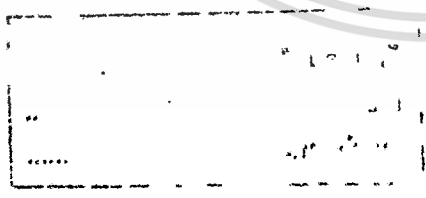




การจำลองการวัดปริมาณน้ำมันในปั้มน้ำมันด้วยไมโครคอมพิวเตอร์

SIMULATION OF OIL QUANTITY MEASUREMENT IN OIL PUMP WITH MICROCOMPUTER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาคตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2537

ปริญญาโท ปีการศึกษา 2537

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขาวิชา วิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การจำลองการวัดปริมาณน้ำมันในปั้มน้ำมันด้วยไมโครคอมพิวเตอร์

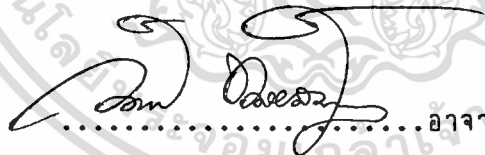
SIMULATION OF OIL QUANTITY MEASUREMENT IN OIL PUMP WITH
MICROCOMPUTER

ผู้จัดทำ

1. นายกิตติพงศ์ มลิตอง 35.102087

2. นายไทรงาม หอมขจร 35.102095

3. นายสมรินทร์ ชูฉัตร 35.102127



.....อาจารย์ที่ปรึกษา

(ผศ. วิชา ทิพย์สุวรรณพร)

การจำลองการวัดปริมาณน้ำมันในปั๊มน้ำมันด้วยไมโครคอมพิวเตอร์

นางกิติพงศ์ มลิกทอง

นายไทรงาม หอมขจร

นายอมรินทร์ ชูฉัตร

อาจารย์ที่ปรึกษา

ผศ. วิชา ทิพย์สุวรรณพร

ปีการศึกษา 2537

บทคัดย่อ

โครงการที่จะกล่าวถึงในปริิถยานี้เป็นการจำลองการวัดปริมาณน้ำมันในปั๊มน้ำมันด้วยไมโครคอมพิวเตอร์ซึ่งอาศัยวงจรกำเนิดสัญญาณพัลส์เพื่อจำลองสัญญาณเข้าที่พุกที่ได้จากหัวจ่ายน้ำมันตั้งแต่ 1-8 หัวจ่าย ในปัจจุบันโดยทั่วไปน้ำมันที่มีจำหน่ายในท้องตลาดมี 4 ชนิดคือ เบนซินพิเศษ เบนซินธรรมดา เบนซินไร้สารตะกั่ว และน้ำมันดีเซล สัญญาณที่จำลองขึ้นดังกล่าวนี้จะถูกตรวจจับด้วยไมโครคอนโทรลเลอร์ จากนั้นจะส่งสัญญาณไปยังไมโครคอมพิวเตอร์ โดยผ่านสายสัญญาณ RS-232 เพื่อให้ไมโครคอมพิวเตอร์ทำการนับจำนวนพัลส์, คำนวณราคา, คำนวณปริมาณที่จ่ายออกไปแต่ละหัวจ่าย, คำนวณปริมาณรวมทั้งหมดที่ได้จำหน่ายในแต่ละวันแต่ละหัวจ่ายและแสดงผลยังหน้าจอหรือเครื่องพิมพ์ตามแต่ผู้ใช้ต้องการ นอกจากนี้ไมโครคอมพิวเตอร์ยังทำหน้าที่ในการรับข้อมูลการเปลี่ยนแปลงเกี่ยวกับราคาน้ำมัน เวลาขณะใช้งานและขนาดของเครื่องมือวัด โดยเฉพาะขนาดของเครื่องมือวัดจะมีการเปลี่ยนแปลงเมื่อมีการติดตั้งปั๊มน้ำมัน โดยทั่วไปปั๊มน้ำมันจะใช้เครื่องมือวัดอัตราไหลแบบกักตัน ที่มีเส้นผ่าศูนย์กลาง 1.5 นิ้ว มีอัตราวัด 3-10 พัลส์ต่อลิตร ซึ่งผู้ใช้สามารถป้อนค่าได้โดยผ่านแป้นพิมพ์

SIMULATION OF OIL QUANTITY MEASUREMENT IN OIL PUMP WITH MICROCOMPUTER

Mr.KITIPHONG MALITHONG

Mr.SAINGAM HOMKACHON

Mr.AMARIN CHUCHATTHA

ADVISOR

Asst.Prof.WIDHAYA THIPSUWANPORN

ACADEMIC YEAR 1994

Abstract

About this project is simulation of oil measurement in oil pump with microcomputer by using pulse generator circuit to simulate the output signal from 1 to 8 nozzle. There are 4 types of oil generally special benzine, common benzine, unlead benzine and diesel. The simulate signal is detected by microcontroller and send data to microcomputer. The microcomputer will count amount of pulse, calculate cost, calculate oil quantity of each nozzle, calculate total cost and total oil quantity each day and show the necessary data on monitor or printer depend on selection of user. Other duty of microcomputer are receive the changing data about oil cost of each day, operating time, size of flow measurement. Size of flow measurement change when installed the oil pump. Most of oil pump are use turbine flow meter generally diameter is 1.5", flow rate measure is 3-10 pulse/litre. The operator can change this data easy by editing on keyboard on Define new data mode.

สารบัญ

หน้า

| | |
|--|----|
| บทนำ | 1 |
| บทที่ 1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51 | 3 |
| 1.1 โครงสร้างของ 8051 | 4 |
| 1.2 การจัดการหน่วยความจำของ 8051 | 7 |
| 1.3 สถาปัตยกรรมของ 8051 | 9 |
| บทที่ 2 การออกแบบตัวควบคุม 8051 | 30 |
| 2.1 Microcontroller Specification | 30 |
| 2.2 Microcontroller Design | 32 |
| 2.3 หน่วยความจำภายนอกและการต่อที่สหน่วยความจำภายนอก | 32 |
| 2.4 วงจร Reset และ Clock | 33 |
| 2.5 Expanding I/O | 34 |
| 2.6 Memory-Mapped I/O | 35 |
| 2.7 Part speed | 36 |
| 2.8 การประยุกต์ใช้งานไอซี 8051 | 37 |
| บทที่ 3 ทฤษฎีการสื่อสารข้อมูลและการจัดการข้อมูล | 41 |
| 3.1 องค์ประกอบเบื้องต้นในการสื่อสารข้อมูล | 41 |
| 3.2 รูปแบบการถ่ายโอนข้อมูล | 41 |
| 3.3 MCS-51 กับการติดต่อสื่อสารทางพอร์ตอนุกรม | 46 |
| 3.4 โปรแกรมการจัดการข้อมูล | 51 |
| 3.4.1 โปรแกรมเกี่ยวกับการจัดเก็บข้อมูล | 51 |
| 3.4.2 โปรแกรมเกี่ยวกับ Database | 53 |
| บทที่ 4 การจำลองการวัดอัตราไหลและการทดลองโครงงาน | 54 |
| 4.1 การจำลองการวัดอัตราไหล | 54 |
| 4.2 การทดสอบโครงงาน | 57 |
| 4.2.1 อุปกรณ์ที่ใช้ | 57 |
| 4.2.2 วิธีการทดลอง | 57 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|--|-----|
| 4.2.3 ผลการทดลอง | 57 |
| บทที่ 5 โปรแกรมประกอบการทดลอง | 80 |
| 5.1 Turbo-C Flowchart | 80 |
| 5.2 C-Program | 82 |
| 5.3 Assembly Flowchart | 126 |
| 5.4 Assembly Program | 130 |
| บทที่ 6 สรุปผลโครงการ | 148 |
| ภาคผนวก | 150 |
| รูป ก-1 แสดงภาพ Flow Simulator ด้านหน้า | 150 |
| รูป ก-2 แสดงภาพ Flow Simulator ด้านหลัง | 151 |
| รูป ก-3 แสดงภาพ Flow simulator ด้านบน | 152 |
| รูป ก-4 แสดงภาพ Counter Console ด้านบน | 153 |
| รูป ก-5 แสดงภาพ Counter Console ด้านหลัง | 154 |
| รูป ก-6 แสดงภาพโดยรวมของการทดลองโครงการ | 155 |
| กิตติกรรมประกาศ | 156 |
| บรรณานุกรม | 157 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ในปัจจุบันธุรกิจปั้มน้ำมันได้ขยายตัวอย่างรวดเร็ว ตามสองข้างถนนแต่ละสายทั้งในกรุงเทพมหานครและในต่างจังหวัดจะเห็นว่าปั้มน้ำมันมากมายปั้มน้ำมันเหล่านี้มีระบบการวัดปริมาณน้ำมันที่แตกต่างกันไป บ้างก็ใช้ระบบวัดแบบ MECHANIC บ้างก็ใช้การวัดแบบ DIGITAL ซึ่งเป็นระบบที่เริ่มจะมีใช้งานมากยิ่งขึ้นในประเทศไทย ถึงแม้ว่าจะได้มีการพัฒนาถึงขั้นนี้แล้ว จะพบว่าไม่มีผู้คิดประดิษฐ์เครื่องมือวิเคราะห์ข้อมูลที่ได้จากการวัดปริมาณน้ำมันเหล่านั้น ทำให้การประกอบธุรกิจปั้มน้ำมันเกิดความล่าช้า กล่าวคือจะต้องมีพนักงานหรือเจ้าของปั้มน้ำมันนำข้อมูลที่ได้จากเครื่องวัดเหล่านั้นมาทำการวิเคราะห์ เช่นจะทราบปริมาณการจ่ายน้ำมันในแต่ละวันก็ต้องใช้เวลาในการรวบรวมข้อมูลซึ่งทำให้เกิดความล่าช้า

ในแต่ละวันราคาน้ำมันจะเปลี่ยนแปลงไป ที่เป็นเช่นนี้เพราะระบบตลาดน้ำมันในประเทศไทยได้ปล่อยให้ราคาน้ำมันลอยตัว กล่าวคือตลาดการค้าน้ำมันปล่อยให้มีการแข่งขันกันอย่างสมบูรณ์ ถึงแม้ว่าการกระทำในลักษณะนี้จะเป็นประโยชน์ต่อกลุ่มลูกค้าก็ตาม แต่นำมาซึ่งความลำบากของลูกค้านายย่อย อาจจะสรุปปัญหาต่างๆ ได้ดังนี้

1. เมื่อราคาน้ำมันเปลี่ยนแปลงไป จะต้องเปลี่ยนแปลงราคาน้ำมันที่ขาย (บาท/ลิตร) ที่อุปกรณ์การวัดปริมาณน้ำมัน เมื่อในแต่ละปั้มน้ำมันมีช่องบริการและขายน้ำมันหลายชนิด แต่ละชนิดมีราคาไม่เท่ากัน ในการป้อนข้อมูลเกี่ยวกับราคาน้ำมันดังกล่าวอาจก่อให้เกิดความผิดพลาดและล่าช้าได้
2. ปริมาณสะสมของน้ำมันที่จ่ายแต่ละ วัน/เดือน/ปี ต้องอาศัยการบันทึกทางบัญชี ทำให้เกิดความผิดพลาดและล่าช้าได้
3. ราคายกจากการขายน้ำมันแต่ละชนิด แต่ละครั้ง ต้องมีการบันทึกทางบัญชีทำให้เกิดความผิดพลาดและล่าช้าได้

จากปัญหาดังกล่าวจำเป็นต้องจัดหาอุปกรณ์ที่สามารถวัด บันทึก และคำนวณปริมาณน้ำมันและราคาน้ำมันในแต่ละวันได้ ซึ่งอุปกรณ์ดังกล่าวได้ใช้ Micro Computer มาใช้งาน โดยวิธีการนำเอาข้อมูลมาวิเคราะห์ ซึ่งต้องประกอบด้วยส่วนต่างๆ ดังนี้

- ส่วนที่ 1 ชุดตรวจวัดปริมาณน้ำมัน ซึ่งจะอาศัยอุปกรณ์วัดอัตราไหลที่มี OUTPUT เป็น Pulse แต่ที่นี้ใช้การจำลองสัญญาณจากเครื่องวัดอัตราไหลดังกล่าว
- ส่วนที่ 2 ชุดบันทึกและคำนวณ โดยนำสัญญาณไฟฟ้าที่ได้จากชุดตรวจวัดปริมาณน้ำมันมาคำนวณและบันทึกไว้เพื่อส่งข้อมูลไปยัง Micro Computer ส่วนที่ 2 นี้ อุปกรณ์ที่ใช้คือ Micro

controller (MCS-51)

ส่วนที่ 3 Micro Computer ทำหน้าที่นำข้อมูลจาก MCS-51 มาวิเคราะห์แสดง
ผลตลอดจนบันทึกเพื่อนำข้อมูลเสนอทาง Printer



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51

Single Chip Microcontroller system-51 family Architectural

1.0 บทนำ

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (Single Chip Microcontroller) คือไมโครคอนโทรลเลอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชิพเดี่ยวเหมาะสำหรับงานควบคุมอุปกรณ์แบบอัตโนมัติ เพราะผู้ใช้สามารถเขียนโปรแกรมควบคุมการทำงานได้ตามต้องการ ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51 หรือ MCS-51 อันได้แก่เบอร์ 8051 และ 8052 ซึ่งมีโครงสร้างและชุดคำสั่งแตกต่างกันเล็กน้อย

MCS-51 ผลิตโดยบริษัท Intel มีการทำงานเป็นแบบ 8 บิตหมายความว่าส่วนที่มีหน้าที่ในการคำนวณ (Arithmetic Logic Unit, ALU) จะทำงานสูงสุดทีละ 8 บิต

MCS-51 มีข้อดีดังนี้

- สามารถนำเอาข้อมูลมา AND, OR หรือทำ Complement ทั้งแบบทีละ 8 บิตและ 1 บิต
- สามารถใช้กับหน่วยความจำสำหรับโปรแกรม (Program Memory) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับเก็บชุดคำสั่งที่จะให้ MCS-51 ทำงานได้สูงสุด 64 กิโลไบต์ (kilobyte) ทำให้เขียนโปรแกรมควบคุมการทำงานได้มาก
- สามารถต่อกับหน่วยความจำสำหรับข้อมูล (Data Memory) ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการทำงานของโปรแกรมได้สูงสุด 64 กิโลไบต์
- ใน 8051 และ 8751 มีหน่วยความจำสำหรับโปรแกรมจำนวน 4 กิโลไบต์ (ใน 8052 และ 8752 มีหน่วยความจำสำหรับโปรแกรมจำนวน 8 กิโลไบต์) อยู่ภายในวงจรรวมทำให้ไม่ต้องต่อหน่วยความจำสำหรับโปรแกรมอยู่ภายนอก ระบบรวมทั้งหมดจึงมีขนาดเล็กและสัญญาณรบกวนจากภายนอกจะทำให้ MCS-51 ทำงานผิดพลาดได้ยาก
- มีพอร์ตแบบขนาน (Parallel Port) สำหรับข้อมูลเข้าและออกจำนวน 32 บิต ที่ข้อมูลแต่ละบิตเป็นอิสระต่อกัน
- มีวงจร Timer/Counter ขนาด 16 บิต 2 ชุด (8052 มี 3 ชุด) ที่ทำงานในโหมดต่างๆได้ถึง 4 โหมด
- มี Universal Asynchronous Receiver Transmitter (UART) สำหรับรับ-ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลอนุกรม (Serial) แบบ Full duplex ที่สามารถเลือกรูปแบบการรับส่งได้ 4 แบบ

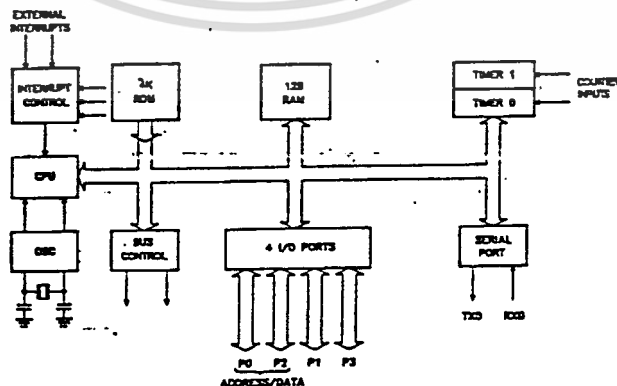
- มีแหล่งกำเนิดสัญญาณขอขัดจังหวะการทำงานของโปรแกรม (Interrupt Request Signal) 6 แหล่งซึ่งสามารถทำการกระโดดไปทำงานตอบสนองการขัดจังหวะ (Interrupt Service Routine) ได้ต่าง ๆ กัน 5 ตำแหน่ง

- สามารถเลือกการทำงานให้อยู่ในโหมด Idle และ Power Down ซึ่งจะประหยัดการใช้กำลังไฟในการทำงาน

ซึ่งจากข้อดีดังกล่าว จึงทำให้ MCS-51 เป็นที่นิยมนำมาใช้ในการควบคุมระบบอัตโนมัติมาก คุณสมบัติดังกล่าวบรรจุไว้ในวงจรรวมเดี่ยว (Single Chip) ขนาด 40 ขา ดังนั้นจึงสามารถออกแบบให้ระบบทั้งหมดมีขนาดเล็ก และการที่ทั้งหมดบรรจุอยู่ในวงจรรวมเดี่ยวจึงทำให้การตรวจสอบหาข้อผิดพลาดในระบบง่ายไม่สลับซับซ้อน รวมทั้งลดปัญหาเรื่องการที่มีสัญญาณรบกวนในระบบจนทำให้การทำงานผิดพลาดไป แต่การที่จะนำเอา MCS-51 มาใช้งานจำเป็นที่จะต้องศึกษาและทำความเข้าใจถึงโครงสร้างและองค์ประกอบของ MCS-51 ให้เป็นไปตามต้องการ ในหนังสือเล่มนี้จะอธิบายการใช้งานของ MCS-51 โดยใช้ 8051 เป็นตัวอย่างในการอธิบาย เพราะไมโครคอนโทรลเลอร์ในตระกูลนี้จะแตกต่างกันเพียงเล็กน้อยเท่านั้น

1.1 โครงสร้างของ 8051

ภายใน 8051 จะประกอบด้วย GATE ต่างๆ เช่น AND, OR, NOT ซึ่ง GATE เหล่านี้จะถูกนำมาออกแบบให้ทำหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง (Instruction Decoder) วงจรสร้างสัญญาณนาฬิกา (Clock Signal Generator) โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยต่างๆ ดังโคอะแกรมในรูปที่ 1.1



รูปที่ 1.1 โคอะแกรมโครงสร้างของ 8051

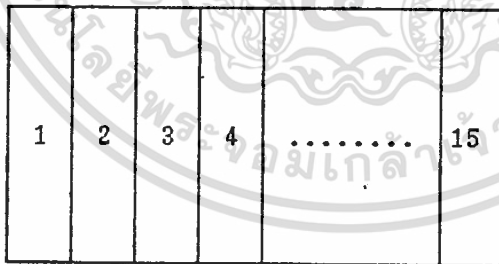
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดอะแกรมในรูปแบบที่ 1.1 เป็นโครงสร้างใหญ่ๆของ 8051 เนื่องจากลักษณะของ 8051 เป็นคอมพิวเตอร์จึงประกอบด้วย 3 ส่วนหลักๆ คือ

ส่วนที่ 1 คือ CPU (Central Processing Unit) หรือตัวประมวลผล ส่วนนี้จะมีส่วนที่ทำหน้าที่สร้างสัญญาณในการควบคุมในการติดต่อกับส่วนอื่นๆ เรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุม ได้แก่สัญญาณสำหรับการติดต่อกับหน่วยความจำ, อุปกรณ์รับข้อมูลเข้า หรือส่งข้อมูลออกจากตัว 8051 ซึ่งส่วนควบคุมการขัดจังหวะ (Interrupt Control) และส่วนควบคุมบัส (Bus Control) ก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณควบคุมจากส่วน CPU นี้จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่ง (Instruction) ตามที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรรอสซิงคลเลเตอร์เพื่อให้ทุกๆ ส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง

ใน CPU นี้ยังประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่าส่วนประมวลผล (Arithmetic Logic Unit) ส่วนนี้จะทำหน้าที่ประมวลผลข้อมูล เช่น การบวก, ลบ, คูณ หรือหารข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในรีจิสเตอร์หรือหน่วยความจำที่ต้องการ

ส่วนที่ 2 คือหน่วยความจำ (Memory) มีไว้สำหรับจัดจำข้อมูล ถ้าจะให้เห็นภาพพจน์ของหน่วยความจำได้ดีก็คือ หน่วยความจำเปรียบเสมือนกล่องใส่เอกสารจำนวนมากที่นำมาต่อเรียงกันไว้ แต่ละกล่องก็มีเอกสารหนึ่งแผ่นดังในรูปแบบที่ 1.3 มีกล่องเอกสารทั้งหมด 15 กล่อง



รูปที่ 1.3 ภาพเสมือนของหน่วยความจำ

ถ้าต้องการเอกสารจากกล่องใด หรือเอาเอกสารไปเก็บที่กล่องใด จะต้องรู้หมายเลขของกล่องข้อมูลเสียก่อน ซึ่งถ้าเป็นหน่วยความจำแล้วหมายเลขของกล่องก็คือตำแหน่งของหน่วยความจำ หรือแอสแอดเรส (Address) นั่นเอง การเอาข้อมูลไปเก็บไว้ที่หน่วยความจำเรียกว่าการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Write) ข้อมูล และการนำเอาข้อมูลออกจากหน่วยความจำเรียกว่าการอ่าน (Read) ข้อมูล ซึ่งแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลได้เพียงค่าเดียวเท่านั้น ในไมโครโปรเซสเซอร์ทั้งหมดรวมทั้ง 8051 นั้นข้อมูลในแต่ละตำแหน่งของหน่วยความจำจะมีค่าได้เพียง 8 หลักของเลขฐานสอง (8 บิตเท่ากับ 1 ไบต์) ดังนั้นแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลมีค่าได้ระหว่าง 0 ถึง 255 (00000000 ถึง 11111111 ในเลขฐาน 2) แต่จำนวนตำแหน่งที่เก็บข้อมูลได้ขึ้นกับไมโครโปรเซสเซอร์แต่ละเบอร์ การติดต่อกับหน่วยความจำจะต้องมีสัญญาณ 3 กลุ่มคือ

1. แอดเดรสหรือค่าตำแหน่งที่ต้องการติดต่อกับหน่วยความจำใน 8051 จะติดต่อกับหน่วยความจำประเภท Program Memory หรือ Data Memory ได้สูงสุดชนิดละ 65536 ตำแหน่ง ดังนั้นการอ้างอิงแต่ละตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น (2^{16} เท่ากับ $64 * 1024 = 65536$)

2. ข้อมูลที่จะอ่านหรือเขียนกับหน่วยความจำที่ตำแหน่งในข้อ 1

3. สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำ เพื่อบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล

สัญญาณเหล่านี้จะถูกรวบรวมภายใน 8051 สร้างมาจากวงจรลอจิกของคำสั่งที่ 8051 อ่านจากหน่วยความจำ Program Memory เข้าไปทำงานนั่นเอง ในรูปที่ 1.2 หน่วยความจำได้แก่ 4K ROM และ 128 Byte RAM ซึ่งขนาดของหน่วยความจำนี้มีขนาดต่างๆกันตามเบอร์ของไมโครคอนโทรลเลอร์ และจะอธิบายโดยละเอียดในข้อ 1.2

ส่วนที่ 3 อุปกรณ์อินพุตและเอาต์พุต (Input/Output Device) เป็นส่วนที่จะต้องใช้ในการส่งข้อมูลเข้าหรือออกจาก 8051 ทำให้ 8051 ติดต่อกับภายนอกได้ ดังในไดอะแกรมดังรูปที่ 1.2 อุปกรณ์อินพุตและเอาต์พุตได้แก่ 4 I/O Port, Timer 0, Timer 1, Serial Port การทำงานของแต่ละส่วนมีดังนี้

1.4 I/O Port คำว่าพอร์ทหมายถึงจุดที่จะติดต่อกับส่วนที่อยู่ภายนอก 4 I/O Port ของ 8051 เป็นที่สำหรับรับ-ส่งข้อมูลซึ่งเป็นสัญญาณดิจิทัลเข้าหรือออกจากตัว MCS-51 พอร์ท มีทั้งหมด 4 พอร์ท โดยแต่ละพอร์ทจะรับ-ส่งข้อมูลได้ 8 บิต มีพอร์ท P0, P1, P2 และ P3 บางพอร์ทจะใช้ทำงานมากกว่า 1 อย่างก็ได้ เช่น พอร์ท P0 และ P2 จะใช้สำหรับการส่งค่าตำแหน่ง (Address) ของหน่วยความจำที่ต้องการติดต่อและพอร์ท P0 จะใช้รับ-ส่งข้อมูลเมื่อติดต่อกับหน่วยความจำได้ด้วย แต่สิ่งเหล่านี้ไม่ได้เกิดขึ้นในเวลาเดียวกัน แต่จะใช้วิธีทำงานตามลำดับโดยควบคุมจากสัญญาณควบคุม (Control) ที่ลอจิกสมาแต่ละคำสั่งที่ให้คอมพิวเตอร์ทำงานนั่นเอง และสัญญาณ

ทั้งหมดจะอ้างอิงกับสัญญาณนาฬิกา

2. Timer 0 และ Timer 1 เป็นวงจรนับที่สามารถกำหนดให้นับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอก 8051 หรือจำนวนไซเคิลของสัญญาณนาฬิกาภายใน 8051 ก็ได้ ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดย CPU

3. Serial Port หรือพอร์ทอนุกรม CPU จะอ่านหรือเขียนข้อมูลกับ Serial Port เป็นแบบ 8 บิต แต่ข้อมูลจะถูกส่งออกจาก 8051 เรียงไปที่ละบิตออกจากขา TXD และในการรับข้อมูลเข้าก็จะรับเข้ามาที่ละบิตทางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิตเพื่อให้ CPU อ่านไปใช้งานต่อไป

8051 มีพอร์ทให้ใช้งานได้หลายแบบทำให้สะดวกต่อการนำไปใช้งานต่างๆมากมาย การจะนำพอร์ทเหล่านี้ไปใช้งานจะต้องเขียนโปรแกรมขึ้นควบคุมที่จะได้กล่าวต่อไป

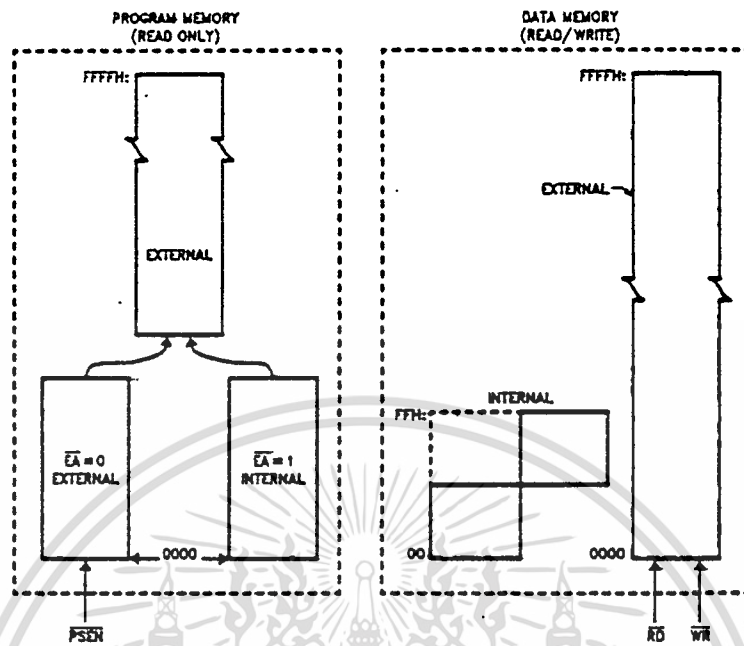
1.2 การจัดการหน่วยความจำของ 8051

หน่วยความจำของ 8051 แบ่งออกไว้เป็น 2 แบบตามลักษณะของการใช้งานคือ

1. Program Memory เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปรหัสภาษาเครื่อง (Machine Language) ซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำประเภทนี้เข้าไปถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่นๆตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้จะต้องเป็นแบบ Read Only Memory (ROM) และผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ (หน่วยความจำแบบ ROM เป็นแบบ Non volatile ซึ่งเมื่อปิดไฟแล้วข้อมูลก็ไม่มีการสูญหาย) การเขียนข้อมูลลงไปใน ROM จะต้องใช้เครื่องมือพิเศษ ในระหว่างการทำงานของ 8051 ผู้ใช้จะไม่สามารถใช้คำสั่งทำการเขียนข้อมูลลงในหน่วยความจำแบบนี้ได้ จำนวนตำแหน่งสูงสุดของหน่วยความจำแบบนี้ที่ 8051 จะใช้งานได้คือ 65536 ตำแหน่ง ค่าของตำแหน่งจะเขียนเป็นเลขฐาน 16 ได้ตั้งแต่ 0000H ถึง FFFFH หน่วยความจำตำแหน่ง 0000H ถึง 0FFFFH จำนวน 4 กิโลไบต์นั้นผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่อยู่ภายในหรือภายนอก 8051 (ไมโครคอนโทรลเลอร์เบอร์อื่นๆเช่น 8052 จะมีขนาดของ ROM ส่วนนี้ได้ถึง 8 กิโลไบต์ ตำแหน่ง 0000H ถึง 1FFFFH) ถ้าต้องการให้ 8051 ทำงานตามคำสั่งที่เก็บไว้ใน ROM ภายใน 8051 ก็ให้ป้อนสัญญาณสถานะลอจิก High (1) เข้าที่ขา EA ของ 8051 แต่ถ้าต้องการให้ทำงานตามโปรแกรมที่เก็บไว้ใน ROM ภายนอก 8051 ก็ให้ต่อลอจิก Low (0) เข้าที่ขา EA ของ 8051 ส่วนหน่วยความจำที่ตำแหน่ง 1FFFFH ถึง FFFFFH จะต้องต่ออยู่ภายนอก 8051 เสมอ ดังแสดงในแผนภูมิหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิพนธ์ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำ (Memory Map) ในรูปที่ 1.3



รูปที่ 1.3 แผนภูมิหน่วยความจำของ 8051

Internal Memory หมายถึงหน่วยความจำนั้นอยู่ภายใน 8051 ส่วน External Memory หมายถึง หน่วยความจำนั้นอยู่ภายนอก 8051

ไมโครคอนโทรลเลอร์เบอร์ 8031, 8051 และ 8751 นั้น โดยโครงสร้างและรหัสคำสั่ง จะเหมือนกันทุกประการจะแตกต่างกันที่

- 8031 จะไม่มี ROM ขนาด 4 กิโลไบต์ที่อยู่ภายใน ผู้ใช้จะต้องเลือกการใช้งาน Program Memory อยู่ภายนอกจบรวมทั้งหมด 64 กิโลไบต์
- 8051 จะมี ROM ขนาด 4 กิโลไบต์ อยู่ภายใน ถ้าต้องการเก็บคำสั่งควบคุมการทำงาน ไว้ในหน่วยความจำส่วนนี้ จะต้องส่งโปรแกรมคำสั่งไปให้โรงงานผู้ผลิตเขียนใส่ ROM ที่ตั้งแต่นั้นขั้นตอนการผลิตจบรวม ผู้ใช้ไม่สามารถแก้ไขวงจรได้เอง ถ้าจะนำมาใช้งานโดยเก็บโปรแกรม ไว้ในหน่วยความจำช่วง 4 กิโลไบต์แรกอยู่ภายนอกก็สามารถทำได้ โดยการต่อ ROM ไว้ภายนอก แล้วต่อขา EA ของ 8051 ไว้กับสัญญาณที่มีสถานะลอจิกเป็น 0
- 8751 จะมีหน่วยความจำขนาด 4 กิโลไบต์ เป็นแบบ EPROM (Erasable Program Read Only Memory) อยู่ภายในจบรวมเอาไว้ ใช้เก็บคำสั่งที่จะให้ 8751 ทำงาน ผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีเพื่อนำไปใช้

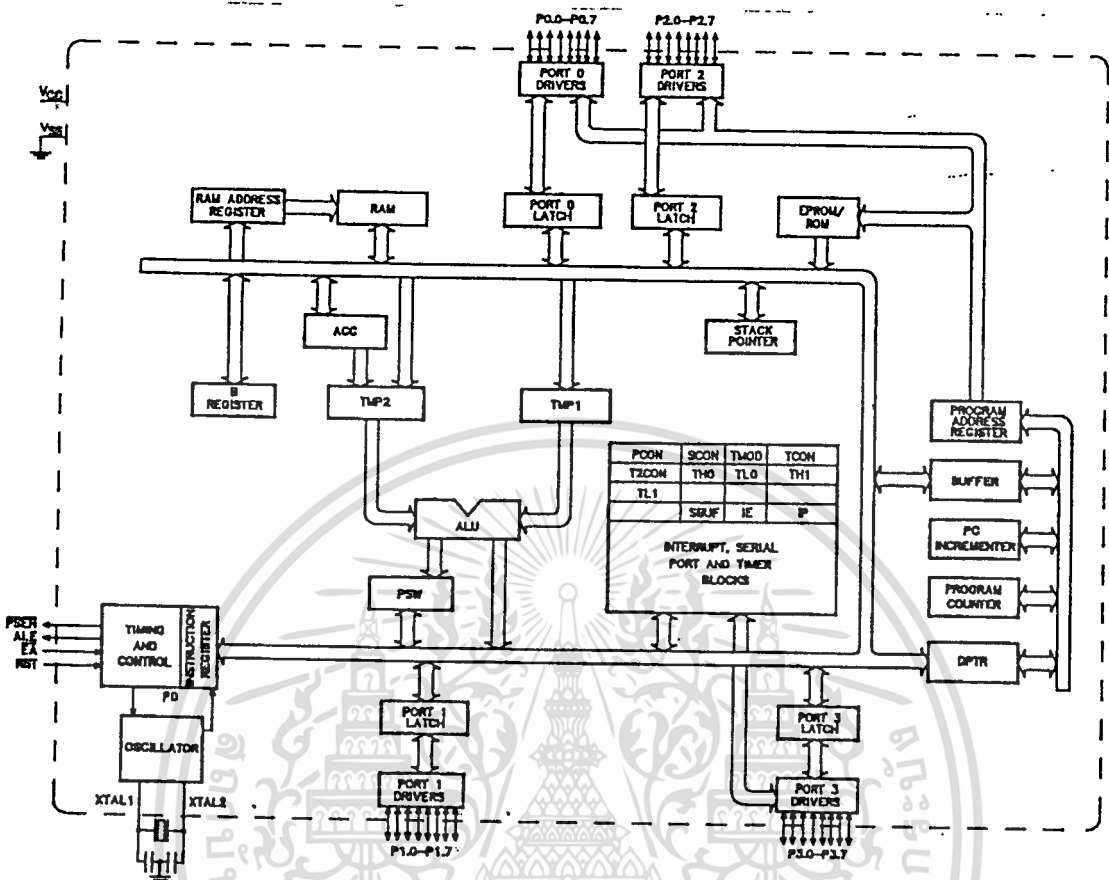


สามารถเขียนโปรแกรมไปใน EPROM ได้โดยการใช้เครื่องมือที่เรียกว่าเครื่องโปรแกรม EPROM (EPROM Programmer) และผู้ใช้สามารถแก้ไขโปรแกรมที่อยู่ใน EPROM ได้โดยการล้างข้อมูลในทุกตำแหน่งของ EPROM ออกด้วยการฉายแสงอุลตราไวโอเล็ต (Ultraviolet) ผ่านกระจกใสบนวงจรรวมเข้าไปยังโปรแกรมภายใน ตามเวลาที่กำหนดไว้ในคู่มือเฉพาะ (Data sheet) ของ 8751 จากนั้นก็ใช้เครื่องโปรแกรม EPROM เขียนโปรแกรมลงไปใหม่ 8751 นี้จะสะดวกมากในการพัฒนาโปรแกรม

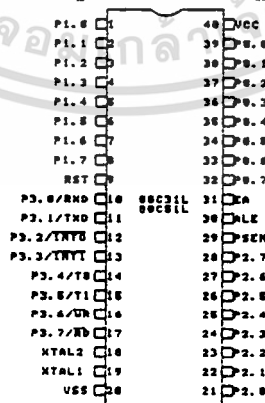
2.Data Memory เป็นหน่วยความจำที่ 8051 จะใช้สำหรับพัก, เก็บข้อมูล แล้วเรียกมาใช้ใหม่ในระหว่างการทำงานของ 8051 การอ่านหรือเขียนข้อมูลจากหน่วยความจำจะกระทำโดยคำสั่งที่เก็บไว้ใน Program Memory หน่วยความจำแบบนี้เป็นประเภท Random Access Memory (RAM) ถ้ามีไฟเลี้ยงอยู่ข้อมูลที่เก็บไว้จะไม่สูญหาย แต่ถ้าปิดเครื่องหรือไม่จ่ายไฟให้แก่ RAM แล้ว ข้อมูลใน RAM ก็จะไม่สูญหาย การสูญหายของข้อมูลไม่ได้หมายความว่า ไม่มีอะไรอยู่เลย แต่เป็นการที่ข้อมูลใหม่ซึ่งไม่ใช่ข้อมูลที่เก็บไว้เดิมเข้ามาอยู่แทนที่ เช่นเดิมเก็บข้อมูล 18H ไว้ที่ตำแหน่ง 1900H เมื่อปิดไฟแล้วเปิดใหม่ ข้อมูลที่ตำแหน่ง 1900H จะไม่ใช่ 18H อาจจะเป็นค่าอะไรก็ได้ ซึ่งเรียกการเกิดลักษณะแบบนี้ว่าข้อมูลสูญหายไป หน่วยความจำแบบ Data Memory ของ 8051 จะมีอยู่ 2 ชุด ชุดหนึ่งอยู่ภายใน 8051 จำนวน 128 ไบท์ที่ตำแหน่ง 00H ถึง 7FH (เบอร์ 8052 จะมี 256 ไบท์อยู่ที่ตำแหน่ง 00H ถึง FFH) และอีกชุดหนึ่งจะตั้งอยู่นอกของวงจรรวม 8051 มีได้สูงสุด 65536 ไบท์ (64 กิโลไบท์) อยู่ที่ตำแหน่ง 0000H ถึง FFFFH ดังแสดงในรูปที่ 1.3 หน่วยความจำแบบ Data Memory ภายใน 8051 ที่ตำแหน่ง 80H ถึง FFH นั้นไม่ได้มีอยู่ทุกตำแหน่ง จะมีเฉพาะในบางตำแหน่งซึ่งเรียกหน่วยความจำบางตำแหน่งนี้ว่า Special Function Register (SFR) เพราะจะใช้หน่วยความจำแบบนี้สำหรับงานพิเศษเท่านั้น แต่ละตำแหน่งของหน่วยความจำแบบ SFR นี้ อาจเป็น RAM หรือวงจรรวม (Counter) วงจรตั้งเวลา (Timer) ก็ได้ เช่นเป็น Timer 0, Timer 1 ดังนั้นใน 8051 จึงไม่ถือว่า SFR เป็น Data Memory ถ้าเป็น 8052 ซึ่งมี Data Memory ขนาด 256 ไบท์ จะใช้บางตำแหน่งของหน่วยความจำ 80H ถึง FFH เป็น SFR ส่วนตำแหน่งที่เหลือก็เป็น RAM เหมือนกับหน่วยความจำช่วง 00H ถึง 7FH นั้นเอง

1.3 สถาปัตยกรรมของ 8051

ในรูปที่ 1.4 เป็นสถาปัตยกรรมภายในของ 8051 ซึ่งจะอธิบายถึงส่วนย่อยๆของภายใน 8051 เพียงขั้นเดียว และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขา (Pin) ของ 8051 ที่มีอยู่ 40 ขา ดังรูปที่ 1.5



รูปที่ 1.4 สถาปัตยกรรมภายในของ 8051



รูปที่ 1.5 โดอะแกรมขาของ 8051 แบบ DIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8051 ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 8051 มีอยู่ข้างละ 20 ขารวมทั้งหมด 40 ขา นั้นจะใช้งานต่าง ๆ กันดังนี้

Vcc

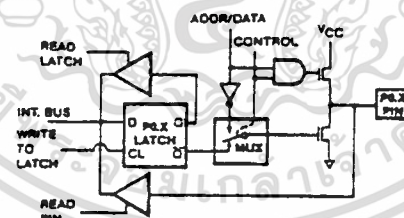
ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลต์เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับโวลต์เตจของลอจิก 0 และ 1 ของ 8051 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง

Vss

ขา 20 เป็นขาที่ต้องต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์เข้าด้วยกัน

Port 0

เป็นพอร์ตขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับดังในรูปที่ 1.5 แต่ละขาอาจจะเขียนว่า P0.0, P0.1, ..., P0.7 นั้น P0.7 หมายถึงบิต 7 ของพอร์ต 0 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด (Most Significant) และ P0.0 คือบิต 0 ของพอร์ต 0 เป็นบิตที่มีนัยสำคัญต่ำสุด (Least Significant) พอร์ต 0 นี้ใช้ได้ทั้งการรับ-ส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ตรับ-ส่งข้อมูลก็ได้ ข้อมูลที่ถูกส่งออกจากพอร์ต 0 จะถูก Latch ไว้ที่ขาของพอร์ต โครงสร้างของแต่ละบิตของพอร์ต 0 เป็นแบบ Open Drain Bidirectional ดังรูปที่ 1.6



A. Port 0 Bit

รูปที่ 1.6 โครงสร้างของ PORT 0

ในรูปที่ 1.6 เมื่อเปรียบเทียบกับรูปที่ 1.4 ส่วนที่ 1 ของรูป 1.6 ก็คือ Port 0 Latch ในรูปที่ 1.4 และส่วนที่ 2 ในรูปที่ 1.6 ก็คือ Port 0 Driver ของรูปที่ 1.4 นั้นเอง

จากโครงสร้างในรูปที่ 1.6 เมื่อมีคำสั่งการเขียนข้อมูลมาส่งพอร์ต 0 ข้อมูลจาก Internal Data Bus จะถูก Latch ไว้ที่ D-FF โดยสัญญาณ "Write to Latch" ที่ถูกสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาจากส่วน Timing and Control และในการอ่านข้อมูลจากพอร์ท 0 จะอ่านได้ 2 แบบคือ การอ่านข้อมูลที่ส่งไปเก็บไว้ที่พอร์ทก็จะมีสัญญาณ Read Latch มาเพื่ออ่านข้อมูลจาก D-FF กลับเข้าไปยัง Internal Data Bus การอ่านข้อมูลอีกแบบก็คือการอ่านสถานะของสัญญาณที่เข้ามาทางพอร์ท 0 ก็จะมีสัญญาณ Read Pin มาควบคุมการอ่าน พอร์ท 0 จะใช้งานหลายอย่างดังนี้

1. ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อด้วย ตำแหน่งหน่วยความจำสูงสุดที่จะติดต่อได้ก็คือ 64 kbyte จึงมีค่าตำแหน่งหน่วยความจำ 16 บิตของเลขฐาน 2 ค่าตำแหน่งหน่วยความจำ 8 บิตล่างจะถูกส่งออกไปทางพอร์ท 0 และ 8 บิตบนจะส่งออกไปทางพอร์ท 2

2. ใช้รับ-ส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory

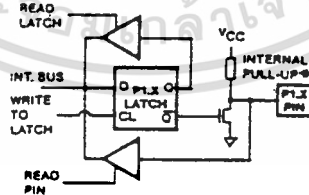
3. ใช้รับ-ส่งข้อมูลผ่านทางพอร์ทโดยตรง ในกรณีที่ไม่มีการใช้หน่วยความจำของ Program Memory หรือ Data Memory ภายนอก

วงจรภายในส่วน Timing and Control จะเป็นตัวสร้างสัญญาณมาควบคุมวงจรในรูปที่ 1.6 เพื่อให้การทำงานในแต่ละอย่างข้างต้น เมื่อแต่ละบิตของพอร์ท 0 ทำงานตามข้อ 1 และ 2 ข้างต้น วงจร Timing and Control จะทำให้สถานะลอจิกของขา Control เป็น 1 ซึ่งทำให้สวิตช์ MUX อยู่ในตำแหน่งข้างบน เมื่อพอร์ท 0 จะส่งข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำหรือข้อมูลที่เขียนออกไปยังหน่วยความจำภายนอกก็จะส่งค่าดังกล่าวมายัง ADDR/DATA ถ้าข้อมูลที่ส่งมาเป็น 1 จะทำให้สัญญาณออกจาก AND GATE เป็น 1 และสัญญาณที่ออกจาก Inverter เป็น 0 ดังนั้น FET ตัวบน ON (สถานะ ON ของ FET คือความต้านทานระหว่างขา D กับ S มีค่าต่ำมาก เสมือนเป็นวงจรปิด) ส่วน FET ตัวล่าง OFF (สถานะ OFF ของ FET คือ ความต้านทานระหว่างขา D กับ S สูงมากเสมือนกับเป็นวงจรเปิด) สถานะลอจิกที่ขา PO.X PIN จะเป็น 1 แต่ถ้าข้อมูลที่ถูกส่งออกมาถึง ADDR/DATA เป็น 0 ก็จะทำให้สัญญาณจาก AND GATE เป็น 0 และสัญญาณที่ออกจาก Inverter เป็น 1 ดังนั้น FET ตัวบนจะ OFF ส่วน FET ตัวล่างจะ ON ทำให้สถานะลอจิกที่ขา PO.X PIN เป็น 0 เมื่อ 8051 ต้องการใช้พอร์ท 0 สำหรับการอ่านข้อมูลจากหน่วยความจำภายนอก หรือใช้ทำงานตามข้อ 3 ข้างบน ก็จะทำให้โดยวงจร Timing and Control ทำให้สถานะลอจิกของสัญญาณ Control ในรูปเป็น 0 ทำให้เข้ากับทุกจาก AND GATE เป็น 0 FET ตัวบนจะ OFF และสวิตช์จะอยู่ในตำแหน่งข้างล่าง ดังนั้น FET ตัวล่างจะ ON หรือ OFF ก็แล้วแต่ข้อมูลที่ขา Q ของ D-FF เมื่อมีการเขียนข้อมูลจาก Internal Data Bus มาถึง D-FF ก็จะมีสัญญาณ Write to Latch มาถึง D-FF ด้วย ถ้าข้อมูลที่เขียนมาเป็น 1 ก็จะทำให้ขา Q มีสถานะลอจิกเป็น 0 ทำให้ FET ตัวล่าง OFF ดังนั้นขา PO.X จะอยู่ในสถานะอิมพี

แดนซ์สูง (High Impedance) เพราะ FET ทั้ง 2 ตัว OFF แต่ถ้าข้อมูลที่เขียนมายัง D-FF เป็น 0 ก็จะทำให้ FET ตัวล่าง ON แต่ตัวบน OFF ทำให้สภาวะลอจิกที่ขา P0.X เป็น 1 ดังนั้น พอร์ต 0 เมื่อให้ทำงานเป็นพอร์ตส่งข้อมูล (ไม่ใช่ส่งตำแหน่งหน่วยความจำ) จะไม่สามารถแสดง สภาวะลอจิก 1 ได้จึงต้องต่อตัวต้านทาน Pull Up อยู่ภายนอก ระหว่างขา P0.X กับไฟเลี้ยง วงจร ถ้าจะใช้พอร์ต 0 สำหรับรับข้อมูลเข้าจะต้องเขียน 1 มาเก็บไว้ยัง D-FF เสียก่อนเพื่อ ให้ขา P0.X อยู่ในสภาวะ High Impedance แล้วจึงใช้คำสั่งอ่านสภาวะลอจิกเข้าไปยัง Internal Data Bus ต่อไป โดยคำสั่งอ่านสภาวะลอจิกทางพอร์ต 0 ก็จะทำให้วงจร Timing and Control สร้างสัญญาณ Read Pin สำหรับการอ่านสภาวะลอจิกข้างต้น ถ้าไม่เขียน 1 มา เก็บไว้ยัง D-FF ก่อนที่จะอ่านข้อมูลแล้วอาจมีข้อมูลค้างอยู่ที่ D-FF ทำให้ Q เป็น 0 และ Q เป็น 1 ซึ่งทำให้ FET ตัวล่าง ON สัญญาณที่ต่อเข้ามาที่ขา P0.X ไม่ว่าจะมีส่วนลอจิกใดจะถูกดึงลง กราวด์ ดังนั้นเมื่ออ่านข้อมูลเข้าไปก็จะพบว่าเป็น 0 เสมอ ในการอ่านข้อมูลจากหน่วยความจำภายนอกนั้น วงจร Timing and Control ก็จะเขียนข้อมูลมายัง D-FF ให้เป็น 1 และสร้างสัญญาณ Control ให้มีลอจิกเป็น 0 ก่อนจะอ่านข้อมูลเข้าไปด้วย

Port 1

เป็นพอร์ตขนานขนาด 8 บิต ในรูปที่ 1.5 คือขา P1.0 ถึง P1.7 (ขา 1-8) P1.0 หมายถึงบิต 0 ของพอร์ต 1 ซึ่งเป็นบิต Least Significant Bit และบิต P1.7 หมายถึงบิตที่ 7 ของพอร์ต 1 ซึ่งเป็นบิต Most Significant Bit โครงสร้างของพอร์ต 1 แต่ละบิตมีดัง รูปที่ 1.7



B. Port 1 Bit

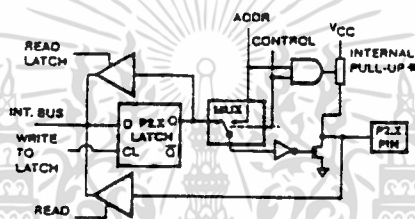
รูปที่ 1.7 โครงสร้างของ PORT 1

ส่วนที่ 1 คือ Port Latch ในรูปที่ 1.4 ซึ่งจะมีการทำงานเหมือนส่วนที่ 1 ของพอร์ต 0 ในรูปที่ 1.6 ส่วนที่ 2 คือ Port Driver ในรูปที่ 1.4 Port 1 Driver นี้จะมีตัวต้านทานต่อ

อยู่เป็น Internal Pull Up พอร์ต 1 นี้จะใช้ทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ถูกส่งออกมาทางพอร์ต 1 จะถูก Latch ไว้แล้วส่งออกไปทางแต่ละขา ก่อนที่จะอ่านข้อมูลเข้าไปทางพอร์ต 1 จะต้องเขียน 1 ไปยังทุกบิตของพอร์ต 1 เสียก่อนเพื่อให้ FET อยู่ในสภาวะ OFF ก่อน มิฉะนั้นแล้วถ้ามีข้อมูล 0 ส่งออกมาค้างอยู่ที่ D-FF จะทำให้ FET อยู่ในสภาวะ ON ดังนั้นถ้าสัญญาณภายนอกส่งเข้ามาที่ขานี้ก็จะถูกลัดวงจรลงกราวด์ โดยไม่สนใจว่าสภาวะลอจิกของสัญญาณที่เข้ามาจะเป็นอะไร ข้อมูลที่อ่านเข้าไปจึงจะเป็น 0 เสมอ

Port 2

พอร์ตขนานขนาด 8 บิตคือขา P2.0 ถึง P2.7 (บิต 0 ถึงบิต 7 ของพอร์ต 2) ในรูปที่ 1.5 โครงสร้างของพอร์ต 2 แต่ละบิตจะมีดังรูปที่ 1.8



C. Port 2 Bit

รูปที่ 1.8 โครงสร้างของ PORT 2

ลักษณะโครงสร้างจะเหมือนกับ Port 0 แตกต่างกันใน Port 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะคือ

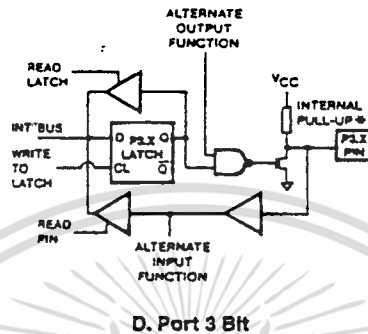
1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิตบนของค่าตำแหน่ง
2. ใช้เป็นพอร์ตรับและส่งข้อมูลกับภายนอก

ดังนั้นภาค Driver ของพอร์ต 2 จึงแตกต่างจาก Driver ของพอร์ต 0 โดยที่ในพอร์ต 2 นั้นจะมีเฉพาะ ADDR (ตำแหน่งหน่วยความจำ) เข้ามาที่ MUX (Multiplexer) เท่านั้น นอกนั้นแล้วการทำงานจะเหมือนกันและที่เข้าทุกของพอร์ต 2 จะมี Internal pull-up ซึ่งเป็นตัวต้านทานและจะทำให้เข้าทุกของพอร์ต 2 แสดงสถานะเป็น 1 ได้ ถ้า FET อยู่ในสภาวะ OFF บางครั้งเรียกว่า "Quasi-bidirectional" เมื่อใช้เป็นพอร์ตอินพุตก็สามารถทำได้โดยการต่อสัญญาณภายนอกเข้ามาโดยตรง ถ้าสัญญาณภายนอกเป็น 0 ก็จะมีกระแสไหลออกจากพอร์ต (Source

Current) ในการที่จะใช้พอร์ทนี้เป็นพอร์ทรับข้อมูลเข้า จะต้องเขียน 1 ไปยังแต่ละบิตของพอร์ทเสียก่อน ดังได้อธิบายในเรื่อง Port 0 และ Port 1

Port 3

คือขา P3.0 ถึง P3.7 หรือขา 10-17 ตามลำดับในรูปที่ 1.5 พอร์ทนี้มรโครงสร้างดังรูปที่ 1.9



รูปที่ 1.9 โครงสร้างของ PORT 3

ส่วนที่ 1 ในรูปที่ 1.9 เป็นส่วน Latch ข้อมูลที่เขียนมาซึ่งพอร์ท 3 ทาง Internal Bus เหมือนกับพอร์ทอื่นๆ และพอร์ท 3 จะมี Internal pull up อยู่ทุกบิต แต่พอร์ท 3 นี้แต่ละบิตจะใช้ในการทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ Alternative Output Function ที่สร้างมาจากส่วน Timing and Control สัญญาณ Alternative Output Function เป็นสัญญาณส่งออกในกรณีที่ใช้พอร์ท 3 ทำงานในฟังก์ชันอื่น และจุด Alternative Input Function เป็นจุดที่จะเอาสัญญาณไปเข้ากับส่วนอื่นตามการทำงานของบิตนั้น แต่ละบิตของพอร์ท 3 จะมีฟังก์ชันอื่นดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/INT0 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/INT1 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

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

P3.5/T1 (Timer/Counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 1 ซึ่งมีการทำงานเหมือนกับ T0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

RST

ขารีเซ็ตขานี้จะใช้ทำการรีเซ็ตการทำงานของ 8051 ที่ขา RST ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ (Ground) ถ้าป้อนสัญญาณที่มีสถานะลอจิก 1 เข้าไปที่ขานี้จะเป็นการรีเซ็ตการทำงานของ 8051 ดังนั้นจึงสามารถต่อตัวเก็บประจุ (Capacitor) ภายนอกระหว่างขา RST กับไฟเลี้ยง +5 โวลต์ เพื่อให้เกิดการรีเซ็ตเมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งเรียกว่า Power on reset การรีเซ็ตจะทำให้ค่าในรีจิสเตอร์ต่างๆเปลี่ยนไปเป็นค่าหนึ่งดังในตารางรูปที่ 1.10

| REGISTER | CONTENT |
|----------|-----------|
| PC | 0000H |
| ACC | 00H |
| B | 00H |
| PSW | 00H |
| SP | 00H |
| DPTR | 0000H |
| PO-P3 | 0FFH |
| IP | 00H |
| IE | 0X000000H |
| TMOD | 00H |
| TCON | 00H |
| T2CON | 00H |
| TH0 | 00H |

รูปที่ 1.10 ค่าของรีจิสเตอร์เมื่อเกิดการรีเซ็ต 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| REGISTER | CONTENT |
|----------|---------------|
| TL0 | 00H |
| TH1 | 00H |
| TL1 | 00H |
| TH2 | 00H |
| TL2 | 00H |
| RCAP2H | 00H |
| RCAP2L | 00H |
| SCON | 00H |
| SBUF | Indeterminate |
| IOCON | 00H |

รูปที่ 1.10 (ต่อ) ค่าของรีจิสเตอร์เมื่อเกิดการรีเซ็ต 8051

ในตารางรูปที่ 1.10 ช่องทางขวาเป็นค่าของรีจิสเตอร์ที่อยู่ทางซ้ายเมื่อสิ้นสุดการรีเซ็ต ในรีจิสเตอร์ SBUF เมื่อสิ้นสุดการรีเซ็ตจะมีค่าไม่แน่นอน และพอร์ตจะอยู่ในสภาวะลอจิก 1 ทุกบิต ตลอดเวลาที่สัญญาณของขา RST เป็น High อยู่

เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจากการรีเซ็ต 8051 จะเริ่มทำงานตามคำสั่งที่อยู่ใน Program Memory ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่ทำงานถูกเปลี่ยนให้เป็น 0000H ในเครื่องไมโครคอมพิวเตอร์แบบบอร์ดเดี่ยว (Single Board Microcomputer) จะมีโปรแกรมที่เขียนเก็บไว้เริ่มจากตำแหน่ง 0000H นี้เรียกว่ามอนิเตอร์โปรแกรม (Monitor Program) ที่จะคอยรับการกดแป้นพิมพ์ (Keyboard) และแสดงผลทางตัวแสดงผล (Display) แบบ 7 Segment

ALE

Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจาก ออกสวิตช์เลเตอร์ สัญญาณนี้จะส่งออกมาตลอดเวลาสัปดาห์บางครั้งที่ติดต่อกับหน่วยความจำ สำหรับข้อมูลภายนอก 8051 สัญญาณนี้จะใช้บอกอุปกรณ์ภายนอก 8051 ว่าขณะนี้สัญญาณนี้ Active

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิตล่างของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อออกไปทางพอร์ท 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุท

PSEN

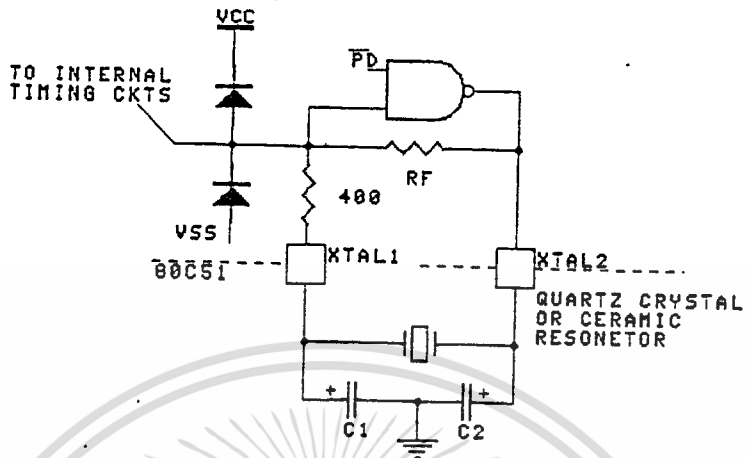
Program Store Enable เป็นขาที่ 29 ในรูปที่ 1.5 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำสำหรับโปรแกรมภายใน 8051 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ถ้า PSEN นี้สามารถต่อไปยังขาอินพุทของ TTL ชนิด LS ได้ถึง 8 อินพุท

EA

External Access ขาที่ 31 ของรูปที่ 1.5 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing and Control ในรูปที่ 1.4 เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก 8051 จะต้องสร้างสัญญาณ PSEN ออกไปยังภายนอกเพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา EA เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ใน 8051 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งจาก ROM ภายใน 8051

XTAL 1

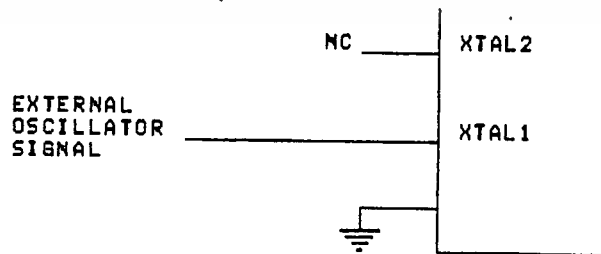
ขาที่ 19 ของรูปที่ 1.5 ขานี้จะต่อเข้ากับขาของ Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟสสัญญาณ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ในรูปที่ 1.11 จะเห็นวงจรภายในของออสซิลเลเตอร์ NAND GATE จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิลเลตหรือไม่ก็ขึ้นกับสัญญาณ PD ซึ่งต่อมาจากบิต PD ของรีจิสเตอร์ PCON ถ้าต้องการให้สัญญาณนาฬิกา (Clock Signal) จากภายนอกมาเป็นสัญญาณนาฬิกาควบคุมการทำงานของ 8051 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้ แต่ถ้าต้องการใช้วงจรออสซิลเลเตอร์ภายในก็ให้ต่อ Crystal หรือเซรามิคเรโซเนเตอร์ดังรูปที่ 1.11 คาปาซิเตอร์ในวงจรควรมีค่าประมาณ 20 PF



รูปที่ 1.11 วงจรออสซิลเลเตอร์ภายใน 8051

XTAL 2

ขาที่ 18 ของรูปที่ 1.5 ขานี้เป็นจุดเข้าที่ทุกของวงจรรขยายแบบกลับเฟสสัญญาณที่ประกอบเป็นวงจรรออสซิลเลเตอร์ (อินพุทคือขา XTAL 1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกมาเป็นสัญญาณนาฬิกาของ 8051 แล้ว ให้ปล่อยขานลอสไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูปที่ 1.12



รูปที่ 1.12 8051 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 การทำงานของ 8051

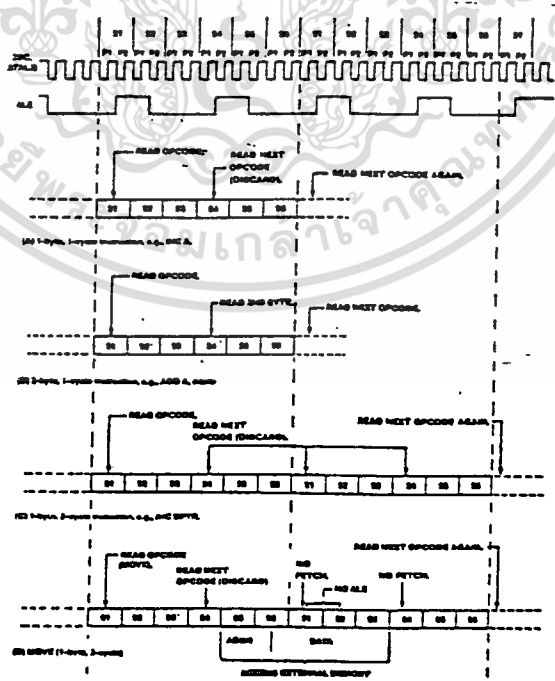
คอมพิวเตอร์จะทำงานด้วยวงจรที่เรียกว่าฮาร์ดแวร์ (Hardware) ประกอบขึ้นมาเพียงอย่างเดียวยังไม่ได้ จะต้องมีการโปรแกรมหรือคำสั่งที่จัดเรียงกันไว้ให้คอมพิวเตอร์ทำงานตามลำดับ ใน 8051 ก็เช่นกัน ผู้ใช้จะต้องเขียนโปรแกรมเป็นภาษาเครื่อง ซึ่งอยู่ในรูปของเลขฐาน 2 เก็บไว้ในหน่วยความจำประเภท Program Memory แต่ละคำสั่งของ 8051 อาจจะถูกเก็บด้วย 1, 2 หรือ 3 ไบต์แล้วแต่ว่าจะเป็นคำสั่งให้ทำงานอะไร คอมพิวเตอร์ก็เหมือนกับคนที่จะต้องทำงานตามคำสั่ง เมื่อรับคำสั่งแล้วก็จะไปทำตามคำสั่งนั้นเสร็จสิ้นแล้วก็มารับคำสั่งต่อไป

จากรูปที่ 1.4 เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งมีวงจร Power on reset ต่ออยู่จะมีการรีเซ็ตเกิดขึ้น การทำงานภายใน 8051 จะเริ่มจากบล็อก Program Counter ซึ่งเป็นวงจรนับ (Counter Circuit) ชนิดหนึ่ง การส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปยังบัส (Bus) หมายเลข 1 บัสนี้มีขนาด 16 บิต ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำ จะปรากฏที่บัส 16 บิตหมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซ็ต ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าจะเป็น ROM ภายในหรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา EA ซึ่งต่ออยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัส (Decoder) แล้วสร้างสัญญาณควบคุมต่อไป ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA จะเป็นการเลือกใช้ ROM ใน 8051 โดยที่วงจร Timing and Control จะสร้างสัญญาณไปยัง ROM ภายในให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ถูกส่งมาทางบัสหมายเลข 2 ข้อมูลจาก ROM จะถูกส่งลงไปยังบัสหมายเลข 3 ที่เรียกว่า Internal Data Bus แล้วนำไปเก็บไว้ที่ Instruction Register (เป็นวงจร Latch) เพื่อส่งต่อไปให้กับวงจร Timing and Control ทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่นๆ ต่อไปแล้วแต่จะเป็นคำสั่งให้ทำงานอะไร ในกรณีที่เลือก ROM จากภายนอก 8051 โดยป้อนสัญญาณลอจิก 1 เข้าไปที่ขา EA จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ต 0 และพอร์ต 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปยังหน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ต 0 ไปยัง Internal Data Bus แล้วไปเก็บที่ Instruction Register เพื่อทำงานต่อไปเหมือนตอนอ่านคำสั่งจาก ROM ภายใน การทำงานในช่วงส่งค่าตำแหน่งหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ใน Instruction Register เรียกว่าเป็นช่วงของการ Fetch (Fetch Cycle)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle เช่นถ้าเป็นคำสั่งให้บวกข้อมูลในรีจิสเตอร์ Accumulator กับข้อมูลจากหน่วยความจำ Data Memory ภายใน RAM คำแหน่ง 23H วงจร Timing and Control ก็จะส่งสัญญาณให้ Instruction Register ส่งค่าตำแหน่งหน่วยความจำ 23H ลงไปยัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ RAM ADDR Register เพื่อใช้ชี้ตำแหน่งหน่วยความจำ RAM จากนั้น Timing and Control จะสั่งให้ RAM ส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ TMP1 (วงจรถ่าย) ขณะเดียวกันวงจร Timing and Control ก็ จะส่งสัญญาณไปยัง TMP2 (วงจรถ่าย) วงจร ALU ซึ่งโครงสร้างเป็นวงจรถ่ายการคำนวณทางคณิตศาสตร์ (บวก,ลบ,คูณ,หาร) และยังสามารถทำงานทางลอจิก (AND,OR,NOT,XOR) จะทำการบวกเลขจาก TMP1 และ TMP2 เข้าด้วยกันผลลัพธ์ที่ได้จะส่งผ่าน Internal Data Bus กลับไปเก็บยัง ACC PSW (Program status word) ซึ่งจะทำหน้าที่เก็บสถานะผลลัพธ์ของการทำงานใน ALU เช่นผลลัพธ์การบวกมีค่าเกิน 8 บิต ก็จะทำให้บิตหนึ่งใน PSW ถูกเซตเป็น 1

การทำงานที่กล่าวมาข้างต้นจะขึ้นกับสัญญาณควบคุมที่สร้างมาจากวงจร Timing and Control และสัญญาณที่สร้างขึ้นนี้จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างมาจากวงจร Oscillator ทำให้การทำงานต่างๆเป็นไปตามลำดับที่ผู้ผลิตได้ออกแบบไว้ ดังในรูปที่ 1.13



รูปที่ 1.13 ลำดับสถานะการทำงานใน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งแต่ละคำสั่งของ 8051 จะใช้เวลาทำงาน 1, 2 หรือ 3 ไชเคิลของเครื่อง (Machine Cycle) แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 ไชเคิลของเครื่องจะใช้เวลา 12 ไชเคิลของสัญญาณนาฬิกา ดังนั้นแต่ละคำสั่งของ 8051 จะใช้เวลาทำงาน 12, 24 หรือ 36 ไชเคิลของสัญญาณนาฬิกานั้นเอง แต่ละไชเคิลของเครื่องจะถูกแบ่งออกเป็น 6 state คือ S1, S2, S3, S4, S5 และ S6 แต่ละ State จะประกอบด้วย 2 ไชเคิลของสัญญาณนาฬิกา ในไชเคิลแรกจะเรียกว่า เฟส 1 (P1) และไชเคิลที่ 2 เรียกว่า เฟส 2 (P2) ในแต่ละเฟสจะนับตั้งแต่ขอบขาของสัญญาณนาฬิกาถึงขอบขาของสัญญาณนาฬิกาที่อยู่ถัดไปดังในรูปที่ 1.13 เมื่อ 8051 ทำงานเสร็จ 1 ไชเคิลของเครื่องก็จะเริ่มทำงาน State 1 Phase 1 (S1P1) ของไชเคิลต่อไป ใน 1 ไชเคิลของเครื่องวงจร Timing and Control จะสร้างสัญญาณ ALE ออกมา 2 ไชเคิลเพื่อ Fetch คำสั่งเข้าไป 2 ครั้งเสมอ ที่บริเวณขอบขาขึ้นของสัญญาณ ALE คำสั่งใดจะมีกี่บิตหรือใช้เวลาทำงานที่ไชเคิลจะดูได้จากตารางชุดคำสั่ง 8051

คำสั่งประเภท 1 บิต 1 ไชเคิลของเครื่องได้แก่คำสั่ง INC A จะมีการอ่านคำสั่งจากหน่วยความจำสำหรับโปรแกรม 2 ครั้ง ที่เวลาประมาณขอบขาขึ้นของสัญญาณ ALE เมื่อคำสั่งแรกถูกอ่านเข้าไปที่เวลาขอบขาขึ้นของสัญญาณ ALE แรก แล้วนำไปเก็บที่ Instruction Register เพื่อให้วงจร Timing and Control ถอดรหัส แล้วเข้าสู่การ Execute ขณะเดียวกันก็จะเริ่มต้นการ Fetch คำสั่งที่อยู่ในหน่วยความจำตำแหน่งถัดไปเข้ามาและคำสั่งที่ 2 จะถูกอ่านเข้ามาที่เวลาขอบขาขึ้นของสัญญาณ ALE ถัดไป วงจร Timing and Control เมื่อถอดรหัสคำสั่งแรกก็จะทราบว่าการทำงานคำสั่งนี้ให้สิ้นสุดจะใช้คำสั่งเพียง 1 บิต ดังนั้นคำสั่งที่ถูกอ่านมาบิตที่ 2 จะไม่ถูกนำมาทำงาน เพียงแต่อ่านเข้ามาแล้วทิ้งไป (Discard) ดังในรูปที่ 1.13a

คำสั่งประเภท 2 บิตและใช้เวลา 1 ไชเคิลของเครื่องได้แก่คำสั่ง ADD A, #data ในหนึ่งไชเคิลของเครื่องนี้จะมีการอ่านเข้ามา 2 บิต เหมือนกับคำสั่งประเภท 1 บิตไชเคิลของเครื่องแตกต่างกันที่บิตที่ 2 จะถูกนำมาใช้งานด้วยไม่ได้ถูกทิ้งไปดังในรูปที่ 1.13b ตัวอย่างของคำสั่ง ADD A, #33H จะเขียนเป็นภาษาเครื่องได้ 2 บิต คือ 24 33 เมื่ออ่านคำสั่งบิตแรก คือ 24 เข้าไปไว้ที่ Instruction Register แล้ว Timing and Control จะถอดรหัสพบว่าเป็นคำสั่งบวกเลข ก็จะส่งสัญญาณไปยัง Accumulator ให้เอาข้อมูลไปไว้ที่ TMP1 เมื่อคำสั่งที่ 2 ถูกอ่านเข้ามาที่ Instruction Register แล้ว Timing and Control จะสั่งให้เอาข้อมูลบิตที่ 2 ส่งลงไปยัง Internal Data Bus ไปเก็บที่ TMP2 จากนั้นวงจร ALU จะนำเอาข้อมูล TMP1 และ TMP2 มาบวกกัน ผลลัพธ์ที่ได้จะส่งออกจาก ALU ไปยัง Internal Data Bus แล้ว

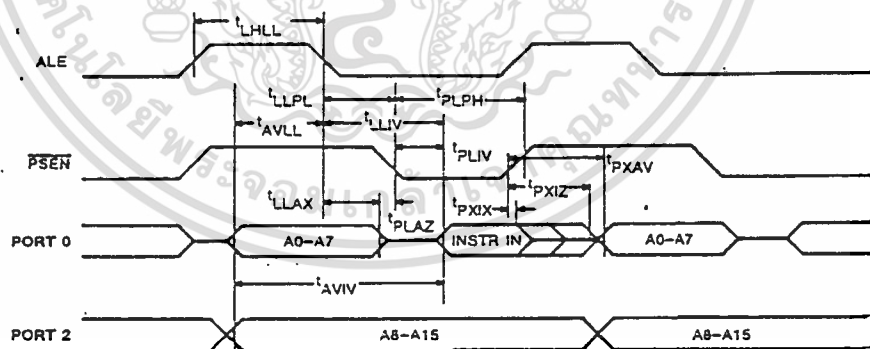
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปเก็บไว้ที่ Accumulator

คำสั่งประเภท 1,2 หรือ 3 ไบท์ที่ใช้เวลาทำงาน 2 ไชเคลของเครื่องเช่นคำสั่ง INC DPTR จะมีการอ่านคำสั่งเข้าไป 4 ครั้งทุกๆรอบขาขึ้นของสัญญาณ ALE ที่มี 2 ครั้งต่อ 1 ไชเคลของเครื่อง ถ้าเป็นคำสั่งของเครื่องประเภท 1,2 หรือ 3 ไบท์ วงจร Timing and Control จะเอาคำสั่ง 1,2 หรือ 3 ไบท์แรกเท่านั้นไปทำงาน ส่วนคำสั่งที่เหลือทิ้งไปดังในรูปที่ 1.14c คำสั่ง 1 ไบท์ที่ใช้เวลาทำงาน 2 ไชเคลของเครื่องที่กล่าวมาแล้วจะไม่รวมคำสั่ง MOVX ซึ่งใช้ในการอ่านหรือเขียนข้อมูลกับหน่วยความจำ Data Memory ภายนอก การทำงานของคำสั่งนี้ จะมีการ Fetch คำสั่งเข้าไป 2 ไบท์ในไชเคลของเครื่องแรก ในไชเคลของเครื่องที่ 2 จะไม่มีการ Fetch คำสั่งเข้าไป แต่จะเป็นช่วงเวลาอ่านหรือเขียนข้อมูลกับ Data Memory ภายนอก สัญญาณ ALE ซึ่งปกติจะเปลี่ยนเป็น 1 ที่ S1P2 ก็จะไม่เปลี่ยนเป็น 1 ในไชเคลของเครื่องที่ 2 โดยจะเป็น 0 อยู่จนกว่าจะถึงเวลา S4P2 ของไชเคลของเครื่องที่ 2 สัญญาณ ALE จะเปลี่ยนเป็น 1 เพื่อทำการอ่านหรือเขียนข้อมูลกับ Data Memory ภายนอก

1.5 ไคอะแกรมเวลาของการติดต่อกับหน่วยความจำ

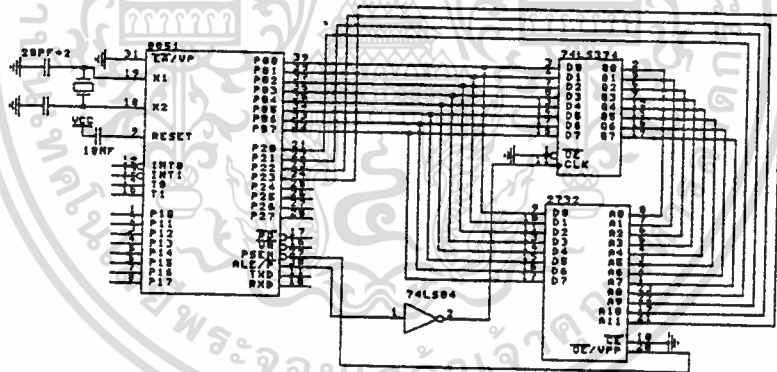
การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 นั้น ลำดับสัญญาณตามเวลา (Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่งมีดังรูปที่ 1.14



รูปที่ 1.14 Timing diagram ของการอ่านโปรแกรมจากหน่วยความจำภายนอก

การอ่านคำสั่ง (Fetch) จาก Program area ภายนอกจะเริ่มจาก 8051 ส่งสัญญาณลอจิก 1 ออกมาทางขา ALE ขณะนี้สัญญาณที่ขา PSEN จะเป็น 1 จากนั้น Port 0 จะส่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าตำแหน่งหน่วยความจำ 8 บิตล่างและ Port 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิตบนออกมา แล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้ขอบขาของสัญญาณ ALE เพื่อ Latch ค่าตำแหน่งหน่วยความจำที่ Port 0 ได้ จากนั้น Port 0 ก็จะยกเลิกการส่งค่าตำแหน่งหน่วยความจำเข้าสู่สภาวะ High Impedance และสัญญาณ PSEN จะเป็น 0 เพื่อเตรียมรับคำสั่งที่ส่งออกจากหน่วยความจำภายนอกเข้าไปยัง 8051 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction Register (ดูรูป 1.4) แล้วสัญญาณ PSEN จะกลับเป็น 1 พร้อมกับสัญญาณ ALE ก็กลับเป็น High เพื่อการอ่านคำสั่งต่อไปทำงาน ข้อมูลใน Port 2 จะคงที่ตลอดเวลา ตั้งแต่สัญญาณ ALE เป็น 1 จนกระทั่งสัญญาณ ALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้งหนึ่งจากนั้นจะเริ่มลำดับการ Fetch ข้อมูลไบต์ที่ 2 จากหน่วยความจำสำหรับโปรแกรม ซึ่งจะมีการเปลี่ยนแปลงของสัญญาณตามเวลาเหมือนกับการ Fetch ไบต์แรกนั่นเอง จาก Timing Diagram ดังกล่าวจะออกแบบวงจรที่ Program Memory อยู่นอก 8051 ได้ดังตัวอย่างในรูปที่ 1.15



รูปที่ 1.15 วงจรที่ Program memory อยู่นอก 8051

74LS374 ในรูปที่ 1.15 จะทำหน้าที่ Latch ค่าตำแหน่งในหน่วยความจำ 8 บิตล่างที่เวลาขอบขาของสัญญาณ ALE ซึ่งสัญญาณ ALE จะถูกกลับให้เป็นตรงข้ามโดย Inverter 74LS04 ก่อนที่จะป้อนให้กับ CLK ของ 74LS374 และขอบขาขึ้นของสัญญาณที่ออกจาก 74LS04 จะ Latch ค่าตำแหน่งหน่วยความจำ ข้อมูลที่ออกจาก 74LS374 จะเป็นค่า 8 บิตล่างของตำแหน่งหน่วยความจำที่ต้องการติดต่อ ในวงจรได้ต่อตำแหน่งหน่วยความจำ 8 บิตเข้ากับ A0 ถึง A7 ของ EPROM และ

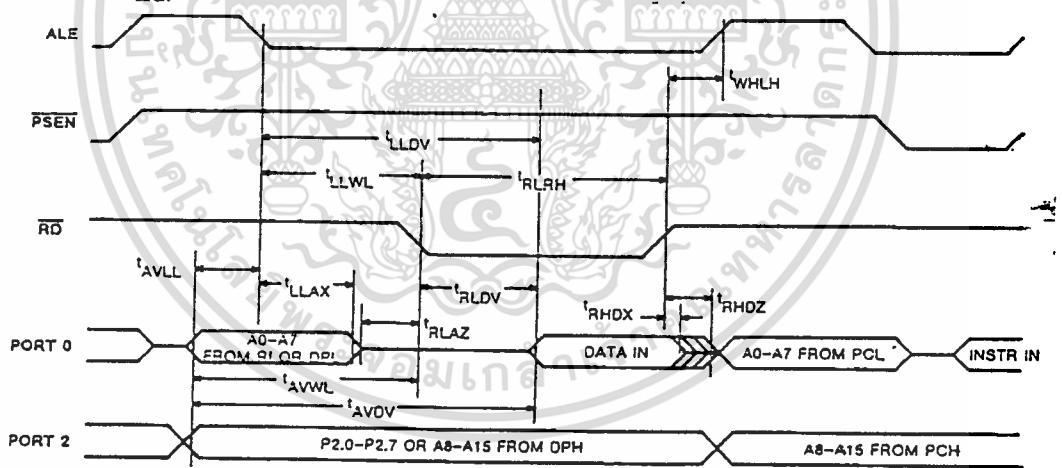
ข้อมูลจาก Port 2 บิต P2.0 ถึง P2.3 จะต่อเข้ากับ A8-A11 ของ EPROM โดยตรงเพราะค่าตำแหน่งหน่วยความจำ 8 บิตบนที่ออกมาจาก Port 2 จะคงที่ตลอดเวลา ขา PSEN ของ 8051 จะถูกต่อเข้ากับขา OE ของ EPROM 2716 ดังนั้นเมื่อสัญญาณ PSEN มีสภาวะลอจิกเป็น 0 ก็จะสามารถส่งค่าสิ่งที่เก็บใน EPROM ณ ตำแหน่งที่ชี้โดยข้อมูลที่ขา A0-A11 ออกมาซึ่ง Port 0 และถูก 8051 เก็บไปทำงานต่อไป

การอ่าน-เขียนข้อมูลกับหน่วยความจำสำหรับข้อมูลภายนอก 8051

การอ่าน-เขียนข้อมูลกับ Data Memory ภายใน 8051 นั้นจะมีสัญญาณสร้างมาจากส่วน Timing and Control โดยที่ผู้ใช้ไม่จำเป็นต้องทำความเข้าใจ แต่การอ่าน-เขียนข้อมูลกับ Data Memory ภายนอก อันเนื่องมาจากคำสั่ง MOVX นั้น เมื่อคำสั่งดังกล่าวถูกอ่านเข้ามาถึง Instruction Register แล้ว Timing and Control จะทำการถอดรหัสแล้วสร้างสัญญาณควบคุมดังนี้

การอ่านข้อมูลจาก External Data Memory จะมีไคและแอมพลิจูดตามเวลาดังรูปที่

1.16

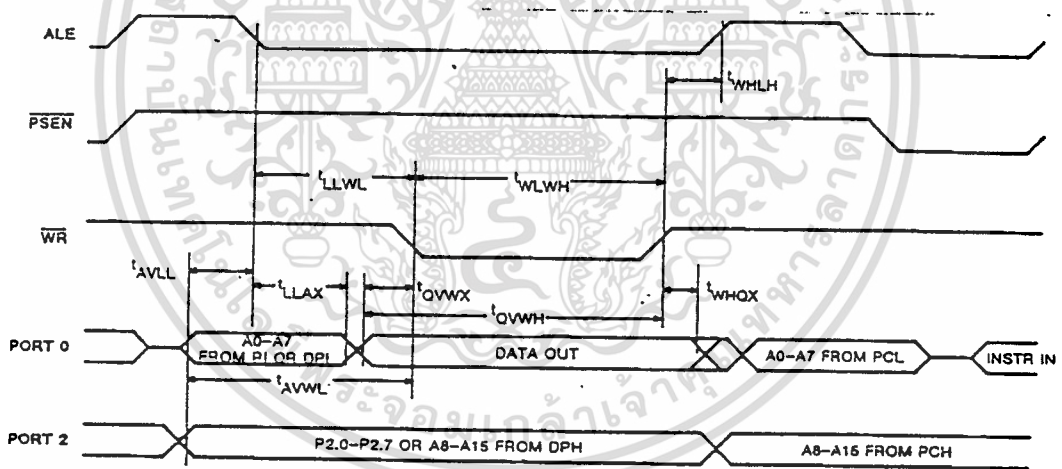


รูปที่ 1.16 Timing diagram ของการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 8051

การทำงานจะเริ่มจากการส่งค่าตำแหน่งหน่วยความจำภายนอก 8 บิตทางออกทางพอร์ต 0 และ 8 บิตบนออกทางพอร์ต 2 เมื่อส่งค่าตำแหน่งแล้ว สัญญาณ ALE ซึ่งเดิมมีลอจิกเป็น 1 จะกลับมาเป็น 0 เพื่อให้อุปกรณ์ภายนอกสามารถ Latch ตำแหน่งหน่วยความจำไว้เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เพื่อส่งไปยังหน่วยความจำ แม้ว่าข้อมูลบนเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ท 0 จะเปลี่ยนแปลงไปก็จะมีค่าตำแหน่งหน่วยความจำส่งไปยังหน่วยความจำในระหว่างการติดต่อกับ Data Memory สัญญาณ PSEN จะเป็น 1 ตลอดเพราะสัญญาณ PSEN จะ Active (เป็น 0) ก็ต่อเมื่อเป็นการติดต่อกับหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เท่านั้น 8051 จะส่งสัญญาณลอจิก 0 ออกมาทางขา RD (P3.7) เพื่อบอกกับหน่วยความจำภายนอกว่าต้องการอ่านข้อมูลเข้าไปเมื่อ 8051 ส่งสัญญาณ RD เป็นลอจิก 0 จะทำให้พอร์ท 0 เข้าสู่สถานะ High Impedance พร้อมทั้งจะให้หน่วยความจำภายนอกส่งข้อมูลมาบนพอร์ท 0 ข้อมูลบนพอร์ท 0 ที่ส่งมาจากหน่วยความจำภายนอกจะถูกอ่านเข้าไปเก็บที่เวลาขอบขาขึ้นของสัญญาณ RD จากนั้นสัญญาณALE ก็จะกลับเป็น 1 เพื่อเริ่มการทำงานในคำสั่งต่อไป ในระหว่างการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอกนี้ พอร์ท 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิตบนออกมาตลอดเวลา

การเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051 จะมีไคอะแกรมสัญญาณตามเวลาดังรูปที่ 1.17

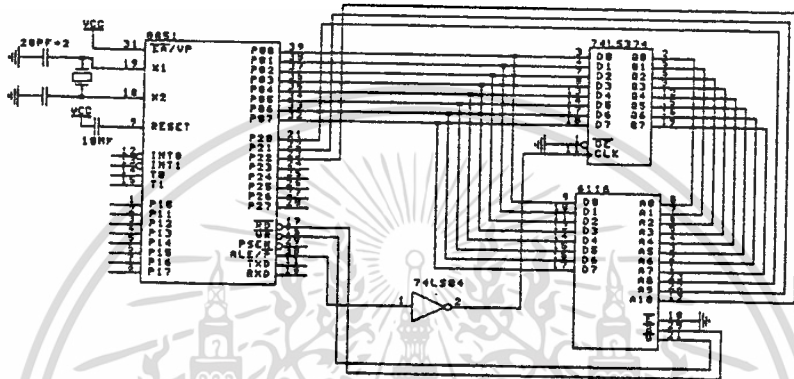


รูปที่ 1.17 Timing diagram ของการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก8051

เมื่อ 8051 ส่งค่าตำแหน่งหน่วยความจำ 8 บิตต่างไปทางพอร์ท 0 และ 8 บิตบนลงไปที่พอร์ท 2 แล้วสัญญาณALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้สัญญาณนี้ในการ Latch ค่าตำแหน่งหน่วยความจำบนพอร์ท 0 เหมือนกับการอ่านข้อมูลจากหน่วยความจำ สำหรับข้อมูลภายนอก เมื่อสัญญาณ ALE เป็น 0 แล้ว 8051 จะส่งข้อมูลที่ต้องการเขียนไปยังพอร์ท 0 แล้วจะให้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ WR เปลี่ยนสถานะลอจิกเป็น 0 ขณะหน่วยความจำภายนอกจะต้องเขียนข้อมูลไปเก็บยังตำแหน่งที่กำหนด จากนั้นสัญญาณ WR จะกลับเป็น 1 เพื่อเป็นการบอกสิ้นสุดการเขียนข้อมูลแล้วสัญญาณ ALE ก็กลับเป็น 1 เพื่อ Fetch คำสั่งต่อไปมาทำงาน

หน่วยความจำสำหรับข้อมูลภายนอกที่สามารถอ่านและเขียนข้อมูลได้ จะสามารถเขียนเป็นวงจรได้ดังรูปที่ 1.18

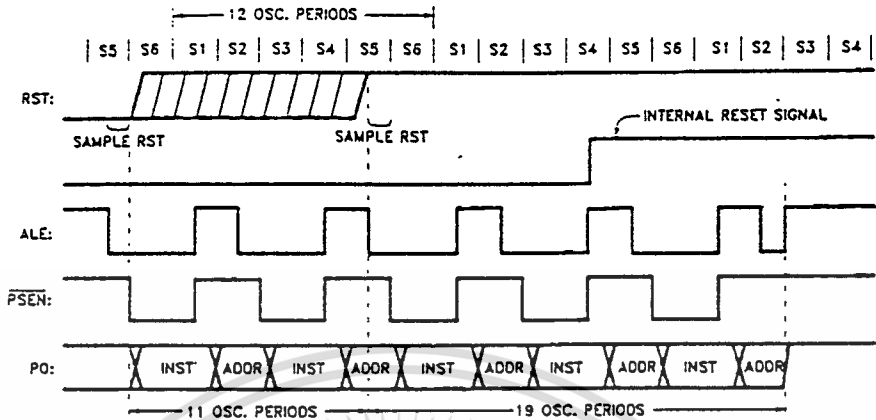


รูปที่ 1.18 วงจรที่มีหน่วยความจำสำหรับข้อมูลที่อยู่นอก 8051

74LS374 ในรูปจะใช้สำหรับ Latch ค่าตำแหน่งหน่วยความจำ 8 บิตต่างไว้ แม้ว่าข้อมูลบนพอร์ท 2 จะเปลี่ยนไปสัญญาณ RD และ WR จะอ่านหรือเขียนข้อมูลจากหน่วยความจำภายนอก 6116 เป็นหน่วยความจำแบบ RAM ที่สามารถอ่านและเขียนข้อมูลได้

1.6 การรีเซ็ต

เมื่อป้อนสัญญาณที่มีสถานะลอจิก 1 เข้าไปทางขา RST จะไม่ได้เกิดการรีเซ็ตขึ้นทันทีทันใด แต่ลำดับเกิดการรีเซ็ตจะแสดงได้ดังโคอะแกรมตามเวลาในรูปที่ 1.19



รูปที่ 1.19 ไคอะแกรมตามเวลาของการรีเซ็ต

ในรูปที่ 1.19 เป็น Timing Diagram ของการรีเซ็ตสภาวะลอจิกของสัญญาณที่ขา RST จะถูกอ่านเข้ามาที่เวลา S5P2 (เฟส 2, State 5) ของทุกๆ ไซเคิลของเครื่อง ในกรณีที่เป็นค่า สิ่งซึ่งมีการทำงานเสร็จสิ้นใน 2 ไซเคิลของเครื่องก็จะตรวจสอบเฉพาะสัญญาณที่อ่านเข้ามาใน ไซเคิลที่ 2 ของการทำงาน ดังนั้นในการรีเซ็ตจะต้องป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขา นี้เป็นเวลาอย่างน้อย 2 ไซเคิลของเครื่องหรือ 24 ไซเคิลของสัญญาณนาฬิกาที่สร้างจากวงจรรอสส์ ซิลเลเตอร์ภายใน 8051 เพื่อให้แน่ใจว่าสัญญาณรีเซ็ตจะถูกอ่านเข้าไปตรวจสอบและทำงาน ขณะที่ ทำการรีเซ็ต 8051 ออสซิลเลเตอร์จึงจะต้องทำงานอยู่ด้วย เมื่อ 8051 สุ่มข้อมูลที่ขา RST แล้ว ตรวจสอบว่าเป็นสภาวะลอจิก 1 ก็จะสร้างสัญญาณรีเซ็ตขึ้นภายใน ที่เวลา S2P4 ของไซเคิลเครื่อง ถัดไป ข้อมูลที่แต่ละพอร์ทส่งออกมาจะยังคงปรากฏที่พอร์ทจนกว่าจะเกิดการรีเซ็ตขึ้นซึ่ง ต้องใช้เวลา 19 ไซเคิลของสัญญาณจากออสซิลเลเตอร์นับตั้งแต่เวลา S5P2 ในไซเคิลของเครื่องที่พบสัญญาณ รีเซ็ต ในระหว่างเวลา 19 ไซเคิลนี้จะต้องมีการ Fetch คำสั่งเข้าไปทำงานได้อยู่

สภาวะของสัญญาณลอจิกที่ RST จะถูกอ่านเข้าไปตรวจสอบที่เวลา S5P2 ของทุกๆ ไซเคิลของเครื่อง ดังนั้นถึงแม้ว่าสัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาก่อนก็ยังไม่เกิดการตรวจสอบสัญญาณรีเซ็ต ดังในรูปที่ 1.20 สัญญาณที่ขา RST อาจเป็น 1 มาตั้งแต่ State ที่ 6 ก็จะไม่ เกิดอะไรขึ้นจนกระทั่ง 1 ไซเคิลของออสซิลเลเตอร์ต่อมาซึ่งเป็นเวลา S5P2 จึงจะเกิดการตรวจสอบสัญญาณที่ขา RST ถ้าคำสั่งนั้นมีการกำหนดมากกว่า 1 ไซเคิลของเครื่อง 8051 ก็จะต้องทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานในคำสั่งนั้นให้เสร็จสิ้นเสียก่อนจึงจะเริ่มการรีเซ็ตได้ โดย 8051 จะดูสถานะของสัญญาณที่ขา RST ของ S5P2 ในไซเคิลของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ของไซเคิลเครื่องแรกๆ ในคำสั่งอาจมีสถานะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไซเคิลของเครื่องสุดท้าย มีสถานะลอจิกที่ขา RST เป็น 0 ก็จะไม่เกิดการรีเซ็ต

ทันทีที่เวลา S5P2 เมื่อตรวจสอบสถานะสัญญาณที่ขา RST แล้วพบว่าเป็น 1 จะต้องรอไปจนถึงเวลา S4P2 ของไซเคิลของเครื่องถัดไปจึงจะทำให้สัญญาณรีเซ็ตภายในเปลี่ยนสถานะลอจิกจาก 0 เป็น 1 ในระหว่างเวลา S5P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนถึง S4P2 ของไซเคิลของเครื่องถัดไปจะยังคงมีการ Fetch คำสั่งเข้าไปทำงานอีก 2 คำสั่ง เมื่อสัญญาณรีเซ็ตภายในเปลี่ยนเป็น 1 ก็จะเริ่มการรีเซ็ต โดยการเขียนข้อมูล 0 ไปยัง Special Function Register ทุกตัวยกเว้นพอร์ท 0 ถึงพอร์ท 3 Stack Pointer และรีจิสเตอร์ SBUF ดังตารางในรูปที่ 1.11 ระหว่างนี้ข้อมูลใน RAM ภายใน 8051 จะไม่เปลี่ยนแปลงข้อมูลในระหว่างการเขียนข้อมูลลงไปยัง SFR จะยังมีการ Fetch คำสั่งเข้ามาอีก 1 คำสั่งจนกว่าจะถึง S3P1 ของไซเคิลของเครื่องที่ 2 (นับแต่ไซเคิลของเครื่องที่ตรวจพบลอจิก 1 ที่ขา RST) ก็จะทำให้สถานะลอจิกที่ขา ALE และ PSEN ค้างอยู่ที่สถานะลอจิก 1 และจะเป็นอย่างนี้ไปจนกว่าสถานะลอจิกที่ขา RST เป็น 0 เวลานั้นตั้งแต่พบลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลาที่ ALE และ PSEN ค้างอยู่ที่ 1 จะเท่ากับ 19 ไซเคิลของออสซิลเลเตอร์เมื่อสัญญาณที่ขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 8051 จะรออีก 1 ถึง 2 ไซเคิลของเครื่อง สัญญาณ ALE และ PSEN จะเริ่มเปลี่ยนแปลงเพื่อเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานจากคำสั่งในหน่วยความจำสำหรับโปรแกรมตำแหน่ง 0000H

บทที่ 2

การออกแบบตัวควบคุม 8051

การแนะนำ ในบทนี้เป็นเรื่องขององค์ประกอบทางด้าน hardware สำหรับตัวควบคุม 8051 ซึ่งจะใช้เป็น hardware สำหรับตัวอย่างการประยุกต์ใช้งานในบทต่อไป

การออกแบบตัวควบคุม เริ่มจากการแยกแยะความต้องการและความว่างเปล่าของกระดาษ หรือจอคอมพิวเตอร์ การพัฒนาของตัวควบคุมจะตามมาในขั้นตอนต่างๆดังนี้

1. การกำหนดรายละเอียด
2. ออกแบบระบบตัวควบคุมตามรายละเอียด
3. เขียน program ตรวจสอบการออกแบบ
4. เขียนและทดสอบ subroutines ที่ใช้บริการต่างๆไป

ขั้นตอนแรกในการออกแบบมีความสำคัญมาก ถ้าขั้นตอนนั้นใช้เป็นแบบของผลผลิตจำนวนมาก (มากกว่า 1000 หน่วย) ดังนั้นจะเป็นภาระที่ต้องวิเคราะห์อย่างระมัดระวัง ความพิถีพิถัน ความรัดกุมในรายละเอียด จะเพิ่มมากขึ้นในส่วนของการลงทุนในอุตสาหกรรม เพื่อให้รายละเอียดลดลง โดยทั่วไปผู้ออกแบบจะเขียนรายละเอียดที่กระชับ

ในความเป็นจริง ส่วนใหญ่จะไม่สามารถออกแบบตัวควบคุมได้ใหม่ หลังจากที่มันถูกพัฒนาจนไกลจากจุดเริ่มต้น เนื่องจากเราไม่สามารถเข้าใจที่มาของตัวควบคุม จากประสบการณ์ของผู้ออกแบบ รายละเอียดที่สมบูรณ์ จะต้องการ PORT I/O มากกว่า 1 PORT และต้องการหน่วยความจำมากกว่า 1 กิโลไบต์

Microcontroller Specification

ตัวอย่างสำหรับการออกแบบตัวควบคุมมีดังนี้ การควบคุมเวลาและข้อมูลที่แน่นอนจำเป็นมากสำหรับตัวควบคุมที่ฉลาด ตัวควบคุมจะเป็นส่วนหนึ่งของระบบเครือข่ายคอมพิวเตอร์ที่ติดต่อกันทางPORT อนุกรม ถ้าตัวควบคุมถูกผลิตออกมาเป็นจำนวนน้อย ตัวอย่างเช่น น้อยกว่า 1000 หน่วยในงานใดงานหนึ่ง ราคาของระบบต้องมีราคาต่ำ

ตระกูลของ 8051 มีลักษณะดังนี้

1. ราคาถูก
2. มีผู้จำหน่ายมาก
3. เหมาะกับการใช้งานร่วมกับ NMOS และ CMOS
4. Software ง่ายและราคาถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Compiler เป็นภาษาชั้นสูง

ลักษณะในข้อ 1-3 มีความสำคัญมากจาก ราคาที่เป็นมาตรฐาน ทำให้สามารถอาศัย software ช่วยลดต้นทุนและทำให้ project สมบูรณ์ได้

ความต้องการในการเปลี่ยนแปลงโปรแกรมเพื่อใช้กับงานใดงานหนึ่ง มักจะต้องการ EPROM จากภายนอก ซึ่งต้องการใช้ PORT 0 เป็นขา address A0-A7 และ PORT 2 เป็นขา address A8-A15 เพราะฉะนั้น PORT 0 และ 2 จะไม่เหมาะสมในการใช้เป็น I/O PORT

เนื่องจากความเป็นไปได้ที่ตัวควบคุมจะมีข้อมูลที่มากเกินไปกว่า RAM ภายนอกจะเก็บไว้ได้ เพราะฉะนั้นอาจจำเป็นต้องใช้ RAM ภายนอก ซึ่งมีสัญญาณควบคุมการเขียน (WR. Active low) บิตที่ 6 ของ PORT 3 และสัญญาณควบคุมการอ่าน (RD. Active low) บิตที่ 7 ของ PORT 3 สำหรับหน่วยความจำภายนอกขนาด 64 K จะมีขาทั้งหมด 28 ขา

ตัวอย่างนี้ใช้ได้กับ EPROM ตั้งแต่ 3 K จนถึง 32K ขนาดของหน่วยความจำที่แตกต่างกัน สามารถเลือกได้จาก jumper เพื่อเพิ่มขา address ให้กับหน่วยความจำขนาดใหญ่และใช้ความต้านทาน pull up เพื่อสามารถใช้กับหน่วยความจำขนาดเล็กได้

ความต้องการในการส่งข้อมูลอนุกรมสามารถใช้ PORT อนุกรมภายนอกซึ่งมีสัญญาณที่ใช้ในการควบคุม bit ที่ 0 ของ PORT 3 เป็นสัญญาณ RXD และ bit ที่ 1 ของ PORT 3 เป็นสัญญาณ TXD เพราะฉะนั้นเราสามารถให้ PORT 1 ทั้งหมดและ PORT 3 pin 2-5 เป็น I/O หรือเป็น interrupt จากภายนอก และ timing input ได้

การสูญเสียความสามารถของ I/O ที่เกิดขึ้นจากการนำไปใช้เป็น port function เพื่อความถูกต้องของ program ข้อเสียนี้จะไม่มาก แต่อย่างไรก็ตามมี 2 วิธีในการขยาย I/O คือ จาก I/O ของคอมพิวเตอร์และ Memory map I/O

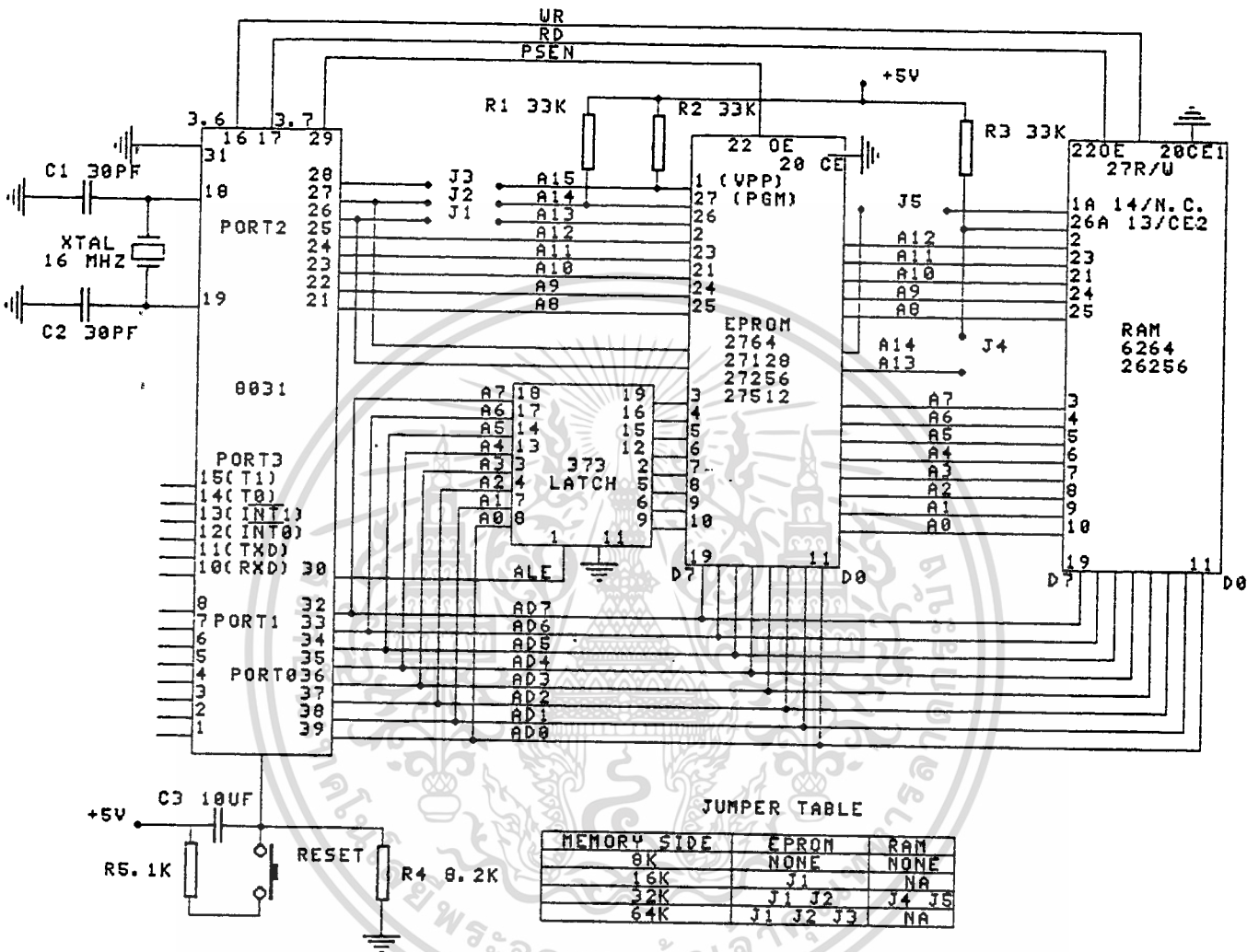
สุดท้ายนี้ เราใช้ crystal ความถี่ 18 MHz รายละเอียดคุณสมบัติดังนี้

- 80C31-1 (ไม่มี ROM ภายนอก) เป็นตัวควบคุม
- EPROM ภายนอกขนาด 64 KByte
- RAM ภายนอกขนาด 32 KByte
- 8 port ที่ใช้ในการทั่วไป
- 4 port ที่ใช้ในการควบคุม program
- 1 full-duplex serial port
- clock ความถี่ 16 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microcontroller Design

รูป 2.1 แสดงการออกแบบตัวควบคุมที่มีรายละเอียดดังนี้



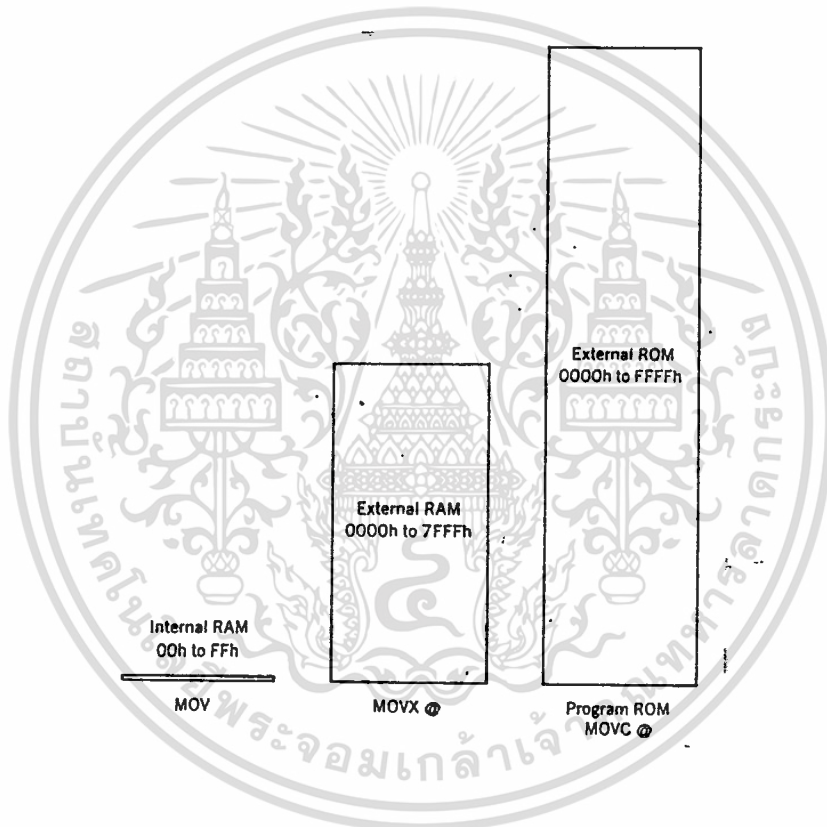
รูปที่ 2.1

หน่วยความจำภายนอกและการถอดรหัสหน่วยความจำภายนอก

ให้ port 0 ทำหน้าที่เป็น port ข้อมูลและเป็น port address byte ต่ำ ส่วน port 2 เป็น port address byte สูง ข้อมูลและ address byte ต่ำของ port 0 จะมี time multiplex ใช้ 373 ที่ต่อกับ port 0 เป็นตัว latch ค่า address byte ต่ำ โดยให้สัญญาณ ALE จาก 8051 เป็นสัญญาณในการ latch และ port 0 จะเป็นได้ทั้ง input และ output สำหรับอ่านและเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

address ของ RAM และ ROM จะต่างกันที่สัญญาณ PSEN (Active low) เมื่อ PSEN Active แสดงว่าเป็นการติดต่อกับ ROM และสัญญาณ WR (Active low) หรือ RD (Active low) จะใช้สำหรับ RAM และผลจาก address ที่แตกต่างกันของหน่วยความจำแต่ละขนาด ทำให้เราต้องใช้ jumper และความต้านทาน pull up ดังนั้นผู้ใช้สามารถใช้ขนาดหน่วยความจำตามต้องการ ตาราง jumper ในรูป 2.1 ใช้สำหรับ EPROM และ RAM รูป 2.2 แสดงหน่วยความจำภายในและภายนอก



รูปที่ 2.2

วงจร Reset และ Clock

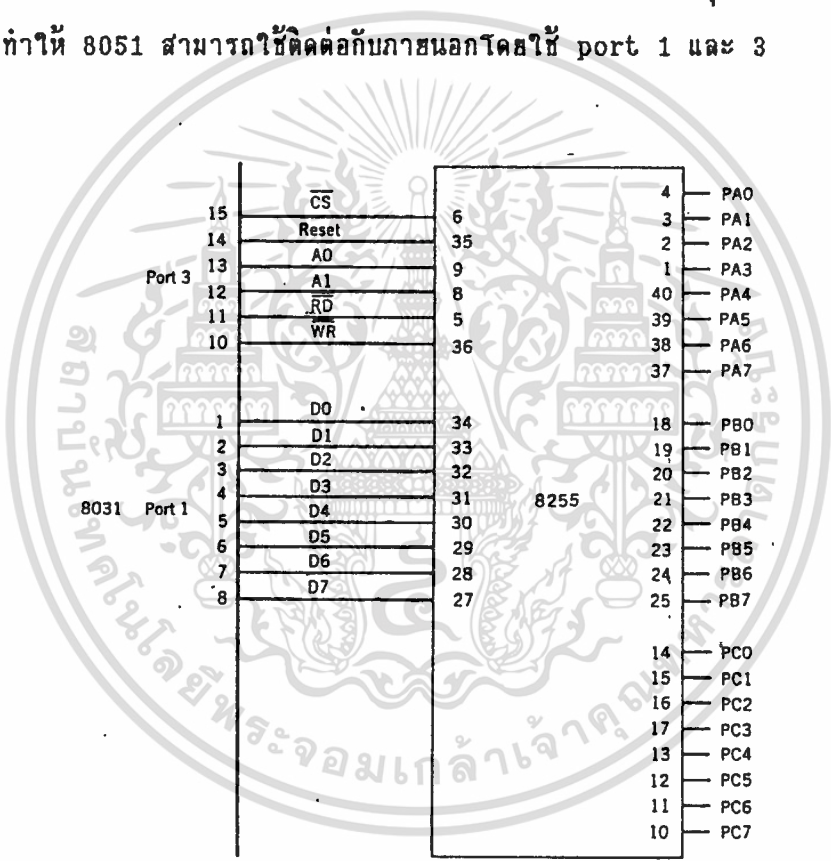
ใน 8051 ภา Reset จะต้องเป็น high ใน 2 machine cycle เมื่อเริ่มจ่ายไฟเลี้ยง และหลังจากนั้นภา Reset จะเป็น low โดยใช้วงจร RC เมื่อเริ่มจ่ายไฟเลี้ยงที่ภา Reset จะเป็น high จนกระทั่งตัวเก็บประจุ charge ไฟไว้ราวๆ 2.5 V จึงทำให้ภา Reset ยังมี Switch สำหรับ Reset ระบบเมื่อผู้ใช้ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Expanding I/O

Port 1 และ 3 สามารถใช้เป็นขาควบคุม และขาเข้าออกของข้อมูล data bus ใช้เชื่อมโยงกับภายนอกเพื่อขยาย I/O chip I/O ที่นิยมใช้เป็น port คือ 8255 ใน 8255 จะมี Register Control word สามารถ set โดย software เพื่อกำหนดให้ port ของ 8255 ที่ชื่อว่า port A,B,C มีความสามารถเป็น input port,output port

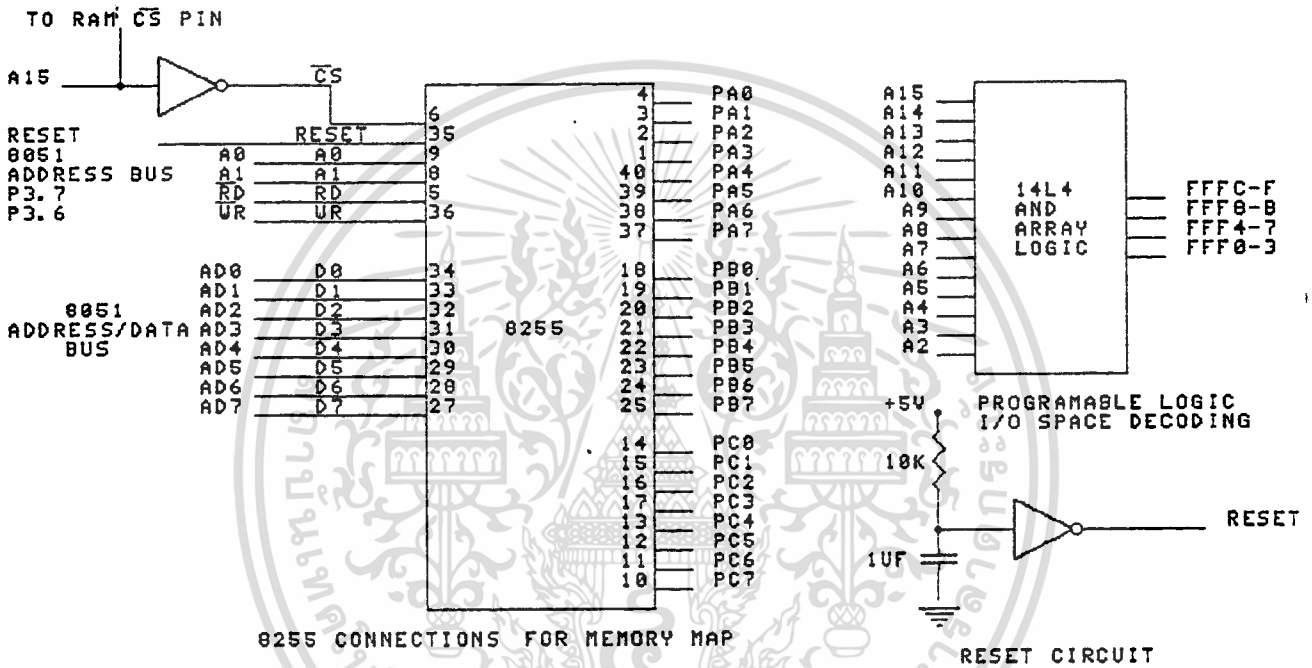
รูป 2.3แสดงวงจรที่เพิ่ม 8255 เข้าไปในระบบที่ทำให้ขณะนี้มี port ขนาด 8 bit อยู่ 3 port ปัญหาของการขยาย I/O คือ ความเร็วที่เกิดจากการที่ต้องการเวลาที่ใช้ในการเขียน control bit ใน port 1 และ 3 ก่อนที่ I/O จะทำงานได้ ส่วนจุดเด่นของการขยาย I/O port คือ ทำให้ 8051 สามารถใช้ติดต่อกับภายนอกโดยใช้ port 1 และ 3



รูปที่ 2.3

Memory-Mapped I/O

chip ที่ใช้ขยาย port สามารถใช้พื้นที่ของหน่วยความจำที่ไม่ถูกใช้ในการออกแบบแสดงไว้ในรูป 2.4 เราสามารถออกแบบได้ใน 31K ด้านบน (8000H หรือสูงกว่า) สำหรับการขยาย port โดย A15 เป็น high เนื่องจาก 32K ของ RAM (7FFFH หรือต่ำกว่า) A15 จะเป็น low ดังนั้นต้องการ inverter เพื่อถอดรหัสระหว่างพื้นที่ของหน่วยความจำ และ I/O



รูปที่ 2.4

การออกแบบควรจะถอดรหัสให้พื้นที่ที่ใช้สำหรับ RAM เป็นจำนวนมากและควรระบุตำแหน่งที่แน่นอนในการติดต่อกับ chip I/O รูป 4 แสดงการออกแบบที่รวม 3 memory-mapped port ที่ address FFF0H-FFF3H, FFF4H-FFF7H, FFF8H-FFFBH และ FFFCH-FFFFH โดย RAM สามารถใช้พื้นที่จาก 0000H-FFE7H ได้

Memory-Mapped I/O มีจุดเด่นที่ไม่ต้องใช้ port ของ 8051 แต่ข้อเสียก็คือ ทำให้สูญเสียพื้นที่ของ RAM เพื่อนำมาใช้เป็น address ของ I/O ลักษณะส่วนใหญ่ของโปรแกรมจะเหมือนกับ port I/O แตกต่างกันที่ใช้คำสั่ง Movx กับ Memory-Mapped I/O

สำหรับการขยาย I/O ใน 2 แบบนั้น จะพิจารณาการเลือกใช้ที่ราคาของระบบซึ่งต้องดูจาก ความต้องการ I/O port มากน้อยเท่าใด และขนาดของหน่วยความจำ

Part Speed

ข้อพิจารณาข้อหนึ่งที่มีจะไม่ปรากฏในการออกแบบ คือส่วนของความถี่ที่ใช้ในระบบหน่วยความจำทั้งหมดจะใช้เวลาในการเข้าถึงข้อมูลเป็น ns (เวลาที่ใช้ในการอ่านหรือเขียนข้อมูลหลังจากกำหนด address แล้ว) เวลาที่ใช้ขึ้นจะต้องมีค่ามากกว่าคาบเวลาในการออกแบบระบบเวลาที่ใช้ในการอ่านหรือเขียนข้อมูลจาก RAM กับเวลาที่ใช้ในการอ่านข้อมูลจาก ROM ซึ่งเวลาทั้งหมดจะเป็นตัวเลือกใช้ความถี่ของระบบ และในตัวอย่างของเรา EPROM มีเวลาที่มากที่สุดในการเข้าถึงข้อมูล คือ 150 ns และ RAM ใช้เวลา 400 ns ในส่วนอื่น เช่น 373 ซึ่งใช้เป็นตัว latch สามารถใช้ตระกูล LSTTL ถึงตระกูล HCMOS เพราะความเร็วของส่วนนี้จะห่างจากความเร็วของ 8051 มาก

สรุป

การออกแบบระบบ microprocessor ใช้ 8051 เป็นตัว controller ร่วมกับ External EPROM และ RAM ที่สามารถเลือกขนาดได้ ระบบที่ออกแบบมีดังนี้

RAM ภายนอกขนาด 8K-32K byte

ROM ภายนอกขนาด 8K-64K byte

I/O port 1-8 bit จาก port ของ 8051

port อื่นๆมี port 3.0 (RXD)

3.1 (TXD)

3.2 (INT0, Active low)

3.3 (INT1, Active low)

3.4 (T0)

3.5 (T1)

crystal ความถี่ 16 MHZ

สามารถใช้ crystal ความถี่อื่นๆกำเนิดจังหวะเวลาให้กับ Timer ได้ต้องเขียน program ขึ้นมาเพื่อทดสอบการอ่านข้อมูลจาก ROM และทดสอบการเขียนและอ่านข้อมูลจาก RAM

Subroutine ที่ถูกพัฒนาโดยคำสั่งใน 8051 เพื่อนำไปประยุกต์ใช้มีดังนี้

Timer delays: software; timer, software polled; timer,

interrupt driven

Lookup Tables: PC base, DPTR base

Serial data communications transmission: time delay, software polled, interrupt driven

Serial data communications reception: software polled, interrupt driven

การประยุกต์ใช้งานไอซี 8051

การวัดพัลส์

เช่น เซอร์คิวท์ใช้สำหรับแอนะล็อกเซ็นเซอร์ ที่ควบคุมทั้งทางอุตสาหกรรมและพาณิชย์ บ่อยครั้งที่สร้างพัลส์ที่มีข้อมูลเกี่ยวกับการตรวจสอบปริมาณ ความถี่เข้าที่ทุกของเซ็นเซอร์ชนิดต่างๆ ให้ duty cycle คงที่ แต่พัลส์ของความถี่ที่เปลี่ยนแปลงได้เป็นตัวชี้การเปลี่ยนแปลงในตัวแปรที่สามารถวัดได้ เป็นแบบที่ธรรมดาที่สุด การเปลี่ยนระยะความกว้างพัลส์มีผลในความถี่ที่คงที่ ความถี่ที่วัด

Timer T0 และ T1 ใช้วัดความถี่ภายนอก โดย timer ตัวหนึ่งเป็น counter และใช้อีกตัวสร้างช่วงเวลา timing ในการนับครั้งแรก ความถี่ของขบวนพัลส์ที่วัดเป็น

$$\text{unknown freq.} = \text{counter/timer}$$

สำหรับตัวอย่าง ถ้า counter นับ 200 พัลส์ ในเวลา 0.1 s ที่สร้างโดย timer ความถี่คือ

$$UF = 200/0.1 = 2000 \text{ Hz}$$

พัลส์อินพุตต้องเปลี่ยนจาก 1 เป็น 0 ใน 2 แมกซ์ไซเคิลสุดท้ายหรือ $f/24$ ถูกนับข้อจำกัดบนการเปลี่ยนแปลงพัลส์นี้ จะให้ความถี่ 667 Hz ซึ่งใช้คริสตอล 16 MHz

ความถี่ต่ำสุดที่สามารถนับ ถูกจำกัดโดยระยะเวลาที่สร้างซึ่งสามารถนานเกินกว่าการใช้แรมทั้งหมดนับ timer rollovers มีข้อจำกัดที่ไม่สามารถทำได้บนความถี่ที่ต่ำที่สุดที่ถุกนับ

โพลีซีที่เซ็นเซอร์เปลี่ยนความถี่ได้ ส่วนมากสร้างสัญญาณที่อยู่ในช่วง 0 ถึง 667 Hz ปกติแล้วสัญญาณอยู่ในช่วง 1,000 ถึง 10,000 Hz

ตัวอย่างที่ใช้เซ็นเซอร์วัดโวลต์เดจ dc จาก 0 ถึง 5 โวลต์ที่ 0 โวลต์เข้าที่ทุกเซ็นเซอร์เป็น 1,000 Hz และที่จำกัด 5 โวลต์ เข้าที่ทุกเป็น 10,000 Hz และเราต้องการวัดโวลต์เดจให้ใกล้ 0.01 โวลต์ มากที่สุด หรือ 10 Hz ช่วงเวลา 1 s สร้างความถี่ที่วัดได้อย่างแน่นอน

คือเข้าใกล้ 1 Hz

การวัดความกว้างพัลส์

ในทางทฤษฎี ถ้ารู้ว่าพัลส์อินพุตเป็น square wave ความถี่ของพัลส์สามารถวัดได้โดยหาเวลาที่คลื่นเป็น high (T_h) ความถี่จะเป็น

$$UF = 1/(T_h * 2)$$

ถ้า T_h เป็น 200 μs ดังตัวอย่างนี้ จะได้ $UF = 2500$ Hz ความแน่นอนในการวัดจะเป็นคลื่นอินพุตที่มี duty cycle 50%

timer x อาจถูกใช้เพื่อให้ clock ภายในถุนับ เฉพาะเมื่อขา INTX เป็น high โดยเซตบิตที่ Gate x ใน TMOD ความแน่นอนในการวัดจะใกล้เคียงกับ 1/2 ของคาบของ timer หรือ 0.375 μs สำหรับคริสตอล 16 MHz ความแน่นอนนี้หาได้ก็ต่อเมื่อเริ่มต้นวัดพัลส์อินพุตที่ low และหยุดเมื่ออินพุตต่อไปเป็น low ความกว้างของพัลส์ที่มากกว่าความสามารถของ counter คือ 49.152 μs สามารถวัดโดยนับ overflow ของแฟลค timer และเพิ่มข้อมูลใน counter

เพื่อจะวัดความแน่นอน TO จะใช้นับ clock ภายใน เมื่อ INTO เป็น high การวัดจะไม่เริ่มจนกว่า INTO เปลี่ยนจาก high เป็น low ใช้เวลาอย่างน้อย 100 μs เพื่อเริ่ม TO การวัดทำในขณะที่ INTO เป็น high และหยุดเมื่อ INTO เป็น low อีกครั้ง ขบวนการนี้สามารถถูกอินเทอร์รัพท์กับ INTO แฟลค IEO สามารถเซตเมื่อ INTO เปลี่ยนจาก high เป็น low เพื่อบอกให้โปรแกรมเริ่มนับความกว้างพัลส์ และต่อไปก็หยุด การอินเทอร์รัพท์หลายอัน

8051 มีอินเทอร์รัพท์จากภายนอก 2 อัน คือที่ขา INTO และ INT1 ซึ่งพอเพียงสำหรับระบบเล็กๆ แต่ความต้องการอาจจะเพิ่มมากกว่า 2 ขานี้ มีอยู่หลายลักษณะที่ขยายจำนวนชุดอินเทอร์รัพท์ ซึ่งขึ้นอยู่กับลักษณะดังนี้

การต่อแหล่งอินเทอร์รัพท์เป็นส่วนร่วม

การกำหนดแหล่งอินเทอร์รัพท์ที่ใช้ซอฟต์แวร์

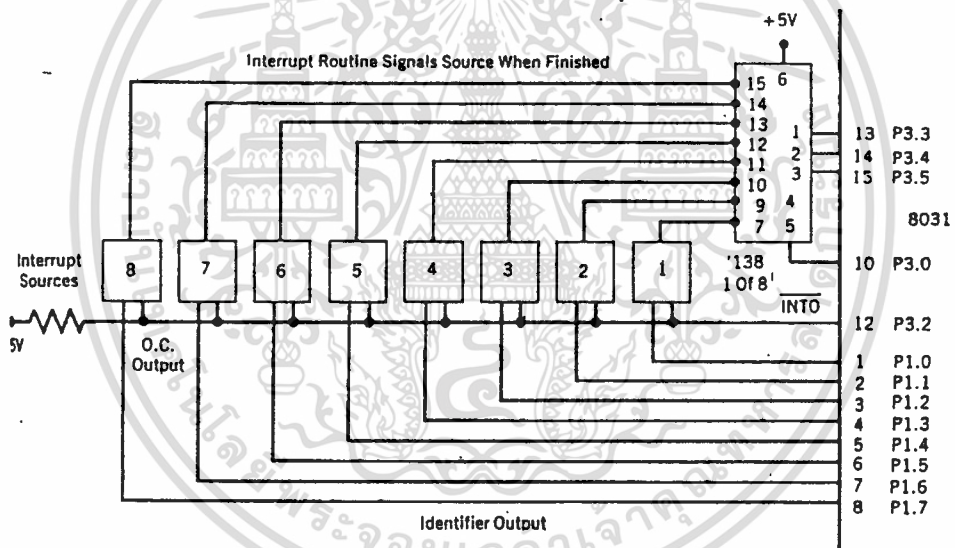
เพราะว่าอินเทอร์รัพท์ภายนอกจะ Active low การต่อจากแหล่งอินเทอร์รัพท์ไปยังขา INTX ต้องใช้ open-collector หรืออุปกรณ์ tri-state

ตัวอย่างการเพิ่ม INTO จาก 1 เป็น 8 ชุด ด้วยรูป 2.5 แต่ละแหล่งจะ Active low เมื่อต้องการอินเทอร์รัพท์ขาของพอร์ตหนึ่งที่สมันจะได้อินเทอร์รัพท์ที่กำหนดมา โปรแกรมอิน

เทอร์ริฟท์จะไปจัดการอินเทอร์ริฟท์ แหล่งอินเทอร์ริฟท์ต้องรับรู้เพื่อว่า สาขอินเทอร์ริฟท์ของแหล่ง สามารถกลับเป็น high พอร์ต 3 ขา 3.3, 3.4 และ 3.5 ให้ผ่าน 3-to-8 decoder สัญญาณ ป้อนกลับจะกลับไปยังแหล่งอินเทอร์ริฟท์ที่ถูกต้อง ตัวถอครหัสใช้โดยขา 3.0

การอินเทอร์ริฟท์หลายๆตัวพร้อมกัน จะจัดการโดยโปรแกรมที่ซับซ้อน ถ้าไม่มีความต้องการอินเทอร์ริฟท์ใดๆอย่างเร่งด่วน แล้วก็สามารถจัดการที่ขาพอร์ต 1 ในการสแกนค่า low

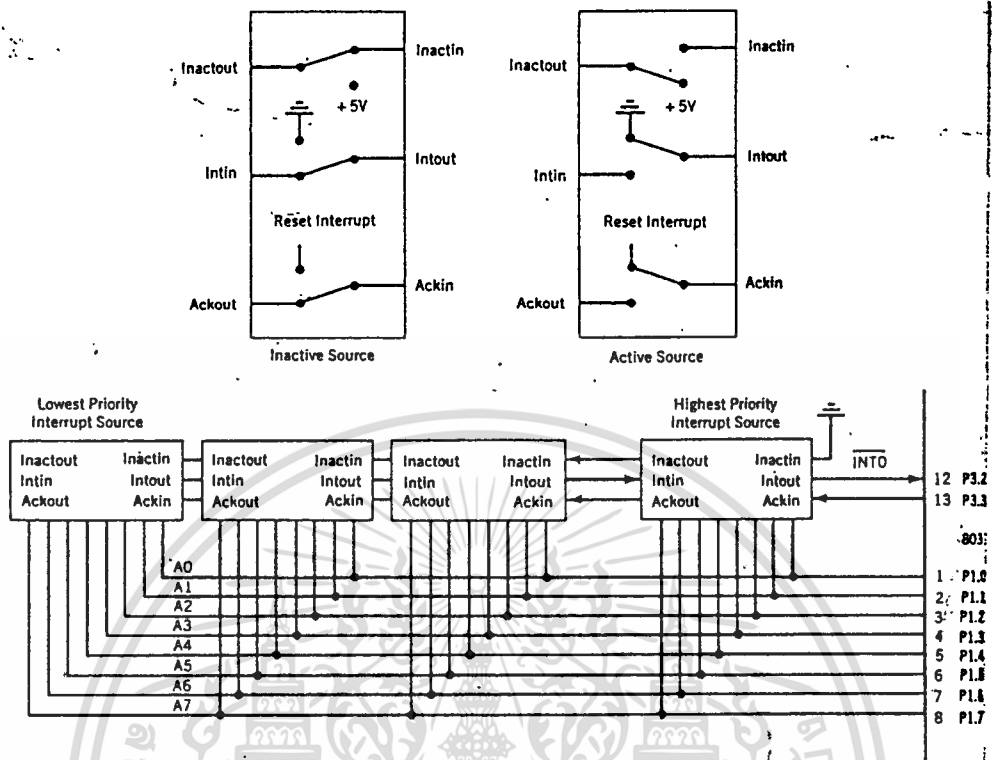
ระบบ priority ขรรมคาสามารถสร้างเมื่อแหล่งอินเทอร์ริฟท์ที่ค่าคัญมากที่สุดตรวจสอบ เป็นลำดับแรก และโปรแกรมอินเทอร์ริฟท์จะทำงานจนกว่าจะเสร็จระบบ priority ที่ละเอีศด เกี่ยวข้องกับลำดับการมาก่อนของแต่ละแหล่ง ระบบที่ละเอีศดจะรู้การอินเทอร์ริฟท์โดยทันที ดังนั้น การรีเซตสาขอินเทอร์ริฟท์และเริ่มจัดการโปรแกรมอินเทอร์ริฟท์ของแหล่ง อินเทอร์ริฟท์ตัวใหม่ที่ค่า คัญกว่า จะทำให้โปรแกรมอินเทอร์ริฟท์ที่ทำอยู่ก่อนหยุด และทำตัวอินเทอร์ริฟท์ใหม่



รูปที่ 2.5

วงจรรีอาร์ดแวร์สำหรับอินเทอร์ริฟท์หลายตัว

ผลของปัญหา เรื่องอินเทอร์ริฟท์ที่มีมากขึ้น จะเน้นการใช้วงจรรภายนอกที่น้อยที่สุดที่จะจัดการกับอินเทอร์ริฟท์หลายๆตัว ที่เกิดพร้อมกัน วิธีทางฮาร์ดแวร์สามารถถกขสาเป็นแหล่งอินเทอร์ริฟท์ 256 แห่งดังรูป 2.6 วงจรนี้เป็นที่นิยมมาก



รูปที่ 2.6

ความคิดทั้งหมดในการออกแบบมีดังนี้

1. แหล่งอินเทอร์รัพท์ที่สำคัญที่สุด ถูกต่อกันในลักษณะลูกโซ่ กับแหล่งที่สำคัญน้อยกว่าแหล่งที่สำคัญน้อยกว่าจะอยู่ข้างหลัง
2. แหล่งอินเทอร์รัพท์สามารถหยุดสัญญาณทั้งหมดที่อยู่ข้างหลังทุกแหล่ง
3. รูทีนของอินเทอร์รัพท์สร้างสัญญาณ ACKIN แก่แหล่งจาก 8051 ที่จุดสิ้นสุดของรูทีน ต่อจากนั้นแหล่งอินเทอร์รัพท์จะเคลื่อนย้าย INOUT และแอดเดรสของแหล่งขนาด 8 บิตเมื่อรับรู้การอินเทอร์รัพท์ แหล่งอินเทอร์รัพท์ต้องนำสาย INOUT เป็น high อย่างน้อยที่สุด 1 แมซซึนไซเคิล อินเทอร์รัพท์สามารถจำการเปลี่ยนจาก high เป็น low บน INTO ได้

บทที่ 3

ทฤษฎีการสื่อสารข้อมูลและการจัดการข้อมูล

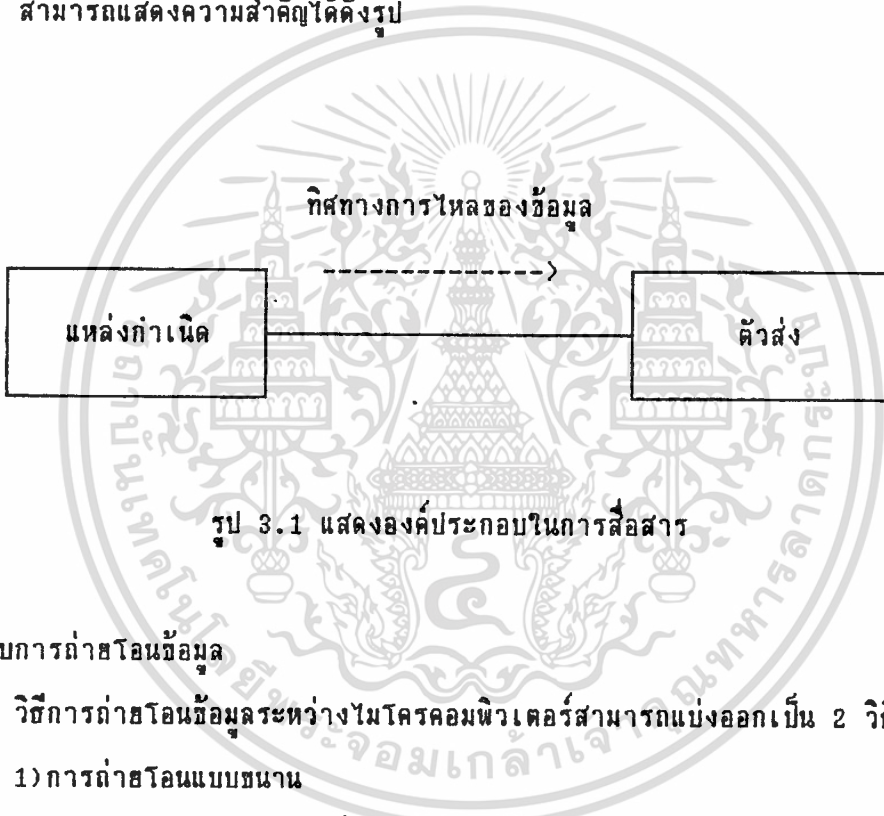
ในบทนี้จะกล่าวถึงหลักการเบื้องต้นของการสื่อสารข้อมูลโดยอาศัยไมโครคอมพิวเตอร์

1. องค์ประกอบเบื้องต้นในการสื่อสารข้อมูล

ในการสื่อสารข้อมูลต้องมีปัจจัยอย่างน้อย 3 ประการคือ

- 1) แหล่งข้อมูล (source)
- 2) ตัวกลาง (medium)
- 3) ตัวรับ (receiver)

สามารถแสดงความสำคัญได้ดังรูป



รูป 3.1 แสดงองค์ประกอบในการสื่อสาร

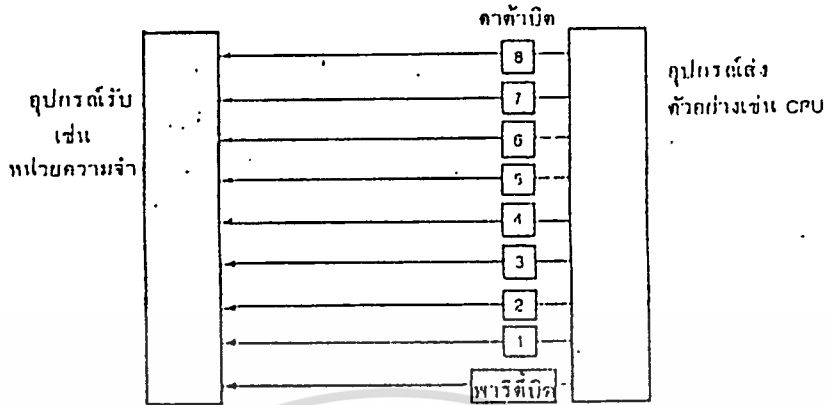
2. รูปแบบการถ่ายโอนข้อมูล

วิธีการถ่ายโอนข้อมูลระหว่างไมโครคอมพิวเตอร์สามารถแบ่งออกเป็น 2 วิธีคือ

1) การถ่ายโอนแบบขนาน

ในระบบไมโครคอมพิวเตอร์ทั่วไป การส่งข้อมูลระหว่างอุปกรณ์ต่างๆที่อยู่บนแผ่นวงจรรล้นแล้วแต่เป็นการส่งแบบขนานทั้งสิ้น การส่งลักษณะนี้ทำได้โดยส่งข้อมูลออกมาพร้อมๆกันทุกๆบิต ดังนั้นเส้นทางของข้อมูลจึงมีเท่ากับจำนวนบิต เช่น ถ้าเป็นไมโครคอมพิวเตอร์ 8 บิต สายส่งข้อมูลก็จะมี 8 เส้นด้วย

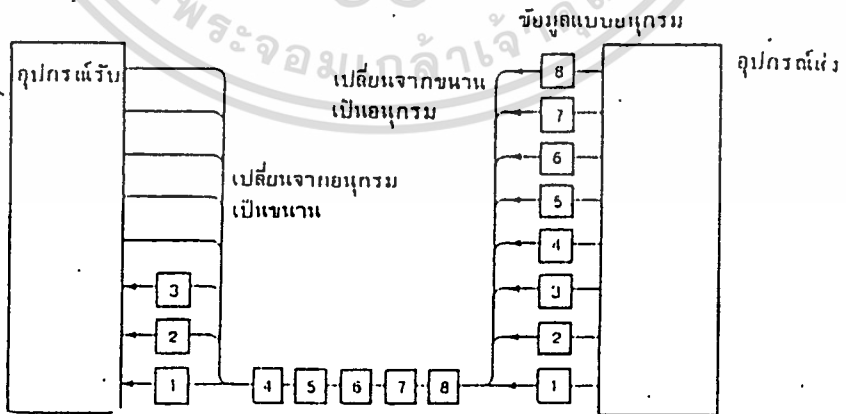
ลักษณะการถ่ายโอนข้อมูลแบบขนานจะเป็นดังรูป 3.2



รูปที่ 3.2 การถ่ายโอนข้อมูลแบบขนาน

2) การถ่ายโอนแบบอนุกรม

การส่งแบบอนุกรมนี้ ข้อมูลแต่ละบิตจะถูกส่ง เรียงกันออกไป เป็นลำดับต่อเนื่องกันทีละบิตระหว่างจุดส่งและจุดรับ จากการศึกษาในลักษณะนี้จึงใช้สายส่งข้อมูลเพียงคู่เดียวเท่านั้น การถ่ายโอนข้อมูลแบบอนุกรมนี้ถึงแม้จะช้าแต่ก็เหมาะกับการส่งระยะไกลๆ ทั้งนี้เพราะประหยัดสายส่งได้เป็นจำนวนมาก ลักษณะการถ่ายโอนข้อมูลจะเป็นดังรูป 3.3



รูปที่ 3.3 การถ่ายโอนข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานและแบบอนุกรม

| | แบบขนาน | แบบอนุกรม |
|-------------------------|--|---|
| 1. ระยะทาง | ปกติจะน้อยกว่า 100 ฟุต | ส่งได้ตั้งแต่ระยะสั้นๆจนถึงระยะทางเป็นไมล์ |
| 2. ความเร็ว | อัตราความเร็วสูงมากในระยะที่ไม่ไกลมากนัก กำหนดได้เป็นจำนวนบิตต่อวินาที | อัตราความเร็วของข้อมูลที่ใช้กันอยู่ทั่วไป จะอยู่ในช่วง 0 ถึง 2 ล้านบิตต่อวินาที |
| 3. ระดับของสัญญาณ | ในการอินเทอร์เฟซจะใช้ระดับของสัญญาณที่ใช้กับอุปกรณ์ TTL คือ สัญญาณลอจิก 1 และ 0 จะแทนด้วยระดับแรงดัน +5 V. และ 0 V. ตามลำดับ | ใช้มาตรฐานของ EIA RS 232C คือมีระดับสัญญาณไฟฟ้าขนาด 12V. หรืออาจใช้มาตรฐาน 20 mA current loop หรืออาจใช้ระดับสัญญาณของ TTL ก็ได้ |
| 4. ความผิดพลาดของสัญญาณ | ถ้าส่งในระยะทางไกลๆความผิดพลาดของข้อมูลจะเกิดได้ง่าย | การผิดพลาดของสัญญาณจะมีน้อยลง |
| 5. ค่าใช้จ่าย | ถ้าส่งในระยะทางไกลๆจะสิ้นเปลืองค่าใช้จ่ายมาก เพราะต้องใช้สายส่งสัญญาณหลายเส้น | สิ้นเปลืองน้อยกว่าหลายเท่า ถึงแม้ว่าจะใช้อุปกรณ์เปลี่ยนสัญญาณของข้อมูลแบบขนานไปเป็นแบบอนุกรมแล้วส่งผ่านสายส่งใช้อุปกรณ์ในการแปลงสัญญาณกลับมาเป็นแบบขนานอีก ก็ยังลงทุนน้อยกว่า |

เนื่องจากความจริงแล้วภาสในไมโครคอมพิวเตอร์มีการรับส่งข้อมูลแบบขนาน ดังนั้นจึงจำเป็นต้องมีการแปลงสัญญาณเข้าให้เหมาะสมเสียก่อน โดยการแปลงสัญญาณให้เป็นแบบขนานซึ่งมีรูปแบบในการรับ (protocol) เพื่อป้องกันความผิดพลาดในการรับ กลไกที่วันนี้แบ่งเป็น 2 แบบคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การรับส่งแบบอะซิงโครนัส (Asynchronous protocol)
- การรับส่งแบบซิงโครนัส (Synchronous protocol)

2.1 การรับส่งแบบอะซิงโครนัส (Asynchronous protocol)

การส่งข้อมูลแบบนี้ข้อมูลที่ส่งออกไปจะมีลักษณะเป็นบล็อกๆ ซึ่งในแต่ละบล็อกจะประกอบด้วย บิตเริ่มต้น (start bit) ส่วนของข้อมูล และบิตสิ้นสุดของข้อมูล (stop bit) โดยบิตเริ่มต้นจะแสดงถึงการมาหรือการเริ่มต้นของข้อมูล แล้วตามด้วยกลุ่มของข้อมูล ในบางกรณีอาจจะมีการเพิ่ม บิตพาริตีเข้าไป เพื่อให้ตรวจสอบความถูกต้องของข้อมูล ส่วนในบิตสุดท้ายนั้นจะเป็นการบอกว่าข้อมูลได้สิ้นสุดเพียงเท่านั้น

- บิตเริ่มต้นในโปรโตคอลของการส่งข้อมูลอนุกรมแบบอะซิงโครนัส กำหนดให้สถานะมาร์ค (marking state) เป็นสัญญาณลอจิก 1 เมื่อทางด้านส่งจะทำการส่งข้อมูลก็จะส่งบิตเริ่มต้นหรือสัญญาณลอจิก 0 ทำให้ทางด้านรับสามารถตรวจสอบสถานะของสายส่งได้ว่าในขณะนั้นกำลังจะมีข้อมูลส่งมา ปัญหาอันหนึ่งที่พบบ่อยๆในการส่งสัญญาณข้อมูลคือสัญญาณสไปร์ค (spike) ซึ่งทำให้สถานะลอจิกสั้นเกินไป ปัญหานี้แก้ได้โดยจะมีส่วนของวงจรสไปร์คดีเทคชั่น (spike detection)

- บิตข้อมูล หลังจากด้านรับสามารถดีเทคสัญญาณบิตเริ่มต้นได้แล้ว ก็ทำการเซตสถานะของชิพรีจิสเตอร์ให้พร้อมที่จะสามารถรับข้อมูลได้ โดยบิตข้อมูลจะมีจำนวนบิตเป็น 5, 6, 7 หรือ 8 บิต ขึ้นอยู่กับจำนวนคาร์เรียดเตอร์

- บิตพาริตี จะทำหน้าที่ในการบอกส่วนรับข้อมูลทราบว่า ข้อมูลที่รับเข้ามาผิดหรือไม่ บิตพาริตีนั้นจะถูกส่งออกมาพร้อมกับบิตข้อมูล ซึ่งบิตนี้จะ เป็น 1 หรือ 0 ขึ้นอยู่กับข้อมูลที่ส่งออกมาว่ามีจำนวนบิต 1 เป็นจำนวนคี่หรือคู่ และยังขึ้นอยู่กับอุปกรณ์รับส่งข้อมูลด้วยว่าถูกออกแบบไว้ให้รับส่งบิตพาริตีคู่หรือคี่อีกด้วย

2.2 การรับส่งข้อมูลแบบซิงโครนัส (Synchronous protocol)

ในการรับส่งข้อมูลแบบซิงโครนัสนั้นต่างจากแบบอะซิงโครนัสคือ การส่งแบบนี้จะไม่มีการส่งสัญญาณ start bit หรือ stop bit ออกไปซิงค์กับตัวรับ แต่จะมีการส่งชุดสถานะของสัญญาณที่เรียกว่า sync character โดยจะส่งสัญญาณนี้ออกไปเรื่อยๆจนกว่าฝ่ายรับจะซิงค์กับตัวส่งได้

ดังนั้นจะเห็นได้ชัดว่า การส่งข้อมูลแบบซิงโครนัสจะมีความรวดเร็วกว่าการส่งข้อมูลแบบอะซิงโครนัส แต่อย่างไรก็ตามความผิดพลาดจากการส่งแบบอะซิงโครนัสย่อมต้องดีกว่า ทั้งนี้เนื่องมาจากไม่มีการตรวจสอบพาริตี สิ่งที่สำคัญมากสิ่งหนึ่งในการส่งข้อมูลแบบอนุกรมทั้งแบบซิงโครนัสและอะซิงโครนัส

ซึ่งโครนส์ก็คือความถี่ที่ใช้ในการส่งข้อมูล ซึ่งจะต้องสัมพันธ์กันระหว่างอุปกรณ์ที่ทำการรับและส่งข้อมูล ความถี่นี้เราเรียกว่า baud rate หรืออาจกล่าวได้ว่า

baud rate คือ "อัตราการรับส่งข้อมูลเป็นบิตต่อวินาที" ถ้าหากเครื่องส่งใช้ baud rate ไม่สัมพันธ์กับเครื่องรับแล้ว ก็จะทำให้การรับส่งข้อมูลเกิดการผิดพลาดขึ้นได้

การติดต่อแบบอนุกรมอาจจะแบ่งตามรูปลักษณะได้ 3 แบบ ตามรูป 3.4

1.แบบซิมเพล็กซ์ (simplex) ข้อมูลส่งได้ในทางเดียวเท่านั้น บางครั้งก็เรียกว่า การส่งทิศทางเดียว (Unidirectional data bus)

2.แบบฮาล์ฟดูเพล็กซ์ (half duplex) ข้อมูลสามารถส่งได้ทั้ง 2 สถานี แต่จะต้องผลัดกันส่งและผลัดกันรับ จะส่งและรับพร้อมกันไม่ได้

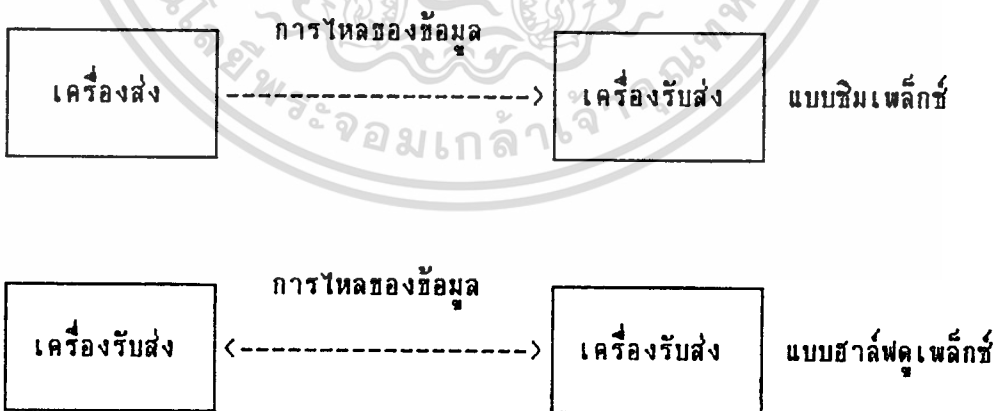
3.แบบฟูลดูเพล็กซ์ (full duplex) ทั้ง 2 สถานีสามารถรับและส่งในเวลาเดียวกัน

ตัวอย่างของระบบที่ใช้การติดต่อในลักษณะดังกล่าวนี้ได้แก่

ระบบซิมเพล็กซ์ : การส่งกระจายเสียงวิทยุหรือโทรทัศน์ หรือ การส่งข้อมูลจากคอมพิวเตอร์ไปให้เครื่องพิมพ์

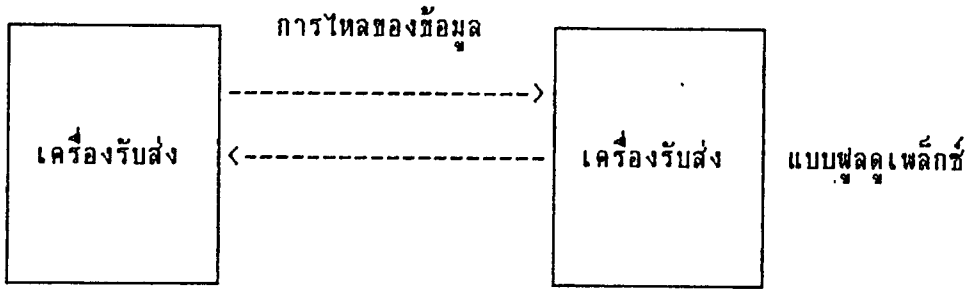
ระบบฮาล์ฟดูเพล็กซ์ : ระบบวิทยุติดต่อ, ระบบ ATM

ระบบฟูลดูเพล็กซ์ : การสื่อสารทางโทรศัพท์ และการสื่อสารคอมพิวเตอร์



รูป 3.4 รูปแบบการติดต่อสื่อสารข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.4 (ต่อ) รูปแบบการติดต่อสื่อสารข้อมูลแบบอนุกรม

MCS-51 กับการติดต่อสื่อสารทางพอร์ทอนุกรม

MCS-51 เป็นชิปเกิดชิพที่ค่อนข้างจะสมบูรณ์ ซึ่งภายในตัว MCS-51 ประกอบด้วยไทเมอร์ ตัวนับ, ออสซิลเลเตอร์, พอร์ท I/O, พอร์ทอนุกรม จากรูปที่ 3.5 เป็นบล็อกการทำงานของ MCS-51 โดยแสดงส่วนที่เกี่ยวข้อง และจัดการกับการสื่อสารข้อมูลทางพอร์ทอนุกรม โดยใช้พอร์ทอนุกรม โดยใช้พอร์ท 3 บิต P3.0 และ P3.1 เป็นส่วนติดต่อข้อมูลโดยที่ P3.0 จะทำหน้าที่รับข้อมูลเข้ามาเรียกว่าขา RXD ส่วนในการส่งข้อมูล MCS-51 จะส่งออกที่ขา P3.1 เรียกว่าขา TXD

1. รีจิสเตอร์ที่เกี่ยวข้อง

การใช้งาน MCS-51 ในด้านสื่อสารข้อมูลทางพอร์ทอนุกรม จะมีรีจิสเตอร์ที่เกี่ยวข้องหลายตัวคือ

ก) PCON : Power Mode Control Register

| | | | | | | | |
|------|---|---|---|-----|-----|----|-----|
| SMOD | - | - | - | GF1 | GFO | PD | IDL |
|------|---|---|---|-----|-----|----|-----|

รูปที่ 3.5 แสดงบิตต่างๆของรีจิสเตอร์ PCON

รีจิสเตอร์ตัวนี้จะใช้งานเพียงบิตเดียวคือ SMOD ถ้า SMOD เป็น "0" ค่า K เป็น 1 SMOD เป็น "1" ค่า K จะเป็น 2

ข) IE : (Interrupt Enable Register)

| | | | | | | | |
|----|---|-----|----|-----|-----|-----|-----|
| EA | - | ET2 | ES | ET1 | EX1 | ETO | EXO |
|----|---|-----|----|-----|-----|-----|-----|

รูปที่ 3.6 แสดงบิตต่างๆของรีจิสเตอร์ IE

รีจิสเตอร์ตัวนี้เป็นตัวที่ใช้กำหนดความสำคัญของการอินเตอร์รัพท์ ในกรณีที่ต้องการให้พอร์ทอนุกรมมีการอินเตอร์รัพท์ขณะทำการรับส่งข้อมูลให้ใช้คำสั่ง EST EA และ SET ES แล้วนำเอาโปรแกรมการรับส่งไปไว้ที่ตำแหน่ง RI&TI (0023H)

ค) IP : (Interrupt Priority Register)

| | | | | | | | |
|---|---|-----|-----|-----|-----|----|----|
| - | - | SM2 | REN | TB8 | RB8 | TI | RI |
|---|---|-----|-----|-----|-----|----|----|

รูปที่ 3.7 แสดงบิตต่างๆของรีจิสเตอร์ IP

ในกรณีที่ต้องการให้พอร์ทอนุกรมมีการอินเตอร์รัพท์ ขณะที่รับส่งข้อมูลรีจิสเตอร์ตัวนี้เป็นตัวที่ใช้กำหนดความสำคัญของการอินเตอร์รัพท์ ถ้าบิตใดเป็น "1" ลักษณะการอินเตอร์รัพท์ในบิตนั้นจะมีความสำคัญสูงสุด คำสั่งที่ใช้ก็ เช่น SETB IP

ง) TCON : (Timer Control Register)

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| TF1 | TR1 | TFO | TRO | IE1 | IT1 | IE0 | IT0 |
|-----|-----|-----|-----|-----|-----|-----|-----|

รูปที่ 3.8 แสดงบิตต่างๆของรีจิสเตอร์ TCON

รีจิสเตอร์นี้มีบิตที่ต้องใช้คือ TRO และ TR1 เป็นตัวควบคุมไทเมอร์และเคาน์เตอร์โดยถ้าให้เป็น "0" จะหมายถึง OFF และถ้าให้เป็น "1" จะหมายถึง ON ฉะนั้นหลังจากเรื่กค่าต่างๆหมดแล้ว เราจะเปิดก็ใช้คำสั่ง SETB TRO หรือ SETB TR1 ถ้าต้องการปิดก็ใช้คำสั่ง CLR TR1 หรือ CLR TR1

จ) TMOD : (Timer Control Register)

| | | | | | | | |
|------|-----|----|----|------|-----|----|----|
| GATE | C/T | M1 | MO | GATE | C/T | M1 | MO |
|------|-----|----|----|------|-----|----|----|

รูปที่ 3.9 แสดงบิตต่างๆของรีจิสเตอร์ TMOD

ในการใช้งานพอร์ทอนุกรมจะใช้งานรีจิสเตอร์นี้บางบิต คือ

- GATE ถ้า GATE เป็น "1" TRx (TRO หรือ TR1 ก็ได้) เป็น "1" ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อ INTx เป็น "1" เท่านั้น และถ้า GATE เป็น "1" เท่านั้น นั่นคือถ้าให้ GATE ของไทเมอร์ 1 เป็น "1" และ TR1 ใน TCON เป็น "1" จะทำงานเมื่อ INT1 เป็น "1" เท่านั้น ไทเมอร์ 0 ก็เช่นเดียวกัน

- C/T ถ้า C/T เป็น "1" จะเป็นโหมดเคาน์เตอร์ โดยจะนับสัญญาณจากขา Tx ถ้า C/T เป็น "0" จะเป็นโหมดไทเมอร์โดยจะเอาสัญญาณนาฬิกาจากภายในชิพขึ้นมาตั้งเวลา

- M1, MO เป็นตัวเลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์ โดยมีรายละเอียดดังตารางที่ 3.1 จะใช้ไทเมอร์ 1 ในการสร้างบอดเรต โดยใช้ไทเมอร์โหมด 2 โดยต้องกำหนด

ค่าการทำงานดังนี้

GATE - "0"

C/T - "0" (โหมดไทเมอร์)

MO - "0"

M1 - "0" (8บิต ออตรีโหลด)

ตารางที่ 3.1 การเลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์โดยใช้ MO, M1

| MO | M1 | โหมด | คำอธิบาย |
|----|----|------|--------------------------------|
| 0 | 0 | 0 | 13บิตไทเมอร์ |
| 0 | 1 | 1 | 13บิตไทเมอร์ |
| 1 | 0 | 2 | 8บิตออตรีโหลด |
| 1 | 1 | 3 | ไทเมอร์/เคาน์เตอร์ 8 บิต 2 ชุด |

a) SCON : (Serial Port Control Register)

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|----|----|
| SM0 | SM1 | SM2 | REN | TB8 | RB8 | T1 | R1 |
|-----|-----|-----|-----|-----|-----|----|----|

รูปที่ 3.10 แสดงบิตต่างๆของรีจิสเตอร์ SCON

การเลือกโหมดการทำงานของ SCON จะต้องให้การควบคุมจาก MO, M1 ดังตารางที่ 3.2 บิตต่างๆที่เกี่ยวข้องกับการสื่อสารทางพอร์ทอนุกรมในรีจิสเตอร์ SCON ได้แก่

- SM2 ถ้า SM2 เป็น "1" ในโหมด 2 หรือ 3 RI จะไม่เป็น "1" ถ้าข้อมูลบิตที่ 9 เป็น "0" ส่วนในโหมด 1 ถ้า SM2 เป็น "1" RI จะไม่เป็น "1" ถ้ายังไม่ได้สตอปบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- REN ถ้าบิตนี้เป็น "0" จะส่งได้ทางเดียว ถ้าเป็น "1" จะทำงานได้ทั้งรับส่ง
- TB8/TR8 ถ้าใช้งานในโหมด 2 หรือ 3 RB8 จะเป็นข้อมูลที่รับบิตที่ 8 และ TB8 จะเป็นข้อมูลที่ต้องการส่งออกบิตที่ 8
- TI จะเป็น "1" เมื่อข้อมูลใน SBUF ได้ส่งไปแล้ว
- RI จะเป็น "1" เมื่อได้รับข้อมูลครบทุกบิตหรือพบสตอบบิตแล้ว

ตารางที่ 3.2 การเลือกโหมดการทำงานบอดเรตของ SCON โดย MO และ M1

| MO | M1 | โหมด | คำอธิบาย | บอดเรต |
|----|----|------|---------------|-------------------------------|
| 0 | 0 | 0 | ซีพริจิสเตอร์ | ความถี่สัญญาณนาฬิกา/2 |
| 0 | 1 | 1 | USART 8 บิต | เลือกค่าได้ |
| 1 | 0 | 2 | USART 9 บิต | ความถี่สัญญาณนาฬิกา/2 หรือ 64 |
| 1 | 1 | 3 | USART 9 บิต | เลือกค่าได้ |

* USART : Universal Asynchronous Receiver Transceiver

การคำนวณบอดเรต

โหมด 1

$$\text{บอดเรต} = 2^{\text{SMOD}} / 32 * 12 * [(256 - (\text{TH1}))]$$

โหมด 0

$$\text{บอดเรต} = \text{ความถี่สัญญาณนาฬิกา} / 12$$

$$\text{TH1} = 256 - [(2^{\text{SMOD}} * \text{ความถี่สัญญาณนาฬิกา} / 384 * \text{บอดเรต})]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 2

บ็อดเรต - $[2^{\text{SMOD}} * \text{ความถี่สัญญาณนาฬิกา}] / 64$

ถ้าใช้ไทมเมอร์ 1 เป็นตัวกำเนิดบ็อดเรต ในการคำนวณจะเหมือนกับกรณีของ 8031, 8051 8751 แต่ถ้าใช้ไทมเมอร์ 2 การคำนวณจะคำนวณได้ดังนี้

บ็อดเรต - $\text{ความถี่สัญญาณนาฬิกา} / 32 * [65536 - (RCAP2H, RCAP2L)]$

$RCAP2H, RCAP2L - 65536 - \text{ความถี่ของสัญญาณนาฬิกา} / 32 * \text{บ็อดเรต}$

โปรแกรมการจัดการข้อมูล

โปรแกรมในการจัดการข้อมูลแบ่งออกเป็น 2 ส่วนคือ

1. โปรแกรมเกี่ยวกับการจัดเก็บข้อมูล
2. โปรแกรมจัดการเกี่ยวกับ Database

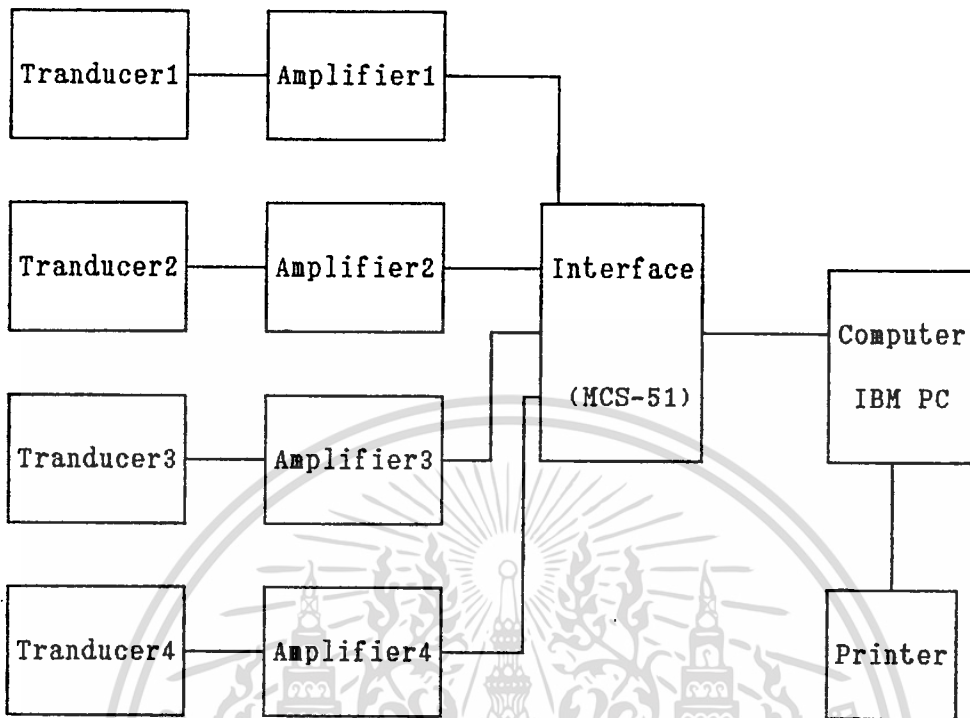
1. โปรแกรมเกี่ยวกับการจัดเก็บข้อมูล

โปรแกรมนี้เป็นการจัดการรวบรวมข้อมูลต่าง ๆ ทั้งหมด มาเก็บบันทึกไว้ในหน่วยความจำภายในเครื่องคอมพิวเตอร์ IBM PC ซึ่งจะใช้โปรแกรมภาษาซีเป็นโปรแกรมจัดเก็บข้อมูลดิบที่ได้ต่าง ๆ ที่เกี่ยวข้อง โดยข้อมูลที่จัดเก็บจะประกอบไปด้วย

1. ชื่อของ Operator ที่ควบคุมการทำงานหน้าจอคอมพิวเตอร์
2. วันที่ที่จัดเก็บข้อมูลในแต่ละวัน
3. จำนวนความถี่ที่เทียบออกมา เพื่อหาปริมาณน้ำมันที่จ่ายไปจริงของน้ำมันแต่ละชนิด
4. ราคาของน้ำมันต่อลิตรที่จ่ายไปทั้งหมดในแต่ละวัน
5. ปริมาณน้ำมันที่จ่ายไปทั้งหมดในแต่ละวัน
6. ราคาน้ำมันรวมของน้ำมันแต่ละชนิดในแต่ละวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกไดอะแกรมแสดงการส่งข้อมูลมาที่เครื่องคอมพิวเตอร์



การรับข้อมูล

การรับข้อมูลจะใช้พอร์ตอนุกรม (Serial port) เป็นอุปกรณ์ที่ใช้สำหรับรับส่งข้อมูล โดยสายสัญญาณที่ใช้ในการรับส่งข้อมูลจะใช้ 3 เส้นคือ GRD, TxD และ RxD ซึ่งการควบคุมการทำงานของพอร์ตอนุกรม โดยการเรียกใช้ผ่าน BIOS อินเทอร์เน็ตหมายเลข 14 อัตราการส่งข้อมูลมีหน่วยเป็น baud (Bit per second) ค่า baud rate ที่ใช้ในเครื่องคอมพิวเตอร์ระดับ IBM PC ที่ใช้จะใช้ค่า baud rate ที่ 9600 baud

พอร์ตแบบอนุกรมส่วนใหญ่ จะมีรูปร่างขึ้นอยู่กับมาตรฐานของ RS-232 คือ มี 25 ขาที่คอนเน็กเตอร์แต่ละปลายสายส่ง (แต่ IBM AT จะมีเพียง 9 ขา) แต่ถึงแม้ว่าจะมีขาจำนวนเท่ากัน แต่พอร์ตส่วนใหญ่จะมีสัญญาณที่ไม่เหมือนสัญญาณของ RS-232 ทั้งหมด เพราะว่าบางสัญญาณไม่จำเป็นต้องใช้หรือบริษัทผู้ผลิตบางบริษัทตั้งใจจะตัดสัญญาณบางสัญญาณออกไปเพื่อจะลดค่าใช้จ่าย สัญญาณพื้นฐานของ RS-232 แสดงไว้ดังตารางข้างล่าง

| สัญญาณ | ชื่อย่อ | หมายเลขขา |
|---------------------|---------|-----------|
| Request to send | RTS | 4 |
| Clear to send | CTS | 5 |
| Data set ready | DSR | 6 |
| Data terminal ready | DTR | 20 |
| Transmit data | TxD | 2 |
| Receive data | RxD | 3 |
| Ground | GRD | 7 |

2. โปรแกรมเกี่ยวกับ Database

เมื่อเราได้ทำการจัดเก็บข้อมูลที่ได้เก็บลงในหน่วยความจำของเครื่องคอมพิวเตอร์แล้วขั้นต่อไปก็คือการเก็บข้อมูลที่ได้นี้มารวบรวมอย่างเป็นระบบในรูปของ Database โดยใช้โปรแกรม FoxPro ซึ่งเป็นโปรแกรมระบบฐานข้อมูลที่มีประสิทธิภาพสูงที่ใช้กับเครื่องคอมพิวเตอร์ส่วนบุคคล IBM PC และคอมแพคทีเบิ้ลทั่วไป ซึ่งสามารถใช้ฟอซโปรในการสร้างแฟ้มข้อมูลที่ประกอบด้วยฟิลด์ข้อมูลต่าง ๆ และแสดงผลออกมาในรูปแบบที่สะดวกในการทำแบบฟอร์มและรายงาน

บทที่ 4

การจำลองการวัดอัตราไหลและการทดลองโครงการ

4.1 การจำลองการวัดอัตราไหล

เนื่องจากเครื่องมือการวัดอัตราไหลไม่ว่าจะเป็นการวัดการไหลโดยใช้ Turbine flow meter หรือ Magnetic flow meter เอ้าท์พุทที่ได้จากเครื่องมือวัดดังกล่าวจะเป็นสัญญาณพัลส์ ทั้งสิ้น ดังนั้นการจำลองสัญญาณเอ้าท์พุทที่ได้จากเครื่องมือเหล่านั้นจึงเป็นสิ่งที่จำเป็น ซึ่งการที่จะจำลองสิ่งเหล่านั้นได้ เราจำเป็นต้องรู้คุณสมบัติของเครื่องมือการวัดอัตราไหลเหล่านั้นก่อน ซึ่งในที่นี่จะขอกล่าวเฉพาะ Turbine flow meter ดังนี้คือ

1. สัญญาณเอ้าท์พุทที่ได้เป็น square pulse
2. range ของความถี่อยู่ระหว่าง 3-10 Hz
3. ขนาดของแรงดันเป็นสัดส่วนกับอัตราการไหล ซึ่งโดยปกติจะอยู่ในช่วง 10-1000 mV

rms

นอกจากนี้ Capacities range ของ Turbine flow meter ก็ยังเป็นสิ่งจำเป็นในการที่จะสร้างชุดจำลองการวัดอัตราไหลให้เป็นไปโดยถูกต้องซึ่งแสดงได้ในตารางที่ 4.1

TABLE 4.1-TYPICAL CAPACITIES FOR A RANGE OF TURBINE FLOW METER FOR BOTH LIQUIDS AND GASES

| | | LIQUIDS | | | |
|------|-------|----------------------|---------|----------------------|---------|
| Bore | | Flow range(standard) | | Flow range(extended) | |
| mm | in | m ³ /hr | lit/min | m ³ /hr | lit/min |
| 12 | 0.5 | 0.09to0.85 | 1.5to14 | 0.05to1.3 | 0.8to22 |
| 16 | 0.625 | 0.22to2.2 | 3.6to36 | 0.11to2.7 | 1.3to45 |
| 16 | 0.625 | 0.4to4 | 6.8to68 | 0.3to4.8 | 5to80 |
| 19 | 0.75 | 0.85to8.5 | 14to140 | 0.5to9.5 | 8to155 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | LIQUIDS | | | |
|------|------|----------------------|-----------|----------------------|----------|
| Bore | | Flow range(standard) | | Flow range(extended) | |
| mm | in | m ³ /hr | lit/min | m ³ /hr | lit/min |
| 25 | 1.0 | 1.6to16 | 27to270 | 0.8to18 | 13to300 |
| 32 | 1.25 | 2.7to27 | 45to450 | 1.9to33 | 32to550 |
| 38 | 1.5 | 4to40 | 68to680 | 2.2to46 | 36to770 |
| 50 | 2.0 | 6.6to66 | 110to1100 | 4to84 | 68to1400 |

| | | GASES(actual flows) | | | |
|------|-------|----------------------|-----------|----------------------|------------|
| Bore | | Flow range(standard) | | Flow range(extended) | |
| mm | in | m ³ /hr | lit/min | m ³ /hr | lit/min |
| 12 | 0.5 | 0.09to0.85 | 0.05to0.5 | 0.09to1.11 | 0.05to0.65 |
| 16 | 0.625 | 0.22to2.2 | 0.13to1.3 | 0.22to2.9 | 0.13to1.7 |
| 16 | 0.625 | 0.4to4 | 0.23to2.3 | 0.4to5.4 | 0.23to3.2 |
| 19 | 0.75 | 0.85to8.5 | 0.5to5 | 0.85to11 | 0.5to6.5 |
| 25 | 1.0 | 1.6to16 | 1to10 | 1.6to22 | 1to13 |
| 32 | 1.25 | 2.7to27 | 1.6to16 | 2.7to35 | 1.6to21 |
| 38 | 1.5 | 4to40 | 2.3to23 | 4to100* | 2.3to58* |
| 50 | 2.0 | 8.5to85 | 5to50 | 8.5to250* | 5to150* |

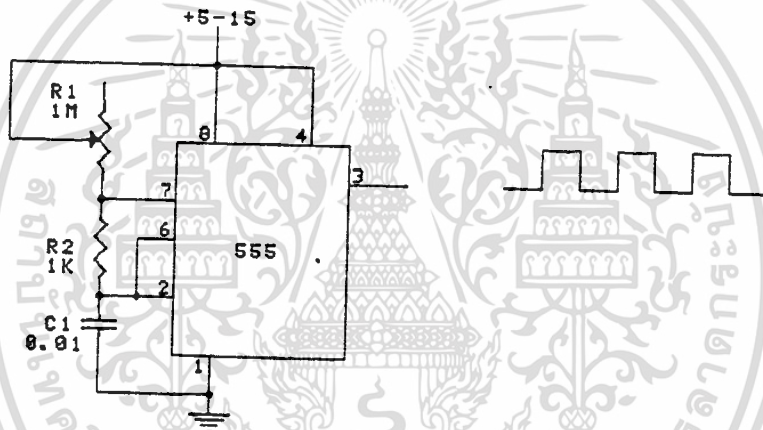
The above flow range are based on water calibrations. Where high viscos

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

city fluids are metered, the linear range is decreased. Gas flow ranges are *actual* flows for pipelines with pressure of 3 bar and above.

ดังนั้นในการสร้างชุดจำลองการวัดอัตราไหลต้องสอดคล้องกับเหตุผลดังกล่าวข้างต้นเพื่อคงไว้ซึ่งความถูกต้อง วงจรการกำเนิดพัลส์นี้นับว่าเป็นประโยชน์มากในการทดลองหรือการทำงานเกี่ยวกับวงจรลอจิกต่างๆวงจรนี้ใช้ฐานเวลาจาก IC ตั้งเวลาเบอร์ 555 ซึ่งทำให้ปรับความกว้างของพัลส์และความถี่ของพัลส์ได้โดยง่าย

วงจรการกำเนิดพัลส์แสดงดังรูปที่ 4.1



รูปที่ 4.1 วงจรกำเนิดพัลส์

วงจรนี้ใช้ฐานเวลาจาก IC เบอร์ 555 โดยมีตัวความต้านทาน R1, R2 และตัวเก็บประจุ C1 ซึ่งสามารถควบคุมช่วงเวลาของพัลส์ และสามารถควบคุมอย่างละเอียดได้โดยการปรับค่า R1

4.2 การทดสอบโครงการ

4.2.1 อุปกรณ์ที่ใช้

- 1) Personal Computer 1 เครื่อง
- 2) Microcontroller 1 เครื่อง
- 3) Pulse Simulate 1 ชุด
- 4) Counter Console 1 ชุด
- 5) Printer 1 เครื่อง
- 6) สายสัญญาณระหว่าง PC กับ Microcontroller

4.2.2 วิธีการทดลอง

- 1) ประกอบชุด Microcontroller และชุดของ PC ดังรูป 4.2
- 2) ทำการ Run โปรแกรมภาษาซี
- 3) ทำการเลือก Channel ที่ Counter console ซึ่งมีทั้งหมด 8 Channel โดยถ้าป้อน 10 เข้าไปจะเป็น Channel 1, ป้อน 20 เข้าไปก็จะเป็น Channel 2, ป้อน 30 เข้าไปก็จะเป็น Channel 3 จนกระทั่งป้อน 80 เข้าไปก็เป็นการเลือก Channel 8 ตามลำดับ ซึ่ง Channel 1 หมายความว่า ในปั้มน้ำมันมีการใช้หัวจ่ายน้ำมันเพียงหัวเดียวและ Channel 2 จะหมายถึง ในปั้มน้ำมันมีการใช้หัวจ่ายน้ำมันอยู่ 2 หัวจ่ายจนถึง Channel 8 ซึ่งมีการใช้หัวจ่ายน้ำมันถึง 8 หัวจ่าย
- 4) ทำการจ่ายน้ำมันในแต่ละหัวจ่าย ซึ่งการจ่ายน้ำมันในแต่ละหัวจ่ายน้ำมันนั้น ได้จำลอง Pulse Simulate ขึ้นมาแทน เพราะสัญญาณที่ได้จากหัวจ่ายน้ำมันในปั้มน้ำมันนั้น มีสัญญาณเข้าที่พบเป็น Pulse ดังได้กล่าวแล้วข้างต้น
- 5) สังเกตการทดลองที่หน้าจอ PC ดังรูปที่ 4.3 ถึง 4.14
- 6) สังเกตผลการทดลองที่ Printer ดังรูปที่ 4.13
- 7) เปรียบเทียบค่าที่ได้จากการทดลองและการคำนวณ

4.2.3 ผลการทดลอง

- รูปที่ 4.3 แสดง Main Menu ที่หน้าจอ PC
- รูปที่ 4.4 ถึง 4.6 แสดงการเลือกใช้ Menu 1
- รูปที่ 4.7 แสดงการใช้ Menu 2
- รูปที่ 4.8 แสดงการใช้ Menu 3

- รูปที่ 4.9 แสดงการใช้ Menu 1 อีกครั้งหลังจากทำการ save ด้วย Menu 3
- รูปที่ 4.10 ถึง 4.11 แสดงการใช้ Menu 4
- รูปที่ 4.12 แสดงการใช้ Menu 5 ซึ่งรูปนี้แสดงผลที่หน้าจอ PC
- รูปที่ 4.13 แสดงการใช้ Menu 5 ซึ่งรูปนี้แสดงผลที่ Printer
- รูปที่ 4.14 แสดงการใช้ Menu 6
- รูปที่ 4.15 ถึง 4.18 แสดงผลที่ได้จากการเปรียบเทียบระหว่างค่าที่วัดได้กับค่า

ที่คำนวณ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

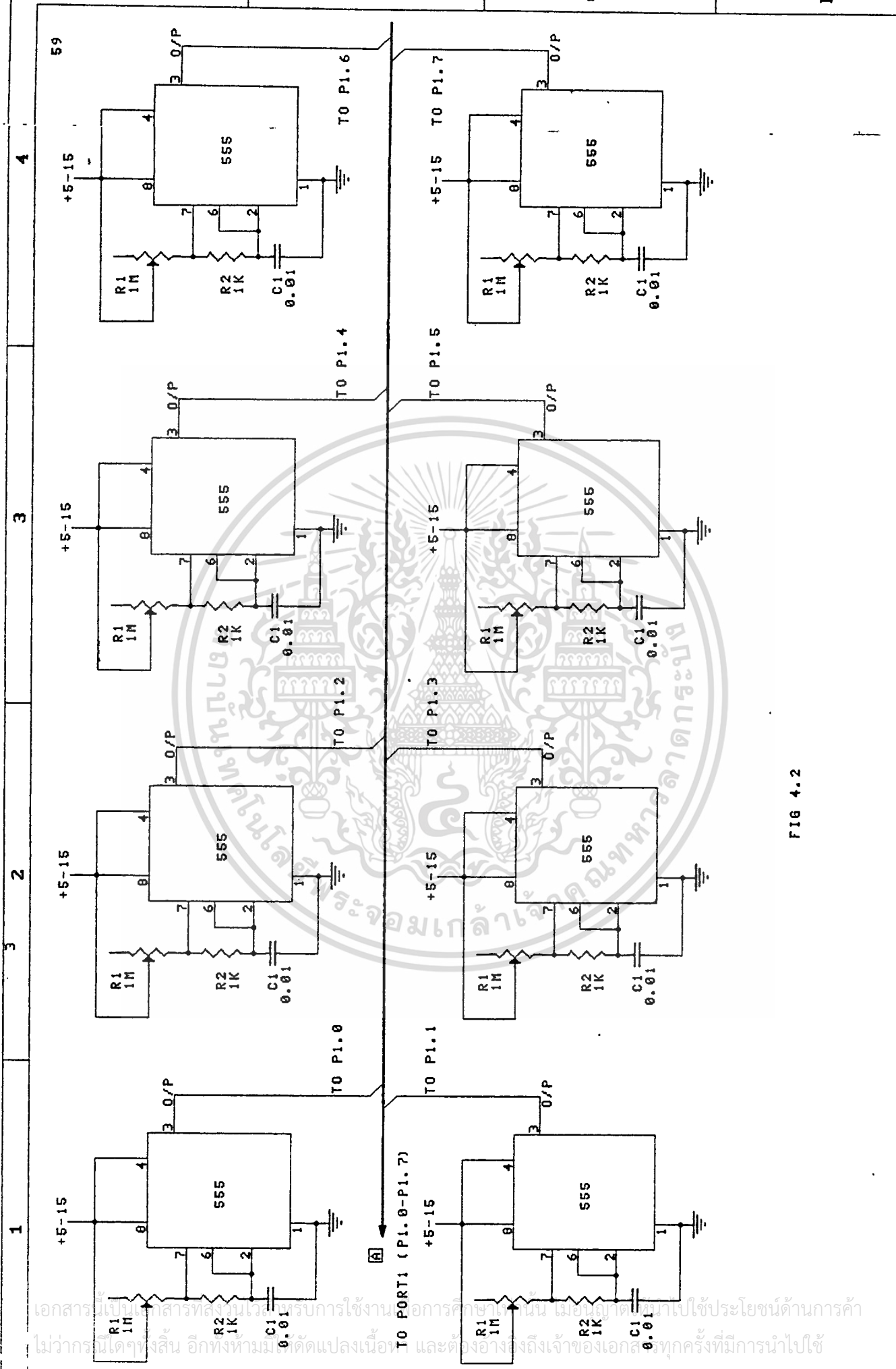


FIG 4.2

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาหรือการให้บริการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อถูกคัดลอกไปใช้โดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีการดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

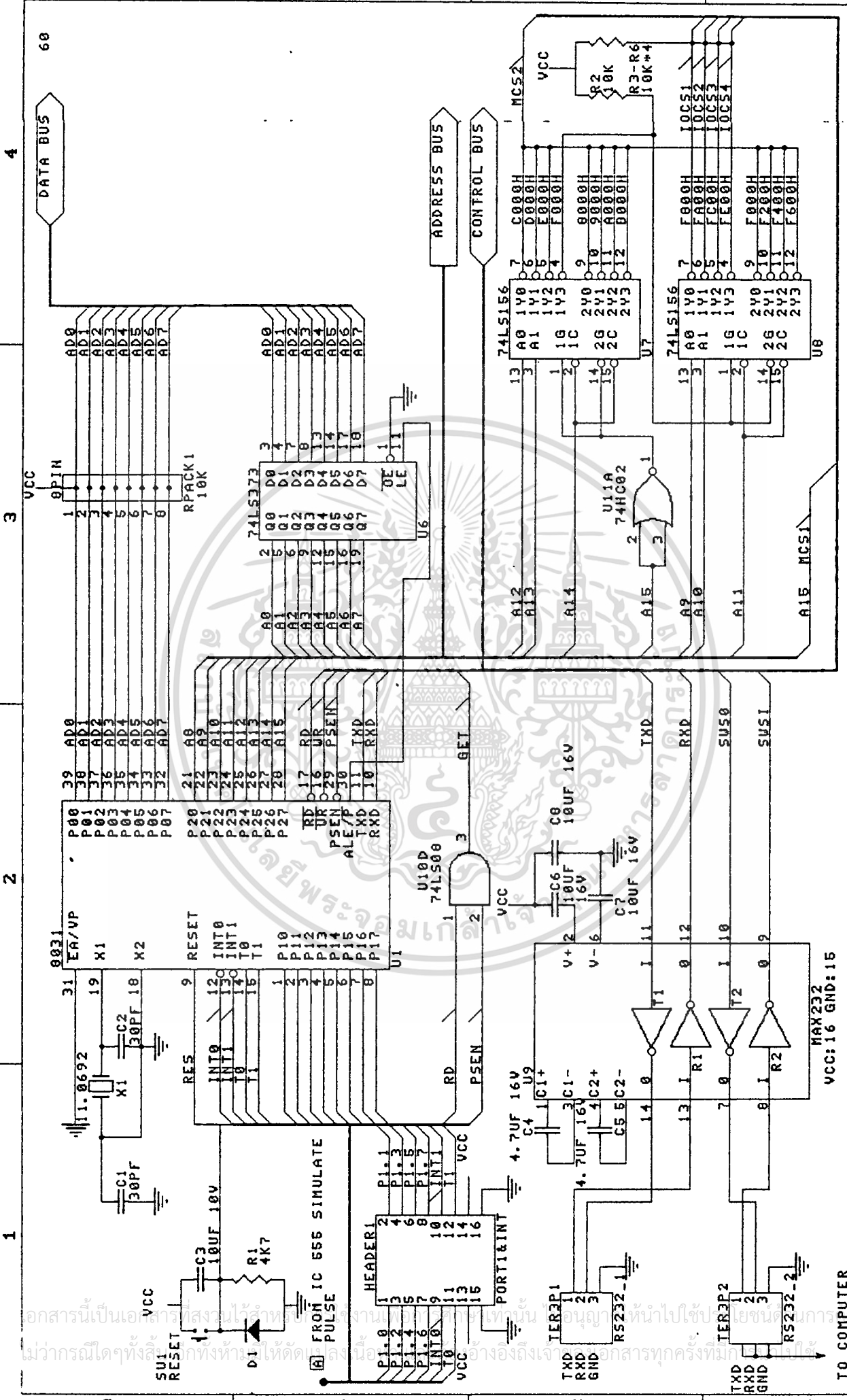


FIG 4.2 (CONT.)

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ห้ามการนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตจากทางสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 หากมีการนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตจากทางสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 จะถือว่าผิดกฎหมายและทางสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ขอสงวนสิทธิ์ใน
 การฟ้องดำเนินคดีตามกฎหมายที่เกี่ยวข้อง

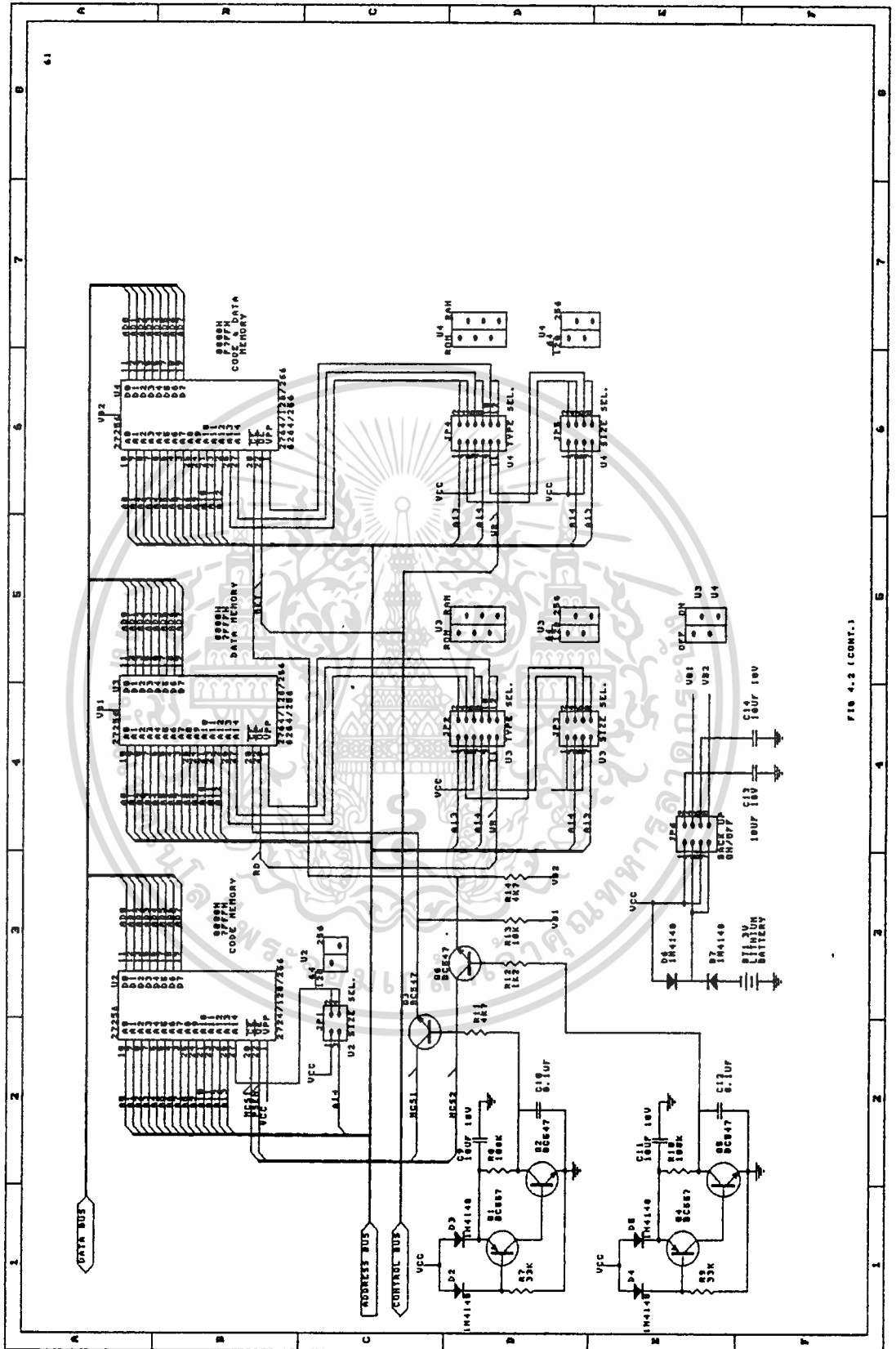


FIG 4-3 (CONT.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 SEGMENT LED DISPLAY * 3

LED1--16

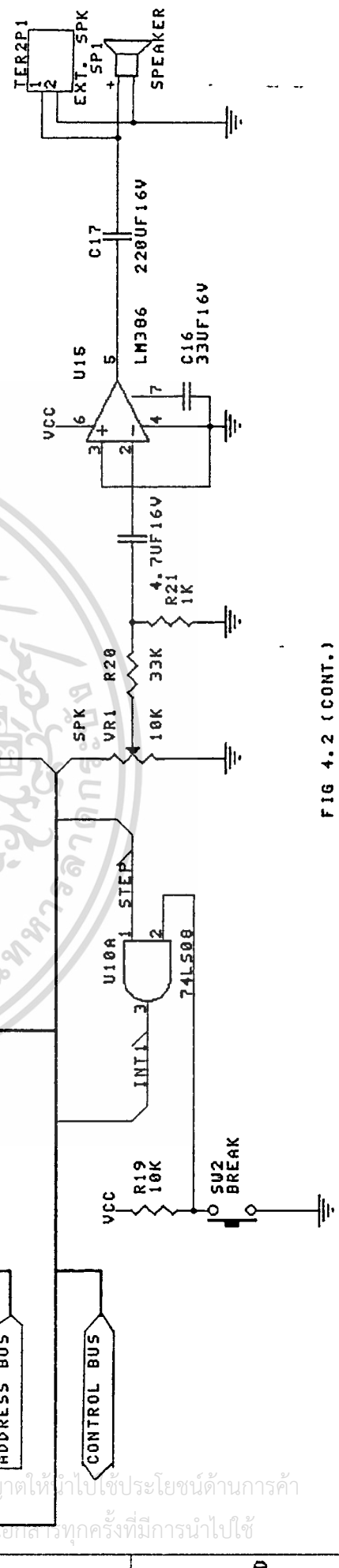
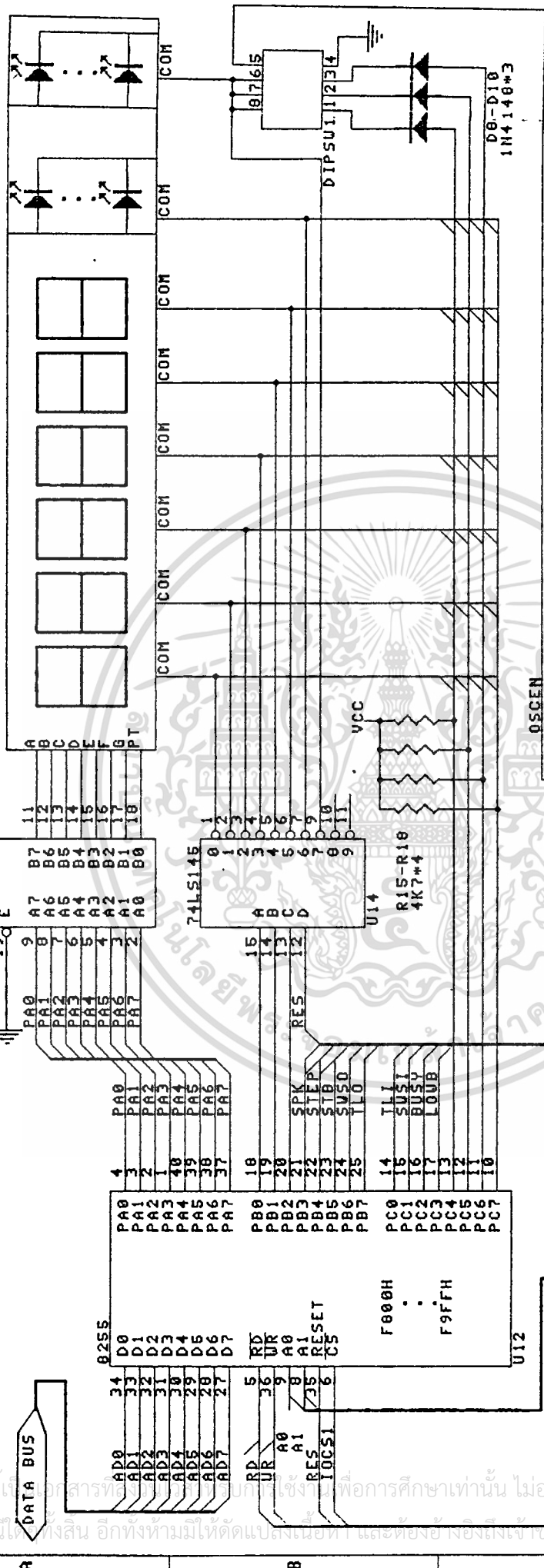
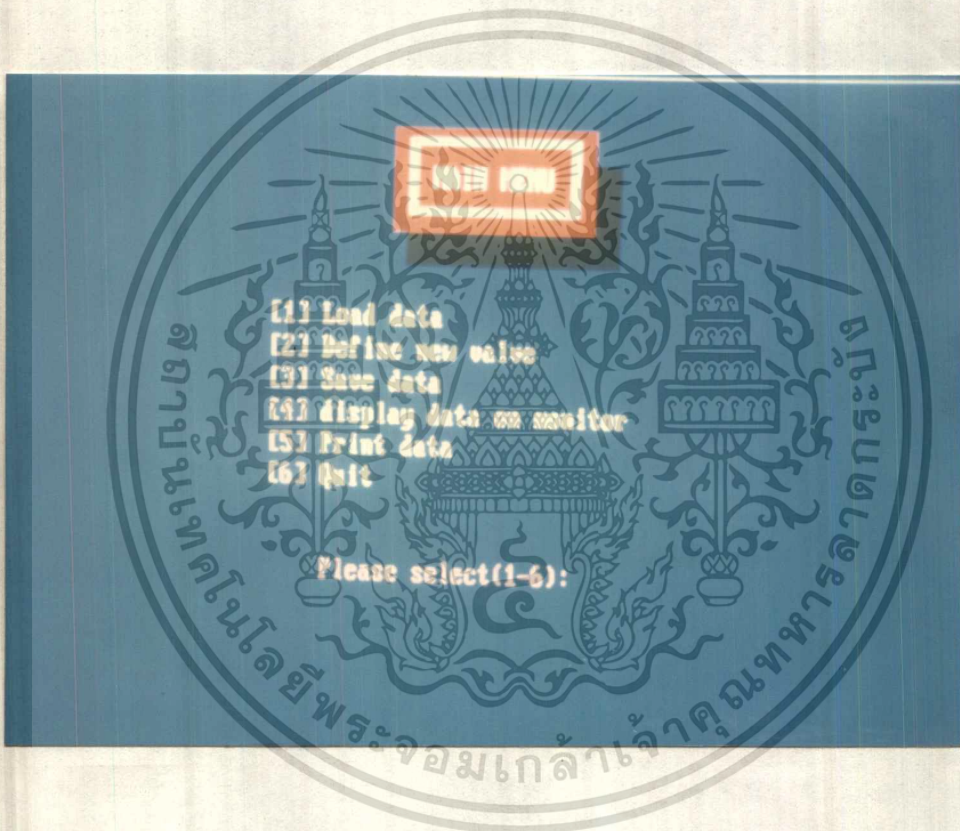


FIG 4.2 (CONT.)



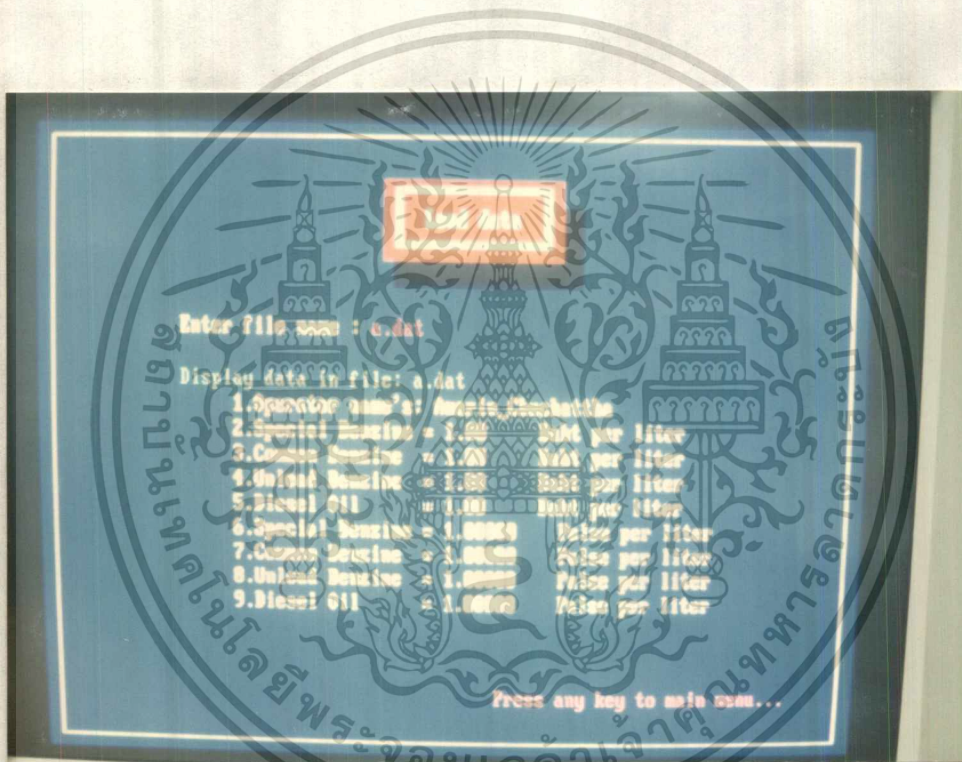
รูปที่ 4.3 แสดง Main Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



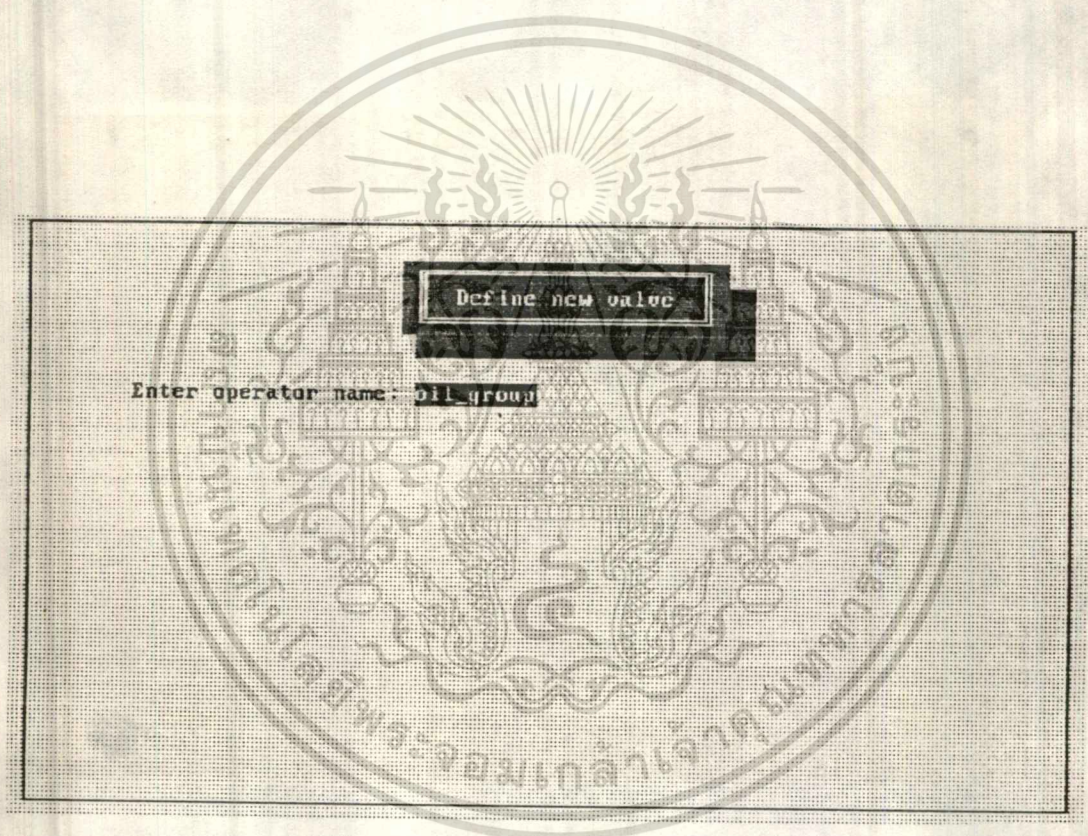
รูปที่ 4.4 แสดงการเลือกใช้ Menu 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



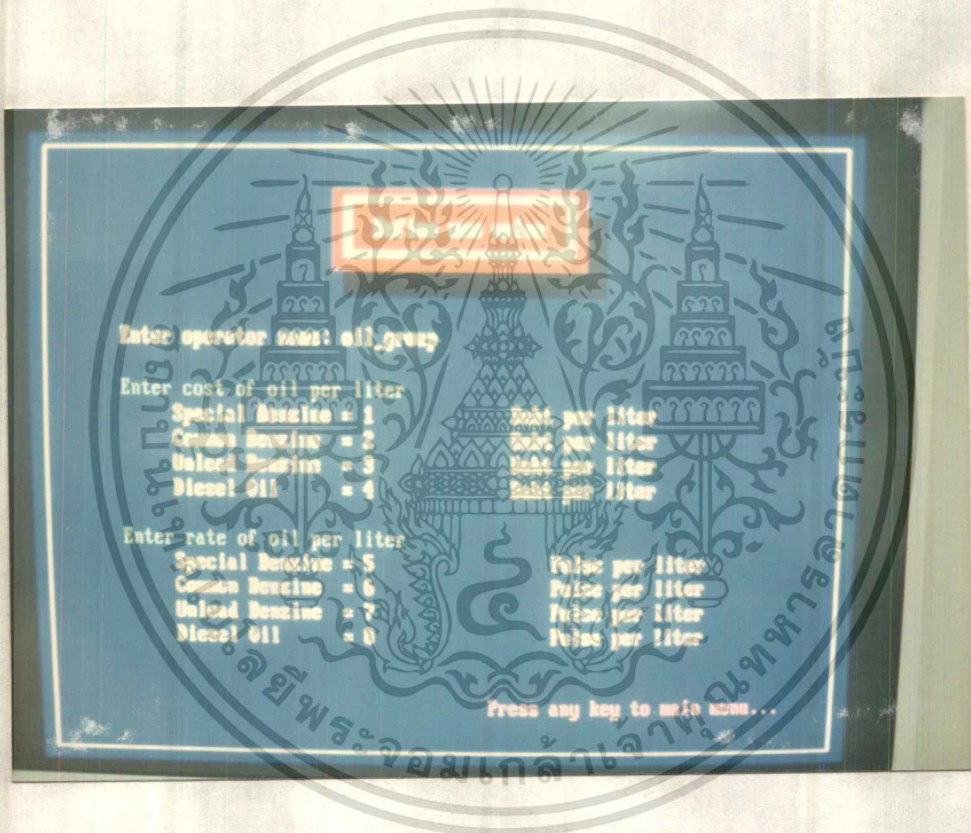
รูปที่ 4.5 แสดงการเลือกใช้ Menu 1 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงการเลือกใช้ Menu 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



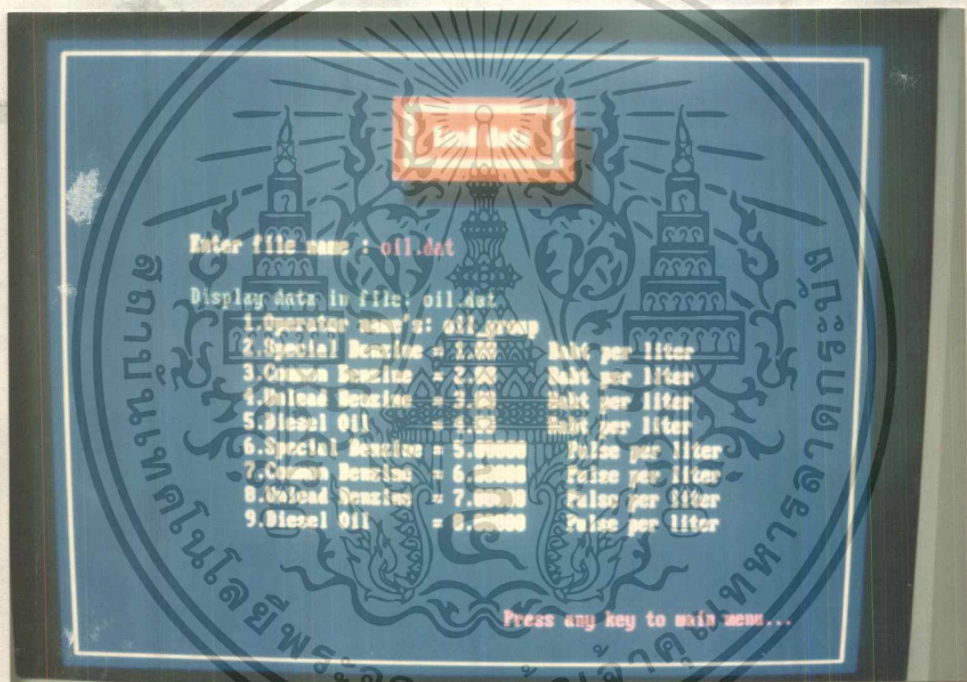
รูปที่ 4.7 แสดงการเลือกใช้ Menu 2 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



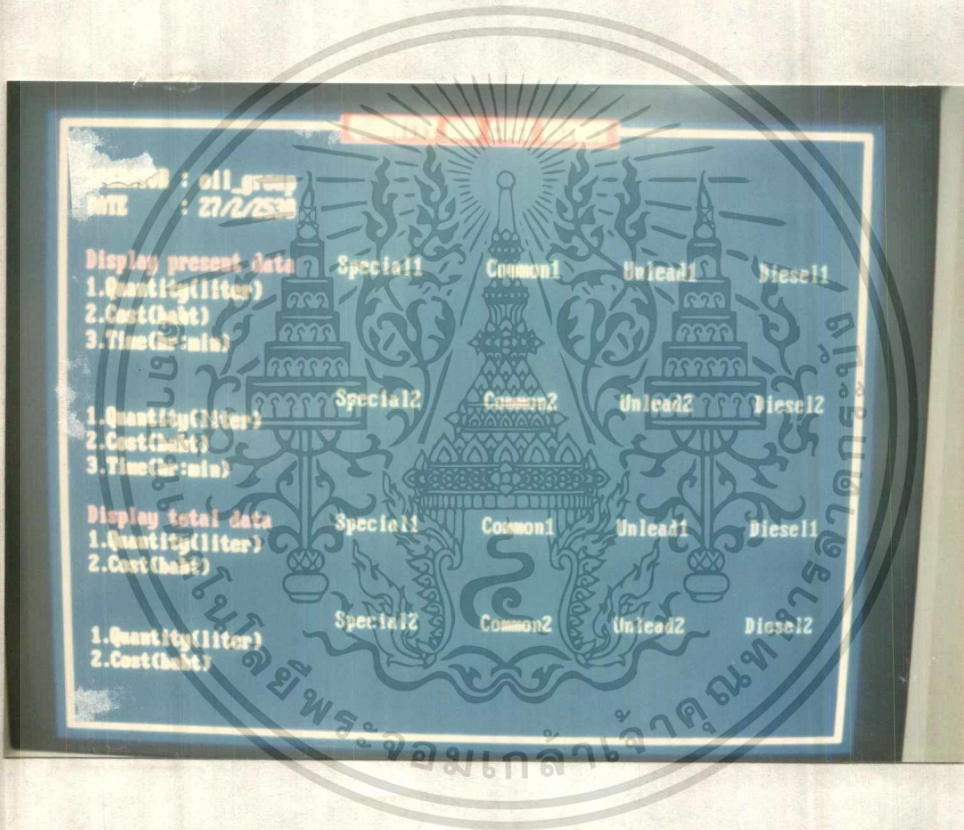
รูปที่ 4.8 แสดงการใช้ Menu 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงการใช้ Menu 1 อีกครั้งหลังจาก save ด้วย Menu 3 แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงการใช้ Menu 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QUANTITY AND COST REPORT

OPERATOR : oil_group
DATE : 27/2/2538

Display present data

| | Special1 | Common1 | Unlead1 | Diesel1 |
|-------------------|----------|---------|---------|---------|
| 1.Quantity(liter) | 7.20 | 31.88 | 268.86 | 9.58 |
| 2.Cost(baht) | 7.20 | 62.88 | 886.52 | 38.00 |
| 3.Time(hr:min) | 17:51 | 17:51 | 17:51 | 17:51 |

| | Special2 | Common2 | Unlead2 | Diesel2 |
|-------------------|----------|---------|---------|---------|
| 1.Quantity(liter) | 1.80 | 12.17 | 8.71 | 13.25 |
| 2.Cost(baht) | 1.80 | 24.33 | 26.14 | 53.00 |
| 3.Time(hr:min) | 17:51 | 17:51 | 17:51 | 17:51 |

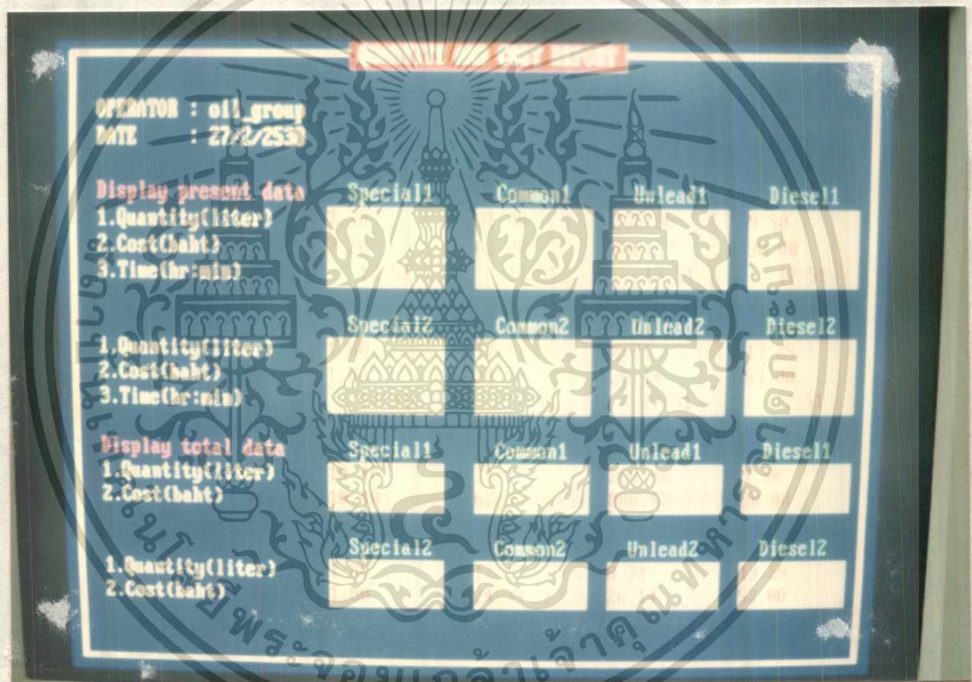
Display total data

| | Special1 | Common1 | Unlead1 | Diesel1 |
|-------------------|----------|---------|---------|---------|
| 1.Quantity(liter) | 18.48 | 31.17 | 268.86 | 9.58 |
| 2.Cost(baht) | 18.48 | 62.33 | 886.57 | 38.00 |

| | Special2 | Common2 | Unlead2 | Diesel2 |
|-------------------|----------|---------|---------|---------|
| 1.Quantity(liter) | 1.80 | 12.17 | 8.71 | 13.25 |
| 2.Cost(baht) | 1.80 | 24.33 | 26.14 | 53.00 |

รูปที่ 4.11 แสดงการใช้ Menu 4 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงการใช้ Menu 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Type:Special Benzine(1)
 Date:25/2/2538
 Time:19:17
 Quantity:167.00 liter
 Cost:167.00 Baht

Type:Special Benzine(2)
 Date:25/2/2538
 Time:19:17
 Quantity:118.00 liter
 Cost:118.00 Baht

Type:Common Benzine(1)
 Date:25/2/2538
 Time:19:17
 Quantity:84.00 liter
 Cost:84.00 Baht

Type:Common Benzine(2)
 Date:25/2/2538
 Time:19:17
 Quantity:39.00 liter
 Cost:39.00 Baht

Type:Unlead Benzine(1)
 Date:25/2/2538
 Time:19:17
 Quantity:281.00 liter
 Cost:281.00 Baht

Type:Unlead Benzine(2)
 Date:25/2/2538
 Time:19:17
 Quantity:376.00 liter
 Cost:376.00 Baht

Type:Diesel Oil(1)
 Date:25/2/2538
 Time:19:17
 Quantity:197.00 liter
 Cost:197.00 Baht

Type:Diesel Oil(2)
 Date:25/2/2538
 Time:19:17
 Quantity:157.00 liter
 Cost:157.00 Baht

รูปที่ 4.13 แสดงการใช้ Menu 5 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงการใช้ Menu 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบค่าที่วัดได้จากการวัดในการทดลองกับค่าที่เกิดจากการคำนวณ
ที่หัวจ่ายน้ำมันหัวที่ 1 CH-1

| FREQ. | TIME(min) | PULSE/LITRE | QUANTITY/LITRE | | ERROR (%) |
|-------|-----------|-------------|----------------|-------|--------------|
| | | | MEAS. | CALC. | |
| 1 | 1 | 20 | 2.98 | 3 | -0.66 |
| 2 | 1 | 20 | 5.99 | 6 | -0.16 |
| 3 | 1 | 20 | 9.02 | 9 | 0.22 |
| 4 | 1 | 20 | 12.03 | 12 | 0.25 |
| 5 | 1 | 20 | 15.06 | 15 | 0.40 |
| 6 | 1 | 20 | 17.97 | 18 | -0.16 |
| 7 | 1 | 20 | 20.95 | 21 | -0.24 |
| 8 | 1 | 20 | 24.08 | 24 | 0.33 |
| 9 | 1 | 20 | 26.95 | 27 | -0.18 |
| 10 | 1 | 20 | 29.90 | 30 | -0.33 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบค่าที่วัดได้จากการวัดในการทดลองกับค่าที่เกิดจากการคำนวณ
ที่หัวจ่ายน้ำมันหัวที่ 1 CH-4

| FREQ. | TIME(min) | PULSE/LITRE | QUANTITY/LITRE | | ERROR (%) |
|-------|-----------|-------------|----------------|-------|--------------|
| | | | MEAS. | CALC. | |
| 1 | 1 | 20 | 2.96 | 3 | -1.33 |
| 2 | 1 | 20 | 5.91 | 6 | -1.50 |
| 3 | 1 | 20 | 9.07 | 9 | 0.77 |
| 4 | 1 | 20 | 12.02 | 12 | 0.16 |
| 5 | 1 | 20 | 15.03 | 15 | 0.20 |
| 6 | 1 | 20 | 17.94 | 18 | -0.33 |
| 7 | 1 | 20 | 20.96 | 21 | -0.19 |
| 8 | 1 | 20 | 24.06 | 24 | 0.25 |
| 9 | 1 | 20 | 26.94 | 27 | -0.22 |
| 10 | 1 | 20 | 29.93 | 30 | -0.23 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบค่าที่วัดได้จากการวัดในการทดลองกับค่าที่เกิดจากการคำนวณ
ที่หัวจ่ายน้ำมันหัวที่ 1 CH-6

| FREQ. | TIME(min) | PULSE/LITRE | QUANTITY/LITRE | | ERROR (%) |
|-------|-----------|-------------|----------------|-------|--------------|
| | | | MEAS. | CALC. | |
| 1 | 1 | 20 | 2.96 | 3 | -1.33 |
| 2 | 1 | 20 | 5.90 | 6 | -1.66 |
| 3 | 1 | 20 | 9.10 | 9 | 1.11 |
| 4 | 1 | 20 | 12.09 | 12 | 0.75 |
| 5 | 1 | 20 | 14.89 | 15 | -0.73 |
| 6 | 1 | 20 | 17.88 | 18 | -0.66 |
| 7 | 1 | 20 | 20.80 | 21 | -0.95 |
| 8 | 1 | 20 | 24.22 | 24 | 0.91 |
| 9 | 1 | 20 | 26.76 | 27 | -0.88 |
| 10 | 1 | 20 | 29.73 | 30 | -0.90 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบค่าที่วัดได้จากการวัดในการทดลองกับค่าที่เกิดจากการคำนวณ
ที่หัวจ่ายน้ำมันหัวที่ 1 CH-8

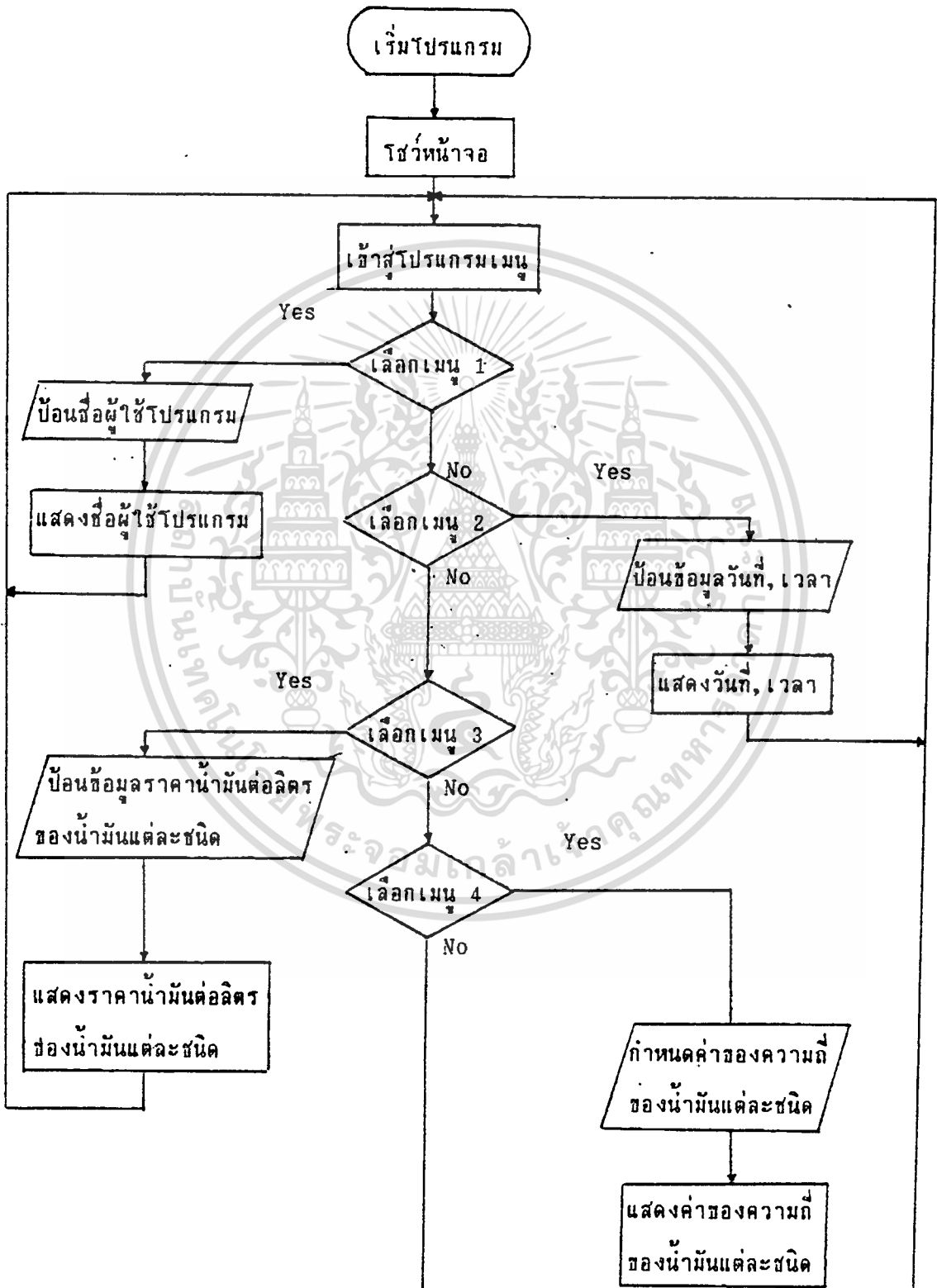
| FREQ. | TIME(min) | PULSE/LITRE | QUANTITY/LITRE | | ERROR (%) |
|-------|-----------|-------------|----------------|-------|-----------|
| | | | MEAS. | CALC. | |
| 1 | 1 | 20 | 2.94 | 3 | -2.00 |
| 2 | 1 | 20 | 5.89 | 6 | -1.83 |
| 3 | 1 | 20 | 9.10 | 9 | 1.11 |
| 4 | 1 | 20 | 12.17 | 12 | 1.41 |
| 5 | 1 | 20 | 14.82 | 15 | -1.20 |
| 6 | 1 | 20 | 17.81 | 18 | -1.05 |
| 7 | 1 | 20 | 20.75 | 21 | -1.19 |
| 8 | 1 | 20 | 24.32 | 24 | 1.33 |
| 9 | 1 | 20 | 26.67 | 27 | -1.22 |
| 10 | 1 | 20 | 29.66 | 30 | -1.13 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

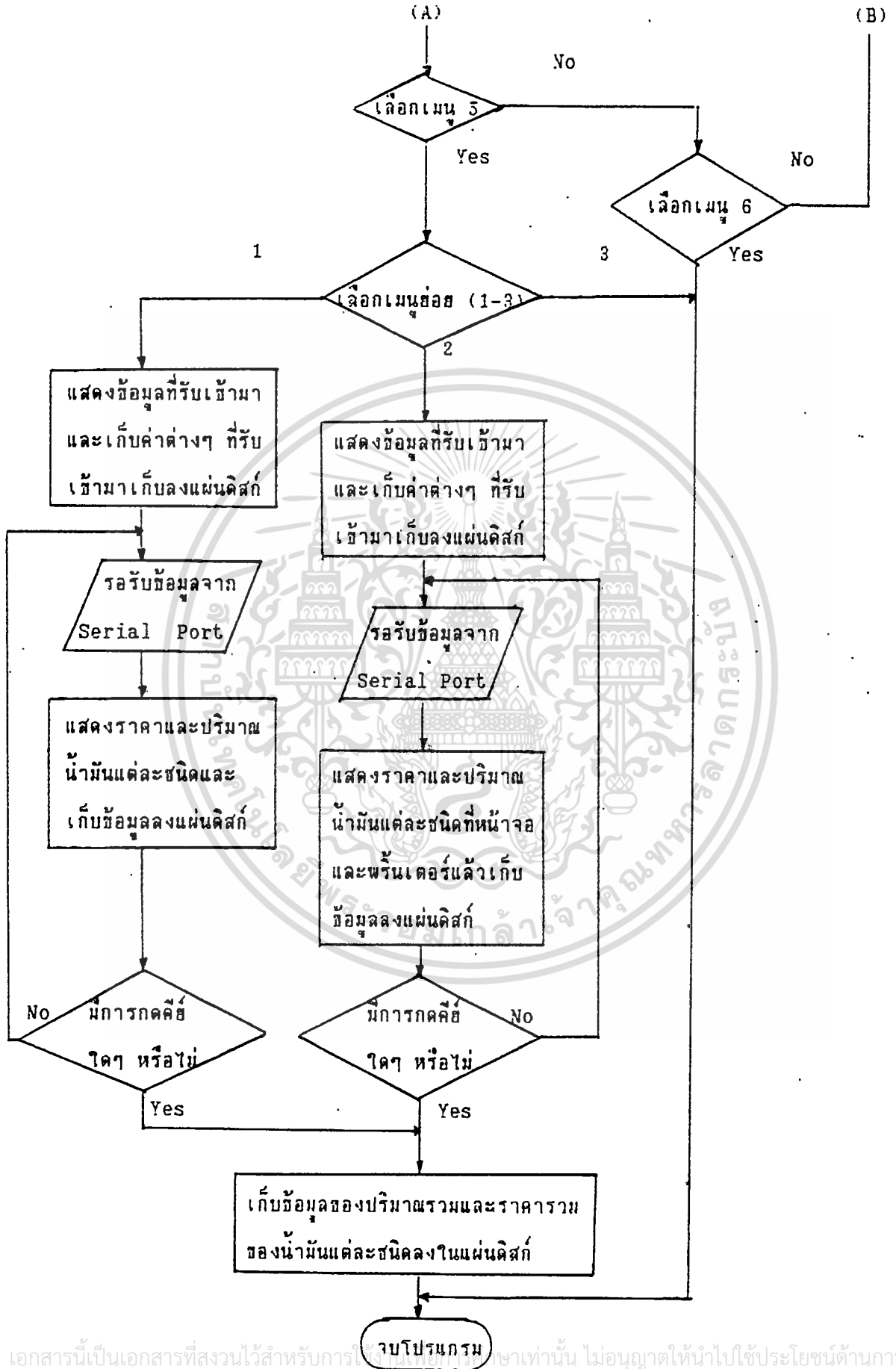
บทที่ 5

โปรแกรมประกอบภาวทดลอง

5.1 Turbo-C Flowchart



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ (A) (B)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 C Program

```

/* ===== X.C ===== */
/* Oil quantity measurement in oil pump with microcomputer
Version 1.0
Subject Project2
Name of adviser
    Mr.Vittaya    Tipsuvanporn
Name of membership
    1. Mr.Ammarin    Chuchattha
    2. Mr.Kittipong    Malitong
    3. Mr.Saingam    Homkachon
Department
    Industrial instrumentation technology
Copyright 1994
All rights reserved
*/

#ifdef_TINY_
#error Graphics demos will not run in the tiny model.
#endif

/* define position of serial port */
#define PORT 0

#include<dos.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<stdarg.h>
#include<ctype.h>
#include<string.h>
#include<math.h>
#include<time.h>
#include<graphics.h>
#include<gprint.i>

```

```

/* data of show1 & show2 */
int GraphDriver;
int GraphMode;
int MaxColors;
int MaxX,MaxY;
int ErrorCode=0;
struct palettetype palette;

/* data of create graph */
void *grimage[3];
int Accounts[4][20]={119,121,132,140,141,139,142,135,133,123,
    121,120,124,111,109,119,122,132,140,142,
    97,99,100,107,119,123,137,148,159,160,
    168,172,167,155,159,163,165,155,151,148,
    59,73,66,49,40,39,41,45,46,52,
    56,59,60,56,51,54,55,53,72,75,
    13,15,16,19,22,20,17,18,19,21,
    24,26,28,27,22,20,19,23,25,24
};
int Years[5]={1990,1991,1992,1993,1994};
char *AccTypes[4]={"Special Benzine",
    "Common Benzine",
    "Unlead Benzine",
    "Diesel Oil"};

int rport(int port), check_stat(int port);

/* make construction of program */
char op[40];
float cs;
float cc;
float cu;
float cd;
double ps;
double pc;
double pu;

```

```

double pd;
double qs1,qs2;
double qc1,qc2;
double qu1,qu2;
double qd1,qd2;
double css1,css2;
double ccc1,ccc2;
double cuu1,cuu2;
double cdd1,cdd2;
double tqs1,tqs2;
double tqc1,tqc2;
double tqu1,tqu2;
double tqd1,tqd2;
double tcs1,tcs2;
double tcc1,tcc2;
double tcu1,tcu2;
double tcd1,tcd2;
struct time now1;
struct time now2;
struct time now3;
struct time now4;
struct time now5;
struct time now6;
struct time now7;
struct time now8;
struct date today;

```

```

/* Data transfer

```

```

    Port initialized to
        9600 baud,
        no parity,
        eight data bits,
        two stop bits.

```

```

*/

```

```

/* start program */
atexit_t main()
{
char choice;

/*showscreen();*/
textmode(3);
textbackground(BLUE);
for(;;)
{
    choice=menu();
    switch(choice)
    {
case'1':load();break;
case'2':define();break;
case'3':save();break;
case'4':display();break;
case'5':print();break;
case'6':quit();break;
    }
}
}

/*get a menu selection*/
menu()
{
char s[35];
cursor_off();
do
{
    clrscr();
    box1(2,1,79,25);
    textbackground(BLACK);
    gotoxy(34,4);cprintf("                ");
    gotoxy(34,5);cprintf("                ");
    gotoxy(34,6);cprintf("                ");

```

```

textbackground(RED);
textcolor(WHITE);
gotoxy(33,3);cprintf("                ");
gotoxy(33,4);cprintf("    MAIN MENU    ");
gotoxy(33,5);cprintf("                ");
        box2(34,3,46,5);
textbackground(BLUE);
textcolor(YELLOW);
gotoxy(24,8);
cprintf("[1] Load data");
gotoxy(24,9);
cprintf("[2] Define new valve");
gotoxy(24,10);
cprintf("[3] Save data");
gotoxy(24,11);
cprintf("[4] display data on monitor");
gotoxy(24,12);
cprintf("[5] Print data");
gotoxy(24,13);
cprintf("[6] Quit");
textmode(3);
textbackground(BLUE);
textcolor(LIGHTMAGENTA);
gotoxy(28,16);
cprintf("Please select(1-6): ");
gets(s);
}
while(!strchr("123456",tolower(*s)));
return tolower(*s);
}

/* load file name */
load()
{
FILE *fp;
char name[80];

```

```

textmode(3);
    sound(800);
delay(100);
nosound();
textbackground(LIGHTGRAY);
textcolor(RED);
gotoxy(24,8);
cprintf("[1] Load data");
textbackground(BLUE);
delay(1000);
clrscr();
textbackground(BLACK);
gotoxy(35,4);cprintf(" ");
gotoxy(35,5);cprintf(" ");
gotoxy(35,6);cprintf(" ");
textbackground(RED);
textcolor(LIGHTCYAN);
gotoxy(34,3);cprintf(" ");
gotoxy(34,4);cprintf(" Load .data ");
gotoxy(34,5);cprintf(" ");
box2(35,3,49,5);

textbackground(BLUE);
textcolor(YELLOW);
cursor_on();
box1(2,1,79,25);
gotoxy(15,8);
cprintf("Enter file name : ");
gets(name);
cursor_off();
if((fp=fopen(name,"rb"))==NULL)
{
    clrscr();
    box1(2,1,79,25);
    textcolor(WHITE);
    gotoxy(30,12);

```

```

    printf("Cannot open file!");
    gotoxy(33,23);
    printf("Press any key... ");
    getch();
    atexit(main());
}
while(fscanf(fp,"%s %f %f %f %f %lf %lf %lf %lf",
op,&cs,&cc,&cu,&cd,&ps,&pc,&pu,&pd)!=EOF)
{
    textbackground(BLUE);
    textcolor(LIGHTCYAN);
    gotoxy(15,10);
    printf("Display data in file: %s",name);
    textbackground(BLUE);
    textcolor(WHITE);
    gotoxy(20,11);
    printf("1.Operator name's: %s",op);
    gotoxy(20,12);
    printf("2.Special Benzine = %.2f      Baht per liter",cs);
    gotoxy(20,13);
    printf("3.Common Benzine   = %.2f      Baht per liter",cc);
    gotoxy(20,14);
    printf("4.Unlead Benzine    = %.2f      Baht per liter",cu);
    gotoxy(20,15);
    printf("5.Diesel Oil        = %.2f      Baht per liter",cd);
    gotoxy(20,16);
    printf("6.Special Benzine = %.5lf      Pulse per liter",ps);
    gotoxy(20,17);
    printf("7.Common Benzine  = %.5lf      Pulse per liter",pc);
    gotoxy(20,18);
    printf("8.Unlead Benzine  = %.5lf      Pulse per liter",pu);
    gotoxy(20,19);
    printf("9.Diesel Oil      = %.5lf      Pulse per liter",pd);
    textcolor(LIGHTRED);
    gotoxy(45,23);
    printf("Press any key to main menu...");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    getch();
}
fclose(fp);
    textbackground(BLUE);
textcolor(WHITE);
}

/* determine data for calculate */
define()
{
    sound(800);
delay(100);
nosound();
textbackground(LIGHTGRAY);
textcolor(RED);
gotoxy(24,9);
cprintf("[2] Define new valve");
delay(1000);
    textbackground(BLUE);
textcolor(YELLOW);
clrscr();
    textbackground(BLACK);
gotoxy(31,4);cprintf(" ");
gotoxy(31,5);cprintf(" ");
gotoxy(31,6);cprintf(" ");
textbackground(RED);
textcolor(LIGHTCYAN);
gotoxy(30,3);cprintf(" ");
gotoxy(30,4);cprintf("    Define new valve    ");
gotoxy(30,5);cprintf(" ");
box2(31,3,52,5);
textbackground(BLUE);
textcolor(YELLOW);
cursor_on();
box1(2,1,79,25);
gotoxy(10,8);

```

```

cprintf("Enter operator name: ");
gets(op);
textcolor(LIGHTCYAN);
gotoxy(10,10);
cprintf("Enter cost of oil per liter");
textcolor(YELLOW);
gotoxy(15,11);
cprintf("Special Benzine =           Baht per liter");
gotoxy(33,11);
scanf("%f",&cs);
gotoxy(15,12);
cprintf("Common Benzine =           Baht per liter");
gotoxy(33,12);
scanf("%f",&cc);
gotoxy(15,13);
cprintf("Unlead Benzine =           Baht per liter");
gotoxy(33,13);
scanf("%f",&cu);
gotoxy(15,14);
cprintf("Diesel Oil =           Baht per liter");
gotoxy(33,14);
scanf("%f",&cd);
gotoxy(10,16);
textcolor(LIGHTCYAN);
cprintf("Enter rate of oil per liter");
textcolor(YELLOW);
gotoxy(15,17);
cprintf("Special Benzine =           Pulse per liter");
gotoxy(33,17);
scanf("%lf",&ps);
gotoxy(15,18);
cprintf("Common Benzine =           Pulse per liter");
gotoxy(33,18);
scanf("%lf",&pc);
gotoxy(15,19);
cprintf("Unlead Benzine =           Pulse per liter");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(3,19);
scanf("%lf",&pu);
gotoxy(15,20);
printf("Diesel Oil      =                Pulse per liter");
gotoxy(33,20);
scanf("%lf",&pd);
cursor_off();
textcolor(LIGHTRED);
gotoxy(45,23);
printf("Press any key to main menu...");
getch();
        textbackground(BLUE);
textcolor(WHITE);

}

/* save data that user input from keyboard to disk */
save()
{
FILE *fp;
char s[80];

sound(800);
delay(100);
nosound();
textbackground(LIGHTGRAY);
textcolor(RED);
gotoxy(24,10);
printf("[3] Save data");
delay(1000);
        textbackground(BLUE);
textcolor(YELLOW);

clrscr();
cursor_on();
box1(2,1,79,25);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(25,12);
cprintf("Enter file name: ");
gets(s);
cursor_off();
if((fp=fopen(s,"wb"))!=NULL)
{
gotoxy(29,14);
cprintf("Cannot open file!");
}
fprintf(fp,"%s %f %f %f %f %lf %lf %lf %lf",
op,cs,cc,cu,cd,ps,pc,pu,pd);
fclose(fp);
}

/* display data on monitor */
display()
{
int Done=0;
double num1,num2,num3,num4,
      num5,num6,num7,num8,
      v1,v2,v3,v4,v5,v6,v7,v8;
char ch;
clock_t start1,start2,start3,start4,
start5,start6,start7,start8,
end1,end2,end3,end4,
end5,end6,end7,end8;
float diff1,diff2,diff3,diff4,
      diff5,diff6,diff7,diff8;
char z[]="          ";
char s1[]="Special1",c1[]="Common1",
u1[]="Unlead1",di1[]="Diesel1",
s2[]="Special2",c2[]="Common2",
u2[]="Unlead2",di2[]="Diesel2";

      sound(800);
delay(100);

```

```

nosound();
textbackground(LIGHTGRAY);
textcolor(RED);
gotoxy(24,11);
printf("[4] Display data on monitor");
delay(1000);
    textbackground(BLUE);
textcolor(WHITE);
clrscr();
port_init(PORT,231);
box2(2,1,79,25);
textbackground(RED);
textcolor(WHITE);
gotoxy(29,1);
printf(" QUANTITY AND COST REPORT ");
textbackground(BLUE);
textcolor(YELLOW);
gotoxy(5,3);
printf("OPERATOR : %s",op);
getdate(&today);
gotoxy(5,4);
printf("DATE      : %d/%d/%d",
    today.da_day,today.da_mon,today.da_year+543);
textbackground(LIGHTBLUE);
textcolor(LIGHTRED);
gotoxy(5,6);
printf("Display present data");
gotoxy(5,16);
printf("Display total data");
textbackground(LIGHTBLUE);
textcolor(LIGHTCYAN);
gotoxy(29,6);printf(s1);
gotoxy(43,6);printf(c1);
gotoxy(56,6);printf(u1);
gotoxy(69,6);printf(di1);
gotoxy(29,11);printf(s2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(43,11);cprintf(c2);
gotoxy(56,11);cprintf(u2);
gotoxy(69,11);cprintf(di2);
gotoxy(29,16);cprintf(s1);
gotoxy(43,16);cprintf(c1);
gotoxy(56,16);cprintf(u1);
gotoxy(69,16);cprintf(di1);
gotoxy(29,20);cprintf(s2);
gotoxy(43,20);cprintf(c2);
gotoxy(56,20);cprintf(u2);
gotoxy(69,20);cprintf(di2);
textbackground(BLUE);
textcolor(WHITE);
gotoxy(5,7);
cprintf("1.Quantity(liter)");
gotoxy(5,8);
cprintf("2.Cost(baht)");
gotoxy(5,9);
cprintf("3.Time(hr:min)");
gotoxy(5,12);
cprintf("1.Quantity(liter)");
gotoxy(5,13);
cprintf("2.Cost(baht)");
gotoxy(5,14);
cprintf("3.Time(hr:min)");
gotoxy(5,17);
cprintf("1.Quantity(liter)");
gotoxy(5,18);
cprintf("2.Cost(baht)");
gotoxy(5,21);
cprintf("1.Quantity(liter)");
gotoxy(5,22);
cprintf("2.Cost(baht)");
textbackground(LIGHTGRAY);
textcolor(RED);
/* initial valve */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

qs1=0;qs2=0;      /* initial quantity */
qc1=0;qc2=0;
qu1=0;qu2=0;
qd1=0;qd2=0;
css1=0;css2=0;    /* initial cost */
ccc1=0;ccc2=0;
cuu1=0;cuu2=0;
cdd1=0;cdd2=0;
tqs1=0;tqs2=0;    /* initial total quantity */
tqc1=0;tqc2=0;
tqu1=0;tqu2=0;
tqd1=0;tqd2=0;
tcs1=0;tcs2=0;    /* initial total cost */
tcc1=0;tcc2=0;
tcu1=0;tcu2=0;
ted1=0;ted2=0;
start1=0;start2=0; /* initial first time */
start3=0;start4=0;
start5=0;start6=0;
start7=0;start8=0;
v1=0;v2=0;        /* initial valve of total cost */
v3=0;v4=0;
v5=0;v6=0;
v7=0;v8=0;
num1=0;num2=0;    /* initial valve of pulse */
num3=0;num4=0;
num5=0;num6=0;
num7=0;num8=0;
while(!Done)
{
    ch=rport(PORT);
    switch(ch)
    {
        case 'w' :    /* Special1 */
end1=clock();
v1++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

diff1=(end1-start1)/CLK_TCK;
if(diff1<=5)
    num1++;
else
    num1=0;
qs1=num1/ps;
css1=qs1*cs;
tqs1=(v1-1)/ps;
tcs1=tqs1*cs;
gotoxy(27,7);cprintf(z);
gotoxy(27,7);cprintf("%.2lf",qs1);
gotoxy(27,8);cprintf(z);
gotoxy(27,8);cprintf("%.2lf",css1);
gettime(&now1);
gotoxy(27,9);cprintf("    %02d:%02d    ",
now1.ti_hour,now1.ti_min);
gotoxy(27,17);cprintf(z);
gotoxy(27,17);cprintf("%.2lf",tqs1);
gotoxy(27,18);cprintf(z);
gotoxy(27,18);cprintf("%.2lf",tcs1);
start1=clock();
break;
    case 'v': /* Special2 */
end2=clock();
v2++;
diff2=(end2-start2)/CLK_TCK;
if(diff2<=5)
    num2++;
else
    num2=0;
qs2=num2/ps;
css2=qs2*cs;
tqs2=(v2-1)/ps;
tcs2=tqs2*cs;
gotoxy(27,12);cprintf(z);
gotoxy(27,12);cprintf("%.2lf",qs2);

```

```

gotoxy(27,13);cprintf(z);
gotoxy(27,13);cprintf("%.2lf",css2);
gettime(&now2);
gotoxy(27,14);cprintf("  %02d:%02d  ",
now2.ti_hour,now2.ti_min);
gotoxy(27,21);cprintf(z);
gotoxy(27,21);cprintf("%.2lf",tqs2);
gotoxy(27,22);cprintf(z);
gotoxy(27,22);cprintf("%.2lf",tcs2);
start2=clock();
break;
    case ']' : /* Common1 */
end3=clock();
v3++;
diff3=(end3-start3)/CLK_TCK;
if(diff3<=5)
    num3++;
else
    num3=0;
qc1=num3/pc;
ccc1=qc1*cc;
tqc1=(v3-1)/pc;
tcc1=tqc1*cc;
gotoxy(41,7);cprintf(z);
gotoxy(41,7);cprintf("%.2lf",qc1);
gotoxy(41,8);cprintf(z);
gotoxy(41,8);cprintf("%.2lf",ccc1);
gettime(&now3);
gotoxy(41,9);cprintf("  %02d:%02d  ",
now3.ti_hour,now3.ti_min);
gotoxy(41,17);cprintf(z);
gotoxy(41,17);cprintf("%.2lf",tqc1);
gotoxy(41,18);cprintf(z);
gotoxy(41,18);cprintf("%.2lf",tcc1);
start3=clock();
break;

```

```

        case 'r' : /* Common2 */
end4=clock();
v4++;
diff4=(end4-start4)/CLK_TCK;
if(diff4<=5)
    num4++;
else
    num4=0;
qc2=num4/pc;
ccc2=qc2*cc;
tqc2=(v4-1)/pc;
tcc2=tqc2*cc;
gotoxy(41,12);cprintf(z);
gotoxy(41,12);cprintf("%.2lf",qc2);
gotoxy(41,13);cprintf(z);
gotoxy(41,13);cprintf("%.2lf",ccc2);
gettime(&now4);
gotoxy(41,14);cprintf(" %02d:%02d ",
now4.ti_hour,now4.ti_min);
gotoxy(41,21);cprintf(z);
gotoxy(41,21);cprintf("%.2lf",tqc2);
gotoxy(41,22);cprintf(z);
gotoxy(41,22);cprintf("%.2lf",tcc2);
start4=clock();
break;
        case 'p' : /* Unlead1 */
end5=clock();
v5++;
diff5=(end5-start5)/CLK_TCK;
if(diff5<=5)
    num5++;
else
    num5=0;
qu1=num5/pu;
cuu1=qu1*cu;
tqu1=(v5-1)/pu;

```

```

tcu1=tqu1*cu;
gotoxy(54,7);cprintf(z);
gotoxy(54,7);cprintf("%.2lf",qu1);
gotoxy(54,8);cprintf(z);
gotoxy(54,8);cprintf("%.2lf",cuu1);
gettime(&now5);
gotoxy(54,9);cprintf(" %02d:%02d ",
now5.ti_hour,now5.ti_min);
gotoxy(54,17);cprintf(z);
gotoxy(54,17);cprintf("%.2lf",tqu1);
gotoxy(54,18);cprintf(z);
gotoxy(54,18);cprintf("%.2lf",tcu1);
start5=clock();
break;
    case '๗' : /* Unlead2 */
end6=clock();
v6++;
diff6=(end6-start6)/CLK_TCK;
if(diff6<=5)
    num6++;
else
    num6=0;
qu2=num6/pu;
cuu2=qu2*cu;
tqu2=(v6-1)/pu;
tcu2=tqu2*cu;
gotoxy(54,12);cprintf(z);
gotoxy(54,12);cprintf("%.2lf",qu2);
gotoxy(54,13);cprintf(z);
gotoxy(54,13);cprintf("%.2lf",cuu2);
gettime(&now6);
gotoxy(54,14);cprintf(" %02d:%02d ",
now6.ti_hour,now6.ti_min);
gotoxy(54,21);cprintf(z);
gotoxy(54,21);cprintf("%.2lf",tqu2);
gotoxy(54,22);cprintf(z);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(54,22);printf("%.2lf",tcu2);
start6=clock();
break;
    case ' ' : /* Diesel1 */
end7=clock();
v7++;
diff7=(end7-start7)/CLK_TCK;
if(diff7<=5)
    num7++;
else
    num7=0;
qd1=num7/pd;
cdd1=qd1*cd;
tqd1=(v7-1)/pd;
tcd1=tqd1*cd;
gotoxy(67,7);printf(z);
gotoxy(67,7);printf("%.2lf",qd1);
gotoxy(67,8);printf(z);
gotoxy(67,8);printf("%.2lf",cdd1);
gettime(&now7);
gotoxy(67,9);printf(" %02d:%02d ",
now7.ti_hour,now7.ti_min);
gotoxy(67,17);printf(z);
gotoxy(67,17);printf("%.2lf",tqd1);
gotoxy(67,18);printf(z);
gotoxy(67,18);printf("%.2lf",tcd1);
start7=clock();
break;
    case ' ' : /* Diesel2 */
end8=clock();
v8++;
diff8=(end8-start8)/CLK_TCK;
if(diff8<=5)
    num8++;
else
    num8=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

qd2=num8/pd;
cdd2=qd2*cd;
tqd2=(v8-1)/pd;
tcd2=tqd2*cd;
gotoxy(67,12);cprintf(z);
gotoxy(67,12);cprintf("%.2lf",qd2);
gotoxy(67,13);cprintf(z);
gotoxy(67,13);cprintf("%.2lf",cdd2);
gettime(&now8);
gotoxy(67,14);cprintf("  %02d:%02d  ",
now8.ti_hour,now8.ti_min);
gotoxy(67,21);cprintf(z);
gotoxy(67,21);cprintf("%.2lf",tqd2);
gotoxy(67,22);cprintf(z);
gotoxy(67,22);cprintf("%.2lf",tcd2);
start8=clock();
break;

    default:break;
}
if(toupper(ch)=='Q') Done ++;
}

getch();
    textbackground(BLUE);
textcolor(WHITE);

}

/* tranfer data to printer */
print()
{
int Done=0;
double num1,num2,num3,num4,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    num5,num6,num7,num8,
    v1,v2,v3,v4,v5,v6,v7,v8;
char ch;
clock_t start1,start2,start3,start4,
start5,start6,start7,start8,
end1,end2,end3,end4,
end5,end6,end7,end8;
float diff1,diff2,diff3,diff4,
    diff5,diff6,diff7,diff8;
char z[]="          ";
char s1[]="Special1",c1[]="Common1",
    u1[]="Unlead1",di1[]="Diesel1",
    s2[]="Special2",c2[]="Common2",
    u2[]="Unlead2",di2[]="Diesel2";

    sound(800);
delay(100);
noisound();
textbackground(LIGHTGRAY);
textcolor(RED);
gotoxy(24,12);
cprintf("[5] Print data");
delay(1000);
    textbackground(BLUE);
textcolor(WHITE);
clrscr();
port_init(PORT,231);
    init_prn();
box2(2,1,79,25);
textbackground(RED);
textcolor(WHITE);
gotoxy(29,1);
cprintf(" QUANTITY AND COST REPORT ");
textbackground(BLUE);
textcolor(YELLOW);
gotoxy(5,3);

```

```

cprintf("OPERATOR : %s",op);
getdate(&today);
gotoxy(5,4);
cprintf("DATE      : %d/%d/%d",
today.da_day,today.da_mon,today.da_year+543);
textbackground(LIGHTBLUE);
textcolor(LIGHTRED);
gotoxy(5,6);
cprintf("Display present data");
gotoxy(5,16);
cprintf("Display total data");
textbackground(LIGHTBLUE);
textcolor(LIGHTCYAN);
gotoxy(29,6);cprintf(s1);
gotoxy(43,6);cprintf(c1);
gotoxy(56,6);cprintf(u1);
gotoxy(69,6);cprintf(di1);
gotoxy(29,11);cprintf(s2);
gotoxy(43,11);cprintf(c2);
gotoxy(56,11);cprintf(u2);
gotoxy(69,11);cprintf(di2);
gotoxy(29,16);cprintf(s1);
gotoxy(43,16);cprintf(c1);
gotoxy(56,16);cprintf(u1);
gotoxy(69,16);cprintf(di1);
gotoxy(29,20);cprintf(s2);
gotoxy(43,20);cprintf(c2);
gotoxy(56,20);cprintf(u2);
gotoxy(69,20);cprintf(di2);
textbackground(BLUE);
textcolor(WHITE);
gotoxy(5,7);
cprintf("1.Quantity(liter)");
gotoxy(5,8);
cprintf("2.Cost(baht)");
gotoxy(5,9);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("3.Time(hr:min)");
gotoxy(5,12);
printf("1.Quantity(liter)");
gotoxy(5,13);
printf("2.Cost(baht)");
gotoxy(5,14);
printf("3.Time(hr:min)");
gotoxy(5,17);
printf("1.Quantity(liter)");
gotoxy(5,18);
printf("2.Cost(baht)");
gotoxy(5,21);
printf("1.Quantity(liter)");
gotoxy(5,22);
printf("2.Cost(baht)");
    textbackground(LIGHTGRAY);
textcolor(RED);
/* initial valve */
qs1=0;qs2=0;    /* initial quantity */
qc1=0;qc2=0;
qu1=0;qu2=0;
qd1=0;qd2=0;
css1=0;css2=0; /* initial cost */
ccc1=0;ccc2=0;
cuu1=0;cuu2=0;
cdd1=0;cdd2=0;
tqs1=0;tqs2=0; /* initial total quantity */
tqc1=0;tqc2=0;
tqu1=0;tqu2=0;
tqd1=0;tqd2=0;
tcs1=0;tcs2=0; /* initial total cost */
tcc1=0;tcc2=0;
tcu1=0;tcu2=0;
tcd1=0;tcd2=0;
start1=0;start2=0; /* initial first time */
start3=0;start4=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

start5=0;start6=0;
start7=0;start8=0;
v1=0;v2=0;          /* initial valve of total cost */
v3=0;v4=0;
v5=0;v6=0;
v7=0;v8=0;
num1=0;num2=0;      /* initial valve of pulse */
num3=0;num4=0;
num5=0;num6=0;
num7=0;num8=0;

while(!Done)
{
    ch=rport(PORT);
    switch(ch)
    {
        case 'w' :    /* Special1 */
end1=clock();
v1++;
diff1=(end1-start1)/CLK_TCK;
if(diff1<=5)
    num1++;
else if((diff1>5)&&(qs1>1))
{
    prn1();
    num1=0;
}

        qs1=num1/ps;

        css1=qs1*cs;
        tq1=v1/ps;
        tcs1=tq1*cs;
        gotoxy(27,7);cprintf(z);
        gotoxy(27,7);cprintf("%.2lf",qs1);
        gotoxy(27,8);cprintf(z);
        gotoxy(27,8);cprintf("%.2lf",css1);
        gettime(&now1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    gotoxy(27,9);printf("  %02d:%02d  ",
now1.ti_hour,now1.ti_min);
    gotoxy(27,17);cprintf(z);
    gotoxy(27,17);cprintf("%.2lf",tqs1);
    gotoxy(27,18);cprintf(z);
    gotoxy(27,18);cprintf("%.2lf",tcs1);
        start1=clock();
break;
    case 'v' :      /* Special2 */
end2=clock();
v2++;
diff2=(end2-start2)/CLK_TCK;
if(diff2<=5)
    num2++;
    else if((diff2>5)&&(qs2>1))
{
    prn2();
    num2=0;
}
    qs2=num2/ps;
    css2=qs2*cs;
    tq2=v2/ps;
    tcs2=tq2*cs;
    gotoxy(27,12);cprintf(z);
    gotoxy(27,12);cprintf("%.2lf",qs2);
    gotoxy(27,13);cprintf(z);
    gotoxy(27,13);cprintf("%.2lf",css2);
    gettime(&now2);
    gotoxy(27,14);printf("  %02d:%02d  ",
now2.ti_hour,now2.ti_min);
    gotoxy(27,21);cprintf(z);
    gotoxy(27,21);cprintf("%.2lf",tqs2);
    gotoxy(27,22);cprintf(z);
    gotoxy(27,22);cprintf("%.2lf",tcs2);
    start2=clock();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;
    case 'j' : /* Common1 */
end3=clock();
v3++;
diff3=(end3-start3)/CLK_TCK;
if(diff3<=5)
    num3++;
else if((diff3>5)&&(qc1>1))
{
    prn3();
    num3=0;
}

qc1=num3/pc;
ccc1=qc1*cc;
tqc1=v3/pc;
tcc1=tqc1*cc;
gotoxy(41,7);cprintf(z);
gotoxy(41,7);cprintf("%.2lf",qc1);
gotoxy(41,8);cprintf(z);
gotoxy(41,8);cprintf("%.2lf",ccc1);
gettime(&now3);
gotoxy(41,9);cprintf(" %02d:%02d ",
now3.ti_hour,now3.ti_min);
gotoxy(41,17);cprintf(z);
gotoxy(41,17);cprintf("%.2lf",tqc1);
gotoxy(41,18);cprintf(z);
gotoxy(41,18);cprintf("%.2lf",tcc1);
start3=clock();
break;
    case 'r' : /* Common2 */
end4=clock();
v4++;
diff4=(end4-start4)/CLK_TCK;
if(diff4<=5)
    num4++;
else if((diff4>5)&&(qc2>1))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    prn4();
    num4=0;
}

qc2=num4/pc;
ccc2=qc2*cc;
tqc2=v4/pc;
tcc2=tqc2*cc;
gotoxy(41,12);cprintf(z);
gotoxy(41,12);cprintf("%.2lf",qc2);
gotoxy(41,13);cprintf(z);
gotoxy(41,13);cprintf("%.2lf",ccc2);
gettime(&now4);
gotoxy(41,14);cprintf("    %02d:%02d    ",
now4.ti_hour,now4.ti_min);
gotoxy(41,21);cprintf(z);
gotoxy(41,21);cprintf("%.2lf",tqc2);
gotoxy(41,22);cprintf(z);
gotoxy(41,22);cprintf("%.2lf",tcc2);
start4=clock();
break;
    case 'l' :    /* Unlead1 */
end5=clock();
v5++;
diff5=(end5-start5)/CLK_TCK;
if(diff5<=5)
    num5++;
else if((diff5>5)&&(qu1>1))
{
    prn5();
    num5=0;
}

qu1=num5/pu;
cuu1=qu1*cu;
tqu1=v5/pu;
tcu1=tqu1*cu;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(54,7);cprintf(z);
gotoxy(54,7);cprintf("%.2lf",qu1);
gotoxy(54,8);cprintf(z);
gotoxy(54,8);cprintf("%.2lf",cuu1);
gettime(&now5);
gotoxy(54,9);cprintf(" %02d:%02d ",
now5.ti_hour,now5.ti_min);
gotoxy(54,17);cprintf(z);
gotoxy(54,17);cprintf("%.2lf",tqu1);
gotoxy(54,18);cprintf(z);
gotoxy(54,18);cprintf("%.2lf",tcu1);
start5=clock();
break;
    case '๗' : /* Unlead2 */
end6=clock();
v6++;
diff6=(end6-start6)/CLK_TCK;
if(diff6<=5)
    num6++;
else if((diff6>5)&&(qu2>1))
{
    prn6();
    num6=0;
}
qu2=num6/pu;
cuu2=qu2*cu;
tqu2=v6/pu;
tcu2=tqu2*cu;
gotoxy(54,12);cprintf(z);
gotoxy(54,12);cprintf("%.2lf",qu2);
gotoxy(54,13);cprintf(z);
gotoxy(54,13);cprintf("%.2lf",cuu2);
gettime(&now6);
gotoxy(54,14);cprintf(" %02d:%02d ",
now6.ti_hour,now6.ti_min);
gotoxy(54,21);cprintf(z);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(54,21);cprintf("%.2lf",tqu2);
gotoxy(54,22);cprintf(z);
gotoxy(54,22);cprintf("%.2lf",tcu2);
start6=clock();
break;
    case '7' : /* Diesel1 */
end7=clock();
v7++;
diff7=(end7-start7)/CLK_TCK;
if(diff7<=5)
    num7++;
else if((diff7>5)&&(qd1>1))
{
    prn7();
    num7=0;
}
qd1=num7/pd;
cdd1=qd1*cd;
tqd1=v7/pd;
ted1=tqd1*cd;
gotoxy(67,7);cprintf(z);
gotoxy(67,7);cprintf("%.2lf",qd1);
gotoxy(67,8);cprintf(z);
gotoxy(67,8);cprintf("%.2lf",cdd1);
gettime(&now7);
gotoxy(67,9);cprintf(" %02d:%02d  ",
now7.ti_hour,now7.ti_min);
gotoxy(67,17);cprintf(z);
gotoxy(67,17);cprintf("%.2lf",tqd1);
gotoxy(67,18);cprintf(z);
gotoxy(67,18);cprintf("%.2lf",ted1);
start7=clock();

break;
    case '8' : /* Diesel2 */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end8=clock();
v8++;
diff8=(end8-start8)/CLK_TCK;
if(diff8<=5)
    num8++;
else if((diff8>5)&&(qd2>1))
{
    prn8();
    num8=0;
}
qd2=num8/pd;
cdd2=qd2*cd;
tqd2=v8/pd;
tcd2=tqd2*cd;
gotoxy(67,12);cprintf(z);
gotoxy(67,12);cprintf("%.2lf",qd2);
gotoxy(67,13);cprintf(z);
gotoxy(67,13);cprintf("%.2lf",cdd2);
gettime(&now8);
gotoxy(67,14);cprintf(" %02d:%02d ",
now8.ti_hour,now8.ti_min);
gotoxy(67,21);cprintf(z);
gotoxy(67,21);cprintf("%.2lf",tqd2);
gotoxy(67,22);cprintf(z);
gotoxy(67,22);cprintf("%.2lf",tcd2);
start8=clock();

break;
    default:break;
}
if(toupper(ch)=='Q') Done ++;
}
textbackground(BLUE);
textcolor(WHITE);
}

```

```

/* initial printer port */
init_prn()
{
union REGS r;
r.h.ah=1; /*initializes printer */
r.x.dx=0; /* printer number */
int86(0x17,&r,&r);
}

/* check printer status */
status()
{
union REGS r;
r.h.ah=2; /*check printer status */
r.x.dx=0; /*select first printer */
int86(0x17,&r,&r);
return(r.h.ah&128);
}

/* display status printer port */
check_prn()
{
if(!(status()&128))
{
gotoxy(40,24);
printf("Send error detected in printer port ");
getch();
atexit(main());
}
}

/* print special1 */
prn1()
{
check_prn();
fprintf(stdprn,"Type:Special Benzine(1)\n");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

getdate(&today);
fprintf(stdprn, "Date:%d/%d/%d\n",
today.da_day, today.da_mon, today.da_year+543);
fprintf(stdprn, "Time:%02d:%02d\n",
now1.ti_hour, now1.ti_min);
fprintf(stdprn, "Quantity:%.2lf liter\n", qs1);
fprintf(stdprn, "Cost:%.2lf Baht\n\n", css1);
}

/* print special2 */
prn2()
{
check_prn();
fprintf(stdprn, "Type:Special Benzine(2)\n");
getdate(&today);
fprintf(stdprn, "Date:%d/%d/%d\n",
today.da_day, today.da_mon, today.da_year+543);
fprintf(stdprn, "Time:%02d:%02d\n",
now2.ti_hour, now2.ti_min);
fprintf(stdprn, "Quantity:%.2lf liter\n", qs2);
fprintf(stdprn, "Cost:%.2lf Baht\n\n", css2);
}

/* print common1 */
prn3()
{
check_prn();
fprintf(stdprn, "Type:Common Benzine(1)\n");
getdate(&today);
fprintf(stdprn, "Date:%d/%d/%d\n",
today.da_day, today.da_mon, today.da_year+543);
fprintf(stdprn, "Time:%02d:%02d\n",
now3.ti_hour, now3.ti_min);
fprintf(stdprn, "Quantity:%.2lf liter\n", qc1);
fprintf(stdprn, "Cost:%.2lf Baht\n\n", ccc1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* print common2 */
prn4()
{
check_prn();
fprintf(stdprn,"Type:Common Benzine(2)\n");
getdate(&today);
fprintf(stdprn,"Date:%d/%d/%d\n",
today.da_day,today.da_mon,today.da_year+543);
fprintf(stdprn,"Time:%02d:%02d\n",
now4.ti_hour,now4.ti_min);
fprintf(stdprn,"Quantity:%.2lf liter\n",qc2);
fprintf(stdprn,"Cost:%.2lf Baht\n\n",ccc2);
}

/* print unlead1 */
prn5()
{
check_prn();
fprintf(stdprn,"Type:Unlead Benzine(1)\n");
getdate(&today);
fprintf(stdprn,"Date:%d/%d/%d\n",
today.da_day,today.da_mon,today.da_year+543);
fprintf(stdprn,"Time:%02d:%02d\n",
now5.ti_hour,now5.ti_min);
fprintf(stdprn,"Quantity:%.2lf liter\n",qu1);
fprintf(stdprn,"Cost:%.2lf Baht\n\n",cuu1);
}

/* print unlead2 */
prn6()
{
check_prn();
fprintf(stdprn,"Type:Unlead Benzine(2)\n");
getdate(&today);
fprintf(stdprn,"Date:%d/%d/%d\n",
today.da_day,today.da_mon,today.da_year+543);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fprintf(stdprn,"Time:%02d:%02d\n",
now6.ti_hour,now6.ti_min);
fprintf(stdprn,"Quantity:%.2lf liter\n",qu2);
fprintf(stdprn,"Cost:%.2lf Baht\n\n",cuu2);
}

/* print diesel1 */
prn7()
{
check_prn();
fprintf(stdprn,"Type:Diesel Oil(1)\n");
getdate(&today);
fprintf(stdprn,"Date:%d/%d/%d\n",
today.da_day,today.da_mon,today.da_year+543);
fprintf(stdprn,"Time:%02d:%02d\n",
now7.ti_hour,now7.ti_min);
fprintf(stdprn,"Quantity:%.2lf liter\n",qd1);
fprintf(stdprn,"Cost:%.2lf Baht\n\n",cdd1);
}

/* print diesel2 */
prn8()
{
check_prn();
fprintf(stdprn,"Type:Diesel Oil(2)\n");
getdate(&today);
fprintf(stdprn,"Date:%d/%d/%d\n",
today.da_day,today.da_mon,today.da_year+543);
fprintf(stdprn,"Time:%02d:%02d\n",
now8.ti_hour,now8.ti_min);
fprintf(stdprn,"Quantity:%.2lf liter\n",qd2);
fprintf(stdprn,"Cost:%.2lf Baht\n\n",cdd2);
}

/* quit programme */
quit()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
char ch;
textbackground(LIGHTGRAY);
textcolor(RED);
gotoxy(24,13);
printf("[6] Quit");
delay(1000);
    textbackground(BLUE);
textcolor(WHITE);
clrscr();
box1(2,1,79,25);
/*laser();
laser();
laser();*/
textbackground(BLACK);
gotoxy(27,12);printf(" ");
gotoxy(27,13);printf(" ");
gotoxy(27,14);printf(" ");
gotoxy(27,15);printf(" ");
gotoxy(27,16);printf(" ");
textbackground(RED);
textcolor(WHITE);
textcolor(LIGHTCYAN);
gotoxy(26,11);printf(" ");
gotoxy(26,12);printf(" ");
gotoxy(26,13);printf(" Are you sure to quit?(y/n) ");
gotoxy(26,14);printf(" ");
gotoxy(26,15);printf(" ");
    box2(27,11,56,15);
ch=getch();
if((toupper(ch))== 'Y')
    exit(0);
textbackground(BLUE);
textcolor(WHITE);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* create box1 */
box1(int xul,int yul,int xlr,int ylr)
{
int i;
char hz=196,vt=179,ul=218,ur=191,lr=217,ll=192;
gotoxy(xul,yul);
for(i=xul;i<=xlr;i++)
    cprintf("%c",hz);
gotoxy(xul,ylr);
for(i=xul;i<=xlr;i++)
    cprintf("%c",hz);
for(i=yul;i<=ylr;i++)
{
    gotoxy(xul,i);cprintf("%c",vt);
    gotoxy(xlr,i);cprintf("%c",vt);
}
gotoxy(xul,yul);cprintf("%c",ul);
gotoxy(xlr,yul);cprintf("%c",ur);
gotoxy(xlr,ylr);cprintf("%c",lr);
gotoxy(xul,ylr);cprintf("%c",ll);
}

/* create box2 */
box2(int xul,int yul,int xlr,int ylr)
{
int i;
char hz=0xCD,
    vt=0xBA,
    ul=0xC9,
    ur=0xBB,
    lr=0xBC,
    ll=0xC8;
gotoxy(xul,yul);
for(i=xul;i<=xlr;i++)
    cprintf("%c",hz);
gotoxy(xul,ylr);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=xul;i<=xlr;i++)
    cprintf("%c",hz);
for(i=yul;i<=ylr;i++)
{
    gotoxy(xul,i);cprintf("%c",vt);
    gotoxy(xlr,i);cprintf("%c",vt);
}
gotoxy(xul,yul);cprintf("%c",ul);
gotoxy(xlr,yul);cprintf("%c",ur);
gotoxy(xlr,ylr);cprintf("%c",lr);
gotoxy(xul,ylr);cprintf("%c",ll);
}

/* Read a character from a port. */
rport(int port) /* i/o port */
{
union REGS r;

/* wait for a character */
while(!(check_stat(port)&256))
if(kbhit())
{ /* abort on keypress */
    getch();
    atexit(main());
}

r.x.dx = port; /* serial port */
r.h.ah = 2; /* read character function */
int86(0x14, &r, &r);
if(r.h.ah & 128)
{
    gotoxy(5,24);
    printf("Read error detected in serial port");
    getch();
    atexit(main());
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return r.h.al;
}

/* Check the status of the serial port. */
check_stat(int port) /* i/o port */
{
    union REGS r;

    r.x.dx = port; /* serial port */
    r.h.ah = 3; /* read status */
    int86(0x14, &r, &r);
    return r.x.ax;
}

/* Initialize the port. */
port_init(int port, unsigned char code)
{
    union REGS r;

    r.x.dx = port; /* serial port */
    r.h.ah = 0; /* initialize port function */
    r.h.al = code; /* initialization code - see text for details */
    int86(0x14, &r, &r);
}

/* set cursor off */
cursor_off()
{
    union REGS reg;
    reg.h.ch=0x20; /*turn on bit 5 */
    reg.h.cl=0;
    reg.h.ah=1; /*call BIOS function */
    int86(0x10,&reg,&reg);
}

/* set cursor on */

```

```

cursor_on()
{
    union REGS reg;
    reg.h.ch=0x00; /*turn on bit 5 */
    reg.h.cl=13;
    reg.h.ah=1; /*call BIOS function */
    int86(0x10,&reg,&reg);
}

#define DELAY 5000
#define RATE 50

/* Create a laser blast effect. */
laser()
{
    unsigned freq;
    unsigned long i;

    union {
        long divisor;
        unsigned char c[2];
    } count;

    unsigned char p;

    p = inportb(97); /* get existing bit pattern */
    outportb(97, p | 3); /* turn on bits 0 and 1 */
    /* laser blast */
    for(freq = 3000; freq>1000; freq-=RATE) {
        count.divisor = 1193280 / freq; /* compute the proper count */
        outportb(67, 182); /* tell 8253 that count is coming */
        outportb(66, count.c[0]); /* send low-order byte */
        outportb(66, count.c[1]); /* send high-order byte */

        for(i=0; i<DELAY; ++i) ;
    }

    outportb(97, p); /* restore original bits to turn off speaker */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/* initial graphic mode */
Initialize()
{
    int xasp,yasp;
    GraphDriver=DETECT;
    initgraph(&GraphDriver,&GraphMode,"");
    ErrorCode=graphresult();
    if(ErrorCode!=grOk)
    {
printf("Graphics System Error:%s\n",
grapherrormsg(ErrorCode));
exit(1);
    }
    MaxX=getmaxx();
    MaxY=getmaxy();
    MaxColors=getmaxcolor()+1;
}

/* stop to display screen */
Pause()
{
    setcolor(LIGHTMAGENTA);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    outtextxy(470,460,"Press any key...");
    while(kbhit()) getch();
    getch();
}

/* display screen no.1 */
Show1()
{
    int width=getmaxx(),height=getmaxy(),left=0,top=0;
    getpalette(&palette);
    setviewport(left,top+5,left+width,top+height+5,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setfillstyle(CLOSE_DOT_FILL,LIGHTGRAY);
rectangle(0,0,MaxX,MaxY);
floodfill(10,10,getcolor());
setcolor(WHITE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,4);
outtextxy(200,50,"KING MONGKUT'S");
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(175,120,"INSTITUTE OF TECHNOLOGY");
outtextxy(250,160,"LADKRABANG");
outtextxy(180,200,"FACULTY OF ENGINEERING");
outtextxy(220,300,"DEPARTMENT OF ");
outtextxy(80,350,"INDUSTRIAL INSTRUMENTATION TECHNOLOGY");
}

/* display screen no.2 */
Show2()
{
    int width=getmaxx(),height=getmaxy(),left=0,top=0;
    cleardevice();
    setviewport(left,top+5,left+width,top+height+5,0);
    getpalette(&palette);
    setfillstyle(XHATCH_FILL,LIGHTRED);
    setcolor(LIGHTBLUE);
    rectangle(0,0,MaxX,MaxY);
    floodfill(10,10,getcolor());
    setcolor(YELLOW);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,4);
    outtextxy(90,90,"OIL QUANTITY MEASUREMENT");
    outtextxy(50,140,"IN OIL PUMP WITH MICROCOMPUTER");
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
    outtextxy(300,220,"BY");
    outtextxy(180,270,"MR. AMARIN CHUCHATTHA");
    outtextxy(180,320,"MR. KITTIPONG MALITHONG");
    outtextxy(180,370,"MR. SAINGAM HOMKACHON");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* make line graph */
Line_Graph()
{
int width=getmaxx(),height=getmaxy(),left=0,top=0;
int i,j,hstep,vstep,vsteps,bottom,x,y,MaxVal=0;
double scale;
for(i=0;i<=3;i++)
    for(j=0;j<=18;j++)
        if(Accounts[i][j]>MaxVal) MaxVal=Accounts[i][j];
setviewport(left,top+5,left+width,top+height+5,0);
width-=30;
hstep=width/20;
vsteps=(MaxVal/25)+2;
vstep=height/vsteps;
bottom=vstep*vsteps;
scale=(double)vstep/(double)25;
if(MaxColors>4)setcolor(LIGHTBLUE);
for(i=0;i<=hstep*32;i+=hstep*4)
    line(i,0,i,vstep*vsteps);
if(MaxColors>4) setcolor(WHITE);
settextjustify(CENTER_TEXT,CENTER_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
for(i=0;i<=4;i++)
{
    x=i*hstep*4+2*hstep;
    y=vstep/2;
    gprintf(&x,&y,"%d",Years[i]);
}
settextjustify(CENTER_TEXT,CENTER_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
for(i=0;i<=bottom;i+=vstep)
{
    if(MaxColors>4) setcolor(LIGHTBLUE);
    line(0,i,hstep*20,i);
    if(MaxColors>4)setcolor(YELLOW);
    x=hstep*20+15;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y=i;
printf(&x,&y,"%d",((bottom-i)/vstep)*25);
}
settextjustify(LEFT_TEXT,TOP_TEXT);
for(i=0;i<=3;i++)
{

setlinestyle(i,0,3);
if(MaxColors>4) setcolor(i+2);
moveto(hstep/2-10,bottom-scale*Accounts[i][0]+14);
outtext(AccTypes[i]);
moveto(hstep/2,bottom-scale*Accounts[i][0]);
    for(j=0;j<=19;j++)
    {
        lineto(hstep/2+j*hstep,bottom-scale*Accounts[i][j]);
        putimage(getx()-10,gety()-10,grimage[i],XOR_PUT);
    }
}
}

/* make each of symbol */
Create_Images()
{
    /*
int i;
int figure[]={10,5,15,5,12,10,17,10,
    10,18,12,13,8,13,10,5};
cleardevice();
setfillstyle(EMPTY_FILL,0);
if(MaxColors>4)setcolor(2);
for(i=0;i<=5;i++)
    pieslice(10,10,i*60,(i*60)+60,10);
grimage[0]=malloc(imagesize(0,0,20,20));
getimage(0,0,20,20,grimage[0]);
putimage(0,0,grimage[0],XOR_PUT);
if(MaxColors>4)setcolor(3);
fillpoly(sizeof(figure)/(2*sizeof(int)),figure);

```

```

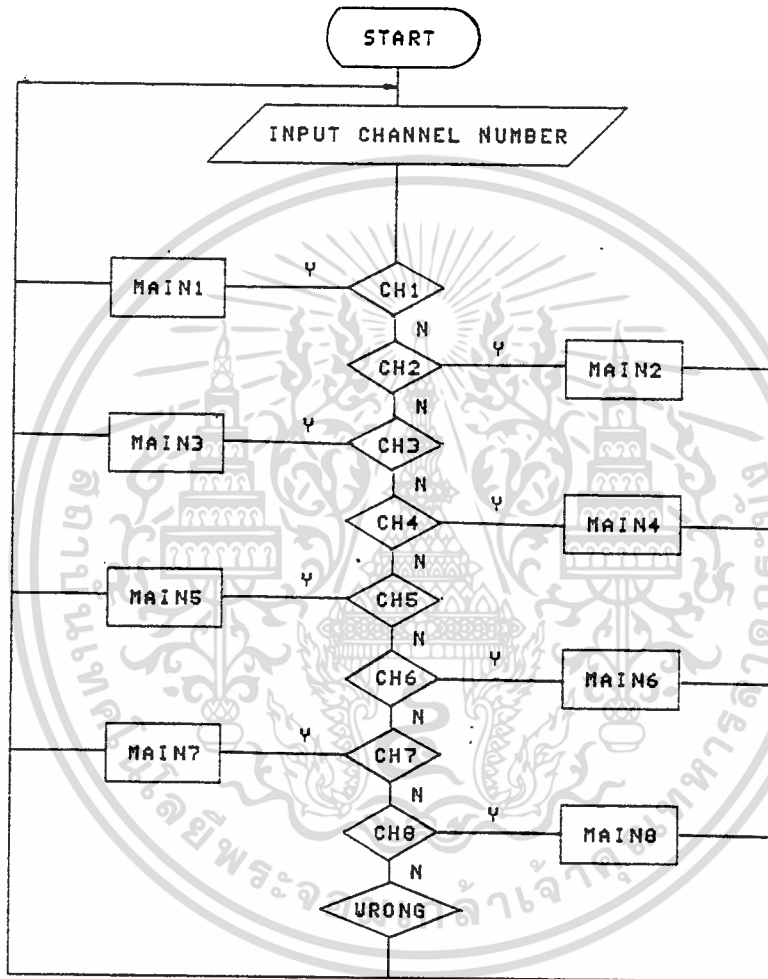
grimage[1]=malloc(imagesize(0,0,20,20));
getimage(0,0,20,20,grimage[1]);
putimage(0,0,grimage[1],XOR_PUT);
if(MaxColors>4) setcolor(4);
setfillstyle(SOLID_FILL,getcolor());
arc(10,10,105,360,8);
ellipse(0,5,300,90,8,3);
ellipse(3,10,0,90,16,8);
floodfill(10,10,getcolor());
grimage[2]=malloc(imagesize(0,0,20,20));
getimage(0,0,20,20,grimage[2]);
putimage(0,0,grimage[2],XOR_PUT);
if(MaxColors>4) setcolor(5);
bar3d(0,10,10,15,5,1);
grimage[3]=malloc(imagesize(0,0,20,20));
getimage(0,0,20,20,grimage[3]);
putimage(0,0,grimage[3],XOR_PUT);
    */
}

/* commulation display screen */
showscreen()
{
Initialize();
Show1();
Pause();
Show2();
Pause();
Create_Images();
Line_Graph();
Pause();
closegraph();
}

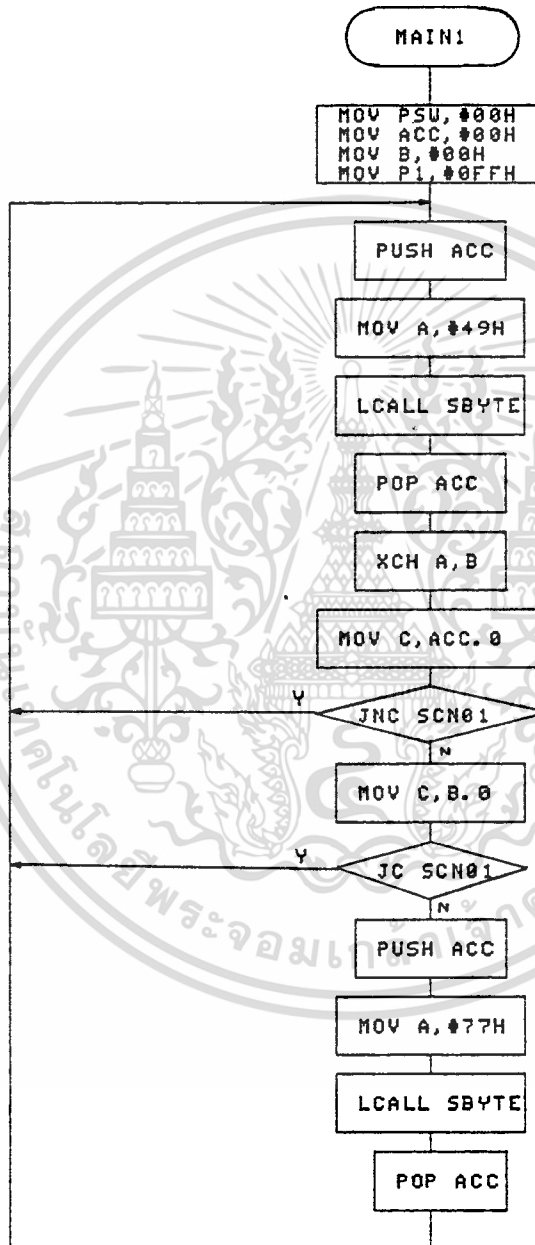
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

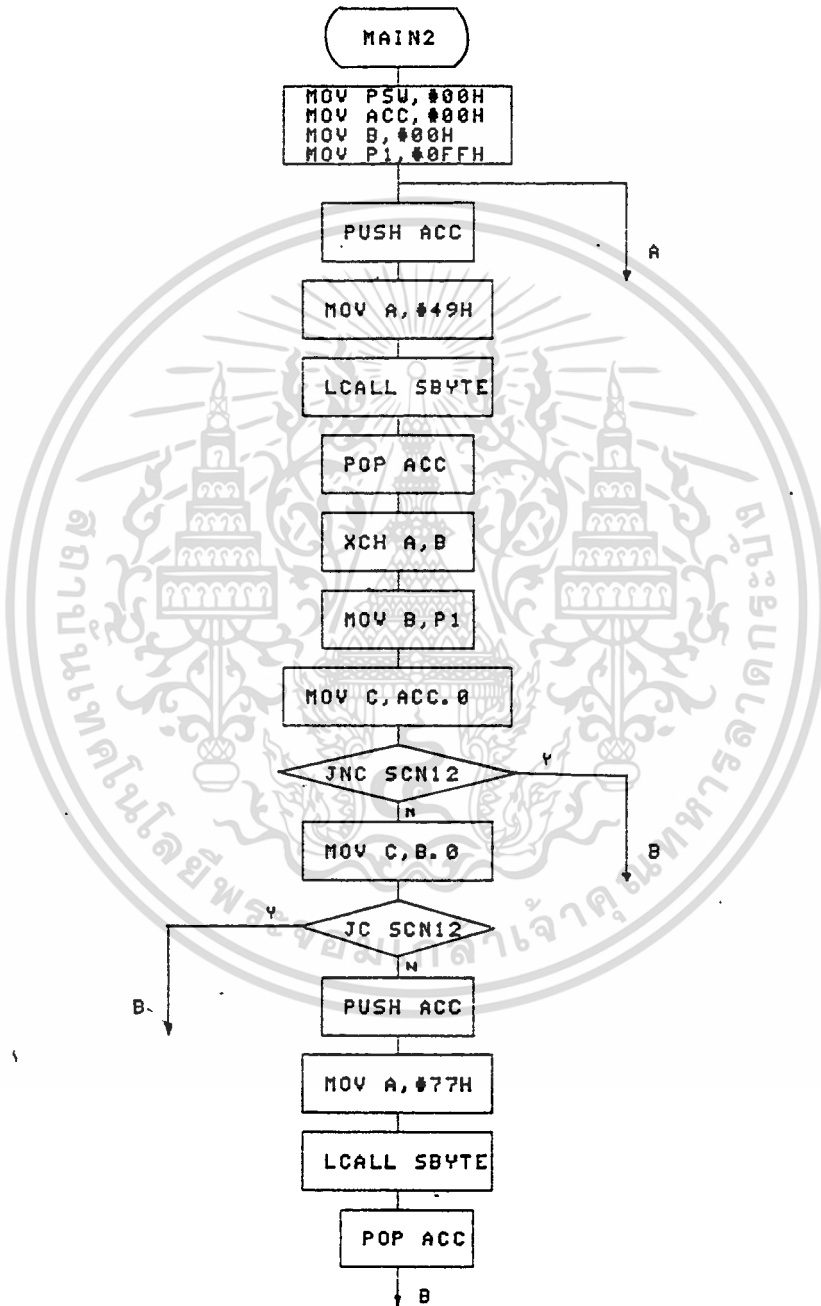
5.3 Assembly Flowchart



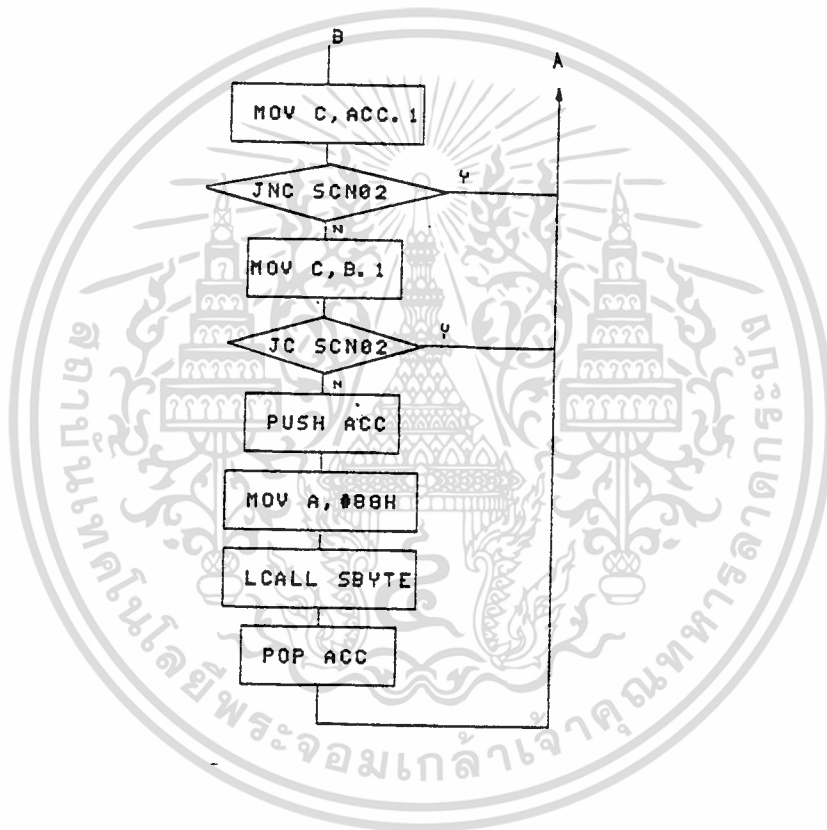
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 Assembly Program

```

;This program is used for count number of pulse that is
;travel to the pin of port1 of 80C31 chip.
;The programe is wOPT 1.5OPT 1.OPT 1.50
;When the pulse changes from high level to be low level ,the
;programe which work as counter
;When the pulse changes from high level to low level ,the
;programe which work as counter is going to count
;Maximam frequency that programe can detect mustn't be
;greater than 54 HZ,So that the performance of this
;programe is quite excellent .For the using of this program
;first press the RST key on the PC CONSOLE board
;then select the number of channel that varies from 1 to 8.
;In the end after selected the number of channel you
;must press the ENT key .The JAZZ31 and PC will work together.
;Before you select the number of channel ,the pumb program
;must be run surely.For selecting the number of channel,certainly
;you have to know amount of pumb.

```

```

;file name 8PINCNTR.ASM
;Hardware Jazz31 Single Board MCS
;Programmer H.Saingam

```

```

#include"j31head.asm"

```

```

org 8000h

```

```

inputno_chn;      lcall inh
                  cjne  a,#10h,inputno_chn1
                  sjmp  main1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

inputno_chn1:    cjne  a,#20h,inputno_chn2
                 ljmp  main2

inputno_chn2:    cjne  a,#30h,inputno_chn3
                 ljmp  main3

inputno_chn3:    cjne  a,#40h,inputno_chn4
                 ljmp  main4

inputno_chn4:    cjne  a,#50h,inputno_chn5
                 ljmp  main5

inputno_chn5:    cjne  a,#60h,inputno_chn6
                 ljmp  main6

inputno_chn6:    cjne  a,#70h,inputno_chn7
                 ljmp  main7

inputno_chn7:    cjne  a,#80h,wrong
                 ljmp  main8

wrong:          ljmp  inputno_chn

```

```

;-----
;
;          ONE CHNS CNTR
;-----

```

```

main1:          mov   psw,#00h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    a,#00h

mov    b,#00h

mov    p1,#0ffh

mov    b,p1

scn01:  push  acc

mov    a,#49h

lcall  sbyte

pop    acc

xch   a,b

mov    c,acc.0

jnc   scn01

mov    c,b.0

jc    scn01

push  acc

mov    a,#77h

lcall  sbyte

pop    acc

sjmp  scn01

;-----
;
;          TWO CHNS CNTR
;-----

main2:  mov    psw,#00h

mov    a,#00h

mov    b,#00h

mov    p1,#0ffh

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

scn02:      push  acc
            mov   a,#49h
            lcall sbyte
            pop   acc
            xch  a,b
            mov  b,p1
            mov  c,acc.0
            jnc  scn12
            mov  c,b.0
            jc   scn12
            push acc
            mov  a,#77h
            lcall sbyte
            pop  acc
scn12:      mov  c,acc.1
            jnc  scn02
            mov  c,b.1
            jc   scn02
            push acc
            mov  a,#88h
            lcall sbyte
            pop  acc
            sjmp scn02

```

```

;-----
;          THREE CHNS CNTR
;-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

main3:      mov    psw,#00h

            mov    a,#00h

            mov    b,#00h

            mov    p1,#0ffh

```

```

scn03:      push   acc

            mov    a,#49h

            lcall  sbyte

            pop    acc

            xch   a,b

            mov    b,p1

            mov    c,acc.0

            jnc   scn13

            mov    c,b.0

            jc    scn13.

            push  acc

            mov    a,#77h

            lcall  sbyte

            pop    acc

```

```

scn13:      mov    c,acc.1

            jnc   scn23

            mov    c,b.1

            jc    scn23

            push  acc

            mov    a,#88h

            lcall  sbyte

            pop    acc

```

```

scn23:      mov    c,acc.2
            jnc    scn03
            mov    c,b.2
            jc     scn03
            push  acc
            mov    a,#99h
            lcall sbyte
            pop   acc
            sjmp  scn03

```

```

;-----
;
;-----

```

```

main4:     mov    psw,#00h
            mov    a,#00h
            mov    b,#00h
            mov    p1,#0ffh

```

```

scn04:     push  acc
            mov    a,#49h
            lcall sbyte
            pop   acc
            xch   a,b
            mov    b,p1
            mov    c,acc.0
            jnc   scn14
            mov    c,b.0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        jc     scn14

        push  acc

        mov   a,#77h

        lcall sbyte

        pop   acc

scn14:   mov   c,acc.1

        jnc  scn24

        mov  c,b.1

        jc   scn24

        push acc

        mov  a,#88h

        lcall sbyte

        pop  acc

scn24:   mov   c,acc.2

        jnc  scn34

        mov  c,b.2

        jc   scn34

        push acc

        mov  a,#99h

        lcall sbyte

        pop  acc

scn34:   mov   c,acc.3

        jnc  scn04

        mov  c,b.3

        jc   scn04

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

push  acc
mov    a,#0aah
lcall sbyte
pop    acc
sjmp  scn04

```

```

;-----
;                               FIVE CHNS CNTR
;-----

```

```

main5:  mov    psw,#00h
        mov    a,#00h
        mov    b,#00h
        mov    p1,#0ffh

```

```

scn05:  xch    a,b
        mov    b,p1
        mov    c,acc.0
        jnc   scn15
        mov    c,b.0
        jc    scn15

```

```

push  acc
mov    a,#77h
lcall sbyte
pop    acc

```

```

scn15:  mov    c,acc.1
        jnc   scn25

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    c,b.1
jc     scn25
push  acc
mov    a,#88h
lcall sbyte
pop   acc

```

```

scn25:  mov    c,acc.2

```

```

jnc    scn35
mov    c,b.2
jc     scn35
push  acc
mov    a,#99h
lcall sbyte
pop   acc

```

```

scn35:

```

```

mov    c,acc.3
jnc    scn45
mov    c,b.3
jc     scn45
push  acc
mov    a,#0aah
lcall sbyte
pop   acc

```

```

scn45:  mov    c,acc.4

```

```

jnc    scn55
mov    c,b.4

```

```

        jc     scn55
        push  acc
        mov   a,#0bbh
        lcall sbyte
        pop   acc

scn55:   push  acc
        mov   a,#49h
        lcall sbyte
        pop   acc
        ljmp  scn05
;-----;
;        SIX CHNS CNTR
;-----;
main6:   mov   psw,#00h
        mov   a,#00h
        mov   b,#00h
        mov   p1,#0ffh

scn06:   xch   a,b
        mov   b,p1
        mov   c,acc.0
        jnc  scn16
        mov   c,b.0
        jc   scn16
        push  acc

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    a,#77h
lcall  sbyte
pop    acc

scn16:  mov    c,acc.1
        jnc    scn26
        mov    c,b.1
        jc     scn26
        push  acc
        mov    a,#88h
        lcall  sbyte
        pop    acc

scn26:  mov    c,acc.2
        jnc    scn36
        mov    c,b.2
        jc     scn36
        push  acc
        mov    a,#99h
        lcall  sbyte
        pop    acc

scn36:  mov    c,acc.3
        jnc    scn46
        mov    c,b.3
        jc     scn46
        push  acc
        mov    a,#0aah

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lcall sbyte
```

```
pop acc
```

```
scn46: mov c,acc.4
```

```
jnc scn56
```

```
mov c,b.4
```

```
jc scn56
```

```
push acc
```

```
mov a,#0bbh
```

```
lcall sbyte
```

```
pop acc
```

```
scn56: mov c,acc.5
```

```
jnc scn66
```

```
mov c,b.5
```

```
jc scn66
```

```
push acc
```

```
mov a,#0cch
```

```
lcall sbyte
```

```
pop acc
```

```
scn66: push acc
```

```
mov a,#49h
```

```
lcall sbyte
```

```
pop acc
```

```
ljmp scn06
```

```

;-----
;                               SEVEN CHNS CNTR
;-----

```

```

main7:      mov    psw,#00h

            mov    a,#00h

            mov    b,#00h

            mov    p1,#0ffh

```

```

scn07:      xch    a,b

            mov    b,p1

            mov    c,acc.0

            jnc   scn17

            mov    c,b.0

            jc    scn17

            push  acc

            mov    a,#77h

            lcall sbyte

            pop   acc

```

```

scn17:      mov    c,acc.1

            jnc   scn27

            mov    c,b.1

            jc    scn27

            push  acc

            mov    a,#88h

            lcall sbyte

            pop   acc

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

scn27:      mov    c,acc.2

            jnc    scn37

            mov    c,b.2

            jc     scn37

            push  acc

            mov    a,#99h

            lcall sbyte

            pop   acc

```

```

scn37:      mov    c,acc.3

            jnc    scn47

            mov    c,b.3

            jc     scn47

            push  acc

            mov    a,#0aah

            lcall sbyte

            pop   acc

```

```

scn47:      mov    c,acc.4

            jnc    scn57

            mov    c,b.4

            jc     scn57

            push  acc

            mov    a,#0bbh

            lcall sbyte

            pop   acc

```

```

scn57:      mov    c,acc.5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

jnc   scn67
mov   c,b.5
jc    scn67
push  acc
mov   a,#0cch
lcall sbyte
pop   acc

```

```

scn67:   mov   c,acc.6
         jnc   scn77
         mov   c,b.6
         jc    scn77
         push  acc
         mov   a,#0ddh
         lcall sbyte
         pop   acc

```

```

scn77:   push  acc
         mov   a,#49h
         lcall sbyte
         pop   acc
         ljmp scn07

```

```

;-----
;           EIGH CHNS CNTR
;-----

```

```

main8:   mov   psw,#00h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    a,#00h
mov    b,#00h
mov    p1,#0ffh

scn08:  xch    a,b
        mov    b,p1
        mov    c,acc.0
        jnc    scn18
        mov    c,b.0
        jc     scn18
        push  acc
        mov    a,#77h
        lcall sbyte
        pop   acc
scn18:  mov    c,acc.1
        jnc    scn28
        mov    c,b.1
        jc     scn28
        push  acc
        mov    a,#88h
        lcall sbyte
        pop   acc

scn28:  mov    c,acc.2
        jnc    scn38
        mov    c,b.2
        jc     scn38

```

```

        push  acc
        mov   a,#99h
        lcall sbyte
        pop   acc

scn38:  mov   c,acc.3
        jnc   scn48
        mov   c,b.3
        jc    scn48
        push  acc
        mov   a,#0aah
        lcall sbyte
        pop   acc
scn48:  mov   c,acc.4
        jnc   scn58
        mov   c,b.4
        jc    scn58
        push  acc
        mov   a,#0bbh
        lcall sbyte
        pop   acc

scn58:  mov   c,acc.5
        jnc   scn68
        mov   c,b.5
        jc    scn68
        push  acc

```

```

mov    a,#0cch
lcall  sbyte
pop    acc

scn68:  mov    c,acc.6
        jnc    scn78
        mov    c,b.6
        jc     scn78
        push  acc
        mov    a,#0ddh
        lcall  sbyte
        pop    acc

scn78:  mov    c,acc.7
        jnc    scn88
        mov    c,b.7
        jc     scn88
        push  acc
        mov    a,#0eeh
        lcall  sbyte
        pop    acc

scn88:  push  acc
        mov    a,#49h
        lcall  sbyte
        pop    acc
        ljmp  scn08
        end

```

บทที่ 6

สรุปผลของโครงการ

6.1 คำนำ

โครงการนี้เป็นโครงการเริ่มแรก ซึ่งก่อให้เกิดปัญหาในการทำงานในด้านต่างๆมากมาย ไม่ว่าจะเป็นเรื่องของข้อมูลและเรื่องของโปรแกรม อาทิเช่น

1. การพัฒนาโปรแกรมบน 8031 ต้องใช้ภาษา Assembly ที่มีคำสั่งในการใช้งานน้อย ยิ่งยากต่อการเขียน software ควบคุม นอกจากนี้เครื่องมือในการพัฒนามีน้อย คุณภาพต่ำ ซึ่งต่างจากการพัฒนาบน PC เครื่องอำนวยความสะดวกมากมายและมีประสิทธิภาพสูง

2. Buad Rate ระหว่าง Microcontroller และ PC ซึ่ง Buad Rate ของไมโครคอนโทรลเลอร์สามารถใช้ได้ถึง 19200 บิตต่อวินาที ขณะที่บน PC ซึ่งเขียนโปรแกรมโดยใช้ภาษาที่สามารถใช้ Buad Rate ได้เพียง 9600 บิตต่อวินาที ทำให้เกิดปัญหาในการติดต่อสื่อสารคือทำให้การทำงานช้าลง

3. รายละเอียดเกี่ยวกับตัว CPU 8031 ค่อนข้างหาได้ยาก ไม่เป็นที่แพร่หลาย จึงต้องใช้เวลาในการศึกษามาก

ซึ่งปัญหาดังกล่าวเป็นปัญหาเริ่มแรก แต่โครงการนี้ก็จัดทำขึ้นเป็นไปตามลำดับและขั้นตอนของการดำเนินงาน จนสำเร็จลุล่วงตามเป้าหมายที่วางไว้และสามารถนำโครงการนี้ไปประยุกต์ใช้วัดปริมาณน้ำมัน, ราคา และอื่นๆได้ตามความเหมาะสม

6.2 สรุปผลที่ได้จากโครงการ

จากการที่ได้สร้างโครงการวัดปริมาณน้ำมันในปั้มน้ำมันด้วยคอมพิวเตอร์ ทำให้ได้เครื่องมือที่ช่วยในการเพิ่มความสะดวกในการทำงานในปั้มน้ำมัน อาทิ

1. ราคาของน้ำมันในปั้มน้ำมันจะเปลี่ยนแปลงไปตามระบบตลาดน้ำมันในประเทศ ทำให้ต้องเปลี่ยนแปลงราคาน้ำมันที่หัวจ่ายน้ำมันบ่อยๆ [ราคาน้ำมันที่ขาย (บาท/ลิตร)] ซึ่งในปั้มน้ำมันแต่ละปั้มนั้นก็จะมีหัวจ่ายน้ำมันอยู่หลายชนิดและราคาน้ำมันแต่ละชนิดก็ไม่เท่ากัน ถ้าใช้การเปลี่ยนแปลงราคาน้ำมันที่หัวจ่ายน้ำมันแต่ละหัวแล้วก็จะเกิดความล่าช้า ซึ่งโครงการนี้ก็สามารที่จะปรับราคาน้ำมันแต่ละหัวจ่ายได้จากเครื่องคอมพิวเตอร์ ซึ่งก่อให้เกิดความสะดวกรวดเร็ว

2. ปริมาณสะสมของน้ำมันที่จ่ายในแต่ละ วัน/เดือน/ปี ต้องอาศัยการบันทึกทางบัญชีถ้าใช้คอมพิวเตอร์เป็นตัวบันทึกจะก่อให้เกิดความถูกต้องและแม่นยำกว่า

3. รายได้จากการขายน้ำมันแต่ละชนิด แต่ละครั้ง ต้องอาศัยการบันทึกทางบัญชี ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

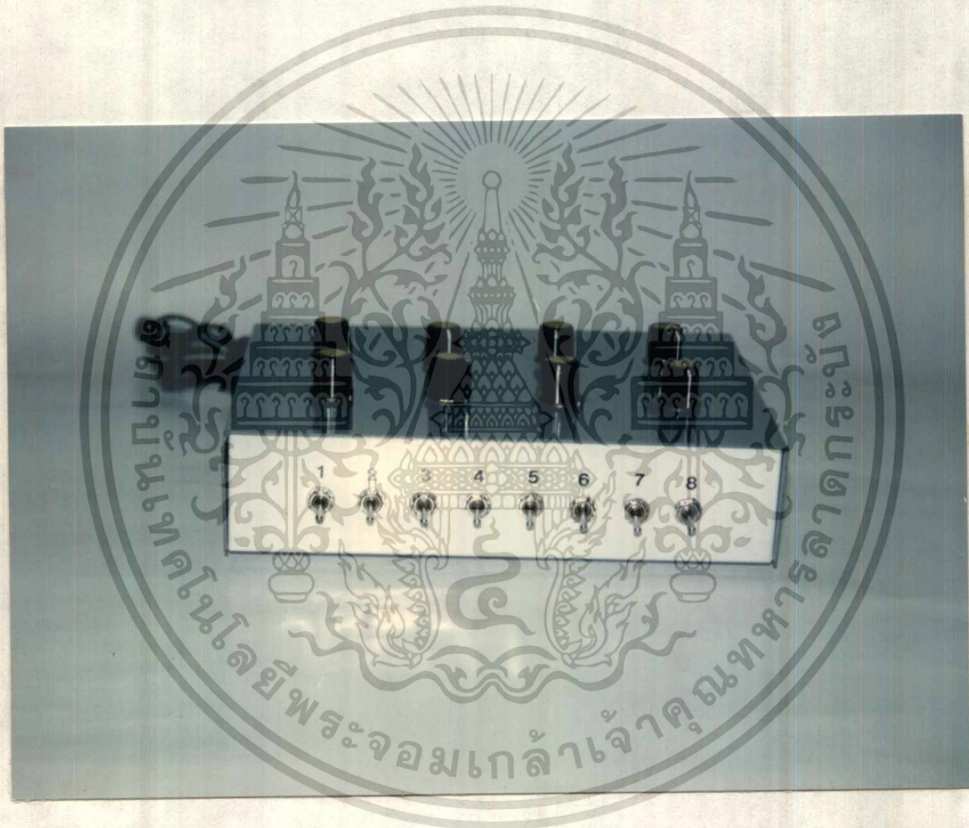
ให้เกิดความผิดพลาดและล่าช้าได้ โครงการนี้ซึ่งใช้คอมพิวเตอร์ในการควบคุมจะช่วยให้การจัดปัญหาดังกล่าวให้หมดไป

ซึ่งทางผู้จัดทำโครงการนี้หวังว่า โครงการนี้จะเป็นประโยชน์ต่อผู้นำไปประยุกต์ใช้ไม่มากนักน



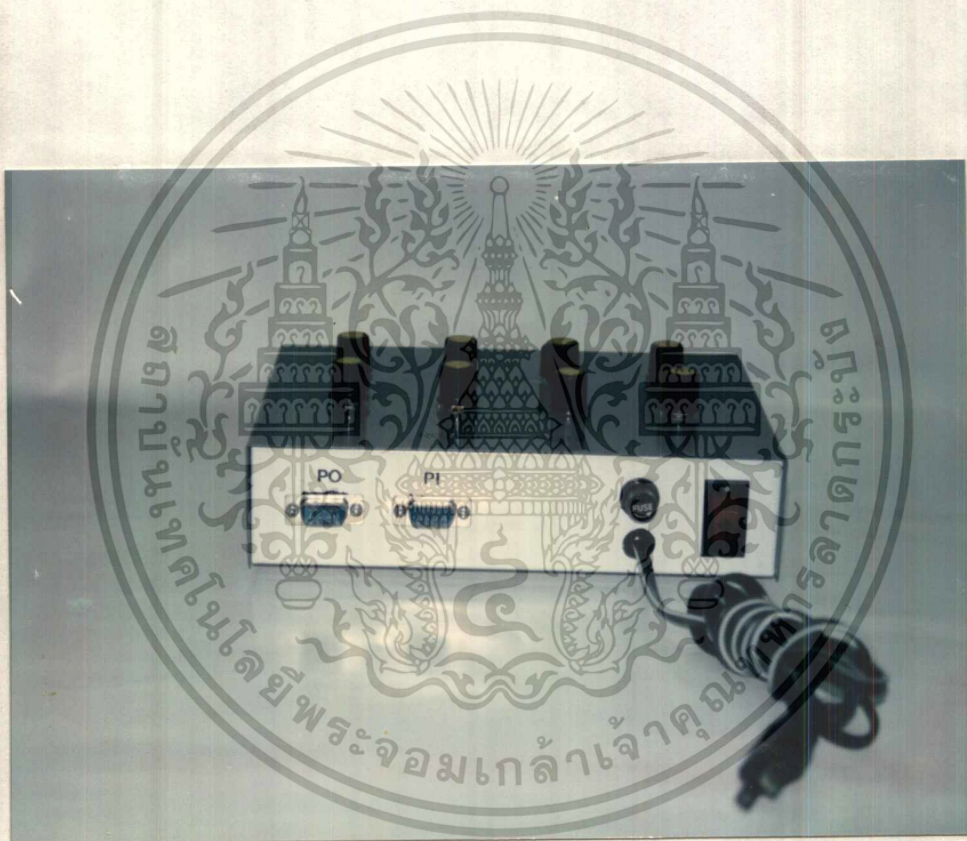
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก



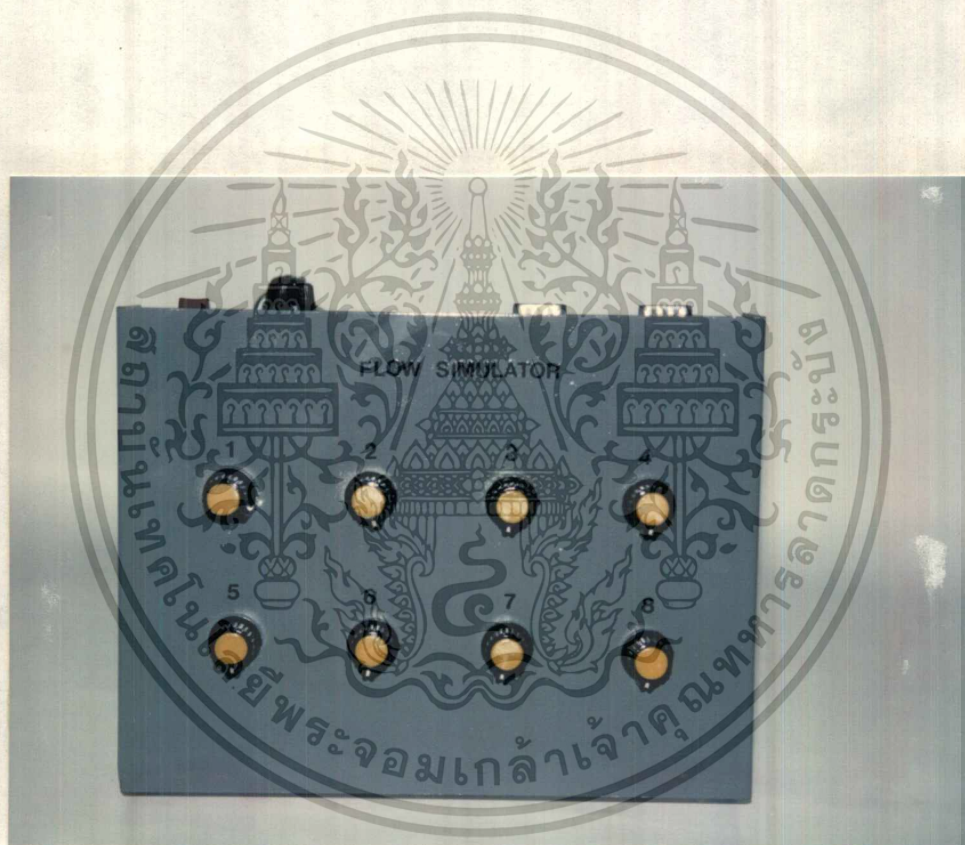
รูป ก-1 แสดงภาพ Flow simulator ด้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



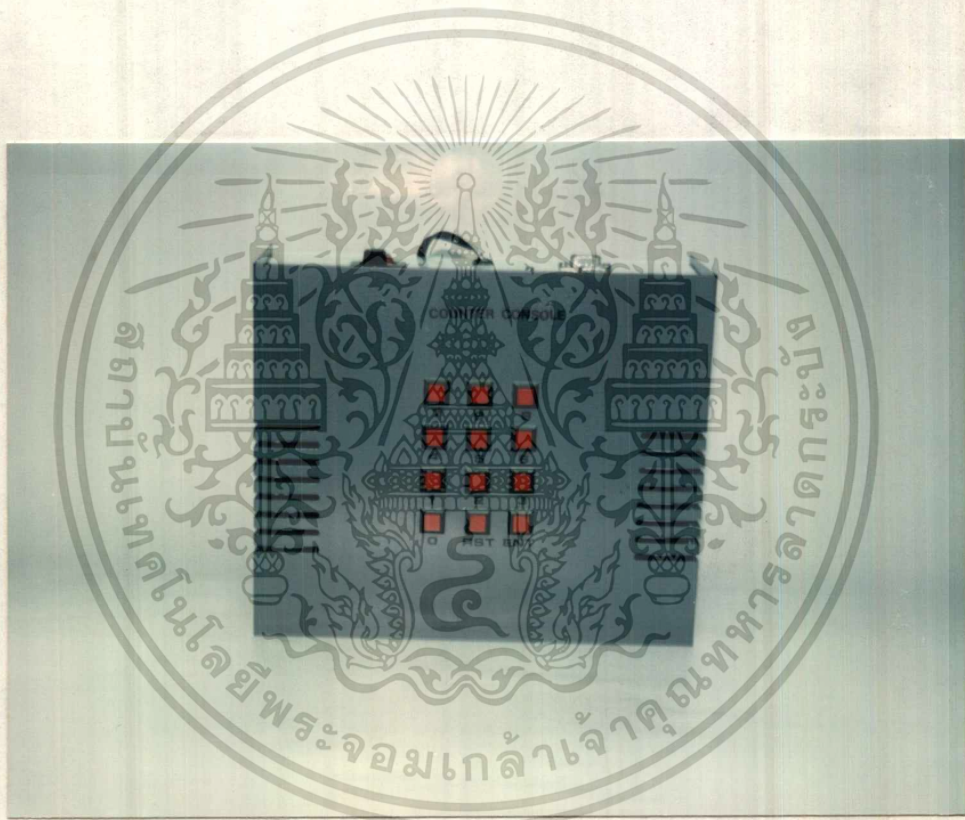
ภาพ ก-2 แสดงภาพ Flow Simulater ด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



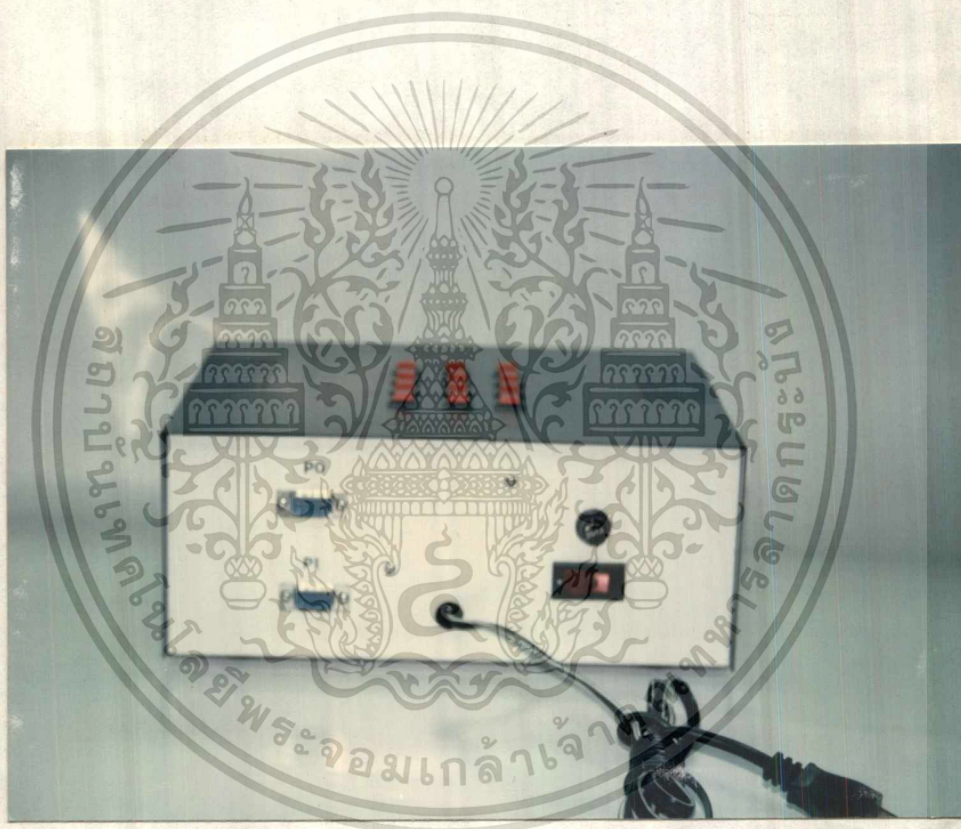
ภาพ ก-3 แสดงภาพ Flow simulator ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



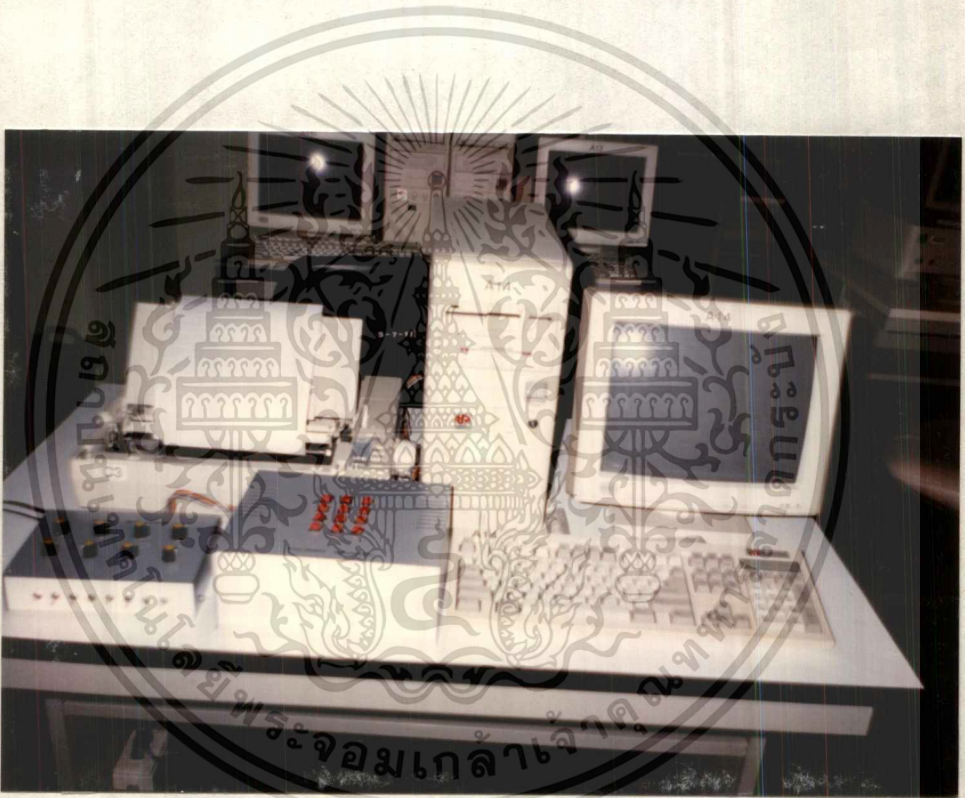
ภาพ ก-4 แสดงภาพ Counter Console ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ก-5 แสดงภาพ Counter Console ด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ก-6 แสดงภาพโดยรวมของการทดลองโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาโทฉบับนี้สำเร็จลงไปได้ด้วยดี เป็นเพราะด้วยความกรุณาให้คำปรึกษา และช่วยเหลือทางด้านเทคนิค ตลอดจนเครื่องมือและอุปกรณ์ต่างๆจากอาจารย์ที่ปรึกษาและภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์ และเพื่อน้องๆทุกคนที่ให้ความช่วยเหลือเป็นอย่างดี

ขอขอบพระคุณ ผศ. วิชา ทิพย์สุวรรณพร เป็นอย่างสูงที่ได้ให้คำแนะนำและคำปรึกษา ที่เป็นประโยชน์ต่อปริญญาโทฉบับนี้

ขอขอบพระคุณ คุณพ่อ-คุณแม่ และคุณอา ของคณะผู้จัดทำ ที่ได้ให้ทุนทรัพย์และให้กำลังใจมาโดยตลอด

คณะผู้จัดทำ



บรรณานุกรม

1. Turbo C Reference Guide, Version 2.0, Borland International, Inc., 1988
2. Turbo C User's Guide, Version 2.0, Borland International, Inc., 1988
3. Pual Chui, Graphics Programming in Turbo C 2.0, Borland International, Inc., 1988
4. SILA RESEARCH, JAZZ-31 SINGLE BOARD MICROCONTROLLER USER MANUAL V1.1, SILA RESEARCH CO., LTD., 1991
5. SILA RESEARCH, MCS-51 SUMMARY, SILA RESEARCH CO., LTD., 1991
6. Kenneth J. Ayala, The 8051 Microcontroller ARCHITECTURE, PROGRAMMING AND APPLICATIONS, WEST PUBLISHING COMPANY, 1991
7. รศ. มัณฑนา ปราการสมุทร, การเขียนชุดคำสั่งภาษาซี, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
8. ธนัท ชัยยุทธ, กณพ แก้วพิชัย, การเขียนโปรแกรมภาษาปาสคาลเบื้องต้น, บริษัท ซีเอ็ดดูเคชั่น จำกัด, 2521
9. ศิววัฒน์ ศิวะบวร, พรชัย จักรธำรงค์, จิรศักดิ์ ชัยวิริยะกุล, การประยุกต์ใช้งานภาษาซี, บริษัท ซีเอ็ดดูเคชั่น จำกัด, 2535
10. พิพัฒน์ เลาสงคราม, ไมโครคอนโทรลเลอร์ MCS-48 MCS-51, ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้