 11, 20: PRINT "4) Pay money report"

LOCATE 12, 20: PRINT "5) Exit to run"

rev: LOCATE 14, 10: INPUT "SELECT (1-5): ", s$

IF LEN(s$) > 1 THEN

    BEEP: LOCATE 13, 23: PRINT STRING$(LEN(s$) + 1, " ")

    GOTO rev

END IF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF s$ >= CHR$(49) AND s$ <= CHR$(53) THEN

    GOTO PASS

ELSE

    BEEP: LOCATE 13, 23: PRINT STRING$(LEN(s$) + 1, " ")

    GOTO rev

END IF

PASS:   SELECT CASE s$

    CASE "1": GOTO DIS

    CASE "2": GOTO PRI

    CASE "3": GOTO DELE

    CASE "4": GOTO PAY

    CASE "5": GOTO RUNIN

END SELECT

PAY: CALL paymoney

    GOTO DISPLAY

PRI:

REM Make sure the line printer is ready

'BIOS DATA AREA

DEF SEG = &H40

'GET STATUS PORT OF LPT1

STATUSPORT& = PEEK(9) * 256 + PEEK(8) + 1

IF INP(STATUSPORT&) <> 223 THEN

    ON TIMER(1) GOSUB BEEPS

TIMER ON

```

เอกสารนี้เป็นเอกสารที่ **DO** อนุญาตให้ใช้งานได้ฟรีสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOCATE 20, 10: PRINT "Printer is not ready"
```

```
LOCATE 21, 10: PRINT "Please correct the problem now.."
```

```
h$ = INKEY$: IF h$ = CHR$(27) THEN GOTO DISPLAY
```

```
    LOOP UNTIL INP(STATUSPORT&) = 223           'Wait till ready
```

```
END IF
```

```
DEF SEG
```

```
    TIMER OFF
```

```
    CALL PRINTS
```

```
    GOTO DISPLAY
```

```
BEEPS: BEEP
```

```
RETURN
```

```
DIS: CALL DISPLAYS
```

```
    GOTO DISPLAY
```

```
DELE: CALL DELETEDATA
```

```
    GOTO DISPLAY
```

```
Setup: CLS : CALL premenu("3")
```

```
    TIMER OFF
```

```
    DEFINT I-N
```

```
    DIM SHARED M(1 TO 16, 38 TO 58)
```

```
MENUSETUP
```

```
    LOCATE 19, 30: PRINT upper
```

```
    LOCATE 20, 30: PRINT LOWER
```

```
    s2$ = " ESC-Abort ": s3$ = " F4-Set Times ": s4$ = " ENTER-__
```

```
    Inc ": s5$ = " F5-Set Demands ": s6$ = " SPACE BAR-Dec "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL colortext(20, 53, 11, 20, 2, 13)

CALL colortext(22, 12, 16, 22, 2, 10)

CALL colortext(22, 35, 14, 22, 2, 11)

CALL colortext(22, 53, 11, 22, 2, 13)

CALL colortext(24, 53, 15, 24, 2, 13)

CALL plotstring(20, 53, 11, s2$)

CALL plotstring(22, 35, 14, s3$)

CALL plotstring(22, 53, 11, s4$)

CALL plotstring(22, 12, 16, s5$)

CALL plotstring(24, 53, 15, s6$)

LINE (415, 273)-(402, 273), 14, , &H9999

LINE -(402, 330), 14, , &H9999

LINE -(415, 330), 14, , &H9999

LINE (415, 301)-(402, 301), 14, , &H9999

LINE -(385, 301), 14

COLOR 15

FOR i = 9 TO 16

LOCATE i, 28: PRINT i - 8

LOCATE i, 38: PRINT M(i, 38)

LOCATE i, 48: PRINT M(i, 48)

LOCATE i, 58: PRINT M(i, 58)

NEXT

CALL savetextwindow(40, 15, 40 + 24, 15 + 5)

CALL restorevaule(upper, LOWER)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOCATE i, 38: PRINT M(i, 38)

LOCATE i, 48: PRINT M(i, 48)

LOCATE i, 58: PRINT M(i, 58)

NEXT

LOCATE 19, 30: PRINT upper

LOCATE 20, 30: PRINT LOWER

ZZZ: Z$ = INKEY$

IF Z$ = CHR$(0) + CHR$(62) THEN GOTO SETTIME

IF Z$ = CHR$(0) + CHR$(63) THEN GOTO DEMANDS

IF Z$ = CHR$(0) + CHR$(59) OR Z$ = CHR$(0) + CHR$(61) THEN

CALL savetextwindow(40, 15, 40 + 24, 15 + 4)

CALL savevaule(upper, LOWER)

GOSUB seti

IF RIGHT$(Z$, 1) = CHR$(59) THEN GOTO DISPLAY

IF RIGHT$(Z$, 1) = CHR$(61) THEN GOTO RUNIN

END IF

GOTO ZZZ

SETTIME:

NY = 16: NX = 58: c = 9: R = 38

DO UNTIL y$ = CHR$(27)

LOCATE c, R, 0: COLOR 12

LOCATE c, R, 0: PRINT M(c, R): COLOR 15

400 y$ = INKEY$: IF y$ = "" THEN 400

LOCATE c, R, 1: PRINT M(c, R)

```

เอกสารนี้เป็นเอกสารที่ IF y\$ = CHR\$(0) + CHR\$(80) THEN GOSUB 520 หน้าไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF y$ = CHR$(0) + CHR$(72) THEN GOSUB 580
IF y$ = CHR$(0) + CHR$(75) THEN GOSUB 620
IF y$ = CHR$(0) + CHR$(77) THEN GOSUB 600
IF y$ = CHR$(13) THEN GOSUB INCDATA
IF y$ = " " THEN GOSUB DELDATA

```

```

LOOP

```

```

y$ = ""

```

```

seti:

```

```

MINOFF%(1) = M(9, 38): MINOFF%(2) = M(10, 38)
MINOFF%(3) = M(11, 38): MINOFF%(4) = M(12, 38)
MINOFF%(5) = M(13, 38): MINOFF%(6) = M(14, 38)
MINOFF%(7) = M(15, 38): MINOFF%(8) = M(16, 38)
MAXOFF%(1) = M(9, 48): MAXOFF%(2) = M(10, 48)
MAXOFF%(3) = M(11, 48): MAXOFF%(4) = M(12, 48)
MAXOFF%(5) = M(13, 48): MAXOFF%(6) = M(14, 48)
MAXOFF%(7) = M(15, 48): MAXOFF%(8) = M(16, 48)
MINON%(1) = M(9, 58): MINON%(2) = M(10, 58)
MINON%(3) = M(11, 58): MINON%(4) = M(12, 58)
MINON%(5) = M(13, 58): MINON%(6) = M(14, 58)
MINON%(7) = M(15, 58): MINON%(8) = M(16, 58)
RETURN
GOTO ZZZ

```

```

DELDATA:

```

```

M(c, R) = M(c, R) - 1

```

เอกสารนี้เป็นเอกสารที่ฝังคำสั่งในโปรแกรม ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RETURN

INCDATA:

$M(c, R) = M(c, R) + 1$

IF  $M(c, R) > 60$  THEN  $M(c, R) = 60$

RETURN

520  $c = c + 1$

IF  $c > NY$  THEN  $c = 9$

RETURN

580  $c = c - 1$

IF  $c < 9$  THEN  $c = NY$

RETURN

600  $R = R + 10$

IF  $R > NX$  THEN  $R = 38$

RETURN

620  $R = R - 10$

IF  $R < 38$  THEN  $R = NX$

RETURN

DEMANDS:

LOCATE 19, 30: PRINT " "

LOCATE 19, 30: INPUT " ", upper

LOCATE 19, 30: PRINT upper

LOCATE 20, 30: PRINT " "

LOCATE 20, 30: INPUT " ", LOWER

LOCATE 20, 30: PRINT LOWER

GOTO ZZZ

เอกสารนี้เป็นเอกสารที่ระบบให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RUNIN: CLS : CALL premenu("2")

CALL MENURUN(LOAD(), upper, LOWER, PD&, DEMANDMAX&)

ON TIMER(1) GOSUB PRINTTIME: TIMER ON

IF Z1 = 0 THEN

FOR KK = 1 TO T - 1

LINE (x(KK), 249 - y(KK - 1))-(x(KK + 1), 249 - y(KK)), 12

NEXT

ELSEIF Z1 = 1 THEN

y(0) = y(15)

y(1) = y(16)

FOR HH = 1 TO T - 1

LINE (x(HH), 249 - y(HH - 1))-(x(HH + 1), 249 - y(HH)), 12

NEXT

END IF

KAKA: LINE (x(T - 1), 249 - YY)-(x(T), 249 - y), 12 'LINE GRAPH

LINE (x(T), 249 - y)-(x(T), 248), 12 'LINE VERTICAL

IF Z1 = 1 AND x(T) = 100 THEN y1 = YY

LINE (x(T), 248)-(82, 248), 12 'LINE HOR

IF Z1 = 1 THEN LINE (82, 249 - y1)-(82, 248), 12

IF x(T) <> 82 AND y <> 0 THEN PAINT (x(T) - 1, 247), 12

LOCATE 23, 10: PRINT TIME$

LOCATE 22, 60: PRINT PD&; " WATT "

LOCATE 23, 60: PRINT acc&; " W/h "

DO

```

เอกสารนี้เป็นเอกสารที่คอมพิวเตอร์ใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SELECT CASE LEN(CODE$)

    CASE 2

        SELECT CASE ASC(MID$(CODE$, 2, 1))

            CASE 59: GOTO DISPLAY

            CASE 60: GOTO Setup

        END SELECT

    END SELECT

END SELECT

day1% = VAL(MID$(DATE$, 4, 2))

day2% = VAL(MID$(Da$, 4, 2))

IF (day1% > day2%) AND day1% = 1 THEN

    datafile% = FREEFILE

    Da$ = LEFT$(DATE$, 6) + RIGHT$(DATE$, 2)

    OPEN "DMAX.DAT" FOR APPEND AS #datafile%

    WRITE #datafile%, Da$, DEMANDMAX, acc$

    CLOSE #datafile%

END IF

V = LOC(1)

IF V > 1 THEN GOSUB DEMAND

LOOP UNTIL DRAWS = 1

DRAWS = 0: GOTO KAKA

PRINTTIME:

LOCATE 23, 10: PRINT TIME$

RETURN

DEMAND:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 YY = y : DRAWS = 1: TIMENOW% = TIMER / 60  
 ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL MAXOFFTIME(TIMEOFF%(), MAXOFF%(), TIMEON%(), TIMENOW%)

INPUT #1, A&, B&

WATT& = (A& * 256 + B&) * 25

IF TT = 16 THEN

    WS& = WATTS& / 15

    CALL OPENFILE(WS&)

    WATTS& = 0

    TT = 1

    IF WS& > DEMANDMAX& THEN DEMANDMAX& = WS&

    LOCATE 6, 65: PRINT DEMANDMAX&; " WATT "

END IF

PD& = (WATTS& + ((16 - TT) * WATT&)) / 15

WATTS& = WATTS& + WATT&; TT = TT + 1

IF upper <= 1250 THEN

    y(T) = (A& * 256 + B&) * 3: IF y(T) > 160 THEN y(T) = 160

    y = (A& * 256 + B&) * 3: IF y > 160 THEN y = 160

ELSEIF upper <= 2500 THEN

    y(T) = (A& * 256 + B&) * 1.5: IF y(T) > 160 THEN y(T) = 160

    y = (A& * 256 + B&) * 1.5: IF y > 160 THEN y = 160

ELSEIF upper <= 5000 THEN

    y(T) = (A& * 256 + B&) * .75: IF y(T) > 160 THEN y(T) = 160

    y = (A& * 256 + B&) * .75: IF y > 160 THEN y = 160

ELSE

    y(T) = (A& * 256 + B&) * .375: IF y(T) > 160 THEN y(T) = 160

    y = (A& * 256 + B&) * .375: IF y > 160 THEN y = 160

```

```

END IF

IF x(T) > 365 THEN LINE (82, 89)-(370, 249), 0, BF

PRO& = WATT&

IF T > 15 THEN Z1 = 1

TOTAL& = WATT& / 60

IF T > 15 THEN T = 1

T = T + 1

acc& = acc& + TOTAL&

LOCATE 22, 60: PRINT PD&; " WATT "
LOCATE 23, 60: PRINT acc&; " W/h "

IF PD& > upper THEN
GOTO OFFLOAD
ELSEIF PD& < LOWER THEN
GOTO ONLOAD
END IF

RETURN

OFFLOAD:

IF LOAD(8) = false GOTO CHECK7 ELSE CALL MINONTIME(8,19,TIMENOW%_
, TIMEOFF%(), TIMEON%(), MINON%())

IF LOAD(8) = true GOTO CHECK7 ELSE RETURN

CHECK7:

IF LOAD(7) = false GOTO CHECK6 ELSE CALL MINONTIME(7,18,TIMENOW%_
, TIMEOFF%(), TIMEON%(), MINON%())

IF LOAD(7) = true GOTO CHECK6 ELSE RETURN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 CHECK6:  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF LOAD(6) = false GOTO CHECK5 ELSE CALL MINONTIME(6,17,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINON%())

IF LOAD(6) = true GOTO CHECK5 ELSE RETURN

CHECK5:

IF LOAD(5) = false GOTO CHECK4 ELSE CALL MINONTIME(5,16,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINON%())

IF LOAD(5) = true GOTO CHECK4 ELSE RETURN

CHECK4:

IF LOAD(4) = false GOTO CHECK3 ELSE CALL MINONTIME(4,15,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINON%())

IF LOAD(4) = true GOTO CHECK3 ELSE RETURN

CHECK3:

IF LOAD(3) = false GOTO CHECK2 ELSE CALL MINONTIME(3,14,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINON%())

IF LOAD(3) = true GOTO CHECK2 ELSE RETURN

CHECK2:

IF LOAD(2) = false GOTO CHECK1 ELSE CALL MINONTIME(2,13,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINON%())

IF LOAD(2) = true GOTO CHECK1 ELSE RETURN

CHECK1:

IF LOAD(1) = true THEN

    CALL MINONTIME(1, 12, TIMENOW%, TIMEOFF%(), TIMEON%(), MINON%())

END IF

RETURN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF LOAD(1) = true GOTO CHEK2 ELSE CALL MINOFFTIME(1,12,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINOFF%())

PRINT #1, "A"

IF LOAD(1) = false GOTO CHEK2 ELSE RETURN

CHEK2:

IF LOAD(2) = true GOTO CHEK3 ELSE CALL MINOFFTIME(2,13,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINOFF%())

PRINT #1, "B"

IF LOAD(2) = false GOTO CHEK3 ELSE RETURN

CHEK3:

IF LOAD(3) = true GOTO CHEK4 ELSE CALL MINOFFTIME(3,14,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINOFF%())

PRINT #1, "C"

IF LOAD(3) = false GOTO CHEK4 ELSE RETURN

CHEK4:

IF LOAD(4) = true GOTO CHEK5 ELSE CALL MINOFFTIME(4,15,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINOFF%())

PRINT #1, "D"

IF LOAD(4) = false GOTO CHEK5 ELSE RETURN

CHEK5:

IF LOAD(5) = true GOTO CHEK6 ELSE CALL MINOFFTIME(5,16,TIMENOW%,_
                                TIMEOFF%(), TIMEON%(), MINOFF%())

PRINT #1, "E"

IF LOAD(5) = false GOTO CHEK6 ELSE RETURN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF LOAD(6) = true GOTO CHEK7 ELSE CALL MINOFFTIME(6,17,TIMENDW%,_
                                TIMEOFF%(), TIMEON%(), MINOFF%())

PRINT #1, "F"

IF LOAD(6) = false GOTO CHEK7 ELSE RETURN

CHEK7:

IF LOAD(7) = true GOTO CHEK8 ELSE CALL MINOFFTIME(7,18,TIMENDW%,_
                                TIMEOFF%(), TIMEON%(), MINOFF%())

PRINT #1, "G"

IF LOAD(7) = false GOTO CHEK8 ELSE RETURN

CHEK8:

IF LOAD(8) = false THEN

CALL MINOFFTIME(8,19,TIMENDW%, TIMEOFF%(), TIMEON%(), MINOFF%())

PRINT #1, "H"

END IF

RETURN

openererror:

badopen% = true%

RESUME NEXT

```

```

SUB colortext (lin1%, col%, nchar%, lin2%, bab%, clr%)

```

```

x1% = col% * 8 - 8

```

```

x2% = (col% + nchar% - 1) * 8

```

```

y1% = lin1% * 14 - 16

```

```

y2% = lin2% * 14 + 1

```

```

IF bab% = 1 THEN

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับอาจารย์อาจารย์และบุคลากรในหน่วยงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LINE (x1%, y1%)-(x2%, y2%), clr%, B
ELSE
LINE (x1%, y1%)-(x2%, y2%), clr%, BF
END IF
END SUB

DEFSNG A-Z
FUNCTION COMPUTEARRAYSIZE (x1%, y1%, x2%, y2%)
COMPUTEARRAYSIZE! = 1 + INT((x2% - x1% + 8) / 8) * (y2% - y1%_
END FUNCTION

DEFINT A-Z
SUB DELETEDATA
LINE (21, 31)-(619, 319), 0, BF
LOCATE 6, 6: INPUT "START DATE :", start$
LOCATE 7, 8: INPUT "END DATE :", END$
OPEN "I", #2, "DEMAND.DAT"
OPEN "O", #3, "COPY"
11 IF EOF(2) THEN 14
INPUT #2, D$, T$, W
IF start$ = LEFT$(D$, LEN(start$)) THEN 12
WRITE #3, D$, T$, W
GOTO 11
12 IF EOF(2) THEN 14

```

เอกสารนี้เป็นเอกสารที่รวบรวมไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF END$ = LEFT$(D$, LEN(END$)) THEN 13

GOTO 12

13 IF EOF(2) THEN 14

INPUT #2, D$, T$, W

IF END$ = LEFT$(D$, LEN(END$)) THEN 13

WRITE #3, D$, T$, W

GOTO 13

14 CLOSE #2: CLOSE #3

KILL "DEMAND.DAT"

NAME "COPY" AS "DEMAND.DAT"

END SUB

SUB DISPLAYS

DIM D$(1 TO 1056)

DIM T$(1 TO 1056)

DIM W(0 TO 1056)

DIM WW(1 TO 1056)

DIM WT(0 TO 97)

DIM V(0 TO 97)

LINE (21, 31)-(619, 319), 0, BF

n = 1: c = 1

LOCATE 6, 6: INPUT "START DATE :", start$

LOCATE 7, 8: INPUT "END DATE :", ends$

LINE (21, 31)-(200, 100), 0, BF

```

เอกสารนี้เป็นเอกสารที่วางจำหน่ายโดยทางมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

60 IF EOF(2) THEN 120

    INPUT #2, D$, T$, W

    IF start$ = LEFT$(D$, LEN(start$)) THEN 100

    GOTO 60

100 D$(n) = D$: T$(n) = T$: W(n) = W: WW(n) = W / 50: n = n + 1

    IF n > 1055 THEN 120

    IF EOF(2) THEN 120

    INPUT #2, D$, T$, W

    IF ends$ = LEFT$(D$, LEN(ends$)) THEN 115

    GOTO 100

115 D$(n) = D$: T$(n) = T$: W(n) = W: WW(n) = W / 50: n = n + 1

    IF n > 1055 THEN 120

    IF EOF(2) THEN 120

    INPUT #2, D$, T$, W

    IF ends$ <> LEFT$(D$, LEN(ends$)) THEN 120

    GOTO 115

120 CLOSE #2

    LOCATE 4,8:PRINT "DATE      TIME      KW/HR  0   200   400   600_
                                     800  1000  1200  1400"

    LINE (240, 60)-(576, 60), 12

    FOR A1 = 240 TO 576 STEP 48

    LINE (A1, 58)-(A1, 62), 12

    NEXT

    A7 = 18

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยอัตโนมัติจากโปรแกรมคอมพิวเตอร์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FOR i = 1 TO A7
```

```
FF = W(i) / 33
```

```
LOCATE i + 5, 6: PRINT D$(i); TAB(16); T$(i); TAB(24); W(i);
```

```
TAB(31); STRING$(FF, 223)
```

```
NEXT
```

```
HITKEY:
```

```
DO
```

```
10 A$ = INKEY$: IF A$ = "" THEN 10
```

```
IF A$ = CHR$(0) + CHR$(72) THEN GOSUB UP
```

```
IF A$ = CHR$(0) + CHR$(80) THEN GOSUB DOWN
```

```
IF A$ = CHR$(0) + CHR$(73) THEN GOSUB PAGEUP
```

```
IF A$ = CHR$(0) + CHR$(81) THEN GOSUB PAGEDOWN
```

```
IF A$ = CHR$(0) + CHR$(64) THEN GOTO DEMANDDAY
```

```
IF A$ = CHR$(27) THEN EXIT SUB
```

```
LOOP
```

```
A$ = ""
```

```
GOTO ENDD
```

```
UP:
```

```
c = c + 1
```

```
IF c > 1056 THEN c = 1056
```

```
GOSUB PRINTS
```

```
RETURN
```

```
DOWN:
```

```
c = c - 1
```

```
IF c < 1 THEN c = 1
```

```
GOSUB PRINTS
```

```
RETURN
```

```
PAGEUP:
```

```
c = c + 18
```

```
IF c > 1056 THEN c = 1056
```

```
GOSUB PRINTS
```

```
RETURN
```

```
PAGEDOWN:
```

```
c = c - 18
```

```
IF c < 1 THEN c = 1
```

```
GOSUB PRINTS
```

```
RETURN
```

```
PRINTS:
```

```
F = c: A5 = 23
```

```
IF n <= 17 THEN A5 = n + 4
```

```
FOR i = 6 TO A5
```

```
IF D$(F) = "" THEN
```

```
LOCATE i, 6: PRINT SPACE$(66)
```

```
GOTO PAS
```

```
END IF
```

```
FF = W(F) / 33
```

```
LOCATE i, 31 + FF: PRINT SPACE$(75 - (31 + FF))
```

```
LOCATE i, 6: PRINT D$(F); TAB(16); T$(F); TAB(24); W(F); TAB_
```

```
(31); STRING$(FF, 223)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

F = F + 1

NEXT

RETURN

DEMANDDAY:

LINE (21, 31)-(619, 339), 0, BF

DAT$ = D$(1): i = 1

LOCATE 5, 20: PRINT "DEMANDS OF DATE IS : ";

COLOR 12: PRINT ; DAT$: COLOR 15

DAYS: DO

Da$ = INKEY$

IF Da$ = CHR$(0) + CHR$(77) THEN GOSUB NEXTDAY
IF Da$ = CHR$(0) + CHR$(75) THEN GOSUB BACKDAY
IF Da$ = CHR$(27) THEN GOTO DISP

LOOP UNTIL Da$ = CHR$(13)

LINE (81, 85)-(561, 249), 0, BF

LINE (80, 85)-(80, 250), 14

LINE (80, 250)-(560, 250), 14

FOR Z = 217 TO 85 STEP -29

LINE (77, Z)-(79, Z), 14

NEXT

FOR x = 100 TO 560 STEP 20

LINE (x, 251)-(x, 253), 14

NEXT

LOCATE 19,11:PRINT "0 2 4 6 8 10 12 14 16_"

```

```

LOCATE 6, 9: PRINT "Watt"

LOCATE 8, 6: PRINT "1250"

LOCATE 10, 6: PRINT "1000"

LOCATE 12, 7: PRINT "750"

LOCATE 14, 7: PRINT "500"

LOCATE 16, 7: PRINT "250"

s = 0: V = 80: V(0) = 81

FOR j = 1 TO 96

V = V + 5: V(j) = V

NEXT

FOR j = 1 TO n

IF DAT$ = D$(j) THEN

s = s + 1

WT(s) = (W(j) / 25) * 3

LINE (V(s - 1), 249 - WT(s - 1))-(V(s), 249 - WT(s)), 12

END IF

NEXT

NODAY: WT(0) = WT(1)

LINE (81, 248)-(81, WT(0)), 12 'LINE ONE

LINE (V(s), 249 - WT(s))-(V(s), 248), 12 'LINE END

LINE (81, 248)-(V(s), 248), 12 'LINE DOWN

IF s > 1 THEN PAINT (82, 247), 12

GOTO DAYS

DISP: LINE (21, 31)-(619, 339), 0, BF

```

เอกสารนี้ LOCATE 4, 8: PRINT "DATE TIME KW/HR 0 200 400 600" ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

800 1000 1200 1400"

```

LINE (240, 60)-(569, 60), 2
FOR i = 240 TO 569 STEP 47
LINE (i, 58)-(i, 62), 2: NEXT
GOSUB PRINTS
GOTO HITKEY
NEXTDAY: i = i + 96
IF i > n THEN i = n - 2
DAT$ = D$(i): COLOR 12
LOCATE 5, 41: PRINT DAT$: COLOR 15
RETURN
BACKDAY: i = i - 96
IF i < 1 THEN i = 1
DAT$ = D$(i): COLOR 12
LOCATE 5, 41: PRINT DAT$: COLOR 15
RETURN
ENDD:
END SUB

```

```

SUB INVERTTEXT (lin1%, col%, nchar%, lin2%)

```

```

    x1% = col% * 8 - 8

```

```

    x2% = (col% + nchar% - 1) * 8

```

```

    y1% = lin1% * 14 - 14

```

```

    y2% = lin2% * 14 - 1

```

```

    arraysize% = COMPUTEARRAYSIZE(x1%, y1%, x2%, y2%)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'$DYNAMIC

REDIM MYARRAY!(arraysize%)

'$STATIC

GET (x1%, y1%)-(x2%, y2%), MYARRAY!

PUT (x1%, y1%), MYARRAY!, PRESET

ERASE MYARRAY!

END SUB

SUB MAXOFFTIME (TIMEOFF%( ), MAXOFF%( ), TIMEON%( ), TIMENOW%)

  IF LOAD(1) = true THEN GOTO CHE2

  IF TIMENOW% >= TIMEOFF%(1) + MAXOFF%(1) THEN

LOAD(1) = true
LOCATE 12, 71: PRINT "ON "
PRINT #1, "A"
TIMEON%(1) = TIMER / 60

  END IF

CHE2:

  IF LOAD(2) = true THEN GOTO CHE3

  IF TIMENOW% >= TIMEOFF%(2) + MAXOFF%(2) THEN

LOAD(2) = true
LOCATE 13, 71: PRINT "ON "
PRINT #1, "B"
TIMEON%(2) = TIMER / 60

  END IF

CHE3:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF LOAD(3) = true THEN GOTO CHE4

IF TIMENOW% >= TIMEOFF%(3) + MAXOFF%(3) THEN

LOAD(3) = true

LOCATE 14, 71: PRINT "ON "

PRINT #1, "C"

TIMEON%(3) = TIMER / 60

END IF

CHE4:

IF LOAD(4) = true THEN GOTO CHE5

IF TIMENOW% >= TIMEOFF%(4) + MAXOFF%(4) THEN

LOAD(4) = true

LOCATE 15, 71: PRINT "ON "

PRINT #1, "D"

TIMEON%(4) = TIMER / 60

END IF

CHE5:

IF LOAD(5) = true THEN GOTO CHE6

IF TIMENOW% >= TIMEOFF%(5) + MAXOFF%(5) THEN

LOAD(5) = true

LOCATE 16, 71: PRINT "ON "

PRINT #1, "E"

TIMEON%(5) = TIMER / 60

END IF

CHE6:

```

```

IF TIMENOW% >= TIMEOFF%(6) + MAXOFF%(6) THEN

LOAD(6) = true

LOCATE 17, 71: PRINT "ON "

PRINT #1, "F"

TIMEON%(6) = TIMER / 60

END IF

CHE7:

IF LOAD(7) = true THEN GOTO CHE8

IF TIMENOW% >= TIMEOFF%(7) + MAXOFF%(7) THEN

LOAD(7) = true

LOCATE 18, 71: PRINT "ON "

PRINT #1, "G"

TIMEON%(7) = TIMER / 60

END IF

CHE8:

IF LOAD(8) = true THEN EXIT SUB

IF TIMENOW% >= TIMEOFF%(8) + MAXOFF%(8) THEN

LOAD(8) = true

LOCATE 19, 71: PRINT "ON "

PRINT #1, "H"

TIMEON%(8) = TIMER / 60

END IF

END SUB

```

เอกสารนี้เป็น DEFSNG ที่ A+Z ไม่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUB MENU

PSET (50, 100), 2

'D

DRAW "c12d40r10e5u30h5l10bm-3,-3d46r15e6u34h6l15bf1p12,12be1br27"

'E

DRAW "c2d46r15u3l12u18r12u4l12u18r12u3l15bf1p2,2be1br22"

'M

DRAW "c10d46r3u40m+10,40m+10,-40d40r3u46l6m-7,26m-7,-26l6bf1p10,10be1br45"

'A

DRAW "c4bu1g13d34r3u23r20d23r3u34h13bd4g10d6r19u6h9be1p4,4bf1bm+18,-4"

'M

DRAW "c14d46r3u40m+25,40r3u46l3d39m-24,-39l3bf1p14,14be1bm+39,3"

'D

DRAW "c12d40r10e5u30h5l10bm-3,-3d46r15e6u34h6l15bf1p12,12be1br90"

'C

DRAW "c8g5d36f5r15e7u4l4d4g3l13h3u33e3r13f3d4r4u4h7l14bf1p8,8be1br31"

'O

DRAW "c11g5d36f5r15e5u36h5l15bd4br1g3d32f3r12e3u32h3l12be1p11,11bf1br26"

'N

DRAW "c14bu3d46r3u40m+25,40r3u46l3d39m-24,-39l3bf1p14,14be1br36"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DRAW "c7d4r10d42r4u42r10u4124bf1p7,7be1br31"

'R

DRAW "c13d46r4u25r15m+5,25r4m-5,-25e5u11h5123bd5br4d12r16e4u5h41\_  
16be1p13,13bf1"

'O

DRAW "c11br36bu4g5d36f5r15e5u36h5115bd4br1g3d32f3r12e3u32h3112be\_  
1p11,11bf1br26"

'L

DRAW "c15bu4d46r20u4116u4214bf1p15,15be1br26"

'L

DRAW "c15d46r20u4116u4214bf1p15,15be1br25"

'E

DRAW "c2d46r15u3112u18r12u4112u18r12u3115bf1p2,2be1br21"

'R

DRAW "c13d46r4u25r15m+5,25r4m-5,-25e5u11h5123bd5br4d12r16e4u5h41\_  
16be1p13,13bf1"

END SUB

DEFINT A-Z

SUB MENURUN (LOAD(), upper, LOWER, PD%, DEMANDMAX&)

LINE (21, 31)-(619, 339), 0, BF

LINE (80, 85)-(80, 250), 14

LINE (80, 250)-(380, 250), 14

FOR Z = 217 TO 85 STEP -29

เอกสารนี้เป็นเอกสารที่... LINE (77, Z)-(79, Z), 14  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NEXT
```

```
FOR x = 115 TO 375 STEP 40
```

```
    LINE (x, 251)-(x, 253), 14
```

```
NEXT
```

```
LOCATE 19, 10: PRINT "0    2    4    6    8    10   12   14 15 _
                                     time"
```

```
LOCATE 6, 9: PRINT "Watt"
```

```
IF upper <= 1250 THEN
```

```
    LOCATE 8, 6: PRINT "1250"
```

```
    LOCATE 10, 6: PRINT "1000"
```

```
    LOCATE 12, 7: PRINT "750"
```

```
    LOCATE 14, 7: PRINT "500"
```

```
    LOCATE 16, 7: PRINT "250"
```

```
ELSEIF upper <= 2500 THEN
```

```
    LOCATE 8, 6: PRINT "2500"
```

```
    LOCATE 10, 6: PRINT "2000"
```

```
    LOCATE 12, 6: PRINT "1500"
```

```
    LOCATE 14, 6: PRINT "1000"
```

```
    LOCATE 16, 7: PRINT "500"
```

```
ELSEIF upper <= 5000 THEN
```

```
    LOCATE 8, 6: PRINT "5000"
```

```
    LOCATE 10, 6: PRINT "4000"
```

```
    LOCATE 12, 6: PRINT "3000"
```

```
    LOCATE 14, 6: PRINT "2000"
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELSE

LOCATE 8, 5: PRINT "10000"

LOCATE 10, 6: PRINT "8000"

LOCATE 12, 6: PRINT "6000"

LOCATE 14, 6: PRINT "4000"

LOCATE 16, 6: PRINT "2000"

END IF

LOCATE 5, 25: PRINT "DEMAND CONTROLLER"

LINE (190, 70)-(327, 70)

LOCATE 20, 20: PRINT "Actual Demand Graph"

LOCATE 22, 5: PRINT "DATE"

LOCATE 23, 5: PRINT "TIME"

LOCATE 22, 10: PRINT DATE\$

LOCATE 6, 51: PRINT "DEMANDS MAX "; DEMANDMAX&; " W"

LOCATE 7, 51: PRINT "UPPER DEMAND "; upper; " W"

LOCATE 8, 51: PRINT "LOWER DEMAND "; LOWER; " W"

UP = upper / 25 \* 3: LO = LOWER / 25 \* 3

LOCATE 23, 43: PRINT "ACCUMLATE ENERGY "; acc&; " W/h "

LOCATE 22, 43: PRINT "PREDICTION DEMAND "; PD&; " WATT "

LOCATE 11, 57: PRINT "DESCRIPTION Status"

LOCATE 12, 57: PRINT "OUT PUT 1"

LOCATE 13, 57: PRINT "OUT PUT 2"

LOCATE 14, 57: PRINT "OUT PUT 3"

LOCATE 15, 57: PRINT "OUT PUT 4"

LOCATE 16, 57: PRINT "OUT PUT 5"

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOCATE 17, 57: PRINT "OUT PUT 6"

LOCATE 18, 57: PRINT "OUT PUT 7"

LOCATE 19, 57: PRINT "OUT PUT 8"

IF LOAD(1) = true THEN

    LOCATE 12, 71: PRINT "ON "

ELSE

    LOCATE 12, 71: COLOR 12: PRINT "OFF": COLOR 15

END IF

IF LOAD(2) = true THEN

    LOCATE 13, 71: PRINT "ON "

ELSE

    LOCATE 13, 71: COLOR 12: PRINT "OFF": COLOR 15

END IF

IF LOAD(3) = true THEN

    LOCATE 14, 71: PRINT "ON "

ELSE

    LOCATE 14, 71: COLOR 12: PRINT "OFF": COLOR 15

END IF

IF LOAD(4) = true THEN

    LOCATE 15, 71: PRINT "ON "

ELSE

    LOCATE 15, 71: COLOR 12: PRINT "OFF": COLOR 15

END IF

IF LOAD(5) = true THEN

```

เอกสารนี้เป็นเอกสารที่นำขึ้นจากอินเทอร์เน็ตเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ELSE

    LOCATE 16, 71: COLOR 12: PRINT "OFF": COLOR 15

END IF

IF LOAD(6) = true THEN

    LOCATE 17, 71: PRINT "ON "

ELSE

    LOCATE 17, 71: COLOR 12: PRINT "OFF": COLOR 15

END IF

IF LOAD(7) = true THEN

    LOCATE 18, 71: PRINT "ON "

ELSE

    LOCATE 18, 71: COLOR 12: PRINT "OFF": COLOR 15

END IF

IF LOAD(8) = true THEN

    LOCATE 19, 71: PRINT "ON "

ELSE

    LOCATE 19, 71: COLOR 12: PRINT "OFF": COLOR 15

END IF

LINE (440, 139)-(602, 267), 15, B

LINE (440, 152)-(602, 152), 15

LINE (545, 139)-(545, 267), 15

LINE (393, 68)-(604, 114), 15, B

LINE (507, 68)-(507, 114), 15

LINE (393, 84)-(604, 84), 15

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEFSNG A-Z

SUB MENUSETUP

LINE (21, 31)-(619, 339), 0, BF

LOCATE 5, 31: PRINT "SETUP PARMETER"

LINE (240, 70)-(350, 70), 14

LOCATE 7, 14: PRINT "DESCRIPTION PRIORITY Min.Off Max.Off \_  
Min.On"

LOCATE 8, 15: PRINT "OUT PUT LEVEL Time Time T\_  
ime"

FOR i = 9 TO 16

LOCATE i, 14: PRINT "OUT PUT"; i - 8

NEXT

LOCATE 18, 14: PRINT "CONTROL MODE AUTOMATIC"

LOCATE 19, 14: PRINT "UPPER DEMAND W"

LOCATE 20, 14: PRINT "LOWER DEMAND W"

LINE (98, 82)-(510, 226), 14, B

LINE (98, 234)-(335, 280), 14, B

LINE (200, 82)-(200, 226), 14

LINE (280, 82)-(280, 226), 14

LINE (350, 82)-(350, 226), 14

LINE (430, 82)-(430, 226), 14

LINE (98, 111)-(510, 111), 14

LINE (210, 234)-(210, 280), 14

LINE (98, 250)-(335, 250), 14

```
END SUB
```

```
DEFINT A-Z
```

```
SUB MINOFFTIME (c, D, TIMENOW%, TIMEOFF%(), TIMEON%(), MINOFF%())
```

```
IF TIMENOW% >= TIMEOFF%(c) + MINOFF%(c) THEN
```

```
    LOAD(c) = true
```

```
    LOCATE D, 71: PRINT "ON "
```

```
    TIMEON%(c) = TIMER / 60
```

```
END IF
```

```
END SUB
```

```
SUB MINONTIME (A, B, TIMENOW%, TIMEOFF%(), TIMEON%(), MINON%())
```

```
IF TIMENOW% >= TIMEON%(A) + MINON%(A) THEN
```

```
    LOAD(A) = false
```

```
    PRINT #1, A
```

```
    LOCATE B, 71: COLOR 12: PRINT "OFF": COLOR 15
```

```
    TIMEOFF%(A) = TIMER / 60
```

```
END IF
```

```
END SUB
```

```
SUB OPENFILE (WS&)
```

```
OPEN "DEMAND.DAT" FOR APPEND AS #2
```

```
DD$ = LEFT$(DATE$, 6) + RIGHT$(DATE$, 2): TT$ = LEFT$(TIME$, 5)
```

```
WRITE #2, DD$, TT$, WS&
```

```
CLOSE #2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END SUB
```

```
SUB paymoney
```

```
    dmax1% = 0
```

```
Gmon: CALL colortext(18, 23, 30, 18, 1, 10)
```

```
    LOCATE 18, 25: COLOR 10
```

```
    INPUT "WHAT 'S MONTH (01-12): ", mon$
```

```
    IF LEN(mon$) = 1 THEN
```

```
        mon$ = "0" + mon$
```

```
    END IF
```

```
    IF LEN(mon$) > 2 THEN
```

```
BEEP: LOCATE 18, 49: PRINT STRING$(LEN(mon$) + 2, " ")
```

```
GOTO Gmon
```

```
    END IF
```

```
    month% = VAL(mon$)
```

```
    IF month = 0 OR month% > 12 THEN GOTO Gmon
```

```
    IF month% = 1 THEN
```

```
        s$ = "JANUARY"
```

```
    ELSEIF month% = 2 THEN
```

```
        s$ = "FEBRUARY"
```

```
    ELSEIF month% = 3 THEN
```

```
        s$ = "MARCH"
```

```
    ELSEIF month% = 4 THEN
```

```
        s$ = "APRIL"
```

```
    ELSEIF month% = 5 THEN
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับอาจารย์และบุคลากรที่ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

s$ = "MAY"

        ELSEIF month% = 6 THEN

s$ = "JUNE"

        ELSEIF month% = 7 THEN

s$ = "JULY"

        ELSEIF month% = 8 THEN

s$ = "AUGUST"

        ELSEIF month% = 9 THEN

s$ = "SEPTEMBER"

        ELSEIF month% = 10 THEN

s$ = "OCTERBER"

        ELSEIF month% = 11 THEN

s$ = "NOVEMBER"

        ELSEIF month% = 12 THEN

s$ = "DECEMBER"

END IF

CLS : LINE (150, 91)-(485, 224), 15, B

CALL colortext(7, 37, LEN(s$) + 2, 7, 2, 0)

LOCATE 7, 38: COLOR 13: PRINT s$

CALL colortext(7, 37, LEN(s$) + 2, 7, 1, 15)

LOCATE 10,24:PRINT "DEMAND MAX :                WATT"

LOCATE 11,29:PRINT "( DATE IS :                )"

LOCATE 13,24:PRINT "ACCUMULATE ENERGY                W/H"

LOCATE 15,30:PRINT CHR$(228)+" MONEY                BAHTS"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OPEN "DMAX.DAT" FOR INPUT AS #datafile%

loop1: IF EOF(datafile%) THEN GOTO loop3

      INPUT #datafile%, day$, dmax, acs&

      IF mon$ = LEFT$(day$, 2) THEN GOTO loop2

      GOTO loop1

loop2:

      INPUT #datafile%, day$, dmax, ac&

      IF dmax > dmax1% THEN

dmax1% = dmax: dday$ = day$

      END IF

      IF mon$ < LEFT$(day$, 2) THEN

ac& = acs&: GOTO loop3

      END IF

      IF EOF(datafile%) THEN

ac& = acs&: GOTO loop3

      END IF

      GOTO loop2

loop3: COLOR 14

      LOCATE 10, 42: PRINT USING "#,###,###"; dmax1%

      LOCATE 11, 41: PRINT dday$

      LOCATE 13, 42: PRINT USING "#,###,###"; ac&

      IF ac& <= 35 THEN

baht& = 94

      ELSEIF ac& <= 150 THEN

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

baht& = 94 + (ac1& * 1.14)

    ELSEIF ac& <= 400 THEN

ac1& = ac& - 35: ac2& = ac& - 115

baht& = 94 + (ac1& * 1.14) + (ac2& * 2.22)

    ELSEIF ac& > 400 THEN

ac1& = ac& - 35: ac2& = ac& - 115: ac3& = ac& - 250

baht& = 94 + (ac1& * 1.14) + (ac2& * 2.22) + (ac3& * 2.53)

    END IF

    LOCATE 15, 42: PRINT USING "#,###,###"; baht&

    LOCATE 22, 32: COLOR 15: PRINT "<ESC-Exit to Prev>"

    CALL premenu("1")

    DO
CODE$ = INKEY$
    LOOP UNTIL CODE$ <> "" AND CODE$ = CHR$(27)
END SUB

SUB PLACECHARACTER (c$, x%, y%)

    DEF SEG = &HF000

    offset = 8 * ASC(c$) + &HFAGE

    FOR newy = 0 TO 7

        Bit = 128

        FOR newx = 0 TO 7

            IF (PEEK(offset + newy) AND Bit) > 0 THEN PSET (x% + newx, _
                                                                    y% + newy)

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของโรงเรียนที่ใช้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NEXT newx

NEXT newy

DEF SEG

END SUB

SUB plotstring (lin%, col%, nchar%, s$)

  x1% = col% * 8 - 8

  x2% = (col% + nchar% - 1) * 8

  y1% = lin% * 14 - 14

  y2% = lin% * 14 - 1

  L = LEN(s$): c = INT((x2% - x1%) / 8)

  IF L > c THEN s$ = LEFT$(s$, c)

  x% = x1% + (x2% - x1% + 1) / 2 - 8 * L / 2 + 4

  y% = y1% + (y2% - y1% - 4) / 2

  WHILE s$ <> ""

    c$ = LEFT$(s$, 1)

    s$ = MID$(s$, 2)

    IF x% >= 0 THEN CALL PLACECHARACTER(c$, x%, y%)

    x% = x% + 8

  WEND

END SUB

DEFSNG A-Z

SUB premenu (type$)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้การคุ้มครองของสำนักงานคณะกรรมการกฤษฎีกา  
 ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

s3$ = " F3 Run ": s4$ = " F10 Quit! "

SELECT CASE type$

CASE "1"

  LINE (20, 15)-(615, 330), 15, B

  LINE (23, 17)-(612, 328), 15, B

  .CALL colortext(24, 35, 12, 24, 2, 6)

  CALL plotstring(24, 35, 12, s4$)

  GOTO ends

CASE "2"

  clr% = 10: y1% = 35: y2% = 58: l1% = 5: s3$ = s4$

  GOSUB smenu

  GOTO ends

CASE "3"

  clr% = 14: y1% = 35: y2% = 58: l1% = 5: s2$ = s3$: s3$ = s4$

  GOSUB smenu

  GOTO ends

CASE "4"

  clr% = 7: y1% = 27: y2% = 44: l1% = 4

  GOSUB smenu

  LOCATE 2, 58: PRINT s4$

  CALL INVERTTEXT(2, 58, 5, 2)

  GOTO ends

END SELECT

smenu:  COLOR 10

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของภาควิชาการศึกษาคณะศึกษาศาสตร์ มหาวิทยาลัยบูรพา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LINE (15, 30)-(625, 340), 15, B

PAINT (15, 10), clr%, 15

LOCATE 2, 10: PRINT s1$

LOCATE 2, y1%: PRINT s2$

LOCATE 2, y2%: PRINT s3$

CALL INVERTTEXT(2, 10, 4, 2)

CALL INVERTTEXT(2, y1%, 4, 2)

CALL INVERTTEXT(2, y2%, 11%, 2)

RETURN

ends:    COLOR 15

END SUB

DEFINT A-Z

SUB PRINTS

LINE (21, 31)-(619, 319), 0, BF

LOCATE 6, 6: INPUT "START DATE :", start$

LOCATE 7, 8: INPUT "END DATE :", ends$

OPEN "I", #2, "DEMAND.DAT"

LPRINT CHR$(27); "4";

LPRINT "REPORT OF DEMAND VALUE DESIGN BY SIRICHAJ DILOGRATPIJIT "

LPRINT CHR$(27); "5"

LPRINT TAB(5); "DATE    TIME    KW\HR  0    200    400    600    800    _
                                     1000  1200  1400"

66 IF EOF(2) THEN 122

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF start$ = LEFT$(D$, LEN(start$)) THEN 101

GOTO 66

101 LPRINT TAB(2); D$; TAB(12); T$; TAB(19); W; TAB(27); STRING$_
                                         (W / 33, 223)

IF EOF(2) THEN 122

INPUT #2, D$, T$, W

IF ends$ = LEFT$(D$, LEN(ends$)) THEN 112

GOTO 101

112 LPRINT TAB(2); D$; TAB(12); T$; TAB(19); W; TAB(27); STRING$_
                                         (W / 33, 223)

IF EOF(2) THEN 122

INPUT #2, D$, T$, W

IF ends$ <> LEFT$(D$, LEN(ends$)) THEN 122

GOTO 112

122 CLOSE #2

END SUB

SUB restoretextwindow (leftcol%, topline%)

x1% = leftcol% * 8 - 8

y1% = topline% * 14 - 14

PUT (x1%, y1%), gwindow!, PSET

ERASE gwindow!

END SUB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SUB restorevaule (upper%, LOWER%)

    CALL colortext(15, 40, 37, 19, 2, 0)

    LOCATE 16, 40: PRINT " DO YOU WANT TO USE DEFAULT VAULES ?"

    LOCATE 18, 50: COLOR 15: PRINT "< Yes > or < No >"

    LOCATE 18, 52: COLOR 3: PRINT "Y"

    LOCATE 18, 63: PRINT "N": COLOR 15

    LINE (315, 198)-(604, 263), 7, B

    LINE (315, 230)-(604, 230), 7

    CALL INVERTTEXT(15, 40, 37, 19)

    DO

DO
    onechar$ = INKEY$
    LOOP WHILE onechar$ = ""
    user$ = UCASE$(onechar$)
    IF user$ = "Y" THEN
        DO
            datafile% = FREEFILE
            ON ERROR GOTO operror

            badopen% = false%

            OPEN "data.ini" FOR INPUT AS #datafile%

            ON ERROR GOTO 0

            IF badopen% THEN

                LOCATE 21, 37: COLOR 12

                PRINT "error opening file"

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOOP WHILE badopen%

INPUT #datafile%, upper, LOWER

FOR i = 9 TO 16

    INPUT #datafile%, M(i, 38), M(i, 48), M(i, 58)

NEXT

CLOSE #datafile%

CALL colortext(15, 40, 37, 19, 2, 0)

ELSEIF user$ = "N" THEN

    CALL colortext(15, 40, 37, 19, 2, 0)

    EXIT DO

ELSEIF user$ <> "Y" AND user$ <> "N" THEN

    BEEP

END IF

    LOOP UNTIL user$ = "Y" OR user$ = "N"

    CALL restoretextwindow(40, 15)

END SUB

SUB savetextwindow (leftcol%, topline%, rightcol%, bottomline%)

x1% = leftcol% * 8 - 8

x2% = rightcol% * 8

y1% = topline% * 14 - 14

y2% = bottomline% * 14 - 1

arraysize% = COMPUTEARRAYSIZE(x1%, y1%, x2%, y2%)

REDIM gwindow!(arraysize%)

GET (x1%, y1%)-(x2%, y2%), gwindow!

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของอาจารย์ผู้สอนเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END SUB

SUB savevaule (upper%, LOWER%)

CALL colortext(15, 40, 36, 19, 2, 0)

LOCATE 16, 40: PRINT " DO YOU WANT TO SAVE THESE VAULES ?"

LOCATE 18, 50: COLOR 15: PRINT "< Yes > or < No >"

LOCATE 18, 52: COLOR 3: PRINT "Y"

LOCATE 18, 63: PRINT "N": COLOR 15

LINE (315, 198)-(596, 263), 7, B

LINE (315, 230)-(596, 230), 7

CALL INVERTTEXT(15, 40, 36, 19)

DO

DO

onechar\$ = INKEY\$

LOOP WHILE onechar\$ = ""

user\$ = UCASE\$(onechar\$)

IF user\$ = "Y" THEN

DO

datafile% = FREEFILE

ON ERROR GOTO openerror

badopen% = .false%

OPEN "data.ini" FOR OUTPUT AS #datafile%

ON ERROR GOTO 0

IF badopen% THEN

LOCATE 21, 37: COLOR 12

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PRINT "error opening file"

END IF

LOOP WHILE badopen%

WRITE #datafile%, upper, LOWER

FOR i = 9 TO 16

    WRITE #datafile%, M(i, 38), M(i, 48), M(i, 58)

NEXT

CLOSE #datafile%

CALL colortext(15, 40, 36, 19, 2, 0)

ELSEIF user$ = "N" THEN

    CALL colortext(15, 40, 36, 19, 2, 0)

EXIT DO

ELSEIF user$ <> "Y" AND user$ <> "N" THEN

    BEEP

END IF

LOOP UNTIL UCASE$(onechar$) = "Y" OR UCASE$(onechar$) = "N"

CALL restoretextwindow(40, 15)

END SUB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การไฟฟ้านครหลวง

อัตราค่าไฟฟ้าใหม่

เริ่มใช้ตั้งแต่วันที่ 1 ธันวาคม 2534

ประเภทที่ 1 มิเตอร์อัตราต่ำ

1.1 ปริมาณการใช้พลังงานไฟฟ้าไม่เกิน 150 หน่วยต่อเดือน

5 หน่วยแรกหรือน้อยกว่า	5.00 บาท
10 หน่วยต่อไป หน่วยละ	0.70 "
10 " " "	0.90 "
10 " " "	1.17 "
65 " " "	1.58 "
50 " " "	1.68 "
250 " " "	2.22 "

เกินกว่า 400 หน่วยขึ้นไป

ค่าไฟฟ้าค่าเสาคือเดือนละ 5.00 บาท

1.2 ปริมาณการใช้พลังงานไฟฟ้าเกินกว่า 150 หน่วยต่อเดือนขึ้นไป

35 หน่วยแรกหรือน้อยกว่า	89.00 บาท
115 หน่วยต่อไป หน่วยละ	1.14 "
250 " " "	2.22 "
เกินกว่า 400 หน่วยขึ้นไป	2.53 "
ค่าไฟฟ้าค่าเสาคือเดือนละ	89.00 บาท

\* ประเภทที่ 2 ปริมาณการใช้พลังงานไฟฟ้า (ต่ำกว่า 30 กิโลวัตต์)

35 หน่วยแรกหรือน้อยกว่า	94.00 บาท
115 หน่วยต่อไป หน่วยละ	1.14 "
250 " " "	2.22 "
เกินกว่า 400 หน่วยขึ้นไป	2.53 "
ค่าไฟฟ้าค่าเสาคือเดือนละ	94.00 บาท

ประเภทที่ 3 ปริมาณการใช้พลังงานไฟฟ้า (30-1,999 กิโลวัตต์)

3.1 ปริมาณการใช้พลังงานไฟฟ้าเฉลี่ย 3 เดือนสุดท้ายไม่เกิน 355,000 หน่วยต่อเดือน

3.1.1 ระดับแรงดันไฟฟ้า 69 กิโลโวลท์ขึ้นไป	
ค่าความต้องการพลังไฟฟ้า: กิโลวัตต์ละ	188.00 บาท
ค่าพลังงานไฟฟ้า : หน่วยละ	1.03 "
3.1.2 ระดับแรงดันไฟฟ้า 12-24 กิโลโวลท์ขึ้นไป	
ค่าความต้องการพลังไฟฟ้า: กิโลวัตต์ละ	210.00 บาท
ค่าพลังงานไฟฟ้า : หน่วยละ	1.07 "
3.1.3 ระดับแรงดันไฟฟ้าต่ำกว่า 12 กิโลโวลท์	
ค่าความต้องการพลังไฟฟ้า: กิโลวัตต์ละ	237.00 บาท
ค่าพลังงานไฟฟ้า : หน่วยละ	1.10 "
ค่าไฟฟ้าค่าเสาคือ : คิดร้อยละ 70 ของความต้องการพลังไฟฟ้าที่ส่งกลับในรอบ 12 เดือน	

3.2 ปริมาณการใช้พลังงานไฟฟ้าเฉลี่ย 3 เดือนสุดท้ายเกินกว่า 355,000 หน่วยต่อเดือน (เริ่มใช้ตั้งแต่วันที่ 1 ธันวาคม 2535)

3.2.1 ระดับแรงดันไฟฟ้า 69 กิโลโวลท์ขึ้นไป	
ค่าความต้องการพลังไฟฟ้า	
เวลา 18.30-21.30 น. : กิโลวัตต์ละ	240.00 บาท
เวลา 08.00-18.30 น. : " "	32.00 "
(คิดเฉพาะความต้องการพลังไฟฟ้าในส่วนที่เกิน ON-Peak)	
เวลา 21.30-08.00 น. : ไม่คิดค่าความต้องการพลังไฟฟ้า	
ค่าพลังงานไฟฟ้า	
ทุกช่วงเวลา : หน่วยละ	1.03 บาท
3.2.2 ระดับแรงดันไฟฟ้า 12-24 กิโลโวลท์	
ค่าความต้องการพลังไฟฟ้า	
เวลา 18.30-21.30 น. : กิโลวัตต์ละ	305.00 บาท
เวลา 08.00-18.30 น. : " "	63.00 "
(คิดเฉพาะความต้องการพลังไฟฟ้าในส่วนที่เกิน ON-Peak)	
เวลา 21.30-08.00 น. : ไม่คิดค่าความต้องการพลังไฟฟ้า	
ค่าพลังงานไฟฟ้า	
ทุกช่วงเวลา : หน่วยละ	1.07 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่จากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3 ระดับแรงดันไฟฟ้าต่ำกว่า 12 กิโลโวลต์

ค่าความต้องการหลังไฟฟ้า

เวลา 18.30-21.30 น. : กิโลวัตต์ละ 356.00 บาท

เวลา 08.00-18.30 น. : " 75.00 "

(คิดเฉพาะความต้องการหลังไฟฟ้าในส่วนที่เกิน ON-Peak)

เวลา 21.30-08.00 น. : ไม่มีค่าความต้องการหลังไฟฟ้า

ค่าพลังงานไฟฟ้า

ทุกช่วงเวลา : หน่วยละ 1.10 บาท

ค่าไฟฟ้าค่าสุด : คิดร้อยละ 70 ของค่าความต้องการหลัง

ไฟฟ้าสูงสุดรวมในรอบ 12 เดือน

ค่าเช่าเวอร์เนอเบิลเตอร์ : เหมือนเดิม

### ประเภทที่ 4 กิจการขนาดเล็ก (2,000 กิโลวัตต์ขึ้นไป)

#### 4.1 ระดับแรงดันไฟฟ้า 69 กิโลโวลต์ขึ้นไป

ค่าความต้องการหลังไฟฟ้า

เวลา 18.30-21.30 น. : กิโลวัตต์ละ 240.00 บาท

เวลา 08.00-18.30 น. : " 32.00 "

(คิดเฉพาะความต้องการหลังไฟฟ้าในส่วนที่เกิน ON-Peak)

เวลา 21.30-08.00 น. : ไม่มีค่าความต้องการหลังไฟฟ้า

ค่าพลังงานไฟฟ้า

ทุกช่วงเวลา : หน่วยละ 1.03 บาท

#### 4.2 ระดับแรงดันไฟฟ้า 12-24 กิโลโวลต์

ค่าความต้องการหลังไฟฟ้า

เวลา 18.30-21.30 น. : กิโลวัตต์ละ 305.00 บาท

เวลา 08.00-18.30 น. : " 63.00 "

(คิดเฉพาะความต้องการหลังไฟฟ้าในส่วนที่เกิน ON-Peak)

เวลา 21.30-08.00 น. : ไม่มีค่าความต้องการหลังไฟฟ้า

ค่าพลังงานไฟฟ้า

ทุกช่วงเวลา : หน่วยละ 1.07 บาท

ค่าไฟฟ้าค่าสุด : คิดร้อยละ 70 ของค่าความต้องการหลัง

ไฟฟ้าสูงสุดรวมในรอบ 12 เดือน

ค่าเช่าเวอร์เนอเบิลเตอร์ : เหมือนเดิม

### ประเภทที่ 5 กิจการเฉพาะอุตสาหกรรม (โรงงาน)

#### 5.1 อีกรูปกัก

##### 5.1.1 ระดับแรงดันไฟฟ้า 69 กิโลโวลต์ขึ้นไป

ค่าความต้องการหลังไฟฟ้า: กิโลวัตต์ละ 236.00 บาท

ค่าพลังงานไฟฟ้า : หน่วยละ 1.03

##### 5.1.2 ระดับแรงดันไฟฟ้า 12-24 กิโลโวลต์ขึ้นไป

ค่าความต้องการหลังไฟฟ้า: กิโลวัตต์ละ 274.00 บาท

ค่าพลังงานไฟฟ้า : หน่วยละ 1.07

##### 5.1.3 ระดับแรงดันไฟฟ้าต่ำกว่า 12 กิโลโวลต์

ค่าความต้องการหลังไฟฟ้า: กิโลวัตต์ละ 296.00 บาท

ค่าพลังงานไฟฟ้า : หน่วยละ 1.10

ค่าไฟฟ้าค่าสุด : คิดร้อยละ 70 ของความต้องการหลัง

ไฟฟ้าที่สูงสุดรวมในรอบ 12 เดือน

ค่าเช่าเวอร์เนอเบิลเตอร์ : เหมือนเดิม

#### 5.2 อัตราค่ารวมระยะเวลาของวัน

##### 5.2.1 ระดับแรงดันไฟฟ้า 69 กิโลโวลต์ขึ้นไป

ค่าความต้องการหลังไฟฟ้า

เวลา 18.30-21.30 น. : กิโลวัตต์ละ 240.00 บาท

เวลา 08.00-18.30 น. : " 32.00 "

(คิดเฉพาะความต้องการหลังไฟฟ้าในส่วนที่เกิน ON-Peak)

เวลา 21.30-08.00 น. : ไม่มีค่าความต้องการหลังไฟฟ้า

ค่าพลังงานไฟฟ้า

ทุกช่วงเวลา : หน่วยละ 1.03 บาท

##### 5.2.2 ระดับแรงดันไฟฟ้า 12-24 กิโลโวลต์

ค่าความต้องการหลังไฟฟ้า

เวลา 18.30-21.30 น. : กิโลวัตต์ละ 305.00 บาท

เวลา 08.00-18.30 น. : " 63.00 "

(คิดเฉพาะความต้องการหลังไฟฟ้าในส่วนที่เกิน ON-Peak)

เวลา 21.30-08.00 น. : ไม่มีค่าความต้องการหลังไฟฟ้า

ค่าพลังงานไฟฟ้า

ทุกช่วงเวลา : หน่วยละ 1.07 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 ระดับแรงดันไฟฟ้าต่ำกว่า 12 กิโลโวลต์  
 ค่าความถี่ของการหลังไฟฟ้า  
 เวลา 18.30-21.30 น. : กิโลวัตต์ละ 356.00 บาท  
 เวลา 08.00-18.30 น. : " 73.00 "  
 (คิดเฉพาะความถี่ของการหลังไฟฟ้าในส่วนที่เกิน ON-Peak)  
 เวลา 21.30-08.00 น. : ไม่คิดค่าความถี่ของการหลัง-  
 ไฟฟ้า

**ค่าพลังงานไฟฟ้า**

ทุกช่วงเวลา : หน่วยละ 1.10 บาท  
 ค่าไฟฟ้าค่าสุด : คิดร้อยละ 70 ของค่าความถี่ของการหลัง  
 ไฟฟ้าสูงสุดรวมในรอบ 12 เดือน  
 ค่าทราเวลอร์มัลเตอร์ : เหมือนเดิม

**ประเภทที่ 7 ส่วนนำเพื่อการเกษตร**

100 หน่วยแรกหรือน้อยกว่า 117.00 บาท  
 เกินกว่า 100 หน่วยขึ้นไป หน่วยละ 1.17 "  
 ค่าไฟฟ้าค่าสุด : เดือนละ 117.00 บาท

**ข้อกำหนดเกี่ยวกับการปรับค่าไฟฟ้าโดยอัตโนมัติ**

หากค่าใช้จ่ายที่ไม่อยู่ในความควบคุมของการไฟฟ้า  
 เช่น ราคาเชื้อเพลิงที่ใช้ในการผลิตไฟฟ้า ฯลฯ เปลี่ยนแปลงไป  
 ตั้งแต่หน่วยละ 2 สลึงค์ การไฟฟ้านครหลวงจะปรับค่าไฟฟ้า  
 เพิ่มขึ้นหรือลดลง โดยส่งจำนวนเงินไว้ในใบเสร็จรับเงินค่า-  
 ไฟฟ้าด้วย

**ประเภทที่ 6 ส่วนราชการและองค์กรที่ไม่แสวงหากำไร**

6.1 ระดับแรงดันไฟฟ้า 69 กิโลโวลต์ขึ้นไป  
 10,000 หน่วยแรกหรือน้อยกว่า เป็นเงิน 14,800.00 บาท  
 เกินกว่า 10,000 หน่วยขึ้นไป หน่วยละ 1.48 "

6.2 ระดับแรงดันไฟฟ้า 12-24 กิโลโวลต์  
 300 หน่วยแรกหรือน้อยกว่า เป็นเงิน 495.00 บาท  
 เกินกว่า 300 หน่วยขึ้นไป หน่วยละ 1.65 "

6.3 ระดับแรงดันไฟฟ้าต่ำกว่า 12 กิโลโวลต์  
 10 หน่วยแรกหรือน้อยกว่า เป็นเงิน 18.70 บาท  
 เกินกว่า 10 หน่วยขึ้นไป หน่วยละ 1.87 "

**ค่าไฟฟ้าค่าสุด :**

ระดับแรงดันไฟฟ้า 69 กิโลโวลต์ขึ้นไป เดือนละ 14,800.00 บาท  
 ระดับแรงดันไฟฟ้า 12-24 กิโลโวลต์ขึ้นไป " 495.00 "  
 ระดับแรงดันไฟฟ้าต่ำกว่า 12 กิโลโวลต์ " 18.70 "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- การใช้งาน Z-80 , ศูนย์ภาษาคอมพิวเตอร์
- MCS 51 ไมโครคอนโทรลเลอร์แบบซีพเคียว , สมยศ ทุ่งามแปลง , วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์ , ฉบับที่ 93 , 2532
- เบ็ดโรภไมโครคอนโทรลเลอร์ , ตอน 1 และ ตอนจบ , ศิริชัย ดิลกรัตนพิจิตร / นรศักดิ์ ตั้งภาณุจนตรี , วารสารเซมิคอนดักเตอร์ , ฉบับที่ 114 และ 115 , 2535
- JAZZ-31 SINGLE BOARD MICROCONTROLLER USER MANUAL v1.0 , SILA RESEARCH CO., LTD.
- THE 8051 MICROCONTROLLER , KENNETH J. AYALA
- Quickbasic, The Complete Reference, Steven Nameroff; Osborne McGraw-Hill



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิติกรรมประกาศ

การที่ปริญญานิพนธ์นี้สามารถสำเร็จลงได้ด้วยดีนั้น ต้องขอขอบพระคุณอาจารย์ วิทยา ทิพย์สุวรรณพร เป็นอย่างสูงที่ได้ช่วยแก้ไขข้อเสนอนั้นที่สำคัญและมีประโยชน์ตลอดมา และขอขอบพระคุณอาจารย์ทุกท่านที่ได้กรุณาให้คำแนะนำและข้อมูลที่มีประโยชน์ ขณะเดียวกันก็ต้องขอขอบใจเพื่อนทุกคนที่ได้ให้ความร่วมมือช่วยเหลือต่าง ๆ จนปริญญานิพนธ์นี้สำเร็จได้ ขอขอบคุณเพื่อนร่วมงานทุกคนด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้