

ริโมทิต้าต้าลือกเกอร์



ปฏิญานินนรนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์

ภาควิชาเทคโนโลยีอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

.....  
(รศ.ดร.สมเกียรติ ศุภเดช)

คณะบดีคณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า

เจ้าคุณทหารลาดกระบัง

คณะกรรมการสอบปริญญาโท

.....ประธานกรรมการ

(  
.....กรรมการ

(  
.....กรรมการ

(

อาจารย์ที่ปรึกษา

ดร.ไพศาล นาคินันต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	รีโมทด้าด้าล็อกเกอร์
ชื่อ	รณชัย หาไชย
อาจารย์ที่ปรึกษา	ดร. ไนศาล นาคพิพันธ์
ภาควิชา	เทคนิคอุตสาหกรรม
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	2536

บทคัดย่อ

โครงการนี้ถูกออกแบบขึ้นเพื่อใช้งานแทนด้าด้าล็อกเกอร์ที่นำเข้ามาจากต่างประเทศ ที่มีราคาค่อนข้างแพง อีกทั้งการใช้งานก็อยู่ในวงจำกัด ไม่สะดวก

ด้าด้าล็อกเกอร์ส่วนใหญ่จะใช้สำหรับเก็บข้อมูลจากงานวิจัย หรืองานทดลองต่างๆ ซึ่งข้อมูลดังกล่าวจะอยู่ในรูปของแรงดันไฟฟ้า ที่เปลี่ยนมาจากข้อมูลทางฟิสิกส์ เคมี หรือค่าวัดอื่นๆ ด้วยตัวตรวจวัด (Sensor) โดยด้าด้าล็อกเกอร์จะเปลี่ยนแรงดันไฟฟ้าดังกล่าวเป็นข้อมูลเชิงตัวเลขตามค่าวัดใช้งานจริงก่อน (ด้วยการนำไปคูณกับค่าสัมประสิทธิ์ (Multiplier) และบวกด้วยค่าความแตกต่างจากศูนย์ (Offset)) แล้วจึงบันทึกหรือจัดเก็บ

การสร้างเริ่มต้นด้วยการออกแบบฮาร์ดแวร์ของโครงการ โดยแบ่งโครงการออกเป็น 2 ส่วน คือ การ์ดที่ทำหน้าที่เชื่อมต่อระบบสื่อสารกับเครื่องไมโครคอมพิวเตอร์ (Interface Card) การออกแบบจะทำบนการ์ดทดลอง (Prototype Card) และเชื่อมต่อวงจรด้วยสายไวร์แวร์ป (Wire Wrap) การทดสอบการทำงานของการ์ดจะใช้โปรแกรมภาษาแอสเซมบลี (Assembly) ส่วนโปรแกรมใช้งานจริงจะเขียนด้วยภาษาปาสคาล (Pascal)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกส่วนหนึ่งก็คือหน่วยเชื่อมต่อ (Terminal Unit) การออกแบบทดลอง จะทำบนบอร์ดทดลองวงจร (Breadboard) การออกแบบและทดสอบการทำงานของส่วนนี้จะใช้ รอม-อิมูเลเตอร์ (Rom Emulator) ซึ่งได้พัฒนาขึ้นเอง โดยเขียนโปรแกรมด้วยภาษาแอสเซมบลี (Z80 Assembly) และแปลเป็นภาษาเครื่อง (Object Code) ด้วยโปรแกรมครอสแอสเซม-เบลอร์ (Cross Assembler) แล้วจึงโหลดลงบนรอมอิมูเลเตอร์เพื่อทดสอบการทำงานเมื่อได้วงจรที่สมบูรณ์ เมื่อได้วงจรที่ทำงานได้ถูกต้องสมบูรณ์แล้ว ก็จะนำวงจรที่ได้ไปออกแบบลายวงจรเป็นแผ่นวงจรพิมพ์ (Print Circuit Board) โดยใช้โปรแกรมโปรเทล พีซีบี (Protel PCB) ช่วยในการออกแบบ

งบประมาณที่ใช้ไปสำหรับโครงการนี้แบ่งออกได้ดังนี้

- ค่าอุปกรณ์ของการ์ดเชื่อมต่อประมาณ	500	บาท
- ค่าอุปกรณ์ของหน่วยเชื่อมต่อ (1 ชุด) ประมาณ	2400	บาท
- ค่าใช้จ่ายสำหรับทำแผ่นวงจรพิมพ์แบบ 2 หน้า จำนวน 3 แผ่นประมาณ	3700	บาท
- ค่าอุปกรณ์เบ็ดเตล็ดที่นำมาประกอบประมาณ	800	บาท
รวมค่าใช้จ่ายประมาณ	7400	บาท

เมื่อฮาร์ดแวร์ของโครงการทั้งหมดเสร็จแล้วจึงเริ่มปรับปรุงโปรแกรมเพื่อให้ผู้ใช้สามารถใช้งานได้สะดวก รวมถึงรูปแบบของการนำเสนอข้อมูลบนจอภาพ และรายงานที่เป็นระเบียบสวยงาม

การทดสอบการทำงานของโปรแกรมจะทำโดยการเปลี่ยนหน้าเมนูของโปรแกรมไปมาระหว่างหน้าเมนูต่างๆอย่างรวดเร็ว และการทดสอบการทำงานของฮาร์ดแวร์ทำโดยเปิดเครื่องใช้งานติดต่อกันตลอด 24 ชั่วโมง โดยป้อนแรงดันที่อินพุทแบบอนาล็อกด้วยแหล่งกำเนิดแรงดันมาตรฐาน พร้อมทั้งดูค่าความผิดพลาดของค่าวัดเทียบกับเครื่องกำเนิดแรงดันมาตรฐาน

สำหรับปัญหาที่เกิดขึ้นในการออกแบบโครงการนี้ จะอยู่ที่ส่วนติดต่อสื่อสาร ทั้งนี้เนื่องจากไม่สามารถตรวจสอบความถูกต้องของข้อมูลที่สื่อสารกันได้ ทำให้เสียเวลามาก

โครงการนี้สามารถพัฒนาขึ้นได้อีก เช่น สามารถเพิ่มจำนวนอินพุทแบบอนาล็อกและดิจิตอล , จำนวนเข้าพุทแบบดิจิตอลให้มากขึ้น เพิ่มเข้าพุทแบบอนาล็อก อินพุทแบบเค้าน์เตอร์

## Abstract

This project was designed for used instead of imported datalogger that was expensive and uncomfortable.

Almost datalogger was used to collected data from researchs and experiments. That data was formed in voltage that converted from physical data , chemical data and other measured data by sensor. Datalogger should converted this data to numerical data (multiplied by multiplier and added with offset) and then collected them.

The designing was begun by designed hardware of the project that was divided into 2 parts. First was a computer card that interfaced communication system to the personal computer (Communication Card). Designing was done on prototype card and connected by wirewrapping cable. To testing the operation of this card used assembly language program. The control program was wrote by used pascal language.

The other was Terminal Unit that designed on bread board and tested by used rom-emulator that designed one myself. The testing was wrote control program by used Z80 assembly language and interpreted by cross assembler program and then loaded to rom-emulator. When the circuits were completed then leted them to designed print circuit board by used Protel PCB program.

The budget of this project

- the budget of components in communication card 500 B.
- the budget of components in terminal unit (1 set) 2400 B.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- the budget for making double-sided PCB (3 plates) 3700 B.
- the budget for accessories 800 B.
- total budget 7400 B.

When the hardware of project was completed then I developed program to easy to used and arranged displaying and reporting.

To testing the operation of program done by changed the menu over fast. And testing the operation of hardware done by operatred them over 24 hours , connected analog input to standard voltage source and focus the measurement display compared with statdard voltage.

The trouble that ocured during designed this project was in communication part because I didn't verified the data that communicated so paid a long time.

This project can developing such as the number of analog input, digital input and digital output will be increase ,adding analog output, counter input. Modify program to stand alone when malfunction in personal computer to protect the lost of data.

## กิตติกรรมประกาศ

ขอแสดงความขอบคุณมายังท่านอาจารย์ที่ปรึกษาโครงการ ที่กรุณาให้ความช่วยเหลือ คำแนะนำ และคำปรึกษาที่เป็นประโยชน์ในการจัดทำโครงการนี้จนเสร็จสมบูรณ์ รวมถึงท่านผู้ แต่งหนังสือ บทความ และคู่มือต่างๆ ที่ได้นำมาใช้อ้างอิง และเป็นแหล่งค้นคว้าข้อมูลที่สำคัญ ถ้าโครงการนี้จะเป็นประโยชน์ประการใด ขอยกความดีทั้งหมดให้กับบุคคลดังกล่าว ข้างต้นทั้งหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	จ
บทที่	
1	1
2	5
3	10
4	45
5	62
บรรณานุกรม	
ภาคผนวก ก	
ภาคผนวก ข	
ภาคผนวก ค	

## สารบัญรูป

รูปที่		หน้า
1.1	แสดงการจัดเครือข่ายของระบบ	1
1.2	แสดงตัวอย่างการใช้งานโครงงาน	2
2.1	แสดงการจัดเครือข่ายแบบ Point To Point	5
2.2	แสดงการจัดเครือข่ายแบบ Multipoint	6
2.3	แสดงส่วนประกอบของ เฟรม (Frame)	8
3.1	แสดงบล็อกไดอะแกรมการทำงานของ Communication Card	10
3.2	แสดงวงจรในส่วนของ Address Select Enable	11
3.3	แสดงวงจรในส่วนของ Address Port Decoder	12
3.4	แสดงวงจรในส่วนของการกำเนิดสัญญาณเวลา 10 mS	13
3.5	แสดงการเชื่อมต่อระหว่าง CPU กับ 8273	14
3.6	แสดงวงจรในส่วนของ DMA Acknowledge Enable	15
3.7	แสดงบล็อกไดอะแกรมของ Terminal Unit	16
3.8	แสดงวงจรในส่วนของ Memory Address Decoder	17
3.9	แสดงวงจรในส่วนของ Port Address Decoder	18
3.10	แสดงวงจรในส่วนของ Analog Multiplexer	18
3.11	แสดงวงจรในส่วนของ Channel Select	19
3.12	แสดงวงจรในส่วนของ Analog To Digital Converter	20
3.13	แสดงไหลวชาร์ทการทำงานของโปรแกรมควบคุม Communication Card	24
3.14	แสดงไหลวชาร์ทการทำงานของโปรแกรมควบคุม Terminal Unit	27
3.15	แสดงโครงสร้างเมนูของโปรแกรม Relog	29

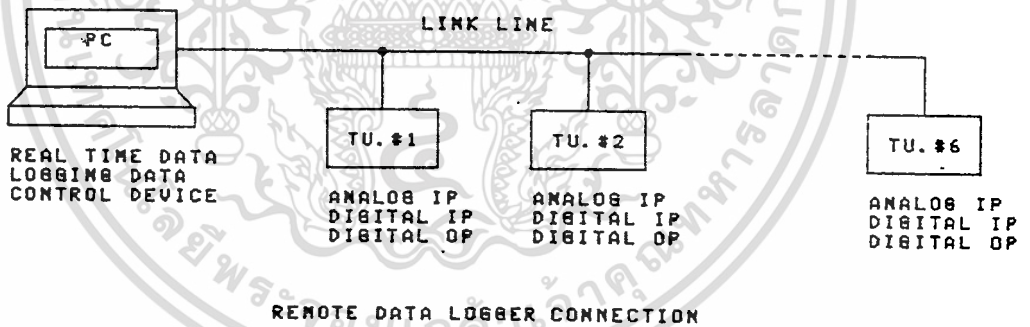
รูปที่	หน้า
3.16 แสดงหน้าเมนู Main Menu ของโปรแกรม Relog	30
3.17 แสดงหน้าเมนู Measure ของโปรแกรม Relog	31
3.18 แสดงหน้าเมนู Status ของโปรแกรม Relog	33
3.19 แสดงหน้าเมนู Control ของโปรแกรม Relog	34
3.20 แสดงโครงสร้างเมนูของ โปรแกรม Edcon	35
3.21 แสดงหน้าเมนู Station ของโปรแกรม Edcon	36
3.22 แสดงหน้าเมนู Measure ของโปรแกรม Edcon	37
3.23 แสดงหน้าเมนู Status ของโปรแกรม Edcon	39
3.24 แสดงหน้าเมนู Control ของโปรแกรม Edcon	40
3.25 แสดงโครงสร้างเมนูของ โปรแกรม View	41
3.26 แสดงหน้าเมนู Main Menu ของโปรแกรม View	42
3.27 แสดงหน้าเมนู View Data ของโปรแกรม View	43
4.1 แสดงการต่อใช้งานระบบกับ Green House	47
4.2 แสดงวงจรต่อเชื่อมกับอุปกรณ์ในระบบที่ทดลองใช้งาน	61

## บทที่ 1

### วัตถุประสงค์ของ โครงการงาน

เนื่องจาก Data Logger ที่มีข้อมูลในปัจจุบันจะทำงานแยกกันเป็นอิสระไม่สามารถนำมาใช้งานร่วมกันเป็นระบบเครือข่ายได้ และการนำข้อมูลที่เก็บบันทึกได้มาใช้ประโยชน์ จะต้องนำ Data Logger นั้นมา Dump ข้อมูลเอง อีกทั้งบางกรณีผู้ใช้ต้องการดูค่าการเปลี่ยนแปลงในระบบที่เวลาปัจจุบัน ก็ทำได้ยากลำบาก

จากข้อด้อยดังกล่าวข้างต้น จึงนำมาเป็นหัวข้อหลักในการออกแบบโครงการนี้ กล่าวคือโครงการนี้จะจัดระบบการทำงานเป็นแบบเครือข่าย (Network) โดยใช้เครื่องคอมพิวเตอร์บุคคล (Personal Computer , PC) เป็นสถานีหลักของระบบ และมีหน่วยจัดเก็บข้อมูล (Terminal Unit, TU.) เป็นสถานีรองหรือลูกข่ายในระบบ การจัดเครือข่ายของระบบจะเป็นแบบ Multipoint ดังแสดงในรูปที่ 1.1

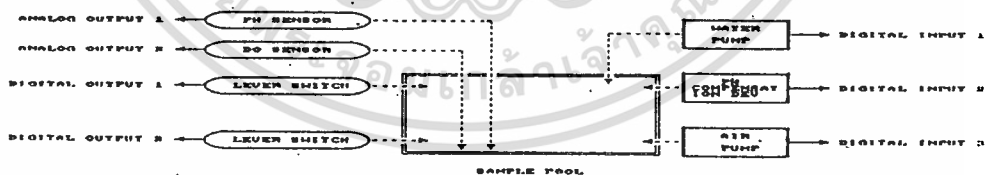


รูปที่ 1.1 แสดงการจัดเครือข่ายของระบบ

สถานีหลักซึ่งต่อไปนี้จะเรียกว่า PC จะทำหน้าที่ติดต่อกับผู้ใช้ (User) และควบคุมการจัดเก็บข้อมูลทั้งหมดในระบบ โดยจะร้องขอข้อมูล จากสถานีรอง ซึ่งต่อไปจะเรียกว่า TU เรียงลำดับกันไปจนครบทุกตัวในระบบ ส่วน TU. จะมีหน้าที่จัดเก็บข้อมูลในระบบที่ต้องการบันทึก และส่งให้กับ PC ผ่านทางสายเชื่อมต่อ (Link Line) เมื่อ PC ได้รับข้อมูลดังกล่าว

แล้วจะนำมาคำนวณ และแสดงผลให้ผู้ใช้เห็นบนจอมอนิเตอร์ ซึ่งเป็นข้อมูลตามเวลาจริง (Real Time Data) พร้อมทั้งจัดเก็บลงในดิสก์ตามช่วงเวลาที่ใช้กำหนด

ข้อดีอีกประการหนึ่งก็คือ มีอินพุท และเอาพุทแบบดิจิตอล (Digital Input Output) โดยอินพุทแบบดิจิตอล จะใช้สำหรับตรวจสอบสถานะ (Status) ของอุปกรณ์ในระบบที่ทำการบันทึกข้อมูล และเอาพุทแบบดิจิตอลใช้สำหรับควบคุมการทำงาน (Control) ของอุปกรณ์ที่ต้องการควบคุมในระบบ ทั้งนี้เพื่อให้ผู้ใช้สามารถเฝ้าดูและควบคุมความเปลี่ยนแปลงที่เกิดขึ้นในระบบที่ทำการทดลองหรือเก็บบันทึกข้อมูลได้ตลอดเวลา ยกตัวอย่างเช่น ระบบการเพาะเลี้ยงและอนุบาลลูกปลา ประกอบด้วย บ่อเพาะฟัก เซ็นเซอร์ตรวจวัดค่าออกซิเจนในน้ำ (DO) เซ็นเซอร์ตรวจวัดค่าความเป็นกรดด่าง (PH) สวิตช์ระดับสำหรับตรวจระดับน้ำในบ่อที่ตำแหน่งสูงสุด และต่ำสุด บัมพ์น้ำ เครื่องเติมอากาศ และเครื่องชดเชยความเป็นกรดด่าง ดังแสดงในรูปที่ 1.2



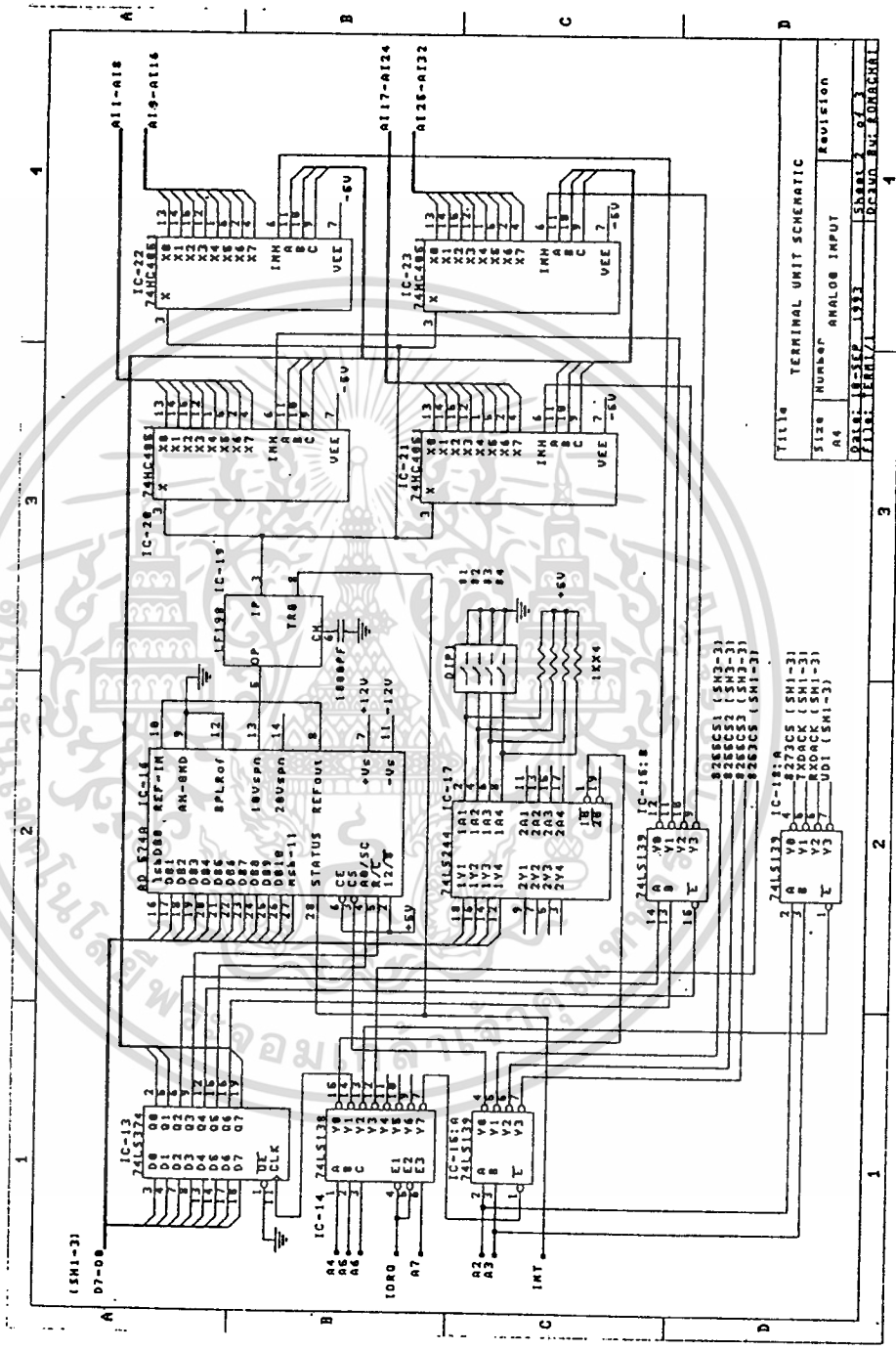
รูปที่ 1.2 แสดงตัวอย่างการใช้งานโครงงาน

ผู้ใช้สามารถดูความเปลี่ยนแปลงของค่าออกซิเจนในน้ำ และค่าความเป็นกรดด่างด้วยอินพุทอนาล็อก ดูระดับน้ำในบ่อด้วยอินพุทแบบดิจิตอล และควบคุมการเปิดปิดของอุปกรณ์ด้วยเอาพุทแบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



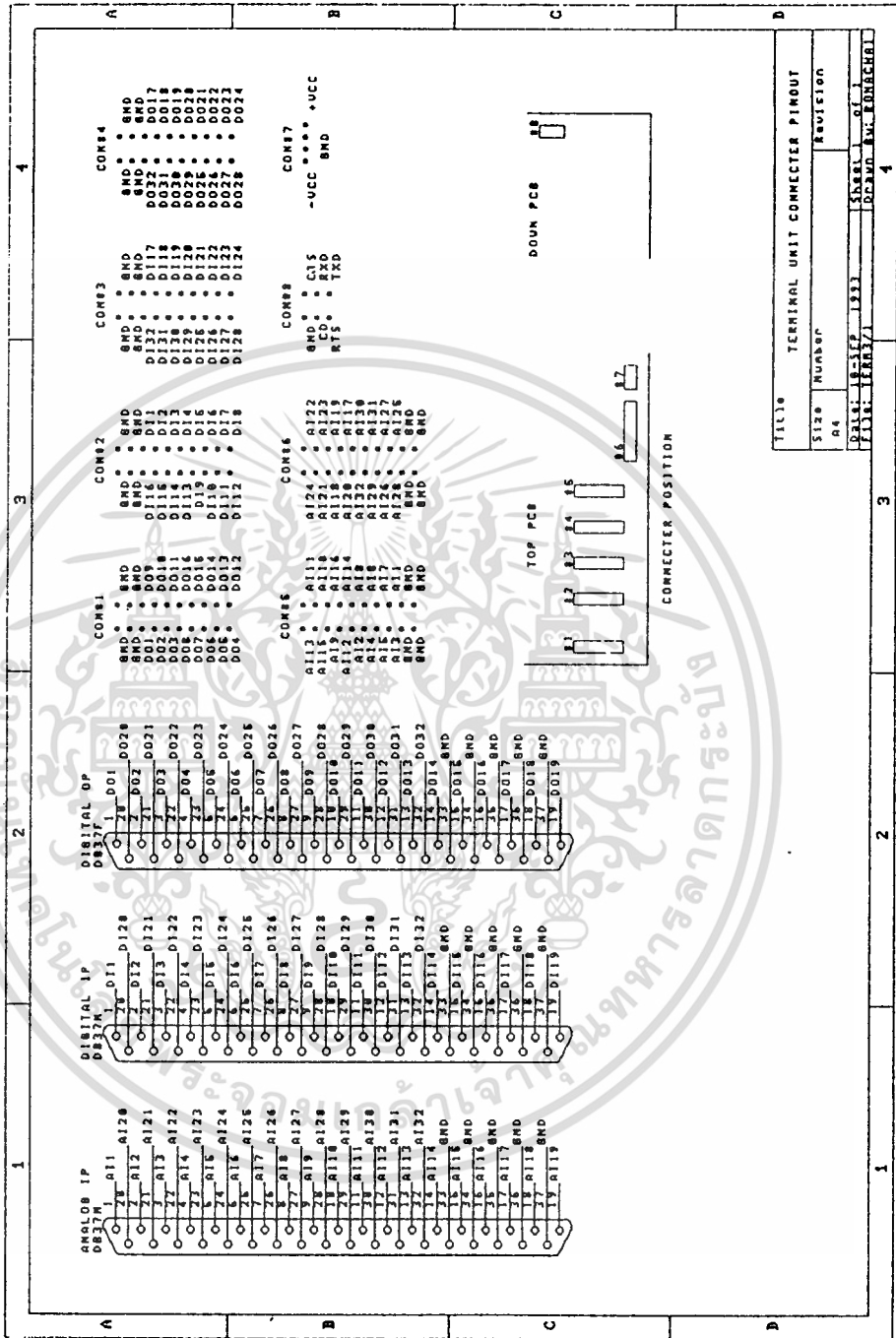




TITLE		
TERMINAL UNIT SCHEMATIC		
SIZE	NUMBER	REVISION
A4	ANALOG INPUT	
DATE: 18-SEP-1983	DESIGNER: S.M.I.-3	REVISION: 0/1
PLATE: 1001/1	IC: 300, BU: 30A300A1	

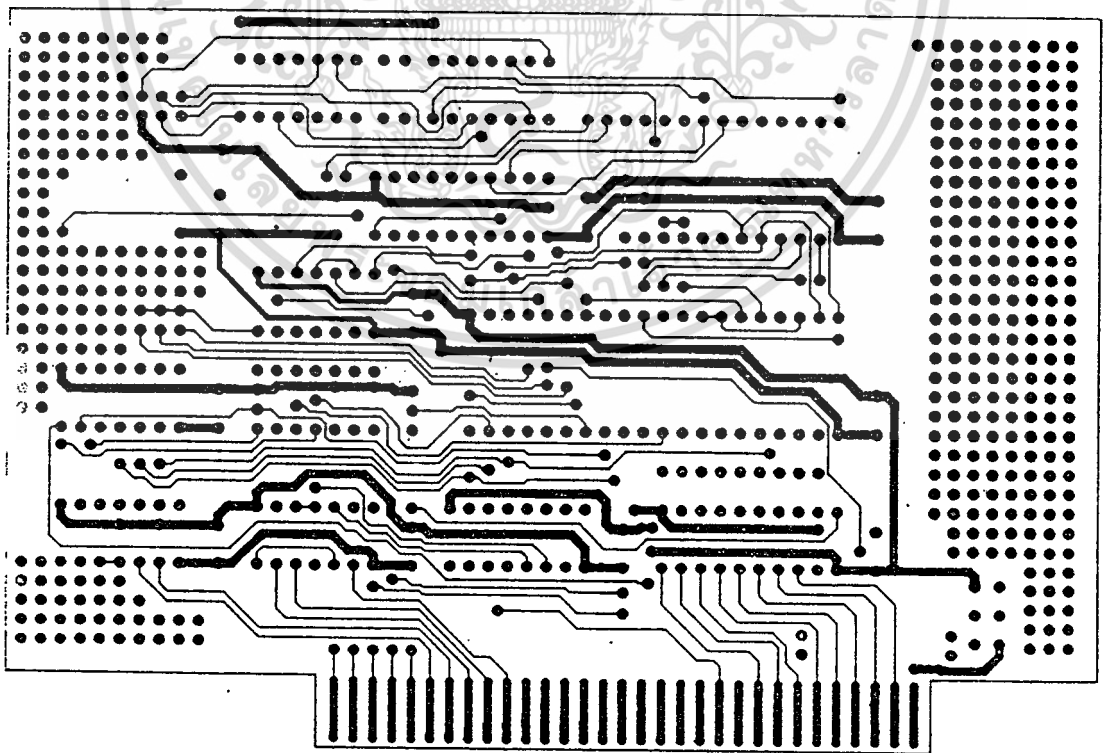
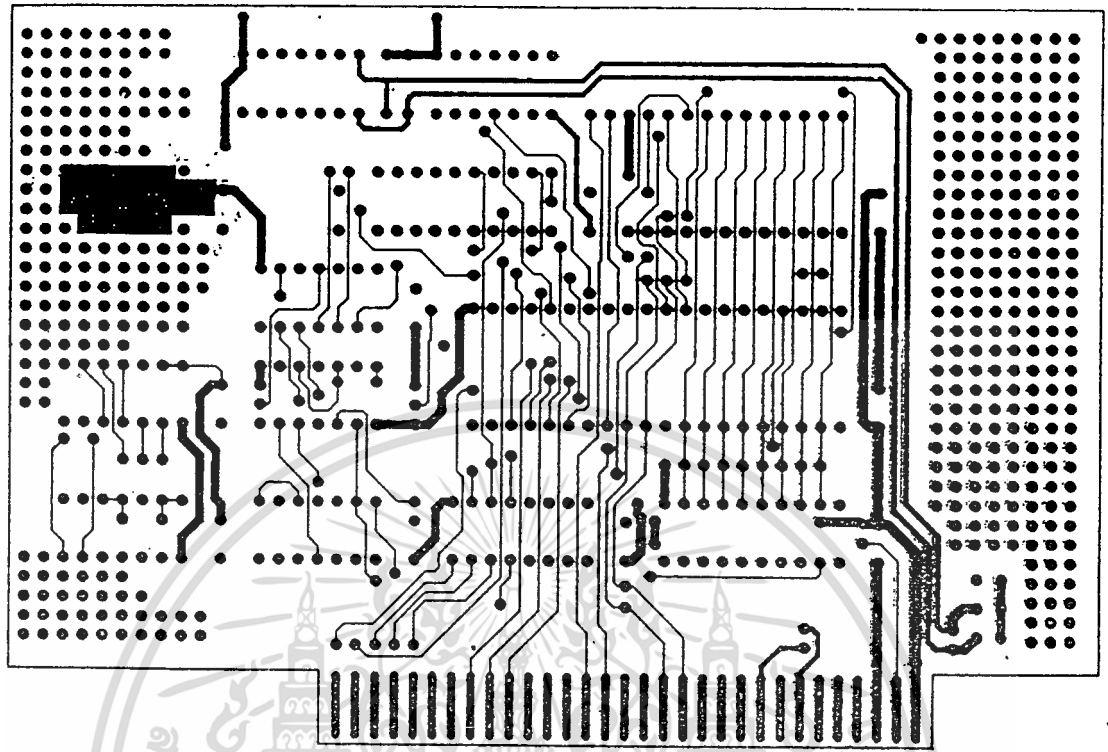
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



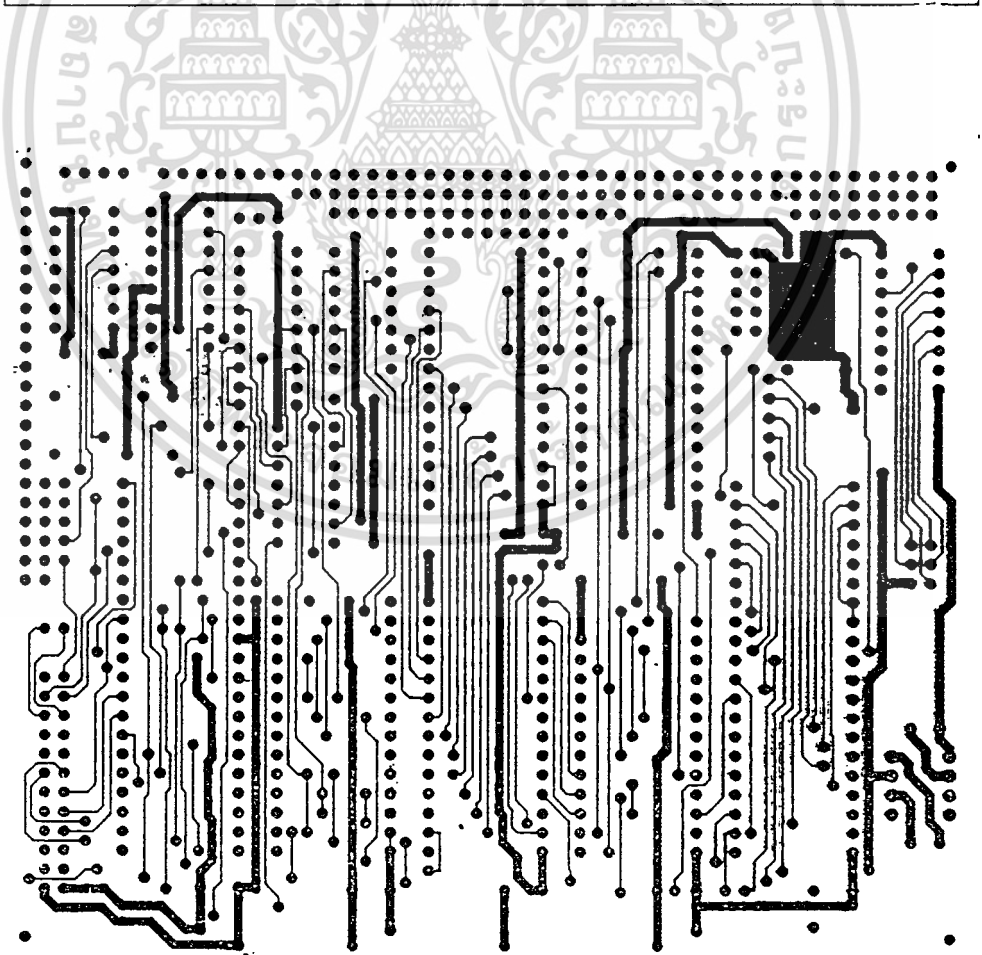
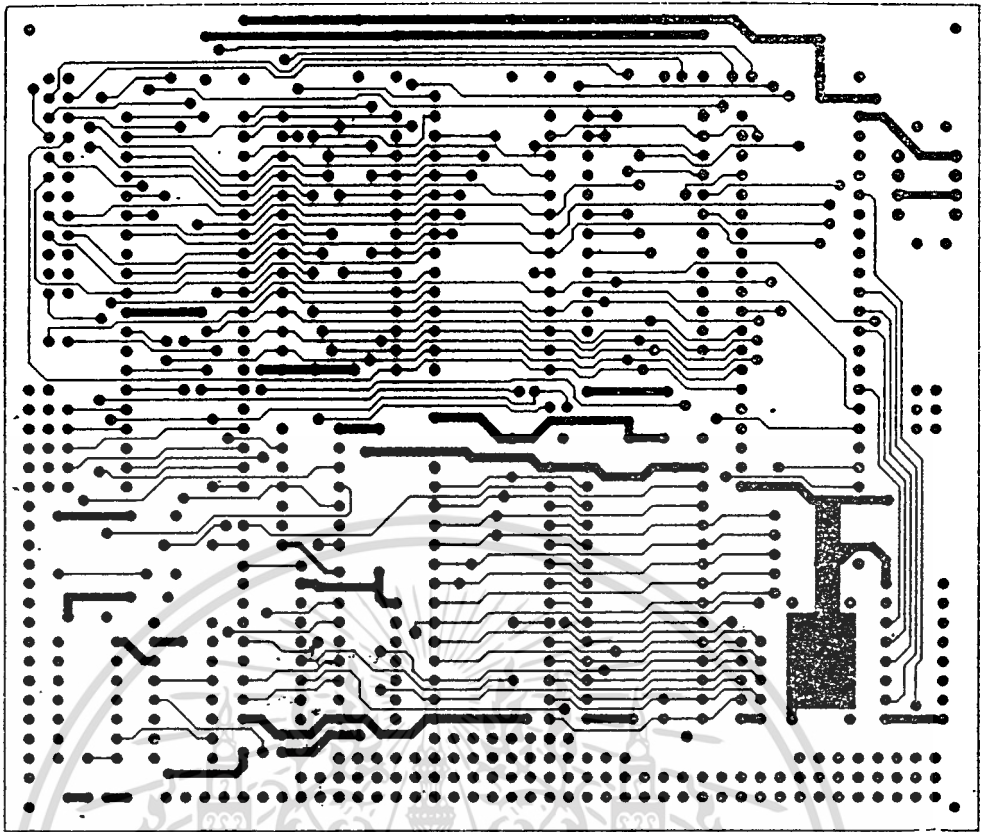


TERMINAL UNIT CONNECTOR PINOUT	
Size	Number
Rev	Revision
DATE: 18-12-1993	Sheet: 1 of 1
FILE: TERA71	DESIGN: ENG. FORACHAI

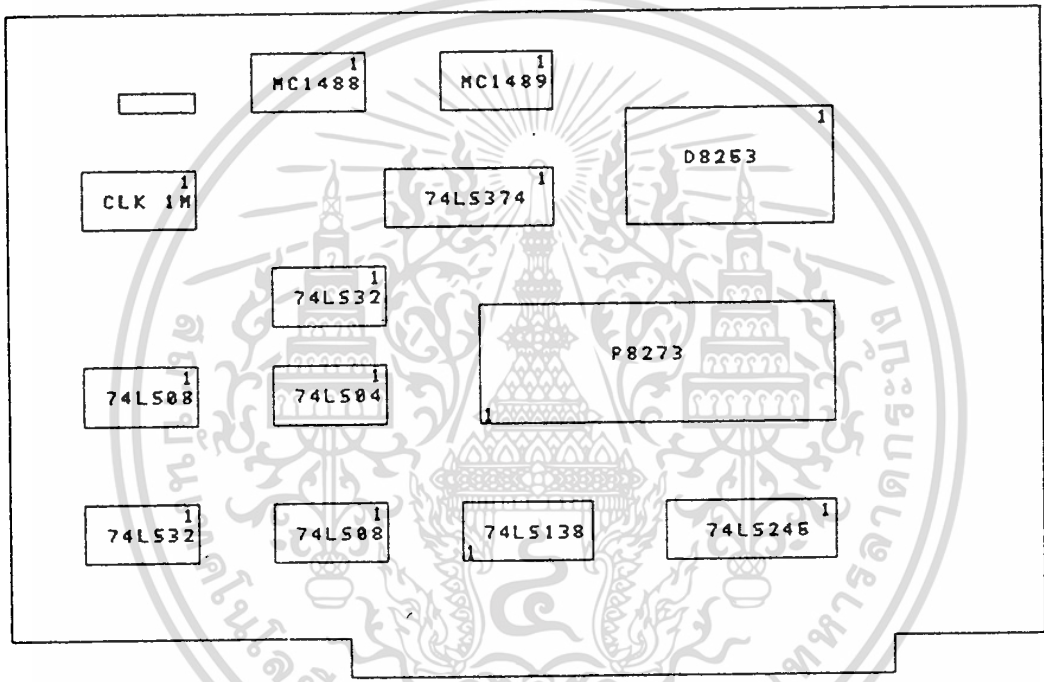
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



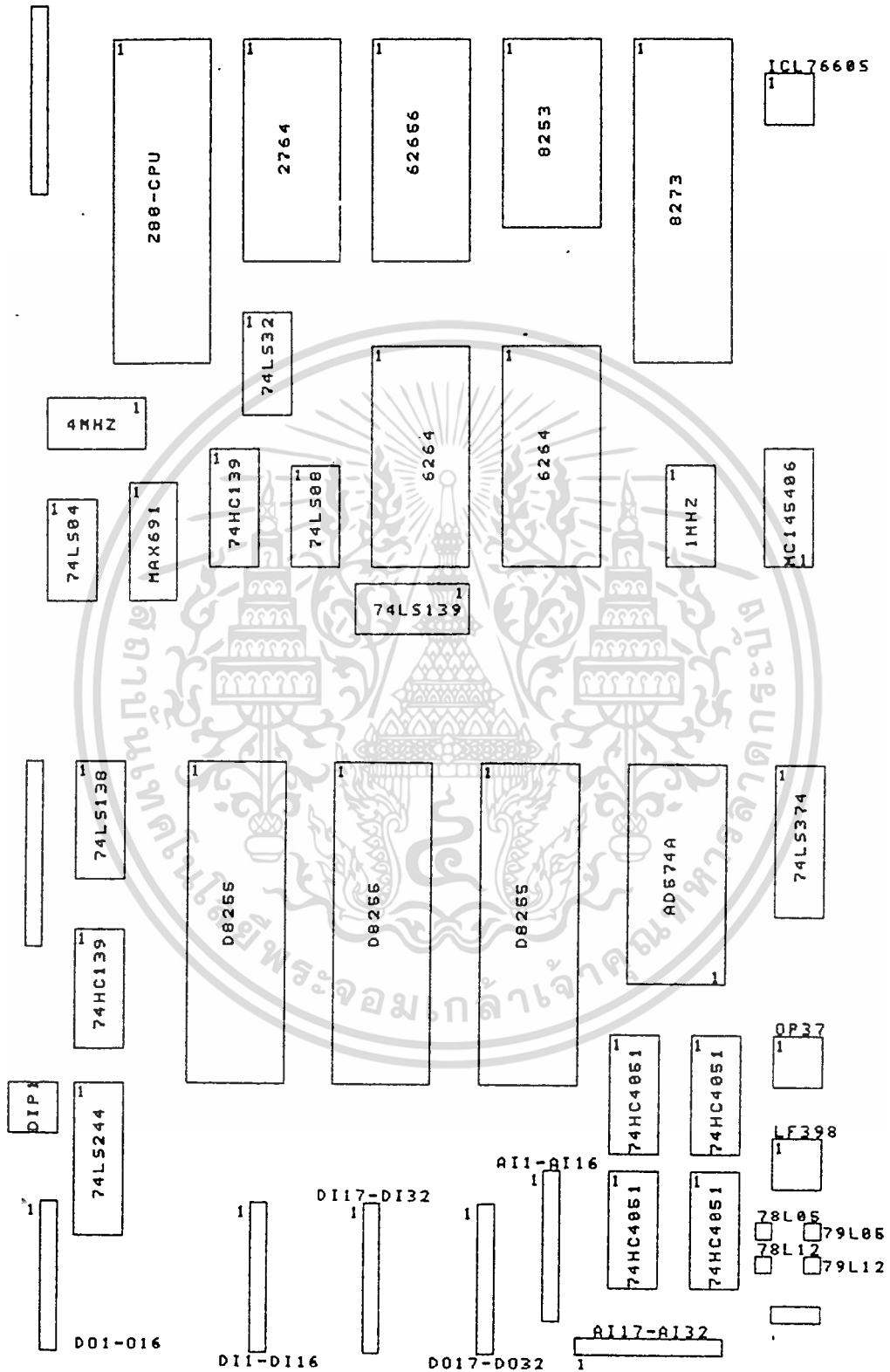
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

- วงจรสมบูร์กซ์ของ COMMUNICATION CARD
- วงจรสมบูร์กซ์ของ TERMINAL UNIT
- แสดงตำแหน่งของ CONNECTER และความหมายของขาต่างๆ
- แสดงลายทองแดงของแผ่นวงจรพิมพ์ของ COMMUNICATION CARD
- แสดงลายทองแดงของแผ่นวงจรพิมพ์ของ TERMINAL UNIT
- แสดงการวางอุปกรณ์บนแผ่นวงจรพิมพ์ของ COMMUNICATION CARD
- แสดงการวางอุปกรณ์บนแผ่นวงจรพิมพ์ของ TERMINAL UNIT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดคุณสมบัติของโครงการไว้ดังนี้คือ Terminal Unit 1 Unit

- มีอินพุตสำหรับวัดสัญญาณทางอนาล็อก (Analog Input) จำนวน 32 อินพุต สามารถรับแรงดันไฟตรงจาก 0 โวลต์ ถึง +5 โวลต์
- ความละเอียดของการแปลงสัญญาณอนาล็อกไปเป็นข้อมูลทางดิจิตอลขนาด 12 บิต
- มีอินพุตสำหรับวัดสัญญาณทางดิจิตอล (Digital Input) จำนวน 32 อินพุต สามารถรับระดับไฟตรงจาก 0 โวลต์ ถึง +5 โวลต์
- มีเอาต์พุตสำหรับสัญญาณทางดิจิตอล (Digital Output) จำนวน 32 เอาต์พุต สามารถจ่ายระดับไฟตรงจาก 0 โวลต์ ถึง +5 โวลต์ โดยสามารถควบคุมสถานะการเปิดปิด โดยผู้ใช้ได้จาก PC

โครงการนี้จะใช้ PC เป็นสถานีหลักในการจัดเก็บข้อมูลจาก TU จึงจำเป็นต้องมีการติดต่อสื่อสารกันระหว่าง PC กับ TU ในช่วงเวลาที่แน่นอน ทั้งนี้เพราะการจัดเก็บข้อมูลลงในดิสก์ จะกระทำในช่วงเวลาที่ผู้ใช้กำหนด จึงเลือกใช้วิธีการติดต่อสื่อสารแบบ ซิงโครนัส (Synchronous Communication) และ โพรโทคอล (Protocol) จะใช้แบบ HDLC (High Level Data Link Control) เนื่องจากมีความถูกต้องของการรับส่งข้อมูลสูง และหา Controller Chip มาสนับสนุนได้ง่าย

สำหรับอุปกรณ์ที่ทำหน้าที่บันทึกข้อมูล จะแบ่งออกเป็น 2 ส่วนคือ

1. อุปกรณ์บันทึกข้อมูลชั่วคราว หรือ Logging Device จะเป็นอุปกรณ์ที่มีความเร็วในการบันทึกข้อมูลสูง เพื่อให้ระบบใช้เวลาสำหรับบันทึกข้อมูลต่ำ เลือกใช้ Ramdisk ของ DOS (Ramdrive) ซึ่งจะใช้เนื้อที่บางส่วนของหน่วยความจำ (Ram) ทำหน้าที่เป็นดิสก์ไดรฟ์ (Disk Drive)

2. อุปกรณ์บันทึกข้อมูลถาวร หรือ Transfer Device เป็นอุปกรณ์บันทึกข้อมูลตามผู้ใช้ที่กำหนดคือ Floppy Disk หรือ Hard Disk จะทำหน้าที่ถ่ายเทข้อมูลจาก Logging Device เมื่อเนื้อที่ของ Logging Device ไม่เพียงพอที่จะใช้เก็บข้อมูล (น้อยกว่า 1 %)

การเลือกใช้ Transfer Device นั้นควรคำนึงถึงเรื่องเวลาที่ใช้ในการเก็บข้อมูล เพราะว่าในขณะที่โปรแกรม ถ่ายเทข้อมูลจาก Logging Device ไปยัง Transfer Device นั้น โปรแกรมจะหยุดการเก็บบันทึกข้อมูลจาก Terminal Unit จนกระทั่งการถ่ายเทข้อมูลเสร็จสิ้นแล้ว จึงจะเริ่มบันทึกข้อมูลใหม่ ถ้าสามารถเลือกใช้ Solidstate Disk เช่น RAMDISK CARD ก็จะช่วยให้ระบบทำงานได้อย่างมีประสิทธิภาพมากขึ้น ทั้งนี้เนื่องจากการทำงานของ Solistate Disk ใช้เวลาในการบันทึกข้อมูลต่ำมาก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**แนวทางการออกแบบ**

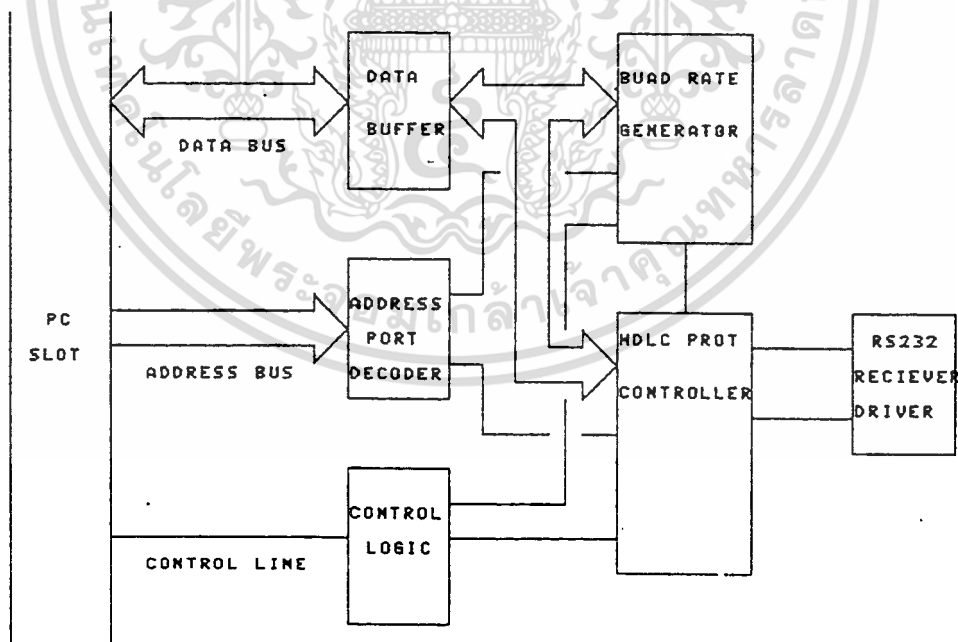
การออกแบบ จะแบ่งฮาร์ดแวร์ของโครงการออกเป็น 2 ส่วน คือ

1. Communication Card เป็นการ์ดสื่อสาร ที่ติดตั้งอยู่ใน PC เพื่อทำหน้าที่สื่อสารข้อมูลกับ TU ใช้โปรแกรมควบคุมการทำงาน

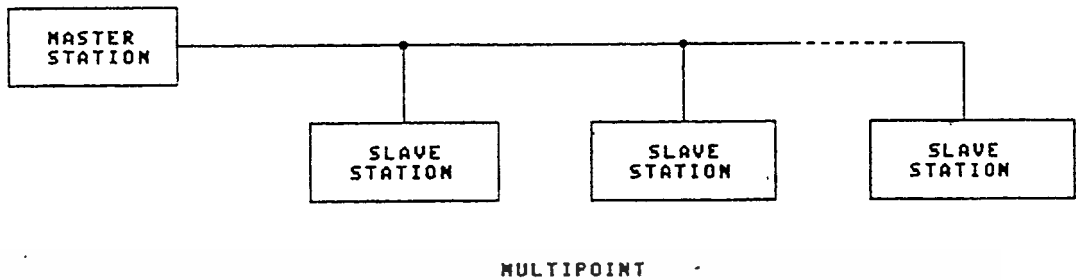
2. Terminal Unit (TU) เป็นระบบไมโครคอมพิวเตอร์ขนาด 8 บิต ที่มีอุปกรณ์ร่วมทำงานในระบบ ดังนี้คือ Analog to Digital Converter Chip 12 Bit, HDLC Protocol Controller Chip , Digital Input/Output Chip ทำหน้าที่เก็บรวบรวมข้อมูลจากจุดวัดต่างๆ และส่งไปให้กับ PC

**COMMUNICATION CARD**

สามารถเขียนแสดงการทำงานเป็นบล็อกไดอะแกรมได้ดังรูปที่ 3.1



รูปที่ 3.1 แสดง บล็อกไดอะแกรมการทำงานของ Communication Card



รูปที่ 2.2 แสดงการจัดเครือข่ายแบบ Multipoint

### 3. Multiple Link เกิดจากการรวมทั้งสองแบบแรกเข้าด้วยกัน

การติดต่อสื่อสารกันระหว่าง Primary และ Secondary Station ผ่านทาง Link Line นั้น จะเป็นการส่งชุดของข้อมูล ที่เป็นรหัสฐานสอง (Binary) ชุดของข้อมูลเหล่านี้ เราจะเรียกว่า เฟรม (Frame) โดยความยาวของเฟรมจะไม่แน่นอน ในแต่ละเฟรมนั้น จะสามารถแบ่งย่อยลงไปได้อีก ส่วนย่อยที่ประกอบกันขึ้นเป็นเฟรมนั้น เรียกว่า ฟิลด์ (Field) ซึ่งในชุดของข้อมูล 1 เฟรม จะประกอบด้วยฟิลด์ต่าง ๆ ดังนี้

- Flag (F) Field ฟิลด์นี้จะมีความยาวเท่ากับ 8 บิตและจะมีรูปแบบที่คงที่คือ ประกอบด้วยลำดับ "01111110" ฟิลด์นี้ใช้สำหรับกำหนดขอบเขตของเฟรมโดยใช้เป็นฟิลด์เริ่มต้นเฟรม (Opening Flag) และฟิลด์สิ้นสุดของเฟรม (Closing Flag) โดยแฟล็กฟิลด์ 1 ฟิลด์ สามารถทำหน้าที่เป็นได้ทั้งฟิลด์สิ้นสุดของเฟรมหนึ่ง และเป็นฟิลด์เริ่มต้นของเฟรมถัดมาได้ และเนื่องจากข้อมูลภายในเฟรมนั้น สามารถที่จะมีรูปแบบของการจัดบิตได้หลากหลาย จึงไม่สามารถรับประกันได้ว่า จะไม่ปรากฏรูปแบบของแฟล็กฟิลด์ ("01111110") ขึ้นภายในเฟรมซึ่งจะทำให้สถานีที่รับเฟรม เข้าใจผิดได้ว่าเป็นฟิลด์สิ้นสุดของเฟรม เพื่อหลีกเลี่ยงปัญหาดังกล่าว จึงต้องใช้กรรมวิธีที่เรียกว่า Bit Stuffing ทำการเปลี่ยนแปลงข้อมูลเพื่อไม่ให้ซ้ำกับแฟล็ก โดยเครื่องส่งจะสอดแทรกบิต "0" พิเศษลงไปข้อมูล ถ้ามีการตรวจพบว่ามีลำดับของบิต "1" จำนวนห้าบิต ("11111") ติดต่อกัน และจะยกเว้นในส่วนที่เป็นแฟล็ก และในภาครับหลังจากที่ได้ตรวจพบ Opening Flag แล้ว ก็จะคอยมองกลุ่มของบิตที่เป็นลำดับของบิต 1 จำนวนห้าบิต

ติดต่อกัน ("11111") ถ้าพบก็จะตรวจสอบบิตที่ หก ถ้าบิตที่หกเป็น "0" ก็จะลบบิตที่หกทิ้ง แต่ถ้าบิตที่หกเป็น "1" และบิตที่เจ็ด เป็น "0" ก็จะมองข้อมูลชุดนั้นว่าเป็นแฟล็กฟิลด์ และถ้าทั้งบิตที่หก และเจ็ด เป็น "1" ทั้งคู่ ก็จะมองว่า Abort Character

- Address (A) Field โดยปกติแล้ว ฟิลด์นี้将有ความยาวเท่ากับ 8 บิต แต่ก็สามารถขยายออกไปได้ (Extended) เป็นจำนวนเท่าของความยาว 7 บิต โดยบิตที่แปดในแต่ละฟิลด์ที่เพิ่มขึ้น จะเป็น "0" หรือ "1" ก็ได้ แต่กำหนดว่าถ้าบิตที่แปด เป็น "1" จะบอกให้รู้ว่าฟิลด์นี้เป็นฟิลด์สุดท้ายของ Address Field ข้อมูลในฟิลด์นี้ จะใช้เป็นตัวบ่งบอก หรือ แยกแยะ Secondary Station ที่ทำการส่ง หรือรับเฟรมในการติดต่อ กล่าวคือเมื่อ Primary Station ต้องการติดต่อกับ Secondary Station ใดก็จะกำหนดให้ Address Field มีข้อมูลตรงกับ Address Field ของ Station นั้น และเมื่อ Secondary Station ตอบรับกลับมา ก็จะกำหนดให้ข้อมูลใน Address Field เป็นของตนเอง และถ้าข้อมูลในฟิลด์นี้เป็น "1" ทั้งหมด ก็จะหมายถึง Secondary Station ทุกตัวในเครือข่าย (Global Station)

- Control (C) Field ข้อมูลในฟิลด์นี้จะเป็นตัวกำหนดชนิดของเฟรมซึ่งมีอยู่ 3 ชนิด คือ Information Frame (I-Frame) ใช้สำหรับการแลกเปลี่ยนข่าวสารข้อมูล, Supervisory Frame (S-Frame) ใช้สำหรับควบคุมการสื่อสารของอุปกรณ์ต่างๆในระบบ และ Unnumbered Frame (U-Frame) เป็นเฟรมพิเศษ ตามแต่จะกำหนดในการใช้ ความยาวของฟิลด์นี้มีขนาด 8 บิต

- Data หรือ Information (I) Field ฟิลด์นี้จะปรากฏขึ้นเฉพาะในเฟรมชนิด I-Frame เท่านั้น จะประกอบด้วยชุดของข้อมูลตามแต่ความต้องการ มีความยาวได้ไม่จำกัด โดยปกติแล้ว จะมีความยาวเป็นจำนวนเท่าของความยาว 8 บิต

- Frame Check Sequence (FCS) Field เนื่องจากความยาว และรูปแบบของฟิลด์ข้อมูลไม่แน่นอน และไม่จำกัดรูปแบบ ดังนั้นจึงจำเป็นต้องมีการตรวจสอบความถูกต้องในการรับ และส่งข้อมูล ข้อมูลในฟิลด์นี้ จะเป็นข้อมูลที่ภาคส่งได้กำเนิดขึ้นหลังจากที่ได้ส่งข้อมูลชุดนั้น ออกไปแล้ว ซึ่งจะมีความสัมพันธ์กับข้อมูล ภาครับก็จะนำข้อมูลในฟิลด์นี้ มาตรวจสอบ

กับข้อมูลที่รับเข้ามา ด้วยความสัมพันธ์แบบเดียวกับภาคส่ง ว่าข้อมูลถูกต้องหรือไม่ พิลด์นี้มีความยาว 16 บิต

ส่วนประกอบของฟิลด์ต่างๆของเฟรมแสดงไว้ดังรูปที่ 2.3

FLAG	ADDRESS	CONTROL	INFORMATION	FCS1	FCS2	FLAG

HDLC PROTOCOL FRAME FOMAT

รูปที่ 2.3 แสดงส่วนประกอบของ เฟรม (Frame)

#### HDLC PROTOCOL CONTROLLER

8273 ของบริษัท Intel เป็นอุปกรณ์ที่ถูกผลิตขึ้นเพื่อสนับสนุนการติดต่อสื่อสารตามมาตรฐานของ ISO/CCITT ภายใต้ข้อตกลง HDLC Protocol สามารถใช้งานติดต่อสื่อสารได้ทั้งแบบ Synchronous และแบบ Asynchronous ภายในประกอบด้วยรีจิสเตอร์ที่สำคัญ 7 ตัวคือ

- Command ใช้สำหรับรับคำสั่งเพื่อกำหนดการทำงาน
- Parameter ใช้สำหรับรับพารามิเตอร์ของคำสั่งบางคำสั่ง
- Result เก็บผลลัพธ์ของการทำตามคำสั่ง
- Transmit Interrupt Result เก็บผลลัพธ์ของการส่งข้อมูล
- Recieve Interrupt Result เก็บผลลัพธ์ของการรับข้อมูล
- Status บอกสถานะต่างๆในการเชื่อมต่อกับ CPU

การทำงานของ 8273 จะประกอบด้วย

- Command Phase เป็นช่วงที่ CPU เขียนคำสั่งและพารามิเตอร์ให้กับ 8273 ลงใน Command และ Parameter Register
- Execution Phase เป็นช่วงที่ 8273 ทำตามคำสั่งที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- Result Phase เป็นช่วงที่ 8273 บอกให้ CPU รู้ว่าทำตามคำสั่งเสร็จสิ้นแล้ว เพื่อที่จะได้อ่านผลของการทำคำสั่งจากรีจิสเตอร์ของ 8273

#### คำสั่งที่สำคัญที่ใช้งาน

- Frame Transmit ใช้สำหรับส่งเฟรมของข้อมูลออกไปทาง Txd คำสั่งนี้จะต้องมีพารามิเตอร์ประกอบด้วยคำสั่ง 4 พารามิเตอร์ คือ แอดเดรสไบต์ คอนโทรลไบต์ ความยาวของเฟรมข้อมูลที่ต้องการส่ง (ไบต์ต่ำ) ความยาวของเฟรมข้อมูลที่ต้องการส่ง (ไบต์สูง)

- Selective Recive ใช้สำหรับรับเฟรมของข้อมูลจาก Station ที่กำหนด มีพารามิเตอร์ประกอบ 4 ตัว คือ แอดเดรสไบต์ คอนโทรลไบต์ ความยาวของเฟรมข้อมูลที่ต้องการรับ (ไบต์ต่ำ) ความยาวของเฟรมข้อมูลที่ต้องการรับ (ไบต์สูง)

**แนวทางการออกแบบ**

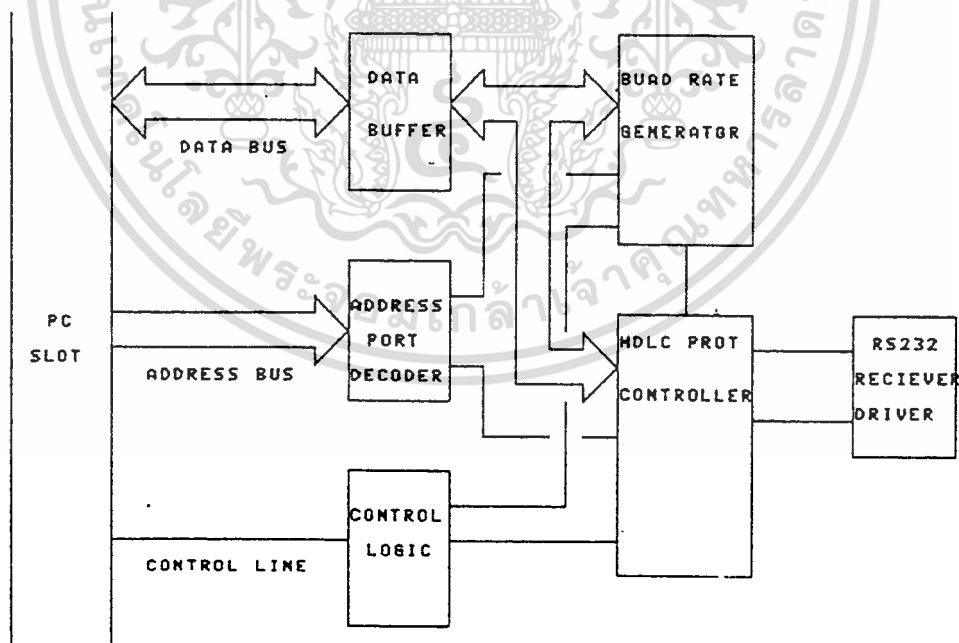
การออกแบบ จะแบ่งฮาร์ดแวร์ของโครงการออกเป็น 2 ส่วน คือ

1. Communication Card เป็นการ์ดสื่อสาร ที่ติดตั้งอยู่ใน PC เพื่อทำหน้าที่สื่อสารข้อมูลกับ TU ใช้โปรแกรมควบคุมการทำงาน

2. Terminal Unit (TU) เป็นระบบไมโครคอมพิวเตอร์ขนาด 8 บิต ที่มีอุปกรณ์ร่วมทำงานในระบบ ดังนี้คือ Analog to Digital Converter Chip 12 Bit, HDLC Protocol Controller Chip , Digital Input/Output Chip ทำหน้าที่เก็บรวบรวมข้อมูลจากจุดวัดต่างๆ และส่งไปให้กับ PC

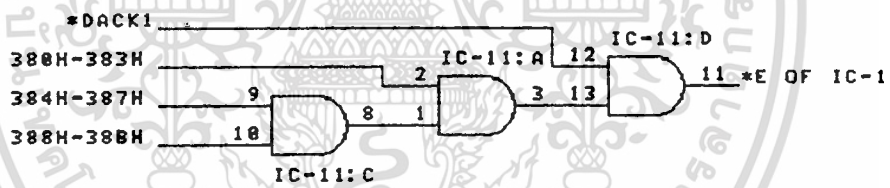
**COMMUNICATION CARD**

สามารถเขียนแสดงการทำงานเป็นบล็อกไดอะแกรมได้ดังรูปที่ 3.1



รูปที่ 3.1 แสดง บล็อกไดอะแกรมการทำงานของ Communication Card

การทำงานของวงจรเป็นดังนี้ 74LS245 (IC-1) ทำหน้าที่เป็น Data Buffer ระหว่าง Data Bus ของ PC กับ Data Bus ของการ์ด เพื่อป้องกันสัญญาณข้อมูลบางส่วนที่ไม่เกี่ยวข้องกับการ์ดนี้ ครอบคลุมการทำงานของอุปกรณ์ภายในการ์ด และทำงานเป็น Buffer แบบสองทาง (Bidirectional Buffer) โดยมีขา Dir (ขาที่ 1) เป็นตัวกำหนดทิศทางของข้อมูลที่จะผ่าน โดยใช้สัญญาณ \*IOR ผ่าน Inverter Gate IC-10:D เป็นสัญญาณควบคุม คือเมื่อ \*IOR แอคทีฟ (เป็น Low) จะเป็นการอ่านข้อมูลของ CPU จากอุปกรณ์ I/O ก็จะทำให้ลอจิกที่ขา Dir เป็น High 74LS245 จะทำหน้าที่เป็น Input Buffer รับข้อมูลจากทางด้าน A1 - A8 ผ่านไปยังด้าน B1 - B8 และในขณะที่ \*IOR ไม่แอคทีฟ ก็จะทำหน้าที่เป็น Output Buffer เพื่อเขียนข้อมูลในการ์ด ส่วนขา \*E (Enable) ได้จากการ AND ของสัญญาณ Address Port Select ทั้งหมดที่ใช้งานในการ์ดนี้ด้วย And Gate IC-11:A , IC-11:C และ IC-11:D ดังแสดงในรูปที่ 3.2

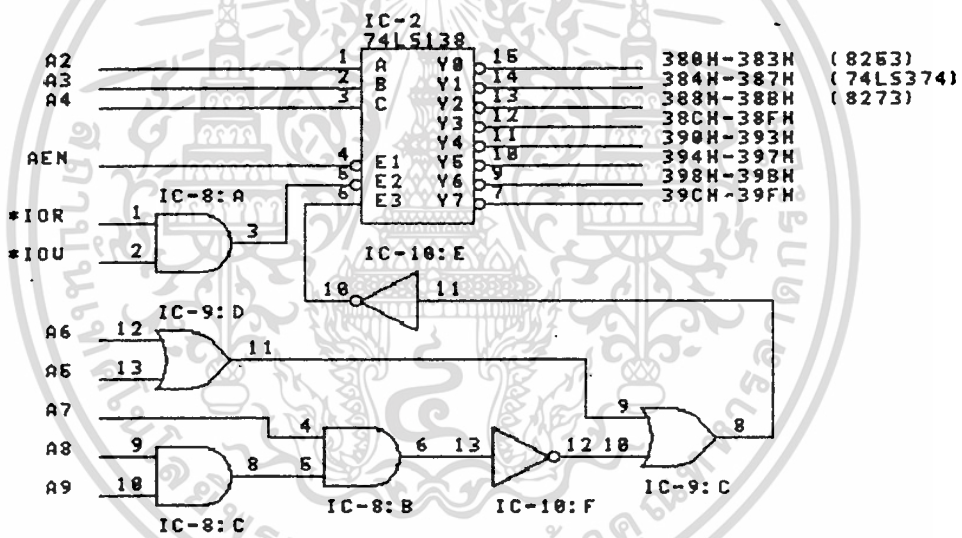


รูปที่ 3.2 แสดงวงจรในส่วนของ Address Select Enable

ส่วนต่อมาก็คือ 74LS138 (IC-2) ทำงานร่วมกับ And Gate IC-8:A , IC-8:B, IC-8:C, OR Gate IC-9:C, IC-9:D และ Inverter Gate IC-10:E, IC-10:F ทำหน้าที่เป็น Address Port Decoder กำหนดให้ Address ของการ์ดนี้อยู่ระหว่าง 380h ถึง 39Fh โดยมีวิธีการถอดรหัสดังนี้

ADDRESS	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
380h	1	1	1	0	0	0	0	0	0	0
39Fh	1	1	1	0	0	1	1	1	1	1

สัญญาณ A1 และ A0 จะไม่นำมาถอดรหัสด้วย เพราะจะต้องนำไปใช้สำหรับถอดรหัสภายในของ 8253 และ 8273 จะเห็นได้ว่า ในช่วง ADDRESS ที่ต้องการ สัญญาณ A9, A8 และ A7 จะเป็น "1" ทั้งหมด จึงนำสัญญาณดังกล่าวมา AND กัน ส่วน A6 และ A5 จะเป็น "0" จึงนำมา OR กัน สัญญาณที่เหลือ คือ A4, A3 และ A2 จะมีการเปลี่ยนแปลงจาก "000" ไปจนถึง "111" นำไปถอดรหัสที่ IC-2 ได้สัญญาณ Address Select ที่ขา Y0 ถึง Y7 เป็น 380h-383h, 384h-387h, 388h-38Bh, ..... 38Ch-39h, ตามลำดับ ที่ \*E2 (ขาที่ 5) จะเป็นตัวกำหนดให้ถอดรหัส Address เฉพาะที่เกี่ยวข้องกับการเขียนหรืออ่านข้อมูลกับ Input/Output เท่านั้นดังแสดงในรูปที่ 3.3

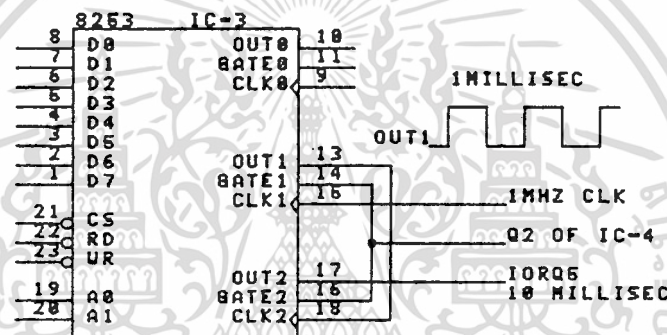


รูปที่ 3.3 แสดงวงจรในส่วนของ Address Port Decoder

8253 Counter/Timer (IC-3) ทำหน้าที่กำหนดสัญญาณนาฬิกาที่ใช้สำหรับกำหนดจังหวะการรับและส่งข้อมูล (Buad Rate Generater) โดยใช้ Counter ใน Channel 0 หารความถี่จาก Clock Oscillator 1 Mhz และถูกโปรแกรมให้ทำงานในโหมด 3 ส่วนอีกสองแชนแนลที่เหลือ จะทำหน้าที่กำเนิดสัญญาณเวลาคงที่ที่ 50 mS โดย Counter Channel 1 โปรแกรมให้ทำงานในโหมด 3 หารความถี่จาก Clock Oscillator 1 Mhz เอาท์พุทที่ได้จาก Channel 1 (Q1) ซึ่งมีคาบเวลาเท่ากับ 1 mS และจะป้อนต่อไปให้กับ Channel 2 ที่โปร

-แกรมให้ทำงานในโหมด 0 ที่เอาพุทของ Channel 2 (Q2) จะมีคาบเวลาเท่ากับ 10 mS และต่อเข้ากับสัญญาณ IORQ 5 ของ PC เพื่อใช้เป็นตัวกำหนดจังหวะและช่วงเวลาที่ใช้ติดต่อสื่อ -สารระหว่าง PC กับ TU โดย TU แต่ละตัวนั้นจะใช้เวลาในการรับ และส่งข้อมูลทั้งหมดรวม 500 mS

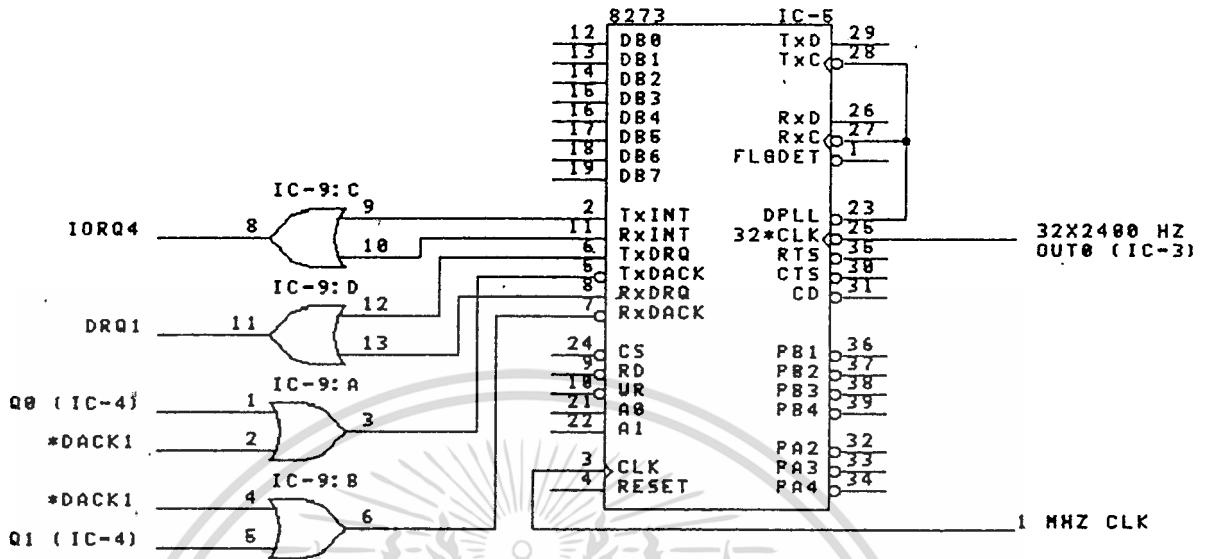
การกำเนิดสัญญาณ 10 mS นี้จะถูกควบคุม ด้วยสัญญาณที่ขา G1 และ G2 ซึ่งได้มาจาก Q2 ของ IC-4 Address Port ของ 8253 นี้จะกำหนดไว้ที่ 380h-383h ดังแสดงใน รูปที่ 3.4



รูปที่ 3.4 แสดงวงจรในส่วนของการกำเนิดสัญญาณเวลา 10 mS

74LS374 (IC-4) ทำหน้าที่เป็น Latch Output Port กำหนดไว้ที่ Address 384h-387h เป็น Control Port สำหรับควบคุม IC-3 และ IC-5 บิต D0 ทำหน้าที่ควบคุม การส่งผ่านข้อมูลทางด้านการส่ง, บิต D1 ทำหน้าที่ ควบคุมการส่งผ่านข้อมูลด้านการรับ ของ IC-5 และบิต D2 ทำหน้าที่ควบคุมการกำเนิดสัญญาณ 10 mS ของ IC-3

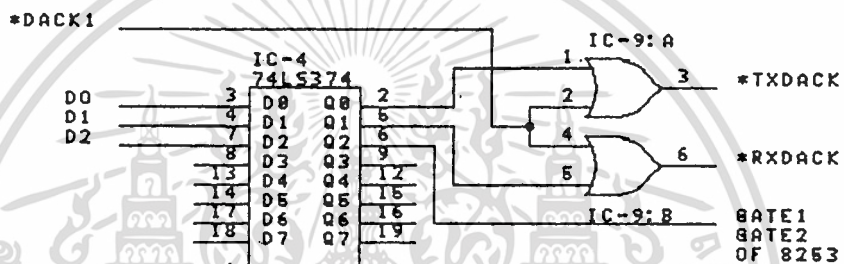
8273 (IC-5) ทำหน้าที่ เป็นส่วนควบคุมการติดต่อสื่อสารทั้งหมดของระบบกำหนด ไว้ที่ Address 388h-38Bh และจะถูกโปรแกรมให้กรรมวิธีการรับส่งข้อมูลระหว่าง CPU กับ 8273 เป็นแบบ DMA (Direct Memory Access) ผ่านทาง DMA Request Channel 1 (DRQ1) ของ PC และการส่งผ่านผลของการรับส่งข้อมูลเป็นแบบ Interrupt ผ่านทาง Inter-rupt Request Channel 4 (IORQ4) ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 แสดงการเชื่อมต่อระหว่าง CPU กับ 8273

การทำงานจะเป็นดังนี้ เมื่อต้องการส่งเฟรมของข้อมูลทำได้โดย CPU จะเขียนคำสั่ง (Command) ให้กับรีจิสเตอร์คำสั่ง (Command Register) พร้อมด้วยกับพารามิเตอร์ให้กับ Parameter Register หลังจากที่ได้รับคำสั่งพร้อมพารามิเตอร์ ครบแล้ว 8273 ก็จะเริ่มทำตามคำสั่ง โดยจะทำให้ขา \*RTS แอ็คทีฟ และรอจนกระทั่งลอจิกที่ขา \*CTS แอ็คทีฟเมื่อแอ็คทีฟแล้ว ก็จะเริ่มต้นส่งแฟล็กนิลด์ (F) ตามด้วยแอดเดรสนิลด์ (A) และ คอนโทรลนิลด์ (C) และในขณะที่กำลังส่งคอนโทรลนิลด์อยู่นั้น ก็จะขอข้อมูลจาก CPU ด้วยการทำให้สัญญาณ TXDRQ แอ็คทีฟ เป็นการขอติดต่อกับหน่วยความจำผ่าน 8257 DMA Controller ในแชนแนลที่ 1 8257 จะอ่านข้อมูลจากหน่วยความจำในบริเวณที่ได้โปรแกรมไว้เป็น Buffer ของนิลด์ข้อมูล และเขียนให้กับ 8273 ครั้งละไบต์ตามที่ 8273 ร้องขอจนหมดหรือครบความยาว ของนิลด์ข้อมูล ด้วยการทำให้สัญญาณ \*DACK แอ็คทีฟ สัญญาณนี้จะผ่านชุด Enable Gate IC-9:A, IC-9:B ให้ปรากฏขึ้นเฉพาะที่ขา \*TXDACK เท่านั้น 8273 ก็จะรู้ว่า CPU ส่งผ่านข้อมูลที่ ร้องขอ และนำข้อมูลดังกล่าว ส่งต่อจากคอนโทรลนิลด์ครั้งละไบต์ต่อเนื่องกันไปจนจบนิลด์ และในระหว่างที่รับข้อมูลแต่ละไบต์นี้ 8273 ก็จะเริ่มคำนวณค่าของ FCS Field ของข้อมูลทั้งหมดในนิลด์ พร้อม

ทั้งส่ง FCS Field ตามฟิลด์ของข้อมูลไป เมื่อส่ง FCS Field ครบความยาว แล้วก็เริ่มส่ง แพลกฟิลด์ พร้อมทั้งทำให้ลอจิกที่ขา TXINT แอ็คทีฟ เพื่อไปอินเทอร์รัพ (Interrupt) CPU เมื่อได้รับการอินเทอร์รัพ CPU ก็จะอ่านผลของการส่ง (TX Result) จากรีจิสเตอร์ภายใน 8273 (Result Register) เพื่อดูผลของการส่ง ว่ามีข้อผิดพลาดหรือไม่ และ ในด้านการรับข้อมูล ก็จะเป็นไปในลักษณะเดียวกัน แต่จะเปลี่ยนสัญญาณที่แอ็คทีฟเป็น \*RXDACK และ RXINT แทน และ 8257 จะทำงานในโหมด การเขียนข้อมูลลงหน่วยความจำแทน ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 แสดงวงจรในส่วนของ DMA Acknowledge Enable

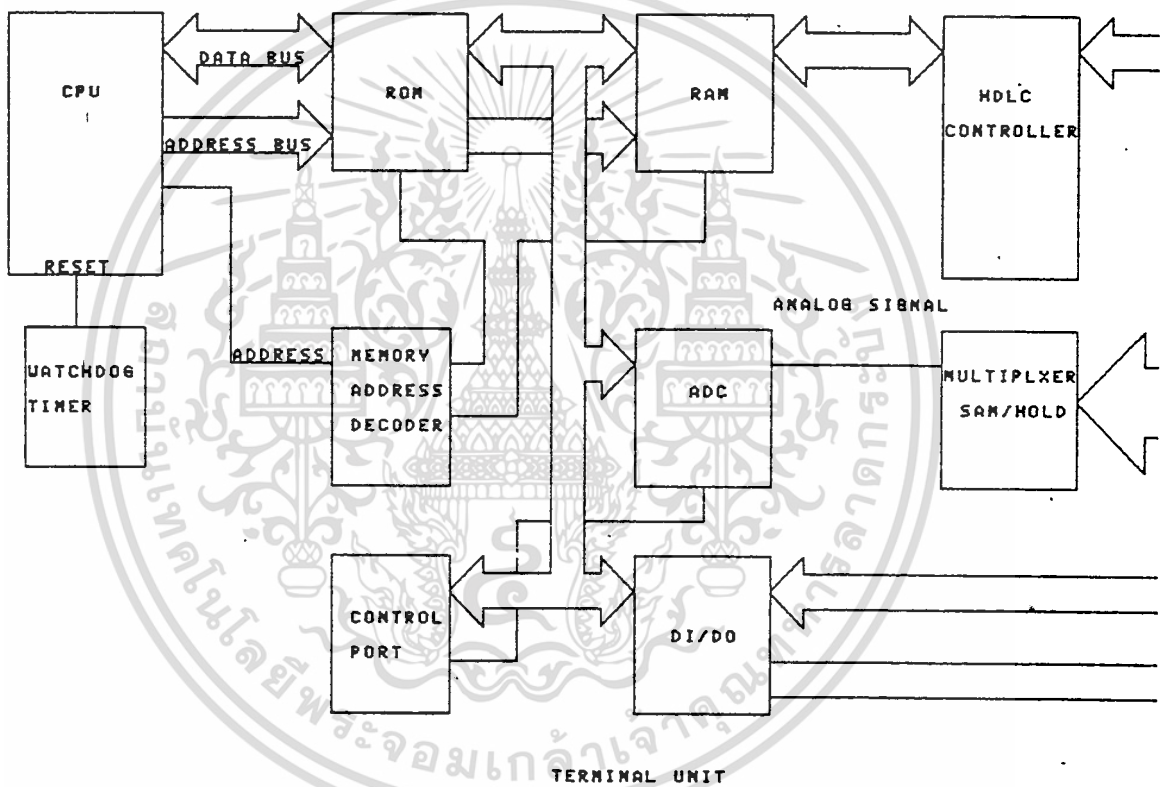
สัญญาณของข้อมูลส่งและรับ (TXD, RXD) รวมทั้งสัญญาณที่จำเป็นต้องใช้ในการควบคุมการส่งและรับ (RTS , CTS , CD) จะบ่อนให้กับ RS-232 Reciever-Driver เพื่อเปลี่ยนระดับแรงดันที่ MC1488 , MC1489 และส่งต่อให้กับ DB-25 Connector สัญญาณนี้จะนำไปใช้ติดต่อสื่อสารโดยตรงผ่าน Leasedline และจำเป็นต้องต่อผ่าน Relay Switch หรือต่อเข้า MODEM เพื่อติดต่อกับ TU ที่อยู่ไกลๆได้ วงจรสมบูรณ์แสดงในภาคผนวก ก

**TERMINAL UNIT**

สามารถแสดงการทำงานเป็นบล็อกไดอะแกรมในรูปที่ 3.7

Z80CPU (IC-1) ทำงานเป็นไมโครโปรเซสเซอร์ที่ความถี่นาฬิกา 4 Mhz จาก Clock Oscillator ในส่วนของหน่วยความจำ (Memory) ประกอบด้วย ROM ขนาด 8KB 2764 (IC-2) เก็บโปรแกรมควบคุมการทำงาน , RAM ขนาด 8KB 6264 (IC-4) และ

ขนาด 32 KB 1 ตัว 62256 (IC-3) เนื้อที่ของ RAM ในส่วนต้นๆ จะทำหน้าที่เป็น System Area และ Stack Area ROM จะกำหนดไว้ที่แอดเดรส 0000h - 3FFFh และ RAM จะกำหนดไว้ที่แอดเดรส 4000h- 5FFFh , 6000h - 7FFFh และ 8000h - FFFFh ตามลำดับ MAX691 (IC-6) ทำหน้าที่เป็น Watch Dog Timer ด้วยช่วงเวลาการรีเซ็ต เท่ากับ

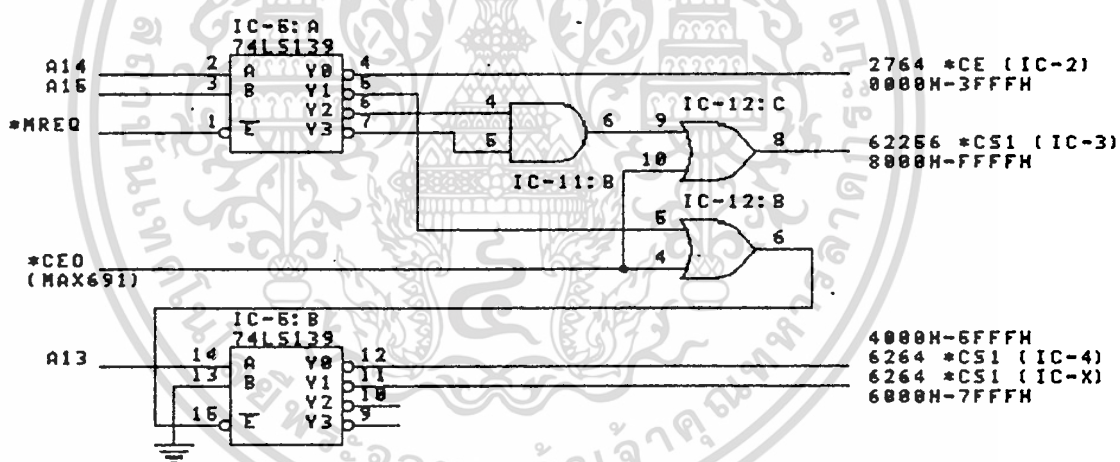


รูปที่ 3.7 แสดงบล็อกไดอะแกรมของ Terminal Unit

1.6 วินาที เพื่อป้องกัน CPU หลุดออกจากโปรแกรม พร้อมกันนั้นก็ทำหน้าที่ตรวจสอบระดับไฟเลี้ยงวงจรผ่านทาง TrimPot ซึ่งจัดเป็นวงจร Divider บ่อนแรงดันให้กับขา PFI วงจรคอมพิวเตอร์ภายใน MAX691 จะเปรียบเทียบแรงดันที่ขา PFI กับแรงดันคงที่ 4.65V ถ้า

แรงดันที่ขา PFI ต่ำกว่า 4.65V ก็จะทำให้ระดับลอจิกที่ขา \*PFO แอ็คทีฟ (LOW) สัญญาณที่ขา นี้จะนำไปควบคุมการเขียนและอ่านข้อมูลกับ RAM ร่วมกับ OR Gate IC-12:B, IC-12:D

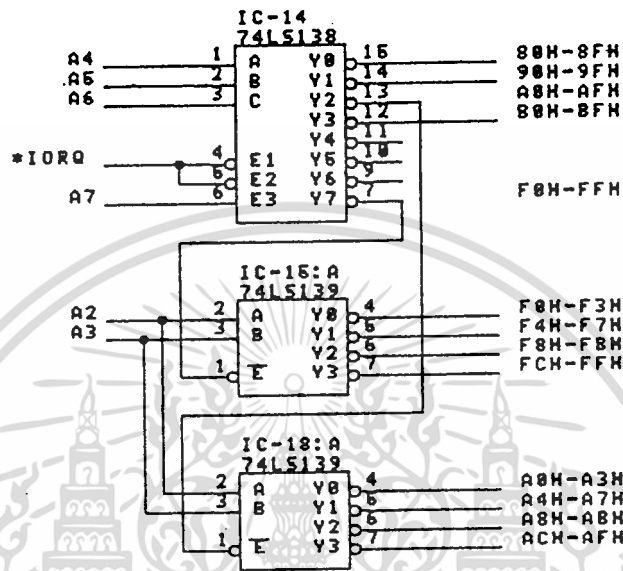
ส่วนต่อมาคือ Address Decoder ประกอบด้วย 74LS139 IC-5:A, IC-5:B, 74LS08 (IC-11:B) และ 74LS32 IC-12:B ทำงานร่วมกันเป็น Memory Address Decoder ถอดรหัสออกเป็นสัญญาณเลือก 4 เส้น โดยแต่ละเส้นจะควบคุมขนาดของหน่วยความจำเท่ากับ 16KB คือที่ขา Y0-Y3 เป็น 0000h - 3FFFh , 4000h - 7FFFh , 8000h - BFFFh และ C000h - FFFFh ตามลำดับ สัญญาณจาก Y2 และ Y3 จะนำไปผ่าน AND Gate IC-11:B เพื่อนำไปควบคุม RAM 62256 ซึ่งมีขนาด 32 KB และสัญญาณที่ขา Y1 จะนำไปถอดรหัสแยกออกเป็น 2 เส้นที่ 74LS139 IC-5:B เพื่อนำไปควบคุม RAM 6264 สองตัว แสดงในรูปที่ 3.8



รูปที่ 3.8 แสดงวงจรในส่วนของ Memory Address Decoder

ต่อมาคือ 74LS138 IC-14 ทำหน้าที่เป็น I/O Address Decoder ถอดรหัสเป็นสัญญาณเลือกจำนวน 8 เส้น เริ่มต้นที่แอดเดรส 80h ไปจนถึง FFh และสัญญาณเลือกที่แอดเดรส F0h - FFh จะนำไปถอดรหัสต่อไปอีกที่ 74LS139 IC-15:A ได้สัญญาณเลือก 4 เส้น คือที่แอดเดรส F0h - F3h , F4h - F7h , F8h - FBh และ FCh - FFh ตามลำดับ และสัญญาณเลือกที่แอดเดรส A0h - AFh จะนำไปถอดรหัสต่อไปอีกที่ 74LS139

IC-18:A ได้สัญญาณเลือก 4 เส้น คือที่แอดเดรส A0h - A3h , A4h - A7h , A8h - ABh และ ACh - AFh ตามลำดับ แสดงในรูปที่ 3.9

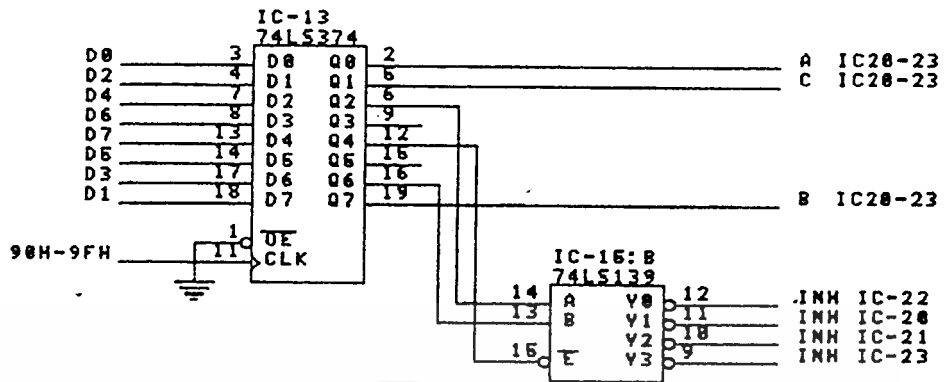


รูปที่ 3.9 แสดงวงจรในส่วนของ Port Address Decoder

74HC4051 IC-20,21,22,23 จำนวน 4 ตัว ทำหน้าที่เป็น Analog Multiplexer 32 แชลแนล รับแรงดันในช่วง +5V ถึง -5V สัญญาณควบคุมการเลือกแชนเนลของ Multiplexer นี้ได้มาจาก IC-13 และ IC-15:B ดังแสดงในรูปที่ 3.10 และ 3.11

รูปที่ 3.10 แสดงวงจรในส่วนของ Analog Multiplexer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

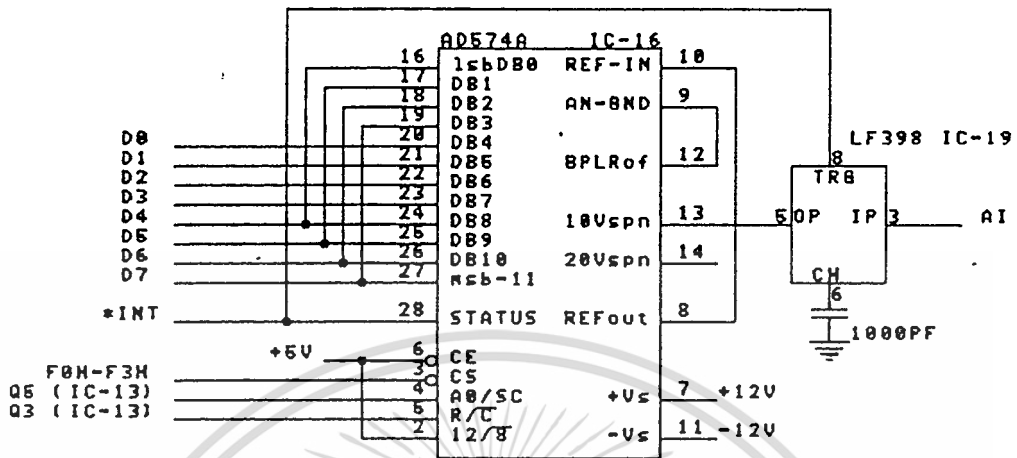


รูปที่ 3.11 แสดงวงจรในส่วนของ Channel Select

เข้าพุทของ Analog Multiplexer จะป้อนให้กับวงจร Sample and Hold ซึ่งทำงานด้วย LF398 IC-19 เพื่อรักษาระดับของแรงดันแอนาล็อกไว้ในช่วงที่ ADC กำลังเปลี่ยนแรงดันเป็นข้อมูลทางดิจิทัลโดยใช้ระดับลอจิกที่ขา TRG เป็นตัวควบคุม คือเมื่อสัญญาณที่ TRG เป็น High จะเป็นช่วง Hold Data

AD574 IC-16 ทำงานเป็น ADC 12 BIT มีภาคกำเนิดสัญญาณนาฬิกาอยู่ภายใน ใช้เวลาในการแปลงข้อมูลแต่ละครั้งเท่ากับ 25 ไมโครวินาที เนื่องจาก Data Bus ของ CPU เป็นขนาด 8 บิต จึงจำเป็นต้องต่อ DB3-DB0 ขนานกับ DB11-DB8 การกระตุ้นการทำงานสามารถทำได้หลายวิธี แต่ที่เลือกใช้นั้น จะกระตุ้นโดยการเปลี่ยนระดับลอจิกที่ขา \*CS ระดับลอจิกที่ขา R/C จะเป็นตัวกำหนดการอ่านข้อมูล และการเปลี่ยนข้อมูล และขา AO จะเป็นตัวกำหนด ไบท์ของข้อมูลที่ต้องการจะอ่านว่าเป็นไบท์ใด กล่าวคือ AO เป็น "1" จะเป็นการอ่านข้อมูลจากไบท์บน (DB11-B8) และ AO เป็น "0" เป็นการอ่านข้อมูล จากไบท์ล่าง (DB3-DB0 , 0000) ดังแสดงในรูปที่ 3.12

74LS374 IC-13 ทำหน้าที่เป็น Output Port เพื่อกำเนิดสัญญาณควบคุมการทำงานของ AD574 โดยใช้บิตที่ D6 และ D7 ส่วนบิตที่เหลือคือ D0-D4 และ D7 จะนำไปใช้ควบคุมภาค Analog Multiplexer ทำงานร่วมกับ 74LS139 IC-15:B สัญญาณที่ขา STATUS ของ AD574 จะใช้ในการ Interrupt การทำงานของ CPU เพื่อให้ CPU อ่านข้อมูลจาก AD574



รูปที่ 3.12 แสดงวงจรในส่วนของ Analog To Digital Converter

8273 HDLC Controller IC-8 ทำหน้าที่ติดต่อสื่อสารกับ PC โดยมี 8253 IC-7 ทำหน้าที่เป็น Buad Rate Generater และ MC145406 IC-9 ทำหน้าที่ RS232 Reciever-Driver การติดต่อกันระหว่าง 8273 กับ CPU จะทำผ่านทาง การ Interrupt ไปยัง CPU ส่วน CPU จะติดต่อกับ 8273 ผ่านทาง I/O Port ICL7660 IC-10 ทำหน้าที่เป็น Voltage Inverter กำเนิดแรงดัน -5V จ่ายให้กับ MC145406

8255 (IC-24,25,26) ทำหน้าที่เป็น Digital Input/Output โดยกำหนดให้ Port B ของ IC-24,25 Port A และ Port C ของ IC-26 ทำหน้าที่เป็น Digital Output ขนาด 32 แชนแนล Port A Port C ของ IC-24 และ IC-25 ทำหน้าที่เป็น Digital Input ขนาด 32 แชนแนลวงจรสมบูรณ์แสดงใน ภาคผนวก ก

ในส่วนของการออกแบบแผ่นวงจรพิมพ์ (Print Circuit Board) จำเป็นต้องใช้แผ่นวงจรพิมพ์แบบ 2 หน้า สำหรับส่วนของ Communication Card จะมีข้อกำหนดในเรื่องของขนาด เพราะจะต้องนำไปติดตั้งใช้งานใน PC และสำหรับส่วนของ Terminal Unit จะมีข้อกำหนดในการออกแบบเพิ่มเติมคือ ลายทองแดงในส่วนของวงจร Sample And Hold จำเป็นจะต้องมี Ground Plane เพื่อป้องกันส่วนที่เป็นสัญญาณอนาล็อกกับส่วนที่เป็นสัญญาณดิจิตอลออก

จากกัน เพื่อป้องกันการรบกวนกัน และลายทองแดงในส่วนของวงจร Analog To Digital Converter คือ การเดินลายทองแดงจาก Analog Ground (pin 9) ไปยัง กราวด์ของ อินพุท จะต้องแยกออกจากกราวด์ของ Digital Ground (pin 15) โดยจะต้องให้ลายทองแดงมีขนาดกว้างพอสมควร กระแสไฟตรงที่ไหลผ่านขา 9 จะมีค่าประมาณ 7 mA และผ่านขา 15 จะมีค่าประมาณ 5.5 mA แผ่นวงจรพิมพ์ที่ออกแบบเสร็จเรียบร้อยแล้ว แสดงในภาคผนวก ก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ,  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การทำงานของโปรแกรมควบคุม Communication Card

โปรแกรมควบคุมการทำงานของ Communication Card แบ่งออกเป็น 3 ส่วน คือ

1. โปรแกรมหลัก (Main Program)
2. โปรแกรมบริการการอินเทอร์รัท (Interrupt Service Routine)
3. โปรแกรมสนับสนุนการดีเอ็มเอ (DMA Support Routine)

โปรแกรมหลัก จะเริ่มทำงานตั้งแต่ การตรวจหา Configuration File (CONFIG.NET) ในไดเรกทอรีใช้งานปัจจุบัน ถ้าไม่พบก็จะเลิกการทำงานของโปรแกรม และกลับออกไปที่ดอส ถ้าพบไฟล์ดังกล่าว ก็จะอ่านค่าของพารามิเตอร์ต่างๆในไฟล์ เพื่อนำไปใช้งานต่อไปในโปรแกรม หลังจากนั้นก็จะเข้าสู่เมนูหลัก (Main Menu) เพื่อแสดงสถานะต่างๆของระบบให้ผู้รู้ และรอคอยการเลือกการทำงานต่อไปจากผู้ใช้ โดยจะตรวจสอบจากสถานะการกดคีย์บอร์ด จากโปรแกรมบริการการอินเทอร์รัท

โปรแกรมบริการการอินเทอร์รัทจะแบ่งออกเป็น 2 ส่วน คือ

1. โปรแกรมบริการการอินเทอร์รัท Timertick (18.2 Hz) อาศัย Hardware Interrupt ของ BIOS ภายในโปรแกรมจะประกอบด้วย การตรวจสอบการกดคีย์บอร์ด และส่งค่าของคีย์ที่กดให้กับโปรแกรมหลัก , เก็บบันทึกค่าของวันเวลาปัจจุบัน (Current Time) และทำหน้าที่ เก็บบันทึกข้อมูลที่รับมาได้จาก TU ลงใน Logging Device (Ramdisk) เมื่อครบช่วงเวลาที่ใช้กำหนด (Logging Time)

2. โปรแกรมบริการการอินเทอร์รัทจาก Communication Card ภายในของ Card จะใช้ Counter Channel 1,2 ของ 8253 ทำงานร่วมกันเป็นตัวกำเนิดสัญญาณอินเทอร์รัทต่อ CPU ผ่านทางการอินเทอร์รัทแชนแนลที่ 5 (IORQ5) ทุกๆช่วงเวลา 10 ms เพื่อใช้เป็นตัวกำหนดจังหวะการตรวจสอบระบบสื่อสารข้อมูลของ 8273 ภายในโปรแกรมประกอบไปด้วยการรับ และการส่งเฟรมของข้อมูลและคำสั่ง , การตรวจสอบความผิดพลาดของการรับและส่ง และกำหนดช่วงเวลาในการถ่ายเทข้อมูลจาก Logging Device ไปยัง Transfer

Device เมื่อเนื้อที่ว่างใน Logging Device เหลือน้อยกว่า 1 %

อีกส่วนคือ โปรแกรมบริการอินเทอร์รัพของ 8273 ในอินเทอร์รัพแชนแนลที่ 4 (IORQ 4) เพื่อใช้สำหรับบอกผลลัพธ์ของการรับและส่งเฟรมต่อ CPU ดังนั้นโปรแกรมในส่วนนี้จะทำหน้าที่อ่านผลลัพธ์จาก Result Register ภายใน 8273 นำมาเก็บไว้ในบัฟเฟอร์ พร้อมทั้งเช็คค่าของแฟล็กเพื่อบอกให้ CPU รู้

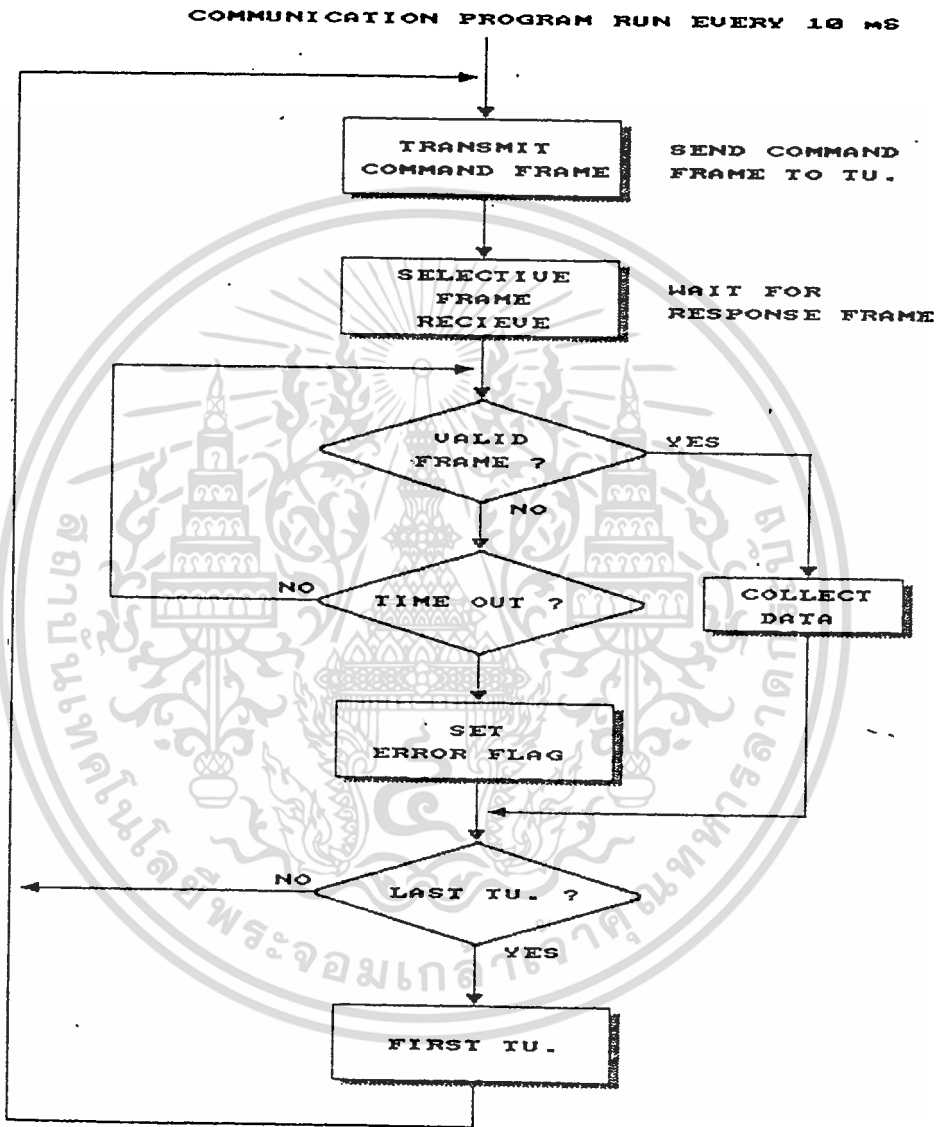
โปรแกรมสนับสนุนการดีเอ็มเอ จะทำหน้าที่สนับสนุนการทำงานของ 8273 ในการรับข้อมูล และส่งข้อมูลกับหน่วยความจำ กล่าวคือเมื่อ 8273 ต้องการส่งเฟรมของข้อมูล 8273 จะขอทำการติดต่อกับหน่วยความจำผ่านทาง DRQ1 ไปยัง 8257 DMA Controller โปรแกรมสนับสนุนการดีเอ็มเอ จะเขียนโหมดการทำงาน ให้เป็นการอ่านข้อมูลจากหน่วยความจำ (Memory to I/O) ,พารามิเตอร์ต่างๆที่จำเป็นในการติดต่อกับหน่วยความจำ คือ ค่าเช็กเมนต์และออฟเซต ของแอดเดรสของข้อมูลที่ 8273 ต้องการและจำนวนไบต์ของการส่งผ่านข้อมูลให้กับ 8257 และเมื่อ 8273 ได้รับเฟรมของข้อมูล ก็จำเป็นจะต้องมีการเขียนข้อมูลดังกล่าวลงในหน่วยความจำก็จะติดต่อกับ DMA ผ่านทาง DRQ 1 เช่นเดียวกัน โปรแกรมก็จะทำงานในลักษณะเดียวกัน แต่จะเปลี่ยนโหมดการทำงานของ 8257 ให้ทำงานในลักษณะเขียนข้อมูลลงในหน่วยความจำแทน (I/O to Memory)

การทำงานของโปรแกรมทั้งหมดสามารถแสดงเป็นโฟลวชาร์ตดังรูปที่ 3.13  
การทำงานของโปรแกรมควบคุม Terminal Unit

โปรแกรมควบคุมจะแบ่งออกเป็น 2 ส่วนด้วยกัน คือ

1. โปรแกรมหลัก (Main Program)
2. โปรแกรมบริการการอินเทอร์รัพท์ (Interrupt Service Routine)

โปรแกรมหลักจะควบคุมการติดต่อสื่อสารของ 8273 เริ่มตั้งแต่ส่งคำสั่งให้ 8273 คอยตรวจสอบเพื่อรับเฟรมคำสั่งจาก Master Station (PC) เมื่อได้รับเฟรมดังกล่าวแล้วก็จะตรวจสอบว่ามีข้อผิดพลาดเกิดขึ้นหรือไม่ ถ้ามีการผิดพลาดก็จะยกเลิกเฟรมดังกล่าว และรอรับเฟรมคำสั่งใหม่ และถ้าไม่มีข้อผิดพลาดเกิดขึ้น ก็จะเริ่มต้นกระตุ้นให้ AD574 อ่านค่าของ



รูปที่ 3.13 แสดง โฟลวชาร์ตการทำงานของโปรแกรมควบคุม Communication

Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า, ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุททางอนาล็อก และแปลงค่าดังกล่าวเป็นข้อมูลทางดิจิตอล และรอจนกระทั่งครบทั้ง 32 แชนแนล เมื่อครบแล้วก็จะอ่านค่าของอินพุททางดิจิตอลทั้ง 32 แชนแนล แล้วส่งให้ 8273 ส่งเฟรมของข้อมูลทีอ่านได้กลับไปยัง Master Station (PC)

ในส่วนที่เป็นเข้าพุทแบบดิจิตอล จะนำข้อมูลจากเฟรมคำสั่งที่ได้รับ ซึ่งจะเป็นส่วนของ Control Byte ตามที่ผู้ใช้ต้องการควบคุมส่งออกไปยัง Digital Output

โปรแกรมบริการการอินเทอร์รัพท์แบ่งออกเป็น 2 ส่วน คือ

1.Maskable Interrupt Service Routine ใช้ตอบสนองการอินเทอร์รัพท์ของ AD574 โดย CPU จะถูกโปรแกรมให้ทำงานในอินเทอร์รัพท์โหมด 0 และ AD574 จะอินเทอร์รัพท์ต่อ CPU เมื่อได้ทำการแปลงข้อมูลทางอนาล็อกเป็นดิจิตอลเรียบร้อยแล้ว (End of Conversion) โปรแกรมในส่วนนี้จะอ่านข้อมูลจาก AD574 ทั้งสองไบท์ และนำไปเก็บไว้ในส่วนของบัฟเฟอร์ข้อมูล และตรวจสอบว่าเปลี่ยนข้อมูลทางอนาล็อกครบทั้ง 32 แชนแนลหรือยัง ถ้ายังไม่ครบก็จะกระตุ้นให้ AD574 เปลี่ยนข้อมูล (Start Conversion) ต่อไปจนครบการติดต่อกันระหว่างโปรแกรมส่วนนี้ กับโปรแกรมหลักจะกระทำผ่านทาง แฟล็ก และบัฟเฟอร์ที่กำหนดไว้ในหน่วยความจำ (Ram)

2.Nonmaskable Service Routine ใช้ตอบสนองการอินเทอร์รัพท์ของ 8273 โดย 8273 จะอินเทอร์รัพท์ CPU เมื่อต้องการขอส่วนของ Data Field ใน Response Frame , ต้องการให้ CPU รับ Data Field ใน Command Frame , บอกลบผลลัพท์ของการส่งเฟรมข้อมูล เมื่อสิ้นสุดการส่ง และบอกลบผลลัพท์ของการรับเฟรมคำสั่งเมื่อสิ้นสุดการรับ ดังนั้นโปรแกรมในส่วนนี้จะทำการตรวจสอบรีจิสเตอร์สถานะ (Status Register) ภายใน 8273 ผ่านทาง I/O Read เพื่อดูว่า 8273 อินเทอร์รัพท์เนื่องจากเหตุการณ์ใดใน 4 หัวข้อดังกล่าว กล่าวคือ เมื่อ 8273 ต้องการ Data Field ใน Response Frame CPU ก็จะนำข้อมูลดังกล่าว จากบัฟเฟอร์ในหน่วยความจำเขียนผ่าน I/O Write ไปยัง 8273 , เมื่อ 8273 รับ Data Field ใน Command Frame CPU ก็จะอ่านข้อมูลผ่านทาง I/O Read จาก 8273 และนำไปเก็บในบัฟเฟอร์ เพื่อส่งผ่านไปยังโปรแกรมหลัก และเมื่อ 8273 ต้อง

การบอกผลลัพธ์ของการรับหรือส่ง CPU ก็จะทำอ่านผลลัพธ์จาก Result Register ภายใน 8273 ผ่านทาง I/O Read การทำงานแสดงเป็นโฟลวชาร์ทในรูปที่ 3.14

### การใช้งานโปรแกรมที่เกี่ยวข้อง

โปรแกรมที่เกี่ยวข้องกับการใช้งาน REMOTE DATA LOGGER จะประกอบด้วย

RAMDRIVE.SYS	6481	Bytes
CONFIG.SYS		
RELOG.EXE	55208	Bytes
EDCON.EXE	63218	Bytes
VIEW.EXE	59978	Bytes
CONFIG.NET	51142	Bytes

โดย 2 ไฟล์แรกจะเป็นไดร์เวอร์ของดอส ในการติดตั้ง โปรแกรมเพื่อใช้งานนั้นจะมีข้อกำหนดดังนี้

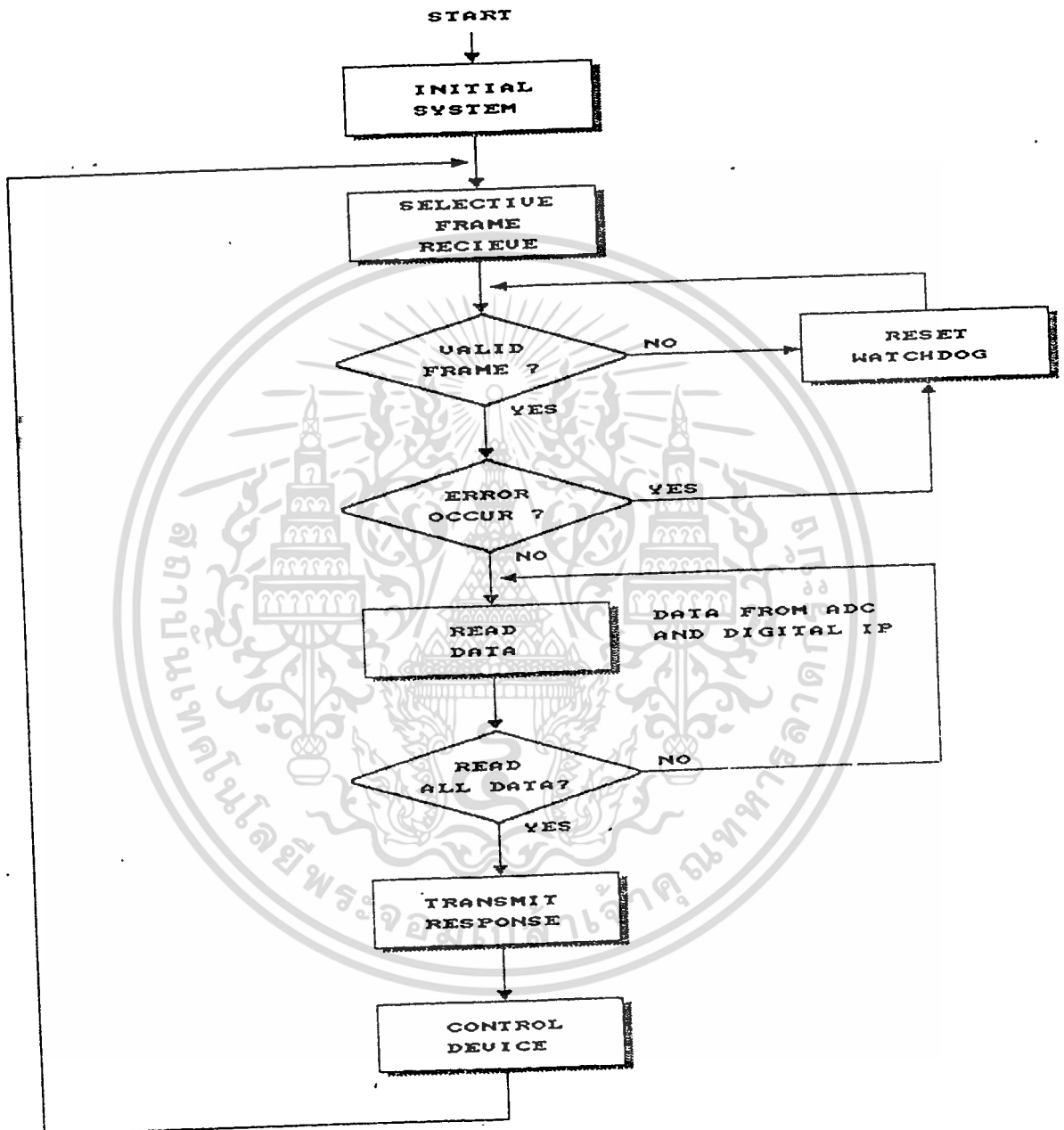
1. โปรแกรม RELOG.EXE และ CONFIG.NET จะต้องอยู่ในไดเรกทอรีเดียวกัน
2. โปรแกรม EDCON.EXE และ CONFIG.NET จะต้องอยู่ในไดเรกทอรีเดียวกัน
3. โปรแกรม VIEW.EXE และ CONFIG.NET จะต้องอยู่ในไดเรกทอรีเดียวกัน

4. จะต้องติดตั้งไดร์เวอร์ในไฟล์ CONFIG.SYS โดยกำหนดว่า

Device = RAMDRIVE.SYS NNN

โดย NNN จะเป็นตัวกำหนดขนาดของหน่วยความจำที่จะนำมาใช้เป็น Ramdisk กำหนดให้มีค่าตั้งแต่ 100 KB ถึง 360KB การติดตั้งโปรแกรมทำได้โดยใช้คำสั่ง COPY ของดอส จากดิสก์ต้นแบบไปยังดิสก์ที่ต้องการได้ตามปกติ

โปรแกรม RELOG.EXE เป็นโปรแกรมหลักที่ใช้ในการควบคุมการติดต่อสื่อสาร,



รูปที่ 3.14 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุม Terminal Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลของข้อมูล, การควบคุมอุปกรณ์ในระบบ และจัดเก็บข้อมูลของระบบ และจะใช้ได้เฉพาะกับเครื่องที่ได้ติดตั้ง Communication Card แล้วเท่านั้น เมื่อเริ่มใช้งานโปรแกรมทำได้โดยคีย์ชื่อของโปรแกรม RELOG ตามด้วยคีย์ Enter เมื่อเริ่มทำงานโปรแกรมจะค้นหาไฟล์ CONFIG.NET ในไดเรกทอรีใช้งานปัจจุบัน และอ่านพารามิเตอร์ต่างๆในไฟล์มาใช้งาน จากนั้นจะให้ผู้ใช้ป้อนข้อมูลของ Transfer Device ซึ่งหมายถึงชื่อของไดรฟ์และไดเรกทอรีที่ผู้ใช้ต้องการให้โปรแกรมนำข้อมูลที่จัดเก็บมาได้จาก Terminal Unit ไปบันทึกไว้ โดยมีรูปแบบดังนี้

ชื่อไดรฟ์:\ชื่อไดเรกทอรีที่ 1\ชื่อไดเรกทอรีที่ 2\...\ชื่อไดเรกทอรีสุดท้าย  
(Enter)

ข้อความทั้งหมดสามารถมีความยาวได้จำนวน 20 ตัวอักษร (รวมเครื่องหมาย ":" และ "\")

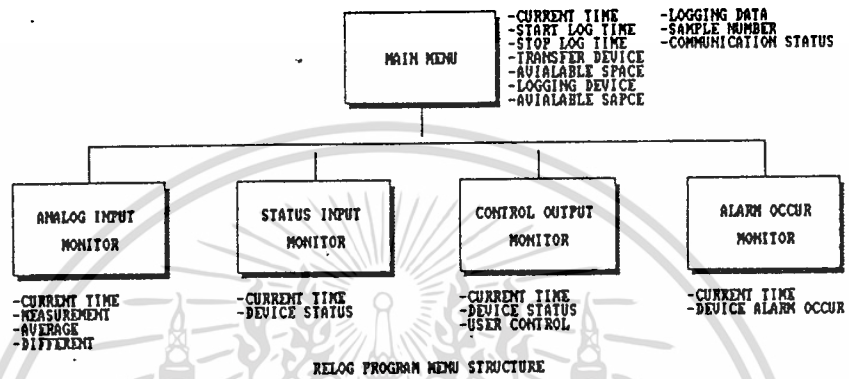
ต่อมาโปรแกรมจะถามถึงข้อมูลของ Logging Device ซึ่งก็คือ ไดรฟ์ที่ติดตั้ง Ramdisk นั้นเอง โดยปกติแล้วดอสจะติดตั้ง Ramdrive ไว้เป็นไดรฟ์สุดท้ายต่อจากไดรฟ์ที่มีใช้งานจริงในเครื่อง เช่น เครื่องมีการใช้งาน Floppy Disk Drive 2 ตัว และ HardDisk 1 ตัว ก็จะมีไดรฟ์ใช้งานจริงคือ A, B และ C ดังนั้น ดอสจะติดตั้ง Ramdisk ให้ใช้งานเป็นไดรฟ์ D เป็นต้น การป้อนข้อมูลจะมีรูปแบบดังนี้

ชื่อไดรฟ์: (Enter)

เมื่อป้อนข้อมูลของ Device ทั้งสองครบแล้ว โปรแกรมจะตรวจสอบความพร้อมของ Device ทั้งสองตัวด้วยการทดลองเขียนและอ่านข้อมูลลงใน Device แต่ละตัว ตัวใดผิดพลาดก็แจ้งให้ผู้ใช้ทราบเพื่อแก้ไขและถ้าเป็นปกติก็จะเข้าไปทำงานในเมนูหลัก (Main Menu) ต่อไป การแสดงผลในโปรแกรม จะแบ่งออกเป็นเมนูย่อยดังมีโครงสร้างดังรูปที่ 3.15

ในหน้า เมนูหลัก โปรแกรมจะแสดงสถานะต่างๆของระบบแต่ละหัวข้อจะมีความหมาย

ดังนี้



รูปที่ 3.15 แสดง โครงสร้างเมนูของ โปรแกรม Reelog

CURRENT TIME - เวลาปัจจุบันในขณะที่ใช้งาน โปรแกรม

TRANSFER DEVICE - ชื่อของ ไดรฟ์และไดเรกทอรีที่ผู้ใช้ต้อง การเก็บข้อมูล

AVIALABLE SPACE - เนื้อที่ว่างที่สามารถใช้เก็บข้อมูลได้ของ Transfer

Device คิดเป็นเปอร์เซ็นต์

START TIME - เวลาที่เริ่มต้นการเก็บบันทึกข้อมูล (กดคีย์ F5)

STOP TIME - เวลาที่สิ้นสุดการเก็บบันทึกข้อมูล(กดคีย์ F6)

LOGGING DEVICE - ชื่อของไดรฟ์ที่ดอสติดตั้ง Ramdisk

AVIALABLE SPACE - เนื้อที่ว่างของ Ramdisk คิดเป็นเปอร์เซ็นต์

LOGGING DATA - สถานะของการเก็บบันทึกข้อมูลจะแสดงเป็น 3 ข้อ

ความคือ

ON - กำลังทำการบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ,  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OFF - หยุดการบันทึกข้อมูล

DUMP- ถ่ายเทข้อมูลที่บันทึกได้จาก Logging Device ไปยัง Transfer Device อันเนื่องมาจากเนื้อที่ว่างของ Logging Device น้อยกว่า 1 %

STATION NAME - ชื่อของ Terminal Unit ที่ผู้ใช้งานกำหนดไว้

LOGGING PERIOD --ช่วงเวลาของการเก็บบันทึกข้อมูลแต่ละชุด มีหน่วยเป็น วินาที

COM.STATUS- สถานะของการติดต่อสื่อสารระหว่าง PC กับ Terminal Unit จะแสดงเป็น 2 ข้อความ คือ

POLING การติดต่อสื่อสารระหว่าง PC กับ TU เป็นปกติ

ABNORMAL การติดต่อสื่อสารระหว่าง PC กับ TU ผิดปกติ อันอาจเกิดจากความผิดพลาดของ Terminal Unit หรือ Communication Line และในบรรทัดล่างสุดจะบอกถึง Function Key ที่ใช้งานและความหมายของแต่ละ Key ดังแสดงในรูปที่ 3.16

<-REMOTE DATA LOGGER->		
CURRENT TIME : 01-AUG-1993 : 10:50:34		LOGGING DEVICE : E:
TRANSFER DEVICE : C:\LOGGER		AVAILABLE SPACE : 98 %
AVAILABLE SPACE : 100%		LOGGING DATA : ON
START LOG TIME : 9:10:00		SAMPLE DATA : 199
STOP LOG TIME : 23:00:00		
STATION NAME	LOGGING (SEC)	COM.LINE
TERMINAL_NAME1	3	POLLING
TERMINAL_NAME2	6	POLLING
TERMINAL_NAME3	9	ABNORMAL
TERMINAL_NAME4	12	POLLING
TERMINAL_NAME5	15	ABNORMAL
TERMINAL_NAME6	18	POLLING
F1-Analog F2-Status F3-Control F4-Alarm F5-Start F6-Stop F10-Quit		

รูปที่ 3.16 แสดงหน้าเมนู Main Menu ของโปรแกรม Relog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า , ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### F1-Measure

การแสดงผลของหน้าจอนี้จะขึ้นอยู่กับข้อกำหนดที่ผู้ใช้กำหนดไว้ใน Analog Input Configuration แสดงค่าวัดของอินพุทแบบอนาล็อก ตามเวลาจริง โดยจะแบ่งเป็นข้อมูลของแต่ละ Terminal Unit และจะแสดงผลเป็นหน้า (page) หน้าละ 16 แชนแนล เรียงลำดับกันไปตามหมายเลขแชนแนล รวม 2 หน้า โดยจะแสดงทั้งค่าวัดจริงปัจจุบัน (Measurement) ค่าเฉลี่ยของการวัด (Average) ซึ่งคิดจากค่าวัดครั้งก่อนรวมกับค่าวัดปัจจุบัน แล้วหารด้วยสอง และค่าความแตกต่าง (Different) คิดจากค่าวัดครั้งก่อนลบด้วยค่าวัดปัจจุบัน ทั้งนี้การแสดงผลจะขึ้นอยู่กับ การกำหนดการใช้งาน (Config) ว่ามีการใช้งานหรือไม่ (Use , Not Use) ถ้าแชนแนลใดไม่มีการใช้งาน ก็จะไม่มีผลการแสดงผลขึ้นบนจอภาพ การเลือกการแสดงผลทำได้ด้วยคีย์ที่กำหนดไว้ในบรรทัดล่างสุดของจอภาพ ตามที่แสดงไว้ในรูปที่ 3.17

ANALOG INPUT MONITOR				
STATION NAME : TERMINAL_NAME1		01-AUG-1993   11:02:34		
DEVICE	MEASURE	AVERAGE	DIFFERENT	UNIT
ANALOG IP1	0.0000	0.0000	0.0000	UNIT1
ANALOG IP2	12.2345	12.4234	-1.2000	UNIT2
ANALOG IP3	230.0980	231.0082	0.2876	UNIT3
ANALOG IP4	1205.0000	1204.5000	-0.3428	UNIT4
ANALOG IP5	9999.9999	9999.9888	-0.0001	UNIT5
ANALOG IP6	1.0000E+01	1.0000E+01	99.0000	UNIT6
ANALOG IP7	12.3456E+20	12.5000E+20	1.0000E+01	UNIT7
ANALOG IP8	365.5673E-19	365.5000E-19	-1.5632E-02	UNIT8
ANALOG IP9	9999.9999E+20	9999.9990E+20	-3.9999E+02	UNIT9
ANALOG IP10	0.0000	0.0000	0.0000	UNIT10
ANALOG IP11	12.2345	12.4234	-1.2000	UNIT11
ANALOG IP12	230.0980	231.0082	0.2876	UNIT12
ANALOG IP13	1205.0000	1204.5000	-0.3428	UNIT13
ANALOG IP14	9999.9999	9999.9888	-0.0001	UNIT14
ANALOG IP15	1.0000E+01	1.0000E+01	99.0000	UNIT15
ANALOG IP16	12.3456E+20	12.5000E+20	1.0000E+01	UNIT16

-Sel stat      -Sel page      Esc-Back to main menu

รูปที่ 3.17 แสดงหน้าจอ Measure ของโปรแกรม Relog

การแสดงผลค่าวัดที่เป็นตัวเลข จะแสดงค่าเป็น 2 แบบ คือ

1. ถ้าค่าที่วัดได้อยู่มากกว่า ค่าที่กำหนดไว้ใน Threshold และ อยู่ในช่วง -9999.9999 ถึง 9999.9999 ก็จะแสดงผลตามปรกติ คือ "+nnnn.nnnn" แต่ถ้าค่าที่วัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้อยู่นอกช่วงดังกล่าว ก็จะเปลี่ยนค่าวัดให้อยู่ในรูปของ Engineering Number คือ "+nnnn.nnnnE+nn" แล้วจึงนำมาแสดงผลบนจอภาพ

2. ถ้าค่าที่วัดได้น้อยกว่า หรือเท่ากับค่าที่กำหนดไว้ใน Threshold ก็จะแสดงผลเป็น "0.0000"

การแสดงผลของค่าเฉลี่ย และค่าความแตกต่างนั้น ถ้าได้กำหนดไว้ใน Average และ Different ว่ามีการใช้งาน (Use) ก็จะมีการแสดงผลเช่นเดียวกับค่าวัด แต่ถ้ากำหนดว่าไม่มีการใช้งาน (Not Use) ก็จะแสดงผลเป็น " - " การแสดงผลจะเปลี่ยนตามค่าวัดปัจจุบัน (Update) ทุกๆเวลา 3 วินาที ดังแสดงในรูปที่ 3.17

#### F2-Status

ในหน้าจอนี้จะแสดงผลของสถานะของอินพุทแบบดิจิทัล โดยจะแบ่งเป็นข้อมูลของแต่ละ Terminal Unit และแสดงครบทั้ง 32 แชนแนลทั้งนี้การแสดงผลจะขึ้นอยู่กับข้อกำหนดใน Digital Input Configuration คือถ้ามีการกำหนดการใช้งาน (Config) ว่ามีการใช้งานหรือไม่ (Use, Not Use) ถ้าแชนแนลใดไม่มีการใช้งาน ก็จะไม่มีการแสดงผลขึ้นบนจอภาพ การเลือกการแสดงผลทำได้ด้วย คีย์ที่กำหนดไว้ในบรรทัดล่างสุดของจอภาพ ตามที่แสดงไว้ในรูปที่ 3.18

การแสดงผลสถานะของอินพุทจะแสดงดังนี้

1. เมื่อแรงดันไฟตรงที่อินพุทแบบดิจิทัล มีค่าอยู่ในช่วง 2.8 โวลต์ถึง 5 โวลต์ ก็จะแสดงผลในช่องของ Status ตามข้อความที่ผู้ใช้กำหนดใน On Message

2. เมื่อแรงดันไฟตรงมีค่าน้อยกว่า 2.8 โวลต์ ก็จะแสดงผลตามข้อความที่กำหนดใน Off Message

การแสดงผลจะเปลี่ยนตามสถานะปัจจุบัน (Update) ทุกๆเวลา 3 วินาทีดังแสดงในรูปที่ 3.18

----- STATUS INPUT MONITOR -----			
STATION NAME : TERMINAL_NAME1		01-AUG-1993 ! 11:05:56	
DEVICE	STATUS	DEVICE	STATUS
DIGITAL_IP1	: OFF	DIGITAL_IP17	: OFF
DIGITAL_IP2	: ON	DIGITAL_IP18	: ON
DIGITAL_IP3	: ON	DIGITAL_IP19	: ON
DIGITAL_IP4	: OFF	DIGITAL_IP20	: OFF
DIGITAL_IP5	: ON	DIGITAL_IP21	: ON
DIGITAL_IP6	: ON	DIGITAL_IP22	: ON
DIGITAL_IP7	: OFF	DIGITAL_IP23	: OFF
DIGITAL_IP8	: ON	DIGITAL_IP24	: ON
DIGITAL_IP9	: OFF	DIGITAL_IP25	: OFF
DIGITAL_IP10	: ON	DIGITAL_IP26	: ON
DIGITAL_IP11	: ON	DIGITAL_IP27	: ON
DIGITAL_IP12	: OFF	DIGITAL_IP28	: OFF
DIGITAL_IP13	: ON	DIGITAL_IP29	: ON
DIGITAL_IP14	: ON	DIGITAL_IP30	: ON
DIGITAL_IP15	: OFF	DIGITAL_IP31	: OFF
DIGITAL_IP16	: ON	DIGITAL_IP32	: ON

Esc-Back to main menu

รูปที่ 3.18 แสดงหน้าจอเมนู Status ของโปรแกรม Relog

### F3-Control

ในหน้าจอนี้จะแสดงผลของสถานะของเอาพุทแบบดิจิตอล ตามการควบคุมของผู้ใช้ การแสดงผลจะเหมือนกับการแสดงผลใน Status Monitor ผู้ใช้สามารถควบคุมสถานะของเอาพุทดังกล่าวได้ว่าต้องการให้ On หรือ Off โดยการควบคุมด้วยคีย์ดังแสดงไว้ในบรรทัดล่างสุดตามที่แสดงไว้ในรูปที่ 3.19

การเลือกการควบคุมของผู้ใช้ จะแสดงแขนแนลที่เลือกด้วยแถบสว่าง (High Light Bar) ปรากฏบนชื่อของ แขนแนลที่ต้องการการตอบสนองของ Terminal Unit หลังจากผู้ใช้เลือกการควบคุมจะใช้เวลานานที่สุด 3 วินาที เพราะต้องใช้เวลาในการติดต่อสื่อสาร

CONTROL OUTPUT MONITOR			
STATION NAME : TERMINAL_NAME1		01-AUG-1993 : 11:05:56	
DEVICE	STATUS	DEVICE	STATUS
DIGITAL_OP1	: OFF	DIGITAL_OP17	: OFF
DIGITAL_OP2	: ON	DIGITAL_OP18	: ON
DIGITAL_OP3	: ON	DIGITAL_OP19	: ON
DIGITAL_OP4	: OFF	DIGITAL_OP20	: OFF
DIGITAL_OP5	: ON	DIGITAL_OP21	: ON
DIGITAL_OP6	: ON	DIGITAL_OP22	: ON
DIGITAL_OP7	: OFF	DIGITAL_OP23	: OFF
DIGITAL_OP8	: ON	DIGITAL_OP24	: ON
DIGITAL_OP9	: OFF	DIGITAL_OP25	: OFF
DIGITAL_OP10	: ON	DIGITAL_OP26	: ON
DIGITAL_OP11	: ON	DIGITAL_OP27	: ON
DIGITAL_OP12	: OFF	DIGITAL_OP28	: OFF
DIGITAL_OP13	: ON	DIGITAL_OP29	: ON
DIGITAL_OP14	: ON	DIGITAL_OP30	: ON
DIGITAL_OP15	: OFF	DIGITAL_OP31	: OFF
DIGITAL_OP16	: ON	DIGITAL_OP32	: ON

-Sel stat    -Sel device    Enter-Accept select    Esc-Back to main menu

รูปที่ 3.19 แสดงหน้าจอเมนู Control ของโปรแกรม Relog

#### F4-Alarm

เป็นหน้าจอเมนูที่ใช้สำหรับแจ้งเหตุผิดปกติที่เกิดขึ้นตามที่ผู้ใช้กำหนด แบ่งได้เป็น 2 ประเภท คือ เหตุผิดปกติที่เกิดจากค่าวัดทางอนาล็อก ซึ่งผู้ใช้ได้กำหนดให้มีการตรวจเช็คค่าวัด (Alarm) กับค่าที่กำหนด (Maximum, Minimum) และเหตุผิดปกติที่เกิดจากอินพุตดิจิทัลที่ผู้ใช้กำหนดให้เป็นเซนแนลแจ้งเหตุ (Alarm) หน้าเมนูแสดงในรูปที่ 3.20

การแสดงผลจะแสดงได้ 18 เหตุการณ์ใน 1 หน้า และจะเลื่อนขึ้นทีละ 1 บรรทัด เมื่อมีเหตุการณ์ใหม่ปรากฏขึ้น การเปลี่ยนแปลงตามเหตุการณ์จริง (Update) จะใช้เวลาทุกๆ 3 วินาทีในแต่ละ Terminal Unit

#### F5-Start, F6-Stop

ใช้ควบคุมการบันทึกข้อมูลว่าต้องการให้มีการบันทึกหรือไม่ หลังจากกดคีย์ F5 โปรแกรมจะแสดงเวลาเริ่มต้นของการบันทึกข้อมูลที่ Start Time พร้อมทั้ง แสดงสถานะของการบันทึกข้อมูลที่ Logging data

การบันทึกข้อมูลจะจัดเก็บไว้เป็นไฟล์ของแต่ละ TU โดยจะนำอักษร 8 ตัวแรกของ

ชื่อ TU มาใช้เป็นชื่อไฟล์ ส่วนนามสกุลจะกำหนด ดังนี้

TU ตัวที่ 1 ใช้เป็น .S#1

TU ตัวที่ 2 ใช้เป็น .S#2

TU ตัวที่ 3 ใช้เป็น .S#3

TU ตัวที่ 4 ใช้เป็น .S#4

TU ตัวที่ 5 ใช้เป็น .S#5

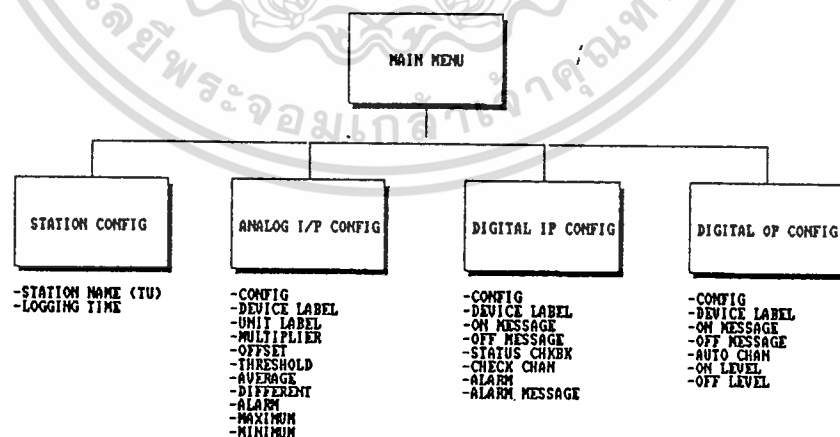
TU ตัวที่ 6 ใช้เป็น .S#6

ส่วนการถ่ายเทไฟล์จาก Logging Device ไปยัง Transfer Device นั้น โปรแกรมจะสร้างไดเรกทอรีย่อย (Sub Directory) ขึ้นใน Transfer Device สำหรับการถ่ายเทไฟล์แต่ละครั้ง โดยจะกำหนดชื่อของไดเรกทอรีย่อยเรียงลำดับกันไป ดังนี้

\SAM1, \SAM2, \SAM3, \SAM4, ... \SAM20

และเมื่อต้องการหยุดการบันทึกข้อมูล ทำได้ด้วยการกดคีย์ F6

โปรแกรม EDCON.EXE ใช้งานสำหรับกำหนดพารามิเตอร์ต่างๆ ที่ใช้งานในระบบ การเรียกใช้งานโปรแกรมทำได้โดยคีย์ชื่อของโปรแกรม EDCON ตามด้วยคีย์ Enter เมื่อโปรแกรมเริ่มทำงานจะเข้าสู่หน้าเมนูหลัก (Main Menu) โครงสร้างของเมนูแสดงในรูปที่ 3.20



EDCON PROGRAM MENU STRUCTURE

รูปที่ 3.20 แสดงโครงสร้างเมนูของโปรแกรม Edcon

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแก้ไขพารามิเตอร์ต่างๆนั้น จะกระทำกับไฟล์ Config.net ซึ่งถ้ามีการใช้งานไฟล์ดังกล่าวอยู่ก่อนแล้ว ก็สามารถโหลดไฟล์เพื่อแก้ไขได้ ด้วยการกดคีย์ F6 โปรแกรมจะถามถึงที่อยู่ของไฟล์ว่าอยู่ในไดรฟ์และไดเรกทอรีใด เมื่อผู้ใช้ป้อนข้อมูลเรียบร้อยแล้ว โปรแกรมจะอ่านไฟล์ดังกล่าวเพื่อทำการแก้ไขต่อไป หรือถ้าเป็นการสร้างไฟล์ใหม่ก็สามารถทำได้โดยกดคีย์ F1 และตามด้วยกดคีย์ Enter หน้าเมนูย่อยของโปรแกรมสำหรับแก้ไขพารามิเตอร์มีดังนี้

Station Configuration (Station) จะใช้สำหรับแก้ไขพารามิเตอร์ของ Terminal Unit มีหัวข้อของการเลือกหัวข้อที่ต้องการแก้ไข แสดงด้วยแถบสว่าง (High Light Bar) ดังแสดงในรูปที่ 3.21 พารามิเตอร์ที่สามารถแก้ไขได้ดังนี้

STATION NAME	LOGGING TIME (x3 SEC)
TERMINAL_NAME1	1
TERMINAL_NAME2	2
TERMINAL_NAME3	3
TERMINAL_NAME4	4
TERMINAL_NAME5	5
TERMINAL_NAME6	6

-Sel topic      Enter-Confirm select      Esc-Back to main menu

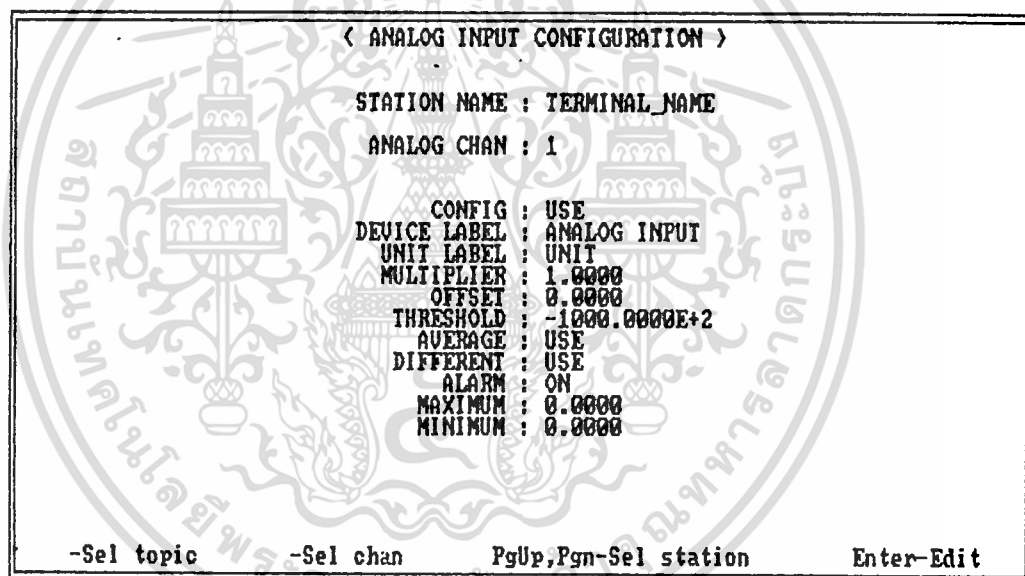
รูปที่ 3.21 แสดงหน้าจอเมนู Station ของโปรแกรม Edcon

Station Name - เป็นการกำหนดชื่อของแต่ละ Terminal Unit ตามที่ต้องการ มีความยาวได้ 15 ตัวอักษรและชื่อที่ผู้ใช้กำหนดนี้ จะนำไปใช้เป็นชื่อไฟล์ ที่ใช้สำหรับจัดเก็บข้อมูลของโปรแกรม RELOG ด้วย โดยจะใช้ตัวอักษร 8 ตัวแรก ไปใช้เป็นชื่อไฟล์

Logging Time - คือระยะเวลาห่าง ของเวลาที่เก็บข้อมูลแต่ละชุดมีหน่วยเป็นวินาที ค่าที่ป้อนจะเป็นตัวเลขจำนวนเต็ม ที่มีค่าอยู่ระหว่าง 0 จนถึง 1200 ค่า

Logging Time จะหาได้จากนำค่าที่บ่อนคูณด้วย 3 เช่น ถ้าบ่อนค่าเท่ากับ 0 ก็จะมีหมายถึงไม่มีการจัดเก็บข้อมูล , บ่อนเท่ากับ 2 จะได้ค่า Logging Time เท่ากับ  $2 \times 3$  เท่ากับ 6 หมายความว่า โปรแกรม Relog จะจัดเก็บข้อมูลจาก Terminal Unit ดังกล่าวทุกๆเวลา 6 วินาที

Analog Input Configuration (Measure) ใช้สำหรับแก้ไขค่าพารามิเตอร์ของค่าวัดทางอนาล็อกจำนวน 32 แชนแนล โดยจะแสดงหมายเลขของแชนแนลที่ Analog Chan การเลือกแชนแนล และพารามิเตอร์ ทำได้ด้วยคีย์ที่แสดงไว้ในบรรทัดล่างสุด พารามิเตอร์ที่เลือก จะแสดงด้วยแถบสว่าง (High Light Bar) ปรากฏบนชื่อของพารามิเตอร์ที่ต้องการ ดังแสดงในรูปที่ 3.22 มีดังนี้



รูปที่ 3.22 แสดงหน้าจอเมนู Measure ของโปรแกรม Edcon

Config - ใช้กำหนดการใช้งานของค่าวัดแชนแนลใด ๆ ว่ามีการใช้งานหรือไม่ (Use, Not Use) ในกรณีที่ผู้ใช้ไม่ต้องการใช้งาน (NotUse) ก็จะไม่มีการเก็บข้อมูล และแสดงผลของแชนแนลดังกล่าว

Device Label - ใช้กำหนดชื่อ ของแชนแนลที่ต้องการ เพื่อให้ง่ายต่อการใช้งานตามความต้องการ มีความยาวได้ 15 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า , ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unit Label - เช่นเดียวกับ Device Label แต่จะใช้กำหนดชื่อของหน่วยวัดแทน

Multiplier - เป็นค่าของตัวเลขที่จะนำมาคูณกับค่าวัด (0-4095) เพื่อให้สำหรับแปลงเป็น ค่าวัดตามหน่วยจริงที่ใช้งานมีค่าอยู่ในช่วง  $-1E20$  ถึง  $1E20$  การบ้อนสามารถบ้อนได้ทั้งแบบตัวเลขธรรมดา และตัวเลขเชิงวิทยาศาสตร์ (Engineering Number)

Offset - คล้ายกับ Multiplier แต่จะเป็นค่าของตัวเลขที่นำมาบวกกับค่าวัดที่คูณกับค่า Multiplier แล้วเพื่อให้สำหรับแปลง เป็นค่าวัดตามหน่วยจริงที่ใช้งาน มีค่าอยู่ในช่วง  $-1E20$  ถึง  $1E20$  การบ้อน สามารถบ้อนได้ทั้งแบบตัวเลขธรรมดาและตัวเลขเชิงวิทยาศาสตร์ (Engineering Number)

Average - เป็นการกำหนดว่าต้องการใช้งานค่าเฉลี่ยของเซนแนลดังกล่าวหรือไม่ (Use , Not Use)

Different - เป็นการกำหนดว่าต้องการใช้งานค่าความแตกต่างของค่าวัดหรือไม่ (Use , Not Use)

Threshold - เป็นค่าของตัวเลขที่ใช้กำหนดค่าวัดต่ำสุดที่ใช้งาน (Lowest Value) ถ้าค่าวัดต่ำกว่า หรือเท่ากับค่าที่กำหนดใน Threshold ก็จะถือว่า ค่าวัดเป็น 0 มีค่าอยู่ในช่วง  $-1E20$  ถึง  $1E20$  การบ้อนสามารถบ้อนได้ทั้งแบบตัวเลขธรรมดา และตัวเลขเชิงวิทยาศาสตร์ (Engineering Number)

Alarm - เป็นการกำหนดว่าต้องการตรวจสอบระดับของค่าวัดหรือไม่ (Use , Not Use) โดยจะเปรียบเทียบกับค่า Maximum และค่า Minimum ถ้าค่าวัดสูงหรือต่ำ จากค่าที่กำหนดนี้ โปรแกรม Relog ก็จะทำให้แจ้งให้รู้ใน Alarm Menu

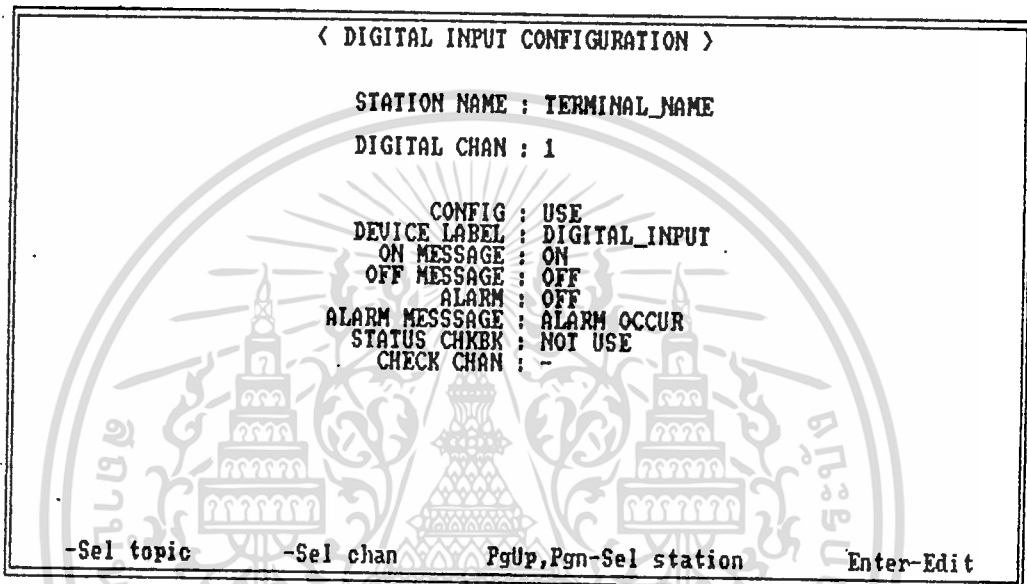
Maximum - ค่าสูงสุดของค่าวัดที่ยอมให้เกิดขึ้นได้

Minimum- ค่าต่ำสุดของค่าวัดที่ยอมให้เกิดขึ้นได้

ทั้งค่า Maximum และ Minimum จะเป็นตัวเลขเช่นเดียวกับค่าของ Multiplier และ Offset และจะสามารถบ้อนค่าได้ ก็ต่อเมื่อได้กำหนดให้มีการตรวจสอบระดับของค่าวัด (Use) เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า , ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digital Input Configuration (Status) ใช้สำหรับแก้ไขพารามิเตอร์ของ อินพุทแบบดิจิตอลจำนวน 32 แชนแนล โดยจะแสดงหมายเลขของแชนแนลที่ Digital Chan การเลือกแชนแนล และพารามิเตอร์ ทำได้ด้วยคีย์ ที่แสดงไว้ในบรรทัดล่างสุด ดังแสดงในรูปที่ 3.23 พารามิเตอร์ที่เลือก จะแสดงด้วยแถบสว่าง (High Light Bar) มีดังนี้



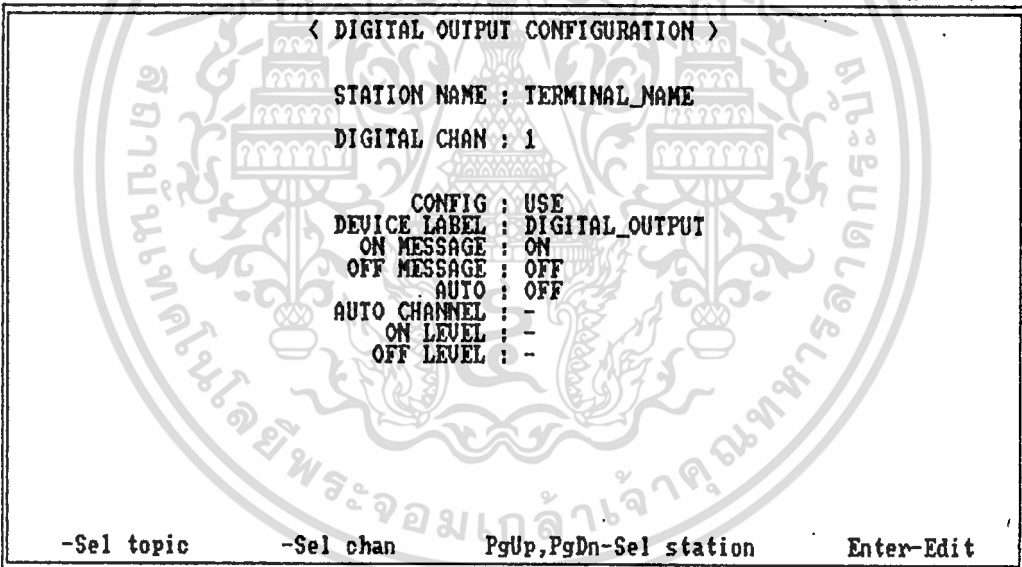
- รูปที่ 3.23 แสดงหน้าเมนู Status ของโปรแกรม Edcon
- Config - ใช้กำหนดการใช้งานของอินพุทดิจิตอลว่ามีการใช้งานหรือไม่ (Use ,Not use)
- Device Label - ใช้กำหนดชื่อของแชนแนลที่ต้องการ มีความยาวได้ 15 ตัวอักษร
- On Message - ใช้กำหนดข้อความที่ผู้ใช้ต้องการให้แสดงบนจอเพื่ออินพุทของ แชนแนลนั้นๆ มีระดับแรงดันไฟตรงสูงกว่า 2.8 โวลท์ (ON) มีความยาวได้ 15 ตัวอักษร
- Off Message - ใช้กำหนดข้อความที่ผู้ใช้ต้องการให้แสดงบนจอ เมื่ออินพุทมี ระดับแรงดันไฟตรง ต่ำกว่า 0.8 โวลท์ (OFF) มีความยาวได้ 15 ตัวอักษร
- Alarm- เป็นการกำหนดว่าอินพุทเป็นแบบ AlarmInput หรือไม่ (On,Off)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า, ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยถ้ากำหนดให้เป็นแบบ Alarm Input เมื่ออินพุทแขนแลดังกล่าว On โปรแกรมก็จะเตือนผู้  
ใช้ให้ทราบใน Alarm Menu ด้วย

Alarm Message - เป็นข้อความที่ผู้ใช้ต้องการให้แสดงบน Alarm Menu มี  
ความยาวได้ 15 ตัวอักษร

Digital Output Configuration (Control) ใช้กำหนดพารามิเตอร์ของ  
เอาต์พุตดิจิทัล จำนวน 32 แขนแล โดยจะแสดงหมายเลขของแขนแลที่ Digital Chan  
การเลือกแขนแล และพารามิเตอร์ ทำได้ด้วยคีย์ ที่แสดงไว้ในบรรทัดล่างสุด ดังแสดงในรูปที่  
3.24 พารามิเตอร์ที่เลือก จะแสดงด้วยแถบสว่าง (High Light Bar) ปรากฏบนชื่อของ  
พารามิเตอร์ ที่ต้องการ มีดังนี้



รูปที่ 3.24 แสดงหน้าเมนู Control ของโปรแกรม Edcon

Config - ใช้กำหนดการใช้งานของเอาต์พุตดิจิทัลว่ามีการใช้งานหรือไม่  
(Use , Not use)

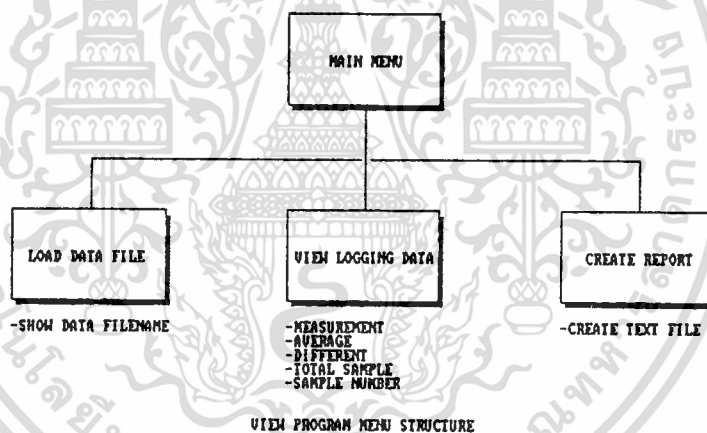
Device Label - ใช้กำหนดชื่อของแขนแลที่ต้องการ มีความยาวได้ 15  
ตัวอักษร

On Message - ใช้กำหนดข้อความที่ผู้ใช้ต้องการให้แสดงบนจอเพื่อเข้าพท  
ของเซนแนลนั้นที่ถูกควบคุมให้ ON มีความยาวได้ 15 ตัวอักษร

Off Message - ใช้กำหนดข้อความที่มีผู้ใช้ต้องการให้แสดงบนจอ เมื่อเข้าพท  
ถูกควบคุมให้ OFF มีความยาวได้ 15 ตัวอักษร

เมื่อแก้ไขพารามิเตอร์ทั้งหมดตามที่ต้องการเรียบร้อยแล้ว สามารถจัดเก็บลงใน  
ไฟล์ CONFIG.NET ได้ด้วยการกดคีย์ F5 และออกจากโปรแกรมเมื่อเลิกใช้งานด้วยคีย์ Esc.

โปรแกรม VIEW.EXE ใช้สำหรับดูข้อมูลที่จัดเก็บบันทึกไว้จากโปรแกรม ReLog  
พร้อมทั้งแปลงข้อมูลดังกล่าวเป็น TextFile เพื่อจัดทำเป็นรายงานต่อไป โครงสร้างหน้าเมนู  
ของ โปรแกรมแสดงในรูปที่ 3.25

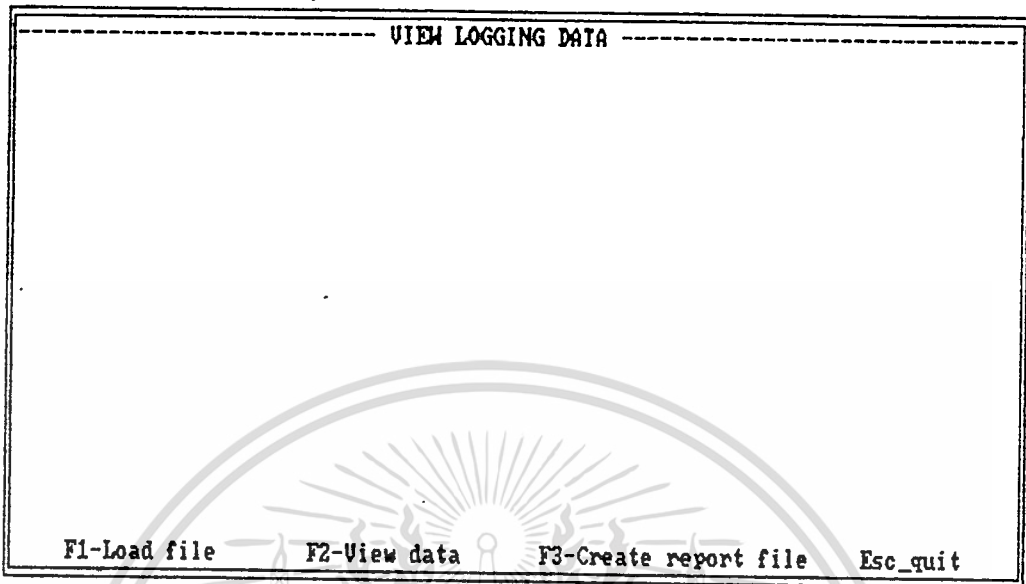


รูปที่ 3.25 แสดง โครงสร้างเมนูของ โปรแกรม View

เมื่อเริ่มรันโปรแกรม จะเข้าสู่หน้าเมนูหลักดังแสดงในรูปที่ 3.26 เมื่อต้องการจะดู  
ข้อมูลในไฟล์ใด ก็จะต้องโหลดไฟล์ด้วยการกดคีย์ F1 โปรแกรมจะถามผู้ใช้ถึงชื่อของไดร์ฟและ  
ไดเร็คทอรีของไฟล์ข้อมูลและไฟล์ Config โดยมีรูปแบบดังนี้

ชื่อไดร์ฟ:\ชื่อไดเร็คทอรีที่ 1\ชื่อไดเร็คทอรีที่ 2\....\ชื่อไดเร็คทอรีสุดท้าย

(Enter)



### รูปที่ 3.26 แสดงหน้าจอเมนู Main Menu ของโปรแกรม View

ข้อความทั้งหมดสามารถมีความยาวได้จำนวน 20 ตัวอักษร และจะอ่านพารามิเตอร์จากไฟล์ Config พร้อมกับแสดงชื่อของไฟล์ข้อมูลที่มีอยู่ไดเรกทอรีดังกล่าวเพื่อให้ผู้ใช้เลือก ด้วยแถบสว่าง (High Light Bar) ปรากฏบนชื่อของไฟล์ที่ต้องการ เมื่อผู้ใช้เลือกไฟล์ได้แล้ว โปรแกรมจะอ่านข้อมูลที่บันทึกจากไฟล์ข้อมูล พร้อมทั้งแสดงจำนวนชุดของข้อมูลที่อ่านจากไฟล์ให้ผู้ใช้รู้ว่า มีจำนวนเท่าใด เมื่อผู้ใช้ต้องการดูข้อมูล ทำได้โดยกดคีย์ F2 โปรแกรมจะเข้าสู่ View Data Menu การแสดงข้อมูลจะมีลักษณะดังแสดงในรูปที่การควบคุมการแสดงผลทำได้ด้วยคีย์ที่แสดงไว้ในบรรทัดล่างสุด ดังแสดงในรูปที่ 3.27

ผู้ใช้สามารถนำข้อมูลที่บันทึกได้นี้ จัดทำเป็นรายงานด้วยการสร้างเป็นไฟล์เอกสาร (Text File) ด้วยการกดคีย์ F3 โปรแกรมจะแสดงชื่อไฟล์ข้อมูลเพื่อให้ผู้ใช้เลือกเช่นเดียวกับการโหลดไฟล์เมื่อเลือกไฟล์ได้แล้ว โปรแกรมจะอ่านข้อมูลจากไฟล์ดังกล่าว พร้อมทั้งบอกให้รู้ถึงจำนวนชุดของข้อมูลที่มีอยู่ในไฟล์ และจะถามถึงชุดของข้อมูลเริ่มต้น (First Sample) และชุดของข้อมูลสิ้นสุด (Last Sample) ที่จะนำไปสร้างเป็นไฟล์เอกสาร เมื่อป้อนเสร็จแล้ว จะถามถึงชื่อของไดเรกทอรีและไดเรกทอรีที่จะใช้เก็บไฟล์เอกสาร โดยมีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ,  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

----- VIEW LOGGING DATA -----				
STATION NAME : TERMINAL_NAME1			TOTAL SAMPLE : 150	
TIME STAMP : 03-AUG-1993 : 13:04:03			SAMPLE NUMBER : 23	
DEVICE	MEASURE	AVERAGE	DIFFERENT	UNIT
ANALOG_IP1	0.0000	0.0000	0.0000	UNIT1
ANALOG_IP2	12.2345	12.4234	-1.2000	UNIT2
ANALOG_IP3	230.0980	231.0082	0.2876	UNIT3
ANALOG_IP4	1205.0000	1204.5000	-0.3428	UNIT4
ANALOG_IP5	9999.9999	9999.9888	-0.0001	UNIT5
ANALOG_IP6	1.0000E+01	1.0000E+01	99.0000	UNIT6
ANALOG_IP7	12.3456E+20	12.5000E+20	1.0000E+01	UNIT7
ANALOG_IP8	365.5673E-19	365.5000E-19	-1.5632E-02	UNIT8
ANALOG_IP9	9999.9999E+20	9999.9999E+20	-3.9999E+02	UNIT9
ANALOG_IP10	0.0000	0.0000	0.0000	UNIT10
ANALOG_IP11	12.2345	12.4234	-1.2000	UNIT11
ANALOG_IP12	230.0980	231.0082	0.2876	UNIT12
ANALOG_IP13	1205.0000	1204.5000	-0.3428	UNIT13
ANALOG_IP14	9999.9999	9999.9888	-0.0001	UNIT14
ANALOG_IP15	1.0000E+01	1.0000E+01	99.0000	UNIT15
ANALOG_IP16	12.3456E+20	12.5000E+20	1.0000E+01	UNIT16

-View next    -View previous    -Toggle page    Esc-back to main menu

รูปที่ 3.27 แสดงหน้าเมนู View Data ของโปรแกรม View

ชื่อไดรฟ์:\ชื่อไดเรกทอรีที่ 1\ชื่อไดเรกทอรีที่ 2\...\ชื่อไดเรกทอรีสุดท้าย

(Enter)

จากนั้นจึงเริ่มสร้างไฟล์เอกสาร ไฟล์เอกสารที่สร้างขึ้นนี้ จะใช้ชื่อเดียวกับไฟล์ข้อมูล แต่จะเปลี่ยนนามสกุลจาก .S# เป็น .RP แทน ดังนี้

TU ตัวที่ 1 ใช้เป็น .RP1  
 TU ตัวที่ 2 ใช้เป็น .RP2  
 TU ตัวที่ 3 ใช้เป็น .RP3  
 TU ตัวที่ 4 ใช้เป็น .RP4  
 TU ตัวที่ 5 ใช้เป็น .RP5  
 TU ตัวที่ 6 ใช้เป็น .RP6

ไฟล์เอกสารที่สร้างขึ้นจะแบ่งออกเป็นหน้า (Page) หน้าละ 25 บรรทัด มีความยาวบรรทัดละ 80 ตัวอักษร โดยชุดของข้อมูล 1 ชุด (1 Sample) จะสร้างเป็นไฟล์เอกสาร 2 หน้า หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรกจะเป็นของแขนงที่ 1-16 หน้าที่สองเป็นของแขนงที่ 17-32 มีลักษณะของข้อความตั้ง  
แสดงในภาคผนวก ข



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### บทที่ 4

จากการนำไปทดลองใช้งานจริง ในการเก็บบันทึกข้อมูลความสามารถในการสังเคราะห์แสงของพืช (Photosynthesis Rate) ในเรือนกระจก (Green House) ที่ถูกควบคุมสภาพแวดล้อม ความสามารถในการสังเคราะห์แสงจะเป็นตัวที่บอกถึงการเจริญเติบโตของพืชที่มีต่อสภาพแวดล้อมที่กำหนด ในระบบการบันทึกข้อมูลดังกล่าวประกอบด้วย

- เซ็นเซอร์ตรวจวัดปริมาณก๊าซคาร์บอนไดออกไซด์ ( $\text{CO}_2$  Analyzer) มีหน่วยเป็น ppm สามารถวัดปริมาณก๊าซได้จาก 0 ถึง 3000 เข้าพุดเป็นแรงดันไฟตรงจาก 0 โวลต์ ถึง 5 โวลต์ จำนวน 1 ตัว
- เซ็นเซอร์ตรวจวัดปริมาณของไอน้ำ ( $\text{H}_2\text{O}$  Analyzer) มีหน่วยเป็น mmol/mol สามารถวัดปริมาณไอน้ำในอากาศได้จาก 0 ถึง 75 เข้าพุดเป็นแรงดันไฟตรงจาก 0 โวลต์ ถึง 5 โวลต์ จำนวน 1 ตัว
- เซ็นเซอร์ตรวจวัดความชื้นสัมพัทธ์ในอากาศ (Relative Humidity Sensor) มีหน่วยเป็น % สามารถวัดปริมาณความชื้นในอากาศได้จาก 0 ถึง 100 เข้าพุดเป็นแรงดันไฟตรงจาก 0 โวลต์ ถึง 4.5 โวลต์ จำนวน 1 ตัว
- เซ็นเซอร์ตรวจวัดอุณหภูมิของใบไม้ (Leaf Temperature Thermocouple) มีหน่วยเป็น Celcius สามารถวัดอุณหภูมิของใบไม้ได้จาก 0 ถึง 60 เข้าพุดเป็นแรงดันไฟตรงจาก 0 โวลต์ ถึง 2.5 โวลต์ จำนวน 1 ตัว
- เซ็นเซอร์ตรวจวัดอุณหภูมิในแชมเบอร์ (Chamber Temperature Thermocouple) มีหน่วยเป็น Celcius สามารถวัดอุณหภูมิในอากาศได้จาก 0 ถึง 50 เข้าพุดเป็นแรงดันไฟตรงจาก 0 โวลต์ ถึง 2.5 โวลต์ จำนวน 1 ตัว
- เซ็นเซอร์ตรวจวัดควอนตัมของแสง (Quantum Sensor) มีหน่วยเป็น  $\text{E S}^{-1}\text{m}^{-2}$  สามารถวัดปริมาณควอนตัมของแสงได้จาก 0 ถึง 2500 เข้าพุดเป็นแรงดันไฟตรงจาก 0 โวลต์ ถึง 1.0 โวลต์ จำนวน 1 ตัว

- เซ็นเซอร์ตรวจวัดความชื้นในดิน (Soil Humidity Sensor) มีหน่วยเป็น % สามารถวัดปริมาณความชื้นในดินได้จาก 0 ถึง 100 แอ้มพุทเป็นแรงดันไฟตรง จาก 0 โวลต์ ถึง 4.5 โวลต์ จำนวน 3 ตัว กำหนดปริมาณความชื้นสูงสุดและต่ำสุดในดิน ไว้ที่ 25 และ 70
- ระบบให้น้ำพืช ควบคุมปริมาณน้ำด้วยมอเตอร์ไฟฟ้ากระแสสลับ (AC Motor) ขนาดแรงดันขดลวด 220 โวลต์ จำนวน 3 ตัว
- ระบบควบคุมอุณหภูมิภายในเรือนกระจก ประกอบด้วย พัดลมระบายอากาศ (Air Flow Fan) ควบคุมการเปิด ปิด ด้วยรีเลย์ขนาดแรงดันขดลวด 12 โวลต์ จำนวน 4 ตัว และเครื่องฉีดละอองน้ำ (Water Vapor Injector) ควบคุมการเปิด ปิด ด้วยรีเลย์ขนาดแรงดันขดลวด 12 โวลต์ จำนวน 4 ชุด
- ระบบควบคุมความเข้มแสง (Shutter) แบบปรับระดับความเข้มของแสงได้ 10 ระดับ จาก 0 ถึง 100 เปอร์เซนต์ มีอินพุทสำหรับควบคุมด้วยแรงดันไฟตรง 12 โวลต์จำนวน 10 อินพุท และแอ้มพุทเป็นไมโครสวิทช์สำหรับแสดงตำแหน่งของความเข้มที่เลือกใช้จำนวน 10 แอ้มพุท จำนวน 1 ชุด

การต่อใช้งานระหว่าง Remote Data Logger กับระบบที่ต้องการสามารถแสดงได้ในรูปที่ 4.1 และต้องกำหนดค่าของพารามิเตอร์ใน Configuration File ดังนี้

- อนาล็อกอินพุทแชนแนลที่ 1

Config : USE

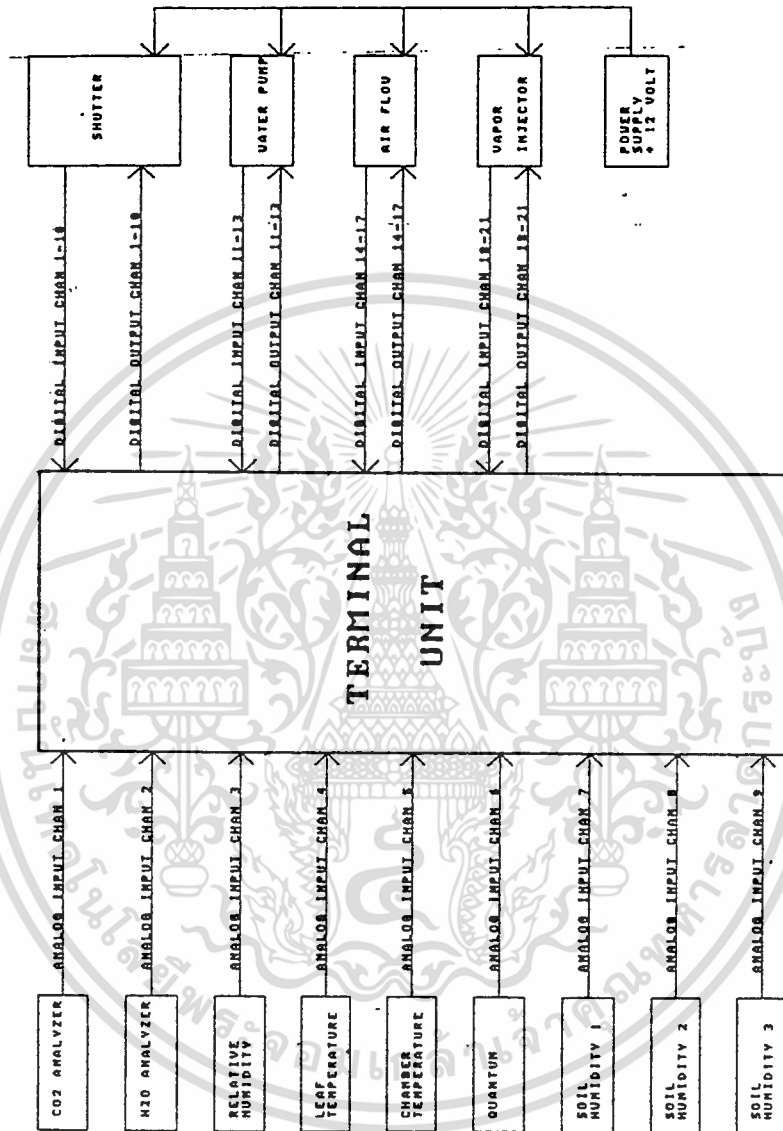
Device Label : VALUE\_OF\_CO<sub>2</sub>

Device Unit : ppm

Multiplier : 1.4648

หาได้จาก แอ้มพุทสูงสุดที่เซ็นเซอร์วัดได้หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ 3000/2048 คูณด้วยอัตราส่วนค่าแรงดันสูงสุดแอ้มพุทของเซ็นเซอร์ กับ ค่าแรงดันอินพุทสูงสุดของ Terminal Unit หรือเท่ากับ 5/5

Offset : 0



รูปที่ 4.1 แสดงการต่อใช้งานระบบกับ Green House

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า, ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Alarm : Off

- อนาล็อกอินพุทเซนแนลที่ 2

Config : USE

Device Label : VALUE\_OF\_H<sub>2</sub>O

Device Unit : mmol/mol

Multiplier :  $36.6210 \times 10^{-03}$

หาได้จาก เ้าพุทสูงสุดที่เซ็นเซอร์วัดได้ หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ  $(75/2048) \times (5/5)$

Offset : 0

Alarm : Off

- อนาล็อกอินพุทเซนแนลที่ 3

Config : USE

Device Label : VALUE\_OF\_RH

Device Unit : %

Multiplier :  $43.9453 \times 10^{-03}$

หาได้จาก เ้าพุทสูงสุดที่เซ็นเซอร์วัดได้ หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ  $(100/2048) \times (4.5/5)$

Offset : 0

Alarm : Off

- อนาล็อกอินพุทเซนแนลที่ 4

Config : USE

Device Label : LEAF\_TEMP

Device Unit : CELCIUS

Multiplier :  $14.6484 \times 10^{-03}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หาได้จาก ให้นำค่าสูงสุดที่เซ็นเซอร์วัดได้หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ  $(60/2048) \times (2.5/5)$

Offset : 0

ALarm : Off

- อนุาล็อกอินพุทเซนแนลที่ 5

Config : USE

Device Label : CHAMBER\_TEMP

Device Unit : CELCIUS

Multiplier :  $12.2070 \times 10^{-03}$

หาได้จาก ให้นำค่าสูงสุดที่เซ็นเซอร์วัดได้หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ  $(50/2048) \times (2.5/5)$

Offset : 0

ALarm : Off

- อนุาล็อกอินพุทเซนแนลที่ 5

Config : USE

Device Label : QUANTUM

Device Unit : uE

Multiplier :  $244.4140 \times 10^{-03}$

หาได้จาก ให้นำค่าสูงสุดที่เซ็นเซอร์วัดได้หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ  $(2500/2048) \times (1/5)$

Offset : 0

ALarm : Off

- อนุาล็อกอินพุทเซนแนลที่ 6

Config : USE

Device Label : SOIL\_HUMIDITY\_1

Device Unit : %

Multiplier :  $43.9453 \times 10^{-03}$

หาได้จาก ค่าสูงสุดที่เซ็นเซอร์วัดได้หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ  $(100/2048) \times (4.5/5)$

Offset : 0

ALarm : 0n

Maximum : 70

Minimum : 20

- อนาล็อกอินพุตแชนแนลที่ 7

Config : USE

Device Label : SOIL\_HUMIDITY\_2

Device Unit : %

Multiplier :  $43.9453 \times 10^{-03}$

หาได้จาก ค่าสูงสุดที่เซ็นเซอร์วัดได้หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ  $(100/2048) \times (4.5/5)$

Offset : 0

ALarm : 0n

Maximum : 70

Minimum : 20

- อนาล็อกอินพุตแชนแนลที่ 8

Config : USE

Device Label : SOIL\_HUMIDITY\_3

Device Unit : %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Multiplier :  $43.9453 \times 10^{-09}$

หาได้จาก แอานพุทสูงสุดที่เซ็นเซอร์วัดได้ หารด้วย ค่าสูงสุดที่แสดงบนจอเมื่อ Multiplier เท่ากับ 1 (2048) หรือเท่ากับ  $(100/2048) \times (4.5/5)$

Offset : 0

ALarm : On

Maximum : 70

Minimum : 20

การควบคุมอุปกรณ์ต่างในระบบที่ทำงานด้วยแรงดันไฟฟ้ากระแสสลับ 220 โวลท์ จำเป็นจะต้องใช้วงจรไดร์เวอร์ (Driver) ซึ่งประกอบด้วยทรานซิสเตอร์ (Transister) และ รีเลย์ (Relay) ดังแสดงในรูปที่ 4.2 เพื่อต่อเชื่อมวงจรระหว่างแรงดันต่ำ กับ แรงดันสูง และกำหนดค่าพารามิเตอร์ใน Configuration File ดังนี้

ดิจิตอลเข้าพุทเซนแนลที่ 1

Config : USE

Device Label : SHUTTER

On Message : OPEN\_10%

Off Message : CLOSE

ดิจิตอลเข้าพุทเซนแนลที่ 2

Config : USE

Device Label : SHUTTER

On Message : OPEN\_20%

Off Message : CLOSE

ดิจิตอลเข้าพุทเซนแนลที่ 3

Config : USE

Device Label : SHUTTER

On Message : OPEN\_30%

Off Message : CLOSE

**ดิจิตอลเข้าพวฯแซนแนลที่ 4**

Config : USE

Device Label : SHUTTER

On Message : OPEN\_40%

Off Message : CLOSE

**ดิจิตอลเข้าพวฯแซนแนลที่ 5**

Config : USE

Device Label : SHUTTER

On Message : OPEN\_50%

Off Message : CLOSE

**ดิจิตอลเข้าพวฯแซนแนลที่ 6**

Config : USE

Device Label : SHUTTER

On Message : OPEN\_60%

Off Message : CLOSE

**ดิจิตอลเข้าพวฯแซนแนลที่ 7**

Config : USE

Device Label : SHUTTER

On Message : OPEN\_70%

Off Message : CLOSE

**ดิจิตอลเข้าพวฯแซนแนลที่ 8**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Config : USE  
 Device Label : SHUTTER  
 On Message : OPEN\_80%  
 Off Message : CLOSE

ดิจิตอลเข้าพทแซนแนลที่ 9

Config : USE  
 Device Label : SHUTTER  
 On Message : OPEN\_90%  
 Off Message : CLOSE

ดิจิตอลเข้าพทแซนแนลที่ 10

Config : USE  
 Device Label : SHUTTER  
 On Message : OPEN\_100%  
 Off Message : CLOSE

ดิจิตอลเข้าพทแซนแนลที่ 11

Config : USE  
 Device Label : WATER\_PUMP#1  
 On Message : PUMPING  
 Off Message : STOP

ดิจิตอลเข้าพทแซนแนลที่ 12

Config : USE  
 Device Label : WATER\_PUMP#2  
 On Message : PUMPING  
 Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 13

Config : USE  
 Device Label : WATER\_PUMP#3  
 On Message : PUMPING  
 Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 14

Config : USE  
 Device Label : AIR\_FLOW\_FAN#1  
 On Message : FLOW  
 Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 15

Config : USE  
 Device Label : AIR\_FLOW\_FAN#2  
 On Message : FLOW  
 Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 16

Config : USE  
 Device Label : AIR\_FLOW\_FAN#3  
 On Message : FLOW  
 Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 17

Config : USE  
 Device Label : AIR\_FLOW\_FAN#4  
 On Message : FLOW

Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 18

Config : USE

Device Label : VAPOR\_INJECTOR1

On Message : INJECT

Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 19

Config : USE

Device Label : VAPOR\_INJECTOR2

On Message : INJECT

Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 20

Config : USE

Device Label : VAPOR\_INJECTOR3

On Message : INJECT

Off Message : STOP

ดิจิตอลเข้าพุทเซนแนลที่ 21

Config : USE

Device Label : VAPOR\_INJECTOR4

On Message : INJECT

Off Message : STOP

การตรวจสอบสถานะของอุปกรณ์ในระบบ สำหรับตัวปรับความเข้มแสงจะมีสวิทซ์  
แสดงตำแหน่งของการเปิด ปิดอยู่ภายในแล้ว ส่วนอุปกรณ์ตัวอื่นๆ จำเป็นจะต้องใช้หน้าสัมผัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Contact) ของรีเลย์มาทำหน้าที่บอกถึงสถานะของอุปกรณ์นั้นๆ และเนื่องจากระดับแรงดันต่างกัน จึงต้องใช้วงจรแบ่งแรงดัน (Voltage Divider) เพื่อลดแรงดันจาก 12 โวลต์เป็นแรงดันในช่วง ทีทีแอล (TTL) ดังแสดงในรูปที่ 4.2 และกำหนดค่าพารามิเตอร์ใน Configuration File ดังนี้

#### ดิจิตอลอินพุทแชนแนลที่ 1

Config : USE  
 Device Label : SHUTTER  
 On Message : 10%\_OPEN  
 Off Message : CLOSE

#### ดิจิตอลอินพุทแชนแนลที่ 2

Config : USE  
 Device Label : SHUTTER  
 On Message : 20%\_OPEN  
 Off Message : CLOSE

#### ดิจิตอลอินพุทแชนแนลที่ 3

Config : USE  
 Device Label : SHUTTER  
 On Message : 30%\_OPEN  
 Off Message : CLOSE

#### ดิจิตอลอินพุทแชนแนลที่ 4

Config : USE  
 Device Label : SHUTTER  
 On Message : 40%\_OPEN  
 Off Message : CLOSE

### ติจิตอลอินพุทเซนแนลที่ 5

Config : USE  
 Device Label : SHUTTER  
 On Message : 50%\_OPEN  
 Off Message : CLOSE

### ติจิตอลอินพุทเซนแนลที่ 6

Config : USE  
 Device Label : SHUTTER  
 On Message : 60%\_OPEN  
 Off Message : CLOSE

### ติจิตอลอินพุทเซนแนลที่ 7

Config : USE  
 Device Label : SHUTTER  
 On Message : 70%\_OPEN  
 Off Message : CLOSE

### ติจิตอลอินพุทเซนแนลที่ 8

Config : USE  
 Device Label : SHUTTER  
 On Message : 80%\_OPEN  
 Off Message : CLOSE

### ติจิตอลอินพุทเซนแนลที่ 9

Device Label : SHUTTER  
 On Message : 90%\_OPEN  
 Off Message : CLOSE

ดิจิทัลอินพุทเซนแนลที่ 10

Config : USE  
 Device Label : SHUTTER  
 On Message : 100%\_OPEN  
 Off Message : CLOSE

ดิจิทัลอินพุทเซนแนลที่ 11

Config : USE  
 Device Label : WATER\_PUMP#1  
 On Message : ON  
 Off Message : OFF

ดิจิทัลอินพุทเซนแนลที่ 12

Device Label : WATER\_PUMP#2  
 On Message : ON  
 Off Message : OFF

ดิจิทัลอินพุทเซนแนลที่ 13

Config : USE  
 Device Label : WATER\_PUMP#3  
 On Message : ON  
 Off Message : OFF

ดิจิทัลอินพุทเซนแนลที่ 14

Config : USE  
 Device Label : AIR\_FLOW\_FAN#1  
 On Message : ON  
 Off Message : OFF

ดิจิตอลอินพุทเซนแนลที่ 15

Config : USE  
 Device Label : AIR\_FLOW\_FAN#2  
 On Message : ON  
 Off Message : OFF

ดิจิตอลอินพุทเซนแนลที่ 16

Config : USE  
 Device Label : AIR\_FLOW\_FAN#3  
 On Message : ON  
 Off Message : OFF

ดิจิตอลอินพุทเซนแนลที่ 17

Config : USE  
 Device Label : AIR\_FLOW\_FAN#4  
 On Message : ON  
 Off Message : OFF

ดิจิตอลอินพุทเซนแนลที่ 18

Config : USE  
 Device Label : VAPOR\_INJECTOR1  
 On Message : ON  
 Off Message : OFF

ดิจิตอลอินพุทเซนแนลที่ 19

Config : USE  
 Device Label : VAPOR\_INJECTOR2  
 On Message : ON

Off Message : OFF

ดิจิตอลอินพุทแชนแนลที่ 20

Config : USE

Device Label : VAPOR\_INJECTOR3

On Message : ON

Off Message : OFF

ดิจิตอลอินพุทแชนแนลที่ 21

Config : USE

Device Label : VAPOR\_INJECTOR4

On Message : ON

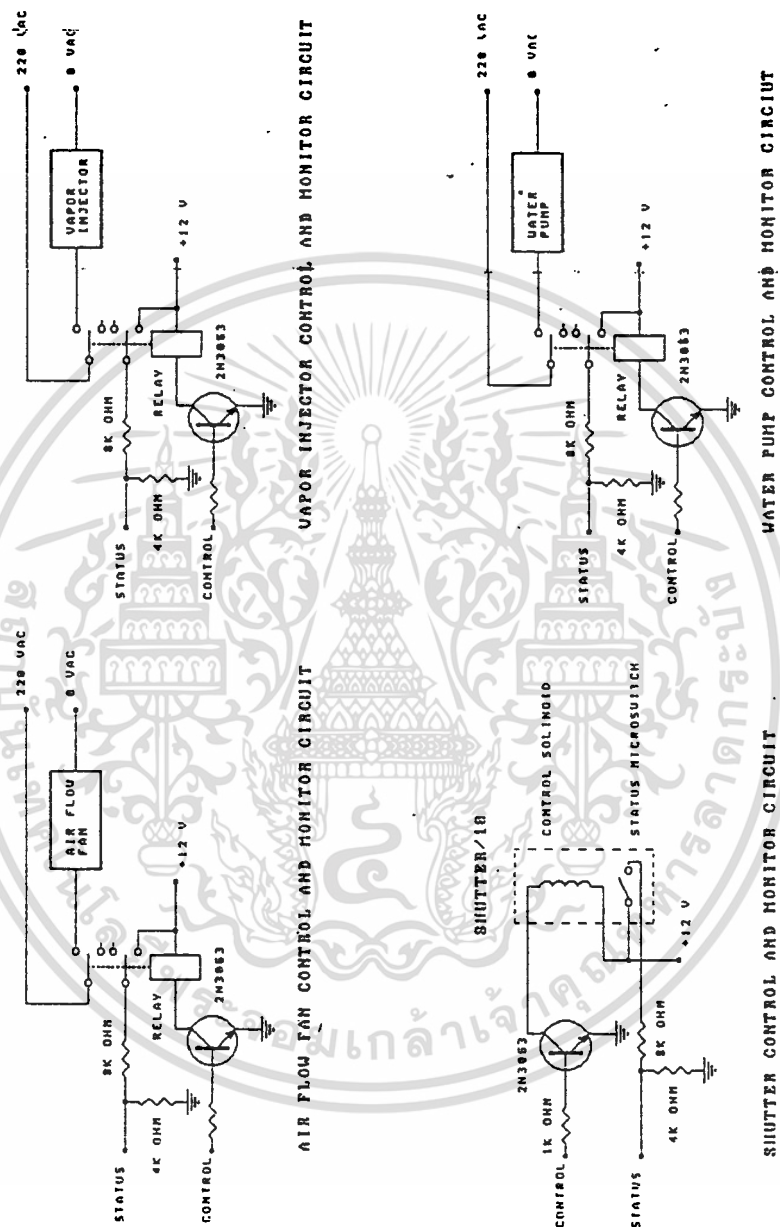
Off Message : OFF

กำหนดค่าพารามิเตอร์ใน Station Configuration ดังนี้

Station name : GREEN\_HOUSE#1

Logging Period : 1

สำหรับแชนแนลที่ไม่ได้ใช้งานที่เหลือ กำหนดใน Config ให้เป็น NOT USE



รูปที่ 4.2 แสดงวงจรต่อเชื่อมกับอุปกรณ์ในระบบที่ทดลองใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

จากการทดลองใช้งานโครงการนี้ พบว่าสามารถพัฒนาให้จำนวนของ Terminal Unit มีมากกว่า 6 Unit ได้ แต่ก็จะทำให้ช่วงเวลาของ Logging Period ซึ่งจากเดิมมีค่าต่ำสุดที่ 3 วินาที เพิ่มขึ้นตามจำนวนของ Terminal Unit ที่เพิ่มขึ้นด้วย คือ 1 Unit จะเพิ่มขึ้นประมาณ 500 milli Second และสามารถลดช่วงเวลาดังกล่าวลงได้เมื่อไม่มีความจำเป็นที่จะต้องใช้งาน Terminal Unit ทั้งหมด ในบางกรณีที่ต้องการใช้งานกับเซ็นเซอร์แบบอื่นๆ เช่น เครื่องวัดปริมาณน้ำฝน (Tripping Blunket) ที่มีเข้าพุทเป็นพัลส์ (Pulse) หรือ การควบคุมความเร็วรอบของมอเตอร์ที่เป็นแบบ ลินีเยร์ (Linear Control) ที่มีอินพุทเป็นแบบอนาล็อก ก็สามารถเพิ่ม วงจรนับเหตุการณ์ (Event Counter) และวงจรเข้าพุทแบบอนาล็อก (Analog Output) ได้ เนื่องจากได้ออกแบบให้บัส (Bus) ของ Terminal Unit สามารถต่อขยายเพื่อใช้งานผ่านคอนเน็คเตอร์ (Connector) ในส่วนของวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (ADC) ที่ใช้อยู่เป็นแบบที่ใช้สัญญาณนาฬิกาภายใน ทำให้ไม่สามารถกำหนดช่วงเวลาที่ใช้แปลงข้อมูล (Conversion Time) ได้ ซึ่งอาจทำให้เกิดการรบกวนจากความถี่ไฟฟ้ากระแสสลับ 50 เฮิตซ์ได้ การแก้ไขทำได้โดยเลือกใช้ ADC แบบที่ใช้สัญญาณนาฬิกาจากภายนอก (External Clock) แทน และกำหนดให้ช่วงเวลาที่ใช้แปลงข้อมูลให้มีค่าเป็นจำนวนเท่าของความถี่ 50 เฮิตซ์ (20 ms)

การเก็บบันทึกข้อมูลของโปรแกรม ReLog จะใช้เนื้อที่ในดิสก์ 76 ไบต์ ต่อจำนวนข้อมูลที่บันทึก 1 ตัวอย่าง (1 Sample) ในส่วนของเข้าพุทแบบดิจิตอล สามารถเพิ่มความสามารถในการควบคุมการเปิดปิดแบบอัตโนมัติได้ โดยควบคุมจากระดับของค่าวัดทางอนาล็อกอินพุท เนื่องจากวงจรที่ใช้กำหนดช่วงเวลากการบันทึกข้อมูลนั้น ใช้วงจร Counter แบบธรรมดา ทำให้ช่วงเวลา Logging Time มีความคลาดเคลื่อนอยู่บ้าง ซึ่งแก้ไขได้ด้วยการใช้วงจร RTC (Real Time Clock) เช่น ชิพเบอร์ MM58174 เป็นวงจรกำหนดช่วงเวลาแทน

การใช้งานโปรแกรม View มีข้อจำกัดอยู่บ้างคือ เนื้อที่ของหน่วยความจำในขณะที่ใช้งานโปรแกรมไม่พอที่จะโหลดไฟล์ข้อมูลได้ทั้งหมด ซึ่งบางครั้งเกิดจากโปรแกรมประเภทเรสซิเด้น (Resident Program) แก้ไขด้วยการโหลดโปรแกรม View ใช้งานเป็นโปรแกรมแรก

การสร้างไฟล์เอกสาร (Text File) จะใช้เนื้อที่ในดิสก์ 363 ไบต์สำหรับส่วนที่เป็น Header Page และใช้เนื้อที่ 2351 ไบต์สำหรับส่วนที่เป็นข้อมูล 1 ตัวอย่าง (1 Sample Data for 2 Data Pages) ไฟล์ที่สร้างขึ้นนี้สามารถนำไปพิมพ์เป็นรายงาน (Text Report) ได้ด้วยคำสั่ง Type ของดอส หรือโปรแกรม Word Processor อื่นๆ เช่น Chula Word , Ratchvithi Word เป็นต้น รวมทั้งสามารถใช้โปรแกรมประเภท Spread Sheet เช่น Lotus อ่านข้อมูลจากไฟล์เอกสารนี้ เพื่อเปลี่ยนรูปแบบรายงานให้เป็นรูปแบบ กราฟ (Graph Report) ได้เช่นเดียวกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

(1) Intel Corporation THE 8086 FAMILY USER'S MANUAL Intel Corpora-  
tion Literature Department 1979

นกุล กระจาย การเขียนโปรแกรม และ ประมวลผลข้อมูลด้วย เทอร์โบ ปาสคาล  
บริษัท ซีเอ็ดยูเคชั่น จำกัด

บริษัท ซีเอ็ดยูเคชั่น จำกัด คู่มือไอซี ชิพทันฟอร์ท และหน่วยความจำ บริษัท ซีเอ็ดยูเคชั่น  
จำกัด 2529

วารสารไมโครคอมพิวเตอร์ฉบับพิเศษ ยูทิลิตี้ (UTILITY) บริษัท ซีเอ็ดยูเคชั่น จำกัด

สุรศักดิ์ สงวนพงษ์ เทคนิคการเขียนโปรแกรมขั้นสูง แอดวานซ์ เทอร์โบ ปาสคาล บริษัท ซี  
-เอ็ดยูเคชั่น จำกัด

Borland International. TURBO PASCAL 4.0 REFERENCE MANUAL

Borland International 1987

Burr-Brown Inc. ACQUISITION DEVICE DATA Burr-Brown Inc. 1989

IBM Inc. PC/XT HARDWARE REFERENCE MANUAL

James W.Coffron Z80 APPLICATION Sybex Inc. 1983

Maxim Integrated Products MICROPROCESSER SUPERVISORY CIRCUIT

Maxim Integrated Products 1989

Motorola Inc. CMOS LOGIC DATA Motorola Inc. Rev3 1991

Motorola Inc. LINEAR/SWITCH MODE VOLTAGE REGULATER HANDBOOK 4th  
Edition Motorola Inc. 1989

Motorola Inc. TELECOMMUNICATION DEVICE DATA Motorola Inc. Rev2

1989

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor Corporation. DATA ACQUISITION LINEAR DEVICE

National Semiconductor Corp. 1989 edition

Russell Rector - George Alexy THE 8086 BOOK Osborne/McGraw-Hill 1980

Zilog Inc. ZILOG INTELLIGENT PERIPHERAL CONTROLLERS Zilog Inc.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

write by ronachai hachai

```
program edcon;
```

```
uses hdlcu3,myscreen,mywin,keyboard,dos,crt;
```

```
type dig = set of 0..14;
```

```
const main_key : array [1..14] of string[18] =
```

```
    ('F1','-Measure ','F2','-Status ','F3','-Control ','F4','-
```

```
Station ',
```

```
    'F5','-Save ','F6','-Load ','ESC','-Quit');
```

```
    odd : dig = [1,3,5,7,9,11,13,15];
```

```
    coor : array [1..2] of integer = (3,25);
```

```
    editwin : array [1..4] of integer =  
        (20,9,60,22);
```

```
var count : integer;
```

```
    quit : boolean;
```

```
    enbedit : boolean;
```

```
    path : string[12];
```

```
    ch : char;
```

```
procedure Showkey;
```

```
type dig = set of 0..10;
```

```
const key : array [1..12] of string[17] =
```

```
    (#24,#25,' -Sel topic ',#27,#26,' -Sel chan ',
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใ้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า  
'PgUp,PgDn',' -Sel station ','Enter',' -Edit ',  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'Esc',' -Back to main menu');

hid : dig = [1,2,4,5,7,9,11];

coor : array [1..2] of integer = (1,25);

var count : integer;

begin

    gotoxy(coor[1],coor[2]);
    for count := 1 to 10 do
        begin
            if count in hid then
                Setattr(highdisplay);
            else
                Setattr(lowdisplay);
            write(key[count]);
        end;
    Setattr(lowdisplay);

end;

function Int_str(i : integer) : stmax;

begin

    str(i,st);

    Int_str := st;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure Read_Dicfg;

const cfg_msg : array [1..9] of string[13] =

    (' STATION NAME', ' CONFIG', ' DEVICE LABEL',

     ' ON MESSAGE', ' OFF MESSAGE', ' ALARM',

     'ALARM MESSAGE', ' STATUS CHKBK', ' CHECK CHAN');

```

```

    enb = 'USE ';

    dis = 'NOT USE';

    on = 'ON ';

    off = 'OFF';

var curcho, prevcho, curchan, curstat : integer;

    quit : boolean;

    xcon, ycon : integer;

    dumst : stmax;

    dumint : integer;

```

```

procedure Wrtchoice;

```

```

var count : integer;

```

```

begin

```

```

    gotoxy(cord[1], cord[2]-5);

```

```

    write(cfg_msg[1], ' : ');

```

```

    for count := 2 to maxdicho do

```

```

        begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และเผยแพร่ไปยังสื่อออนไลน์ใดๆ ทุกครั้งที่มีการนำไปใช้

```

            gotoxy(cord[1], cord[2]+(count-2));

```

```

        write(cfg_msg[count], ' : ');

    end;

end;

```

```

procedure Wrtbar(var curr : integer;
                var prev : integer);

var xb,yb : integer;

begin
    xb := cord[1];
    if (prev <> 1) then
    begin
        yb := cord[2]+(prev-2);
        gotoxy(xb,yb);
        write(cfg_msg[prev]);
    end
    else
    begin
        gotoxy(cord[1],cord[2]-5);

        write(cfg_msg[1]);

    end;

    Setattr(reversehigh);

    if (curr <> 1) then

    begin

```

```

        yb := cord[2]+(curr-2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ,  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ **gotoxy(xb,yb);** ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        write(cfg_msg[curr]);

    end
else
    begin
        gotoxy(cord[1],cord[2]-5);
        write(cfg_msg[1]);
    end;
Setattr(lowdisplay);
end;

procedure Wrt_DIcfg(var rec : di);
var count : integer;
    xc,yc : integer;
begin
    xc := cord[1]+16;
    yc := cord[2];
    with rec do
        begin
            gotoxy(xc,yc);

            case di_enb of

                true : write(enb);

                false : write(dis);

            end;

            gotoxy(xc,yc+1);write(di_lab,' ');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(xc,yc+3);write(off_msg,' ');

gotoxy(xc,yc+4);

case enb_alarm of

    true : write(on);

    false : write(off);

end;

gotoxy(xc,yc+5);write(alarm_msg,' ');

gotoxy(xc,yc+6);

case chk_back of

    true : write(enb);

    false : write(dis);

end;

gotoxy(xc,yc+7);

case chk_back of

    true : write(Int_str(chk_chan),' ');

    false : write('- ');

end;

end;

end;

begin

    clrscr;

    quit := false;

    curstat := 1;

    curchan := 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

curcho := 2;

Oncursor;

Offcursor;

prevcho := maxdicho;

Setattr(highdisplay);

gotoxy(25,1);write('< DIGITAL INPUT CONFIGURATION >');

Setattr(lowdisplay);

gotoxy(cord[1],cord[2]-3);write(' DIGITAL CHAN : ');

Showkey;

Wrtchoice;

repeat
with netw_rec[curstat] do
begin
gotoxy(cord[1]+16,4);
write(stat_lab,' ');

Wrt_Dlcfg(di_cfg[curchan]);

end;

gotoxy(cord[1]+16,cord[2]-3);write(curchan,' ');

Wrtbar(curcho,prevcho);

Readfunckey(key);

case key of
    Rt_key : if (curchan = maxdichan) then
                curchan := 1
            else
                curchan := curchan+1;
        end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิง curchan := curchan+1; นำไปใช้

```

Lt_key : if (curchan = 1) then
    curchan := maxdichan
else
    curchan := curchan-1;
Dn_key : if (curcho = maxdicho) then
    begin
        prevcho := curcho;
        curcho := 2;
    end
else
    begin
        prevcho := curcho;
        curcho := curcho+1;
    end;
Up_key : if (curcho = 2) then
    begin
        prevcho := curcho;
        curcho := maxdicho;
    end
else
    begin
        prevcho := curcho;
        curcho := curcho-1;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ,  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PgUp\_key : if (curstat = maxstation) then

```

curstat := 1

else

curstat := curstat+1;

PgDn_key : if (curstat = 1) then

curstat := maxstation

else

curstat := curstat-1;

Return_key : with netw_rec[curstat].di_cfg

[curchan] do

begin

xcon := whereX;

ycon := whereY;

gotoxy(xcon+2,ycon);

clrcol;

gotoxy(xcon+2,ycon);

case curcho of

2 : begin

Readfunc

(di_enb,enb,dis);

end;

3 : begin

dumst :=

Readstr(15);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ,  
 '') then  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

di_lab :=
dumst;

end;

4 : begin
dumst :=
Readstr(15);
if (dumst <>
'') then
on_msg :=
dumst;
end;
5 : begin
dumst :=
Readstr(15);
if (dumst <>
'') then
off_msg :=
dumst;
end;
6 : Readfunc
(enb_alarm,on,off);
7 : begin
if enb_alarm
then
dumst :=

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

di_lab :=
dumst;

end;

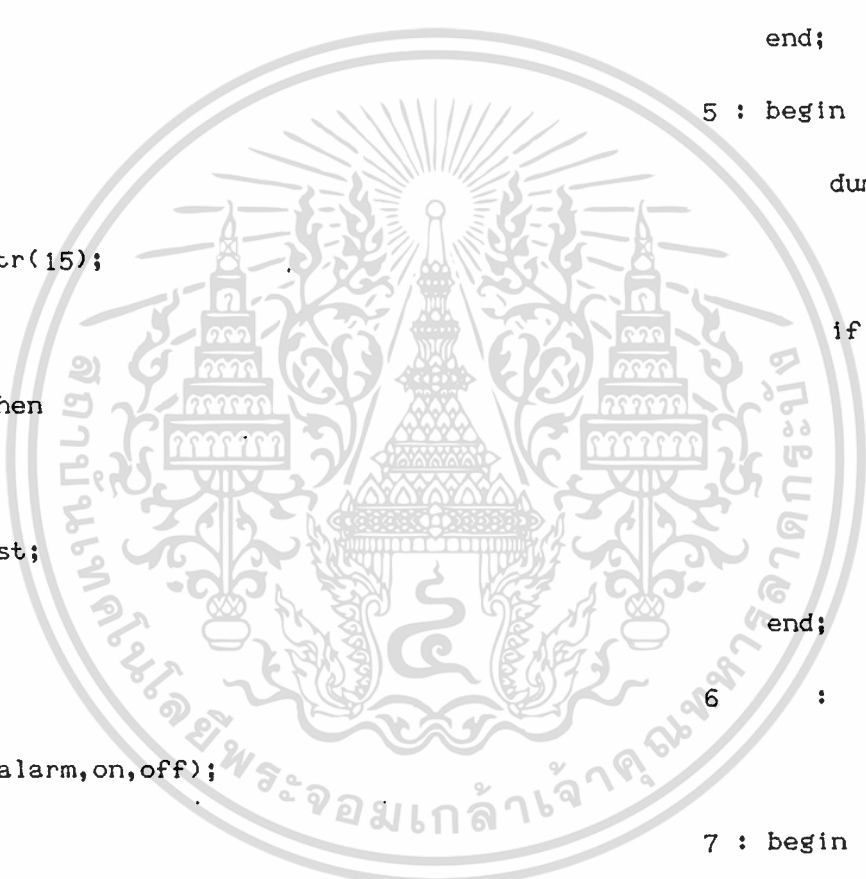
4 : begin
dumst :=
Readstr(15);
if (dumst <>
'') then
on_msg :=
dumst;
end;

5 : begin
dumst :=
Readstr(15);
if (dumst <>
'') then
off_msg :=
dumst;
end;

6 : Readfunc
(enb_alarm,on,off);

7 : begin
if enb_alarm
then
dumst :=

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
else
begin

```

case auto of

```

        true : write(on);

        false : write(off);

end;

gotoxy(xc,yc+5);

if auto then

    write(Int_str(auto_ai),' ')

else

    write(' - ');

    gotoxy(xc,yc+6);

    case auto of

        true : begin

            st := Realtrostr(on_lev);

            Delblank(st);

            write(st,' ');

        end;

        false : write(' - ');

    end;

    gotoxy(xc,yc+7);

case auto of

    true : begin

        st := Realtrostr(off_lev);

        Delblank(st);

        write(st,' ');

    end;

    false : write(' - ');

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการเรียนเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

end;

gotoxy(cord[1]+16,cord[2]-3);write(curchan,' ');

Wrtbar(curcho,prevcho);

Readfunckey(key);

case key of

    Rt_key : if (curchan = maxdochan) then

                curchan := 1

            else

                curchan := curchan+1;

    Lt_key : if (curchan = 1) then

                curchan := maxdochan

            else

                curchan := curchan-1;

    Dn_key : if (curcho = maxdicho) then

                begin

                    prevcho := curcho;

                    curcho := 2;

                end

            else

                begin

                    prevcho := curcho;

                    curcho := curcho+1;

                end;

    Up_key : if (curcho = 2) then

                begin

                    case curcho of

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(xc,yc);

case al_enb of

    true : write(enb);

    false : write(dis);

end;

```

```
Rt_key : if (curchan = maxaichan) then
```

```
    curchan := 1
```

```
else
```

```
    curchan := curchan+1;
```

```
Lt_key : if (curchan = 1) then
```

```
    curchan := maxaichan
```

```
else
```

```
    curchan := curchan-1;
```

```
Dn_key : if (curcho = maxaicho) then
```

```
begin
```

```
    prevcho := curcho;
```

```
    curcho := 2;
```

```
end
```

```
else
```

```
begin
```

```
    prevcho := curcho;
```

```
    curcho := curcho+1;
```

```
end;
```

```
Up_key : if (curcho = 2) then
```

```
begin
```

```
    prevcho := curcho;
```

```
    curcho := maxaicho;
```

```
end
```

```
else
```

```
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

```

prevcho := curcho;

curcho := curcho-1;

end;

PgUp_key : if (curstat = maxstation) then

curstat := 1

else

curstat := curstat+1;

PgDn_key : if (curstat = 1) then

curstat := maxstation

else

curstat := curstat-1;

Return_key : with netw_rec[curstat].ai_cfg

[curchan] do

begin

xcon := whereX;

ycon := whereY;

gotoxy(xcon+2,ycon);

clreol;

gotoxy(xcon+2,ycon);

case curcho of

2 : begin

Readfunc

(ai_enb,enb,dis);

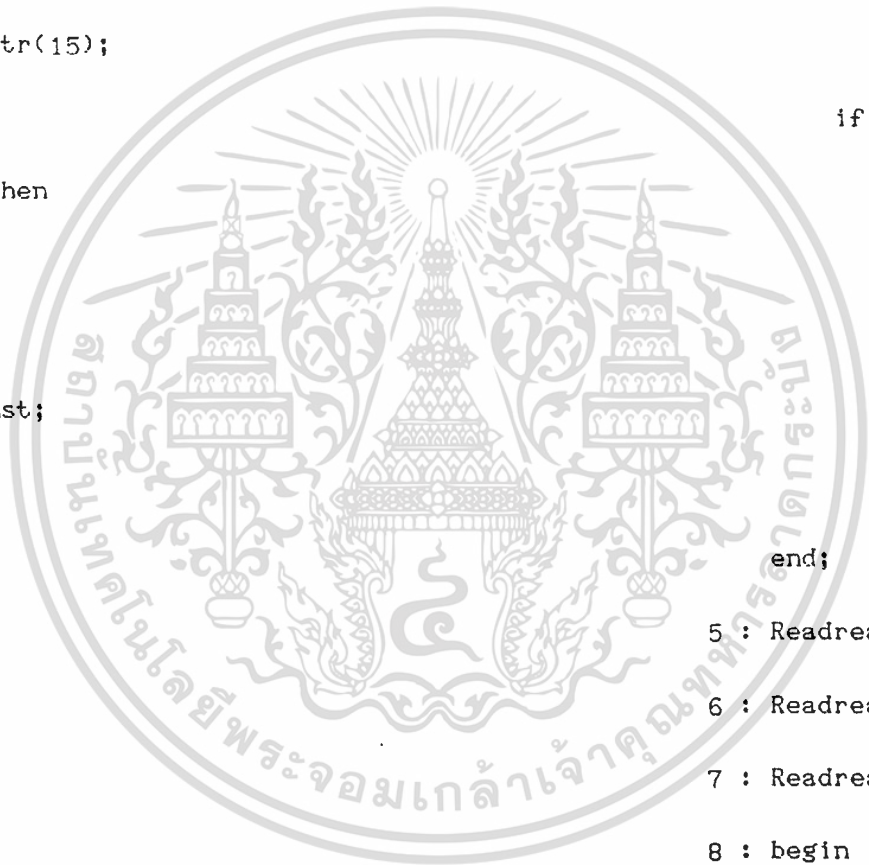
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร หากมีการนำไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dumst :=
Readstr(15);
if (dumst <>
'' ) then
begin
ai_lab := dumst;
end;
end;
4 : begin
dumst :=
Readstr(15);
if (dumst <>
'' ) then
begin
scal :=
= dumst;
end;
end;
5 : Readreal(multip);
6 : Readreal(offst);
7 : Readreal(thresh);
8 : begin
Readfunc
(averg,enb,dis);

```



เอกสารนี้เป็นทรัพย์สินของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังขอให้อัปเดตแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure Wrtchoice;
var count : integer;
begin
for count := 1 to 12 do
begin
gotoxy(st_cord[1],ch_cord[count]);
write(netw_rec[stat_no[count]].stat_lab);

```

```

begin
    gotoxy(st_cord[2],ch_cord[curr]);
    write(netw_rec[stat_no[curr]].log_time);
end;
Setattr(lowdisplay);
end;

```

```
begin
```

```

clrscr;
Oncursor;
Offcursor;
prevcho := maxcho;
Setattr(highdisplay);
gotoxy(27,1);write('< STATION CONFIGURATION >');
Setattr(lowdisplay);
gotoxy(st_cord[1],5);
write('STATION NAME');
gotoxy(st_cord[2],5);
write('LOGGING TIME (x3 SEC)');
gotoxy(5,25);
for curcho := 1 to 7 do
begin
    if (curcho in head) then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ,  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ **else** ่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Setattr(lowdisplay);

        write(st_key[curcho]);

    end;

    curcho := 1;

Wrtchoice;

repeat

    Wrtbar(curcho,prevcho);

    Readfunkey(key);

    case key of

        Dn_key : if (curcho = maxcho) then
                    begin
                        prevcho := curcho;
                        curcho := 1;
                    end
                else
                    begin
                        prevcho := curcho;
                        curcho := curcho+1;
                    end;

        Up_key : if (curcho = 1) then
                    begin
                        prevcho := curcho;
                        curcho := maxcho;
                    end
                else
                    begin
                        prevcho := curcho;
                        curcho := curcho-1;
                    end;
    end;
end repeat;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    prevcho := curcho;
    curcho := curcho-1;
end;

Return_key : begin
    ycor := ch_cord[curcho];
    if (curcho in st_cho) then
        begin
            xcor := st_cord[1];
            gotoxy(xcor-1,ycor);
            write(' ');
            gotoxy(xcor-1,ycor);
            dumst := Readstr(15);
            if (dumst <> '') then
                netw_rec[stat_no
[curcho]].stat_lab := dumst;
            end
        end
    else
        begin
            xcor := st_cord[2];
            gotoxy(xcor-1,ycor);
            write(' ');
            gotoxy(xcor-1,ycor);
            Readint(netw_rec[stat_no].stat_val);
            if (stat_val < 0) then
                stat_val := 0;
            end if;
            netw_rec[stat_no].stat_val := stat_val;
            netw_rec[stat_no].log_time,1000);
        end
    end if;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้ทำซ้ำหรือเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

end;

end;

until

key = Esc_key;

end;

procedure Saveconfig;
var count,result : integer;
begin
assign(cfg_fi,path+cfg_name);
{$I-}
reset(cfg_fi);
{$I+}
result := IOResult;
if (result <> 0) then
rewrite(cfg_fi);

Windowopen(10,5,70,7);

Windowbox(10,5,70,7);

gotoxy(10,1);

write(' Please wait Saving configuration file ');

begin

for count := 1 to maxstation do

begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไข{\$I-}และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(cfg_fi,netw_rec[count]);

{$!+}

result := 10result;

if result <> 0 then

begin

clrscr;

gotoxy(10,1);

Setattr(blinkhigh);

write(' file write error press Esc to

exit ');

repeat

Readfunckey(key);

until

(key = Esc_key);

Setattr(lowdisplay);

end;

end;

close(cfg_fi);

end;

Windowclose;

end;

procedure Readconfig;

label again;

var count : integer;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีสิทธิ์เปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rewrite(cfg_fi);

Windowopen(10,5,70,7);

Windowbox(10,5,70,7);

gotoxy(10,1);

write(' Please wait Saving configuration file ');

begin
    for count := 1 to maxstation do
        begin
            {$I-}
            write(cfg_fi,netw_rec[count]);
            {$I+}
            result := 10result;
            if result <> 0 then
                begin
                    clrscr;
                    gotoxy(10,1);
                    Setattr(blinkhigh);
                    write(' file write error press Esc to
exit ');

                    repeat

                        Readfunckey(key);

                    until

                        (key = Esc_key);

                    Setattr(lowdisplay);

                end;
            end;
        end;
    end;
    close(cfg_fi);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{$!+}

result := 10result;

if (result = 0) then
    begin
        for count := 1 to maxstation do
            begin
                {$!-}
                read(cfg_fi,netw_rec[count]);
                {$!+}
                result := 10result;
                if result <> 0 then
                    begin
                        gotoxy(1,1);
                        Setattr(blinkhigh);
                        write(' file read error
press Esc to exit or Enter to continue');
                        repeat
                            Readfunkey(key);
                        until
                            (key = Esc_key) or (key
= Return_key);

                        if (key = Return_key) then
                            goto again;

                        Setattr(lowdisplay);

                    end;
                end;
            end;
        end;
    end;
    close(cfg_fi);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end

    else

        begin

            gotoxy(1,1);

            Setattr(blinkhigh);

            write(' file not found press Esc to exit or
Enter to continue');

            repeat
                Readfunckey(key);
            until
                (key = Esc_key) or (key = Return_key);
            end;
            if (key = Return_key) then
                goto again;
            Setattr(lowdisplay);
            enbedit := true;
            end;
            Windowclose;

        end;

    procedure Doprnt;

    const pkey : array [1..6] of string[20] =
        ('Up,Dn','-Select station ','Enter','-Confirm
select ',
        'Esc','-Back to main menu');

    var pcount,lcount : integer;
        exit,break : boolean;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
```

```
    exit := false;
```

```
    break := false;
```

```
    clrscr;
```

```
    gotoxy(31,1);
```

```
    write('PRINT CONFIGURATION');
```

```
    gotoxy(5,25);
```

```
    for lcount := 1 to 6 do
```

```
        begin
```

```
            if lcount in odd then
```

```
                Setattr(highdisplay)
```

```
            else
```

```
                Setattr(lowdisplay);
```

```
                write(pkey[lcount]);
```

```
            end;
```

```
    Windowopen(30,8,50,15);
```

```
    Setattr(lowdisplay);
```

```
    for pcount := 1 to maxstation do
```

```
        with netw_rec[pcount] do
```

```
            begin
```

```
                gotoxy(3,pcount);
```

```
                write(stat_lab,blank(15-length(stat_lab)));
```

```
            end;
```

```
        pcount := 1;
```

```
    repeat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gotoxy(3,pcount);
```



```

write(netw_rec[pcount].stat_lab);

lcount := pcount;

Readfunckey(key);

case key of

    Up_key : if pcount = 1 then

                pcount := maxstation

            else

                pcount := pcount-1;

            Dn_key : if pcount = maxstation then

                pcount := 1

            else

                pcount := pcount+1;

            Esc_key : exit := true;

            Return_key : begin

                                end;

            end;

end;

Setattr(lowdisplay);

gotoxy(2,lcount);

write(netw_rec[lcount].stat_lab);

until

    exit;

Windowclose;

end;

```

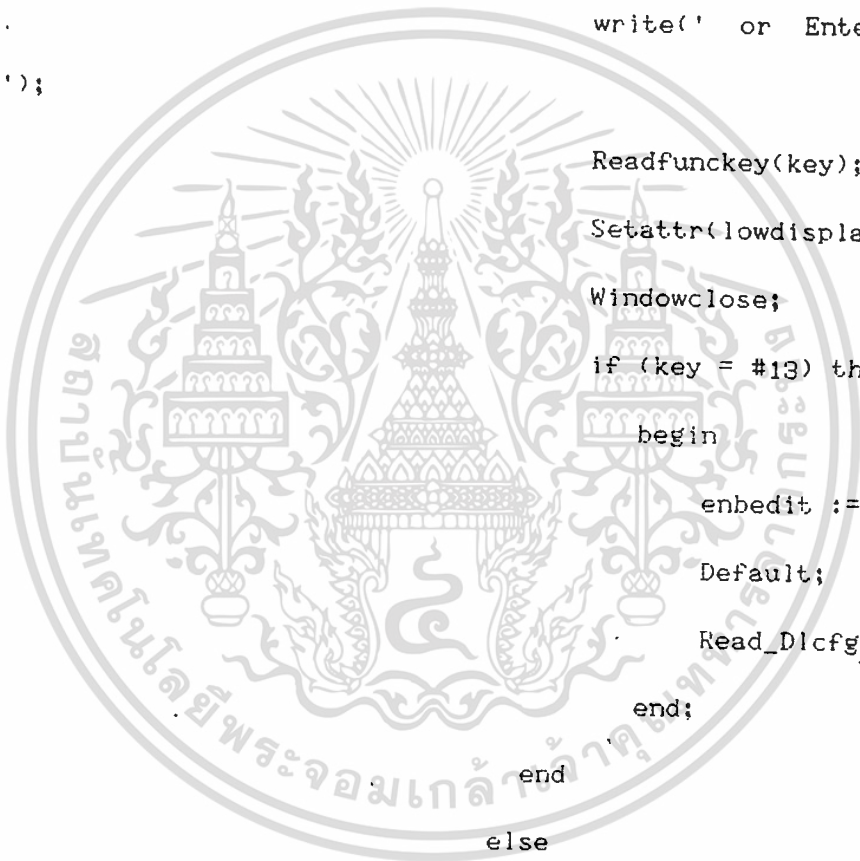
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Setattr(blinkhigh);

gotoxy(1,1);

write('      Please      load
configuration before edit press Esc.');
```



```

gotoxy(1,2);

write(' or Enter for new
file ');

Readfunckey(key);

Setattr(lowdisplay);

Windowclose;

if (key = #13) then
begin
enbedit := true;

Default;

Read_Dlcfg;

end;
else

Read_Dlcfg;

end;

F3_key : begin

if not(enbedit) then

begin

Windowopen(10,5,70,8);

Windowbox(10,5,70,8);

Setattr(blinkhigh);

gotoxy(1,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(netw_rec[pcount].stat_lab);

lcount := pcount;

Readfunckey(key);

case key of

    Up_key : if pcount = 1 then

                pcount := maxstation

            else

                pcount := pcount-1;

            Dn_key : if pcount = maxstation then

                pcount := 1

            else

                pcount := pcount+1;

            Esc_key : exit := true;

            Return_key : begin

                end;

            end;

end;

Setattr(lowdisplay);

gotoxy(2,lcount);

write(netw_rec[lcount].stat_lab);

until

    exit;

Windowclose;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

begin

```

directvideo := true;

quit := false;

enbedit := false;

path := '';

oncursor;

offcursor;

repeat
  clrscr;
  Setattr(highdisplay);
  gotoxy(1,1);
  write('----- EDIT CONFIGURATION -----');
  gotoxy(coor[1],coor[2]);
  for count := 1 to 14 do
  begin
    if count in odd then
      Setattr(highdisplay)
    else
      Setattr(lowdisplay);
    write(main_key[count]);
  end;

  Setattr(lowdisplay);

  Readfunckey(key);

  case key of

    F1_key : begin
      if not(enbedit) then
        begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Windowopen(10,5,70,8);

Windowbox(10,5,70,8);

Setattr(blinkhigh);

gotoxy(1,1);

write(' Please load
configuration before edit press Esc');

gotoxy(1,2);

write(' or Enter for new
file ');

Readfunckey(key);
Setattr(lowdisplay);
Windowclose;
if (key = #13) then
begin
enbedit := true;
Default;
Read_Alcfg;
end;

end

else

Read_Alcfg;

end;

F2_key : begin

if not(enbedit) then

begin
Windowopen(10,5,70,8);
Windowbox(10,5,70,8);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากนำไปใช้

```

write(' Please load
configuration before edit press `Esc`');

gotoxy(1,2);

write(' or Enter for new
file ');

Readfunckey(key);
Setattr(lowdisplay);
Windowclose;

if (key = #13) then
begin
enbedit := true;
Default;
Read_D0cfg;
end;
else
Read_D0cfg;
end;
F4_key : begin
if not(enbedit) then
begin
Windowopen(10,5,70,8);
Windowbox(10,5,70,8);
Setattr(blinkhigh);
gotoxy(1,1);
write(' Please load
configuration before edit press Esc');

```

```
newnode := getnode;
```

```
newnode^.name := name;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านอื่น  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

newnode^.attr := attr;

temp := atnode^.right;

temp^.left := newnode;

newnode^.right := temp;

newnode^.left := atnode;

atnode^.right := newnode;

end;

procedure Insafter(name : dirname; attr : byte);
var
    last : dirptr;
begin
    last := headnode^.left;
    Insrightright(last,name,attr);
end;

procedure Searchdir(name : dirname; var post : byte; var pt : dirptr);
begin
    post := 0;
    pt := headnode;
    repeat
        post := post+1;
        pt := pt^.right;
    until
        (name = pt^.name) or (pt = headnode);
    if (name = pt^.name) then
        post := post+1;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ (name = pt^.name) or (pt = headnode); รังที่มีการนำไปใช้

```

write(' Please load
configuration before edit press 'Esc');

gotoxy(1,2);

write(' or Enter for new
file ');

```

```

Readfunckey(key);
Setattr(lowdisplay);
Windowclose;
if (key = #13) then
begin
enbedit := true;
Default;
Read_DCcfg;
end;
end
else
Read_DCcfg;
end;

```

```

F4_key : begin

```

```

if not(enbedit) then

```

```

begin

```

```

Windowopen(10,5,70,8);

```

```

Windowbox(10,5,70,8);

```

```

Setattr(blinkhigh);

```

```

gotoxy(1,1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
 configuration before edit press 'Esc');

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
getoxy(1,2);  
write(' or Enter for new  
file ');
```

```
Readfunckey(key);  
Setatr(lowdisplay);  
Windowlose;  
if (key = #13) then  
begin  
embedt := true;  
Default;  
Read_STcf;g;  
end;  
else  
Read_STcf;g;  
end;  
F5_key : Saveconf;g;  
F6_key : Readconf;g;  
F7_key : begin
```

```
if not(embedt) then  
begin  
Windowopen(10,5,70,7);  
Windowbox(10,5,70,7);  
Setatr(blinkh;g);  
getoxy(1,1);  
write(' Please load
```

configuration file before print press ESC ');

```

ch := ' ';

repeat
    ch := readkey;
until
    (ch = #27);

Setattr(lowdisplay);
Windowclose;

end
else
Doprint;
gotoxy(coor[1],coor[2]);
for count := 1 to 16 do
begin
if count in odd then
Setattr(highdisplay)
else
Setattr(lowdisplay);
write(main_key[count]);

end;

Setattr(lowdisplay);

end;

Esc_key : quit := true;

end;

until quit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write by ronachai hachai

program view;

uses myscreen,mywin,keyboard,hdlcu3,dos,crt;

{$M 65520,0,655360}

const xdim = 5;

      lineup = #13;

      ftag : array [1..maxstation] of string[4] =
              ('.RP1','.RP2','.RP3','.RP4','.RP5','.RP6');
      headline = 'DEVICE MEASURE AVERAGE DIFFERENT UNIT';
      dash = '-----';
-----';
      con1_key : array [1..8] of string[20] =
              (#25,' -View next ',#24,' -View previous ',
              #26,' -Toggle page ','ESC',' -Back to main
menu');

      con2_key : array [1..10] of string[20] =
              (#25,' -View next ',#24,' -View previous ',
              #26,' -Toggle page ','PgDn','-Next block ',
              'ESC',' -Back to main menu');

      odd : dig = [1,3,5,7,9,11];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dptr = ^datrec;

datrec = record

    drec : analog_rec;

    prev : dptr;

    next : dptr;

end;

dirptr = ^dirlist;
dirname = string[12];
dirlist = record
    name : dirname;
    attr : byte;
    left : dirptr;
    right : dirptr;
end;
st12 = string[12];
text_rec = array [1..25] of string[80];

var

fi : file of analog_log;

temp : analog_log;

firstd,lastd,curdptr : dptr;

newitem,firdat,lastdat : dptr;

maxdat,mindat : array [1..maxaichan] of dptr;

st : stmax;

fi_name : stmax;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
done,quit : boolean;

page : integer;

dsam,maxdsam : integer;

count,dircount : integer;

dirinfo : searchrec;

headnode : dirptr;

pointer : dirptr;

post : byte;

fi_len : integer;

p,n : string;

timrec : array [1..6] of word;

stat : integer;

temprec : analog_rec;

path : string[20];

ready : boolean;

cfgrdy : boolean;

pre_ai,cur_ai : array [1..maxaichan] of real;

firstentry : boolean;

tag : string[1];

tagx : char;

code : integer;

ferr : boolean;

rep_rec : text_rec;

notend : boolean;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function Getnode : dirptr;

var p : dirptr;

begin

    new(p);

    getnode := p;

end;

procedure Freedir(var p : dirptr);

begin

    dispose(headnode);

end;

procedure Initheader;

begin

    headnode := Getnode;

    headnode^.left := headnode;

    headnode^.right := headnode;

end;

procedure Insright(atnode : dirptr; name : dirname; attr : byte);

var

    newnode,temp : dirptr;

begin

    newnode := getnode;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

newnode^.attr := attr;

temp := atnode^.right;

temp^.left := newnode;

newnode^.right := temp;

newnode^.left := atnode;

atnode^.right := newnode;

end;

procedure Insafter(name : dirname; attr : byte);
var
    last : dirptr;
begin
    last := headnode^.left;
    Insrightright(last, name, attr);
end;

procedure Searchdir(name : dirname; var post : byte; var pt : dirptr);
begin
    post := 0;
    pt := headnode;
    repeat
        post := post+1;
        pt := pt^.right;
    until
        (name = pt^.name) or (pt = headnode);
    if (name = pt^.name) then
        post := post+1;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ (name = pt^.name) or (pt = headnode); รังที่มีการนำไปใช้

```

        if pt = headnode then

            post := 0;

        end;

procedure Puttolist;

begin

    with dirinfo do

        begin

            if ((attr and sysfile) = 0) then

                begin

                    Insafter(name,attr);

                    dircount := dircount+1;

                end;

            end;

        end;

    end;

procedure Listdir(mask : string);

begin

    Initheader;

    dircount := 0;

    findfirst(mask,anyfile,dirinfo);

    while doserror = 0 do

        begin

            if (doserror = 0) then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาลงในเอกสารนี้;อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        findnext(dirinfo);

    end;

end;

function Getpath : string;
var
    path : string[20];
begin
    getdir(0,path);
    if path[length(path)] = '\' then
        Getpath := path;
    else
        Getpath := path+'\'
    end;
end;

procedure Findname(var p : integer);
var i : byte;
begin
    pointer := headnode;

    for i := 1 to p do
        pointer := pointer^.right;
    end;
end;

procedure Name(x,y : integer);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
**begin**  
 ไม่ว่าจะผิดๆ อย่างไรก็ดี ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(x,y);

write(' ':14);

gotoxy(x,y);

write(' ',pointer^.name,' ':11-length(pointer^.name));

end;

```

```

function Selname(var max : integer) : st12;

```

```

var offset,maxoffset : integer;

```

```

    curname,lastname,maxname : integer;

```

```

    x,y,z : integer;

```

```

    start,fin : integer;

```

```

    quit,select : boolean;

```

```

begin

```

```

    oncursor;

```

```

    offcursor;

```

```

    quit := false;

```

```

    select := false;

```

```

    curname := 1;

```

```

    offset := 0;

```

```

    if (max > 16) then

```

```

        begin

```

```

            maxoffset := max-16;

```

```

            maxname := 16;

```

```

        end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin

    maxoffset := 0;

    maxname := max;

end;

z := 1;

repeat

    y := 1;

    start := offset+1;

    fin := offset+maxname;

    Setattr(lowdisplay);

    for x := start to fin do
        begin
            Findname(x);

            Name(2,y);

            y := y+1;

        end;

    repeat

        Setattr(reversehigh);

        Findname(curname);

        Name(2,z);

        lastname := curname;

        x := z;

        Readfunckey(key);

        if funckey then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาใดๆอย่างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Up_key : if (curname > 1) then
    begin
        curname := curname-
1;
        if (z > 1) then
            z := z-1
        else
            z := 1;
        end
    else
        begin
            curname := 1;
            z := 1;
        end;
Dn_key : if (curname < max) then
    begin
        curname :=
curname+1;
        if (z < maxname)
            z := z+1
        else
            z := maxname;
    end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    curname := max;
    z := maxname;
end;

end

else
    case key of
        Esc_key : quit := true;
        Return_key : select := true;
    end;
    Setattr(lowdisplay);
    Findname(lastname);
    Name(2,x);
until
    (curname > fin) or (curname < start) or quit or
select;

    if (curname > fin) and (offset < maxoffset) then
        offset := offset+1;

    if (curname < start) and (offset > 0) then
        offset := offset-1;

until
    quit or select;

if select then
    begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
**Findname(curname);**  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Selname := pointer^.name;

    end

else

    Selname := '';

    key := ' ';

end;

```

```

procedure Readdir(mask : string;
    var path,name : string);

```

type

```

string80 = string[80];

```

var

```

    post : byte;

```

```

    p : byte;

```

```

    xindx,yindx : byte;

```

```

    n : dirname;

```

```

    curdir : string80;

```

```

    startdir : string;

```

```

    pt : dirptr;

```

```

    finish : boolean;

```

begin

```

    offcursor;

```

```

    Getdir(0,startdir);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    curdir := '';

```

```

finish := false;

Listdir(mask);

if dircount = 0 then

    begin

        Windowopen(10,5,70,7);

        Windowbox(10,5,70,7);

        gotoxy(1,1);

        Setattr(blinkhigh);

        write(' Logging file not found ! press Enter');

        readln;

        Setattr(lowdisplay);

        Windowclose;

    end

else

    begin

        Windowopen(03,05,20,22);

        Windowbox(03,05,20,22);

        fi_name := Selname(dircount);

        Freedir(headnode);

        Windowclose;

    end;

end;

end;

```

**procedure Convtime;**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

week : array [1..7] of string =
    ('SUN', 'MON', 'TUE', 'WED', 'THR', 'FRI', 'SAT');
month : array [1..12] of string =
    ('JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG',
     'SEP', 'OCT', 'NOV', 'DEC');

var intext : string[2];
    ytext : string[4];

begin
    date := temp.log_time[1];
    mon := temp.log_time[2];
    year := temp.log_time[3];
    hour := temp.log_time[4];
    min := temp.log_time[5];
    sec := temp.log_time[6];
    str(date, intext);
    if date < 10 then
        begin
            timestamp := '0'+intext+'-'+month[mon]+'-';
            str(year, ytext);
            timestamp := timestamp+ytext;
        end
    else
        begin
            timestamp := intext+'-'+month[mon]+'-';
            str(year, ytext);
            timestamp := timestamp+ytext;
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น; ระเบียบด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ข้อมูลนี้ออกไปยังผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        timestamp := timestamp+ytext;

    end;

    timestamp := timestamp+' : ';

    str(hour,intext);

    if (hour < 10) then

        begin

            timestamp := timestamp+'0'+intext+':';

        end

    else

        begin

            timestamp := timestamp+intext+':';

        end;

    str(min,intext);

    if (min < 10) then

        begin

            timestamp := timestamp+'0'+intext+':';

        end

    else

        begin

            timestamp := timestamp+intext+':';

        end;

    str(sec,intext);

    if (sec < 10) then

        begin

            timestamp := timestamp+'0'+intext;


```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ `new(newitem)`; เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        newitem^.drec := temprec;

    until

        keypressed;

    end;

    Windowclose;

```

```
end;
```

```
procedure Ahead;
```

```
const ai_head : array [1..5] of stmax =
```

```
    ('DEVICE', 'MEASURE', 'AVERAGE', 'DIFFERENT', 'UNIT');
```

```
    cord : array [1..5] of integer =
```

```
    (1,21,37,53,66);
```

```
var count : integer;
```

```
begin
```

```
    window(1,2,80,25);
```

```
    clrscr;
```

```
    window(1,1,80,25);
```

```
    Setattr(lowdisplay);
```

```
    gotoxy(1,3);
```

```
    write('STATION NAME : ');
```

```
    gotoxy(1,4);
```

```
    write('TIME STAMP : ');
```

```
    gotoxy(60,3);
```

```
    write('TOTAL SAMPLE : ');
```

```
    gotoxy(60,4);
```

```
    write('SAMPLE NUMBER : ');
```

```
    for count := 1 to 5 do
```

```
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ `gotoxy(cord[count],6);` ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและข้อมูลอ้างอิงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
`write(ai_head[count]);`

```

        timestamp := timestamp+text;

    end;

    timestamp := timestamp+' ';

    str(hour,intext);

    if (hour < 10) then

        begin

            timestamp := timestamp+'0'+intext+'';

        end

    else

        begin

            timestamp := timestamp+intext+'';

        end;

        str(min,intext);

        if (min < 10) then

            begin

                timestamp := timestamp+'0'+intext+'';

            end

        else

            begin

                timestamp := timestamp+intext+'';

            end;

        str(sec,intext);

        if (sec < 10) then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        timestamp := timestamp+'0'+intext;

```

```

        end

    else

        begin

            timestamp := timestamp+intext;

        end;

    end;

end;

procedure Readconfig;
var stat,count : integer;
begin
    Windowopen(10,5,70,7);
    Windowbox(10,5,70,7);
    gotoxy(1,1);
    assign(cfg_fi,path+'\'+cfg_name);
    {$I-}
    reset(cfg_fi);
    {$I+}

    done := IOresult <> 0;

    if not done then

        begin

            write(' Please wait loading Configuration file');

            for stat := 1 to maxstation do

                read(cfg_fi,netw_rec[stat]);

                close(cfg_fi);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรรณการเพื่อกรรณาการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัด ready:=true; ้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end

else

begin

write(' Configuration file not found ! press any
key');

ready := false;

repeat

until keypressed;

end;

Windowclose;

end;

procedure Caldata(var arec : analog_log);

var count : integer;

chan : integer;

mul,offset,aux : real;

begin

Convtime;

temprec.stamp := timestamp;

for count := 0 to maxaichan-1 do

if (netw_rec[stat].ai_cfg[count+1].ai_enb) then

begin

mul := netw_rec[stat].ai_cfg[count+1].multip;

offset := netw_rec[stat].ai_cfg[count+1].offst;

with temprec.analog_val[count+1] do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในองค์กรเท่านั้น ไม่ควรเผยแพร่หรือแจกจ่ายโดยไม่ได้รับอนุญาต  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงที่มาของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin

    name := netw_rec[stat].ai_cfg[count+1]

.ai_lab;

    uname      :=      netw_rec[stat].ai_cfg
[count+1].scal;

    aux      :=      (arec.ai_val[count+1]*mul)
+offset;

    cai_val := aux;
    cur_ai[count+1] := aux;
    if firstentry then
begin
        avr_val := aux;
        dif_val := 0;
    end
    else
begin
        avr_val := (((cur_ai[count+1]
+pre_ai[count+1]) / 2)*mul)+offset;

        dif_val := ((cur_ai[count+1] -
pre_ai[count+1])*mul)+offset;

    end;

    pre_ai[count+1] := cur_ai[count+1];

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังเป็นที่ต้องแจ้งแก่ผู้ดูแล และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
```

```
procedure Readfi(var fi_name : stmax);
```

```
label err;
```

```
var
```

```
    mstart,mstop : longint;
```

```
begin
```

```
    ferr := true;
```

```
    stat := length(fi_name);
```

```
    repeat
```

```
        tag := copy(fi_name,stat,1);
```

```
        stat := stat-1;
```

```
    until
```

```
        (tag = '1') or (tag = '2') or
```

```
        (tag = '3') or (tag = '4') or
```

```
        (tag = '5') or (tag = '6') or
```

```
        (stat = 1);
```

```
    val(tag,stat,code);
```

```
    while (firstd <> nil) do
```

```
        begin
```

```
            newitem := firstd^.next;
```

```
            dispose(firstd);
```

```
            firstd := newitem;
```

```
        end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้ง **maxdsam := 0;** มีข้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

assign(fi,path+'\'+fi_name);

{$I-}

reset(fi);

{$I+}

done := IOresult <>.0;

Windowopen(10,5,70,7);

Windowbox(10,5,70,7);

Setattr(lowdisplay);

if not(done) then
begin
    firstentry := true;
    repeat
        gotoxy(1,1);
        write(' Please wait loading ',fi_name,' for
sample : ',maxdsam);
        seek(fi,maxdsam);
    {$I-}

    read(fi,temp);

    {$I+}

    if (IOresult <> 0) then
        goto err;

    Caldata(temp);

    mstart := memavail;

    new(newitem);
    newitem^.drec := temprec;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if firstd = nil then

    firstd := newitem

else

    lastd^.next := newitem;

    mstop := memavail;

    newitem^.prev := lastd;

    lastd := newitem;

    lastd^.next := nil;

    maxdsam := maxdsam+1;

until
eof(fi) or (memavail < (mstart-mstop));
close(fi);
ferr := false;
end
else

err: begin

    gotoxy(1,1);

    write(' Reading ',fi_name,' error ! press any
key');

    ferr := true;

    repeat

    until

        keypressed;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังขอสงวนสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

end;

procedure Showsample(var rec : analog_rec; pg : integer);
const cord : array [1..5] of integer =
    (1,21,37,53,66);
var count : integer;
    st : string[maxmsg];
begin
    with rec do
        begin
            gotoxy(16,4);
            write(stamp);
            gotoxy(76,4);
            write(dsam,' ');
            if pg = 1 then
                for count := 1 to 16 do
                    with analog_val[count] do
                        begin
                            gotoxy(cord[1],count+7);

                            write(name,blank(15-length(name)));

                            gotoxy(cord[2]-3,count+7);

                            write(realtostr(cai_val),' ');

                            gotoxy(cord[3]-3,count+7);

                            write(realtostr(avr_val),' ');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่สามารถให้ไปใช้ในประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        gotoxy(cord[4]-3,count+7);

        write(realtostr(dif_val),' ');

        gotoxy(cord[5],count+7);

        write(uname,blank(15-length(uname)));

    end

else

    for count := 17 to maxaichan do

        with analog_val[count] do

            begin

                gotoxy(cord[1],count-9);

                write(name,blank(15-length(name)));

                gotoxy(cord[2]-3,count-9);

                write(realtostr(cai_val),' ');

                gotoxy(cord[3]-3,count-9);

                write(realtostr(avr_val),' ');

                gotoxy(cord[4]-3,count-9);

                write(realtostr(dif_val),' ');

                gotoxy(cord[5],count-9);

                write(uname,blank(15-length(uname)));

            end;

        end;

    end;

end;

procedure Showfirst;

begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if count in odd then

```

```

            Setattr(highdisplay)

```

```

        else

```

```

            Setattr(lowdisplay);

```

```

end;

Up_key : begin

    if dsam > 1 then

        dsam := dsam-1;

        Upsample;

    end;

Dn_key : begin

    if dsam < maxdsam then

        dsam := dsam+1;

        Dnsample;

    end;

    Esc_key : quit := true;

end;

if not(quit) then

    Showsample(curdptr^.drec,page);

until

    quit;

key := ' ';

end;

```

```

procedure Wrtpage(var tfl : text;

                 tpg : text_rec);

```

```

var linecount : integer;

```

```

begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        writeln(tfi,tpg[linecount]);

end;

procedure Caltext(var rec : analog_rec;
                 page : integer);

var chan : integer;

    tempst : string[15];

begin

    with rec do
        begin
            rep_rec[1] := '';
            str(dsam,tempst);
            rep_rec[2] := blank(5)+'SAMPLE NUMBER : '+tempst;
            rep_rec[3] := blank(5)+' LOGGING TIME : '+stamp+'';
            rep_rec[6] := dash+'';
            rep_rec[5] := headline+'';
            rep_rec[7] := '';

            case page of

                1 : begin

                    for chan := 1 to 16 do

                        with analog_val[chan] do

                            begin

                                rep_rec[chan+7] := name;

                                tempst := Realtostr(cai_val);
                                rep_rec[chan+7] := rep_rec[chan+7]+blank(18-

```

```

length(rep_rec[chan+7]));

        rep_rec[chan+7] := rep_rec[chan+7]+tempst;

        tempst := Realtostr(avr_val);

        rep_rec[chan+7] := rep_rec[chan+7]+blank(34-
length(rep_rec[chan+7]));

        rep_rec[chan+7] := rep_rec[chan+7]+tempst;

        tempst := Realtostr(dif_val);

        rep_rec[chan+7] := rep_rec[chan+7]+blank(50-
length(rep_rec[chan+7]));

        rep_rec[chan+7] := rep_rec[chan+7]+tempst;

        rep_rec[chan+7] := rep_rec[chan+7]+blank(66-
length(rep_rec[chan+7]));

        rep_rec[chan+7] := rep_rec[chan+7]+uname'';

    end;

end;

2 : begin

    for chan := 17 to maxaichan do

        with analog_val[chan] do

            begin

                rep_rec[chan-9] := name;

                tempst := Realtostr(cai_val);

                rep_rec[chan-9] := rep_rec[chan-9]+blank(18-
length(rep_rec[chan-9]));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ rep\_rec[chan-9] := rep\_rec[chan-9]+tempst; ค่า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
tempst := Realtostr(avr\_val);



```

begin
again: Windowopen(10,5,70,7);

      Windowbox(10,5,70,7);

      Setattr(lowdisplay);

      gotoxy(1,1);

      write(' Please enter path to save :');

      textpath := Readstr(15);

      Windowopen(15,6,65,10);

      Windowbox(15,6,65,10);

      Setattr(lowdisplay);

      gotoxy(1,1);

      write('total sample = ',maxdsam);

      gotoxy(1,2);

      write('start sample :');

      Readint(start,maxdsam);

      gotoxy(1,3);

      write(' stop sample :');

      Readint(stop,maxdsam);

      Windowclose;

      assign(rep_fi,textpath+'\''+netw_rec[stat].stat_lab+ftag
[stat]);

      {$I-}

      rewrite(rep_fi);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งการแก้ไขต้องแจ้งผู้ดูแล และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin

    rep_rec[1] := '';

    rep_rec[2] := blank(20)+'REPORT LOGGING DATA'+'';

    rep_rec[3] := '';

    rep_rec[4] := '';

    rep_rec[5] := '';

    rep_rec[6] := blank(5)+'STATION NAME : '+netw_rec
[stat].stat_lab+'';

    rep_rec[7] := '';
    str(netw_rec[stat].log_time*3,dstr);
    rep_rec[8] := blank(5)+'LOGGING TIME : '+dstr+'
SECOND'+'';

    rep_rec[9] := '';
    str(maxdsam,dstr);
    rep_rec[10] := blank(5)+'TATAL SAMPLE : '+dstr+'';
    rep_rec[11] := '';
    rep_rec[12] := blank(5)+' START TIME : '+firstd^.
drec.stamp+'';

    rep_rec[13] := '';

    rep_rec[14] := blank(5)+' STOP TIME : '+lastd^.
drec.stamp+'';

    for line := 15 to 25 do

        rep_rec[line] := '';

        Wrtpage(rep_fi,rep_rec);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dsam := 1;

firdat := firstd;

curdptr := firstd;

curdptr := curdptr^.next;

lastdat := curdptr^.prev;

clrscr;

gotoxy(1,1);

write(' Please wait creating text file for sample
number : ');

repeat
gotoxy(54,1);
write(dsam);
if (dsam >= start) and (dsam <= stop) then
begin
Caltext(curdptr^.drec,i);
Wrtpage(rep_fi,rep_rec);
Caltext(curdptr^.drec,2);
Wrtpage(rep_fi,rep_rec);

end;

Dnsample;

dsam := dsam+1;

until

(dsam = maxdsam+1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end

```

else
    begin
        clrscr;
        gotoxy(1,1);
        write(' File creating error ! ESC -quit , Enter -
continue');
        repeat
            Readfunckey(key);
        until
            (key = #27) or (key = #13);
        if (key = #13) then
            begin
                Windowclose;
                goto again;
            end;
        end;
    end;
Windowclose;

end;

procedure Mainmenu;
const main_key : array [1..8] of string[25] =
    ('F1','-Load file ', 'F2','-View data ',
    'F3','-Create report file ',
    'Esc','-Quit');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

firstd := nil;

lastd := nil;

repeat

    clrscr;

    Setattr(highdisplay);

    gotoxy(1,1);

    write('----- VIEW LOGGING
DATA -----');

    gotoxy(5,25);

    for count := 1 to 8 do
    begin
        if (count in odd) then
            Setattr(highdisplay)
        else
            Setattr(lowdisplay);
        write(main_key[count]);
    end;

    Setattr(lowdisplay);

    Readfunckey(key);

    case key of

        F1_key : begin

            if not(cfgrdy) then

                begin

                    Windowopen(10,5,70,7);
                    Windowbox(10,5,70,7);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการแก้ไข

```

gotoxy(1,1);

write(' Please enter

path :');

path := Readstr(20);

clrscr;

Windowclose;

end;

if (path <> '') then
begin
if not(cfgrdy) then
begin
Readconfig;
cfgrdy := true;
end;
if ready then
begin
while (firstd
<> nil) do

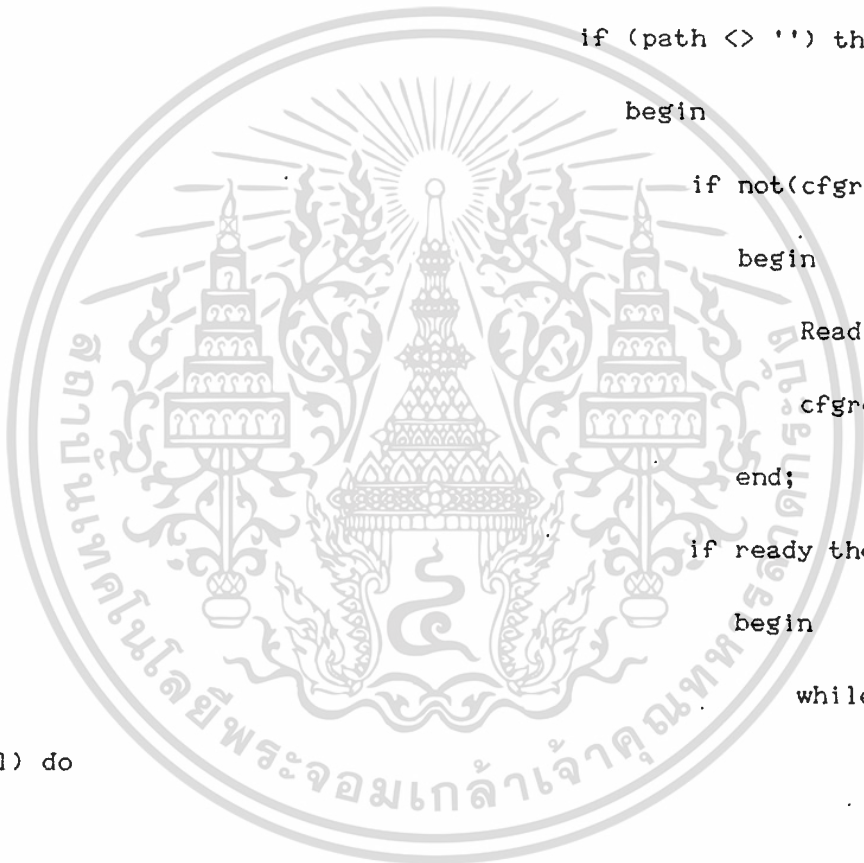
begin

newitem := firstd^.next;

dispose(firstd);

firstd := newitem;

```



```
alr_buf : array [1..18] of string[60];
```

```
alr_ptr : integer;
```

```
chcount : integer;
```

```
mul,offset,thr,aux : real;
```

```
logf_name : string;
```

```
col,line,chan : integer;
```

```
const
```

```
do_key : array [1..10] of string[18] =
```

```
(#27,#26,'-Select stat ',
```

```
#24,#25,'-Select dev ',
```

```
'ENTER','-Accept select ',
```

```
'ESC','-Back to main menu');
```

```
hid : dig = [1,2,4,5,7,9];
```

```
mask : array [1..8] of byte =
```

```
($1,$2,$4,$8,$10,$20,$40,$80);
```

```
hi_alr : string[16] =
```

```
('OVER MAX-VALUE');
```

```
lo_alr : string[16] =
```

```
('UNDER MIN_VALUE');
```

```
cord : array [1..4] of integer =
```

```
(3,21,49,67);
```

```
do_head : array [1..4] of string[10] =
```

```
('DEVICE','STATUS','DEVICE','STATUS');
```

```
di_head : array [1..4] of string[10] =
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อวัตถุประสงค์เฉพาะเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

('DEVICE','STATUS','DEVICE','STATUS');

ai_head : array [1..5] of string[10] =

('DEVICE','MEASURE','AVERAGE','DIFFERENT',

'UNIT');

head_pos : array [1..5] of integer =

(1,21,37,53,66);

ai_key : array [1..8] of string[20] =

(#24,#25,' -Select station ',

#27,#26,' -Select page ',

'Esc',' -Back to main memu');

procedure ChkDI(var stat : integer);

var

cdi,pdi : byte;

pos : integer;

tempdi : byte;

begin

with netw_rec[stat] do

for count := 1 to 4 do

begin

for pos := 1 to 2 do

begin

cdi := cur_di[stat,count];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่ควรออกให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

with di_cfg[count*pos] do
    if (enb_alarm) and (di_enb) then
        begin
            tempdi := (cdi xor pdi) and mask[pos];
            if (tempdi <> 0) then
                begin
                    alr_buf[alr_ptr] := timestamp+'
+stat_lab+
                    blank(15-length(stat_lab))+
                    di_lab+blank(15-length(di_lab))
+ ' '+
                    alarm_msg;
                    if (alr_ptr < 18) then
                        alr_ptr := alr_ptr+1
                    else
                        begin
                            for alr_ptr := 1 to 17 do
                                alr_buf[alr_ptr] :=
alr_buf[alr_ptr+1];
                            alr_ptr := 18;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภายในเท่านั้น ไม่ควรเผยแพร่ไปยังภายนอกโดยไม่ได้รับอนุญาต  
 pre\_di[stat,count] := cur\_di[stat,count];  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

end;

procedure Chkover(var stat : integer);

begin

    with netw_rec[stat] do

        for count := 0 to maxaiChan-1 do

            with ai_cfg[count+1] do

                begin

                    if (ai_enb and alarm) then

                        begin

                            if (ai_buf[stat,count+1].cai_val >= max_val)

                                then

                                    begin

                                        alr_buf[alr_ptr] := timestamp'+stat_lab+

                                            blank(18-length(stat_lab))+

                                                ai_lab+blank(15-length(ai_lab))+

                                                    hi_alr;

                                        alr_ptr := alr_ptr+1;

                                        if (alr_ptr = 19) then

                                            begin

                                                for alr_ptr := 1 to 17 do

                                                    begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
alr_buf[alr_ptr] :=
```

```

alr_buf[alr_ptr+1];

end;

alr_ptr := 18;

end;

end;

if (ai_buf[stat,count+1].cai_val <= min_val)

then

begin

alr_buf[alr_ptr] := timestamp+'

'+stat_lab+

blank(18-length(stat_lab))+

ai_lab+blank(15-length(ai_lab))+

lo_alr;

alr_ptr := alr_ptr+1;

if (alr_ptr = 19) then

begin

for alr_ptr := 1 to 17 do

begin

alr_buf[alr_ptr] :=

alr_buf[alr_ptr+1];

end;

alr_ptr := 18;

end;

end;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

end;

procedure Collectdata(var stat : integer);

begin

    for chcount := 0 to maxaichan-1 do

        begin

            if (netw_rec[stat].ai_cfg[chcount+1].ai_enb) then

                begin

                    mul := netw_rec[stat].ai_cfg[chcount+1].multip;
                    offset := netw_rec[stat].ai_cfg[chcount+1].
offst;
                    thr := netw_rec[stat].ai_cfg[chcount+1].thresh;
                    cur_ai[stat,chcount+1] := ((rxbuffer[(chcount)
*2] and $f0) div 16) +
(rxbuffer[((chcount)
*2)+1] * 16);

                    ai_log.ai_val[chcount+1] := cur_ai[stat,
chcount+1];

                    aux := (cur_ai[stat,chcount+1]*mul)+offset;

                    if (aux >= thr) then

                        ai_buf[stat,chcount+1].cai_val := aux

                    else

                        ai_buf[stat,chcount+1].cai_val := thr;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin

    ai_buf[stat,chcount+1].avr_val := aux;

    ai_buf[stat,chcount+1].dif_val := 0;

end

else

begin

    aux := (((cur_ai[stat,chcount+1]+pre_ai
[stat,chcount+1]) / 2)*mul)+offset;

    ai_buf[stat,chcount+1].avr_val := aux;

    aux := ((cur_ai[stat,chcount+1] -
pre_ai[stat,chcount+1])*mul)+offset;

    ai_buf[stat,chcount+1].dif_val := aux;

end;

pre_ai[stat,chcount+1] := cur_ai[stat,
chcount+1];

end;

end;

cur_di[stat,1] := rxbuffer[64];

cur_di[stat,2] := rxbuffer[65];

cur_di[stat,3] := rxbuffer[66];

cur_di[stat,4] := rxbuffer[67];

if firstentry[stat] then

begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ firstentry[stat] := false; ของเอกสารทุกครั้งที่มีการนำไปใช้

```

pre_di[stat,1] := cur_di[stat,1];

pre_di[stat,2] := cur_di[stat,2];

pre_di[stat,3] := cur_di[stat,3];

pre_di[stat,4] := cur_di[stat,4];

end;

for chcount := 1 to 6 do

with ai_log do

begin

log_time[1] := date;

log_time[2] := mon;

log_time[3] := year;

log_time[4] := hour;

log_time[5] := min;

log_time[6] := sec;

end;

end;

procedure LogAI(var stat : integer);

begin

logf_name := netw_nec[stat].stat_lab+site_ana[stat];

assign(ai_fi,(logdev+logf_name));

{$I-}

reset(ai_fi);

{$I+}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rewrite(ai_fi)

else

seek(ai_fi,filesize(ai_fi));

write(ai_fi,ai_log);

close(ai_fi);

end;

procedure ShowAthead;

begin

clrscr;

gotoxy(1,1);

Setattr(highdisplay);

write('-----ANALOG INPUT MONITOR
-----');

Setattr(lowdisplay);

gotoxy(1,3);

write('STATION NAME :');

for count := 1 to 5 do

begin

gotoxy(head_pos[count],5);

write(ai_head[count]);

end;

gotoxy(5,25);

for count := 1 to 8 do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (count in hid) then
            Setattr(highdisplay)
        else
            Setattr(lowdisplay);
        write(ai_key[count]);
    end;

    Setattr(lowdisplay);
end;

procedure MonitorAI(var stat : integer);
var chcount : integer;
begin
    gotoxy(58,3);write(timestamp);
    st := netw_rec[stat].stat_lab;
    gotoxy(16,3);write(st,Blank(15-length(st)));
    line := 7;
    if (page = 0) then
        with netw_rec[stat] do
            begin
                for chcount := 1 to 16 do
                    if (ai_cfg[chcount].ai_enb) then
                        begin
                            gotoxy(head_pos[1],line);

                                st := netw_rec[stat].ai_cfg[chcount].

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(st,Blank(15-length(st)));

st := Realtrostr(ai_buf[stat,chcount],
cai_val);

gotoxy(head_pos[2]-3,line);

write(st,' ');

if ai_cfg[chcount].aveng then
st := Realtrostr(ai_buf[stat,chcount],
avr_val)
else
st := ' - ';
gotoxy(head_pos[3]-3,line);
write(st,' ');
if ai_cfg[chcount].diffft then
st := Realtrostr(ai_buf[stat,chcount],
dif_val)
else
st := ' - ';
gotoxy(head_pos[4]-3,line);

write(st,' ');

gotoxy(head_pos[5],line);

st := netw_rec[stat].ai_cfg[chcount].scal;

write(st,Blank(15-length(st)));

line := line+1;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 end  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else

with netw_rec[stat] do

begin

for chcount := 17 to maxaichan do

if ai_cfg[chcount].ai_enb then

begin

gotoxy(head_pos[1],line);

st := netw_rec[stat].ai_cfg[chcount].

ai_lab;

write(st,Blank(15-length(st)));

st := Realtostr(ai_buf[stat],chcount).

cai_val);

gotoxy(head_pos[2]-3,line);

write(st,' ');

if ai_cfg[chcount].averg then

st := Realtostr(ai_buf[stat],chcount).

avr_val)

else

st := ' - ';

gotoxy(head_pos[3]-3,line);

write(st,' ');

if ai_cfg[chcount].difft then

st := Realtostr(ai_buf[stat],chcount).

dif_val)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        st := ' - ';

        gotoxy(head_pos[4]-3,line);
        write(st,' ');

        gotoxy(head_pos[5],line);

        st := netw_rec[stat].ai_cfg[chcount].scal;;

        write(st,Blank(15-length(st)));

        line := line+1;
    end;
end;

procedure ShowDIhead;
begin
    Setattr(highdisplay);
    gotoxy(1,1);
    write('----- STATUS INPUT MONITOR
-----');

    Setattr(lowdisplay);

    gotoxy(3,3);

    write('STATION NAME : ');

    for count := 1 to 4 do
        begin
            gotoxy(cord[count],5);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ end; ปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(58,25);

Setattr(highdisplay);

write('Esc');

Setattr(lowdisplay);

write('-Exit to main menu');

end;

procedure MonitorDI(var stat : integer);
begin
    col := 1;
    line := 7;
    gotoxy(58,3);write(timestamp);
    st := netw_rec[stat].stat_lab;
    gotoxy(19,3);write(st,Blank(15-length(st)));
    for count := 1 to 4 do
        for chan := 1 to 8 do
            if netw_rec[stat].di_cfg[((count-1)*8)+(chan)].
di_enb then

                begin

                    gotoxy(cord[col],line);

                    st := netw_rec[stat].di_cfg[((count-1)*8)+(chan)]
                    .di_lab;

                    write(st,Blank(14-length(st)), ' :');

                    gotoxy(cord[col+1],line);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปตีพิมพ์หรือจำหน่าย  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

then

    begin

        st := netw_rec[stat].di_cfg[((count-1)*8)
+(chan)].on_msg;

        write(st,Blank(14-length(st)));

    end

else

    begin

        st := netw_rec[stat].di_cfg[((count-1)*8)
+(chan)].off_msg;

        write(st,Blank(14-length(st)));

    end;

    line := line+1;

    if (line > 22) then

        begin

            line := 7;

            col := 3;

        end;

    end;

end;

end;

procedure ShowDOhead;

begin

```

```

    Setattr(highdisplay);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุที่เบี่ยงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write('----- CONTROL OUTPUT
MONITOR -----');

Setattr(lowdisplay);

gotoxy(3,3);

write('STATION NAME : ');

for count := 1 to 4 do
begin
gotoxy(cord[count],5);
write(do_head[count]);
end;
gotoxy(1,25);
for count := 1 to 10 do
begin
if (count in hid) then
Setattr(highdisplay)
else
Setattr(lowdisplay);
write(do_key[count]);
end;

Setattr(lowdisplay);

end;

procedure MonitorD0(var stat : integer);
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line := 7;

gotoxy(58,3);write(timestamp);

st := netw_rec[stat].stat_lab;

gotoxy(19,3);write(st,Blank(15-length(st)));

for count := 1 to 4 do

    for chan := 1 to 8 do

        if netw_rec[stat].do_cfg[((count-1)*8)+chan];

do_endb then

        begin

            gotoxy(cord[col],line);

            st := netw_rec[stat].do_cfg[((count-1)*8)

+(chan)].do_lab;

            if seldol(((count-1)*8)+(chan)] then

                Setattr(reversehigh);

                write(st,Blank(15-length(st)));

                Setattr(lowdisplay);

                write(' ');

                gotoxy(cord[col+1],line);

                if ((cur_dol[stat,count] and mask[chan])

<> 0) then

                    begin

                        st := netw_rec[stat].do_cfg

[(((count-1)*8)+(chan)].on_msg;

                        write(st,Blank(15-length(st)));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else

begin

st := netw_rec[stat].do_cfg

[((count-1)*8)+(chan)].off_msg;

write(st,Blank(15-length(st)));

end;

line := line+1;

if (line > 22) then

begin

line := 7;

col := 3;

end;

end;

end;

procedure Readconfig;

begin

fin := false;

Windowopen(19,5,61,7);

Windowbox(19,5,61,7);

gotoxy(1,1);

write(' Please wait loading configuration file');

assign(cfg_fi,cfg_name);

{$!-}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 reset(cfg\_fi);  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{$I+}

if (IOresult = 0) then

    begin

        for station := 1 to maxstation do

            read(cfg_fi,netw_rec[station]);

            close(cfg_fi);

            delay(500);

        end

    else

        begin

            cireol;

            write(' Configuration file not found!');

            .delay(800);

            fin := true;

        end;

    Windowclose;

end;

```

{\$F+}

procedure Chkkey;

{\$F-}

interrupt;

begin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 inline(\$fa);  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Stamptime;

if keypressed then
  begin
    key := readkey;
    kflag := true;
  end;

if logenb[cursite] and enblog then
  begin
    logenb[cursite] := false;
    logrem := diskfree(lognum)/(disksize(lognum));
    if ((logrem*100) >= 1) then
      begin
        logai(cursite);
        tranflag := false;
        if (cursite = maxstation) then
          begin
            if (samcnt < 99999) then
              samcnt := samcnt+1
            else
              samcnt := 0;
          end;
        end;
      end
    else
      tranflag := true;
    end;
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ,  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        eoi;

end;

{$F+}

procedure Netwctrl;

{$F-}

interrupt;

begin

    inline($fa);
    rescount;
    if (curtime >= maxtime) then
        curtime := 1
    else
        curtime := curtime+1;
    if (curtime = 1) then
        begin
            if (comflag[cursite] <> txser) then
                begin
                    txbuffer[0] := cur_do[cursite,1];
                    txbuffer[1] := cur_do[cursite,2];
                    txbuffer[2] := cur_do[cursite,3];
                    txbuffer[3] := cur_do[cursite,4];
                    controlf := reqpoll;
                    txlen := polllen;
                    Fram_TX(cursite);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
```

```

        if (cursite >= maxstation) then
            begin
                presite := maxstation;
                cursite := 1;
            end
        else
            begin
                presite := cursite;
                cursite := cursite+1;
            end;
        end;
    eoi
end;

{$F+}
procedure Datalink;
{$F-}
interrupt;
begin
    inline($fa);
    if TXINT then
        begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    txrest := port[txresult];
    if (txrest = txfrcom) then
        begin
            comflag[cursite] := txrdy;
            rxrest[0] := $f;
        end
    else
        comflag[cursite] := txerror;
    end;
end
else
    begin
        if RXIRA then
            begin
                while RXINT do
                    begin
                        if RXIRA then
                            begin
                                rxrest[rxrptr] := port
[rxresult];
                                rxrptr := rxrptr+1;
                                if (rxrptr > 0) and (rxrest.
[0] <> rx8b) then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

comflag[cursite] :=
rxerror;

end

else
if (rxrest[0] = rx8b) and
(rxrptr = 5) then
begin
comflag[cursite] :
= rxrdy;
end;
end;
end;
Eoi;
end;

procedure Setintr;

begin

getintvec($1C, intr1c);

getintvec(13, intr13);

getintvec(12, intr12);

setintvec($1C, @Chkkey);

setintvec(13, @netwctrl);

setintvec(12, @datalink);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Interp(on);

end;

procedure Mainmenu;

begin

    clrscr;

    Setattr(highdisplay);

    gotoxy(1,1);
    write(' <- REMOTE DATA LOGGER ->');

    Setattr(lowdisplay);

    gotoxy(1,4);
    write(' CURRENT TIME : ');

    gotoxy(1,5);
    write(' TRANSFER DEVICE : ');

    gotoxy(1,6);
    write(' AVIALABLE SPACE : %');

    gotoxy(1,7);

    write(' START LOG TIME : ');

    gotoxy(1,8);

    write(' STOP LOG TIME : ');

    gotoxy(59,5);

    write(' LOGGING DEVICE : ');

    gotoxy(58,6);

    write(' AVIALABLE SPACE : %');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(' LOGGING DATA : ');

gotoxy(59,8);

write(' SAMPLE NUMBER : ');

gotoxy(1,9);

write('-----
-----');

gotoxy(1,11);

write('-----
-----');

gotoxy(1,18);

write('-----
-----');

gotoxy(3,25);
for c := 1 to 14 do
begin
if c in odd then
Setattr(highdisplay)

else

Setattr(lowdisplay);

write(main_key[c]);

end;

Setattr(lowdisplay);

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
**procedure Showsys;**  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

var

```
stn : integer;
```

```
st3 : real;
```

begin

```
Setattr(lowdisplay);
```

```
gotoxy(19,4);
```

```
write(timestamp);
```

```
gotoxy(19,5);
```

```
write(tradev, ' ');
```

```
for stn := 1 to 3 do
```

```
begin
```

```
gotoxy(top_pos[stn],10);
```

```
write(topic[stn]);
```

```
end;
```

```
for stn := 1 to maxstation do
```

```
with netw_rec[stn] do
```

```
begin
```

```
gotoxy(top_pos[1],stn+11);
```

```
write(stat_lab, ' ');
```

```
gotoxy(top_pos[2]+5,stn+11);
```

```
write(log_time*3, ' ');
```

```
gotoxy(65,stn+11);
```

```
if (site_stas[stn] = abnormal) then
```

```
write('ABNORMAL')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    write('POLLING ');

end;

gotoxy(76,5);

write(logdev);

gotoxy(76,6);

st3 := (logrem*100);

if (st3 >= 0) and (st3 <= 9) then
    write(st3:1:0, ' ');
if (st3 >= 10) and (st3 <= 99) then
    write(st3:2:0, ' ');
if (st3 >= 100) and (st3 <= 999) then
    write(st3:3:0);
gotoxy(19,6);
st3 := (trarem*100);

if (st3 >= 0) and (st3 <= 9) then
    write(st3:1:0, ' ');

if (st3 >= 10) and (st3 <= 99) then
    write(st3:2:0, ' ');

if (st3 >= 100) and (st3 <= 999) then
    write(st3:3:0);

gotoxy(76,7);

if enblog then

    write('ON ')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        write('OFF ');

gotoxy(76,8);

write(samcnt);

gotoxy(80,presite+11);

write(#0);

Setattr(highdisplay);

gotoxy(80,cursite+11);

write(#27);

Setattr(lowdisplay);

gotoxy(19,7);

write(timestart,' ');

gotoxy(19,8);

write(timestop,' ');

end;

procedure Ctrlcmd(var stat,point : integer);

var po,ch : integer;

begin

    gotoxy(1,25);

    clreol;

    gotoxy(1,25);

    for ch := 1 to 6 do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และเผยแพร่ไปยังเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Setattr(highdisplay)

    else

        Setattr(lowdisplay);

    write(op_key[ch]);

end;

ch := ((point-1) div 8)+1;
po := ((point-1) mod 8)+1;
repeat
    MonitorDO(stat);
    gotoxy(70,25);
    case key of
        F1_key : begin
            cur_do[stat,ch] := cur_do[stat,
ch] or
                                do_mask[po];
            write('ON DEVICE ');
            end;
        F2_key : begin
            cur_do[stat,ch] := cur_do[stat,
ch] and
                                not(do_mask
[po]);

            write('OFF DEVICE');
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมี end; เปล่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

until
    key = Esc_key;

gotoxy(1,25);

clrscr;

gotoxy(1,25);

for ch := 1 to 10 do
    begin
        if ch in hid then
            Setattr(highdisplay)
        else
            Setattr(lowdisplay);
            write(do_key[ch]);
        end;
    end;

end;

procedure Readlogdev;

label try,f,logerr,traerr;

var stlog : string[15];

    samf : text;

    logfail : integer;

begin

    clrscr;

    Windowopen(10,5,70,8);

    Windowbox(10,5,70,8);

    try: timok := false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

repeat
    Readfunckey(key);

until
    (key = #13) or (key = #27);

clrscr;

if (key = #13) then
    begin
        key := ' ';
        goto try;
    end
else
    begin
        timok := false;
        goto f;
    end
end

else
    begin
        close(samf);
        erase(samf);

    end;

lognum := 0;

repeat
    lognum := lognum+1;
until

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (logdev = logdrv[lognum]) or (lognum > 7);

    if (lognum > 7) then
        goto logerr;

    tranum := 0;

    repeat
        tranum := tranum+1;
    until
        ((copy(tradev,1,2)) = logdrv[tranum]) or (tranum >
7);

    if (tranum > 7) then
        goto traerr;
    assign(samf,logdev+'\SAM.TXT');
    {$I-}
    rewrite(samf);
    {$I+}
    logfail := IOresult;
    if (logfail <> 0) then

logerr: begin

        clrscr;

        gotoxy(1,1);

        write(' LOGGING DEVICE not found ! ENTER to
continue,ESC to quit');

        repeat

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ Readfunckey(key); ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        (key = #13) or (key = #27);

    clrscr;

    if (key = #13) then

        begin

            key := ' ';

            goto try;

        end

    else

        begin

            timok := false;

            goto f;

        end;

    end

else

begin

close(samf);

erase(samf);

timok := true;

end;

f: Windowclose;

end;

procedure Showalarm;

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(timestamp, ' ');

for alr_cnt := 1 to 18 do

    begin

        gotoxy(1,alr_cnt+5);

        write(air_buf[alr_cnt]);

    end;

end;

```

```

procedure Clralarm;

begin

    for alr_cnt := 1 to 18 do

        air_buf[alr_cnt] := ' ';

        alr_ptr := 1;

    end;

```

```

procedure Transfer;

var temp_rec : analog_log;

    in_fi,ot_fi : file of analog_log;

    rec_cnt : integer;

    sta_cnt : integer;

    fin_name : string;

    fot_name : string;

begin

```

```

    enblog := false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีสิทธิ์เปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Retintr;

```

```

gotoxy(76,7);

Setattr(highdisplay);

write('DUMP ');

mkdir(tradev+samp[cursam]);

for sta_cnt := 1 to maxstation do

    with netw_rec[sta_cnt] do

        begin

            fin_name := stat_lab+site_ana[sta_cnt];
            fot_name := stat_lab+site_ana[sta_cnt];
            assign(in_fi,logdev+fin_name);
            assign(ot_fi,tradev+samp[cursam]+'\''+fot_name);
            reset(in_fi);
            rewrite(ot_fi);
            rec_cnt := 0;
            repeat

                seek(in_fi,rec_cnt);
                read(in_fi,temp_rec);

                write(ot_fi,temp_rec);

                rec_cnt := rec_cnt+1;

            until

                eof(in_fi) or (l0result <> 0);

            rewrite(in_fi);

            close(in_fi);

            close(ot_fi);
        end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Setattr(lowdisplay);

trarem := diskfree(tranum)/(disksize(tranum));

if (cursam < 20) then

    cursam := cursam+1

else

    cursam := 1;

    Initnet;

    Setintr;

    enblog := true;

    tranflag := false;

end;

label exit;

begin

    clrscr;

    Default;

    Readconfig;

    Readlogdev;

    if not(timok) then

        goto exit;

    rxrptr := 0;

    rxdptr := 0;

    txdptr := 0;

    cursam := 1;

    samcnt := 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

enblog := false;

tranflag := false;

kflag := false;

logrem := 1;

Clralarm;

presite := 1;

cursite := 1;

timestart := '';
timestop := '';
timestamp := '';
trarem := diskfree(tranum)/(disksize(tranum));
Initnet;
Setintr;
repeat
    mainmenu;
while not(quit) do
    begin
        showsys;

        if tranflag then

            Transfer;

        if kflag then

        case key of

            F1_key : begin

```

```

ShowAhead;

station := 1;

quit := false;

repeat

    MonitorAl(station);

    case key of

        Dn_key : if station
= maxstation then

station := 1

station := station+1;

        Up_key : if station
= 1 then

station := maxstation

        else

station := station-1;

        Lt_key,Rt_key : if

page = 0 then

page :=

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

page :=

0;

Esc\_key : quit :=

true;

end;

key := ' ';

until

quit;

end;

F2\_key : begin

clnscr;

ShowDIhead;

station := 1;

lstat := station;

quit := false;

repeat

MonitorDI(station);

case key of

Dn\_key : if station

= maxstation then

station := 1

else

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

station := station+1;

```

Up_key : if station
= 1 then

station := maxstation

else

station := station-1;

Esc_key : quit :=
true;

end;
key := ' ';
until
quit;
end;
F3_key : begin
clrscr;
ShowD0head;

station := 1;

quit := false;

for cdo := 1 to maxdochan do

seldo[cdo] := false;

cdo := 1;

pdo := 1;

repeat
seldo[pdo] := false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งเมื่อให้นำไปใช้

```

seldo[cdo] := true;

MonitorDO(station);

case key of

    Rt_key : if station

= maxstation then

station := 1

else

station := station+1;

Lt_key : if station

= 1 then

station := maxstation

else

station := station-1;

Dn_key : begin

    pdo :=

cdo;

    if

(cdo = maxdochan) then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
 else

```
cdo := cdo+1;
```

```
end;
```

```
Up_key : begin
```

```
pdo :=
```

```
cdo;
```

```
if
```

```
(cdo = 1) then
```

```
cdo := maxdochan
```

```
else
```

```
cdo := cdo-1;
```

```
end;
```

```
Return_key : begin
```

```
Ctrlcmd(station,cdo);
```

```
end;
```

```
Esc_key : quit :=
```

```
true;
```

```
end;
```

```
key := ' ';
```

```
until
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
```

```

F4_key : begin

    clrscr;

    Setattr(highdisplay);

    gotoxy(1,1);

    write('-----

- ALARM OCCUR MONITOR -----');

    gotoxy(1,25);
    write('F1');
    gotoxy(27,25);
    write('Esc');
    Setattr(lowdisplay);
    gotoxy(1,3);
    write('CURRENT TIME : ');
    gotoxy(1,5);
    write('TIME STATION DEVICE OCCUR');

    gotoxy(4,25);

    write('-clear alarm occur ');

    gotoxy(30,25);

    write('-back to main menu');

    repeat

        Showalarm;

        case key of

            F1_key : begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะโดยทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Clralarm;

```

key :
= ' ';

end;

Esc_key : quit :=
true;

end;

until
quit;
key := ' ';
end;
F5_key : begin
if not(enblog) then
begin
enblog := true;
timestamp := copy
(timestamp, (length(timestamp)-7), 8);
timestop := ' ';
for cdo := 1 to maxstation
do
begin
logtim[cdo] :=
netw_rec[cdo].log_time;
logenb[cdo] :=
false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
logcount[cdo] := 0;
```

```

end;

end;

end;

F6_key : begin

    if (enblog) then

        begin

            enblog := false;

            timestop := copy(timestamp,

(length(timestamp)-7),8);

            end;

            end;

            F10_key : begin

                fin := true;

                quit := true;

                end;

            end;

            key := ' ';

        end;

    quit := false;

until

    fin;

Retintr;

exit: res_DEV;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write by ronachai hachai

unit hdlcu3;

interface

uses dos,crt,myscreen,mywin,keyboard;

const

    counter = $380;

    count0 = counter;

    count1 = counter+1;

    count2 = counter+2;

    cntctrl = counter+3;

    protc = $388;

    txdata = $390;

    rxdata = $38c;

    commd = protc;

    status = protc;

    param = protc+1;

    result = protc+1;

    resetp = protc+2;

    txresult = protc+2;

    rxresult = protc+3;

    tmr = protc+2;

    enb_func = $384;

```

```

req_reg = $0009;

mask_reg = $000a;

mode_reg = $000b;

clr_bptr = $000c;

clrm_reg = $000e;

wrtm_reg = $000f;

page1_reg = $0083;

page3_reg = $0082;

base1_add = $0002;

base1_cnt = $0003;

base3_add = $0006;

base3_cnt = $0007;

{ 8237 DMA control byte }

wrt_mode1 = $45;

red_mode1 = $49;

enb_dma1 = $1;

dis_dma1 = $5;

wrt_mode3 = $47;

red_mode3 = $4b;

enb_dma3 = $3;

dis_dma3 = $7;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 { 8253 control word }  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
 selcnt0 = 0;

```

selcnt1 = $40;

selcnt2 = $80;

latch = 0;

lsb = $10;

msb = $20;

imsb = $30;

mode0 = $0;

mode1 = $2;

mode2 = $4;

mode3 = $6;

mode4 = $8;

mode5 = $c;

binary = 0;

bcd = 1;

on = 1;

off = 0;

```

{ 8273 command }

```

SOBDcmd = $a4;

ROBDcmd = $64;

SDTMcmd = $97; { use interrupt for data }

RDTMcmd = $57; { use DMA for data }

SOPMcmd = $91;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ซึ่งการใช้นั้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้ง SSIOcmd = \$a0; หา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RSIOMcmd = \$60;

GRxcmd = \$c0;

SRxcmd = \$c1;

DRxcmd = \$c5;

TxFcmd = \$c8;

TxTcmd = \$c9;

AbtTxFr = \$cc;

AbtTxTr = \$cd;

RdPortA = \$22;

RdPortB = \$23;

SetPBcmd = \$a3;

ResPBcmd = \$63;

ResDEV1 = \$1;

ResDEV2 = \$0;

{ 8273 parameter }

sobdb = \$80;

robdb = \$7f;

sdtmb = \$1;

rdtmb = \$fe;

flagstrb = \$1;

presynb = \$2;

buffb = \$4;

earintrb = \$8;

eopintrb = \$10;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hdlcb = $20;

nrzib = $1;

txrxcb = $2;

loopbkb = $4;

flagb = $20; { port k D5 }

dtrb = $2; { port b D1 }

rtsb = $1; { port b D0 }

dsrb = $4; { port a D2 }

cdb = $2; { port a D1 }

ctsb = $1; { port a D0 }

{ 8273 status }

cbsyb = $80;

cbfb = $40;

cpbfb = $20;

crbfb = $10;

rxintb = $8;

txintb = $4;

rxirab = $2;

txirab = $1;

```

```
{ tx result byte }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
txearly = $c;
txfrcom = $d;
```

```
ctserr = $f;  
abtcp = #10;  
txerror = $f0;
```

```
{ rx result byte }
```

```
Rx8b = $e0;  
Rx7b = $60;  
Rx6b = $a0;  
Rx5b = $20;  
Rx4b = $c0;  
Rx3b = $40;  
Rx2b = $80;  
Rx1b = $0;  
A1mat = $0;  
A2mat = $1;  
CRCerr = $3;  
Abtdet = $4;  
IDLdet = $5;  
EOPdet = $6;  
FRless = $7;  
Memovf = $8;  
CDfail = $a;  
RxIntOvr = $b;
```

```
rxerror = $f1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ protocol control field }
```

```
reqpoll = $1;
```

```
reqdnld = $3;
```

```
reqdump = $5;
```

```
reqsync = $7;
```

```
reptx = $9;
```

```
initsys = $b;
```

```
reqctrl = $d;
```

```
pollen = 4;
```

```
dnldlen = 2;
```

```
dumpln = 2;
```

```
synclen = 8;
```

```
replen = 2;
```

```
initlen = 30;
```

```
ctrlrlen = 10;
```

```
abnormal = $ff;
```

```
txser = $8;
```

```
rxser = $4;
```

```
rxrddy = $2;
```

```
txrddy = $1;
```

```
abnorm = $f;
```

```
start = $99;
```

เอกสารนี้เป็นเอกสารที่ **maxrx = 255**; ซึ่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
**master = \$ff**;

```
enb_tx = #6;
```

```
enb_rx = #5;
```

```
dis_tr = #7;
```

```
site_1 = 1;
```

```
site_2 = 1;
```

```
site_3 = 1;
```

```
site_4 = 1;
```

```
site_5 = 1;
```

```
site_6 = 1;
```

```
site : array [1..6] of byte =  
      (site_1,site_2,site_3,site_4,  
       site_5,site_6);
```

```
cord : array [1..2] of integer = (25,9);
```

```
maxaicho = 12;
```

```
maxaichan = 32;
```

```
maxdicho = 9;
```

```
maxdichan = 32;
```

```
maxdocho = 7;
```

```
maxdochan = 32;
```

```
maxstation = 6;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stat_def : array [1..6] of string[8] =
```

```
('STATION1', 'STATION2', 'STATION3', 'STATION4',  
'STATION5', 'STATION6');
```

```
aienb = true;  
ailog = true;  
ailab = 'ANALOG I/P';  
aiscal = 'UNIT';  
aimult = 1;  
aioffs = 0;  
aithres = -100000;  
aiavr = false;  
aidiff = false;  
aialrm = false;  
dienb = true;  
dilog = true;  
dilab = 'DIGITAL I/P';  
onmsg = 'ON';  
offmsg = 'OFF';  
dialrm = false;  
almsg = 'ALARM OCCUR';  
chkback = false;
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
doenb = true;  
dolog = true;
```

```

dolab = 'DIGITAL O/P';

doauto = false;

maxlab = 8;

maxmsg = 15;

cfg_name = 'CONFIG.NET';

site_ana : array [1..maxstation] of string[4] =
    ('.S#1', '.S#2', '.S#3', '.S#4', '.S#5', '.S#6');

site_dig : string[4] =
    ('.DIG');

{ action bit define}

high = $80;

low = $0;

dip = $0;

dom = $20;

dol = $40;

doc = $60;

type

dig = set of 1..20;

stmax = string[20];

ai = record
    ai_enb : boolean;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ai_lab : string[15];

scal : string[15];

multip : real;

offst : real;

thresh : real;

averg : boolean;

diffst : boolean;

alarm : boolean;

max_val : real;

min_val : real;

end;
di = record
di_enb : boolean;
di_lab : string[15];
on_msg : string[15];
off_msg : string[15];
enb_alarm : boolean;
alarm_msg : string[15];
chk_back : boolean;
chk_chan : integer;

end;

dop = record

do_enb : boolean;
do_lab : string[15];
on_msg : string[15];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

off_msg : string[15];

auto : boolean;

auto_ai : integer;

on_lev : real;

off_lev : real;

end;

ai_rec = record
    name : string[maxmsg];
    cai_val : real;
    avr_val : real;
    dif_val : real;
    uname : string[7];
end;

digital_log = record
    time_stmp : string[22];
    stat_name : integer;
    action : byte;
end;

```

{ action bit definition

D7 - ON,OFF

D6,D5 - 0 0 : DI

0 1 : DO max level

1 0 : DO min level

1 1 : DO chk back

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D4-D0 - POINT NUMBER (1-32) 3

```
station_cfg = record

    stat_lab : string[15];

    log_time : integer;

    ai_cfg : array [1..maxaichan] of ai;
    di_cfg : array [1..maxdichan] of di;
    do_cfg : array [1..maxdochan] of dop;
end;

analog_log = record

    log_time : array [1..6] of word;
    ai_val : array [1..maxaichan] of
word;
end;

analog_rec = record

    stamp : string[22];

    analog_val : array [1..maxaichan] of

ai_rec;

end;

const odd : dig = [1,3,5,7,9,11,13,15,17,19];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
opr_mode : byte;
```

```
sio_mode : byte;

srx_par : array [1..4] of byte;

fra_tx : array [1..4] of byte;

port_b : byte;

resultb : byte;

cntx : integer;

txrest : byte;

txdptr : integer;

txlen : integer;

txbuffer : array [0..9] of byte;

rxrest : array [0..9] of byte;

rxdptr : integer;

rxrptr : integer;

rxlen : integer;

rxbuffer : array [0..80] of byte;

stas : byte;

txaddf : byte;

controlf : byte;

netw_rec : array [1..maxstation] of station_cfg;

cfg_fi : file of station_cfg;

ai_log : analog_log;

ai_fi : file of analog_log;

dg_log : digital_log;

dg_fi : file of digital_log;

cur_ai : array [1..maxstation,1..maxaichan] of integer;
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ใช้เฉพาะภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pre_ai : array [1..maxstation,1..maxaichan] of integer;
ai_buf : array [1..maxstation,1..maxaichan] of ai_rec;
cur_di : array [1..maxstation,1..4] of byte;
pre_di : array [1..maxstation,1..4] of byte;
cur_do : array [1..maxstation,1..4] of byte;
comflag : array [1..maxstation] of byte;
site_stas : array [1..maxstation] of byte;
quit,fin,kflag : boolean;
intr12 : pointer;
intr13 : pointer;
intr1C : pointer;
hour,min,sec,s100 : word;
date,day,mon,year : word;
firstentry : array [1..maxstation] of boolean;
st,sx,sy : stmax;
timestamp : string[22];

regs : registers;

enblog : boolean;

curtime,cursite : integer;

logdev : string[15];

```

```
procedure Offcursor;
```

```
procedure Oncursor;
```

เอกสารนี้จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น สิ่งนี้จัดทำขึ้นเพื่อวัตถุประสงค์เฉพาะ และต้องอ้างถึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function Packreal(var num : real) : stmax;
```

```

function Realsostr(var num : real) : stmax;

procedure Readreal(var inre : real);

procedure Readint(var inre : integer;
                  max : integer);

function Readstr(len : integer) : stmax;

procedure Readfunc(var infunc : boolean;
                  onmsg,offmsg : stmax);

procedure Delblank(var st : stmax);

procedure Stamptime;

function CBSY : boolean;

function CPBF : boolean;

function CBF : boolean;

function CRBF : boolean;

function RXINT : boolean;

function TXINT : boolean;

function RXIRA : boolean;

function TXIRA : boolean;

procedure wait;

procedure set_OBD;

procedure res_OBD;

procedure set_DTM;

procedure res_DTM;

procedure set_OPM (var mode : byte);

procedure res_OPM (var mode : byte);

procedure set_SIOM (var mode : byte);

```

```

procedure res_SIOM (var mode : byte);

procedure res_DEV;

procedure dis_RX;

procedure sel_RX(var num : integer);

procedure RTS (opr : integer);

procedure FLAG (opr : integer);

procedure fram_TX(var num : integer);

procedure abort_FTX;

procedure Initnet;

procedure interp (func : integer);

procedure DMA1_write(bcount : word);

procedure DMA1_read(bcount : word);

procedure eoi;

procedure retintr;

function Blank(len : integer) : stmax;

procedure Rescount;

```

implementation

```

procedure Offcursor;

```

```

begin

```

```

    regs.CH := $21;

```

```

    intr($10,regs);

```

```

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure Oncursor;

begin

    regs.AH := $1;

    regs.CH := $1;

    regs.CL := $b;

    intr($10,regs);

end;

```

```

procedure Default;
var count,stat : integer;
begin
    enblog := false;
    kflag := false;
    fin := false;
    key := ' ';
    directvideo := true;
    curtime := 0;
    cursite := 1;

    oncursor;

    offcursor;

    for stat := 1 to maxstation do

        with netw_rec[stat] do

            begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ stat\_lab := stat\_def[stat]; เอกสารทุกครั้งที่มีการนำไปใช้

```

for count := 1 to maxaichan do
  with ai_cfg[count] do
    begin
      ai_enb := aienb;
      ai_lab := ailab;
      scal := aiscal;
      multip := aimult;
      offst := aioffs;
      thresh := aithres;
      averg := aiavr;
      difft := aidiff;
      alarm := aialrm;
      max_val := 0;
      min_val := 0;
    end;
  for count := 1 to maxdichan do
    with di_cfg[count] do
      begin
        di_enb := dienb;
        di_lab := dilab;
        on_msg := onmsg;
        off_msg := offmsg;
        enb_alarm := dialrm;
        alarm_msg := almsg;
        chk_back := chkback;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        chk_chan := 0;

    end;

    for count := 1 to maxdochan do

        with do_cfg[count] do

            begin

                do_enb := doenb;

                do_lab := dolab;

                on_msg := onmsg;

                off_msg := offmsg;

                auto := false;

                auto_ai := 1;

                on_lev := 0;

                off_lev := 0;

            end;

        end;

    end;

```

```

function Packreal(var num : real) : stmax;

begin

    st := '';

    str(num,st);

    delete(st,11,3);

    packreal := st;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function Realtostr(var num : real) : stmax;

var dstr,estr,rstr,mstr : string[20];

    code,eint : integer;

begin

    str(num,rstr);

    estr := copy(rstr,15,3);

    val(estr,eint,code);

    if (eint >= 0) then
        begin
            if (eint <= 3) then
                begin
                    mstr := copy(rstr,2,(eint+2));
                    dstr := copy(rstr,(eint+4),4);
                    eint := 0;
                end
            else
                begin
                    mstr := copy(rstr,2,5);
                    dstr := copy(rstr,7,4);
                    eint := eint-3;
                end;
            delete(mstr,2,1);
            str(eint,estr);

            if (eint = 0) then
                begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (num >= 0) then

            mstr := '.'+mstr+'.'+dstr

        else

            mstr := '-' +mstr+'.'+dstr;

        end

    else

        begin

            if (num >= 0) then

                mstr := '.'+mstr+'.'+dstr+'E'+testr

            else

                mstr := '-' +mstr+'.'+dstr+'E'+testr;

            end;

        end

    else

        begin

            mstr := copy(rstr,1,7);

            dstr := copy(rstr,14,4);

            mstr := mstr+dstr;

        end;

        code := 0;

        repeat

            code := code+1;

        until

            mstr[code]= '.';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น for code := 1 to (6-code) do อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        mstr := ' '+mstr;

    Realsostr := mstr;

end;

```

```

procedure Readreal(var inre : real);

```

```

var rnum : real;

```

```

    code,ep : integer;

```

```

    ch : char;

```

```

    xc,yc : integer;

```

```

const mr : set of char =

```

```

    ['0'..'9','E','e','-','.'];

```

```

label rloop;

```

```

begin

```

```

    xc := wherex+1;

```

```

    yc := wherex;

```

```

rloop:

```

```

    Oncursor;

```

```

    st := '';

```

```

    ch := ' ';

```

```

    gotoxy(xc,yc);

```

```

    write(' ');

```

```

    gotoxy(xc,yc);

```

```

    repeat

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if ch in mr then

```

```

begin
    st := st+ch;
    gotoxy(xc,yc);
    Offcursor;
    write(st);
    Oncursor;
end;
if ch = (^H) then
begin
    delete(st,length(st),1);
    gotoxy(xc+length(st),yc);
    write(' ');
    gotoxy(xc,yc);
    Offcursor;
    write(st);
    Oncursor;
end;
until
    (ch = #13) or (ch = #27);
if (ch <> #27) and (st <> '') then
begin
    sx := '';
    sy := '';

```

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับภาวใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ch := st[code];

if (ch <> 'E') and (ch <> '') then

    sx := sx+st[code];

code := code+1;

until

    (code > length(st)) or (ch = 'E') or (ch = 'e');

if (ch = 'E') or (ch = 'e') then

    repeat

        ch := st[code];

        sx := sx+st[code];

        code := code+1;

    until

        (code > length(st));

end;

if (st <> '') then

    val(sx, rnum, code);

if (ch <> #27) then

    begin

        if (code = 0) then

            begin

                if (sy <> '') then

                    begin

                        val(sy, ep, code);

                        if (ep > 0) then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ for code := 1 to ep do ครั้งที่มีการนำไปใช้

```

        rnum := rnum*10;

    if (ep < 0) then

        for code := 1 to ep do

            rnum := rnum/10;

        end;

    inre := rnum;

    if (rnum > 1E+20) or (rnum < (-1E+20)) then

        begin

            gotoxy(xc,yc);

            write('VALUE OVER RANGE press ESC');

            repeat

                ch := readkey;

            until

                (ch = #27);

            goto rloop;

        end;

    end

else

    if (ch <> #13) then

        begin

            gotoxy(xc,yc);

            write('INCORRECT FORMAT press ESC');

            ch := ' ';

            repeat

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและข้อมูลใดๆของเอกสารทุกครั้งที่มีการนำไปใช้

```

until
    (ch = #27);
goto rloop;

end;

end;

```

```
Offcursor;
```

```
end;
```

```
procedure Readint(var inre : integer;
                  max : integer);
```

```
var inum : integer;
```

```
code,ep : integer;
```

```
ch : char;
```

```
xc,yc : integer;
```

```
const mr : set of char =
```

```
['0'..'9'];
```

```
label rloop;
```

```
begin
```

```
xc := wherex+1;
```

```
yc := wherey;
```

```
rloop:
```

```
Oncursor;
```

```
ch := ' ';
```

```
st := ' ';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น `gotoxy(xc,yc)`; แปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(' ');

gotoxy(xc,yc);

repeat

    ch := readkey;

    if ch in mr then

        begin

            st := st+ch;

            gotoxy(xc,yc);

            Offcursor;

            write(st);

            Oncursor;

        end;

        if ch = (^H) then

            begin

                delete(st,length(st),1);

                gotoxy(xc+length(st),yc);

                write(' ');

                gotoxy(xc,yc);

                Offcursor;

                write(st);

                Oncursor;

            end;

    until

```

```
(ch = #13) or (ch = #27);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตีพิมพ์เผยแพร่ และจะถือว่าเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    val(st, inum, code);

    if (code = 0) and (ch <> #27) then
        begin
            if (inum > max) then
                begin
                    gotoxy(xc,yc);
                    Offcursor;
                    Setattr(blinkhigh);
                    write('OVER RANGE press ESC');
                    repeat
                        ch := readkey;
                    until
                        ch = #27;
                    Setattr(lowdisplay);
                    goto rloop
                end;
            end;

            if (code =0) and (inum <= max) then
                inre := inum
            else
                inre := 0;
            Offcursor;
        end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function Readstr(len : integer) : stmax;
```

```

label ex;

var count : integer;

      xc,yc : integer;

      ch : char;

const mr : set of char =

      [#33..#125];

begin

Oncursor;

st := '';

xc := whereX+1;

yc := whereY;

gotoxy(xc,yc);

repeat

      ch := readkey;

      if ch in mr then

begin

          st := st+upcase(ch);

          gotoxy(xc,yc);

          Offcursor;

          write(st);

          oncursor;

end;

      if ch = #0 then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ch = (^H) then
    begin
        delete(st, length(st), 1);
        gotoxy(xc+length(st), yc);
        write(' ');
        gotoxy(xc, yc);
        Offcursor;
        write(st);
        Oncursor;
    end;
until
    (ch = #13) or (ch = #27) or (length(st) = len);
if (ch = #13) and (st = '') then
    goto ex;
if (ch <> #27) then
    Readstr := st
else
    Readstr := '';

```

```
ex: Offcursor;
```

```
end;
```

```
procedure Readfunc(var Infunc : boolean;
```

```
onmsg, offmsg : stmax);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่าจะในรูปแบบใดก็ตาม ลิขสิทธิ์ทั้งหมดนี้ให้แด่บรรดาผู้แต่ง และต้องอ้างถึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
const fkey : array [1..6] of string[20] =
```

```

('Ins,Insert','-Select function ','Enter',
'-Accept function ','Esc','-Exit');

var func : boolean;

      xf,yf,xc : integer;

      ch : char;

begin

      xf := whereX+1;
      yf := whereY;
      gotoxy(1,25);
      clreol;
      gotoxy(8,25);
      for xc := 1 to 6 do
      begin
            if xc in odd then
                  Setattr(highdisplay)
            else
                  Setattr(lowdisplay);

            write(fkey[xc]);

      end;

      func := infunc;

      Setattr(blinkhigh);

      repeat

            gotoxy(xf,yf);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามละเมิดลิขสิทธิ์ของเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        true : write(onmsg);

        false : write(offmsg);

    end;

    ch := readkey;

    if (ch = #82) then

        func := not(func);

    until

        (ch = #27) or (ch = #13);

    if (ch = #13) then

        ifunc := func;

        Setattr(lowdisplay);

        gotoxy(xf,yf);

        case func of

            true : write(onmsg);

            false : write(offmsg);

        end;

    end;

end;

```

```

procedure Delblank(var st : stmax);

var' msg : stmax;

begin

    msg := st;

    repeat

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไขหรือเผยแพร่ข้อมูลไปยังผู้อื่นโดยเด็ดขาด

```

until
    (msg[1] <> ' ');

st := msg;

end;

procedure StampTime;

const
    week : array [1..7] of string =
        ('SUN', 'MON', 'TUE', 'WED', 'THR', 'FRI', 'SAT');
    month : array [1..12] of string =
        ('JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG',
        , 'SEP', 'OCT', 'NOV', 'DEC');

var intext : string[2];
    ytext : string[4];

begin
   _gettime(hour,min,sec,s100);
   _getdate(year,mon,date,day);

    str(date,intext);

    if date < 10 then
        begin
            timestamp := '0'+intext+'-'+month[mon]+'-';

            str(year,ytext);

            timestamp := timestamp+ytext;
        end
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น **else** มิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin

    timestamp := intext+'-'+month[mon]+'-';

    str(year,ytext);

    timestamp := timestamp+ytext;

end;

timestamp := timestamp+' : ';

str(hour,intext);

if (hour < 10) then
begin
    timestamp := timestamp+'0'+intext+'.';
end
else
begin
    timestamp := timestamp+intext+'.';
end;

str(min,intext);

if (min < 10) then

begin

    timestamp := timestamp+'0'+intext+'.';

end

else

begin

    timestamp := timestamp+intext+'.';

end;

str(sec,intext);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งหากมีการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (sec < 10) then
    begin
        timestamp := timestamp+'0'+intext;
    end
else
    begin
        timestamp := timestamp+intext;
    end;
end;

function CBSY : boolean;
var flag : boolean;
begin
    flag := false;
    stas := port[status];
    stas := stas and cbsyb;
    if (stas = cbsyb) then
        flag := true;
    CBSY := flag;
end;

```

```
function CPBF : boolean;
```

```
var flag : boolean;
```

```
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกข้อหนึ่งคือ flag := false; นี้อาจต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stas := port[status];  
stas := stas and cpbfb;  
if (stas = cpbfb) then  
    flag := true;  
CPBF := flag;  
end;
```

```
function CBF : boolean;  
var flag : boolean;  
begin  
    flag := false;  
    stas := port[status] and cbfb;  
    if (stas = cbfb) then  
        flag := true;  
    CBF := flag;  
end;
```

```
function CRBF : boolean;  
var flag : boolean;  
begin  
    flag := false;  
    stas := port[status];  
    stas := stas and crbfb;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
flag := true;
```

```

CRBF := flag;

end;

function RXINT : boolean;

var flag : boolean;

begin

    flag := false;
    stas := port[status];
    stas := stas and rxintb;
    if (stas = rxintb) then
        flag := true;
    RXINT := flag;

end;

function TXINT : boolean;

var flag : boolean;

begin

    flag := false;
    stas := port[status];
    stas := stas and txintb;
    if (stas = txintb) then
        flag := true;
    TXINT := flag;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 end; ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function RXIRA : boolean;

var flag : boolean;

begin

    flag := false;

    stas := port[status];

    stas := stas and rxirab;

    if (stas = rxirab) then

        flag := true;

        RXIRA := flag;

end;

function TXIRA : boolean;

var flag : boolean;

begin

    flag := false;

    stas := port[status];

    stas := stas and txirab;

    if (stas = txirab) then

        flag := true;

        TXIRA := flag;

end;

procedure wait;

begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามใช้ข้อมูลเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
 inline(\$90);

```

inline($90);

end;

procedure set_OBD;

begin

    while (CBSY) do

        wait;

        port[commd] := SOBDCmd;

        while (CPBF) do

            wait;

            port[param] := sobdb;

end;

procedure res_OBD;

begin

    while (CBSY) do

        wait;

        port[commd] := ROBDCmd;

        while (CPBF) do

            wait;

            port[param] := robdb;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะในรูปแบบใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (CBSY) do
    wait;

port[commd] := SDTMcmd;

while (CPBF) do
    wait;

port[param] := sdtmb;

end;

```

```

procedure res_DTM;
begin
    while (CBSY) do
        wait;

port[commd] := RDTMcmd;

while (CPBF) do
    wait;

port[param] := rdtmb;

end;

```

```

procedure set_OPM (var mode : byte);

begin

    while (CBSY) do

        wait;

port[commd] := SOPMcmd;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือที่สร้างขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        port[param] := mode;

end;

procedure res_OPM (var mode : byte);

begin

    while (CBSY) do

        wait;

        port[comnd] := ROPMcmd;

        while (CPBF) do

            wait;

            port[param] := mode;

        end;

    procedure set_S!OM (var mode : byte);

    begin

        while (CBSY) do

            wait;

            port[comnd] := SS!OMcmd;

            while (CPBF) do

                wait;

                port[param] := mode;

            end;

        end;

    procedure res_S!OM (var mode : byte);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 begin  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (CBSY) do
    wait;

    port[commd] := RS10Mcmd;

    while (CPBF) do
        wait;

        port[param] := mode;
    end;
end;

```

```

procedure res_DEV;
begin
    port[tmr] := ResDEV1;
    wait;
    port[tmr] := ResDEV2;
end;

```

```

procedure dis_RX;
begin
    while (CBSY) do
        wait;

        port[commd] := DRXcmd;
    end;
end;

```

```

procedure sel_RX(var num : integer);

```

```

begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    port[enb_func] := enb_rx;
end;

```

```

dma1_write(68);

rxdptr := 0;

rxrptr := 0;

srx_par[1] := maxrx;

srx_par[2] := maxrx;

srx_par[3] := master;

srx_par[4] := master;

while (CBSY) do
    wait;
port[commd] := srxcmd;
while (CPBF) do
    wait;
port[param] := srx_par[1];
while (CPBF) do
    wait;

port[param] := srx_par[2];

while (CPBF) do

    wait;

port[param] := srx_par[3];

while (CPBF) do

    wait;

port[param] := srx_par[4];

comflag[num] := rxser;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 end;  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure RTS (opr : integer);
var cmd : byte;

begin
    case opr of
        on : begin
            port_b := port_b or rtsb;
            cmd := SetPBcmd;
            end;
        off : begin
            port_b := port_b and not(rtsb);
            cmd := ResPBcmd;
            end;
    end;
    while (CBSY) do
        wait;
    port[commd] := cmd;
    while (CPBF) do
        wait;
    port[param] := port_b;

end;

procedure FLAG (opr : integer);
var cmd : byte;

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

on : begin
    port_b := port_b or flagb;
    cmd := SetPBcmd;
end;

off : begin
    port_b := port_b and not(flagb);
    cmd := ResPBcmd;
end;

while (CBSY) do
    wait;
port[commd] := cmd;
while (CPBF) do
    wait;
port[param] := port_b;
end;

```

```

procedure fram_TX(var num : integer);

```

```

begin

```

```

    port[enb_func] := enb_tx;

```

```

    dma1_read(4);

```

```

    txdptr := 0;

```

```

    txrest := 0;

```

```

    fra_tx[1] := txlen;

```

```

    fra_tx[2] := 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fra_tx[3] := site[num];

fra_tx[4] := controlf;

while CBSY do

    wait;

port[commd] := TXFcmd;

while CPBF do

    wait;

port[param] := fra_tx[1];

while CPBF do

    wait;

port[param] := fra_tx[2];

while CPBF do

    wait;

port[param] := fra_tx[3];

while CPBF do

    wait;

port[param] := fra_tx[4];

comflag[num] := txser;

end;

```

```

procedure abort_FTX;

```

```

begin

```

```

    while CBSY do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

port[commd] := abtTxFr;

```

```
end;
```

```
procedure initnet;
```

```
begin
```

```
{ init buad rate generater }
```

```
port[cntctrl] := selcnt0+mode3+lmsb+bcd;
```

```
port[count0] := 13;
```

```
port[count0] := 0;
```

```
port[cntctrl] := selcnt1+mode3+lmsb+bcd;
```

```
port[count1] := 00;
```

```
port[count1] := 10;
```

```
port[cntctrl] := selcnt2+mode0+lmsb+bcd;
```

```
port[count2] := 10;
```

```
port[count2] := 00;
```

```
{ init 8273 }
```

```
res_DEV;
```

```
wait;
```

```
res_DTM;
```

```
opr_mode := buffb+presynb+hdlcb;
```

```
set_OPM(opr_mode);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้ง sio\_mode := \$f8; ;ฯ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

res_SIOM(sio_mode);

port[enb_func] := dis_tr;

for cntx := 1 to maxstation do

    comflag[cntx] := start;

end;

procedure Rescount;
begin
    port[count2] := 10;
    port[count2] := 00;
end;

procedure Interp (func : integer);
const intr_p = $021;
      disb = $30;
      enb = $cf; { intr 4,5 }
var intr_b : byte;
begin
    intr_b := port[intr_p];

    case func of
        on : intr_b := intr_b and enb;
        off : intr_b := intr_b or disb;
    end;

    port[intr_p] := intr_b;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure Eoi;

const eoi_port = #020;

      eoi_comd = #20;

begin

      port[eoi_port] := eoi_comd;

      inline($fb);

end;

```

```

procedure Retintr;

begin

      Interp(offs);

      setintvec(#1C, intr1C);

      setintvec(13, intr13);

      setintvec(12, intr12);

end;

```

```

procedure DMA1_write( bcount : word);

var

      hibyte, lobyte, page : byte;

      segadd, ofsadd, blen : word;

begin

      blen := bcount;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีก ofsadd := ofs(rxbuffer[0]); อนึ่งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

port[clr_bptr] := 0;

port[mode_reg] := wrt_modei;

lobyte := lo(segadd+ofsadd);

hibyte := hi(segadd+ofsadd);

port[base1_add] := lobyte;

port[base1_add] := hibyte;

segadd := seg(rxbuffer[0]);

ofsadd := (ofsadd shr 4) and #fff;

page := (hi(segadd+ofsadd) shr 4) and #f;

port[page1_reg] := page;

lobyte := lo(blen);

hibyte := hi(blen);

port[base1_cnt] := lobyte;

port[base1_cnt] := hibyte;

port[mask_reg] := enb_dma1;

end;

```

```

procedure DMA1_read( bcount : word);

var

    hibyte,lobyte,page : byte;

    segadd,ofsadd,blen : word;

begin

    blen := bcount;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีลิขสิทธิ์ที่คุ้มครองโดยมหาวิทยาลัยราชภัฏวชิรเวศน์บุรีรัมย์ จึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

port[clr_bptr] := 0;

port[mode_reg] := red_mode1;

lobyte := lo(segadd+ofsadd);

hibyte := hi(segadd+ofsadd);

port[base1_add] := lobyte;

port[base1_add] := hibyte;

segadd := seg(txbuffer[0]);

ofsadd := (ofsadd shr 4) and $fff;

page := (hi(segadd+ofsadd) shr 4) and $f;

port[page1_reg] := page;

lobyte := lo(blen);

hibyte := hi(blen);

port[base1_cnt] := lobyte;

port[base1_cnt] := hibyte;

port[mask_reg] := enb_dma1;

end;

```

```
function Blank(len : integer) : stmax;
```

```
var blk : char;
```

```
st : stmax;
```

```
bc : integer;
```

```
begin
```

```
st := '';
```

```
blk := ' ';
```

```
for bc := 1 to len do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกข้อควรระวังคือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
st := st+blk;
```

```
Blank := st;
```

```
end:
```

```
end.
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้