



ปีการศึกษา 2533

ระบบเก็บข้อมูลล่วงหน้า



ปริญญาโทปีการศึกษา 2533


ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเก็บข้อมูลส่วนหน้า (FRONT END DATA ACQUISITION)

จัดทำโดย

นางสาวสุติยา	พรหมสกลิต	รหัส 301068
นายศรารุช	บานทรงกิจ	รหัส 301272
นายสมชาย	กาทอง	รหัส 301289

  
\_\_\_\_\_  
( อาจารย์ขนิษฐา แซ่ตั้ง ) อาจารย์ที่ปรึกษา

## ระบบเก็บข้อมูลส่วนหน้า

นางสาวฐิติยา พรหมสถิต	301068
นายศรารุช บานทรงกิจ	301272
นายสมชาย กาทอง	301289
อาจารย์ขนิษฐา แซ่ตั้ง	อาจารย์ที่ปรึกษา ปีการศึกษา 2533

### บทคัดย่อ

ในปัจจุบันระบบการเก็บข้อมูลด้วยไมโครโปรเซสเซอร์มีความสำคัญมากทั้งในด้านวิทยาศาสตร์และอุตสาหกรรม ระบบจัดเก็บข้อมูลส่วนหน้าหรือที่เรียกว่า ฟรอนต์เอนด์ดาตาแอคควิซิชันเป็นระบบการจัดเก็บข้อมูลแบบหนึ่งที่มีข้อได้เปรียบคือสามารถที่จะทำงานร่วมกับระบบคอมพิวเตอร์ได้โดยตรง ดังนั้นการแสดงผล และการกำหนดการทำงานของระบบจึงสามารถสั่งได้จากเครื่องคอมพิวเตอร์ ทำให้สะดวกต่อการใช้งานและการแปลผลข้อมูลต่างๆ นอกจากนี้ยังสามารถส่งสัญญาณเอาท์พุทจากเครื่องกลับไปควบคุมระบบได้อีกด้วย จึงเหมาะกับการที่ต้องการการควบคุมคุณภาพทุกชั้นตอน

## FRONT END DATA ACQUISITION

MS. THITIYA      PROMSTIT      301068  
MR. SARAWUTH    BANSONGKIT    301272  
MR. SOMCHAI      KARTHONG      301289  
MS. KANITTA      SAETANG      ADVISOR

1990.

### ABSTRACT

Now, the Data Acquisition is very importance for sciences and industrials. The Front End Data Acquisition is a kind of Data Acquisition which have advantage. Because it's able to link which computer unit. So display and set up system can control by computer. Thus it's easy to use and analyze data. Moreover it is able to supply outputs to make a closed loop control for a good quality control system.

## สารบัญ

บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎี	2
บทที่ 3	การสร้างระบบเก็บข้อมูลส่วนหน้า	19
บทที่ 4	การทดลองและผลการทดลอง	32
บทที่ 5	บทวิจารณ์และสรุป	33

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

ในขณะที่โลกเรากำลังรุดหน้าไปด้วยเทคโนโลยี ( Technology ) อันทันสมัย การเก็บข้อมูลก็มีความสำคัญมากขึ้น ดังจะเห็นได้จากในระบบอุตสาหกรรมทั่ว ๆ ไปที่จำเป็นต้องมีการควบคุมคุณภาพทุกขั้นตอน โดยจะทำการเก็บข้อมูลจากจุดต่างๆแล้วนำไปวิเคราะห์และแปรผลข้อมูลที่ได้ เพื่อลดข้อผิดพลาดของระบบให้น้อยที่สุด

จากการที่คอมพิวเตอร์ ( Computer ) เริ่มมีบทบาทมากขึ้นทุกทีในปัจจุบัน ทำให้คอมพิวเตอร์กลายเป็นอุปกรณ์คำนวณ ประมวลผล และวิเคราะห์ผลข้อมูลได้อย่างถูกต้องรวดเร็ว เพื่อทำงานในส่วนที่ยุ้งยากนี้แทนมนุษย์ ดังนั้นการนำคอมพิวเตอร์มาใช้กับการเก็บ และประมวลผลข้อมูล จึงเป็นการเหมาะสมอย่างยิ่ง ทางผู้จัดทำได้เห็นถึงความสำคัญในเรื่องนี้ จึงได้สร้างอุปกรณ์เพื่อศึกษาระบบเก็บข้อมูลส่วนหน้า ( Front End Data Acquisition ) ซึ่งเป็นระบบเก็บข้อมูลและประมวลผลชนิดหนึ่งที่มีประสิทธิภาพสูง อีกทั้ง ยังสามารถเก็บสัญญาณ ( Signal ) ทั้งที่เป็นสัญญาณแบบอนาล็อก ( Analog Signal ) และสัญญาณแบบดิจิตอล ( Digital Signal ) ได้หลายช่องสัญญาณ ( Channel ) พร้อมกัน นอกจากนี้ยังสามารถประมวลผลข้อมูลที่ได้รับเข้ามา เพื่อไปควบคุมการทำงานของระบบอีกทีหนึ่ง ซึ่งข้อมูลที่ได้นี้จะถูกนำมาแสดงผลที่คอมพิวเตอร์ จึงทำให้การแสดงผลทำได้ในหลายลักษณะ และการวิเคราะห์ข้อมูลก็ทำได้สะดวกยิ่งขึ้น ซึ่งทางผู้จัดทำหวังว่าโครงการนี้ จะเป็นประโยชน์ในการนำเทคโนโลยีการเก็บ และการประมวลผลข้อมูลมาใช้งานทางอุตสาหกรรมได้อย่างมีประสิทธิภาพ

## บทที่ 2

### ทฤษฎี

#### 2.1 ระบบเก็บข้อมูล (Data Acquisition)

เนื่องจากข้อมูลมีมากมายหลายประเภท ดังนั้นเราจึงสามารถจัดแบ่งข้อมูลได้หลายลักษณะ เช่น

ก) แบ่งตามเวลา ได้แก่

- ข้อมูลระยะสั้น
- ข้อมูลระยะกลาง
- ข้อมูลระยะยาว

ข) แบ่งตามลักษณะของสัญญาณ ได้แก่

- สัญญาณอนาล็อก (Analog Signal) ซึ่งเป็นข้อมูลแบบต่อเนื่อง
- สัญญาณดิจิทัล (Digital Signal) ซึ่งเป็นข้อมูลที่แสดงค่าเป็น

ระดับการเปลี่ยนแปลงของสัญญาณอย่างทันทีทันใด

ค) แบ่งตามลักษณะอื่นๆ

ข้อมูลเหล่านี้จะถูกจัดเก็บ ประมวลผล วิเคราะห์ และแสดงผล โดยระบบเก็บข้อมูล คือ การกระทำของข้อมูลที่อยู่ในรูปของสัญญาณดิจิทัล ตามขั้นตอนการต่อไปนี้

- การเก็บบันทึกข้อมูล (Storage)
- การประมวลสัญญาณหรือการประมวลผล (Processing)
- การส่งสัญญาณจากจุดหนึ่งไปอีกจุดหนึ่ง (Transmission)
- การแสดงผล (Display)

เนื่องจากในระบบเก็บข้อมูลนี้ สัญญาณที่ใช้เป็นสัญญาณแบบดิจิทัล ดังนั้นการแปลงสัญญาณไปมาระหว่างสัญญาณอนาล็อกกับสัญญาณดิจิทัลจึงมีความสำคัญมาก เพราะในธรรมชาติสัญญาณส่วนใหญ่จะอยู่ในรูปของสัญญาณอนาล็อก ฉะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการอ่านสัญญาณเข้ามาก็จะต้องแปลงเป็นสัญญาณดิจิทัลเสียก่อน และเช่นเดียวกัน ในการควบคุมระบบ (Control System) ก็จะต้องมีการแปลงสัญญาณดิจิทัลให้เป็นสัญญาณแบบอนาลอกก่อน แล้วจึงค่อยส่งสัญญาณนี้ไปควบคุมต่อไป

## 2.2 การแปลงดิจิทัลเป็นอนาลอก (Digital to Analog Converter)

เมื่อข้อมูลมีการประมวลผลเสร็จสิ้นแล้ว ก็จะนำเอาผลนี้ไปใช้งาน ซึ่งจำเป็นต้องมีการแปลงข้อมูลนั้นให้กลับมาเป็นสัญญาณอนาลอก วงจรแปลงนี้มีหลายชนิด คือ

- 1) สวิตช์กระแส (Current switching DAC)
- 2) สวิตช์แรงดัน (Voltage switching DAC)
- 3) แลตเตอร์เน็ตเวิร์ค (Ladder network)
- 4) สวิตช์ประจุ

1) สวิตช์กระแส ใช้ตัวต้านทานค่าต่าง ๆ ปรับให้กระแสที่ไหลจากจุดแรงดันอ้างอิงเข้าสู่วงจรผลบวก โดยบิตที่มีค่าต่ำจะใช้ความต้านทานสูงวงจรนี้จึงต้องใช้ตัวต้านทานที่มีค่าต่างกันมากๆ ดังรูป 2.1 ดังนั้น การสร้างเป็นวงจรรวมจึงยุ่งยาก

2) สวิตช์แรงดันใช้ตัวต้านทานที่มีค่าเท่ากันมาต่อแบ่งแรงดันเป็นจำนวนขั้นเท่ากับ  $2^n$  ขั้น แล้วใช้วงจรเกทต่อกันเป็นสวิตช์ปิดเปิด โดยผลของการปิดเปิดสวิตช์ จุดที่ต่อจากตัวต้านทาน ถึงวงจรขยายแบบบัฟเฟอร์เพียงจุดเดียว วงจรดังกล่าวมีลักษณะดังรูป 2.2 วงจรนี้นิยมสร้างด้วยเทคโนโลยีรวมแบบมอส

3) แลตเตอร์เน็ตเวิร์ค เป็นลักษณะผสมระหว่างสวิตช์กระแส และสวิตช์แรงดันโดยใช้ความต้านทานเพียง 2 ค่า คือ R, 2R ต่อสลับกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นันไต ดังรูป 2.3 ทำให้มีการแบ่งแรงดันที่แต่ละจุด และแต่ละจุดจะมีตัวต้านทานต่อเข้าไปยังวงจรผลบวก ซึ่งจะได้ผลบวกเป็นค่าแรงดัน โดยตัวต้านทานที่ต่อเข้ามาวงจรรวมจะต้องมีกระแสไหลคงที่

4) สวิตช์ประจุ การต่อตัวเก็บประจุร่วมกับวงจรที่เหมาะสมก็สามารถสร้างวงจรแบ่งแรงดันได้ ดังรูปที่ 2.4 วงจรนี้เหมาะกับการสร้างด้วยเทคโนโลยีแบบมอส เนื่องจากสามารถสร้างตัวเก็บประจุได้ง่าย

### 2.3 ระบบควบคุมกระบวนการ (Process control system)

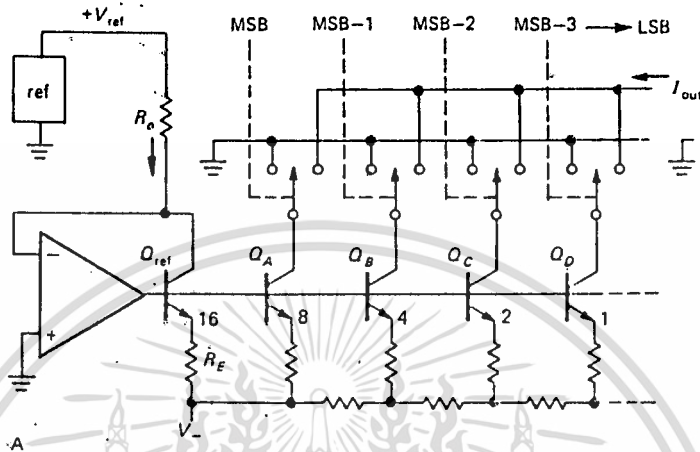
ในระบบควบคุมกระบวนการโดยทั่วไป มีวิธีการควบคุมอยู่ 2 แบบ คือ การควบคุมแบบซีควีนเชียล (Sequential) หรือการทำซ้ำตามลำดับวิธีการจนสัมฤทธิ์ผล และการควบคุมแบบป้อนกลับ (Feedback) ซึ่งเป็นการควบคุมที่มีการส่งสัญญาณป้อนกลับให้กระบวนการเป็นตัวแปรที่ควบคุมได้

จากรูป 2.5 เป็นรูปการควบคุม (Control loop) ที่แทนด้วยการควบคุมตัวแปรเดียว (single variable control) แต่ละบล็อกอธิบายได้ดังนี้

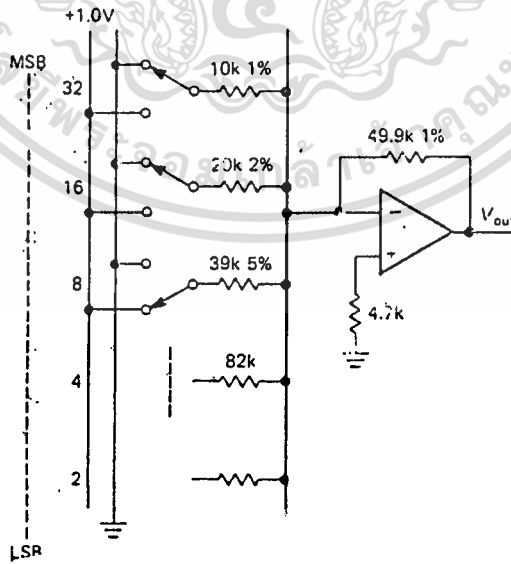
1) กระบวนการ (Process) คือกระบวนการที่สนใจทั้งหมดที่จะควบคุม ประกอบด้วยตัวแปรมากมาย ที่มีผลต่อการดำเนินการของกระบวนการ แต่ในรูป 2.5 เป็นการควบคุมเพียงตัวแปรเดียว ซึ่งตัวแปรของกระบวนการก็คือสิ่งต่างๆ ที่เมื่อแปรเปลี่ยนค่าแล้วจะมีผลกระทบต่อคุณสมบัติของกระบวนการ เช่น อุณหภูมิ ความดัน อัตราการไหล ระดับของเหลว ที่มีผลต่อสี ความหนาแน่น ส่วนประกอบทางเคมี ขนาด ของผลิตภัณฑ์ในกระบวนการผลิต ซึ่งการควบคุมกระบวนการนี้จะดำเนินไปโดยอ้างอิงกับระบบตลอดเวลา

2) เครื่องมือวัด (Measurement) คือ ส่วนที่แทนการทำงานเพื่อแสดงค่าตัวแปรที่ควบคุม โดยส่วนเอาต์พุต (Output) ของส่วนนี้คือ ค่าการวัด ที่มีรูปแบบต่างๆ คือ ค่าความต่างศักย์ กระแสไฟฟ้า หรือสัญญาณดิจิทัล

3) เออเรอร์ดีเทคเตอร์ (Error detector) การทำงานของส่วนนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

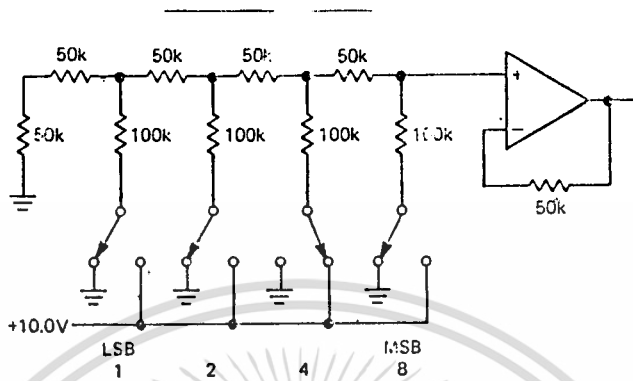


รูป 2.1 วงจรแปลงสัญญาณดิจิทัลเป็นอนาลอกแบบสวิตช์กระแส

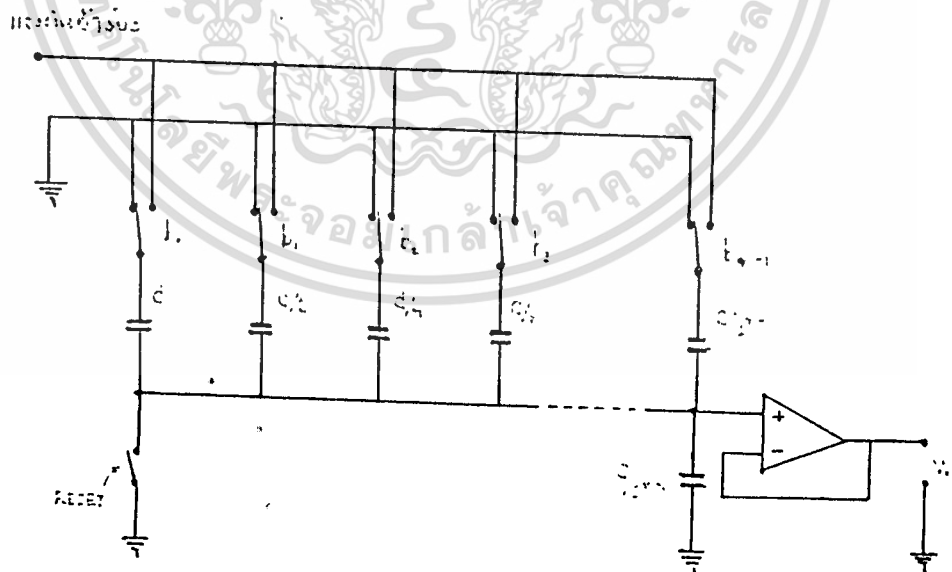


รูป 2.2 วงจรแปลงสัญญาณดิจิทัลเป็นอนาลอกแบบสวิตช์แรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.3 วงจรแปลงสัญญาณดิจิทัลเป็นอนาลอกแบบแลตเตอร์เนทเวอร์ค



รูป 2.4 วงจรแปลงสัญญาณดิจิทัลเป็นอนาลอกแบบสวิตช์ประจุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็นการเปรียบเทียบค่าการวัดตัวแปรที่ควบคุม กับค่าอ้างอิง (Set point) ค่าแตกต่างของทั้งสองค่านี้จะ เป็นเอาต์พุทของเออเรอร์ดีเทคเตอร์และสำหรับระบบควบคุมที่ใช้คอมพิวเตอร์ส่วนนี้มักจะอยู่ในส่วนของโปรแกรม

4) ตัวควบคุม (Controller) เป็นส่วนที่กำหนดการเปลี่ยนแปลงของตัวแปรควบคุม เพื่อจะกำจัดค่าผิดพลาดจากตัวแปรที่ควบคุม ซึ่งส่วนนี้ทำหน้าที่คล้ายกับเป็นสมองของระบบควบคุมเอาต์พุทของตัวควบคุม คือ สัญญาณป้อนกลับที่จะส่งต่อไปให้กับกระบวนการ เพื่อให้ตัวแปรควบคุมแปรค่าไปให้ตรงตามที่กำหนด

5) คอนโทรลลิเมนต์ (Control element) ส่วนสุดท้ายในส่วนควบคุมนี้จะ เป็นส่วนที่เปลี่ยนสัญญาณที่ได้จากตัวควบคุมซึ่งเป็นตัวแปรควบคุม เพื่อเป็นค่าอินพุท (Input) ที่กระบวนการยอมรับ ในทางปฏิบัติแล้วส่วนสุดท้ายของคอนโทรลลิเมนต์นี้จะ เป็นส่วนหนึ่งภายในกระบวนการซึ่งมีผลต่อการแปรค่าของตัวแปรกระบวนการการทำงานของลูปการควบคุมเริ่มจากตัวควบคุมจะทำหน้าที่แก้สัญญาณค่าผิดพลาดที่คลาดเคลื่อนไปจากค่าอ้างอิงซึ่งสัญญาณที่เปรียบเทียบกับค่าอ้างอิง ได้มาจากเอาต์พุท ของเครื่องมือวัดที่วัดค่ามาจากกระบวนการที่สนใจ โดยสัญญาณแก้ที่ได้จะถูกส่งไปควบคุมคอนโทรลลิเมนต์ เพื่อส่งค่าเป็นเอาต์พุทให้กับกระบวนการที่สนใจต่อไป โดยจะทำเช่นนี้ไปเรื่อยๆเป็นลูปควบคุม

#### 2.4 คอนโทรลเลอร์แอคชั่น (Controller action)

ในลูปควบคุมทั่วไป อุปกรณ์แต่ละตัว จะมีคุณสมบัติของทรานเฟอร์ฟังก์ชัน (Transfer function) และ คุณลักษณะ (Characteristic) ที่แน่นอน (ยกเว้นตัวควบคุมซึ่งสามารถเลือกรูปแบบได้) ดังนั้น เราควรเลือกโหมดการควบคุมให้เหมาะสมกับกระบวนการนั้นๆ ซึ่งเราสามารถแบ่งโหมดการควบคุมออกได้เป็น 4 โหมดด้วยกัน คือ

1) ออนออฟ คอนโทรลโหมด (On/Off Control Mode) เป็นโหมดการควบคุมที่ง่ายที่สุดใน 4 โหมดนี้ ซึ่งเป็นการควบคุมแบบ 2 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 บล็อกไดอะแกรมแสดงระบบควบคุมกระบวนการโดยทั่วไป



ดังรูป 2.6 เอาท์พุทของตัวควบคุมจะมีอยู่เพียง 2 สถานะ คือ 0% กับ 100% ในการควบคุม เมื่อสัญญาณค่าผิดพลาดมีค่าเกินกว่าพิคัดที่กำหนดไว้เอาท์พุทจากตัวควบคุมก็จะเปลี่ยนจาก 0% เป็น 100% และในทางกลับกันเอาท์พุทของตัวควบคุมจะลดลงจาก 100% เป็น 0% เมื่อค่าผิดพลาดลดต่ำลงกว่าพิคัด ซึ่งการเปลี่ยนแปลงสัญญาณของตัวควบคุมจะไม่พอดี กับการเปลี่ยนของสัญญาณค่าผิดพลาดทำให้เกิดเดดแบนด์(Dead band) รอบจุดค่าผิดพลาดศูนย์ ตัวอย่างการควบคุมในโหมดนี้เช่นการควบคุมอุณหภูมิของเครื่องปรับอากาศ การควบคุมในโหมดนี้จะเกิดการออสซิลเลต(Oscillate) ถ้าหากความถี่ของการเปลี่ยนแปลงของสัญญาณค่าผิดพลาดมาก ลมการของการควบคุมในโหมดนี้คือ

$$C = 0\% \quad ; \quad e < \alpha$$

$$C = 100\% \quad ; \quad e > \alpha$$

โดยที่ C คือ เอาท์พุทของตัวควบคุม และ  $\alpha$  คือ 1/2 ของเดดแบนด์

2) พรופןโพรชันนอล คอนโทรลโหมด (Proportional control mode) โหมดนี้เอาท์พุทของตัวควบคุมจะเป็นสัดส่วนกับสัญญาณค่าผิดพลาด โดยสัญญาณค่าผิดพลาดจะเปลี่ยนไปมากน้อยตามขนาดของอัตราขยาย ดังรูป 2.7 ความสัมพันธ์ของเอาท์พุทของตัวควบคุมกับสัญญาณค่าผิดพลาดสามารถเขียนเป็นสมการได้คือ

$$C = K_p * e + C_0$$

โดยที่ C คือ เอาท์พุทของตัวควบคุม

$K_p$  คือ อัตราขยายพรופןโพรชันนอล

e คือ ค่าผิดพลาด

$C_0$  คือ เอาท์พุทของตัวควบคุมเมื่อไม่มีค่าผิดพลาด

จากรูป 2.7 พีบิ(PB) คือพรופןโพรชันนอลแบนด์(Proportional Band)ซึ่งหมายถึงค่าเปอร์เซ็นต์ของสัญญาณค่าผิดพลาดที่เปลี่ยนไป แล้วทำให้สัญญาณเอาท์พุทของตัวควบคุมเปลี่ยนไป 100%

3) อินทิกรัล คอนโทรลโหมด(Integral control mode) ในโหมดนี้จะเป็นการอินทิเกรตค่าสัญญาณค่าผิดพลาดที่เกิดขึ้นตั้งแต่เริ่มทำการควบคุมจนถึงเวลาในปัจจุบัน เพื่อแก้ค่าผิดพลาด ตามสมการ

$$C_i(t) = K_i \int_0^t e(T) d(T) + C_i(0)$$

โดยที่  $C_i(t)$  คือ เอาท์พุทของตัวควบคุมเมื่อเวลา  $t$

$K_i$  คือ อัตราขยายอินทิกรัล

$C_i(0)$  คือ เอาท์พุทของตัวควบคุมเมื่อเวลา  $t = 0$

$e(t)$  คือ ค่าผิดพลาดที่เวลา  $t$

4) เดอริเวทีฟคอนโทรลโหมด(Derivative control mode) ในโหมดนี้ตัวควบคุมจะแก้สัญญาณค่าผิดพลาดตามการเปลี่ยนแปลงของสัญญาณค่าผิดพลาดที่เปลี่ยนแปลงไป สมการการควบคุมในโหมดนี้ คือ

$$C_d(t) = K_d * d e(t)/dt$$

โดยที่  $C_d(t)$  คือ เอาท์พุทของตัวควบคุมที่เวลา  $t$

$K_d$  คือ อัตราขยายเดอริเวทีฟ

$e(t)$  คือ ค่าผิดพลาดที่เวลา  $t$

สำหรับการใช้โหมดการควบคุมแบบอินทิกรัล และเดอริเวทีฟในระบบควบคุมทั่ว ๆ ไปจะไม่ใช้เพียงลำพังโหมดเดียว แต่จะใช้ร่วมกับโหมดพรอพอร์ชันนอล คือ

1. พีไอ โหมด

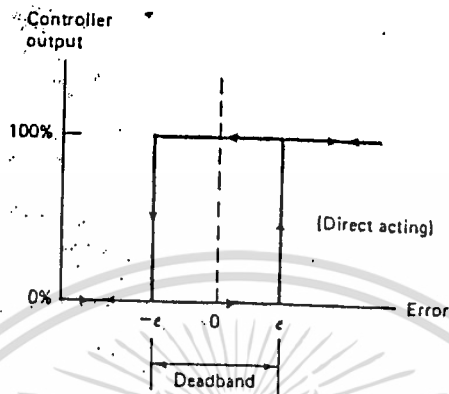
$$C_{pi} = K_p * e + K_i \int_0^t e(T) d(T)$$

2. พีดี โหมด

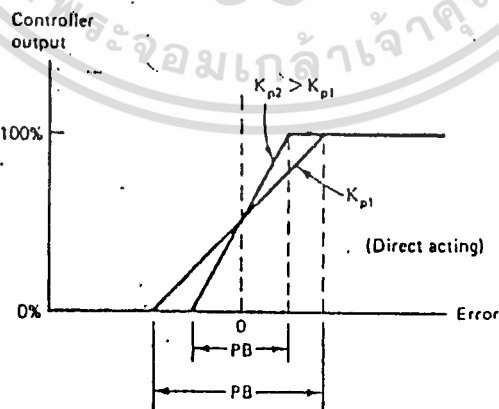
$$C_{pd} = K_p * e + K_d * d e(t)/dt + C_o$$

3. พีไอดี โหมด

$$C_{pid} = K_p * e + K_i \int_0^t e(T) d(T) + K_d * d e(t)/dt$$



รูป 2.6 การควบคุมแบบออน-ออฟ (ON-OFF CONTROL)



รูป 2.7 การควบคุมแบบพรอนโพรชันนอล (PROPORTIONAL CONTROL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 ชิปเก็ลชีพไมโครคอนโทรลเลอร์ (Single-Chip Microcontrollers)

ชิปเก็ลชีพ 8031 เป็นไมโครคอนโทรลเลอร์ในตระกูล เอ็มซีเอส 51 (MCS-51) ซึ่งไม่มีหน่วยความจำชนิดรอม (ROM) อยู่บนชิป แต่มีลักษณะเด่นดังนี้

1. ใช้ HMOS เทคโนโลยีสร้าง และ ทำงานได้ด้วยแหล่งจ่ายไฟขนาด 5 โวลต์
2. ชิพมีขนาด 8 บิท
3. มีวงจรรอสซิลเลเตอร์ (Oscillator) บนชิป
4. มีชุดแบ่งคีย์จิสเตอร์ (Bank Register) 4 ชุด
5. มีตัวตั้งเวลาและตัวนับ (Timer/Counter) ขนาด 16 บิท 2 ตัว
6. มีพอร์ตขนานไอโอแบบสองทิศทาง (Bidirectional I/O Port) 4 พอร์ต
7. มีพอร์ตการสื่อสารแบบอนุกรมที่สามารถโปรแกรมการรับ-ส่ง แบบ ฟูลดูเพล็กซ์ (Full Duplex) ที่มีความเร็วในการสื่อสารสูง
8. มีซอฟต์แวร์แฟล็ก (Software Flag) สำหรับผู้ใช้ ที่สามารถกำหนดเองได้ถึง 128 ตำแหน่งบิท
9. สามารถใช้งานแบบบูลีน (Boolean) ได้
10. มีคำสั่งการคำนวณทางคณิตศาสตร์ และสามารถใช้ได้ทั้งตัวเลขแบบ เดซิมีล (Decimal) และแบบไบนารี (Binary)
11. การกำหนดพื้นที่สแต็ก (Stack Area) สำหรับโปรแกรมย่อยต่าง ๆ ทำได้กว้าง

โครงสร้างทั่วไปของไมโครคอนโทรลเลอร์ เอ็มซีเอส 51 มีดังนี้

### 2.5.1) โครงสร้างภายนอกของ 8031

จากรูป 2.8 เป็นการจัดขาตามลักษณะภายนอกของ 8031 ซึ่งมีรายละเอียด ดังนี้

-ขา V<sub>CC</sub> (ขา 20) เป็นขาสำหรับต่อลงกราวด์ (Ground)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ขา  $V_{cc}$  (ขา 40) เป็นขาสำหรับต่อกับไฟเลี้ยง (Supply)

-พอร์ท 0 (ขา 32-39) เป็นพอร์ทไอโอ 8 บิตแบบสองทิศทาง ซึ่งสามารถขับโหลดที่ที่แอลได้สูงสุดถึง 8 ตัว และพอร์ทนี้ยังเป็นการรวมสัญญาณระหว่างแอดเดรสไบต์ต่ำ กับข้อมูลในการอ้างอิงหน่วยความจำภายนอกด้วย

-พอร์ท 1 (ขา 1-8) เป็นพอร์ทไอโอ 8 บิตแบบสองทิศทางที่สามารถขับโหลดที่ที่แอลได้สูงสุด 4 ตัว

-พอร์ท 2 (ขา 21-28) เป็นพอร์ทไอโอ 8 บิตแบบสองทิศทาง ซึ่งสามารถขับโหลดที่ที่แอลได้สูงสุด 4 ตัว พอร์ทนี้จะถูกใช้งานเป็นตัวส่งแอดเดรสไบต์สูงด้วย เมื่อใช้งานร่วมกับหน่วยความจำภายนอก เพื่อจะได้อ้างอิงแอดเดรสได้ถึง 16 บิต

-พอร์ท 3 (ขา 10-17) เป็นพอร์ทไอโอ 8 บิตแบบสองทิศทาง ที่สามารถขับโหลดที่ที่แอลได้สูงสุดถึง 4 ตัว และยังสามารถใช้งานตามฟังก์ชันพิเศษได้อีก

-ขา RST (ขา 9) ต้องคงสถานะสูง เป็นเวลาอย่างน้อย 2 วัฏจักร (Cycle) ขณะที่ต้องการรีเซ็ตทั้งระบบงาน โดยจะต่อรีซิสเตอร์พูลดาวน์ (8.2 กิโลโอห์ม) จากขา RST ลงกราวด์ และเพื่อให้ตัวชิพสามารถรีเซ็ตตัวเองได้ โดยอัตโนมัติขณะเปิดไฟจะใช้คาปาซิเตอร์ (10 ไมโครฟารัด) ต่อคร่อมระหว่างขา RST กับขาไฟเลี้ยง

-ขา ALE/PROG (ขา 30) เป็นขาแอดเดรสแลทซ์อานาเบิ้ล ใช้สำหรับแลทซ์ค่าแอดเดรสไบต์ต่ำจากพอร์ท 0 ในระหว่างการเข้าถึงหน่วยความจำภายนอก

-ขา PSEN (ขา 29) เป็นสัญญาณสโตบ (strobe) การอ่านข้อมูล จากโปรแกรมหน่วยความจำภายนอก ขา PSEN นี้จะไม่มีพัลส์ส่งออกเมื่อชิพทำงานด้วยโปรแกรมหน่วยความจำภายใน

-ขา EA/V<sub>pp</sub> (ขา 31) ถ้าขานี้มีสถานะสูง ตัวชิพจะทำงานตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่อยู่ในหน่วยความจำภายใน แต่ถ้าขานี้มีสถานะต่ำตัวซีพียูก็จะทำงานตามโปรแกรมที่อยู่ในหน่วยความจำภายนอก

-ขา XTAL1 (ขา 19) ใช้เป็นขาอินพุตเข้าสู่ตัวออสซิลเลเตอร์ในชิพ

-ขา XTAL2 (ขา 18) ใช้เป็นขาเอาต์พุตจากตัวออสซิลเลเตอร์ในชิพ

### 2.5.2) การจัดการทางสถาปัตยกรรม

จากรูป 2.9 เป็นบล็อกที่ถูกแบ่งตามลักษณะงาน ในการจัดการภายในของเอ็มซีเอส 51 โดยซึ่งเก็ลชีพแต่ละตัวของตระกูลนี้จะประกอบด้วยหน่วยประมวลผลกลางหน่วยความจำ พอร์ตอินพุต พอร์ตเอาต์พุต ไทม์ริจิสเตอร์สถานะและ ข้อมูล ส่วนวงจรตรรกในการสุ่ม(Random)ที่จำเป็นสำหรับตัวแปรของฟังก์ชัน การต่อพ่วงส่วนต่างๆที่กล่าวมานี้จะติดต่อกันด้วยบัลข้อมูลขนาด 8 บิต และ จะมีบัฟเฟอร์(Buffer)สำหรับการติดต่อข้อมูลกับภายนอกผ่านพอร์ตไอโอ

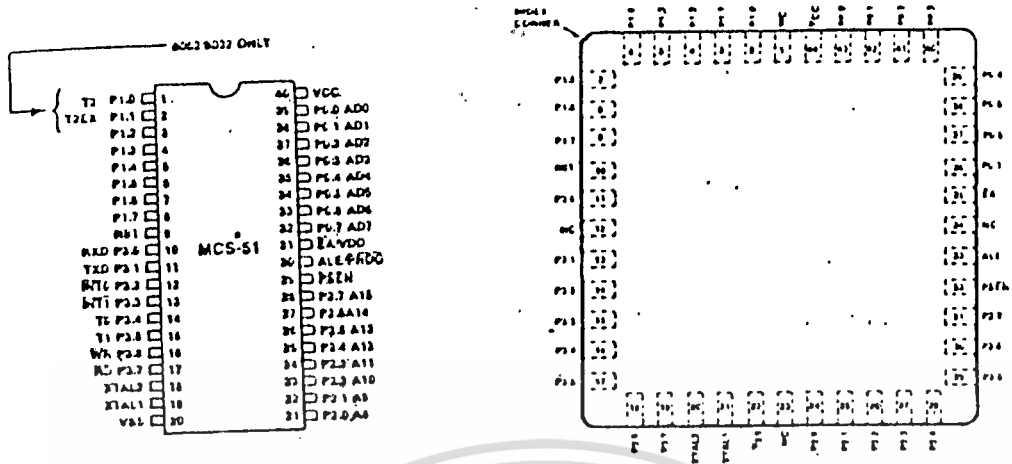
### 2.5.3) การสื่อสารทางพอร์ทอนุกรมของ 8031

การติดต่อทางพอร์ทอนุกรมของเอ็มซีเอส51นี้ จะต้องผ่านทางรีจิสเตอร์ SBUF(Serial Port Receive and Transimit Buffer Register) และ เราสามารถจะควบคุมการรับส่งให้เป็นลักษณะใด นั้น โดยการเลือกโหมด(Mode)การทำงานโหมดใดโหมดหนึ่งภายใน 4 โหมด ซึ่งในโครงงานนี้ใช้โหมด 1 คือ จะมีการรับส่งข้อมูลขนาด 10 บิต ( โดยเป็นข้อมูล 8 บิต สตาร์ทบิต(Start Bit) 1 บิต และเป็นสตอปบิต(Stop Bit) อีก 1 บิต ) และเรายังสามารถเปลี่ยนบอดเรต(Baud Rate)ได้ตามสูตรดังนี้

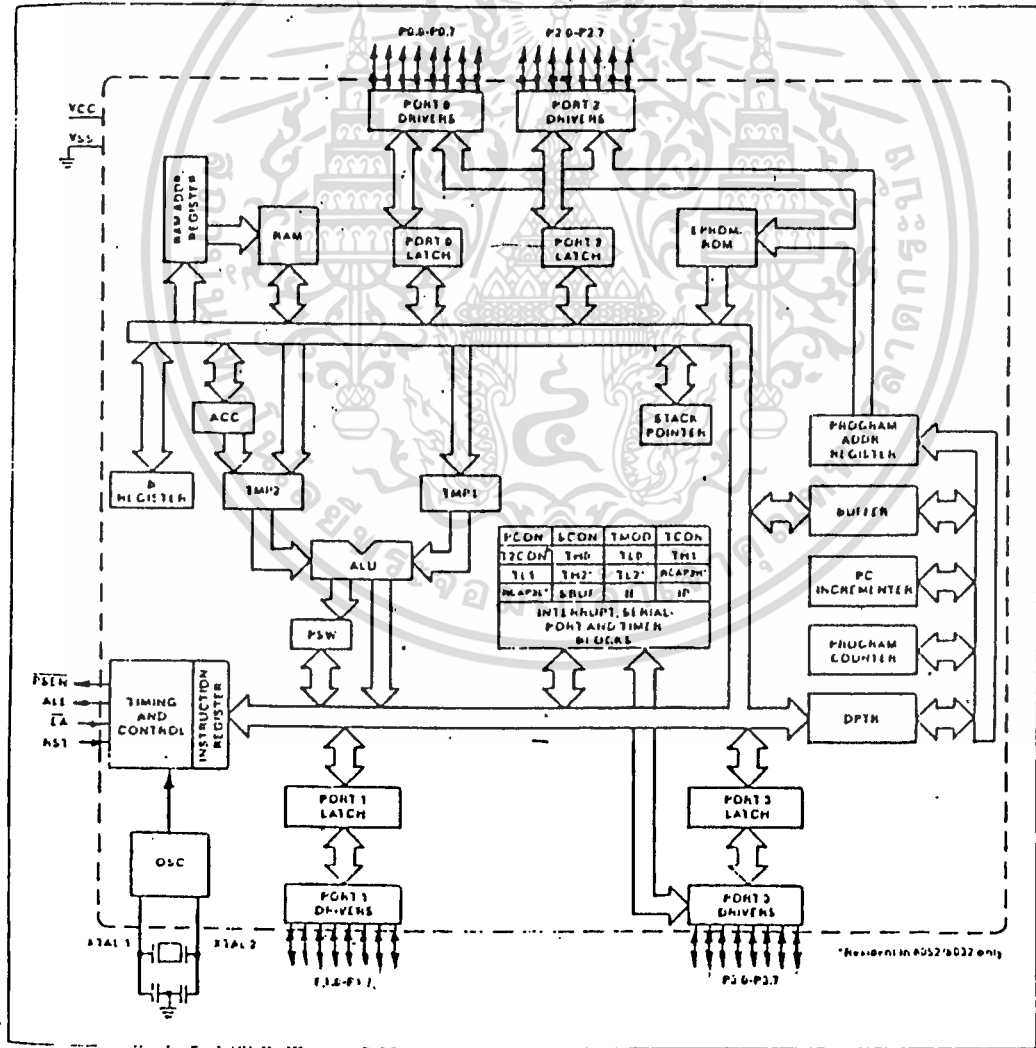
$$BAUD\ RATE = (2^{SMOD} / 32) * (OSC.\ FREQ. / \{ 12 * [ 256 - (TH1) ] \})$$

โดย SMOD เป็นบิตที่อยู่ใน PCON

TH1 เป็นแรมที่อยู่ในแรมภายใน(Internal Ram)ของ 8031 ซึ่งเราสามารถกำหนดค่าของมันได้



รูป 2.8 การจัดขาภายนอกของ 8031



รูป 2.9 สถาปัตยกรรมของ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6) การส่งข้อมูลด้วยพอร์ทอนุกรม

การส่งข้อมูลแบบอนุกรมมีข้อดีคือ เป็นการส่งข้อมูลที่ใช้สายสัญญาณน้อย เส้น และส่งได้ในระยะทางไกล ๆ โดยจะส่งข้อมูลในลักษณะเป็นขบวนของ สัญญาณดิจิทัลในขนาดที่เป็นคำ หรือเป็นกลุ่มของคำในลักษณะเป็นขบวนที่จะ เรียงลำดับจากข้อมูลค่าต่ำไปสูง ซึ่งมีการส่งอยู่สองลักษณะคือ

- 1) ส่งแบบซิงโครนัส (Synchronous)
- 2) ส่งแบบอะซิงโครนัส (Asynchronous)

ทั้งสองแบบจะมีรูปแบบมาตรฐานในการส่งที่ต่างกันที่จะขอเน้นในที่นี้ คือ การส่งแบบอะซิงโครนัสซึ่งเป็นที่นิยมใช้มากกว่า เนื่องจากสามารถใช้งานได้ง่าย ข้อมูลเชื่อถือได้สูง และต่อเข้ากับโมเด็มได้

การส่งข้อมูลระหว่างพอร์ทอนุกรมนั้นข้อมูลที่ส่งต้องมีรูปแบบตามที่กำหนด ในรูปแบบการส่ง (Protocol) ดังนี้

- 1) มีบิตเริ่มต้น (Start Bit) เป็นบิตที่แสดงสถานะต่างจากสถานะว่าง ซึ่งเป็นหนึ่ง ดังนั้นบิตเริ่มต้นจึงต้องมีสถานะเป็นศูนย์
- 2) จำนวนบิตข้อมูลเลือกได้สองขนาด คือ 7 บิตและ 8 บิต
- 3) บิตร่วมสถานะ (Parity Bit) บิตนี้ทำให้ผลรวมของจำนวนบิตที่เป็นหนึ่งมีจำนวนรวมเป็นคู่หรือคี่ที่แน่นอน
- 4) บิตหยุด (Stop Bit) เพื่อแสดงว่าค่านั้น ๆ สิ้นสุดแล้ว โดยจะ ให้สถานะเป็นหนึ่งซึ่งเป็นสถานะเดียวกับสัญญาณว่าง

พอร์ทการส่งข้อมูลแบบอนุกรมนั้น จะประกอบไปด้วย

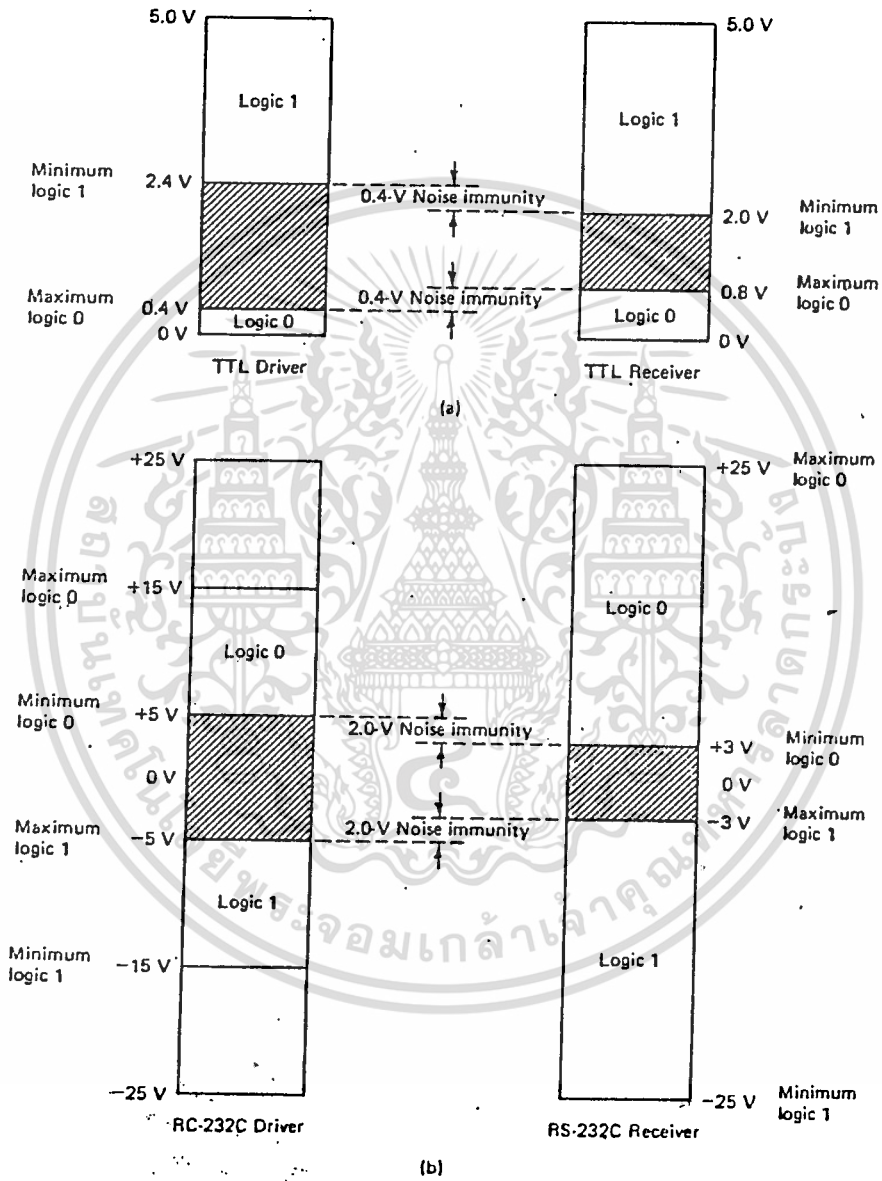
- 1) สัญญาณนาฬิกาของการส่ง (Baud Rate) ซึ่งมีค่ามาตรฐานคือ 9600 4800 2400 1200 600 300 150 110 และ 75 บิตต่อวินาที เพื่อให้พอร์ทที่ทำการรับและส่งสามารถทำงานประสานกันได้อย่างถูกต้อง

- 2) สัญญาณควบคุมการอ่านเขียนจากซีพียู ซึ่งเป็นตัวกำหนดการให้ใช้งานว่าจะให้ใช้งานเมื่อใด และให้ทำงานในลักษณะการอ่านหรือเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) สัญญาณควบคุมโมเด็มในการต่อพอร์ทอนุกรมเป็นพอร์ทสื่อสารในระยะไกลนั้นทำได้โดยใช้โมเด็ม ซึ่งโมเด็มนี้เองจะเป็นตัวช่วยประสานงานให้การติดต่อถูกต้องยิ่งขึ้น เช่น ช่วยในการเชื่อมต่อระบบเข้ากับระบบโทรศัพท์

4) ตัวแปลงข้อมูลและเก็บข้อมูลชั่วคราว เนื่องจากข้อมูลที่ได้รับจากซีพียูนั้นเป็นข้อมูลแบบขนาน จึงต้องแปลงข้อมูลให้เป็นอนุกรมเสียก่อน โดยมีซีพริซิลเตอร์เป็นตัวทำหน้าที่นี้ นอกจากนี้ข้อมูลที่ส่งไปไม่ทันกับการทำงานของซีพียู จึงต้องมีการรอโดยจะเก็บข้อมูลไว้ในหน่วยความจำชั่วคราว (Buffer) ก่อนสำหรับวงจรรวมที่ใช้เป็นพอร์ทอนุกรมที่รู้จักกันดีได้แก่ ไอซี 8250 และ 8251 เนื่องจากสัญญาณที่เอาท์พุทของวงจรรวมเป็นสัญญาณระดับ 0 และ 5 โวลต์ ซึ่งเป็นสัญญาณที่มีช่วงแคบและ วัตถุประสงค์การรับทวนจึงได้มีข้อตกลงให้ใช้สัญญาณใหม่โดยให้สถานะสูงอยู่ที่ช่วงแรงดัน -5 โวลต์ถึง -15 โวลต์และสถานะต่ำอยู่ที่ 5 โวลต์ถึง 15 โวลต์ จากรูป 2.10 จะเปรียบเทียบช่วงสัญญาณของทีทีแอล และ อาร์เอส 232ซี จะเห็นได้ว่าภูมิต้านทานต่อสัญญาณรบกวน (Noise Immunity) ในทีทีแอลเท่ากับ 0.4 โวลต์ แต่ในอาร์เอส 232ซีเท่ากับ 2 โวลต์ดังนั้นการส่งข้อมูลแบบอนุกรมตามมาตรฐานอาร์เอส 232ซี จึงเหมาะสมกับการใช้งานสื่อสารระยะไกล เช่น การออนไลน์ (Online) ผ่านโครงข่ายโทรศัพท์ รวมทั้งระบบเก็บข้อมูลส่วนหน้านี้ด้วย



รูป 2.10 เปรียบเทียบภูมิต้านทานต่อสัญญาณรบกวนระหว่าง  
สัญญาณแบบทีทีแอลกับอาร์เอส 232 ซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

## การสร้างระบบเก็บข้อมูลส่วนหน้า

### 3.1 วัตถุประสงค์ของระบบเก็บข้อมูลส่วนหน้า

1) เป็นตัวกลางในการติดต่อระหว่าง ระบบคอมพิวเตอร์กับส่วนรับข้อมูล (input port) จำนวนมาก ซึ่งส่วนรับข้อมูลเหล่านี้จะอยู่ในสภาวะใดก็ได้ ซึ่งบางสภาวะของส่วนรับข้อมูลนั้นไม่เหมาะสมที่สิ่งมีชีวิตจะอยู่ได้ เครื่องมือชิ้นนี้ จึงอำนวยความสะดวกในการติดต่อกับระบบที่เราต้องการจะควบคุม โดยผู้ใช้ (user) สามารถสั่งงานได้โดยตรงจากคอมพิวเตอร์

2) เนื่องจากในระบบเก็บข้อมูลส่วนหน้านี้ มีหน่วยประมวลผลกลาง (CPU) ภายใน เพราะฉะนั้นหน่วยประมวลผลกลางภายในคอมพิวเตอร์จึงทำหน้าที่ในการส่ง และรับข้อมูลจากระบบเก็บข้อมูลส่วนหน้าและพิมพ์กราฟเท่านั้น ทำให้ไม่เสียเวลาในการประมวลผล

3) การตอบสนองต่อการเปลี่ยนแปลงของสัญญาณอินพุตเป็นไปอย่างรวดเร็ว รวมทั้งสามารถเลือกช่อง (channel) ในการตอบสนองได้ถึง 256 ช่องสัญญาณ

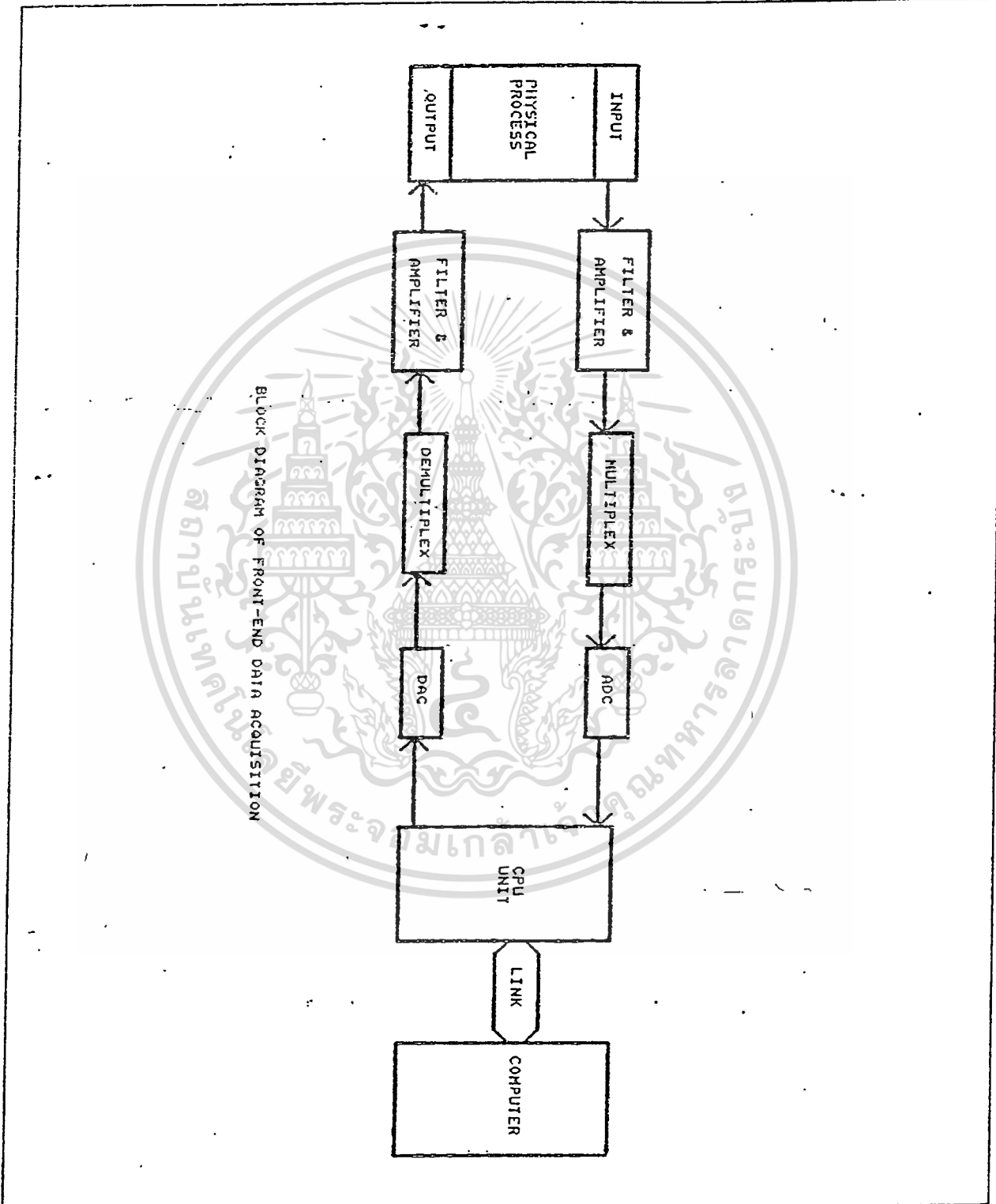
4) เนื่องจากในระบบเก็บข้อมูลส่วนหน้านี้ ไม่ได้กำหนดชนิดของสัญญาณอินพุตไว้ เพราะฉะนั้นจึงสามารถนำไปใช้กับสัญญาณอินพุตชนิดใดก็ได้

5) ช่วยลดความเสี่ยงและค่าใช้จ่ายในการทำงาน เพราะในสภาวะแวดล้อมที่ไม่เหมาะสม สามารถให้ระบบเก็บข้อมูลส่วนหน้าเข้าไปควบคุมการทำงานแทนได้

### 3.2 บล็อกไดอะแกรม (Block Diagram) ของระบบเก็บข้อมูลส่วนหน้า

บล็อกไดอะแกรมการทำงานของระบบเก็บข้อมูลส่วนหน้า แสดงได้ดังรูป

3.1 ซึ่งเริ่มจากจะมีการส่งผ่านคำสั่งต่าง ๆ จากคอมพิวเตอร์เข้ามาเก็บไว้ที่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



BLOCK DIAGRAM OF FRONT-END DATA ACQUISITION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้บล็อกโดยคณะกรรมการทำงานของระบบเก็บข้อมูลส่วนหน้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเก็บข้อมูล เช่น ช่องสัญญาณที่ต้องการติดต่อด้วยเป็นต้นเมื่อมีการรับคำสั่งต่างๆเรียบร้อยแล้วก็จะเริ่มทำงาน โดยจะมีการรับข้อมูลจากตัวทรานสดิวเซอร์ (transducer) ซึ่งเป็นตัวเซนเซอร์ (Sensor) สัญญาณต่าง ๆ เข้ามา สัญญาณที่เข้ามานี้อาจจะอยู่ในรูปของศักดา (Voltage) หรือกระแส (Current) ก็ได้ ซึ่งมีขนาดของสัญญาณต่ำมากๆ จึงต้องมีการส่งสัญญาณนี้ไปที่ วงจรขยายสัญญาณ (Amplifier) เพื่อขยายสัญญาณ แล้วส่งเข้าวงจรกรองความถี่เพื่อลดสัญญาณรบกวน (Noise) และเนื่องจากระบบเก็บข้อมูลนี้มีการติดต่อกับอินพุทหลายช่องสัญญาณจึงต้องมีการรวมสัญญาณ (multiplex) เพื่อให้อินพุทหลายช่องสัญญาณสามารถรวมกันได้ โดยแบ่งช่องเวลาดังตามลำดับ หลังจากนั้นข้อมูลก็จะถูกส่งไปที่ส่วนเอดีซี (ADC) เพื่อแปลงข้อมูลที่เซนเซอร์รับเข้ามาให้อยู่ในรูปที่ระบบเก็บข้อมูลสามารถนำไปใช้ในการประมวลผลได้ ต่อจากนั้นก็ทำการประมวลผลข้อมูลจนเสร็จ ข้อมูลที่ได้นี้จะถูกส่งไปที่คอมพิวเตอร์เพื่อแสดงผล และก็จะถูกนำไปใช้ในการควบคุมการทำงานของส่วนเอาต์พุท (Output Port) ที่ระบบเก็บข้อมูลนี้ควบคุมอยู่ เนื่องจากข้อมูลที่ถูส่งไปที่ ส่วนเอาต์พุทนั้นต้องเป็นสัญญาณแบบอนาล็อก ดังนั้นจึงต้องมีการแปลงสัญญาณที่ส่งมาจากระบบเก็บข้อมูลให้เป็นสัญญาณแบบอนาล็อกก่อน โดยการส่งสัญญาณนี้ไปที่ส่วนดีเอซี (DAC) และเนื่องจากระบบเก็บข้อมูลนี้มีการติดต่อกับส่วนเอาต์พุทหลายช่องสัญญาณ จึงต้องส่งสัญญาณนี้ไปที่ตัวแยกสัญญาณ (demultiplex) เพื่อเลือกช่องสัญญาณเอาต์พุทให้ตรงกับที่เราต้องการติดต่อด้วย

### 3.3 โครงสร้างทางฮาร์ดแวร์ (Hardware) ของระบบเก็บข้อมูลส่วนหน้า

จากบล็อกไดอะแกรมตามรูป 3.1 สามารถออกแบบสร้างวงจรได้ดังนี้

- 1) ส่วนประมวลผลกลาง (CPU) และส่วนกำเนิดความถี่ (Oscillator)
- 2) ส่วนถอดรหัส (Decoder)
- 3) ส่วนหน่วยความจำ (Memory)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) ส่วนแสดงผล(Display)
- 5) ส่วนการสื่อสารแบบอนุกรม(Serial Port)
- 6) ส่วนเอาต์พุต(Output Port)

1) ส่วนประมวลผลกลางและส่วนกำเนิดความถี่

เนื่องจากพอร์ท 0 (Port 0) เป็นการรวมสัญญาณระหว่างแอดเดรสกับข้อมูล ดังนั้นการติดต่อของซีพียูกับหน่วยความจำภายนอกทำได้โดยส่งค่าพอร์ท 0 ออกไปแลตช์(Latch)ไว้ที่ U2 (เบอร์ 74LS373)ก่อน เมื่อซีพียูต้องการส่งแอดเดรสก็จะมีการส่งสัญญาณ ALE (Address Latch Enable) เป็น '1' ออกไปอีน่าเบิล(Enable) U2 ให้ทำงาน จึงจะมีการส่งแอดเดรสผ่าน U2 ได้ ส่วนพอร์ท 2 ใช้เป็นแอดเดรสไบต์สูง(A8-A15)ได้เลยเพราะไม่มีการรวมสัญญาณระหว่างแอดเดรสกับข้อมูล

ในโครงงานนี้ใช้คริสตอล(Crystal)ขนาด 11.0592 MHz เป็นตัวสร้างสัญญาณนาฬิกา(Clock)และใน 1 แมชชีนไซเคิล(Machine Cycle) จะประกอบด้วย 6 สเตท(states) และในแต่ละสเตทจะประกอบด้วย 2 เฟส(Phase)และแต่ละเฟสจะใช้เวลาเท่ากับ 1 สัญญาณนาฬิกาและใน 1 แมชชีนไซเคิล จะมีการส่งสัญญาณ ALE ออกมา 2 ลูก เพื่อนำไปใช้ในการอีน่าเบิล U2 ให้ทำงาน

2) ส่วนถอทรหัส

วงจรส่วนนี้ จะทำหน้าที่ในการถอทรหัสค่าแอดเดรสที่ส่งมาจากซีพียูว่าต้องการติดต่อกับส่วนใดของระบบเก็บข้อมูล เช่น ส่วนหน่วยความจำ ส่วนแสดงผล หรือส่วนเอาต์พุต เป็นต้น ซึ่งได้แบ่งแอดเดรสของระบบไว้ดังนี้

- 2.1) รอม(Rom) มีแอดเดรสอยู่ในช่วง 0000H-1FFFH
- 2.2) แรม(Ram) มีแอดเดรสอยู่ในช่วง 2000H-3FFFH
- 2.3) ส่วนที่ทำการติดต่อกับช่องสัญญาณต่างๆ แบ่งออกเป็น

2.3.1) ส่วนแสดงผล(LED Port) มีแอดเดรสอยู่ในช่วง 4000H-43FFFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2) ส่วนบอดเรต(Baud Rate) มีแอดเดรสอยู่ในช่วง 4400H-47FFH

2.3.3) ส่วนช่องสัญญาณอินพุท มีแอดเดรสอยู่ในช่วง 5000H-57FFH

2.3.4) ส่วนช่องสัญญาณเอาต์พุท มีแอดเดรสอยู่ในช่วง 5800H-5FFFH

### 3) ส่วนหน่วยความจำ

วงจรส่วนนี้เป็นส่วนของหน่วยความจำ รอม(Rom) และแรม(Ram) โดยจะเลือกหน่วยความจำชนิดใดนั้น ขึ้นอยู่กับสัญญาณที่ได้จากการถอดรหัสและสัญญาณต่างๆจากซีพียู กล่าวคือ ถ้าเลือกรอมก็จะมีคำสั่งสัญญาณ PSEN (Program Store Enable)หรือสัญญาณ RD (Read)จากซีพียู มาที่ขา OE (Output Enable)ของรอม พร้อมทั้งสัญญาณที่ได้จากการถอดรหัสก็ต้องมาอีน่าเบิลรอมที่ขา CE (Control Enable)ด้วย และเมื่อต้องการติดต่อกับแรมก็จะมีสัญญาณที่ได้จากการถอดรหัสมาอีน่าเบิลแรมที่ขา CS1(ขา enable) รวมทั้งซีพียูต้องส่งสัญญาณ RD หรือสัญญาณ WR (Write)ออกมาด้วย เมื่อต้องการอ่านหรือ เขียนหน่วยความจำตามลำดับ

### 4) ส่วนแสดงผล

วงจรในส่วนนี้จะมีแอลอีดี(LED)เป็นตัวแสดงผล ซึ่งแอลอีดีจะสว่างเมื่อมีข้อผิดพลาดในการทำการตรวจสอบตัวเอง(Self Test) และจะบอกว่าอุปกรณ์ชนิดใดในระบบที่ยังไม่พร้อมที่จะทำงาน และในขณะที่ระบบเก็บข้อมูลทำการประมวลผลอยู่นั้น ถ้าข้อมูลที่รับเข้ามาเกิดการผิดพลาดขึ้น กล่าวคือ ข้อมูลมีค่าสูงกว่าหรือต่ำกว่าค่าอ้างอิง(Set point)ที่ได้กำหนดไว้ ก็จะมีแอลอีดีอีกชุดหนึ่งเป็นตัวแสดงผลเช่นกัน

### 5) ส่วนการสื่อสารแบบอนุกรม

ส่วนนี้เป็นส่วนของการสื่อสารแบบอนุกรมแบบที่ง่ายที่สุด กล่าวคือ ไม่มีการส่งสัญญาณแฮนด์ เซ็ค(Handshake)กันระหว่างคอมพิวเตอร์ กับระบบเก็บข้อมูล ซึ่งข้อมูลจากคอมพิวเตอร์จะต้องมีการปรับระดับของสัญญาณให้เป็นระดับทีทีแอล(TTL)ก่อน โดยใช้ไอซีเบอร์ MC1489 และเช่นเดียวกัน ข้อมูล

จากระบบเก็บข้อมูลซึ่งเป็นระดับทิกิตแอล ก็ต้องถูกแปลงให้อยู่ในระดับมาตรฐาน RS-232C ก่อน โดยใช้ไอซีเบอร์ MC1488

6) ส่วนเอาต์พุต

ส่วนนี้เป็นส่วนของการส่งค่าเอาต์พุตออกไปควบคุมระบบ ซึ่งจะใช้พอร์ท 0 ในการส่งข้อมูลแต่พอร์ทนี้มีการรวมสัญญาณระหว่างแอดเดรสกับข้อมูลอยู่ ดังนั้นจึงต้องทำการอินาเบิ้ลตัวเลขข้อมูลให้ทำงานก่อน แล้วข้อมูลก็จะถูกป้อนเข้าสู่ไอซี DAC0808 เพื่อแปลงให้เป็นสัญญาณอนาลอก ซึ่งจะให้ค่าเอาต์พุตตามสูตร

$$V_o = V_{REF} [A_0 2^0 + A_1 2^{-1} + A_2 2^{-2} + \dots + A_{N-1} 2^{-(N-1)}]$$

ข้อมูลจาก DAC0808 นี้เป็นสัญญาณกระแส จึงต้องผ่านเข้าสู่ไอซี LF351N เพื่อทำการแปลงให้เป็นสัญญาณโวลเตจก่อน สัญญาณอนาลอกที่ได้นี้จะผ่านเข้าสู่อนาลอกสวิตช์ (Analog Switch) ซึ่งมีการเลือกช่องสัญญาณจากข้อมูลที่ส่งมาจากซีพียู ข้อมูลที่ออกมาจากอนาลอกสวิตช์นี้จะถูกส่งไปยังช่องสัญญาณเอาต์พุตที่ได้ทำการเลือกแล้ว เพื่อควบคุมระบบต่อไป

3.4) โครงสร้างทางซอฟต์แวร์

เป็นส่วนประกอบที่สำคัญอีกส่วนหนึ่ง ของระบบเก็บข้อมูลส่วนหน้าเสมือน เป็นหัวใจของตัวระบบ การทำงานของส่วนซอฟต์แวร์ เริ่มโดยการอ่านโปรแกรม (ซึ่งถูกแปลงเป็นซอร์สโค้ด (Source Code) เก็บไว้ใน อีพ็รอม) เข้ามาในซีพียู 8031 และทำการประมวลผลตามโปรแกรมที่อ่านเข้ามา แล้วส่งผลไปแสดงผ่านทางส่วนฮาร์ดแวร์ โดยมี โปรแกรมเคาน์เตอร์ (Program Counter) เป็นตัวแสดงว่า ต้องอ่านโปรแกรมที่ตำแหน่งเท่าไร ความเร็วในการทำงานของส่วนซอฟต์แวร์ ขึ้นอยู่กับ สัญญาณนาฬิกาที่ส่งให้ ซีพียูทำงานดังนั้นจะเห็นว่าการทำงานที่จะทำความเข้าใจถึงการออกแบบและ การทำงานของระบบรับข้อมูลส่วนหน้านี้ เราต้องเข้าใจถึงส่วนประกอบทางฮาร์ดแวร์และขั้นตอนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บวนการทำงานของส่วนซอฟต์แวร์ให้ละเอียดเสียก่อน ส่วนทางฮาร์ดแวร์ได้  
กล่าวมาแล้วในหัวข้อก่อน ในที่นี้จะกล่าวถึงส่วนทางซอฟต์แวร์

ส่วนโปรแกรมทางซอฟต์แวร์แบ่งเป็น 3 ส่วน ดังนี้

1) โปรแกรมควบคุมการทำงานของระบบเก็บข้อมูลส่วนหน้า

ในการทำงานของโปรแกรมส่วนนี้ จะเป็นการทำงานของการจัดระบบ  
ต่างๆ ตั้งแต่เซตค่าคอนโทรลรีจิสเตอร์ หรือ รีจิสเตอร์ฟังก์ชันพิเศษ รับค่า  
พารามิเตอร์ในโหมดควบคุม กำหนดขั้นตอนการทำงาน การรับข้อมูลที่ผ่านมา  
ทาง เอพูตี การส่งข้อมูลที่ประมวลผลได้ ผ่านทาง ดิทัเอ การแสดงผลข้อมูล  
ทางส่วนแสดงผล การรับค่าข้อมูลจากผู้ใช้ผ่านทางคอมพิวเตอร์ การจัดส่วน  
หน่วยความจำข้อมูลภายนอก (แรม 6264) ในการเก็บข้อมูลต่างๆ ตลอดจน  
การตรวจสอบเงื่อนไขการทำงานของระบบเก็บข้อมูล ในการเขียนโปรแกรม  
ส่วนนี้จะต้องทำความเข้าใจถึง ขั้นตอน กระบวนการทำงาน ของส่วนวงจร  
ฮาร์ดแวร์ให้ละเอียดว่ามีการทำงานอย่างไรระหว่างการทำงานของโปรแกรม

2) โปรแกรมทวงคณิตศาสตร์ ในการทำงานของส่วนนี้ จะเป็นการทำ  
งานโดยนำข้อมูลที่มีอยู่ มาประมวลผล ซึ่งได้แก่ การ บวก ลบ คูณ หาร ซึ่ง  
ขั้นตอนการทำงานในการประมวลผลจะมีส่วนสำคัญมาก เพราะถ้าโปรแกรมที่  
เขียนนั้นทำงานได้รวดเร็ว ก็จะส่งผลให้ การทำงานของระบบเก็บข้อมูลนั้น  
เร็วขึ้นด้วย

ลักษณะของการเก็บข้อมูลในโครงงานนี้ ข้อมูล 1 ค่า จะถูกเก็บโดยใช้

4 ไบต์

ไบต์ที่ 1 แสดงเครื่องหมายโดย 00 = + , FF = -

ไบต์ที่ 2,3 แสดงส่วน จำนวนเต็ม

ไบต์ที่ 4 แสดงส่วนทศนิยม

ตัวอย่าง เช่น

00 00 01 00 = 256.5 ส่วน 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00 FF 01 00 = 272055 ฐาน 10

### 3) โปรแกรมฟังก์ชันการควบคุมแบบพีไอดี

โดยมีหลักการคือรับสัญญาณอนาลอก จากกระบวนการมาประมวลผลด้วย สมการดิฟเฟอเรนเชียล เพื่อให้ 8031 ประมวลผลได้ แล้วแปลงเป็นสัญญาณอนาลอก เป็นเอาต์พุตของตัวควบคุม

การแปลงสมการในโหมดการควบคุมแบบ พีไอดี

$$\begin{aligned}
\text{จาก } C &= K_p + K_i \int E dt + K_d dE/dt \\
&= K_p + (E + K_i/K_p \int E dt + K_d/K_p dE/dt) + K \\
&= K_p + (E + 1/T_i \int E dt + T_d dE/dt) + K
\end{aligned}$$

$$C_n = K_p [E_{n+1}/T_i * E_1(t) + T_d(E_n - E_{n-1})/t] + K$$

$$C_{n-1} = K_p [E_{n+1} + \Delta t/T_i * E_1(t) + T_d/\Delta t(E_{n-1} - E_{n-2})] + K$$

$$\begin{aligned}
C_n - C_{n-1} &= K_p [(E_n - E_{n-1}) + t/T_i * E_n + t_d/t(E_n - 2E_{n-1} - E_{n-2})] \\
&= K_p (1 + \Delta t/T_i + T_d/\Delta t) E_n - K_p (1 + 2T_d/\Delta t) E_{n-1} \\
&\quad + K_p (T_d/\Delta t) E_{n-2}
\end{aligned}$$

กำหนดให้

$$A_0 = K_p (1 + \Delta t/T_i + T_d/t)$$

$$A_1 = K_p (1 + 2T_d/\Delta t)$$

$$A_2 = K_p T_d/\Delta t$$

$$\text{เราจะได้ } C_n = C_{n-1} + A_0 * E_n - A_1 * E_{n-1} + A_2 * E_{n-2}$$

ซึ่งรายละเอียดของซอฟต์แวร์ในส่วนของระบบเก็บข้อมูล

ประกอบด้วย 5 ส่วนใหญ่ ๆ ตามรูป 3.2 คือ

1) ส่วนเซตการเริ่มต้น (Initialization) เป็นส่วนที่ทำหน้าที่ในการเซตค่าเริ่มต้นต่าง ๆ ที่เกี่ยวข้องกับการทำงานของระบบเก็บข้อมูล

2) ส่วนตรวจสอบระบบ (self test) เป็นส่วนที่ทำหน้าที่ในการตรวจสอบ

ความพร้อมของอุปกรณ์ที่อยู่ในส่วนของระบบเก็บข้อมูล หรือส่วนที่ระบบเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ข้อมูลติดต่อด้วย

3) ส่วนอินพุท(Input) เป็นส่วนที่ทำหน้าที่ ควบคุมการติดต่อระหว่างอินพุท ต่างๆ กับระบบเก็บข้อมูล

4) ส่วนเอาต์พุท(Output) เป็นส่วนที่ทำหน้าที่ควบคุมการติดต่อระหว่างเอาต์พุท กับระบบเก็บข้อมูล

5) ส่วนประมวลผล(Process) เป็นส่วนประมวลผลข้อมูลที่ได้รับเข้ามา เพื่อจะได้นำไปใช้ต่อไป

ส่วนเซ็คการเริ่มต้น

ประกอบด้วยส่วนต่าง ๆ ดังนี้คือ

1) ส่วนเซ็คการเริ่มต้นของพอร์ท เป็นการเซ็คค่าต่างๆ ให้กับพอร์ทซึ่งในส่วนนี้ยังแบ่งออกเป็นส่วนย่อย ๆ คือ

1.1) ส่วนเซ็คการเริ่มต้นของ พอร์ทอนุกรมทาง อินพุทและเอาต์พุท ส่วนนี้เป็นการวางรูปแบบการติดต่อระหว่างตัวระบบเก็บข้อมูล กับคอมพิวเตอร์ โดยผ่านทางพอร์ทอนุกรม มาตราฐาน อาร์เอส232

1.2) ส่วนเซ็คการเริ่มต้นของข้อมูล เนื่องจากในส่วนนี้ มีการติดต่อระหว่างระบบเก็บข้อมูล กับพอร์ทอินพุทและเอาต์พุท รวมถึงมีการถ่ายเทข้อมูลระหว่างระบบเก็บข้อมูล กับ คอมพิวเตอร์ จึงต้องมีการกำหนดส่วนของหน่วยความจำที่จะใช้เก็บข้อมูลว่าส่วนใดจะทำหน้าที่ใด เช่น ส่วนที่ทำหน้าที่ในการเก็บโปรแกรมทางจอภาพ และส่วนที่ใช้เก็บข้อมูลจากคอมพิวเตอร์ ซึ่งส่วนนี้ยังประกอบด้วยส่วนย่อย ๆ คือ

1.2.1) ส่วนที่ทำหน้าที่ในการเก็บข้อมูลที่ถูกอ่านมาจากอินพุท(Unprocessed data) ซึ่งข้อมูลในส่วนนี้จะถูกส่งไปทำการประมวลผลข้อมูลบนระบบเก็บข้อมูล และ บนคอมพิวเตอร์ ต่อไป

1.2.2) ส่วนที่ทำหน้าที่ในการเก็บข้อมูลที่ผ่านขบวนการ พิโอดี (Processes data)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.3) ส่วนเก็บข้อมูลที่ผิดปกติ (Alarm Condition) ส่วนนี้จะ เป็น  
เนื้อที่ในการเก็บข้อมูลที่แสดงว่ามีความผิดปกติของข้อมูล  
ส่วนตรวจสอบระบบ

ส่วนนี้จะเป็นการตรวจสอบการทำงานของอุปกรณ์ต่าง ๆ ภายในระบบ  
ซึ่งจะแบ่งเป็น ส่วนต่าง ๆ ดังนี้คือ

1) ส่วนตรวจสอบระบบ ของรีจิสเตอร์ภายใน (internal register)  
คือการตรวจสอบการทำงานของรีจิสเตอร์ ที่อยู่ภายในระบบ ซึ่งในซีพียู 8031  
นี้จะประกอบด้วยรีจิสเตอร์ ต่างๆ เช่น ตัวคำนวณ (accumulator) เป็นต้น  
ซึ่งโปรแกรม ในส่วนนี้จะตรวจสอบการทำงานของรีจิสเตอร์ โดยการรับ ข้อมูลเข้าไปที่ รีจิส  
เตอร์ แล้วทำการอ่านข้อมูลออกมา แล้วเปรียบเทียบว่าข้อมูลที่เขียนเข้าไปกับ  
ที่อ่านออกมาเหมือนกันหรือไม่ ถ้าไม่เหมือนกันแสดงถึง ความผิดพลาดในการ  
ทำงานของรีจิสเตอร์

2) ส่วนตรวจสอบระบบ ของ หน่วยความจำภายนอก (external  
memory) ซึ่งแบ่งเป็นส่วนต่าง ๆ ดังนี้คือ

2.1) ส่วนของแรม (ram) ซึ่งจะใช้ในการเก็บข้อมูลต่างๆ เช่น ข้อมูล  
ที่อ่านจาก พอร์ตอินพุท เป็นต้น ซึ่งส่วนนี้จะทำการตรวจสอบโดยการใส่ข้อมูล  
เข้าไปแล้วอ่านข้อมูลออกมา (read/write storage) แล้วทำการตรวจสอบ  
ว่าเหมือนกัน หรือไม่ ถ้าไม่เหมือนกันแสดงการผิดพลาดของแรม ที่ทุกตำแหน่ง  
ของแรม และเมื่อพบข้อผิดพลาดก็ต้องมีการแจ้งไปที่ผู้ใช้ (user)

ส่วนพอร์ตอินพุท

ส่วนนี้คือส่วนที่ทำหน้าที่ในการรับข้อมูลต่าง ๆ เข้ามาให้กับ ระบบเก็บ  
ข้อมูล แบ่งออกเป็น ส่วนต่าง ๆ ดังนี้คือ

1) ส่วนของอินพุท ที่ถ่ายเทข้อมูลผ่าน พอร์ตอินพุท ซึ่งจะทำการถ่าย  
เทข้อมูลจากคอมพิวเตอร์ ให้กับ ระบบเก็บข้อมูล ซึ่งข้อมูลที่จะถ่ายเทแบ่งเป็น

1.1) ส่วนเลือกประมวลผล (process selection) คือส่วนที่ทำหน้า  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ในการเลือกแชลแนล(channel) ในการควบคุมหรือเลือกสภาวะการทำงานต่างๆ ซึ่งการเลือกข้อมูลในส่วนนี้จะต้องถูกส่งให้กับ ระบบเก็บข้อมูล เพื่อควบคุมการทำงานให้ถูกต้อง และการคัดเลือก ในส่วนนี้จะแบ่งออกเป็น

1.1.1) การเลือกช่องสัญญาณทางอินพุต(input select channel) ส่วนนี้ทำหน้าที่ในการเลือกแชลแนล ในการควบคุมการทำงาน ซึ่งในการออกแบบจะมีแชลแนล ทั้งหมด 256 แชลแนล ซึ่งสามารถเลือก แชลแนล ใดก็ได้

1.1.2) การเลือกอุณหภูมิ(temperature select) ส่วนนี้ทำหน้าที่ในการตั้ง ค่าอุณหภูมิ ที่ต้องการควบคุมการทำงาน ของระบบประกอบด้วยค่าต่าง ๆ ดังนี้

1.1.2.1) ค่าเซตพอยท์(set point) คือค่าอุณหภูมิที่ต้องการควบคุม

1.1.2.2) ค่าสูงสุด(maximum point) คือค่าสูงสุด ของอุณหภูมิที่ระบบยอมรับได้ ถ้าเกินจากค่านี้ต้องมีการแจ้งให้ผู้ใช้ทราบ

1.2) ข้อมูลที่ผ่านขบวนการพีไอดี (PID Control) คือข้อมูลที่ผ่านขบวนการของ พีไอดี เพื่อนำเอาผลลัพธ์ไปให้กับระบบเพื่อใช้ในการควบคุมการทำงานให้เป็นไปตามความต้องการ ส่วนประมวลผล

ส่วนนี้จะทำหน้าที่เกี่ยวข้องกับขบวนการต่าง ๆ ที่เกิดขึ้นบนตัวระบบเก็บข้อมูล ในส่วนนี้ประกอบด้วยส่วนต่าง ๆ คือ

1) การให้ค่า เซ็ตพอยท์ ,ค่าสูงสุด ,ค่าต่ำสุด และ แชลแนล ที่เลือกไว้และส่งผ่านค่าเหล่านี้ไปที่ระบบเก็บข้อมูล

2) ทำการเปรียบเทียบข้อมูลของ พอร์ทขนาน กับค่าของเซตพอยท์ ,ค่าสูงสุด ,ค่าต่ำสุด

ส่วนเอาต์พุต

ส่วนนี้จะทำหน้าที่ส่งข้อมูลต่าง ๆ จากระบบเก็บข้อมูล ไปที่ส่วนต่าง ๆ แบ่งออกเป็นส่วนต่าง ๆ ดังนี้

1) พอร์ตอนุกรม คือการส่ง ข้อมูลที่เป็นการแสดงสภาวะการทำงานซึ่งจะแบ่งออกเป็น

1.1) ข้อมูลที่แสดงถึงความผิดพลาดในการควบคุมการทำงานของระบบซึ่งจะถูกส่งกลับไปให้ ผู้ใช้ ทราบเพื่อทำการแก้ไขต่อไป

1.3) ข้อมูลที่ได้จากการตรวจสอบพบความผิดพลาดภายใน ของระบบเก็บข้อมูล และมีการแจ้งให้ ผู้ใช้ ทราบเพื่อทำการแก้ไขต่อไป

2) ข้อมูลแบบขนาน จากระบบเก็บข้อมูล ซึ่งในส่วนนี้แบ่งออกเป็น

2.1) ข้อมูลที่เป็นดิจิทัล(digital) ซึ่งแสดงสภาวะการทำงานของระบบที่เกิดจากขบวนการบนระบบเก็บข้อมูล

2.2) ข้อมูลที่เป็นอนาล็อก(analog) เป็นข้อมูล ที่ได้รับจาก คอมพิวเตอร์แล้วผ่านตัว ดิจิเอ เพื่อส่งออกไปที่ พอร์ตเอาต์พุต ในการควบคุมระบบ

รายละเอียดของซอฟต์แวร์ ที่อยู่บน คอมพิวเตอร์

เนื่องจากระบบเก็บข้อมูล นี้สามารถทำงานได้ด้วยตัวเอง โดยไม่ต้องพึ่งระบบคอมพิวเตอร์ ดังนั้น คอมพิวเตอร์จะทำหน้าที่ควบคุมการส่งผ่านคำสั่ง และการเชื่อมต่อการทำงานต่างๆของระบบเก็บข้อมูล ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้

1) ส่วนเริ่มต้น(Initialization) ส่วนนี้จะเช็คสภาวะเริ่มต้นการทำงานของระบบให้ทำงานได้ดียิ่งขึ้น เช่น การจองเนื้อที่ในหน่วยความจำ การเช็คความเร็วของการส่งข้อมูล เพื่อให้เหมาะสมกับการทำงานของระบบ เป็นต้น

2) ส่วนรับข้อมูล(Input) ส่วนนี้เป็นการนำเอาข้อมูลทั้งจากระบบเก็บข้อมูล และโปรแกรมการควบคุมต่างๆจากภายนอกเข้ามาเพื่อใช้ในการควบคุม ซึ่งส่วนนี้จะแบ่งออกเป็น

2.1) ส่วนของคีย์บอร์ด(Keyboard) ซึ่งจะรับคำสั่งต่าง ๆ จากผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้ามา ซึ่งสามารถแยกออกเป็นส่วนตัวต่าง ๆ ได้ดังนี้คือ

2.1.1) ในส่วนของการเลือกช่องสัญญาณนี้จะเป็นการเลือกพอร์ตอินพุตที่ต้องการติดต่อด้วยเพื่อนำข้อมูลมาใช้ในการควบคุม ส่วนช่องสัญญาณใดไม่ถูกเลือกก็จะไม่ทำการพูลลิ่งข้อมูล(Polling Data)เข้ามา

2.1.2) การเซ็ตอุณหภูมิ ซึ่งจะทำการปรับค่าอุณหภูมิของแต่ละช่องสัญญาณ ให้เป็นไปตามความต้องการของผู้ใช้ ซึ่งจะเซ็ตค่าต่างๆ ดังนี้คือ ค่าคาดหมาย , ค่าสูงสุด , ค่าต่ำสุด

2.1.3) สตาร์ท(Start) ก่อนที่มีการเลือกออพชั่น (Set Option)นี้ ต้องมีการเลือกช่องสัญญาณ และตั้งค่าอุณหภูมิเรียบร้อยแล้ว

2.1.4) สตอป(Stop) เป็นการหยุดการทำงานของระบบเก็บข้อมูล

2.1.6) พิมพ์กราฟ(Print Graph) ส่วนนี้จะทำหน้าที่ในการแสดงกราฟการทำงานของระบบที่กำลังควบคุมตลอดเวลา

3) เอาท์พุท(Output) เป็นส่วนที่แสดงถึงการทำงานของระบบที่ส่งมาจากคอมพิวเตอร์เพื่อควบคุมการทำงานให้ถูกต้อง แบ่งออกเป็นหลายๆส่วนดังนี้

3.1) เอาท์พุททางจอภาพของคอมพิวเตอร์ ที่แสดงถึงเมนู(Menu)การเซ็ตค่าต่าง ๆ ทางคีย์บอร์ด(Keyboard)ของผู้ใช้

3.2) เอาท์พุทในส่วนของดิสค์(disk) ซึ่งเป็นส่วนของหน่วยความจำภายนอกซึ่งแบ่งออกเป็น

3.2.1) การเก็บข้อมูลที่อ่านมาจากอินพุทพอร์ต

3.2.2) การเก็บผลลัพธ์ของพีไอดี เพื่อตรวจสอบความถูกต้องในการทำงาน

3.2.3) เอาท์พุทในส่วนของระบบเก็บข้อมูล คือข้อมูลต่างๆ ที่ต้องการส่งจากคอมพิวเตอร์ไปยังตัวระบบเก็บข้อมูล เพื่อควบคุมการทำงานของเอาท์พุทพอร์ตต่าง ๆ

3.2.3.1) ส่งผลลัพธ์ของพีไอดีไปที่ระบบเก็บข้อมูล เพื่อให้มีการส่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลเหล่านี้ไปควบคุมเอาที่ผู้ที่ทำการควบคุมอยู่ต่อไป

3.2.3.2) ส่งโค้ดคำสั่ง (Command Code) ไปที่ระบบเก็บข้อมูลคือ การที่คอมพิวเตอร์มีการส่งคำสั่งต่างๆไปที่ระบบเก็บข้อมูล แล้วระบบเก็บข้อมูล ก็จะนำเอาโค้ดที่ส่งไปเทียบกับโค้ดที่เก็บไว้ที่ระบบเก็บข้อมูล เพื่อให้รู้ว่าเป็น โค้ดใด แล้วจึงทำงานตามคำสั่งนั้นๆ



## บทที่ 4

### การทดลองและผลการทดลอง

1. สร้างวงจรอาร์ดแวร์ตามที่ได้ออกแบบในบทที่ 3 ซึ่งมีรายละเอียดดังนี้
  - 1.1 หน่วยประมวลผล ชิงเกิ้ลชิพ 8031 ขนาด 8 บิต
  - 1.2 หน่วยความจำโปรแกรมภายนอก 8 กิโลไบต์
  - 1.3 หน่วยความจำข้อมูลภายนอก 8 กิโลไบต์
  - 1.4 สัญญาณอินพุตและเอาต์พุตมีขนาด 1-5 โวลต์ หรือ 4-20 โวลต์
  - 1.5 มีช่องสัญญาณอินพุตและเอาต์พุตอย่างละ 256 ช่องสัญญาณ
2. การทดลองในส่วนของโปรแกรม ทำได้ดังนี้
  - 2.1 ทดสอบการทำงานของระบบเก็บข้อมูลส่วนหน้า ทาง  
ด้านวงจรอาร์ดแวร์ในส่วนต่างๆดังนี้
    - 2.1.1 โปรแกรมส่วนทดสอบการติดต่อระหว่างไมโครคอน  
โทรลเลอร์ 8031 กับหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลภายนอก
    - 2.1.2 โปรแกรมส่วนทดสอบการทำงานของวงจรดิจิทัลลอจิก  
อนาล็อกคอนเวอร์เตอร์
    - 2.1.3 โปรแกรมทดสอบการรับค่าจากคอมพิวเตอร์
    - 2.1.4 โปรแกรมทดสอบการแสดงผล ผ่านหน่วยแสดงผล
3. ทดสอบการทำงานของระบบเก็บข้อมูลส่วนหน้า โดยกำหนดค่าสัญญาณ  
อินพุตให้กับตัวควบคุมและให้ผู้ใช้กำหนดค่าพารามิเตอร์ตามการอินนิเชียลของ  
ระบบเก็บข้อมูลส่วนหน้า คือ ค่าเวลาหน่วงของโหมดควบคุม (Kp, Ti, Td)  
ค่าขีดจำกัดสูงสุดและต่ำสุดของสัญญาณอินพุต และค่าอ้างอิงของระบบ วัด  
สัญญาณเอาต์พุตที่ได้แล้วนำมาเปรียบเทียบกับค่าที่ได้ทางทฤษฎี

FRONT END DATA ACQUISITION

MAIN MENU

SELECT CHANNEL  
LOAD FID  
LOAD DATA  
PRINT GRAPH  
START  
STOP  
EXIT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRONT END DATA ACQUISITION

MAIN MENU

SELECT CHANNEL  
LOAD PID  
LOAD DATA  
PRINT GRAPH  
START  
STOP  
EXIT

SUB MENU

select channels for temperature  
-----  
(m,M) modified ch&temp  
(q,Q) quit  
-----

Message

FRONT END DATA ACQUISITION

MAIN MENU

SELECT CHANNEL  
LOAD PID  
LOAD DATA  
PRINT GRAPH  
START  
STOP  
EXIT

SUB MENU

select channels for value pid  
-----  
(m,M) modified ch & pid  
(q,Q) quit  
-----

Message

FRONT END DATA ACQUISITION

MAIN MENU

SELECT CHANNEL  
LOAD PID  
LOAD DATA  
PRINT GRAPH  
START  
STOP  
EXIT

SUB MENU

select channel to print graph  
  
what's the card (1-32).?  
card.....1  
  
what channel (1-8)....1

Message

FRONT END DATA ACQUISITION

MAIN MENU

SELECT CHANNEL  
LOAD PID  
LOAD DATA  
PRINT GRAPH  
START  
STOP  
EXIT

SUB MENU

select your baud rate...  
9600 4800 2400 1200  
1200

Message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป

1. เราสามารถทำการติดต่อระหว่างไมโครคอนโทรลเลอร์ เอ็มซีเอส 51 ไมโครคอมพิวเตอร์ผ่านทาง อาร์เอส 232ซีได้ และสามารถนำมาใช้ประโยชน์ในงานควบคุมได้อย่างมีประสิทธิภาพตามลักษณะโปรแกรมที่เขียนขึ้น เพื่อจะได้นำมาประยุกต์ใช้กับงานควบคุมที่เหมาะสม

2. ระบบเก็บข้อมูลส่วนหน้ามีความเหมาะสมกับการควบคุมแบบอัตโนมัติ โดยเฉพาะระบบที่ต้องควบคุมผ่านทางไมโครคอมพิวเตอร์เพราะ ในระบบเก็บข้อมูลส่วนหน้านี้เป็นระบบที่ต่ออินพุตและเอาต์พุตได้หลายช่องสัญญาณ และสัญญาณเหล่านี้จะอยู่ในรูปแบบไคท์ได้ อาทิเช่น ในรูปของกระแสหรือโวลต์เตจ เป็นต้น นอกจากนี้การทำงานของระบบเก็บข้อมูลส่วนหน้าทำให้คอมพิวเตอร์รับภาระ เพียงแค่การแสดงผลและการบันทึกข้อมูลเพื่อเก็บใส่หน่วยความจำเท่านั้น ดังนั้นคอมพิวเตอร์จึงมีเวลาที่จะประมวลสัญญาณอื่นได้ เราจึงสามารถต่อระบบเก็บข้อมูลส่วนหน้าหลาย ๆ ระบบเข้ากับคอมพิวเตอร์เพียงเครื่องเดียวได้โดยผ่านโครงข่ายการสื่อสารสาธารณะ เช่น โครงข่ายโทรศัพท์ทำให้ควบคุมได้จากระยะไกลและลดปัญหายุ่งยากในการวางโครงข่ายเนื่องจากเป็นโครงข่ายที่ใช้สายสัญญาณเพียง 3 เส้น ระบบนี้จึงใช้มากในการควบคุมระบบขนาดใหญ่

## บทวิจารณ์

### อาร์ตแวร์

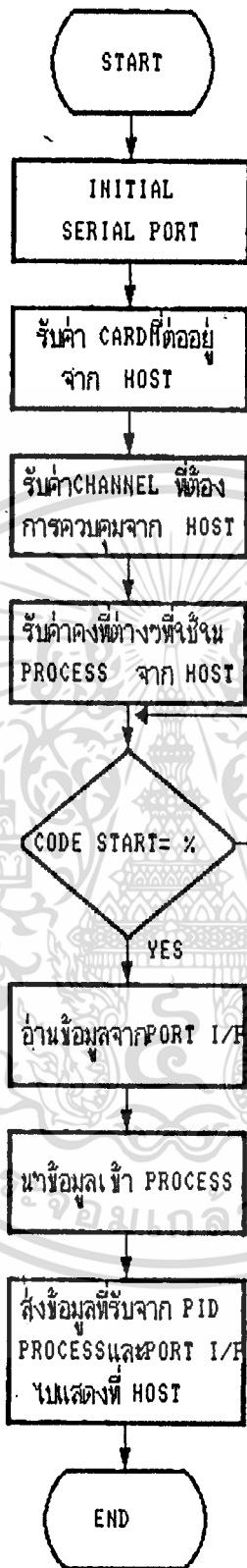
สำหรับระบบเก็บข้อมูลส่วนหน้าที่ได้สร้างขึ้น โดยใช้ไมโครคอนโทรลเลอร์ 8031 เป็นหลักนี้ ได้ออกแบบวงจรหน่วยความจำโปรแกรมภายนอกสำรองเอาไว้ 8 กิโลไบต์ เพื่อเอาไว้ใช้พัฒนาโปรแกรมทางด้านออปติ้มัมคอนโทรล ซึ่งจะสามารถพัฒนาต่อไปได้ในอนาคต

### ซอฟต์แวร์

ในการควบคุมขบวนการ โดยใช้ตัวควบคุมแบบอนาล็อก จะมีปัญหาในการปรับเปลี่ยนค่า แต่ในระบบเก็บข้อมูลส่วนหน้าที่ใช้โปรแกรมในการปรับเปลี่ยนค่าจึงสามารถที่จะจัดเก็บข้อมูลและนำข้อมูลนั้นมาทำการประมวลผลได้โดยสะดวก. นอกจากนี้ ข้อมูลที่ยังไม่ได้ทำการประมวลผล และผ่านการประมวลผลแล้วจะถูกนำไปแสดงผลออกเป็นกราฟ ทำให้สามารถที่จะพิจารณาถึงผลของการทำงานของระบบเก็บข้อมูลส่วนหน้าได้อย่างมีประสิทธิภาพมากขึ้น

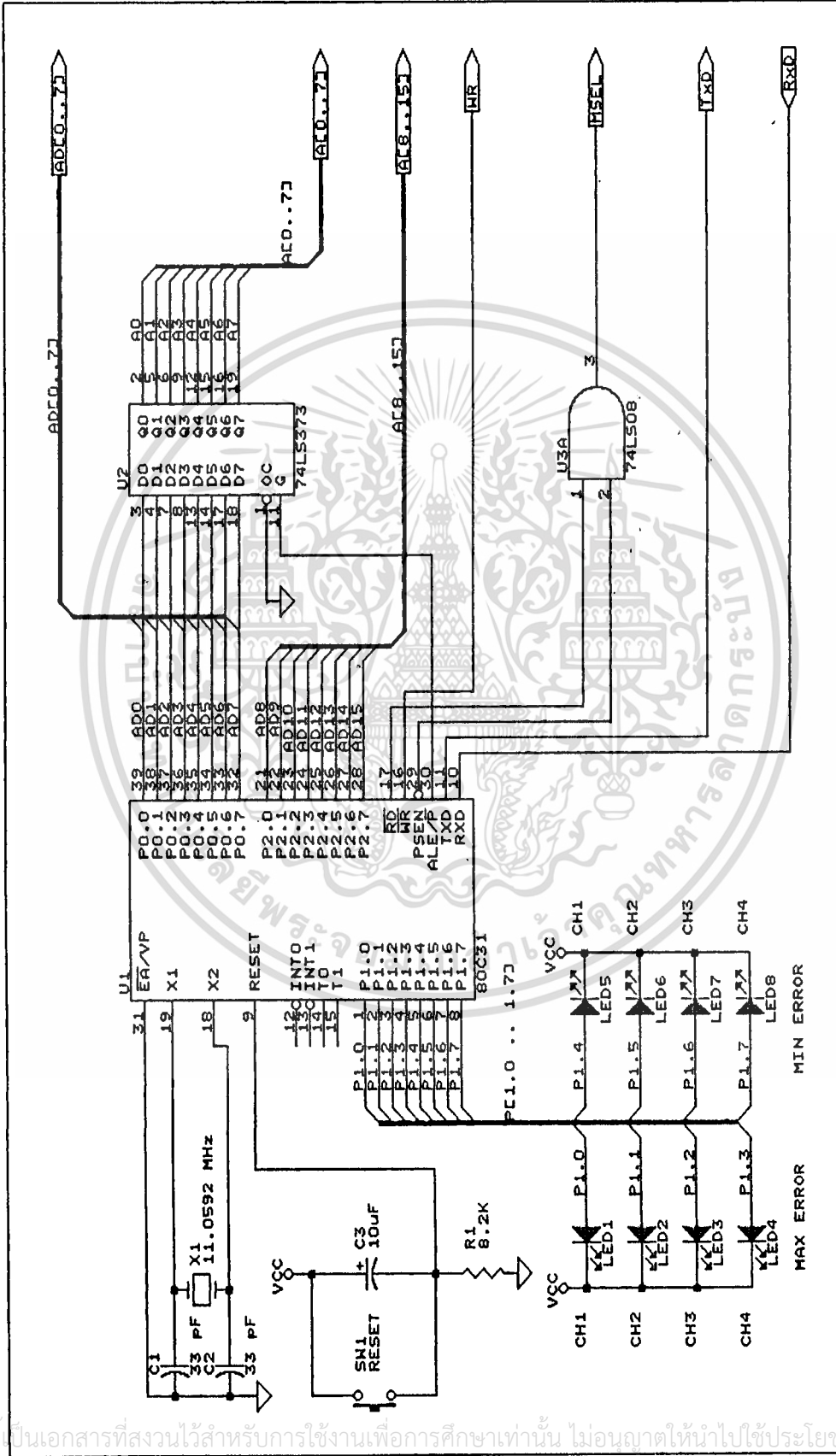


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



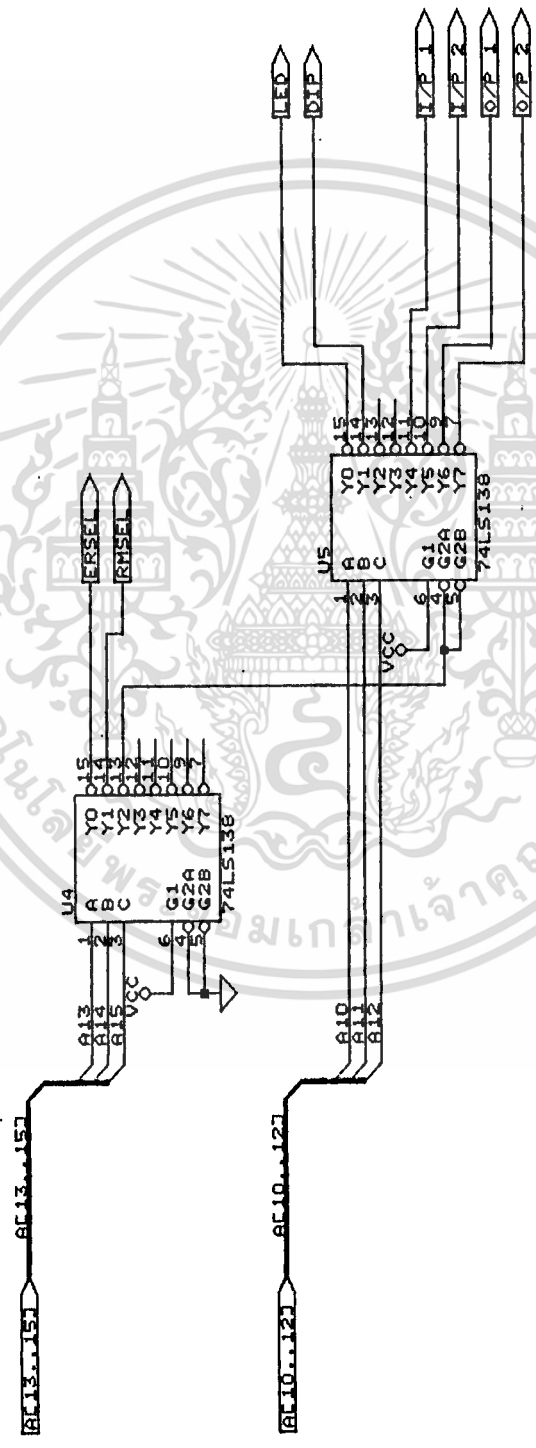
แผนภูมิของการทำงานของ FRONT END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



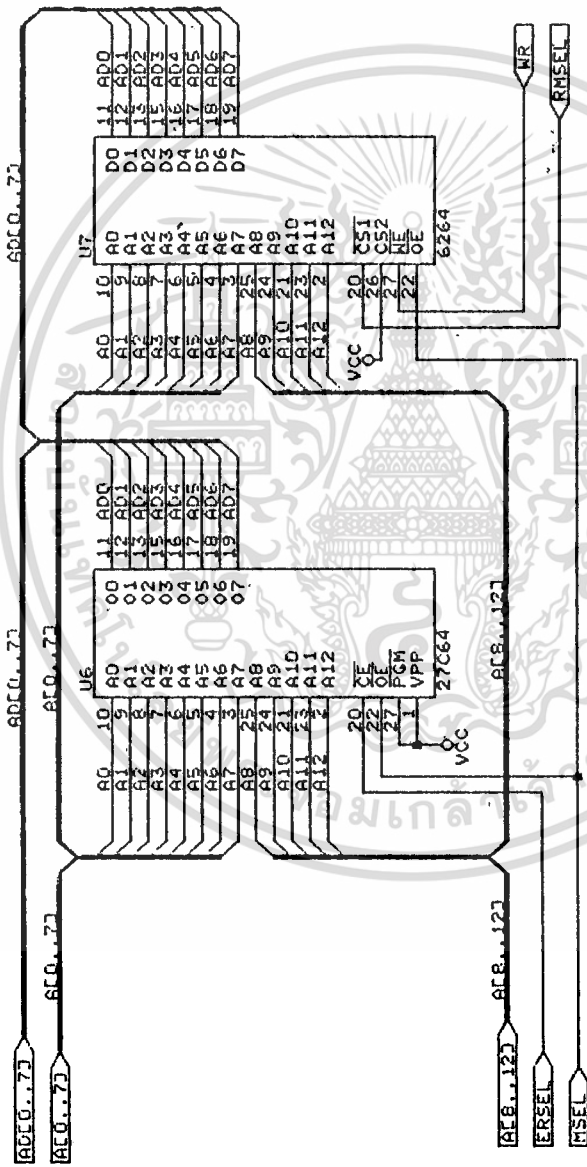
FRONT-END DATA ACQUISITION		
Title	MAIN CPU	
Size	Document Number	1
REV	A	A
Date:	March 21, 1991 Sheet 1 of 7	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



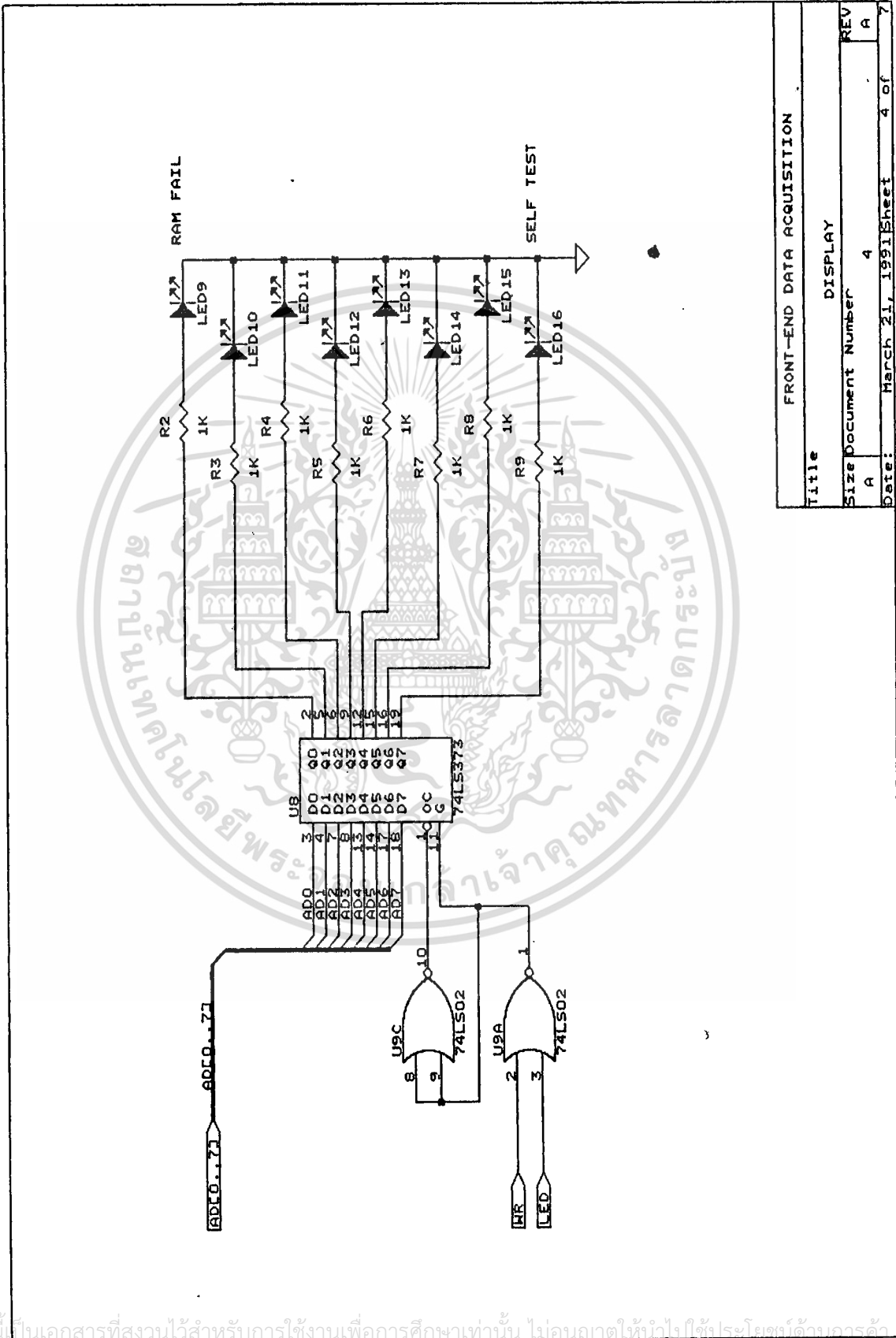
Title		FRONT-END DATA ACQUISITION
Size		A
Document Number		2
Date:		March 21, 1991
REV	Sheet	2 of 7
A		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		FRONT-END DATA ACQUISITION	
Size		MEMORY	
Document Number	3	REV	A
Date:	October 24, 1990	Sheet	3 of 5

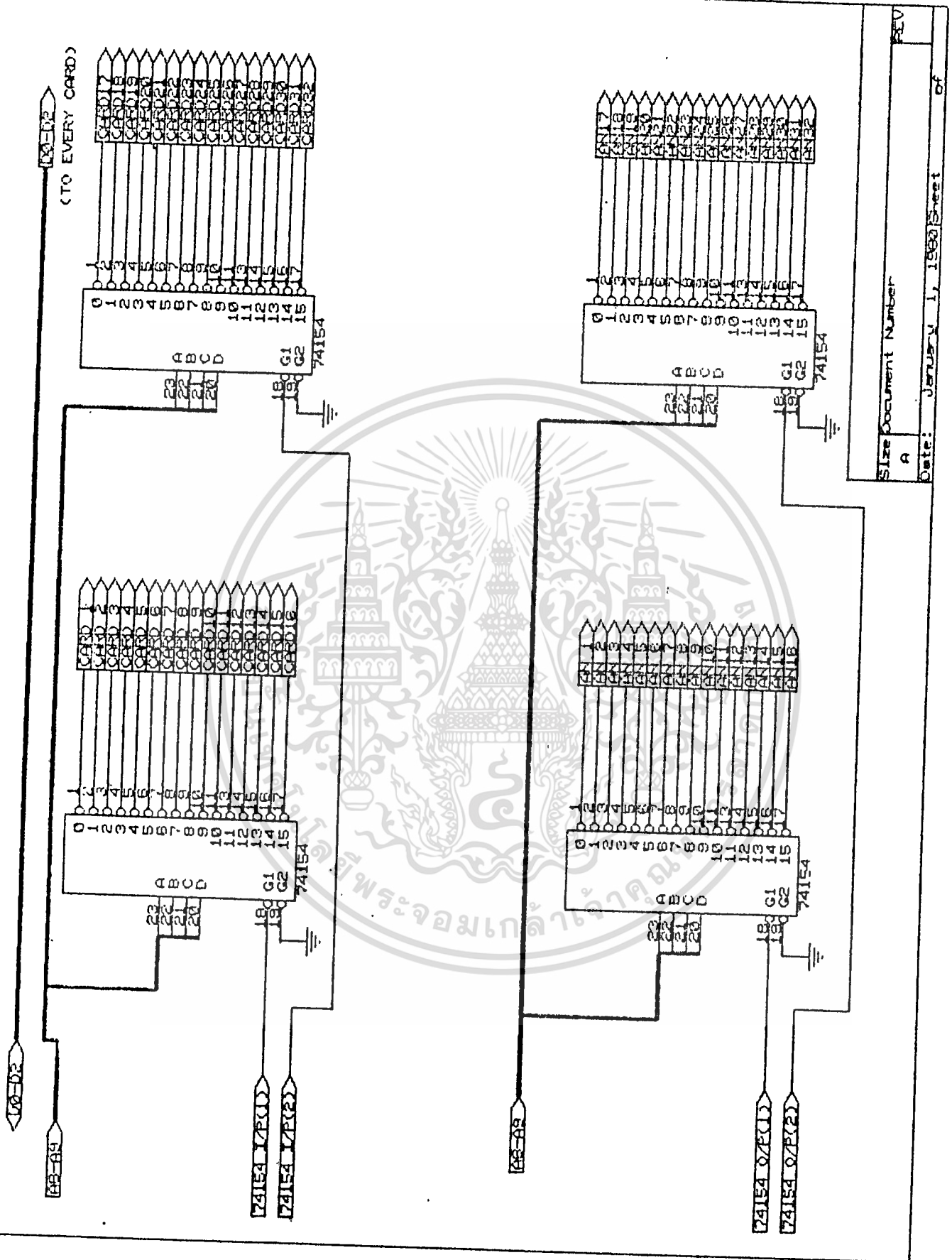
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		FRONT-END DATA ACQUISITION	
Size		A	
Document Number		4	
REV		A	
Date:	March 21, 1991	Sheet	4 of 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์อื่นใด  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



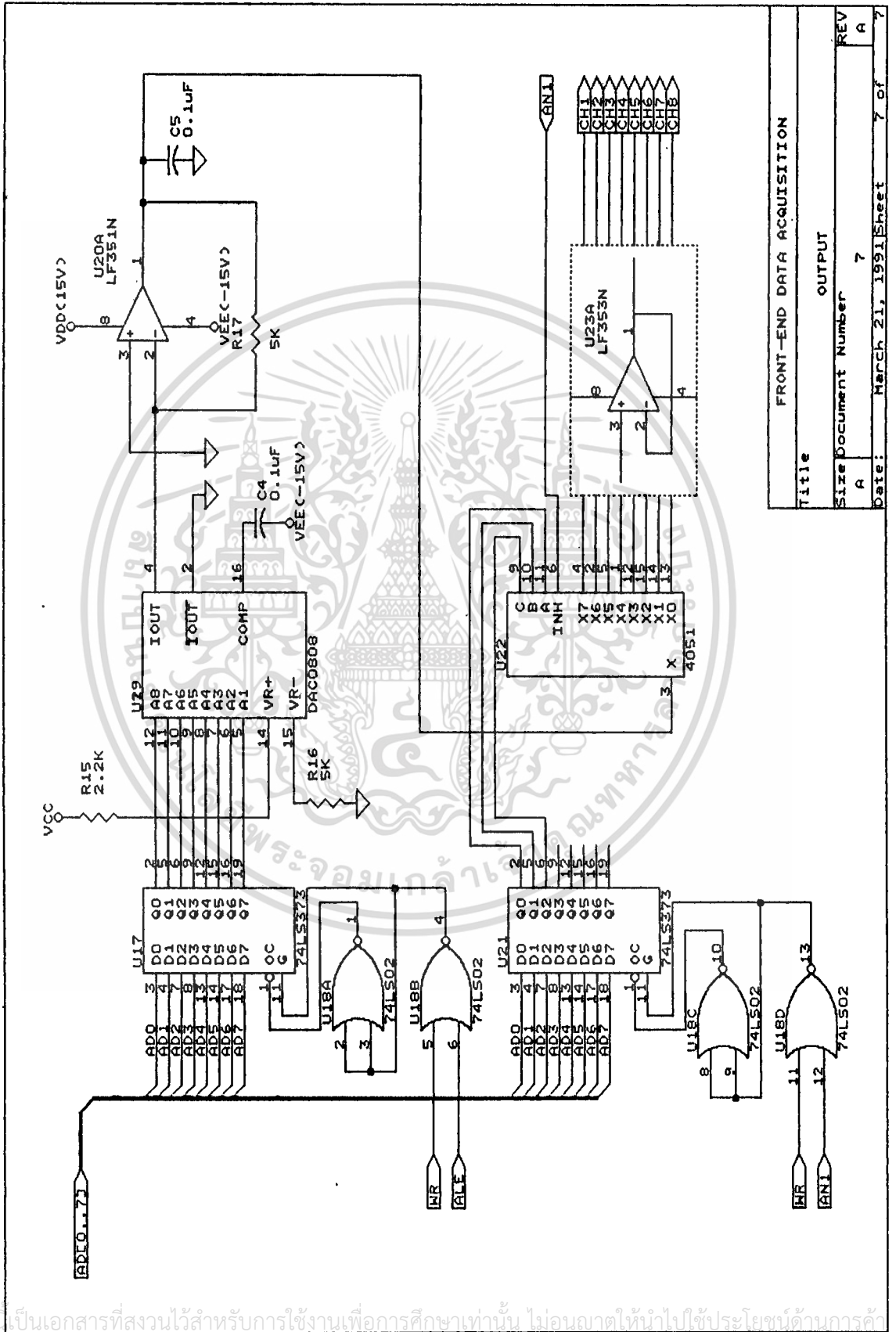


Size Document Number  
A

Date: January 1, 1980 15-est of

REV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		FRONT-END DATA ACQUISITION
Size Document Number		OUTPUT A 7
Date:		March 21, 1991 Sheet 7 of 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*-----*
* MONITOR PROGRAM OF MCS-8031 *
* FRONT END DATA ACQUISITION *
*-----*

```

```

0000 CPU "8051.TBL"
0000 HOF "INT8"

```

```

;-----;
; MCS-51 INTERNAL REGISTERS ;
;-----;

```

```

00F0 = B: EQU 0F0H
00E0 = A: EQU 0E0H
00D0 = PSW: EQU 0D0H
00C8 = IP: EQU 0B8H
00C0 = P3: EQU 0B0H
00A8 = IE: EQU 0A8H
00A0 = P2: EQU 0A0H
0099 = SBUF: EQU 99H
0098 = SCON: EQU 98H
0090 = P1: EQU 90H
008D = TH1: EQU 8DH
008C = TH0: EQU 8CH
0089 = TMOD: EQU 89H
0088 = TCON: EQU 88H
0087 = PCON: EQU 87H
0083 = DPH: EQU 83H
0082 = DPL: EQU 82H
0081 = SP: EQU 81H
0080 = P0: EQU 80H

```

```

;-----;
; MCS-51 INTERNAL BIT ADDRESS ;
;-----;

```

```

00DF = C: EQU 0DFH
00D6 = AC: EQU 0D6H
0097 = TI: EQU 97H
0098 = RI: EQU 98H
00E0 = A0: EQU 0E0H
00E1 = A1: EQU 0E1H
00E2 = A2: EQU 0E2H
00E3 = A3: EQU 0E3H
00E4 = A4: EQU 0E4H
00E5 = A5: EQU 0E5H
00E6 = A6: EQU 0E6H
00E7 = A7: EQU 0E7H
0090 = P1_0: EQU 90H
0091 = P1_1: EQU 91H
0092 = P1_2: EQU 92H
0093 = P1_3: EQU 93H
0094 = P1_4: EQU 94H
0095 = P1_5: EQU 95H
0096 = P1_6: EQU 96H
0097 = P1_7: EQU 97H

```

```

*-----*
; INITIAL PROGRAM
*-----*

```

```

0000 ORG 0000H
0000 7800 CLEAR_IN_RAM: MOV R0,#00H
0002 C3 CLEAR_IN: CLR C
0003 7600 MOV @R0,#00H
0005 08 INC R0
0006 B880F9 CJNE R0,#80H,CLEAR_IN
0009 C3 CLR C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายใน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่อนุญาตให้นำไปตีพิมพ์ซ้ำ และถ้าหากมีให้ตัดแปลงเนื้อหา และตีพิมพ์ซ้ำโดยไม่ขออนุญาตทุกครั้งที่มีการนำไปใช้

```

000A 7CFF          CLEAR_EX_RAM: MOV  R4,#0FFH
000C 903000      MOV  DPTR,#3000H
000F 7400          CLEAR_EX:  MOV  A,#00H
0011 F0          MOVX @DPTR,A
0012 A3          INC  DPTR
0013 DCFA        DJNZ R4,CLEAR_EX
0015 00          NOP

```

```

0016 759850      INIT_SERIAL: MOV  SCON,#50H
0019 758700      MOV  PCON,#00H
001C 758920      MOV  TMOD,#20H
001F 758848      MOV  TCON,#48H
0022 75A894      MOV  IE,#94H
0025 75B810      MOV  IP,#10H
0028 758107      MOV  SP,#07H
002B 904400      MOV  DPTR,#4400H
002E E0          MOVX A,@DPTR

```

; BAUD RATE ADD

```

;
; CHECK BAUD RATE SELECT
;

```

```

002F 20E01A      JB   A0,BAUD_1200
0032 20E112      JB   A1,BAUD_2400
0035 20E20A      JB   A2,BAUD_4800
0038 20E302      JB   A3,BAUD_9600
003B 800F      SJMP BAUD_1200

```

```

003D 758DFD      BAUD_9600: MOV  TH1,#0FDH
0040 800D      SJMP START
0042 758DFA      BAUD_4800: MOV  TH1,#0FAH
0045 8008      SJMP START
0047 758DF4      BAUD_2400: MOV  TH1,#0FAH
004A 8003      SJMP START
004C 758DEF      BAUD_1200: MOV  TH1,#0E8H

```

```

; *-----*
; * INITIAL CHECK & SEND ACKNOWLEDGE *
; * AND RECEIVE CARD SELECT *
; *-----*

```

```

004F 7F04          START:  MOV  R7,#04H
0051 7858          MOV  R0,#58H
0053 120071        LCALL CHECK0
0056 120082        LCALL SEND_POINT
0059 209802        CHECK2: JB   RI,CHECK3
005C 80FB          SJMP CHECK2
005E C298          CHECK3: CLR  RI
0060 E599          MOV  A,SBUF
0062 F6          MOV  @R0,A
0063 C299          CLR  SBUF
0065 7426          MOV  A,#26H
0067 C299          CLR  TI
0069 F599          MOV  SBUF,A
006B 08          INC  R0
006C DFEB          DJNZ R7,CHECK2
006E 020089        LJMP RECEIVE_CH

```

; CHECK '?'

```

0071 209802        CHECK0: JB   RI,CHECK1
0074 80FB          SJMP CHECK0
0076 C3          CHECK1: CLR  C
0077 C298          CLR  RI
0079 E599          MOV  A,SBUF
007B B43FF8        CJNE A,#3FH,CHECK1

```

; CHECK '?' ROU

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการวิจัยและพัฒนาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า

โปรดแจ้งสินค้า อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และแจ้งเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

007E C3          CLR C
007F C299        CLR SBUF
0081 22          RET

0082 742E        SEND_POINT: MOV A,#2EH
0084 C299        CLR TI
0086 F599        MOV SBUF,A
0088 22          RET

```

```

;-----*
;* RECEIVE CHANNAL SELECT FROM HOST COMPUTER *
;-----*

```

```

0089 7860        RECEIVE_CH: MOV R0,#60H
008B 7958        MOV R1,#58H
008D 120071      LCALL CHECK0           ; CHECK ' ?
0090 C3          NEXT_CARD: CLR C
0091 E7          MOV A,@R1
0092 7A08        MOV R2,#08H
0094 C3          ROTATE_CARD: CLR C
0095 13          RRC A
0096 FB          MOV R3,A
0097 4012        JC CH_SELECT
0099 C3          ROTATE_AGAIN: CLR C
009A 08          INC R0
009B DAF7        DJNZ R2,ROTATE_CARD
009D 09          INC R1
009E B95CEF      CJNE R1,#5CH,NEXT_CARD
00A1 C3          CLR C
00A2 742E        MOV A,#2EH
00A4 C299        CLR TI
00A6 F599        MOV SBUF,A
00A8 0200CA      LJMP RECEIVE_CONST

00AB C3          CH_SELECT: CLR C
00AC 120082      LCALL SEND_POINT
00AF 209802      RECEIVE_CH0: JB RI,RECEIVE_CH1
00B2 80FB        SJMP RECEIVE_CH0
00B4 C298        RECEIVE_CH1: CLR RI
00B6 E599        MOV A,SBUF
00B8 F6          MOV @R0,A
00B9 C299        CLR SBUF
00BE 7426        MOV A,#26H
00BD C299        CLR TI
00BF F599        MOV SBUF,A
00C1 00          NOP
00C2 120071      LCALL CHECK0           ; CHECK ' ?
00C5 EB          MOV A,R3
00C6 C3          CLR C
00C7 020099      LJMP ROTATE_AGAIN

```

```

;-----*
;* RECEIVE MAX, MIN, SET_POINT, Kp, Ti & Td *
;-----*

```

```

00CA 903000      RECEIVE_CONST: MOV DPTR,#3000H
00CD 1200D9      LCALL RECEIVE
00D0 903010      MOV DPTR,#3010H
00D3 1200D9      LCALL RECEIVE
00D6 02010E      LJMP CHECK_START

00D9 120071      RECEIVE: LCALL CHECK0
00DC 7F03        RECEIVE_AGAIN: MOV R7,#03H

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามมิให้คัดลอกหรือเผยแพร่โดยไม่ได้รับอนุญาต

```

00DE 120082          LCALL SEND_POINT
00E1 1200F8          LCALL RECEIVE_CONST1
00E4 209802          CHECK_END0: JB RI,CHECK_END1
00E7 80FB            SJMP CHECK_END0
00E9 C298            CHECK_END1: CLR RI
00EB E599            MOV A,SBUF
00ED B440EC          CJNE A,#40H,RECEIVE_AGAIN
00F0 C3              CLR C
00F1 7423            MOV A,#23H
00F3 C299            CLR TI
00F5 F599            MOV SBUF,A
00F7 22              RET

```

```

00F8 209802          RECEIVE_CONST1: JB RI,RECEIVE0
00FB 80FB            SJMP RECEIVE_CONST1
00FD C298            RECEIVE0: CLR RI
00FF E599            MOV A,SBUF
0101 F0              MOVX @DPTR,A
0102 C299            CLR SBUF
0104 A3              INC DPTR
0105 7426            MOV A,#26H
0107 C299            CLR TI
0109 F599            MOV SBUF,A
010B DFEB            DJNZ R7,RECEIVE_CONST1
010D 22              RET

```

```

;-----*
; * CHECK ACKNOWLEDGE FOR START FRONT END *
; * TO RECEIVE DATA FROM INPUT *
;-----*

```

```

010E 120120          CHECK_START: LCALL SEND_SHARP
0111 209802          CHECK_START1: JB RI,PROMPT_RUN
0114 80FB            SJMP CHECK_START1
0116 C299            PROMPT_RUN: CLR TI
0118 E599            MOV A,SBUF
011A B425F1          CJNE A,#25H,CHECK_START
011D 020127          LJMP RECEIVE_DATA

```

```

0120 7423            SEND_SHARP: MOV A,#23H
0122 C299            CLR TI
0124 F599            MOV SBUF,A
0126 22              RET

```

```

;-----*
; * ROTATE CHANNELS FOR RECEIVE INPUT DATA *
;-----*

```

```

0127 7C10            RECEIVE_DATA: MOV R4,#10H
0129 903020          MOV DPTR,#3020H
012C 758147          MOV SP,#47H ; STACK FOR DATA
012F C082            PUSH DPL
0131 C083            PUSH DPH
0133 7860            KEEP_DATA_AGAIN: MOV R0,#60H
0135 7E20            MOV R6,#20H
0137 7F04            MOV R7,#04H
0139 75814F          MOV SP,#4FH ; STACK FOR CARD
013C 905000          MOV DPTR,#5000H
013F C082            PUSH DPL
0141 C083            PUSH DPH
0143 B6001F          ROTATE_CHANNEL: CJNE @R0,#00H,ROTATE_8_BITS
0146 758151          CONTINUE: MOV SP,#51H
0149 D083            POP DPH
014B D082            POP DPL

```

```

014D C3          CLR C
014E E582        MOV A,DPL
0150 2440        ADD A,#40H
0152 F582        MOV DPL,A
0154 DF03        DJNZ R7,ROTATE_NEXT_CARD
0156 020196      LJMP INCREASE_DPH
0159 C082        ROTATE_NEXT_CARD: PUSH DPL
015B C083        PUSH DPH
015D 08          INC R0
015E DEE3        DJNZ R6,ROTATE_CHANNA
0160 DCD1        DJNZ R4,KEEP_DATA_AGAIN
0162 02019D      LJMP KEEP_OUTPUT_CARD

```

```

0165 C3          ROTATE_8_BITS: CLR C
0166 7A08        MOV R2,#08H
0168 7DFF        MOV R5,#0FFH
016A E6          MOV A,@R0
016B 0D          ROTATE: INC R5
016C 13          RRC A
016D FB          MOV R3,A
016E 4005        JC POLLING
0170 DAF9        ROTATE_AGAIN1: DJNZ R2,ROTATE
0172 020146      LJMP CONTINUE

```

```

0175 C3          POLLING: CLR C
0176 ED          MOV A,R5
0177 758151      MOV SP,#51H
017A D083        POP DPH
017C D082        POP DPL
017E F0          MOVX @DPTR,A
017F E4          CLR A
0180 E0          MOVX A,@DPTR
0181 C082        PUSH DPL
0183 C083        PUSH DPH
0185 758149      MOV SP,#49H
0188 D083        POP DPH
018A D082        POP DPL
018C F0          MOVX @DPTR,A
018D A3          INC DPTR
018E C082        PUSH DPL
0190 C083        PUSH DPH
0192 EB          MOV A,R3
0193 020170      LJMP ROTATE_AGAIN1

```

```

0196 0583        INCREASE_DPH: INC DPH
0198 7F04        MOV R7,#04H
019A 020159      LJMP ROTATE_NEXT_CARD

```

```

; *-----*
; * CHECK CARD & CHANNA OUTPUT FOR CONTROL *
; *-----*

```

```

019D 787F        KEEP_OUTPUT_CARD: MOV R0,#7FH
019F 7E20        MOV R6,#20H
01A1 7F04        MOV R7,#04H
01A3 758117      MOV SP,#17H
01A6 905FC0      MOV DPTR,#5FC0H

```

; STACK FOR OUT

```

01A9 C3          NEXT: CLR C
01AA B60014      CJNE @R0,#00H,KEEP_DPTR
01AD E582        NEXT_OUTPUT: MOV A,DPL
01AF 943F        SUBB A,#3FH
01B1 C3          CLR C
01B2 C2D6        CLR AC

```

เอกสารนี้เป็นทรัพย์สินของสำนักงานคณะกรรมการอาหารและยา กระทรวงสาธารณสุข  
 เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าวิจัยและพัฒนาเท่านั้น  
 ไม่สามารถทำซ้ำหรือเผยแพร่โดยไม่ได้รับอนุญาตจากของเอกสารทุกครั้งที่มีการนำไปใช้

```

01B4 F582          MOV   DPL,A
01B6 DF03          DJNZ  R7,AGAIN
01B8 02021C       LJMP  NEXT_OUTPUT_CARD
01BB 18           AGAIN: DEC  R0
01BC DEEB          DJNZ  R6,NEXT
01BE 020223       LJMP  COMPARE_MAX_MIN

```

```

01C1 C3           KEEP_DPTR: CLR  C
01C2 C2D6          CLR  AC
01C4 C082          PUSH DPL
01C6 C083          PUSH DPH
01C8 E6           MOV  A,@R0
01C9 20E018       CHECK_AGAIN1: JB  A0,CH_1
01CC 20E11C       CHECK_AGAIN2: JB  A1,CH_2
01CF 20E220       CHECK_AGAIN3: JB  A2,CH_3
01D2 20E324       CHECK_AGAIN4: JB  A3,CH_4
01D5 20E428       CHECK_AGAIN5: JB  A4,CH_5
01D8 20E52C       CHECK_AGAIN6: JB  A5,CH_6
01DB 20E630       CHECK_AGAIN7: JB  A6,CH_7
01DE 20E734       CHECK_AGAIN8: JB  A7,CH_8
01E1 0201AD       END_CHECK_AGAIN: LJMP NEXT_OUTPUT

```

```

01E4 7D00         CH_1: MOV  R5,#00H
01E6 C005         PUSH 05H
01E8 0201CC       LJMP  CHECK_AGAIN2

```

```

01EB 7D01         CH_2: MOV  R5,#01H
01ED C005         PUSH 05H
01EF 0201CF       LJMP  CHECK_AGAIN3

```

```

01F2 7D02         CH_3: MOV  R5,#02H
01F4 C005         PUSH 05H
01F6 0201D2       LJMP  CHECK_AGAIN4

```

```

01F9 7D03         CH_4: MOV  R5,#03H
01FB C005         PUSH 05H
01FD 0201D5       LJMP  CHECK_AGAIN5

```

```

0200 7D04         CH_5: MOV  R5,#04H
0202 C005         PUSH 05H
0204 0201D8       LJMP  CHECK_AGAIN6

```

```

0207 7D05         CH_6: MOV  R5,#05H
0209 C005         PUSH 05H
020B 0201DB       LJMP  CHECK_AGAIN7

```

```

020E 7D06         CH_7: MOV  R5,#06H
0210 C005         PUSH 05H
0212 0201DE       LJMP  CHECK_AGAIN8

```

```

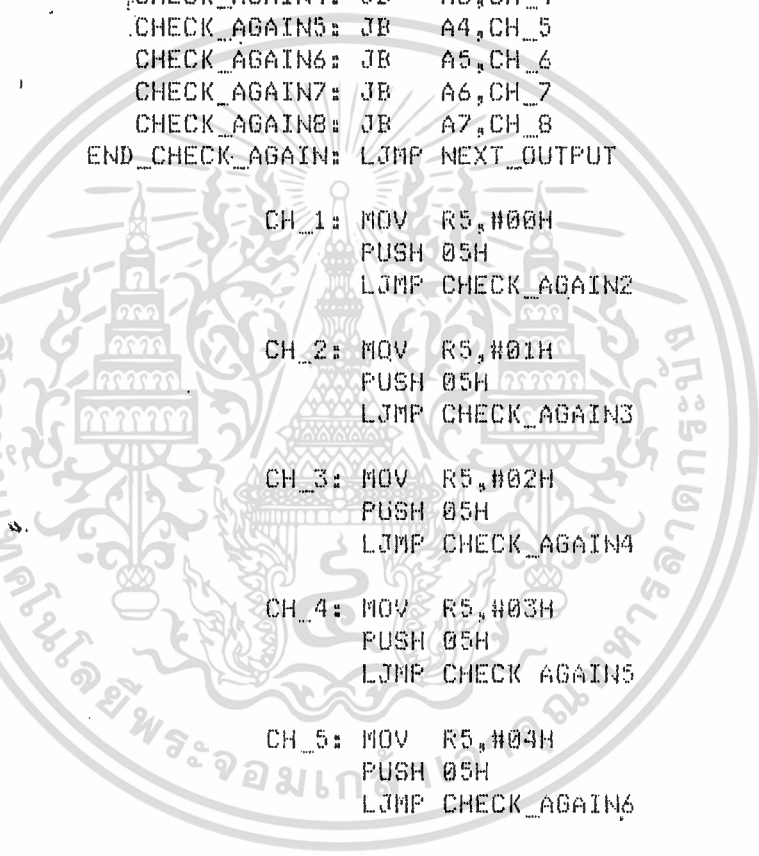
0215 7D07         CH_8: MOV  R5,#07H
0217 C005         PUSH 05H
0219 0201E1       LJMP  END_CHECK_AGAIN

```

```

021C 1583         NEXT_OUTPUT_CARD: DEC  DPH
021E 7F04         MOV  R7,#04H
0220 0201BB       LJMP  AGAIN

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น \* \* \* \* \* COMPARE BETWEEN DATA & MAX, MIN VALUE \* \* \* \* \*

```

0223 758127       COMPARE_MAX_MIN: MOV  SP,#27H ; STACK FOR KE

```

```

0226 903070          MOV  DPTR,#3070H
0229 C082           PUSH DPL
022B C083           PUSH DPH
022D 75813F         MOV  SP,#3FH          ; STACK FOR PID
0230 903010         MOV  DPTR,#3010H
0233 C082           PUSH DPL
0235 C083           PUSH DPH
0237 758145         MOV  SP,#45H          ; STACK FOR SET
023A 903002         MOV  DPTR,#3002H
023D C082           PUSH DPL
023F C083           PUSH DPH
0241 75814F         MOV  SP,#4FH          ; STACK FOR COM
0244 903000         MOV  DPTR,#3000H
0247 C082           PUSH DPL
0249 C083           PUSH DPH
024B 758147         MOV  SP,#47H          ; STACK FOR DAT
024E 903020         MOV  DPTR,#3020H
0251 C082           PUSH DPL
0253 C083           PUSH DPH
0255 7D01           MOV  R5,#01H
0257 75100F         MOV  10H,#0FH        ; USE IN SEND R
025A 7D10           COMPARE_NEXT_CH: MOV R3,#10H          ; CHECK 16 DATA
025C 7804           MOV  R0,#04H          ; ACCEPT ERROR
025E 7C04           MOV  R4,#04H          ; ACCEPT ERROR
0260 02028C         COMPARE_NEXT_DATA: LJMP COMPARE_MAX
0263 4072           CHECK_MAX_ERROR:  JC  OVER_MAX
0265 1202B0         NEXT_COMPARE:    LCALL COMPARE_MIN
0268 5072           CHECK_MIN_ERROR: JNC LOWER_MIN
026A 0203DD         PROCESS_CONTROL: LJMP PID_PROCESS
026D 0202E1         LOAD_NEXT_DATA:  LJMP INCREASE_CHANNEL
0270 DBEE           CHECK_NEXT_DATA: DJNZ R3,COMPARE_NEXT_DATA
0272 0D           INC  R5
0273 020383         LJMP CHECK_DATA
0276 758151         RETURN_CHECK:    MOV  SP,#51H          ; STACK FOR COM
0279 D083           POP  DPH
027B D082           POP  DPL
027D A3           INC  DPTR
027E A3           INC  DPTR
027F A3           INC  DPTR
0280 E582           MOV  A,DPL
0282 C082           PUSH DPL
0284 C083           PUSH DPH
0286 F506D1         CJNE A,#06H,COMPARE_NEXT_CH
0289 020974         LJMP SEND_DATA

```

```

028C 758151         COMPARE_MAX:    MOV  SP,#51H          ; STACK FOR COM
028F D083           POP  DPH
0291 D082           POP  DPL
0293 E0           MOVX A,@DPTR
0294 F507           MOV  07H,A           ; R7 <-- MAX
0296 C082           PUSH DPL
0298 C083           PUSH DPH
029A 758149         MOV  SP,#49H          ; STACK FOR DATA
029D D083           POP  DPH
029F D082           POP  DPL
02A1 E0           MOVX A,@DPTR
02A2 C082           PUSH DPL
02A4 C083           PUSH DPH
02A6 F5F0           MOV  B,A
02A8 E507           MOV  A,#07H          ; A <-- MAX
02AA C3           CLR  C
02AB 95F0           SUBB A,B

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เอกสารนี้สงวนลิขสิทธิ์ หากมีให้ดัดแปลงเนื้อหา หรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการแก้ไข

```

02A0 020263                LJMP CHECK_MAX_ERROR

02B0 758151                COMPARE_MIN: MOV SP,#51H                ; STACK FOR COM
02B3 D083                  POP DPH
02B5 D082                  POP DPL
02B7 A3                    INC DPTR
02B8 E0                    MOVX A,@DPTR
02B9 1582                  DEC DPL
02BB C082                  PUSH DPL
02BD C083                  PUSH DPH
02BF F507                  MOV 07H,A
02C1 758149                ; STACK FOR DAT
02C4 D083                  POP DPH
02C6 D082                  POP DPL
02C8 E0                    MOVX A,@DPTR
02C9 C082                  PUSH DPL
02CB C083                  PUSH DPH
02CD F5F0                  MOV B,A
02CF E507                  MOV A,07H
02D1 C3                    CLR C
02D2 95F0                  SUBB A,B
02D4 020268                LJMP CHECK_MIN_ERROR

02D7 D88C                  OVER_MAX: DJNZ R0,NEXT_COMPARE
02D9 020347                LJMP MAX_ERROR

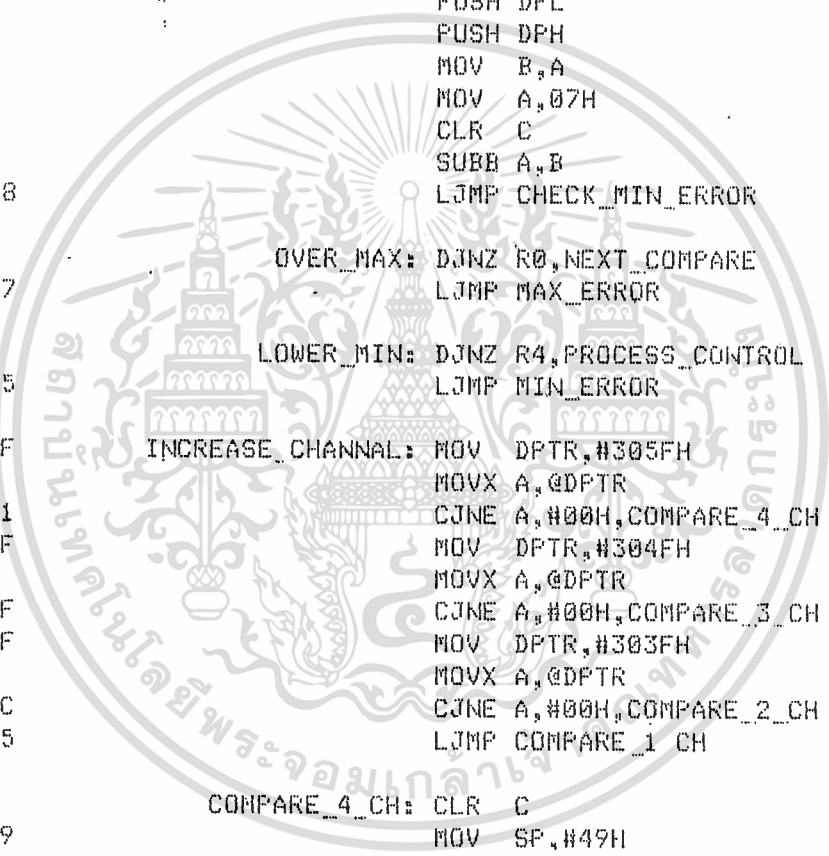
02DC DC8C                  LOWER_MIN: DJNZ R4,PROCESS_CONTROL
02DE 020365                LJMP MIN_ERROR

02E1 90305F                INCREASE_CHANNEL: MOV DPTR,#305FH
02E4 E0                    MOVX A,@DPTR
02E5 B40011                CJNE A,#00H,COMPARE_4_CH
02E8 90304F                MOV DPTR,#304FH
02EB E0                    MOVX A,@DPTR
02EC B4001F                CJNE A,#00H,COMPARE_3_CH
02EF 90303F                MOV DPTR,#303FH
02F2 E0                    MOVX A,@DPTR
02F3 B4002C                CJNE A,#00H,COMPARE_2_CH
02F6 020335                LJMP COMPARE_1_CH

02F9 C3                    COMPARE_4_CH: CLR C
02FA 758149                MOV SP,#49H                ; STACK FOR DAT
02FD D083                  POP DPH
02FF D082                  POP DPL
0301 A3                    INC DPTR
0302 A3                    INC DPTR
0303 A3                    INC DPTR
0304 A3                    INC DPTR
0305 C082                  PUSH DPL
0307 C083                  PUSH DPH
0309 7E0C                  MOV R6,#0CH
030B 020270                LJMP CHECK_NEXT_DATA

030E C3                    COMPARE_3_CH: CLR C
030F 758149                MOV SP,#49H                ; STACK FOR DAT
0312 D083                  POP DPH
0314 D082                  POP DPL
0316 A3                    INC DPTR
0317 A3                    INC DPTR
0318 A3                    INC DPTR
0319 C082                  PUSH DPL
031B C083                  PUSH DPH

```



เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานอกระบบของมหาวิทยาลัยราชภัฏบึงกาฬ  
 ใบอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ใบอนุญาตให้คัดลอกและเผยแพร่ได้โดยไม่เสียค่าใช้จ่ายของเอกสารทุกครั้งที่มีการนำไปใช้

031D 7E09 MOV R6,#09H  
031F 020270 LJMP CHECK\_NEXT\_DATA

0322 C3 COMPARE\_2\_CH: CLR C  
0323 758149 MOV SP,#49H ; STACK FOR D  
0326 D083 POP DPH  
0328 D082 POP DPL  
032A A3 INC DPTR  
032B A3 INC DPTR  
032C C082 PUSH DPL  
032E C083 PUSH DPH  
0330 7E06 MOV R6,#06H  
0332 020270 LJMP CHECK\_NEXT\_DATA

0335 C3 COMPARE\_1\_CH: CLR C  
0336 758149 MOV SP,#49H ; STACK FOR D  
0339 D083 POP DPH  
033B D082 POP DPL  
033D A3 INC DPTR  
033E C082 PUSH DPL  
0340 C083 PUSH DPH  
0342 7E03 MOV R6,#03H  
0344 020270 LJMP CHECK\_NEXT\_DATA

0347 00 MAX\_ERROR: NOP  
0348 BD0105 CH1\_MAX\_ERROR: CJNE R5,#01H,CH2\_MAX\_ERROR  
034B B290 CPL P1\_0  
034D 020265 LJMP NEXT\_COMPARE  
0350 BD0205 CH2\_MAX\_ERROR: CJNE R5,#02H,CH3\_MAX\_ERROR  
0353 B291 CPL P1\_1  
0355 020265 LJMP NEXT\_COMPARE  
0358 BD0305 CH3\_MAX\_ERROR: CJNE R5,#03H,CH4\_MAX\_ERROR  
035B B292 CPL P1\_2  
035D 020265 LJMP NEXT\_COMPARE  
0360 B293 CH4\_MAX\_ERROR: CPL P1\_3  
0362 020265 LJMP NEXT\_COMPARE

0365 00 MIN\_ERROR: NOP  
0366 BD0105 CH1\_MIN\_ERROR: CJNE R5,#01H,CH2\_MIN\_ERROR  
0369 B294 CPL P1\_4  
036B 02026A LJMP PROCESS\_CONTROL  
036E BD0205 CH2\_MIN\_ERROR: CJNE R5,#02H,CH3\_MIN\_ERROR  
0371 B295 CPL P1\_5  
0373 02026A LJMP PROCESS\_CONTROL  
0376 BD0305 CH3\_MIN\_ERROR: CJNE R5,#03H,CH4\_MIN\_ERROR  
0379 B296 CPL P1\_6  
037B 02026A LJMP PROCESS\_CONTROL  
037E B297 CH4\_MIN\_ERROR: CPL P1\_7  
0380 02026A LJMP PROCESS\_CONTROL

0383 758149 CHECK\_DATA: MOV SP,#49H  
0386 D083 POP DPH  
0388 D082 POP DPL  
038A E582 MOV A,DPL  
038C B4600A CJNE A,#60H,L1  
038F 903021 MOV DPTR,#3021H ; CH\_2

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา  
ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ออกให้ตัดแปลงเนื้อหา และ  
L1: CJNE A,#61H,L2 ; CH\_3

0392 C082 PUSH DPL  
0394 C083 PUSH DPH  
0396 020276 LJMP RETURN CHECK ทุกครั้งที่มีการนำไปใช้  
0399 B4610A L1: CJNE A,#61H,L2  
039C 903022 MOV DPTR,#3022H ; CH\_3  
039E C082 PUSH DPL

```

03A1 C083          PUSH DPH
03A3 020276       LJMP RETURN_CHECK
03A6 B4620A       L2: CJNE A,#62H,L3
03A9 903023       MOV DPTR,#3023H ; CH_4
03AC C082          PUSH DPL
03AE C083          PUSH DPH
03B0 020276       LJMP RETURN_CHECK
03B3 B4500A       L3: CJNE A,#50H,L4
03B6 903021       MOV DPTR,#3021H ; CH_2
03B9 C082          PUSH DPL
03BB C083          PUSH DPH
03BD 020276       LJMP RETURN_CHECK
03C0 B4510A       L4: CJNE A,#51H,L5
03C3 903022       MOV DPTR,#3022H ; CH_3
03C6 C082          PUSH DPL
03C8 C083          PUSH DPH
03CA 020276       LJMP RETURN_CHECK
03CD B4400A       L5: CJNE A,#40H,L6
03D0 903021       MOV DPTR,#3021H ; CH_2
03D3 C082          PUSH DPL
03D5 C083          PUSH DPH
03D7 020276       LJMP RETURN_CHECK
03DA 020276       L6: LJMP RETURN_CHECK

```

```

; *-----*
; PID PROCESS CONTROL
; *-----*

```

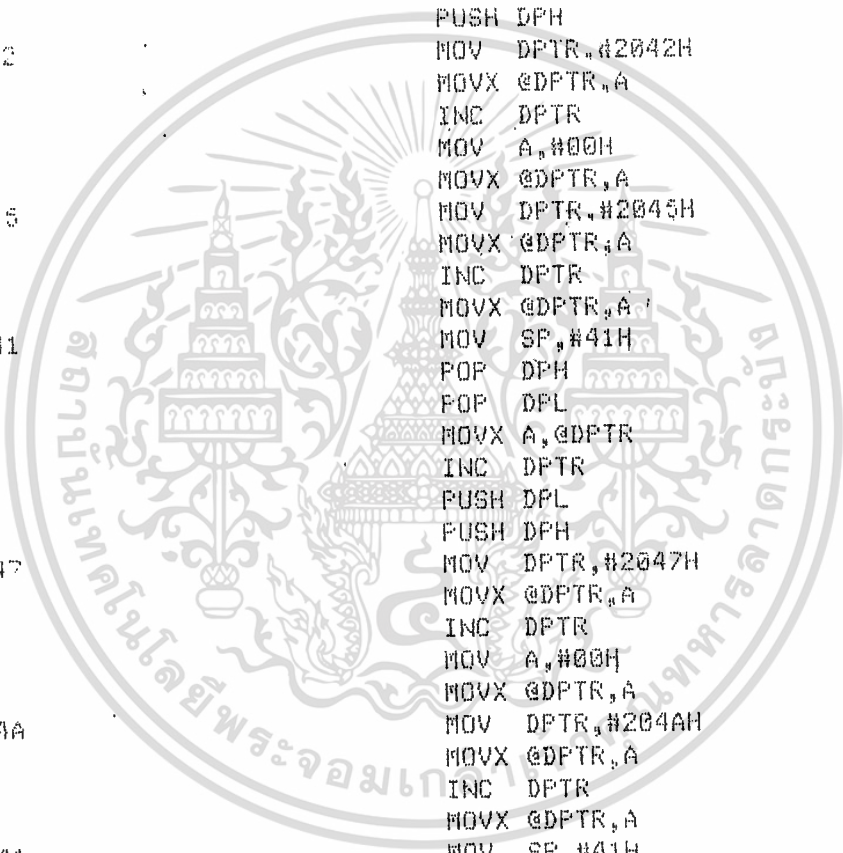
```

03DD 903100       PID_PROCESS: MOV DPTR,#3100H ; KEEP R0-R7 T
03E0 E500          MOV A,#00H
03E2 F0           MOVX @DPTR,A
03E3 A3           INC DPTR
03E4 E501          MOV A,#01H
03E6 F0           MOVX @DPTR,A
03E7 A3           INC DPTR
03E8 E502          MOV A,#02H
03EA F0           MOVX @DPTR,A
03EB A3           INC DPTR
03EC E503          MOV A,#03H
03EE F0           MOVX @DPTR,A
03EF A3           INC DPTR
03F0 E504          MOV A,#04H
03F2 F0           MOVX @DPTR,A
03F3 A3           INC DPTR
03F4 E505          MOV A,#05H
03F6 F0           MOVX @DPTR,A
03F7 A3           INC DPTR
03F8 E506          MOV A,#06H
03FA F0           MOVX @DPTR,A
03FB A3           INC DPTR
03FC E507          MOV A,#07H
03FE F0           MOVX @DPTR,A
03FF 00           NOP
0400 902000       MOV DPTR,#2000H ; DATA @2000H
0403 7400          MOV A,#00H
0405 F0           MOVX @DPTR,A
0406 A3           INC DPTR
0407 F0           MOVX @DPTR,A
0408 750149       MOV SP,#49H ; STACK FOR DA
040B D083          POP DPH
040D D082          POP DPL ; DPL ของเอกสารทุกครั้งที่มีการนำไปใช้
040F E0           MOVX A,@DPTR
0410 C082          PUSH DPL

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้ถือว่าผิดกฎหมาย

0412 C083	PUSH DPH	
0414 902002	MOV DPTR,#2002H	
0417 F0	MOVX @DPTR,A	
0418 A3	INC DPTR	
0419 7400	MOV A,#00H	
041B F0	MOVX @DPTR,A	
041C 902040	MOV DPTR,#2040H	; Kp @2040H @20
041F F0	MOVX @DPTR,A	
0420 A3	INC DPTR	
0421 F0	MOVX @DPTR,A	
0422 758141	MOV SP,#41H	; STACK FOR p
0425 D082	POP DPH	
0427 D082	POP DPL	
0429 E0	MOVX A,@DPTR	; (A) <-- Kp
042B A3	INC DPTR	
042E 30	PUSH DPL	
0430 1000	PUSH DPH	
0432 902042	MOV DPTR,#2042H	
0435 F0	MOVX @DPTR,A	; @2042H <-- Kp
0438 A3	INC DPTR	
043A 7400	MOV A,#00H	
043C F0	MOVX @DPTR,A	
0437 902045	MOV DPTR,#2045H	; T1 @2045H-@20
043A F0	MOVX @DPTR,A	
043B A3	INC DPTR	
043C F0	MOVX @DPTR,A	
043D 758141	MOV SP,#41H	; STACK FOR T1
0440 D083	POP DPH	
0442 D082	POP DPL	
0444 E0	MOVX A,@DPTR	; (A) <-- T1
0445 A3	INC DPTR	
0446 C082	PUSH DPL	
0448 C083	PUSH DPH	
044A 902047	MOV DPTR,#2047H	
044D F0	MOVX @DPTR,A	; @2047H <-- T1
044E A3	INC DPTR	
044F 7400	MOV A,#00H	
0451 F0	MOVX @DPTR,A	
0452 90204A	MOV DPTR,#204AH	; Td @204AH-@20
0455 F0	MOVX @DPTR,A	
0456 A3	INC DPTR	
0457 F0	MOVX @DPTR,A	
0458 758141	MOV SP,#41H	; STACK FOR Td
045B D083	POP DPH	
045D D082	POP DPL	
045F E0	MOVX A,@DPTR	
0460 1582	DEC DPL	
0462 1582	DEC DPL	
0464 C082	PUSH DPL	
0466 C083	PUSH DPH	
0468 90204C	MOV DPTR,#204CH	
046B F0	MOVX @DPTR,A	
046C A3	INC DPTR	
046D 7400	MOV A,#00H	
046F F0	MOVX @DPTR,A	
0470 90205A	MOV DPTR,#205AH	; SETPOINT @20
0473 F0	MOVX @DPTR,A	
0474 A3	INC DPTR	
0475 F0	MOVX @DPTR,A	
0478 758147	MOV SP,#47H	; STACK FOR SET
0479 D083	POP DPH	
047B D082	POP DPL	

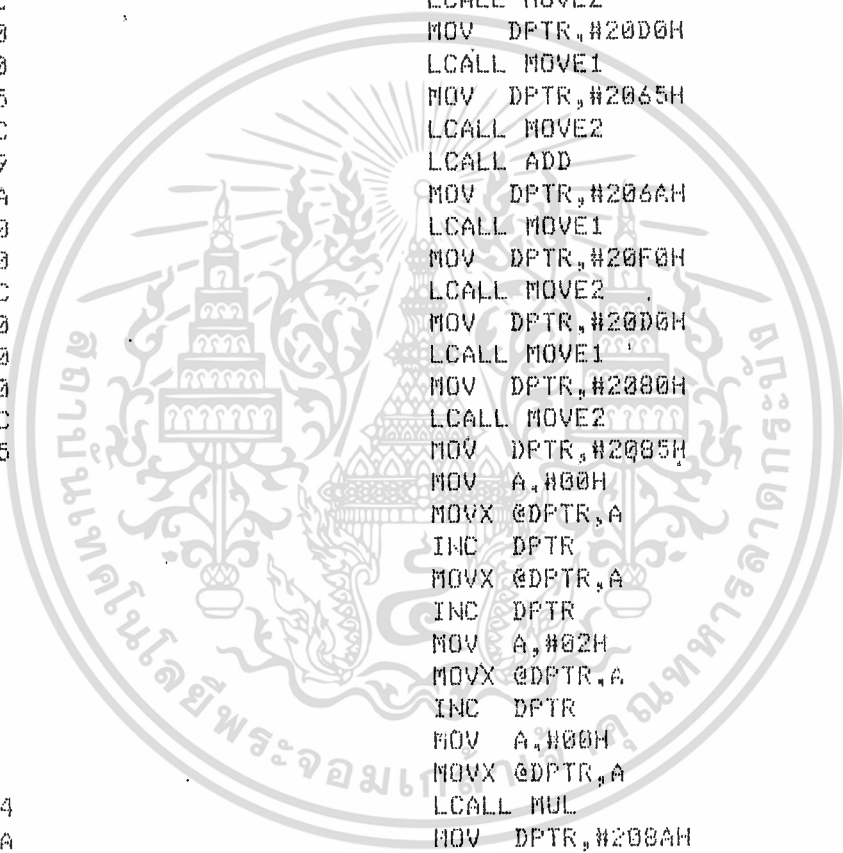


เอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ออกนอกลูกข่ายอื่น อีกทั้งห้ามมีให้ดัดแปลงเนื้อหา และสงวนลิขสิทธิ์ของเอกสารทุกครั้งที่มีการใช้

```

047D E0 MOVX A,@DPTR
047E C082 PUSH DPL
0480 C083 PUSH DFH
0482 90205C MOV DPTR,#205CH
0485 F0 MOVX @DPTR,A ; @205C <-- SETP
0486 A3 INC DPTR
0487 7400 MOV A,#00H
0489 F0 MOVX @DPTR,A
048A 75D008 F_PID: MOV FSW,#08H ; BANK1 FOR PID
048D 758130 MOV SF,#30H ; STACK FOR PID
0490 120678 LCALL FIA0 ; Kp(1+del T/Ti)
0493 120700 LCALL PD ; Kp(Td/del T)
0496 9020C5 MOV DPTR,#20C5H
0499 120660 LCALL MOVE1
049C 902060 MOV DPTR,#2060H
049F 12066C LCALL MOVE2
04A2 9020D0 MOV DPTR,#20D0H ; PD = A2
04A5 120660 LCALL MOVE1
04A8 902065 MOV DPTR,#2065H
04AB 12066C LCALL MOVE2
04AE 120809 LCALL ADD ; A0
04B1 90206A MOV DPTR,#206AH
04B4 120660 LCALL MOVE1
04B7 9020F0 MOV DPTR,#20F0H ; A0 --> @20F0H
04BA 12066C LCALL MOVE2
04BD 9020D0 MOV DPTR,#20D0H ; Kp(1+2Td/del T)
04C0 120660 LCALL MOVE1
04C3 902080 MOV DPTR,#2080H
04C6 12066C LCALL MOVE2
04C9 902085 MOV DPTR,#2085H
04CC 7400 MOV A,#00H
04CE F0 MOVX @DPTR,A
04CF A3 INC DPTR
04D0 F0 MOVX @DPTR,A
04D1 A3 INC DPTR
04D2 7402 MOV A,#02H
04D4 F0 MOVX @DPTR,A
04D5 A3 INC DPTR
04D6 7400 MOV A,#00H
04D8 F0 MOVX @DPTR,A
04D9 120754 LCALL MUL ; 2*PD
04DC 90208A MOV DPTR,#208AH
04DF 120660 LCALL MOVE1
04E2 902060 MOV DPTR,#2060H
04E5 12066C LCALL MOVE2
04E8 902040 MOV DPTR,#2040H
04EB 120660 LCALL MOVE1
04EE 902065 MOV DPTR,#2065H
04F1 12066C LCALL MOVE2
04F4 120809 LCALL ADD ; A1
04F7 90206A MOV DPTR,#206AH
04FA 120660 LCALL MOVE1
04FD 9020F5 MOV DPTR,#20F5H ; A1 --> @20F5H
0500 12066C LCALL MOVE2
0503 90205A PID: MOV DPTR,#205AH ; SP - INPUT = E
0506 120660 LCALL MOVE1
0509 902060 MOV DPTR,#2060H
050C 12066C LCALL MOVE2
050F 902000 MOV DPTR,#2000H
0512 120660 LCALL MOVE1
0515 902065 MOV DPTR,#2065H
0518 12066C LCALL MOVE2

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติให้นำไปใช้ประโยชน์ด้านการค้า  
 อื่นๆโดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัยราชภัฏบรียรัมย์ หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม  
 กรุณาติดต่อฝ่ายประชาสัมพันธ์ โทร. 043-821111 หรือ 043-821112 ในวันและเวลาราชการ

051B	120839	LCALL SUB	; En
051E	90206A	MOV DPTR,#206AH	; En * A0
0521	120660	LCALL MOVE1	
0524	902080	MOV DPTR,#2080H	
0527	12066C	LCALL MOVE2	
052A	9020F0	MOV DPTR,#20F0H	
052D	120660	LCALL MOVE1	
0530	902085	MOV DPTR,#2085H	
0533	12066C	LCALL MOVE2	
0536	120754	LCALL MUL	; K0
0539	90208A	MOV DPTR,#208AH	
053C	120660	LCALL MOVE1	
053F	902120	MOV DPTR,#2120H	; K0 --> @2120H
0542	12066C	LCALL MOVE2	
0545	902130	MOV DPTR,#2130H	; En-1 * A1
0548	120660	LCALL MOVE1	; En-1 --> @2131
054B	902080	MOV DPTR,#2080H	
054E	12066C	LCALL MOVE2	
0551	9020F5	MOV DPTR,#20F5H	
0554	120660	LCALL MOVE1	
0557	902085	MOV DPTR,#2085H	
055A	12066C	LCALL MOVE2	
055D	120754	LCALL MUL	; K1
0560	90208A	MOV DPTR,#208AH	
0563	120660	LCALL MOVE1	
0566	902125	MOV DPTR,#2125H	; K1 --> @2125H
0569	12066C	LCALL MOVE2	
056C	902130	MOV DPTR,#2130H	
056F	120660	LCALL MOVE1	
0572	902135	MOV DPTR,#2135H	; En-2 <-- En-1
0575	12066C	LCALL MOVE2	
0578	90206A	MOV DPTR,#206AH	
057B	120660	LCALL MOVE1	
057E	902130	MOV DPTR,#2130H	; En-1 <-- En
0581	12066C	LCALL MOVE2	
0584	902140	MOV DPTR,#2140H	; En-2 * A2
0587	120660	LCALL MOVE1	; En-2 --> @2140
058A	902080	MOV DPTR,#2080H	
058D	12066C	LCALL MOVE2	
0590	9020D0	MOV DPTR,#20D0H	
0593	120660	LCALL MOVE1	
0596	902085	MOV DPTR,#2085H	
0599	12066C	LCALL MOVE2	
059C	120754	LCALL MUL	; K2
059F	90218A	MOV DPTR,#218AH	
05A2	120660	LCALL MOVE1	
05A5	90212A	MOV DPTR,#212AH	; K2 --> @212AH
05A8	12066C	LCALL MOVE2	
05AB	902135	MOV DPTR,#2135H	
05AE	120660	LCALL MOVE1	
05B1	902140	MOV DPTR,#2140H	; En-2 <-- En-1
05B4	12066C	LCALL MOVE2	
05B7	90212A	MOV DPTR,#212AH	
05BA	120660	LCALL MOVE1	
05BD	902060	MOV DPTR,#2060H	
05C0	12066C	LCALL MOVE2	
05C3	902120	MOV DPTR,#2120H	
05C6	120660	LCALL MOVE1	
05C9	902165	MOV DPTR,#2165H	
05CC	12066C	LCALL MOVE2	
05CF	120809	LCALL ADD	; K2 + K0
05D2	90206A	MOV DPTR,#206AH	

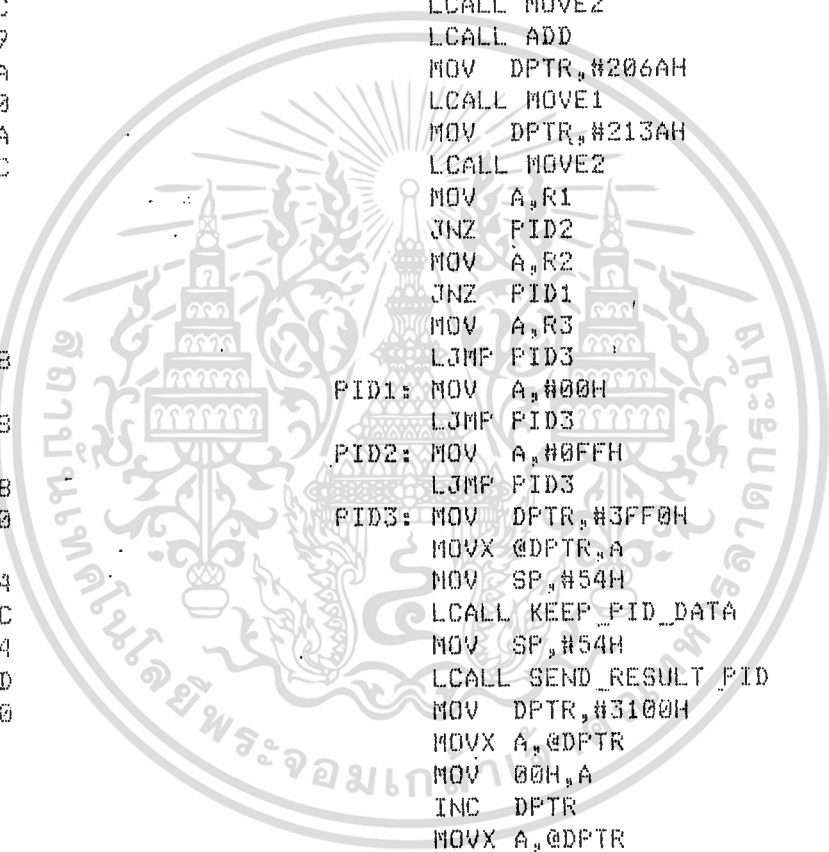


เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติให้นำไปใช้ประโยชน์ด้านการค้า  
 สืบ อีกรั้งห้ามมิให้ดัดแปลงเนื้อหา และสงวนลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

```

05D5 120660          LCALL MOVE1
05D8 902060          MOV DPTR,#2060H
05DB 120660          LCALL MOVE2
05DE 902125          MOV DPTR,#2125H
05E1 120660          LCALL MOVE1
05E4 902065          MOV DPTR,#2065H
05E7 120660          LCALL MOVE2
05EA 120839          LCALL SUB           ; (K2 + K0) -K1
05ED 90213A          MOV DPTR,#213AH    ; DON'T CARE LABEL
05F0 120660          LCALL MOVE1
05F3 902060          MOV DPTR,#2060H
05F6 120660          LCALL MOVE2
05F9 90206A          MOV DPTR,#206AH
05FC 120660          LCALL MOVE1
05FF 902065          MOV DPTR,#2065H
0602 120660          LCALL MOVE2
0605 120809          LCALL ADD           ; Cn-1 + (K0-K1)
0608 90206A          MOV DPTR,#206AH
060B 120660          LCALL MOVE1
060E 90213A          MOV DPTR,#213AH    ; Cn-1 <-- Cn
0611 120660          LCALL MOVE2
0614 E9              MOV A,R1
0615 700C             JNZ PID2
0617 EA              MOV A,R2
0618 7004             JNZ PID1
061A EB              MOV A,R3
061E 020628           LJMP PID3
0620 7400             PID1: MOV A,#00H
0623 74FF             LJMP PID3
0625 020628           PID2: MOV A,#0FFH
0628 903FF0           LJMP PID3
062B F0              PID3: MOV DPTR,#3FF0H
062C 758154           MOVX @DPTR,A
062F 1208CC           MOV SP,#54H
0632 758154           LCALL KEEP_PID_DATA
0635 1208DD           MOV SP,#54H
0638 903100           LCALL SEND_RESULT_PID
063B E0              MOV DPTR,#3100H    ; RETURN BANK0
063C F500           MOVX A,@DPTR
063E A3              MOV 00H,A
063F E0              INC DPTR
0640 F501           MOVX A,@DPTR
0642 A3              MOV 01H,A
0643 E0              INC DPTR
0644 F502           MOVX A,@DPTR
0646 A3              MOV 02H,A
0647 E0              INC DPTR
0648 F503           MOVX A,@DPTR
064A A3              MOV 03H,A
064B E0              INC DPTR
064C F504           MOVX A,@DPTR
064E A3              MOV 04H,A
064F E0              INC DPTR
0650 F505           MOVX A,@DPTR
0652 A3              MOV 05H,A
0653 E0              INC DPTR
0654 F506           MOVX A,@DPTR
0656 A3              MOV 06H,A
0657 E0              INC DPTR
0658 F507           MOVX A,@DPTR
065A 75D000           MOV 07H,A
065B 75D000           MOV PSW,#00H      ; RETURN TO BANK

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่สามารถทำซ้ำได้ทั้งหมด มีให้ดาวน์โหลดเนื้อหา และอื่นๆ อีกมากมายของเอกสารทุกครั้งที่มีการนำไปใช้

065D 02026D LJMP LOAD\_NEXT\_DATA

```

0660 E0 MOVE1: MOVX A,@DPTR
0661 F9 MOV R1,A
0662 A3 INC DPTR
0663 E0 MOVX A,@DPTR
0664 FA MOV R2,A
0665 A3 INC DPTR
0666 E0 MOVX A,@DPTR
0667 FB MOV R3,A
0668 A3 INC DPTR
0669 E0 MOVX A,@DPTR
066A FC MOV R4,A
066B 22 RET

```

```

066C E9 MOVE2: MOV A,R1
066D F0 MOVX @DPTR,A
066E A3 INC DPTR
066F EA MOV A,R2
0670 F0 MOVX @DPTR,A
0671 A3 INC DPTR
0672 EB MOV A,R3
0673 F0 MOVX @DPTR,A
0674 A3 INC DPTR
0675 EC MOV A,R4
0676 F0 MOVX @DPTR,A
0677 22 RET

```

```

0678 902080 FIA0: MOV DPTR,#2080H
067B 7400 MOV A,#00H
067D F0 MOVX @DPTR,A
067E A3 INC DPTR
067F F0 MOVX @DPTR,A
0680 A3 INC DPTR
0681 7464 MOV A,#64H
0683 F0 MOVX @DPTR,A
0684 A3 INC DPTR
0685 7400 MOV A,#00H
0687 F0 MOVX @DPTR,A
0688 902045 MOV DPTR,#2045H
068B 120660 LCALL MOVE1
068E 902085 MOV DPTR,#2085H
0691 12066C LCALL MOVE2
0694 00 NOP
0695 120754 LCALL MUL
0698 902090 MOV DPTR,#2090H
069B 7400 MOV A,#00H
069D F0 MOVX @DPTR,A
069E A3 INC DPTR
069F F0 MOVX @DPTR,A
06A0 A3 INC DPTR
06A1 7401 MOV A,#01H
06A3 F0 MOVX @DPTR,A
06A4 A3 INC DPTR
06A5 7400 MOV A,#00H
06A7 F0 MOVX @DPTR,A
06A8 902081 MOV DPTR,#208AH
06AB 120660 LCALL MOVE1
06AE 902095 MOV DPTR,#2095H
06B1 12066C LCALL MOVE2
06B4 00 NOP
06B5 1208AD LCALL DIVS

```

; Kp(1+del T/Ti)

; 100\*Ti

; 16/(100\*Ti)

เอกสารนี้เป็นทรัพย์สินที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 การวิจัย หรือ สืบค้น อื่นๆ ที่ไม่มีให้ตัดแปลงเนื้อหา และเผยแพร่ต่อสาธารณะโดยไม่ได้รับอนุญาตของเอกสารทุกครั้งที่มีการนำไปใช้

06B8 90209A  
 06BB 120660  
 06BE 902060  
 06C1 12066C  
 06C4 902065  
 06C7 7400  
 06C9 F0  
 06CA A3  
 06CB F0  
 06CC A3  
 06CD 7401  
 06CF F0  
 06D0 A3  
 06D1 7400  
 06D3 F0  
 06D4 120809  
 06D7 902060  
 06DA 120660  
 06DD 902080  
 06E0 12066C  
 06E3 902040  
 06E6 120660  
 06E9 902085  
 06EC 12066C  
 06EF 00  
 06F0 120754  
 06F3 90208A  
 06F6 120660  
 06F9 9020C5  
 06FC 12066C  
 06FF 22

MOV DPTR,#209AH  
 LCALL MOVE1  
 MOV DPTR,#2060H  
 LCALL MOVE2  
 MOV DPTR,#2065H  
 MOV A,#00H  
 MOVX @DPTR,A  
 INC DPTR  
 MOVX @DPTR,A  
 INC DPTR  
 MOV A,#01H  
 MOVX @DPTR,A  
 INC DPTR  
 MOV A,#00H  
 MOVX @DPTR,A  
 LCALL ADD  
 MOV DPTR,#206AH  
 LCALL MOVE1  
 MOV DPTR,#2080H  
 LCALL MOVE2  
 MOV DPTR,#2040H  
 LCALL MOVE1  
 MOV DPTR,#2085H  
 LCALL MOVE2  
 NOP  
 LCALL MUL  
 MOV DPTR,#208AH  
 LCALL MOVE1  
 MOV DPTR,#20C5H  
 LCALL MOVE2  
 RET

; 1 + 16/(100\*F)

; Kp(1 + 16/(100\*F))

; PIA0

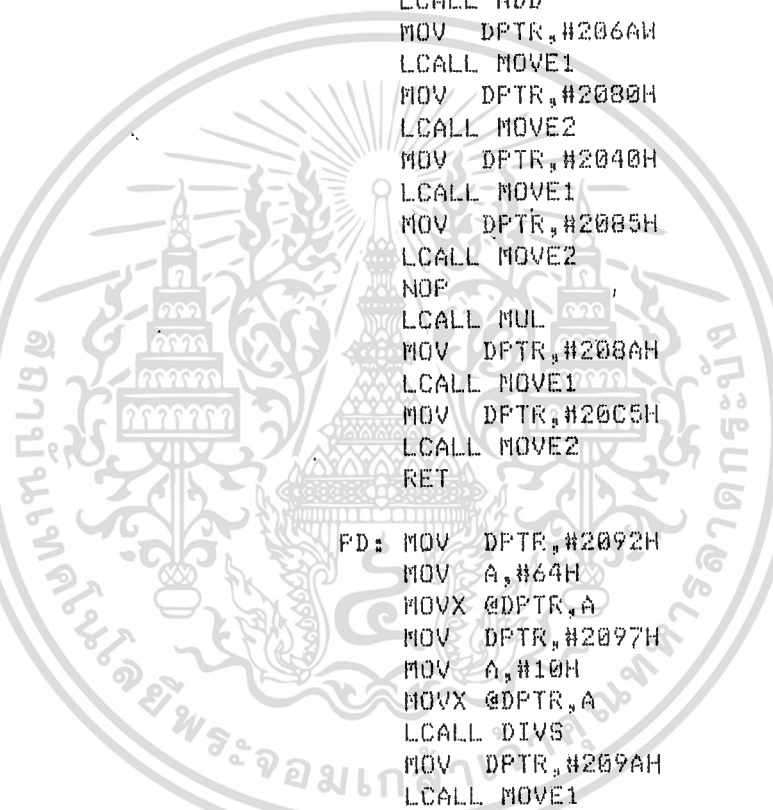
; Kp \* Td/0.16

; 100/16 = 10

; Td/0.16

; Kp \* Td/0.16

; PD



เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ในกรณีที่ท่านมีให้ตัดแปลงเนื้อหา และเอกสารทุกครั้งที่มีการนำไปใช้



07AA 6015  
 07AC FC  
 07AD 8BF0  
 07AF C3  
 07B0 A4  
 07B1 2E  
 07B2 FE  
 07B3 E5F0  
 07B5 3D  
 07B6 FD  
 07B7 EC  
 07B8 8AF0  
 07BA A4  
 07BB 2D  
 07BC FD  
 07BD E5F0  
 07BF 7008  
 07C1 1207FD  
 07C4 00  
 07C5 1207D6  
 07C8 22

07C9 90208A  
 07CC 74FF  
 07CE F0  
 07CF A3  
 07D0 F0  
 07D1 A3  
 07D2 F0  
 07D3 A3  
 07D4 F0  
 07D5 22

07D6 902080  
 07D9 E0  
 07DA F9  
 07DB 902085  
 07DE E0  
 07DF 90208A  
 07E2 C3  
 07E3 99  
 07E4 6004  
 07E6 74FF  
 07E8 F0  
 07E9 22

07EA 7400  
 07EC F0  
 07ED 22

07EE 7900  
 07F0 7A00  
 07F2 7B00  
 07F4 7C00  
 07F6 7D00  
 07F8 7E00  
 07FA 7F00

07FC 22  
 07FD 90208B  
 0800 ED  
 0801 F0

JZ M3  
 MOV R4,A  
 MOV B,R3  
 CLR C  
 MUL AB  
 ADD A,R6  
 MOV R6,A  
 MOV A,B  
 ADDC A,R5  
 MOV R5,A  
 MOV A,R4  
 MOV B,R2  
 MUL AB  
 ADD A,R5  
 MOV R5,A  
 MOV A,B  
 JNZ ERROR2

M3: LCALL SAVE  
 NOP  
 LCALL SIGN  
 RET

ERROR2: MOV DPTR,#208AH  
 MOV A,#0FFH  
 MOVX @DPTR,A  
 INC DPTR  
 MOVX @DPTR,A  
 INC DPTR  
 MOVX @DPTR,A  
 INC DPTR  
 MOVX @DPTR,A  
 INC DPTR  
 MOVX @DPTR,A  
 RET

SIGN: MOV DPTR,#2080H  
 MOVX A,@DPTR  
 MOV R1,A  
 MOV DPTR,#2085H  
 MOVX A,@DPTR  
 MOV DPTR,#208AH  
 CLR C  
 SUBB A,R1  
 JZ SG1  
 MOV A,#0FFH  
 MOVX @DPTR,A  
 RET

SG1: MOV A,#00H  
 MOVX @DPTR,A  
 RET

SET: MOV R1,#00H  
 MOV R2,#00H  
 MOV R3,#00H  
 MOV R4,#00H  
 MOV R5,#00H  
 MOV R6,#00H  
 MOV R7,#00H

RET

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าจะโดยวิธีใดก็ตาม ห้ามมิให้ดัดแปลงเนื้อหา หรือแก้ไข DPTR, #208BH ทุกครั้งที่มีการนำไปใช้  
 0800 ED  
 0801 F0  
 MOVX @DPTR,A

```

0802 A3          INC  DFTR
0803 EE          MOV  A,R6
0804 F0          MOVX @DFTR,A
0805 A3          INC  DFTR
0806 EF          MOV  A,R7
0807 F0          MOVX @DFTR,A
0808 22          RET

0809 02080C      ADD: LJMP STARTA          ; 02060 - 02063
080C 902060      STARTA: MOV DFTR,#2060H ; 02065 - 02068
080F E0          MOVX A,@DFTR          ; RESULT 0206A
0810 F9          MOV  R1,A
0811 902065      MOV  DFTR,#2065H
0814 E0          MOVX A,@DFTR
0815 C3          CLR  C
0816 99          SUBB A,R1
0817 702D        JNZ  SUBS
0819 E9          ADDS: MOV A,R1
081A FF          MOV  R7,A
081B 12088B      LCALL DATA
081E C3          CLR  C
081F 2B          ADD  A,R3
0820 FB          MOV  R3,A
0821 EA          MOV  A,R2
0822 120834      LCALL CH_CARRY
0825 3D          ADDC A,R5
0826 FA          MOV  R2,A
0827 E9          MOV  A,R1
0828 120834      LCALL CH_CARRY
082B 3C          ADDC A,R4
082C F9          MOV  R1,A
082D 90206A      MOV  DFTR,#206AH
0830 1208A1      LCALL SAVEAS
0833 22          RET

0834 5002        CH_CARRY: JNC  RETURN0
0836 04          INC  A
0837 C3          CLR  C
0838 22          RETURN0: RET

0839 902060      SUB: MOV  DFTR,#2060H          ; 02060-02063
083C E0          MOVX A,@DFTR          ; 02065-02068
083D F9          MOV  R1,A          ; RESULT 0206A
083E 902065      MOV  DFTR,#2065H
0841 E0          MOVX A,@DFTR
0842 C3          CLR  C
0843 99          SUBB A,R1
0844 70D3        JNZ  ADDS
0846 12088B      SUBS: LCALL DATA
0849 FE          MOV  R6,A
084A EB          MOV  A,R3
084B C3          CLR  C
084C 9E          SUBB A,R6
084D FB          MOV  R3,A
084E EA          MOV  A,R2
084F 9D          SUBB A,R5
0850 FA          MOV  R2,A
0851 E9          MOV  A,R1
0852 9C          SUBB A,R4
0853 F9          MOV  R1,A
0854 501E        JNC  COR
0856 12088B      LCALL DATA

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 หรือธุรกิจทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และเผยแพร่ต่อสาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

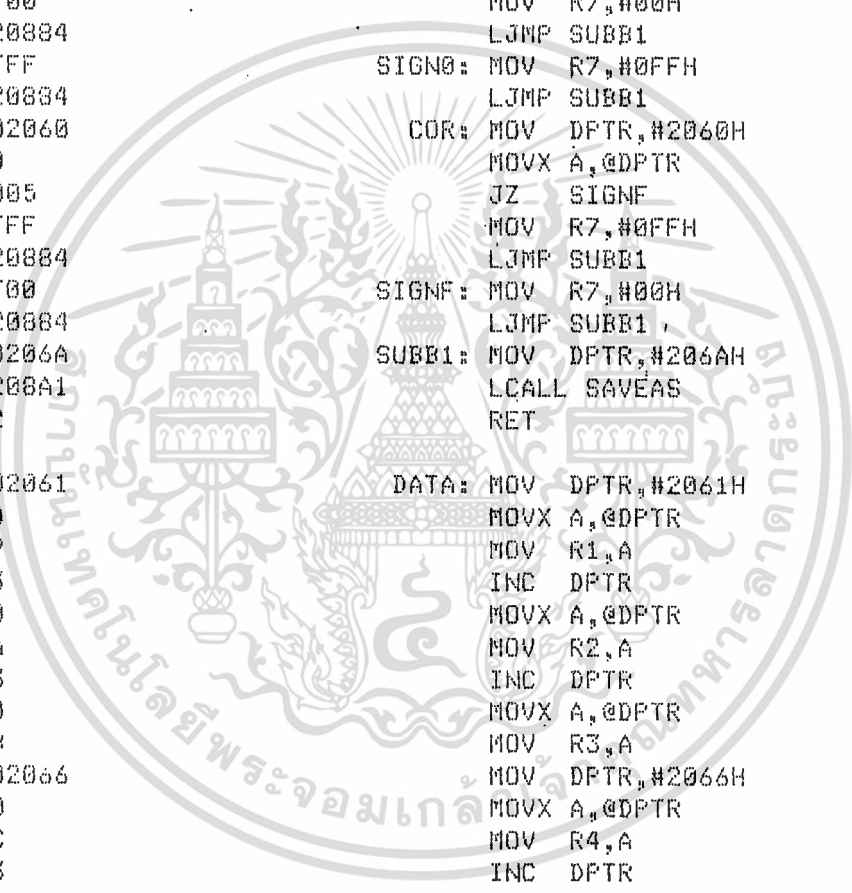
0859 FB      MOV R3,A
085A EE      MOV A,R6
085B C3      CLR C
085C 9B      SUBB A,R3
085D FB      MOV R3,A
085E ED      MOV A,R5
085F 9A      SUBB A,R2
0860 FA      MOV R2,A
0861 EC      MOV A,R4
0862 99      SUBB A,R1
0863 F9      MOV R1,A
0864 902060  MOV DPTR,#2060H
0867 E0      MOVX A,@DPTR
0868 6005     JZ SIGN0
086A 7F00     MOV R7,#00H
086C 020884   LJMP SUBB1
086F 7FFF     SIGN0: MOV R7,#0FFH
0871 020884   LJMP SUBB1
0874 902060  COR:  MOV DPTR,#2060H
0877 E0      MOVX A,@DPTR
0878 6005     JZ SIGNF
087A 7FFF     MOV R7,#0FFH
087C 020884   LJMP SUBB1
087F 7F00     SIGNF: MOV R7,#00H
0881 020884   LJMP SUBB1
0884 90206A   SUBB1: MOV DPTR,#206AH
0887 1208A1   LCALL SAVEAS
088A 22      RET

088B 902061   DATA: MOV DPTR,#2061H
088E E0      MOVX A,@DPTR
088F F9      MOV R1,A
0890 A3      INC DPTR
0891 E0      MOVX A,@DPTR
0892 FA      MOV R2,A
0893 A3      INC DPTR
0894 E0      MOVX A,@DPTR
0895 FB      MOV R3,A
0896 902066   MOV DPTR,#2066H
0899 E0      MOVX A,@DPTR
089A FC      MOV R4,A
089B A3      INC DPTR
089C E0      MOVX A,@DPTR
089D FD      MOV R5,A
089E A3      INC DPTR
089F E0      MOVX A,@DPTR
08A0 22      RET

08A1 EF      SAVEAS: MOV A,R7
08A2 F0      MOVX @DPTR,A
08A3 A3      INC DPTR
08A4 E9      MOV A,R1
08A5 F0      MOVX @DPTR,A
08A6 A3      INC DPTR
08A7 EA      MOV A,R2
08A8 F0      MOVX @DPTR,A
08A9 A3      INC DPTR
08AA EB      MOV A,R3
08AB F0      MOVX @DPTR,A
08AC 22      RET

08AD 802092   PLUS: MOV DPTR,#2092H

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าในกรณีใดๆ หากมีให้ตัดแปลงเนื้อหา และต่อจากเอกสารทุกครั้งที่มีการนำไปใช้

```

08B0 E0          MOVX A,@DPTR          ; @2095H-@2098H
08B1 F8          MOV R0,A              ; RESULT @209AH
08B2 902097      MOV DPTR,#2097H
08B5 E0          MOVX A,@DPTR
08B6 F9          MOV R1,A
08B7 E8          MOV A,R0
08B8 89F0        MOV B,R1
08BA 84          DIV AB
08BB 90209C      MOV DPTR,#209CH
08BE F0          MOVX @DPTR,A
08BF A3          INC DPTR
08C0 E5F0        MOV A,B
08C2 F0          MOVX @DPTR,A
08C3 90209A      MOV DPTR,#209AH
08C6 7400        MOV A,#00H
08C8 F0          MOVX @DPTR,A
08C9 A3          INC DPTR
08CA F0          MOVX @DPTR,A
08CE 22          RET

```

```

; *-----*
; * SAVE PID DATA FOR SEND TO HOST COMPUTER
; *-----*

```

```

08CC 758129      KEEP_PID_DATA: MOV SP,#29H          ; STACK FOR KEEP
08CF D003        POP DPH
08D1 D082        POP DPL
08D3 F0          MOVX @DPTR,A
08D4 A3          INC DPTR
08D5 C082        PUSH DPL
08D7 C083        PUSH DPH
08D9 758156      MOV SP,#56H
08DC 22          RET

```

```

; *-----*
; * SEND RESULT OF PID PROCESS TO CONTROL OUTPUT
; *-----*

```

```

08DD F5F0        SEND_RESULT_PID: MOV B,A          ; (B) <-- PID DA
08DF 75D010      MOV PSW,#10H        ; REG. BANK2
08E2 7823        MOV R0,#23H
08E4 B6000D      CJNE @R0,#00H,SEND_OUT_CH_1
08E7 7820        MOV R0,#20H
08E9 B60027      CJNE @R0,#00H,SEND_OUT_CH_2
08EC 781D        MOV R0,#1DH
08EE B60041      CJNE @R0,#00H,SEND_OUT_CH_3
08F1 020951      LJMP SEND_OUT_CH_4

```

```

08F4 758123      SEND_OUT_CH_1: MOV SP,#23H
08F7 D023        POP 23H
08F9 E523        MOV A,23H
08FB D083        POP DPH
08FD D082        POP DPL
08FF F0          MOVX @DPTR,A
0900 E5F0        MOV A,B
0902 F0          MOVX @DPTR,A
0903 75D008      MOV PSW,#08H        ; RETURN BANK1 T
0906 D51067      DJNZ 10H,RETURN1
0909 752300      MOV 23H,#00H
090C 75100F      MOV 10H,#0FH
090F 758156      MOV SP,#56H
0912 22          RET

```

```

0913 758120      SEND_OUT_CH_2: MOV SP,#20H

```

```

0916 D020          FOP  20H
0918 E520          MOV  A,20H
091A D083          FOP  DFH
091C D082          FOP  DPL
091E F0            MOVX @DPTR,A
091F E5F0          MOV  A,B
0921 F0            MOVX @DPTR,A
0922 75D008        MOV  PSW,#00H          ; RETURN BANK1
0925 D51048        DJNZ 10H,RETURN1
0928 752000        MOV  20H,#00H
092B 75100F        MOV  10H,#0FH
092E 758156        MOV  SP,#56H
0931 22            RET

```

```

0932 75811D        SEND_OUT_CH_3: MOV  SP,#1DH
0935 D01D          FOP  1DH
0937 E51D          MOV  A,1DH
0939 D083          FOP  DFH
093B D082          FOP  DPL
093D F0            MOVX @DPTR,A
093E E5F0          MOV  A,B
0940 F0            MOVX @DPTR,A
0941 75D008        MOV  PSW,#00H          ; RETURN BANK1
0944 D51029        DJNZ 10H,RETURN1
0947 751D00        MOV  1DH,#00H
094A 75100F        MOV  10H,#0FH
094D 758156        MOV  SP,#56H
0950 22            RET

```

```

0951 75811A        SEND_OUT_CH_4: MOV  SP,#1AH
0954 D01A          FOP  1AH
0956 E51A          MOV  A,1AH
0958 D083          FOP  DFH
095A D082          FOP  DPL
095C F0            MOVX @DPTR,A
095D E5F0          MOV  A,B
095F F0            MOVX @DPTR,A
0960 75D008        MOV  PSW,#00H          ; RETURN BANK1
0963 D5100A        DJNZ 10H,RETURN1
0966 751A00        MOV  1AH,#00H
0969 75100F        MOV  10H,#0FH
096C 758156        MOV  SP,#56H
096F 22            RET

```

```

0970 758156        RETURN1: MOV  SP,#56H
0973 22            RET

```

```

*-----*
* SEND PROCESS & UNPROCESS DATA TO HOST COMPUTER
*-----*

```

```

0974 75D000        SEND_DATA: MOV  PSW,#00H
0977 758155        MOV  SP,#55H
097A 90305F        MOV  DPTR,#305FH
097D E0            MOVX A,@DPTR
097E B40014        CJNE A,#00H,SEND_4_CH
0981 C3            CLR  C
0982 90304F        MOV  DPTR,#304FH
0985 E0            MOVX A,@DPTR
0986 B40033        CJNE A,#00H,SEND_3_CH
0989 C3            CLR  C
098A 90303F        MOV  DPTR,#303FH
098D E0            MOVX A,@DPTR

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา  
 ไม่สงวนลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า  
 อื่นๆ หากมีให้ตัดแปลงเนื้อหา และแจ้งเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

098E D4004B      CJNE A,#00H,SEND_2_CH
0991 C3          CLR C
0992 0209F3      LJMP SEND_1_CH

```

```

0995 C3          SEND_4_CH: CLR C
0996 903020      MOV DPTR,#3020H
0999 7F0E        MOV R7,#0EH
099B 120A02      LCALL SEND_DATA_4_CH
099E 903021      MOV DPTR,#3021H
09A1 7F0E        MOV R7,#0EH
09A3 120A02      LCALL SEND_DATA_4_CH
09A6 903022      MOV DPTR,#3022H
09A9 7F0E        MOV R7,#0EH
09AB 120A02      LCALL SEND_DATA_4_CH
09AE 903023      MOV DPTR,#3023H
09B1 7F0E        MOV R7,#0EH
09B3 120A02      LCALL SEND_DATA_4_CH
09B6 120A71      LCALL SEND_DATA_END
09B9 020A7F      LJMP ENDED

```

```

09BC C3          SEND_3_CH: CLR C
09BD 903020      MOV DPTR,#3020H
09C0 7F0F        MOV R7,#0FH
09C2 120A17      LCALL SEND_DATA_3_CH
09C5 00          NOP
09C6 903021      MOV DPTR,#3021H
09C9 7F0F        MOV R7,#0FH
09CB 120A17      LCALL SEND_DATA_3_CH
09CE 903022      MOV DPTR,#3022H
09D1 7F0F        MOV R7,#0FH
09D3 120A17      LCALL SEND_DATA_3_CH
09D6 120A71      LCALL SEND_DATA_END
09D9 020A7F      LJMP ENDED

```

```

09DC C3          SEND_2_CH: CLR C
09DD 903020      MOV DPTR,#3020H
09E0 7F0F        MOV R7,#0FH
09E2 120A2B      LCALL SEND_DATA_2_CH
09E5 903021      MOV DPTR,#3021H
09E8 7F0F        MOV R7,#0FH
09EA 120A2B      LCALL SEND_DATA_2_CH
09ED 120A71      LCALL SEND_DATA_END
09F0 020A7F      LJMP ENDED

```

```

09F3 C3          SEND_1_CH: CLR C
09F4 903020      MOV DPTR,#3020H
09F7 7F0F        MOV R7,#0FH
09F9 120A3E      LCALL SEND_DATA_1_CH
09FC 120A71      LCALL SEND_DATA_END
09FF 020A7F      LJMP ENDED

```

```

0A02 120A78      SEND_DATA_4_CH: LCALL ACK2          ; SEND '#'
0A05 120A60      LCALL PROMPT_SEND      ; CHECK '?'
0A08 E0          SEND_DATA_AGAIN4: MOVX A,@DPTR
0A09 C299        CLR TI
0A0B F599        MOV SBUF,A
0A0D 120A51      LCALL PROMPT_RECEIVE3  ; CHECK '#'
0A10 A3          INC DPTR
0A11 A3          INC DPTR
0A12 A3          INC DPTR
0A13 A3          INC DPTR
0A14 DFF2        DJNZ R7,SEND_DATA_AGAIN4

```

```

0A15 22                                RET

0A17 120A78          SEND_DATA_3_CH: LCALL ACK2                ; SEND '$'
0A1A 120A60          LCALL PROMPT_SEND                ; CHECK '?'
0A1D E0              SEND_DATA_AGAIN3: MOVX A,@DPTR
0A1E C299            CLR TI
0A20 F599            MOV SBUF,A
0A22 120A51          LCALL PROMPT_RECEIVE3            ; CHECK '#'
0A25 A3              INC DPTR
0A26 A3              INC DPTR
0A27 A3              INC DPTR
0A28 DFF3            DJNZ R7,SEND_DATA_AGAIN3
0A2A 22              RET

0A2B 120A78          SEND_DATA_2_CH: LCALL ACK2                ; SEND '$'
0A2E 120A60          LCALL PROMPT_SEND                ; CHECK '?'
0A31 E0              SEND_DATA_AGAIN2: MOVX A,@DPTR
0A32 C299            CLR TI
0A34 F599            MOV SBUF,A
0A36 120A51          LCALL PROMPT_RECEIVE3            ; CHECK '#'
0A39 A3              INC DPTR
0A3A A3              INC DPTR
0A3B DFF4            DJNZ R7,SEND_DATA_AGAIN2
0A3D 22              RET

0A3E 120A78          SEND_DATA_1_CH: LCALL ACK2                ; SEND '$'
0A41 120A60          LCALL PROMPT_SEND                ; CHECK '?'
0A44 E0              SEND_DATA_AGAIN1: MOVX A,@DPTR
0A45 C299            CLR TI
0A47 F599            MOV SBUF,A
0A49 120A51          LCALL PROMPT_RECEIVE3            ; CHECK '#'
0A4C A3              INC DPTR
0A4D A3              INC DPTR
0A4E DFF4            DJNZ R7,SEND_DATA_AGAIN1
0A50 22              RET

0A51 209802          PROMPT_RECEIVE3: JB RI,RECEIVE_ACK2
0A54 80FB            SJMP PROMPT_RECEIVE3
0A56 C3              RECEIVE_ACK2: CLR C
0A57 C298            CLR RI
0A59 E599            MOV A,SBUF
0A5B B423F3          CJNE A,#23H,PROMPT_RECEIVE3
0A5E C3              CLR C
0A5F 22              RET

0A60 209802          PROMPT_SEND: JB RI,RECEIVE_ACK3
0A63 80FB            SJMP PROMPT_SEND
0A65 C3              RECEIVE_ACK3: CLR C
0A66 C298            CLR RI
0A68 E599            MOV A,SBUF
0A6A B43FF3          CJNE A,#3FH,PROMPT_SEND
0A6D C299            CLR SBUF
0A6F C3              CLR C
0A70 22              RET

0A71 7440            SEND_DATA_END: MOV A,#40H
0A73 C299            CLR TI
0A75 F599            MOV SBUF,A
0A77 22              RET

0A78 7424            ACK2: MOV A,#24H
0A7A C299            CLR TI

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
 ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 และไม่อนุญาตให้นำไปใช้  
 อื่นๆ หากมีให้ตัดแปลงเนื้อหา และต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0A7C F599  
0A7E 22

NOV SBUF, A.  
RET

0000

ENDED: END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00E0	A	00E0	A0	00E1	A1
00E2	A2	00E3	A3	00E4	A4
00E5	A5	00E6	A6	00E7	A7
00D6	AC	0A78	ACK2	0809	ADD
0819	ADDS	01BB	AGAIN	00F0	B
004C	BAUD_1200	0047	BAUD_2400	0042	BAUD_4800
003D	BAUD_9600	00DF	C	0348	CH1_MAX_ERROR
0366	CH1_MIN_ERROR	0350	CH2_MAX_ERROR	036E	CH2_MIN_ERROR
0358	CH3_MAX_ERROR	0376	CH3_MIN_ERROR	0360	CH4_MAX_ERROR
037E	CH4_MIN_ERROR	0071	CHECK0	0076	CHECK1
0059	CHECK2	005E	CHECK3	01C9	CHECK_AGAIN1
01CC	CHECK_AGAIN2	01CF	CHECK_AGAIN3	01D2	CHECK_AGAIN4
01D5	CHECK_AGAIN5	01D8	CHECK_AGAIN6	01DB	CHECK_AGAIN7
01DE	CHECK_AGAIN8	0383	CHECK_DATA	00E4	CHECK_END0
00E9	CHECK_END1	0263	CHECK_MAX_ERROR	0268	CHECK_MIN_ERROR
0270	CHECK_NEXT_DATA	010E	CHECK_START	0111	CHECK_START1
01E4	CH_1	01EB	CH_2	01F2	CH_3
01F9	CH_4	0200	CH_5	0207	CH_6
020E	CH_7	0215	CH_8	0834	CH_CARRY
00AB	CH_SELECT	000F	CLEAR_EX	000A	CLEAR_EX_RAM
0002	CLEAR_IN	0000	CLEAR_IN_RAM	0335	COMPARE_1_CH
0322	COMPARE_2_CH	030E	COMPARE_3_CH	02F9	COMPARE_4_CH
028C	COMPARE_MAX	0223	COMPARE_MAX_MIN	0280	COMPARE_MIN
025A	COMPARE_NEXT_CH	0260	COMPARE_NEXT_DATA	0146	CONTINUE
0874	COR	088B	DATA	08AD	DIVS
0083	DPH	0082	DPL	0A7F	ENDED
01E1	END_CHECK_AGAIN	07C9	ERROR2	00A8	IE
02E1	INCREASE_CHANNAL	0196	INCREASE_DPH	0016	INIT_SERIAL
00B8	IF	0133	KEEP_DATA_AGAIN	01C1	KEEP_DPTR
019D	KEEP_OUTPUT_CARD	08CC	KEEP_PID_DATA	0399	L1
03A6	L2	03B3	L3	03C0	L4
03CD	L5	03DA	L6	026D	LOAD_NEXT_DATA
02DC	LOWER_MIN	0780	M1	07A6	M2
07C1	M3	0347	MAX_ERROR	0365	MIN_ERROR
0660	MOVE1	066C	MOVE2	0754	MUL
01A9	NEXT	0090	NEXT_CARD	0265	NEXT_COMPARE
01AD	NEXT_OUTPUT	021C	NEXT_OUTPUT_CARD	02D7	OVER_MAX
0080	P0	0090	P1	0090	P1_0
0091	P1_1	0092	P1_2	0093	P1_3
0094	P1_4	0095	P1_5	0096	P1_6
0097	P1_7	00A0	P2	00B0	P3
0087	PCON	0700	PD	0678	PIA0
0503	PID	061E	PID1	0623	PID2
0628	PID3	03DD	PID_PROCESS	0175	POLLING
026A	PROCESS_CONTROL	0A51	PROMPT_RECEIVE3	0116	PROMPT_RUN
0A60	PROMPT_SEND	00D0	PSW	048A	P_PID
00D9	RECEIVE	00FD	RECEIVE0	0A56	RECEIVE_ACK2
0A65	RECEIVE_ACK3	00DC	RECEIVE_AGAIN	0089	RECEIVE_CH
00AF	RECEIVE_CH0	00B4	RECEIVE_CH1	00CA	RECEIVE_CONST
00F8	RECEIVE_CONST1	0127	RECEIVE_DATA	0838	RETURN0
0970	RETURN1	0276	RETURN_CHECK	0098	RI
016B	ROTATE	0165	ROTATE_8_BITS	0099	ROTATE_AGAIN
0170	ROTATE_AGAIN1	0094	ROTATE_CARD	0143	ROTATE_CHANNAL
0159	ROTATE_NEXT_CARD	07FD	SAVE	08A1	SAVEAS
0099	SBUF	0098	SCON	09F3	SEND_1_CH
09DC	SEND_2_CH	09BC	SEND_3_CH	0995	SEND_4_CH
0974	SEND_DATA	0A3E	SEND_DATA_1_CH	0A2B	SEND_DATA_2_CH
0A17	SEND_DATA_3_CH	0A02	SEND_DATA_4_CH	0A44	SEND_DATA_AGAIN1
0A31	SEND_DATA_AGAIN2	0A1D	SEND_DATA_AGAIN3	0A08	SEND_DATA_AGAIN4
0A71	SEND_DATA_END	08F4	SEND_OUT_CH_1	0913	SEND_OUT_CH_2
0932	SEND_OUT_CH_3	0951	SEND_OUT_CH_4	0082	SEND_POINT
08DD	SEND_RESULT_PID	0120	SEND_SHARP	07EE	SET



```

{
/* int w=0; */
/* sel_ch(); */
/* dis(); */
/* while(!kbhit()) */
/* { */
port_init();
sendp();
rec_file();

/* } */
}
c3()
{
sel_graph();
dis();
/* graph(1,use1,use2,use3,use4,0); */
}
dis()
{
clrscr();
gotoxy(3,3);cprintf("use1 ch is ..%3d",use1);
gotoxy(3,4);cprintf("use2 ch is ..%3d",use2);
gotoxy(3,5);cprintf("use3 ch is ..%3d",use3);
gotoxy(3,6);cprintf("use4 ch is ..%3d",use4);
while (!kbhit());
}
dis1()
{
int n;
window(34,6,74,16);
clrscr();
for(n=0;n<32;n++) {
gotoxy(2,n+1);
cprintf("card[%d]=%d c_h[%d]=%d",n,card[n],n,c_h[n]);
}
while (!kbhit());
clrscr();
for(n=0;n<=3;n++) {
gotoxy(2,n+1);
cprintf("ucard[%d] ..%d",n,ucard[n]);
}
}
/* receive file */
rec_file()
{
FILE *fp;
char ch;
union
{
char c[2];
long unsigned int count;
} cnt;
get_data(data4); /* receive data */
/* gotoxy(2,9); */
printf(" receiving file %s\n ",data4);
remove(data4);
if(!( fp=fopen("data4","wb") ))
gotoxy(2,9);
cprintf(" cannot open output file ");
exit(1);
}
fclose(fp);

```

```

}
/* receive data4[256][14] */
get_data()
{
    int i,j;
    sport('%'); /* code start */
    /* gotoxy(2,8) */
    printf(" receiver waiting... ");
    do {
        rport(); } while ( R != '$' );
    sport('?');
    for(j=0;j<=255;j++) {
        for(i=0;i<=13;i++)
            data4[j][i] = 0;
    }

    for(j=0;j<=255;j++) {
        for(i=0;i<=13;i++) {
            rport();
            data4[j][i] = R;
            printf(" receive data4[%d][%d] is %d ",j,i,data4[j][i]);
            sport('#');
        }
        do {
            rport(); } while ((R != '@') && (R != '$'));
        if ( R != '$')
            break;
        else
            sport('?');
    }
}
/*send parameter*/
int dd[256];
sendp()
{
    int i,j,mn;
    do { sport('?');
        rport();
    } while ( R != '.' );
    for(i=0;i<=3;i++)
    { sport((char)ucard[i]);
        waitb(); }
    sport('?');
    wait();
    for(j=0;j<=31;j++)
    { if(c_h[j]!=0)
        { sport((char)c_h[j]);
            waitb();
            sport('?');
            wait();
        }
    }

    for(j=0;j<=31;j++)
    {
        if(c_h[j]!=0)
        {
            if((c_h[j]&1))
            { sport('?');wait();setemp(j,0);}
            if((c_h[j]&2))
            { sport('?');wait();setemp(j,1);}
            if((c_h[j]&4))

```

เอกสารนี้เป็นเอกสารใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ หากมีข้อผิดพลาด กรุณาแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { sport('?');wait();setemp(j,2);}
if((c_h[j]&8))
    { sport('?');wait();setemp(j,3);}
if((c_h[j]&16))
    { sport('?');wait();setemp(j,4);}
if((c_h[j]&32))
    { sport('?');wait();setemp(j,5);}
if((c_h[j]&64))
    { sport('?');wait();setemp(j,6);}
if((c_h[j]&128))
    { sport('?');wait();setemp(j,7);}

printf(" channel...%d ",c_h[j]);
}
}
sport('@');
wait4();

for(j=0;j<=31;j++)
{
    if(c_h[j]!=0)
    {
        if((c_h[j]&1))
            { sport('?');wait();setpid(j,0);}
        if((c_h[j]&2))
            { sport('?');wait();setpid(j,1);}
        if((c_h[j]&4))
            { sport('?');wait();setpid(j,2);}
        if((c_h[j]&8))
            { sport('?');wait();setpid(j,3);}
        if((c_h[j]&16))
            { sport('?');wait();setpid(j,4);}
        if((c_h[j]&32))
            { sport('?');wait();setpid(j,5);}
        if((c_h[j]&64))
            { sport('?');wait();setpid(j,6);}
        if((c_h[j]&128))
            { sport('?');wait();setpid(j,7);}
        printf(" channel...%d ",c_h[j]);
    }
}
}
sport('@');
wait4();
}
setemp(m,n)
int m,n;
{
    sport((char)t_max[m][n]);
    waitb();
    sport((char)t_min[m][n]);
    waitb();
    sport((char)t_set[m][n]);
    waitb();
}
setpid(m,n)
int m,n;
{
    sport((char)Kp[m][n]);
    waitb();
    sport((char)Ti[m][n]);
    waitb();
    sport((char)Td[m][n]);
}

```



เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่สามารถเผยแพร่ซ้ำโดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

waitb();
}
/* select channel graph */
int use1,use2,use3,use4;
sel_graph()
{
    int m1,m,a,ucard[4],uch[4],key12,ya1;
    char key10,ya2[2],key11;
    key10 = 0;
    do
    {
        if((key10 == 'c')||(key10 == 'C'))
        {
            m1 = 1;
            clrscr();
            gotoxy(2,2);
            cprintf("select number...(1-4)");
            gotoxy(4,4);
            for(a=1;a<=4;a++)
                { cprintf("[%d]card=%2d channel=%d",a,ucard[a-1],uch[a-1]);
                  gotoxy(4,4+a);}
            do
            {
                key11 = getch();
                key12 = key11-48;
            }while((key12<1)||((key12>4)));
            }
            else m1=4;
            gotoxy(2,2);
            cprintf("select channel to print graph");
            for(m=1;m<=m1;m++)
            {
                clrscr();
                gotoxy(2,2);
                cprintf("select channel to print graph");
                if(m1==4) key12=m;
                gotoxy(8,4);cprintf("what's the card (1-32).?");
                do
                {
                    gotoxy(9,5);
                    delline();
                    insline();
                    cprintf(" card.....");
                    gotoxy(9,20);
                    gets(ya2);
                    ya1 = atoi(ya2);
                    ucard[key12-1] = ya1;
                    ya2[0]='\0';ya2[1]='\0';
                    if((ucard[key12-1]<=0)||((ucard[key12-1]>32))
                    {
                        gotoxy(9,5);
                        delline();
                        insline();
                        gotoxy(10,6);
                        cprintf("only card 1-32 select again");
                    }
                    else
                    {
                        gotoxy(10,6);
                        delline();
                    }
                }while((ucard[key12-1]<=0)||((ucard[key12-1]>32)));
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น กรุณาแจ้งและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

else
{
    gotoxy(10,6);
    delline();
}
}while((card[m]<=0);!(card[m]>32));
gotoxy(6,8);cprintf("-----");
gotoxy(7,9); cprintf("(a,A) set all chananel");
gotoxy(7,10); cprintf("(s,S) set some channel");
key2 = getch();
if((key2 == 'a')||(key2 == 'A')) /*modified all channels*/
{
    r_temp(t_set[card[m]-1],t_max[card[m]-1],t_min[card[m]-1],&value,&c_h[card[
    clrscr();
    gotoxy(8,3);cprintf("what's the card (1-32).?");
}
gotoxy(7,10);
textcolor(RED+BLINK);
cprintf("press * when success");
textcolor(YELLOW);
if((key2 == 's')||(key2 == 'S'))
{
    clrscr();
    gotoxy(7,2);
    textcolor(RED);
    cprintf("enter modified channel");
    gotoxy(10,10);
    textcolor(RED+BLINK);
    cprintf("press * when succes");
    textcolor(YELLOW);
    gotoxy(35,2);
    key4 = getch();if(key4=='*'); /* printf("\07"); */
    gotoxy(7,2);
    cprintf("enter modified channel");
    gotoxy(10,10);delline();insline();
    while(((key4>56);!(key4<49))&&(key4!='*')) {
        gotoxy(35,2);
        key4 = getch();if(key4=='*'); /* printf("\07"); */
    }
}
c_h[card[m]-1] = 0;
while(key4!='*') {
    key3 = key4-48;
    value = 1;
    t_set[card[m]-1][key3-1] = t_max[card[m]-1][key3-1] = t_min[card[m]-1][
    gotoxy(5,4);
    cprintf("channel#%d",key3);
    gotoxy(5,5);
    cprintf("enter new setpoint      max      min");
    gotoxy(5,6);
    delline();
    insline();
    scanf("%d",&t_set[card[m]-1][key3-1]);
    gotoxy(10,6);
    scanf("%d",&t_max[card[m]-1][key3-1]);
    gotoxy(15,6);
    scanf("%d",&t_min[card[m]-1][key3-1]);
    gotoxy(5,6);delline();insline();
    if(key3 ==1)
        c_h[card[m]-1] = value;
else
{
    value = 1;
    for(num =1; num<key3; num++)

```

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดก็ตาม หากมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

} /*end of while*/
for(k=0;k<=3;k++) ucard[k]=0;
for(i=0;i<=31;i++) {
    if(card[i]!=0)
    { book = 1;
      if((card[i]>=1)&&(card[i]<=8)) {t = card[i]-1;k=0;}
      if((card[i]>=9)&&(card[i]<=16)) {t = card[i]-9;k=1;}
      if((card[i]>=17)&&(card[i]<=24)) {t = card[i]-17;k=2;}
      if((card[i]>=25)&&(card[i]<=32)) {t = card[i]-25;k=3;}
      if(t==0) book=1;
      else for(j=1;j<=t;j++) book *=2;
      ucard[k] += book;}
}
} /* end of setch() */

```

/\*received setpoint max min for every channels\*/

```

r_temp(t_set,t_max,t_min,va,ch,m)
int t_set[],t_max[],t_min[];
int *va,*m,*ch;

{
    int num;
    int i,n,key6,key3,key4;

    clrscr();
    for(n=0;n<=7;n++)
    t_set[n] = t_max[n] = t_min[n] = 0;
    n = 0; *ch = 0;
    while(n<=7)
    {
        gotoxy(5,4);
        cprintf("channel#%d ",n+1);
        *va = 1;
        if(n == 0)
            *ch = 1;
        else
        {
            for(num =0; num <n; num++)
                *va *=2;
            *ch += *va;
        } /*end of else*/
        gotoxy(5,5);
        cprintf("enter setpoint max min");
        gotoxy(5,6);
        delline();
        insline();
        scanf("%d",t_set+n);
        gotoxy(10,6);
        scanf("%d",t_max+n);
        gotoxy(15,6);
        scanf("%d",t_min+n);
        gotoxy(5,7);
        n = n++;
    } /*end of while*/
    clrscr();
    for (n = 0; n<8;n++)
    { gotoxy(2,2+n);
      cprintf("channel#%d max %5d min %5d",n+1,t_set[n],t_max[n],t_min[n]);
      gotoxy(5,3+n);
      cprintf("ch[%d] = %d",*m,*ch);
      gotoxy(5,10);
      cprintf("press 'c' to improve");
    }
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร สำนักงานปลัดกระทรวงการอุดมศึกษา วิทยาศาสตร์ วิจัยและนวัตกรรม

ไม่ว่าการคัดลอกหรือการเผยแพร่โดยไม่ได้รับอนุญาตจากศูนย์ฯ จะต้องมีแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

key6 = getch();printf("\07");
clrscr();
if((key6=='c')||(key6=='C'))
{
    clrscr();
    gotoxy(7,2);
    textcolor(RED);
    cprintf("enter modified channel ");
    gotoxy(10,10);
    textcolor(RED+BLINK);
    cprintf("press * when succes");
    textcolor(YELLOW);
    gotoxy(35,2);
    key4 = getch();printf("\07");
    gotoxy(7,2);cprintf("enter modified channel");
    gotoxy(10,10);delline();insline();
    while(key4!='*')
    {
        key3 = key4-48;
        while((key3 <=8)&&(key3 >0))
        {
            t_set[key3-1] = t_max[key3-1] = t_min[key3-1] = 0;
            gotoxy(5,4);
            cprintf("channel#%d ",key3);
            gotoxy(5,5);
            cprintf("enter new setpoint      max      min");
            gotoxy(5,6);
            delline();
            insline();
            scanf("%d",&t_set[key3-1]);
            gotoxy(10,6);
            scanf("%d",&t_max[key3-1]);
            gotoxy(15,6);
            scanf("%d",&t_min[key3-1]);
            cprintf("");
            gotoxy(7,2);
            textcolor(RED);
            cprintf("enter modified channel ");
            gotoxy(10,10);
            gotoxy(35,2);
            key4 = getch();
            textattr(YELLOW);
            key3 = key4 - 48;
        }
        /*end of while*/
    }
    /*end of while*/
    clrscr();
    for (n = 0; n<8;n++)
{ gotoxy(2,2+n);
  cprintf("channel#%d  %5d  %5d  %5d",n+1,t_set[n],t_max[n],t_min[n]);}
  gotoxy(5,3+n);
  cprintf("  ch[%d] = %d",*m,*ch);

    } /*end of if*/
} /*end of r_temp*/

```

```

/* receive keyboard value pid for some card and channel */
set_value_pid()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าในรูปแบบใดทั้งสิ้น ไม่ผิดแต่เพียงผู้เดียว และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char key1,key2,key6,key7;
char ya2[2];
int num;
int i,j,k,t,n,m,x,value;

```

```

int      key,ya1,book,key3,key4,key5;

for(m=0;m<=31;m++) {
    for(n=0;n<=7;n++)
        Kp[m][n] = Ti[m][n] = Td[m][n] =0;
}
clrscr();
menu2();
key1 = getch();
while((key1 != 'q')&&(key1 !='Q'))
{
    if((key1 == 'm')||(key1 == 'M'))
    {
        m = 0;
        clrscr();
        gotoxy(8,3);cprintf("what's the card (1-32).?");
do
{
    do
    {
        textcolor(RED);
        gotoxy(9,5);
        cprintf(" card.....");
        textcolor(YELLOW);
        gotoxy(9,20);
        gets(ya2);
        ya1 = atoi(ya2);
        card[m] = ya1;
        if((card[m]<=0)||((card[m]>32)))
        {
            gotoxy(9,5);
            delline();
            insline();
            gotoxy(10,6);
            cprintf("only card 1-32 select again");
        }
        else
        {
            gotoxy(10,6);
            delline();
        }
    }while((card[m]<=0)||((card[m]>32)));
    gotoxy(6,8);cprintf("-----");
    gotoxy(7,9); cprintf("(a,A) set all chananel");
    gotoxy(7,10); cprintf("(s,S) set some channel");
    key2 = getch();
    if((key2 == 'a')||(key2 == 'A'))          /*modified all channels*/
    {
        r_pid(Kp[card[m]-1],Ti[card[m]-1],Td[card[m]-1],&value,&c_h[card[m]-1],&car
        clrscr();
        gotoxy(8,3);cprintf("what's the card (1-32).?");
    }
    gotoxy(7,10);
    textcolor(RED+BLINK);
    cprintf("press * when success");
    textcolor(YELLOW);
    if((key2 == 's')||(key2 == 'S'))
    {
        clrscr();
        gotoxy(7,2);
        textcolor(RED);
        cprintf("enter modiflied channel");
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ขอสงวนสิทธิ์ในสิ่งที่ปรากฏ ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(10,10);
textcolor(RED+BLINK);
cprintf("press * when succes");
textcolor(YELLOW);
gotoxy(35,2);
key4 = getch();if(key4=='*');
gotoxy(7,2);
cprintf("enter modiflied channel");
gotoxy(10,10);delline();insline();
while(((key4>56)!;(key4<49))&&(key4!='*')) {
    gotoxy(35,2);
    key4 = getch();if(key4=='*');
}
c_h[card[m]-1] = 0;
while(key4!='*') {
key3 = key4-48;
value = 1;
    Kp[card[m]-1][key3-1] = Ti[card[m]-1][key3-1] = Td[card[m]-1][key3-1] =
    gotoxy(5,4);
    cprintf("channel#%d",key3);
    gotoxy(5,5);
    cprintf("enter new value pid Kp Ti Td");
    gotoxy(5,6);
    delline();
    insline();
    scanf("%d",&Kp[card[m]-1][key3-1]);
    gotoxy(10,6);
    scanf("%d",&Ti[card[m]-1][key3-1]);
    gotoxy(15,6);
    scanf("%d",&Td[card[m]-1][key3-1]);
    gotoxy(5,6);delline();insline();
    if(key3 ==1)
        c_h[card[m]-1] = value;
    else
    { value = 1;
      for(num =1; num<key3; num++)
        value *= 2;
      c_h[card[m]-1] += value;
    }
    /*end of else*/
clrscr();
gotoxy(7,2);
textcolor(RED);
cprintf("enter modiflied channel");
gotoxy(10,10);
textcolor(RED+BLINK);
cprintf("press * when succes");
textcolor(YELLOW);
gotoxy(35,2);
key4 = getch();if(key4=='*'); /* printf("\07"); */
gotoxy(7,2);
cprintf("enter modiflied channel");
gotoxy(10,10);delline();insline();
while(((key4>56)!;(key4<49))&&(key4!='*'))
{
    gotoxy(35,2);
    key4 = getch();
}
}
/*end of while*/
clrscr();
for (n = 0; n<8;n++)
{ gotoxy(2,2+n);
  cprintf("channel#%d %5d %5d %5d",n+1,Kp[card[m]-1][n],Ti[card[m]-1][n],T

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ gotoxy(2,2+n);
  cprintf("channel#%d %5d %5d %5d",n+1,Kp[card[m]-1][n],Ti[card[m]-1][n],T

```

```

gotoxy(5,3+n);
cprintf("  c_h[%2d] = %d",card[m],c_h[card[m]-1]);
m++;
gotoxy(4,10);
textcolor(RED+BLINK);
cprintf("press space bar");
textcolor(YELLOW);
do { key7 = getch();
    /* printf("\07"); */ } while(key7 != ' ');
clrscr();
gotoxy(6,3);cprintf("what's the card (1-32)..?");
gotoxy(7,10);
textcolor(RED+BLINK);
cprintf("press * when success");
textcolor(YELLOW);
} /*end of if*/
key6 = getch(); /* printf("\07"); */
gotoxy(7,10);delline();insline();
}while(key6 != '*');
}/*end of else*/
window(6,19,74,21);
gotoxy(4,2);
textcolor(RED+BLINK);
cprintf("press space bar to menu");
textcolor(YELLOW);
gotoxy(27,2);
key5 = getch(); /* printf("\07"); */
while(key5 != 0x20) {
gotoxy(27,2);
key5 = getch(); /* printf("\07"); */
}
clrscr();
window(34,6,74,16);
clrscr();
menu2();
key1 = getch();
} /*end of while*/
for(k=0;k<=3;k++) ucard[k]=0;
for(i=0;i<=31;i++) {
if(card[i]!=0)
{ book = 1;
if((card[i]>=1)&&(card[i]<=8)) {t = card[i]-1;k=0;}
if((card[i]>=9)&&(card[i]<=16)) {t = card[i]-9;k=1;}
if((card[i]>=17)&&(card[i]<=24)) {t = card[i]-17;k=2;}
if((card[i]>=25)&&(card[i]<=32)) {t = card[i]-25;k=3;}
if(t==0) book=1;
else for(j=1;j<=t;j++) book *=2;
ucard[k] += book;}
}
} /* end of setch() */

/*received setpoint max min for every channels*/
r_pid(Kp,Ti,Td,va,ch,m)
float Kp[],Ti[],Td[];
int *va,*m,*ch;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่จำกัดวงทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clrscr();
for(n=0;n<=7;n++)

```

```

Kp[n] = Ti[n] =Td[n] = 0;
n = 0; *ch = 0;
while(n<=7)
{
    gotoxy(5,4);
    cprintf("channel#%d   ",n+1);
    *va = 1;
    if(n == 0)
        *ch = 1;
    else
    {
        for(num =0; num <n; num++)
            *va *=2;
            *ch += *va;
        } /*end of else*/
        gotoxy(5,5);
        cprintf("enter setpoint max min");
        gotoxy(5,6);
        delline();
        insline();
        scanf("%d",&Kp[n]);
        gotoxy(10,6);
        scanf("%d",&Ti[n]);
        gotoxy(15,6);
        scanf("%d",&Td[n]);
        gotoxy(5,7);
        n = n++;
    } /*end of while*/
clrscr();
for (n = 0; n<8;n++)
{ gotoxy(2,2+n);
  cprintf("channel#%d   %5d   %5d   %5d",n+1,Kp[n],Ti[n],Td[n]);}
gotoxy(5,3+n);
cprintf("   ch[%d] = %d",*m,*ch);
gotoxy(5,10);
cprintf("press 'c' to improve");
key6 = getch(); /* printf("\07"); */
clrscr();
if((key6=='c')||!(key6=='C'))
{
    clrscr();
    gotoxy(7,2);
    textcolor(RED);
    cprintf("enter modifilied channel ");
    gotoxy(10,10);
    textcolor(RED+BLINK);
    cprintf("press * when succes");
    textcolor(YELLOW);
    gotoxy(35,2);
    key4 = getch(); /* printf("\07"); */
    gotoxy(7,2);cprintf("enter modifilied channel");
    gotoxy(10,10);delline();insline();
    while(key4!='*')
    {
        key3 = key4-48;
        while((key3 <=8)&&(key3 >0))
        {
            Kp[key3-1]=*Ti[key3-1]+Td[key3-1] *ไม่เอา;
            gotoxy(5,4);
            cprintf("channel#%d",key3);
            gotoxy(5,5);
            cprintf("enter new setpoint   max   min");

```

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(5,6);
delline();
insline();
scanf("%d",&Kp[key3-1]);
gotoxy(10,6);
scanf("%d",&Ti[key3-1]);
gotoxy(15,6);
scanf("%d",&Td[key3-1]);
cprintf("");
gotoxy(7,2);
textcolor(RED);
cprintf("enter modified channel ");
gotoxy(10,10);
gotoxy(35,2);
key4 = getch();
textattr(YELLOW);
key5 = key4 - 48;
/*end of while*/
/*end of while*/
clrscr();
for (n = 0; n<8;n++)
gotoxy(2,2+n);
printf("channel%d %5d %5d %5d\n",n+1,Kp[n],Ti[n],Td[n]);
gotoxy(5,3+n);
cprintf(" ch[%d] = %d",*m,*ch);
} /*end of if*/
/*end of r_temp*/
case(a1,b1,a2,b2)
int *a1,*b1,*a2,*b2;
{
int blockx,blocky;
int c1 = *a1+1; int d1 = *b1+1;
int c2 = *a2-1; int d2 = *b2-1;
clrscr();
window(c1,d1,c2,d2);
blockx = wherex();
blocky = wherex();
gotoxy(blockx,blocky);
}
#define attrhead 7
#define attrchar 14
#define head0 " FRONT END DATA ACQUISITION "
#define head1 " MAIN MENU "
#define head2 " SUB MENU "
#define head3 " Message "

windowbox(ax1,ay1,ax2,ay2,an)
int *ax1,*ay1,*ax2,*ay2,*an;
{
static int typeofbox[9] = {186,205,205,186,205,201,187,188,200};
int top = 2; int left = 3;
int right = 3; int bottom = 4;
int upleft = 5; int upright = 6;
int lolleft = 8; int loright = 7;
int *x,*y,*attrwin;
int dx1,dx2,dy1,dy2,dn;
char *headwindow;
dx1 = *ax1; dx2 = *ax2;
dy1 = *ay1; dy2 = *ay2;

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่สามารถแก้ไขหรือเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dn = %an;
if (dn==1)
    headwindow = head1;
else
{ if (dn==2) headwindow = head2;
  else
    { if(dn==3)headwindow = head3;
      else headwindow = head0; }
}
if(%an==0)attrwin=5;else attrwin=1;
textbackground(attrwin);
window(dx1,dyl,dx2,dy2);
clrscr();
window(1,1,80,25);
textattr(0+(5<<4));

/* top */
gotoxy(dx1,dyl);
cprintf("%c",typeofbox[upleft]);
for (x = dx1+1 ; x <= dx2-1 ; x++ )
cprintf("%c",typeofbox[top]);
cprintf("%c",typeofbox[upright]);
/* side */
for (y = dyl+1 ; y <= dy2-1 ; y++ )
{
    gotoxy(dx1,y);
    cprintf("%c",typeofbox[left]);
    gotoxy(dx2,y);
    cprintf("%c",typeofbox[right]);
}
/*bottom */
gotoxy(dx1,dy2);
cprintf("%c",typeofbox[loleft]);
for (x = dx1+1 ; x <= dx2-1 ; x++ )
cprintf("%c",typeofbox[bottom]);
cprintf("%c",typeofbox[loright]);
if(%an==0)textattr(RED+(CYAN<<4));else textattr(attrhead);
gotoxy((dx1+dx2-strlen(*headwindow))/2,dyl);
cprintf("%s",headwindow);
window(dx1+1,dy1+1,dx2-1,dy2-1);
textbackground(attrwin);
textcolor(attrchar);
}
menu(y)
int *y;
{
    int row =2,col = 3;
    char *msg[] = { " SELECT CHANNEL ",
                   " LOAD PID          ",
                   " LOAD DATA          ",
                   " PRINT GRAPH         ",
                   " START               ",
                   " STOP                ",
                   " EXIT                ", };
    int i,rowl,ch;
    *y = 0;
    clrscr();
    rowl = row;
    for(i=0;i<7;i++)
    {
        gotoxy(col,rowl);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าในรูปแบบใดก็ตามผู้ถือลิขสิทธิ์ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    gotoxy(col,rowl);

```

```

    cputs(msg[i]);
    row1++;
}
cursor(6,7);
textcolor(4);
cputs(msg[*y]);
while((ch = getch()) != '\r')
{
    if(ch==0x48)
    {
        if(*y==0) row=row;
        else { row = row-1;
              *y = *y-1;
              gotoxy(col,row+1);
              textcolor(YELLOW);
              cputs(msg[*y+1]); }
    }
    if(ch==0x50)
    {
        if(*y==6) row = row;
        else { row = row + 1;
              *y = *y+1;
              gotoxy(col,row-1);
              textcolor(YELLOW);
              cputs(msg[*y-1]); }
    }
    gotoxy(col,row);
    textcolor(4);
    cputs(msg[*y]);
}
/* printf("\07"); */
}
cursor(int row,int col)
{
    _AH = 0x02; _BH = 0x00; _DH = row; _DL = col;
    geninterrupt(0x10);
    _AH = 0x01; _CH = 11; _CL = 13;
    geninterrupt(0x10);
}

menu1()
{
    gotoxy(5,2); cprintf("select channels for temperature");
    gotoxy(3,3); cprintf("-----");
    gotoxy(7,5); cprintf("(m,M) modified ch&temp");
    gotoxy(7,7); cprintf("(q,Q) quit");
    gotoxy(3,9); cprintf("-----");
}

menu2()
{
    gotoxy(6,2); cprintf("select channels for value pid");
    gotoxy(3,3); cprintf("-----");
    gotoxy(7,5); cprintf("(m,M) modified ch & pid");
    gotoxy(7,7); cprintf("(q,Q) quit");
    gotoxy(3,9); cprintf("-----");
}

/* PID program */
float kp,ki,kd;
pid()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ใบความรู้ฉบับนี้จัดทำขึ้นโดยผู้จัดทำทั้งหมดมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pid()

{

```

FILE *fp;
char ch[20],chaf;
union {
    char c[2];
    long unsigned count;
} cnt;
port_init();
clrscr();
gotoxy(3,3);
cprintf("enter value of Kp..");
scanf("%f",&kp);
gotoxy(3,4);
cprintf("enter value of Ki..");
scanf("%f",&ki);
gotoxy(3,5);
cprintf("enter value of Kd..");
scanf("%f",&kd);
window(1,1,60,25);
clrscr();
system("dir/w a:*.");
gotoxy(2,16);
cprintf(" print PID PROCESS name...");
scanf("%s",ch);
if((fp=fopen(ch,"rb"))==0)
{
    cprintf("cannot open PID file");
    exit(1);
}

/* round out the size of the file */
fseek(fp,0,2);
cnt.count = ftell(fp);
cprintf("%d",cnt.count);

/* send size */
sport('?');
/* wait();*/
sport(cnt.c[0]);
/* waitb(); */
sport(cnt.c[1]);
fseek(fp,0,0);
do {
    char = getc(fp);
    if(ferror(fp)) {
        cprintf(" error reading input file ");
        break;
    }

    /* wait until reciever is ready */
    if(!feof(fp)) {
        /* waitb(); */
        sport(chaf);
        cprintf("%c",chaf);
    }
} while(!feof(fp));
/* waitb();*/
fclose(fp);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* initial serial port
    8 data bits , 2 stop bits, no parity bits */

```

```

port_init()
{
    int port = ser_p;
    int sel;
    unsigned int code;
    union REGS r;
    int co=1;
    clrscr();
    while(co){
        gotoxy(2,2);
        cprintf(" select your baud rate...");
        gotoxy(2,3);
        cprintf(" 9600  4800  2400  1200 ");
        gotoxy(3,5);
        scanf("%d",&sel);
        switch(sel)
        {
            case 9600:code = 0xe3;co=0;break;
            case 4800:code = 0xc3;co=0;break;
            case 2400:code = 0xa3;co=0;break;
            case 1200:code = 0x83;co=0;break;
            default:co=1;gotoxy(3,7);
                cprintf("it can't set this baud rate");
                return port_init();
        }
    }
    /* end of while*/
    r.x.dx = port; /* serial port */
    r.h.ah = 0; /* initialize port function */
    r.h.al = code; /* initialization code - see text for details */
    int86(0x14,&r,&r);
}

/* check the status of the serial port */
check_stat()
{
    int port = ser_p;
    union REGS r;
    r.x.dx = port; /* serial port */
    r.h.ah = 3; /* read status */
    int86(0x14, &r, &r);
    return r.x.ax;
}

/* send a character out the serial port */
sport(c)
char c; /* character to send */
{
    int port = ser_p;
    union REGS r;

    r.x.dx = port; /* serial port */
    r.h.al = c; /* char to send*/
    r.h.ah = 1; /* send character function */
    int86(0x14, &r, &r);
    /* gotoxy(2,6); */
    printf(" sendd %x ",c);
    if(r.h.ah & 128) {
        gotoxy(2,7);
        printf("send error detected in serial port ");
        exit(1);
    }
}
}

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตก่อนใช้เพื่อการเรียนการสอน

```

/* wait for a response */
wait()
{
    int co = 10;
    do { co--;
        rport(); }
        while (( R != '.')&&(co != 0));
        if (( R != '.')&&(co == 0))
        {
            printf(" receiver (.) error");
            exit(1);
        }
    }
}

/* wait for byte */
waitb()
{
    int co = 10;
    do { co--;
        rport(); }
        while (( R != '&')&&(co != 0));
        if (( R != '&')&&(co == 0))
        {
            printf(" receiver (&) error");
            exit(1);
        }
    }
}

/* wait for receive*/
waitf()
{
    int co=10;
    do { co--;
        rport(); }
        while (( R != '?')&&(co != 0));
        if (( R != '?')&&(co == 0))
        {
            printf(" receiver (?) error");
            exit(1);
        }
    }
}

wait4()
{
    int co = 10;
    do { co--;
        rport();
        } while (( R != '#')&&(co != 0));
        if (( R != '#')&&(co == 0))
        {
            printf(" receiver (#) error ");
            exit(1);
        }
    }
}

/* read a character from a port */
rport()
{
    int port = ser_p;
    union REGS r;
    while(!(check_stat()&256))
        if ( kbhit() )

```

เอกสารนี้เป็นของกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์ ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากกรมส่งเสริมการค้าระหว่างประเทศ

ไม่ว่ากรณีใดๆก็ตาม กรมส่งเสริมการค้าระหว่างประเทศ ขอสงวนสิทธิ์ในข้อมูลและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* wait for a character */
while(!(check_stat()&256))
    if ( kbhit() )

```

```

    {
        getch();
        exit(1);
    }
    r.x.dx = port; /* serial port */
    r.h.sh = 2; /* read character function */
    int86(0x14, &r, &r);
    R = r.h.al;
    /* gotoxy(2,7); */
    printf(" received %x ",R);
    if(r.h.sh & 128) {
        gotoxy(2,8);
        printf("read error detected in serial port ");
        exit(1);
    }
}
/* pid init and serial port init */
float kp,kd,ki;
pidinit()
{
    port init();
    clrscr();
    gotoxy(3,3);
    cprintf("enter value of Kp..");
    scanf("%f",&kp);
    gotoxy(3,4);
    cprintf("enter value of Ki..");
    scanf("%f",&ki);
    gotoxy(3,5);
    cprintf("enter value of Kd..");
    scanf("%f",&kd);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <alloc.h>
```

```
#define max_dot 700
#define PI 3.14159
#define step 20
#define Esc '\x1B'
#define Enter '\x0D'
#define F3_key '\x3D'
#define spcl '\x0'
#define true 1
#define false 0
```

```
int col1_gph1, col2_gph1;
int col1_gph2, col2_gph2;
int col1_gph3, col2_gph3;
int col1_gph4, col2_gph4;
int col1_bar1, col2_bar1;
int col1_bar2, col2_bar2;
int col1_bar3, col2_bar3;
int col1_bar4, col2_bar4;
int col1_mtr2, col2_mtr2;
int col1_mtr4, col2_mtr4;
float row1_gph1, row2_gph1;
float row1_gph2, row2_gph2;
float row1_gph3, row2_gph3;
float row1_gph4, row2_gph4;
float row1_box1, row2_box1;
float row1_box2, row2_box2;
float row1_box3, row2_box3;
float row1_box4, row2_box4;
float row1_rng1, row2_rng1;
float row1_rng2, row2_rng2;
float row1_rng3, row2_rng3;
float row1_rng4, row2_rng4;
float set_runrange[8];
float set_setscale[8];
float set_setpoint[8];
float ch1;
float ch2;
float ch3;
float ch4;
int col_pix;
int row_pix;
int num1_exp;
int num2_exp;
int graphdriver;
int str_lng = 0;
int expand = false;
int ch_slct = false;
int enter1 = false;
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์ หากต้องการนำเอกสารนี้ไปใช้ กรุณาติดต่อเจ้าของลิขสิทธิ์เพื่อขอ  
อนุญาตก่อนทุกครั้ง

```
char check_key();
```

```

void write_ch();
void data_ch1(float ,float *,int);
void data_ch2(float ,float *,int);
void data_ch3(float ,float *,int);
void data_ch4(float ,float *,int);
void draw_ch1(int ,float *);
void draw_ch2(int ,float *);
void draw_ch3(int ,float *);
void draw_ch4(int ,float *);
void meter(float ,float ,int ,int ,int );
void bar_slice(float ,int ,int ,int);
void dot_plot(float *,int ,int ,int ,int ,int );
void box_line(int ,int ,int);
void warning(int);
void box_del(int,int,int,int);

float find_ch1 (float ); /* set range of data in 0 to 100 */
float find_ch2 (float ); /* "-----"*/
float find_ch3 (float ); /* "-----"*/
float find_ch4 (float ); /* "-----"*/

void find_mean(float ,float ,float ,float );
char channel4[10];
static int s_ch1,s_ch2,s_ch3,s_ch4,chu,grad;
/* extern float data[3][256]; */
graph (gra,sel_ch1,sel_ch2,sel_ch3,sel_ch4,chud)
int gra,sel_ch1,sel_ch2,sel_ch3,sel_ch4,chud;
{
    int j;
    if(gra)
    {
        clrscr();
        cprintf("please input file name\n");
        gotoxy(1,3);
        cscanf("%s",channel4);
    }

    s_ch1 = sel_ch1; s_ch2 = sel_ch2;
    s_ch3 = sel_ch3; s_ch4 = sel_ch4;
    grad = gra;
    chu = chud;
    detect_monit();
    bound_ch1();
    bound_ch2();
    bound_ch3();
    bound_ch4();
    four_window();
    display_window();
    getch();
    closegraph();
}

detect_monit()
{
    int graphmode;
    graphdriver = DETECT;
    initgraph(&graphdriver,&graphmode," ");
    if (graphdriver == EGA)
        col_pix = 639;
        row_pix = 349;
    }
    else /* for Hercmonohi only */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ขอสงวนสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else /* for Hercmonohi only */

```

```
{
    col_pix = 719;
    row_pix = 347;
}
```

```
bound_ch1()
```

```
    col1_bar1 = 20;
    col2_bar1 = 40;
    row1_box1 = 10;
    row2_box1 = 150;
    row1_gph1 = set_setscale[1] = 0;
    row2_gph1 = set_setscale[0] = 25;
    row1_rng1 = set_runrange[1] = 15;
    row2_rng1 = set_runrange[0] = 20;
    if (graphdriver == EGA)
    {
        col1_gph1 = 90;
        col2_gph1 = 300;
    }
    else /* for Hercules only */
    {
        col1_gph1 = 90;
        col2_gph1 = 340;
    }
}
```

```
bound_ch2()
```

```
{
    row1_gph2 = set_setscale[3] = 0;
    row2_gph2 = set_setscale[2] = 110;
    row1_rng2 = set_runrange[3] = 60;
    row2_rng2 = set_runrange[2] = 80;
    row1_box2 = 60;
    row2_box2 = 150;
    if (graphdriver == EGA)
    {
        col1_gph2 = 370;
        col2_gph2 = 620;
        col1_mtr2 = 370;
        col2_mtr2 = 480;
    }
    else
    {
        col1_gph2 = 410;
        col2_gph2 = 700;
        col1_mtr2 = 410;
        col2_mtr2 = 560;
    }
}
```

```
bound_ch3()
```

```
{
    col1_bar3 = 20;
    col2_bar3 = 40;
    row1_box3 = 185;
    row2_box3 = 330;
    row1_gph3 = set_setscale[5] = 0;
    row2_gph3 = set_setscale[4] = 1000;
    row1_rng3 = set_runrange[5] = 700;
```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

row2_rng3 = set_runrange[4] = 900;
if (graphdriver == EGA)
{
    col1_gph3 = 90;
    col2_gph3 = 300;
}
else
{
    col1_gph3 = 90;
    col2_gph3 = 340;
}
}

```

```

bound_ch4()
{
    row1_gph4 = set_setscale[7] = 0;
    row2_gph4 = set_setscale[6] = 100;
    row1_rng4 = set_runrange[7] = 70;
    row2_rng4 = set_runrange[6] = 90;
    row1_box4 = 240;
    row2_box4 = 330;
    if (graphdriver == EGA)
    {
        col1_gph4 = 370;
        col2_gph4 = 620;
        col1_mtr4 = 370;
        col2_mtr4 = 480;
    }
    else
    {
        col1_gph4 = 410;
        col2_gph4 = 700;
        col1_mtr4 = 410;
        col2_mtr4 = 560;
    }
}

```

```

four_window(
{
    rectangle(0,0,col_pix,row_pix);
    line(col_pix/2,0,col_pix/2,row_pix);
    line(0,row_pix/2,col_pix,row_pix/2);
}

```

```

display_window()
{
    int    number_dot;
    float  win_lev1[max_dot];
    float  win_lev2[max_dot];
    float  win_lev3[max_dot];
    float  win_lev4[max_dot];
    float  out_ch1;
    float  out_ch2;
    float  out_ch3;
    float  out_ch4;

```

```

    char   ch_key;

```

เอกสาร FILE เอกสาร \*fp; วนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆก็ตาม ห้ามแก้ไขหรือดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    number_dot = 1;

```

```

    ch_key = ' ';

```

```

    if ( (fp = fopen("channel4","r")) != NULL)

```





```

        if (graphdriver == EGA) x1_loc = 520;
        else x1_loc = 600;
        box_del(x1_loc,30,x1_loc+110,50);
        expand = false;
        ch_slct = false;
        enter1 = false;
        enter2 = false;
        for (i = 0; i<=6; i++) str_exp[i] = '';
        str_lng = 0;
    }
}
return(ch_key);
}

void draw_ch1(int num_dot,float level_ch[])
{
    int max_arr;
    if (num_dot >= (col2_gph1-col1_gph1))
    {
        max_arr = col2_gph1-col1_gph1;
        bar_slice(level_ch[max_arr],col1_bar1,col2_bar1,1);
        dot_plot(level_ch,num_dot,BLACK,col1_gph1,col2_gph1,1);
    }
    else
    {
        bar_slice(level_ch[num_dot],col1_bar1,col2_bar1,1);
        dot_plot(level_ch,num_dot,BLUE,col1_gph1,col2_gph1,1);
    }
    box_line(col1_gph1,col2_gph1,1);
    setcolor(WHITE);
}

void draw_ch2(int num_dot,float level_ch[])
{
    int max_arr;
    if (num_dot >= (col2_gph2-col1_gph2))
    {
        max_arr = col2_gph2-col1_gph2;
        meter(level_ch[max_arr],level_ch[max_arr-1],col1_mtr2,col2_mtr2,2);
        dot_plot(level_ch,num_dot,BLACK,col1_gph2,col2_gph2,2);
    }
    else
    {
        meter(level_ch[num_dot],level_ch[num_dot-1],col1_mtr2,col2_mtr2,2);
        dot_plot(level_ch,num_dot,BLUE,col1_gph2,col2_gph2,2);
    }
    box_line(col1_gph2,col2_gph2,2);
    setcolor(WHITE);
}

void draw_ch3(int num_dot,float level_ch[])
{
    int max_arr;
    if (num_dot >= (col2_gph3-col1_gph3))
    {
        max_arr = col2_gph3-col1_gph3;
        bar_slice(level_ch[max_arr],col1_bar3,col2_bar3,3);
        dot_plot(level_ch,num_dot,BLACK,col1_gph3,col2_gph3,3);
    }
    else
    {
        bar_slice(level_ch[num_dot],col1_bar3,col2_bar3,3);
        dot_plot(level_ch,num_dot,BLUE,col1_gph3,col2_gph3,3);
    }
    box_line(col1_gph3,col2_gph3,3);
    setcolor(WHITE);
}

```

เอกสารนี้เป็น if (num\_dot >= (col2\_gph3-col1\_gph3)) นั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่ข้อมูลใดๆถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
else
{
    bar_slice(level_ch[num_dot], col1_bar3, col2_bar3, 3);
    dot_plot(level_ch, num_dot, BLUE, col1_gph3, col2_gph3, 3);
}
box_line(col1_gph3, col2_gph3, 3);
setcolor(WHITE);
}

```

```
void draw_ch4(int num_dot, float level_ch[])
```

```

{
    int max_arr;
    if (num_dot >= (col2_gph4 - col1_gph4))
    {
        max_arr = col2_gph4 - col1_gph4;
        meter(level_ch[max_arr], level_ch[max_arr - 1], col1_mtr4, col2_mtr4,
        dot_plot(level_ch, num_dot, BLACK, col1_gph4, col2_gph4, 4);
    }
else
{
    meter(level_ch[num_dot], level_ch[num_dot - 1], col1_mtr4, col2_mtr4, 4);
    dot_plot(level_ch, num_dot, BLUE, col1_gph4, col2_gph4, 4);
}
box_line(col1_gph4, col2_gph4, 4);
setcolor(WHITE);
}

```

```
void data_ch1(float ch1, float *chnel_1, int dot_ch)
```

```

{
    int num;
    if (dot_ch <= (col2_gph1 - col1_gph1)) chnel_1[dot_ch] = ch1;
else
{
    for (num = 1; num < (col2_gph1 - col1_gph1); num++)
        chnel_1[num] = chnel_1[num + 1];
    chnel_1[col2_gph1 - col1_gph1] = ch1;
}
}

```

```
void data_ch2(float ch2, float *chnel_2, int dot_ch)
```

```

{
    int num;
    if (dot_ch <= (col2_gph2 - col1_gph2)) chnel_2[dot_ch] = ch2;
else
{
    for (num = 1; num < (col2_gph2 - col1_gph2); num++)
        chnel_2[num] = chnel_2[num + 1];
    chnel_2[col2_gph2 - col1_gph2] = ch2;
}
}

```

```
void data_ch3(float ch3, float *chnel_3, int dot_ch)
```

```

{
    int num;
    if (dot_ch <= (col2_gph3 - col1_gph3)) chnel_3[dot_ch] = ch3;
else
{
    for (num = 1; num < (col2_gph3 - col1_gph3); num++)
        chnel_3[num] = chnel_3[num + 1];
}
}

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        chnel_3[col2_gph3-col1_gph3] = ch3;
    }
}

void data_ch4(float ch4,float *chnel_4,int dot_ch)
{
    int num;
    if (dot_ch <= (col2_gph4-col1_gph4)) chnel_4[dot_ch] = ch4;
    else
    {
        for (num =1; num < (col2_gph4-col1_gph4); num++)
            chnel_4[num] = chnel_4[num+1];
        chnel_4[col2_gph4-col1_gph4] = ch4;
    }
}

void meter(float level_mtr,float level_old,int col1_mtr,int col2_mtr,int channel)
{
    int center_col;
    int width;
    float center_row;
    float row1_metr;
    float row2_metr;
    float row1_mtr;
    float row2_mtr;
    float row1_var;
    float row2_var;
    float long_row;
    float angle;
    float radius;
    float row_val;
    float x_mtr;
    float y_mtr;
    float x_plot;
    float y_plot;
    float xold_plot;
    float yold_plot;
    float ratio1;
    float ratio2;

    switch (channel)
    {
        case 1 : row1_mtr = row1_box1; row2_mtr = row2_box1;
                 row1_var = row1_gph1; row2_var = row2_gph1;
        case 2 : row1_mtr = row1_box2; row2_mtr = row2_box2;
                 row1_var = row1_gph2; row2_var = row2_gph2;
        case 3 : row1_mtr = row1_box3; row2_mtr = row2_box3;
                 row1_var = row1_gph3; row2_var = row2_gph3;
        case 4 : row1_mtr = row1_box4; row2_mtr = row2_box4;
                 row1_var = row1_gph4; row2_var = row2_gph4;
    }

    long_row = (row2_mtr-row1_mtr);
    row1_metr = row2_mtr-long_row-53;
    row2_metr = row2_mtr-long_row-3;
    rectangle(col1_mtr,row1_metr,col2_mtr,row2_metr);
    radius = 2*(row2_mtr-row1_mtr)/PI;
    center_col = (col2_mtr-col1_mtr)/2+col1_mtr;
    center_row = row2_metr;
    ratio1 = 0.82;
    setcolor(BLUE);
    for (width = 0; width <= 3; width++)
        arc(center_col,center_row,45,134,radius-width);
}

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า หากท่านมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(WHITE);
ratio2      = (row2_var-row1_var)/(row2_mtr-row1_mtr);
row_val     = (level_old-row1_var)/ratio2+row1_mtr;
angle       = (row_val-row1_mtr)*PI/(2*long_row);
if (angle < 0) angle = 0;
if (angle > PI/2) angle = PI/2;
x_mtr      = radius*cos(angle+PI/4);
y_mtr      = radius*sin(angle+PI/4);
xold_plot  = center_col-ratio1*x_mtr;
yold_plot  = center_row-ratio1*y_mtr;
setcolor(BLACK);
line(center_col,center_row,xold_plot,yold_plot);
row_val     = (level_mtr-row1_var)/ratio2+row1_mtr;
angle       = (row_val-row1_mtr)*PI/(2*long_row);
if (angle < 0) angle = 0;
if (angle > PI/2) angle = PI/2;
x_mtr      = radius*cos(angle+PI/4);
y_mtr      = radius*sin(angle+PI/4);
x_plot     = center_col-ratio1*x_mtr;
y_plot     = center_row-ratio1*y_mtr;
setcolor(RED);
line(center_col,center_row,x_plot,y_plot);
setcolor(WHITE);
xold_plot  = x_plot;
yold_plot  = y_plot;
}

```

```
void bar_slice(float level_bar,int col1_bar,int col2_bar,int channel)
```

```

{
    int    color_bar;
    float row1_bar;
    float row2_bar;
    float row1_var;
    float row2_var;
    float run1_rng;
    float run2_rng;
    float lne1_rng;
    float lne2_rng;
    float line1;
    float line2;
    float row_val;
    float ratio;

    switch (channel)
    {
        case 1 : row1_bar = row1_box1; row2_bar = row2_box1;
                  row1_var = row1_gph1; row2_var = row2_gph1;
                  run1_rng = set_runrange[1];
                  run2_rng = set_runrange[0]; break;
        case 2 : row1_bar = row1_box2; row2_bar = row2_box2;
                  row1_var = row1_gph2; row2_var = row2_gph2;
                  run1_rng = set_runrange[2];
                  run2_rng = set_runrange[3]; break;
        case 3 : row1_bar = row1_box3; row2_bar = row2_box3;
                  row1_var = row1_gph3; row2_var = row2_gph3;
                  run1_rng = set_runrange[5];
                  run2_rng = set_runrange[4]; break;
        case 4 : row1_bar = row1_box4; row2_bar = row2_box4;
                  row1_var = row1_gph4; row2_var = row2_gph4;
    }

    setcolor(WHITE);
    rectangle(col1_bar-1,row1_bar-1,col2_bar+1,row2_bar+1);
}

```

เอกสารนี้เป็นเอกสารที่สแกนขึ้นโดยอัตโนมัติ หากพบข้อผิดพลาดประการใด กรุณาแจ้งให้เราทราบ  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(LIGHTRED);
ratio      = (row2_var+row1_var)/(row2_bar-row1_bar);
line1     = (run1_rng-row1_var)/ratio+row1_bar;
lne1_rng  = row2_bar-line1+row1_bar;
if ((lne1_rng < row1_bar) || (lne1_rng > row2_bar)) lne1_rng = row1_bar;
line2     = (run2_rng-row1_var)/ratio+row1_bar;
lne2_rng  = row2_bar-line2+row1_bar;
if ((lne2_rng < row1_bar) || (lne2_rng > row2_bar)) lne2_rng = row2_bar;
line(col1_bar,lne1_rng,col2_bar,lne1_rng);
line(col1_bar,lne2_rng,col2_bar,lne2_rng);
row_val   = (level_bar-row1_var)/ratio+row1_bar;
level_bar = row2_bar-row_val+row1_bar;
if (level_bar <= row1_bar) level_bar = row1_bar;
if (level_bar >= row2_bar) level_bar = row2_bar;
setfillstyle(EMPTY_FILL,BLUE);
bar(col1_bar,row1_bar,col2_bar,row2_bar);
line(col1_bar,lne1_rng,col2_bar,lne1_rng);
line(col1_bar,lne2_rng,col2_bar,lne2_rng);
if ((level_bar < run1_rng) || (level_bar > run2_rng)) setfillstyle(SOLID_
else setfillstyle(SOLID_FILL,BLUE);
bar(col1_bar,row2_bar,col2_bar,level_bar);
} /*end bar_slice*/

```

```

void dot_plot(float level_plot[],int number_plot,int color_dot,int col1_plt,int c

```

```

(
int start;
int plot;
int width_col;
int width_rest;
float row_line;
float row1_plt;
float row2_plt;
float row1_var;
float row2_var;
float run1_rng;
float run2_rng;
float ratio;
float row_val;

start = 1;
switch (channel)
{
case 1 : row1_plt = row1_box1; row2_plt = row2_box1;
row1_var = row1_gph1; row2_var = row2_gph1;
run1_rng = set_runrange[1];
run2_rng = set_runrange[0]; break;
case 2 : row1_plt = row1_box2; row2_plt = row2_box2;
row1_var = row1_gph2; row2_var = row2_gph2;
run1_rng = set_runrange[3];
run2_rng = set_runrange[2]; break;
case 3 : row1_plt = row1_box3; row2_plt = row2_box3;
row1_var = row1_gph3; row2_var = row2_gph3;
run1_rng = set_runrange[5];
run2_rng = set_runrange[4]; break;
case 4 : row1_plt = row1_box4; row2_plt = row2_box4;
row1_var = row1_gph4; row2_var = row2_gph4;
run1_rng = set_runrange[7];
run2_rng = set_runrange[6]; break;
}

```

เอกสารนี้เป็นเอกสารที่ case 4 หรือ row1\_plt = row1\_box4; row2\_plt = row2\_box4; เป็นการคำนวณค่า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และลิขสิทธิ์ของเอกสารนี้ไปใช้

```

width_col = col2_plt-col1_plt;
if (number_plot > width_col) width_rest = width_col;
else width_rest = number_plot;
for (plot = 1; plot <= width_rest; plot++)
{
    if (color_dot == BLACK)
    {
        ratio      = (row2_var-row1_var)/(row2_plt-row1_plt);
        row_val    = (level_plot[start-1]-row1_var)/ratio+row1_plt;
        _row_line  = row2_plt-row_val+row1_plt;
        if (_row_line <= row1_plt) _row_line = row1_plt;
        if (_row_line >= row2_plt) _row_line = row2_plt;
        putpixel(plot+col1_plt, _row_line,color_dot);
    }
    ratio      = (row2_var-row1_var)/(row2_plt-row1_plt);
    row_val    = (level_plot[start]-row1_var)/ratio+row1_plt;
    _row_line  = row2_plt-row_val+row1_plt;
    if (_row_line <= row1_plt) _row_line = row1_plt;
    if (_row_line >= row2_plt) _row_line = row2_plt;
    putpixel(plot+col1_plt, _row_line, BLUE);
    start = start+1;
}
if ((level_plot[start-1] > run2_rng) ||
    (level_plot[start-1] < run1_rng)) warning(channel);
else box_del(col1_plt, row2_plt+4, col1_plt+60, row2_plt+13)
} /* end dot_plot */

void warning(int channel)
{
    int col1_wrn;
    int row2_wrn;
    switch(channel)
    {
        case 1 : col1_wrn = col1_gph1; row2_wrn = row2_box1; break;
        case 2 : col1_wrn = col1_gph2; row2_wrn = row2_box2; break;
        case 3 : col1_wrn = col1_gph3; row2_wrn = row2_box3; break;
        case 4 : col1_wrn = col1_gph4; row2_wrn = row2_box4; break;
    }
    outtextxy(col1_wrn, row2_wrn+5, "Warning");
}

void box_del(int col1_del, int row1_del, int col2_del, int row2_del)
{
    setfillstyle(EMPTY_FILL, BLACK);
    bar(col1_del, row1_del, col2_del, row2_del);
}

void box_line(int col1_ch, int col2_ch, int channel)
{
    int center_row;
    int center_var;
    int row_line;
    int row_lvar;
    int step_var;
    int run1_box;
    int run2_box;
    int run3_box;
    int _row_line;
    int col, row;
    int number;
    float run1_rng;

```

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่มีการฉ้อโกงในสื่อสิ่งพิมพ์ให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float run2_rng;
float row1_ch;
float row2_ch;
float row1_var;
float row2_var;
float row_val;
float set_pnt;
float ratio;

switch (channel)
{
    case 1 : row1_ch = row1_box1; row2_ch = row2_box1;
            row1_var = row1_gph1; row2_var = row2_gph1;
            run1_rng = row1_rng1; run2_rng = row2_rng1;
            set_pnt = set_setpoint[0] = 18; break;
    case 2 : row1_ch = row1_box2; row2_ch = row2_box2;
            row1_var = row1_gph2; row2_var = row2_gph2;
            run1_rng = row1_rng2; run2_rng = row2_rng2;
            set_pnt = set_setpoint[1] = 77; break;
    case 3 : row1_ch = row1_box3; row2_ch = row2_box3;
            row1_var = row1_gph3; row2_var = row2_gph3;
            run1_rng = row1_rng3; run2_rng = row2_rng3;
            set_pnt = set_setpoint[2] = 885; break;
    case 4 : row1_ch = row1_box4; row2_ch = row2_box4;
            row1_var = row1_gph4; row2_var = row2_gph4;
            run1_rng = row1_rng4; run2_rng = row2_rng4;
            set_pnt = set_setpoint[3] = 85; break;
}

setcolor(WHITE);
rectangle(col1_ch,row1_ch,col2_ch,row2_ch);
center_row = (row2_ch-row1_ch)/2 + row1_ch;
setcolor(LIGHTBLUE);
line(col1_ch,center_row,col2_ch,center_row);
setcolor(RED);
setlinestyle(DASHED_LINE,0x3333,NORM_WIDTH);
ratio = (row2_var-row1_var)/(row2_ch-row1_ch);
row_val = (run1_rng-row1_var)/ratio+row1_ch;
run1_box = row2_ch-row_val+row1_ch;
if ((run1_box < row1_ch) || (run1_box > row2_ch)) run1_box = row1_ch;
line(col1_ch,run1_box,col2_ch,run1_box);
row_val = (run2_rng-row1_var)/ratio+row1_ch;
run2_box = row2_ch-row_val+row1_ch;
if ((run2_box < row1_ch) || (run2_box > row2_ch)) run2_box = row2_ch;
line(col1_ch,run2_box,col2_ch,run2_box);
row_val = (set_pnt-row1_var)/ratio+row1_ch;
run3_box = row2_ch-row_val+row1_ch;
if ((run3_box < row1_ch) || (run3_box > row2_ch)) run3_box = row2_ch;
line(col1_ch,run3_box,col2_ch,run3_box);
setlinestyle(SOLID_LINE,0xFFFF,NORM_WIDTH);
row_line = center_row;
center_var = (row2_var-row1_var)/2+row1_var;
row_lvar = center_var;
number = (row2_ch-row1_ch)/step;
step_var = (row2_var-row1_var)/number;
col = col1_ch-40;
row = center_var;
setcolor(WHITE);
do
{
    row_line = row_line-step;
    row_lvar = row_lvar-step_var;
    line(col1_ch,row_line,col2_ch,row_line);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆก็ตาม ผู้ใช้ต้องแจ้งไปยังฝ่ายที่ดูแลเรื่องนี้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        _row_line = row2_var+row1_var-row_lvar;
        row      = row_line-4;
        gprintf(&col,&row,"%d",_row_line);
    } while ((row_line-step) > row1_ch);
    row      = center_row-4;
    gprintf(&col,&row,"%d",center_var);
    row_line = center_row;
    row_lvar  = center_var;
    row      = center_var;
do
    {
        row_line = row_line+step;
        row_lvar = row_lvar+step_var;
        line(col1_ch,row_line,col2_ch,row_line);
        _row_line = row2_var+row1_var-row_lvar;
        row      = row_line-4;
        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        gprintf(&col,&row,"%d",_row_line);
    } while ((row_line+step) < row2_ch);
} /* end box_line */

/*
/*      GPRINTF: Used like PRINTF except the output is sent to the
/*      screen in graphics mode at the specified co-ordinate.
/*
/*
int gprintf( int *xloc, int *yloc, char *fmt, ... )
{
    va_list argptr;          /* Argument list pointer      */
    char str[140];          /* Buffer to build sting into */
    int cnt;                /* Result of SPRINTF for return */

    va_start( argptr, format );          /* Initialize va_ functions */

    cnt = vsprintf( str, fmt, argptr );  /* prints string to buffer   */
    outtextxy( *xloc, *yloc, str );     /* Send string in graphics mode */
    *yloc += textheight( "H" ) + 2;     /* Advance to next line     */

    va_end( argptr );                /* Close va_ functions      */

    return( cnt );                    /* Return the conversion count */
}

/*void find_mean(float ch1,float ch2,float ch3,float ch4)
{
    sum1 += ch1;
    sum2 += ch2;
    sum3 += ch3;
    sum4 += ch4;
    mean1 = sum1/count;
    mean2 = sum2/count;
    mean3 = sum3/count;
    mean4 = sum4/count;
    dev1 = ch1 - mean1;
    dev2 = ch2 - mean2;
    dev3 = ch3 - mean3;
    dev4 = ch4 - mean4;
    count++;
}
*/

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆก็ตาม ห้ามนำเนื้อหาในเอกสารนี้ไปเผยแพร่ในที่สาธารณะหรือทำซ้ำโดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float find_ch1 ( float ch1)
{
    ch1 += 20;
    return( ch1);
}

float find_ch2 ( float ch2)
{
    ch2 += 65;
    return ( ch2);
}

float find_ch3 ( float ch3)
{
    ch3 += 198;
    return ( ch3);
}

float find_ch4 ( float ch4)
{
    ch4 += 245;
    return ( ch4);
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

โครงการนี้สามารถประสบผลสำเร็จได้ ก็เนื่องจากได้รับคำปรึกษา  
รวมถึงอุปการะต่างๆจาก อาจารย์ ขนิษฐา แซ่ตั้ง อาจารย์ที่ปรึกษา  
จึงขอขอบคุณมา ณ ที่นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หนังสืออ้างอิง

- 1) IBM, "TECHNICAL REFERENCE", PP.(1-215) - (1-243),  
REVISED EDITION, 1983
- 2) INTEL, "MICROCONTROLLER HANDBOOK", PP.(7-1) -  
(11-8), 1985
- 3) HERBERT SCHILDT, "C: POWER USER'S GUIDE", OSBORNE  
MCGRAW-HILL, PP. 207-208

