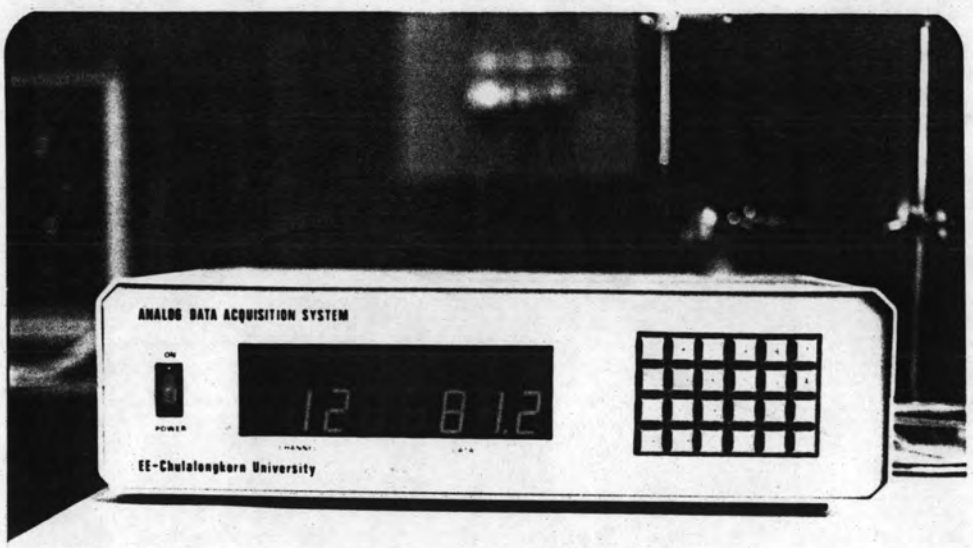
บทที่ 4

การออกแบบโปรแกรมควบคุมระบบเก็บข้อมูลแบบอะนาลอก

4.1 การทำงานของระบบเก็บข้อมูลแบบอะนาลอก

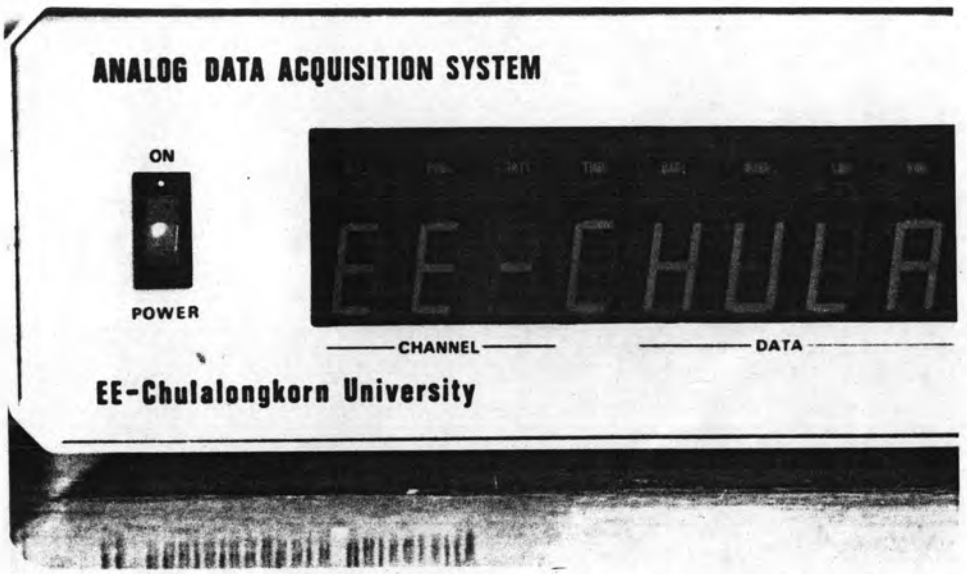
การออกแบบ สร้าง และประกอบวงจรระบบเก็บข้อมูลแบบอะนาลอกตามข้อกำหนด ดังแสดงเครื่องต้นแบบในรูปที่ 4.1 แต่ยังไม่สามารถจะนำไปใช้งานได้ จะต้องเขียนโปรแกรมให้เครื่องต้นแบบสามารถทำงานสัมพันธ์กันทั้งระบบ ในการออกแบบโปรแกรมควบคุมการทำงานของระบบเก็บข้อมูลแบบอะนาลอก จะต้องทราบถึงข้อกำหนดหรือความสามารถในการใช้งานที่ต้องการของระบบว่า สามารถทำอะไรได้บ้าง มีการแสดงผลข้อมูลใด แป้นกดข้อมูลแต่ละปุ่มกดมีหน้าที่ใด รูปแบบการพิมพ์รายงาน และรูปแบบข้อมูลที่ระบบไมโครคอมพิวเตอร์อื่นต้องการ เป็นต้น

4.1.1 การแสดงผลข้อมูล ภาคแสดงผลข้อมูลประกอบด้วยตัวเลขจำนวน 8 หลัก และ



รูปที่ 4.1 แสดงระบบเก็บข้อมูลแบบอะนาลอกที่สร้างขึ้น

มีหลอดแสดงสภาวะของเครื่องจำนวน 8 หลอด โดยอยู่เหนือตัวเลขแต่ละหลัก ดังแสดงในรูปที่ 4.2 สภาวะต่าง ๆ ของหลอด ได้แก่



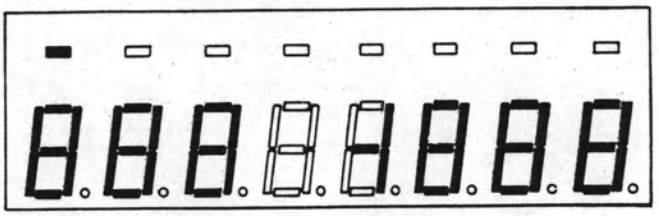
รูปที่ 4.2 แสดงหน้าปัทม์การแสดงผล

1) HOLD ระบบสามารถรับคำสั่งจากแป้นกดข้อมูลได้
2) PROG ระบบสามารถรับการโปรแกรมหรือเปลี่ยนแปลงค่าที่ตั้งไว้จากแป้นกดข้อมูล
3) INTV ข้อมูลที่แสดงอยู่เป็นข้อมูลของช่องเวลาการพิมพ์
4) TIME ข้อมูลที่แสดงอยู่เป็นข้อมูลของเวลาหรือเวลาที่จะตั้ง
5) DATE ข้อมูลที่แสดงอยู่เป็นข้อมูลของ วัน เดือน ปี หรือ วัน เดือน ปี ที่จะตั้ง
6) HIGH ข้อมูลที่แสดงอยู่เป็นข้อมูลของค่ากำหนดสูงสุดหรือค่าที่จะตั้ง
7) LOW ข้อมูลที่แสดงอยู่เป็นข้อมูลของค่ากำหนดต่ำสุดหรือค่าที่จะตั้ง
8) RNGE ข้อมูลที่แสดงอยู่เป็นข้อมูลบอกชนิดของทรานสดิวเซอร์หรือที่จะตั้ง

การแสดงผลที่เป็นตัวเลขจะมีรูปแบบ (Format) ของการแสดงผลตามลักษณะของข้อมูลที่ต้องการทราบ เช่น ข้อมูลที่อ่านจากทรานสดิวเซอร์ ตัวเลขแสดงเวลา เป็นต้น สามารถแบ่ง

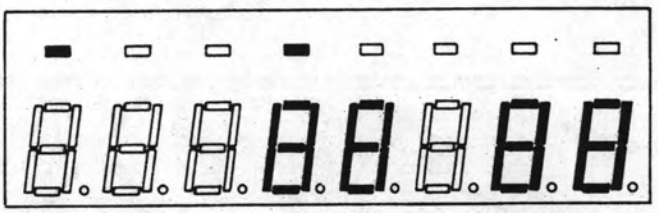
รูปแบบการแสดงผลได้ 5 รูปแบบคือ

ก. แสดงผลข้อมูลที่อ่านจากทรานสดิวเซอร์ ประกอบด้วย หมายเลขช่องสัญญาณ 3 หลัก และข้อมูลจำนวน 4 หลัก โดยมีช่องว่างระหว่างกัน 1 หลัก ดังแสดงในรูปที่ 4.3 รูปแบบการแสดงผลนี้จะใช้กับการแสดงผลข้อมูลที่อ่านได้จากทรานสดิวเซอร์ ค่าสูงสุดหรือค่าต่ำสุดที่ตั้งไว้

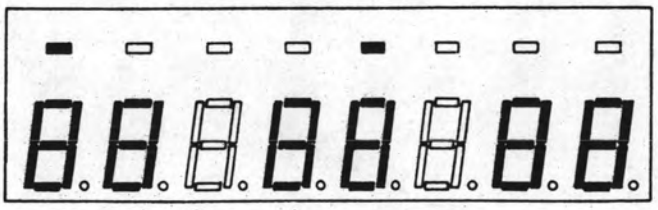


รูปที่ 4.3 แสดงรูปแบบการแสดงผลข้อมูลที่อ่านจากทรานสดิวเซอร์

ข. แสดงเวลา ประกอบด้วย ชั่วโมง 2 หลัก และ นาที 2 หลัก โดยมีช่องว่างระหว่างกัน 1 หลัก ดังแสดงในรูปที่ 4.4 รูปแบบการแสดงผลนี้จะใช้กับการแสดงเวลา และช่วงเวลาการพิมพ์



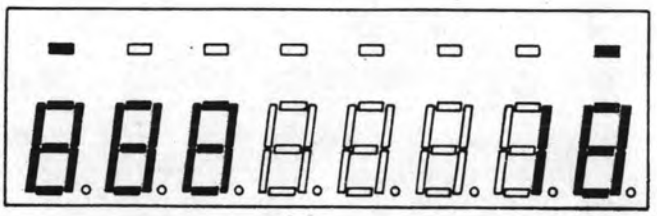
รูปที่ 4.4 แสดงรูปแบบการแสดงเวลา



รูปที่ 4.5 แสดงรูปแบบการแสดง วัน เดือน ปี

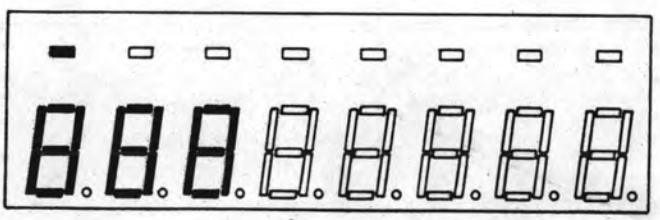
ค. แสดงวัน เดือน ปี ประกอบด้วย วันที่ 2 หลัก เดือน 2 หลัก และปี 2 หลัก โดยมีช่องว่างระหว่างกัน 1 หลัก ดังแสดงในรูปที่ 4.5 รูปแบบการแสดงผลนี้ใช้สำหรับการแสดง วัน เดือน และ ปี เท่านั้น

ง. แสดงชนิดของทรานสดิวเซอร์ ประกอบด้วย หมายเลขช่องสัญญาณ 3 หลักและ ชนิดของทรานสดิวเซอร์ 2 หลัก ดังแสดงในรูปที่ 4.6 รูปแบบการแสดงผลนี้ใช้สำหรับการแสดง ชนิดของทรานสดิวเซอร์เท่านั้น



รูปที่ 4.6 แสดงรูปแบบการแสดงชนิดของทรานสดิวเซอร์

จ. แสดงหมายเลขช่องสัญญาณ ประกอบด้วยหมายเลขช่องสัญญาณ 3 หลัก เท่านั้น ดังแสดงในรูปที่ 4.7 รูปแบบการแสดงผลนี้ใช้สำหรับที่จะเปลี่ยนช่องสัญญาณแสดงผลเท่านั้น

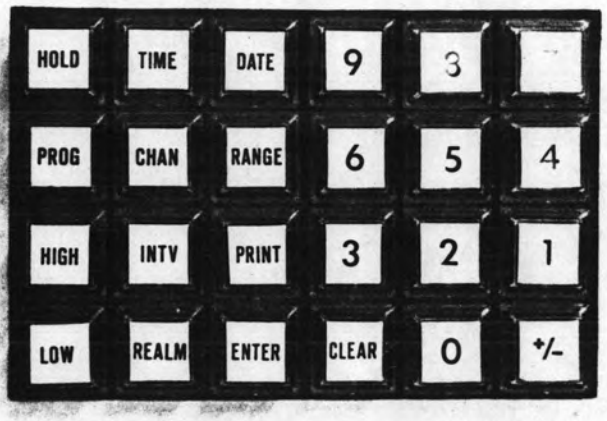


รูปที่ 4.7 แสดงรูปแบบการแสดงหมายเลขช่องสัญญาณ

4.1.2 แป้นกดข้อมูล ประกอบด้วยปุ่มกด 24 ปุ่ม ดังแสดงในรูปที่ 4.8 สามารถแบ่ง ออกเป็นกลุ่มต่าง ๆ ตามหน้าที่ได้ดังนี้คือ

ก. MODE KEY มีหน้าที่เปลี่ยนลักษณะการทำงานของระบบได้แก่

1) HOLD ใช้สำหรับเปลี่ยนลักษณะการทำงานของระบบระหว่าง การแสดงผลและพิมพ์ ข้อมูลแบบอัตโนมัติ กับการรับคำสั่งจากแป้นกดข้อมูล



รูปที่ 4.8 แสดงหน้าปัทม์แป้นกดข้อมูล

2) PROG ใช้สำหรับกำหนดแป้นกดข้อมูลให้แสดงผลข้อมูลหรือเซทค่าให้กับระบบ

ข. FUNCTION KEY มีหน้าที่กำหนดลักษณะของการแสดง หรือพิมพ์ผลข้อมูล ซึ่งมีทั้งหมด 8 ปุ่ม ด้วยกันคือ

1) HIGH ใช้สำหรับอ่าน หรือเซทค่าขีดกำหนดสูงสุด

2) LOW ใช้สำหรับอ่าน หรือเซทค่าขีดกำหนดต่ำสุด

3) TIME ใช้สำหรับอ่าน หรือเซทค่าเวลา ชั่วโมง นาที และวินาที

4) DATE ใช้สำหรับอ่าน หรือเซทค่าวัน เดือน และปี

5) INTV ใช้สำหรับอ่าน หรือเซทค่าช่วงเวลาการพิมพ์ ในการพิมพ์รายงานแบบอัตโนมัติ

6) CHAN ใช้สำหรับเปลี่ยนหมายเลขช่องของสัญญาณเข้า

7) RANGEใช้สำหรับอ่านหรือเซทค่าชนิดของทรานสดิวเซอร์

8) PRINTใช้สำหรับส่งข้อมูลเพื่อที่จะพิมพ์รายงานตามคำสั่งแป้นกดข้อมูล

ค. NUMBER KEY มีหน้าที่สำหรับการเซทค่าตัวเลขต่าง ๆ ให้กับระบบ มีทั้งหมด 12 ปุ่ม ด้วยกัน คือ

1) ตัวเลข 0 - 9 ใช้สำหรับกำหนดตัวเลข

2) +/- ใช้สำหรับกำหนดเครื่องหมาย

3) CLEAR ใช้สำหรับแก้ตัวเลขที่ป้อนผิด

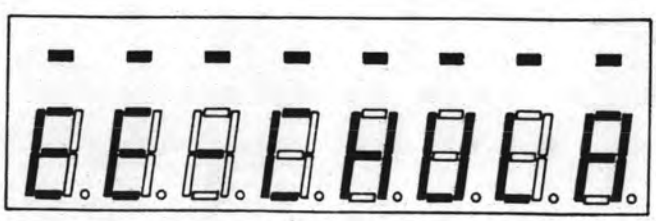
ง. SPECIAL KEY มีหน้าที่อิสระได้แก่

1) REALM ใช้สำหรับหยุดเสียงเตือนเมื่อค่าเกินขีดกำหนด

2) ENTER ใช้สำหรับนำข้อมูลจากปุ่มกดเข้าไปในเครื่อง

4.1.3 การใช้งานของระบบเก็บข้อมูลแบบอะนาลอก สามารถแบ่งการใช้งานตามขั้น-ได้ 4 โมดคือ

ก. INITIAL MODE คือ การทำงานของระบบเมื่อเปิดเครื่องขึ้น โดยระบบจะทำการตรวจดูแบตเตอรีที่เลี้ยงหน่วยความจำ ถ้าแบตเตอรีไม่ทำงานก็จะเซทค่าที่โปรแกรมไว้ทั้งหมดใหม่ โดยเซทให้เป็นศูนย์ทั้งหมด แต่ถ้าแบตเตอรีทำงานปกติระบบจะทำการเซทเฉพาะค่าสภาวะและข้อมูลชั่วคราวเท่านั้น จากนั้นจะแสดงบนภาคแสดงผลเป็น EE-CHULA และหลอดแสดงสภาวะจะติดหมด ดังแสดงในรูปที่ 4.9 ในขณะที่แสดงผลนี้เครื่องพิมพ์จะพิมพ์หัวข้อเริ่มต้น (Heading) เป็น CHULALONGKORN UNIVERSITY ที่เป็นตัวอักษรใหญ่ (Enlarge) และตามด้วยตัวอักษรปกติเป็น ELECTRICAL ENGINEERING การแสดงผลนี้จะใช้เวลาประมาณ 10 วินาที ต่อจากนั้นจะแสดงผลของเวลาขณะนั้น ซึ่งใช้เวลาประมาณ 10 วินาที เช่นกัน แล้วเปลี่ยนไปแสดงผลของวัน เดือน ปี ในช่วงการแสดงผลนี้ เครื่องจะพิมพ์ วัน เดือน ปี และ เวลา ออกที่เครื่องพิมพ์



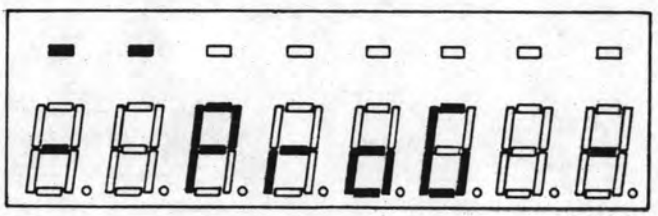
รูปที่ 4.9 แสดงการแสดงผลเมื่อเปิดเครื่อง

ช่วงเวลาการแสดงวัน เดือน ปี ประมาณ 10 วินาที จากนั้นจะเปลี่ยนไปทำงานในขั้นต่อไป คือ PROGRAM MODE สำหรับการพิมพ์ข้อมูลในโมดนี้ ดังแสดงในรูปที่ 4.10

CHULALONGKORN
UNIVERSITY
ELECTRICAL ENGINEERING
DATE 03-05-28 TIME 13:15

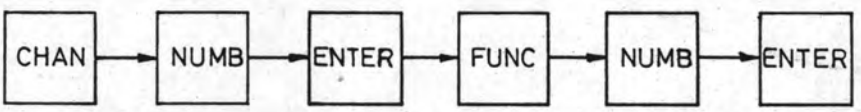
รูปที่ 4.10 แสดงการพิมพ์ข้อมูลใน INITIAL MODE

ข. PROGRAM MODE เป็นขั้นตอนหลังจาก INITIAL MODE ขณะที่ยังไม่มีการกดปุ่มใด ๆ ระบบจะแสดงผลเป็น - PROG - ดังแสดงในรูปที่ 4.11 โดยหลอด HOLD และ PROG ติดอยู่ เมื่ออยู่ในสภาวะนี้ ถ้าเครื่องยังไม่ได้รับการโปรแกรมหรือเซทค่าใด ๆ อยู่ก่อนแล้ว จะไม่สามารถเปลี่ยนไปยังขั้นตอนอื่นได้ การโปรแกรมเซทค่าให้กับระบบทำได้ดังนี้ คือ



รูปที่ 4.11 แสดงการแสดงผลในสภาวะปกติของ PROGRAM MODE

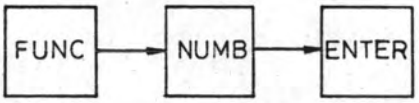
1) การโปรแกรมเซทค่าขีดกำหนดสูงสุดหรือต่ำสุด ทำตามลำดับการกดปุ่มดังแสดงในรูปที่ 4.12 โดยที่ปุ่ม FUNC คือ ปุ่ม HIGH และ LOW และปุ่ม NUMB คือปุ่มเลข 0-9 +/- และ CLEAR



รูปที่ 4.12 แสดงขั้นตอนการกดปุ่มเพิ่มเซทค่าขีดกำหนดสูงสุดหรือต่ำสุด

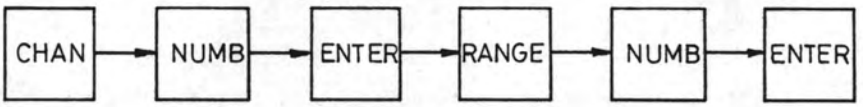
2) การโปรแกรมเซทค่าเวลา วัน เดือน ปี และช่วงเวลาการพิมพ์ ทำตามลำดับการกดปุ่ม ดังแสดงในรูปที่ 4.13 โดยที่ปุ่ม FUNC คือ ปุ่ม TIME DATE และ INTV และปุ่ม NUMB

คือปุ่มตัวเลข 0-9 และ CLEAR



รูปที่ 4.13 แสดงขั้นตอนการกดปุ่มเพื่อเซทค่าเวลา วัน เดือน ปี และช่วงเวลาการพิมพ์

3) การโปรแกรมเซทค่าชนิดของทรานสดิวเซอร์ ทำตามลำดับการกดปุ่มดังแสดงในรูปที่ 4.14 โดยที่ปุ่ม NUMB คือเลข 0-9 และ CLEAR



รูปที่ 4.14 แสดงขั้นตอนการกดปุ่มเพื่อเซทค่าชนิดของทรานสดิวเซอร์

4) การพิมพ์ข้อมูลค่าที่โปรแกรมเซทเอาไว้ เมื่ออยู่ใน PROGRAM MODE ถ้ากดปุ่ม PRINT เครื่องพิมพ์จะพิมพ์ วัน เดือน ปี และเวลาขณะนั้น ช่วงเวลาการพิมพ์ ขีดกำหนดสูงสุดและต่ำสุด ชนิดของทรานสดิวเซอร์ และหน่วยที่ใช้ โดยพิมพ์ตั้งแต่ช่องที่ 1 ถึง 240 ดังแสดงในรูปที่ 4.15

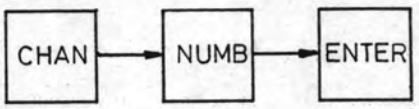
DATE 03-05-28 TIME 13:16
PROGRAM PRINT TIME 00:03

CH NO.	HLIMT	LLIMT	RANGE	UNIT
001	199.9	-199.9	9	C
002	199.9	-199.9	9	C
003	199.9	199.9	9	C
004	199.9	199.9	9	C
005	199.9	-199.9	9	C
006	199.9	199.9	9	C
007	199.9	199.9	9	C
	199.9	199.9	9	C
			9	C
237				
238	0	0	0	
239	0	0	0	mV
240	1999	- 1999	0	mV

รูปที่ 4.15 แสดงการพิมพ์ข้อมูลใน PROGRAM MODE

เมื่ออยู่ใน PROGRAM MODE จะเปลี่ยนไปยังโมดอื่นได้โดยการกดปุ่ม PROG ระบบจะเปลี่ยนไปอยู่ใน HOLD MODE หรือถ้ากดปุ่ม HOLD ระบบจะเปลี่ยนไปอยู่ใน RUN MODE แต่การเปลี่ยนกลับมาอยู่ใน PROGRAM MODE คงเดิม จะต้องกดปุ่ม PROG ในขณะที่ระบบอยู่ใน HOLD MODE เท่านั้น

ค. HOLD MODE เป็นขั้นตอนสำหรับเลือกอ่านข้อมูลที่ต้องการทราบ จากแป้นกดข้อมูล โดยในสภาวะปกติระบบจะแสดง ช่องสัญญาณ และข้อมูล การอ่านค่าต่าง ๆ ของระบบให้หมดปุ่ม FUNCTION KEY ตามที่ต้องการ ระบบจะแสดงผลของข้อมูลที่ต้องการตามปุ่มกด เช่นต้องการดูเวลา ก็ให้กดปุ่ม TIME เป็นต้น เมื่อต้องการให้ระบบเข้าสู่สภาวะปกติ ให้กดปุ่ม ENTER หรือกดปุ่ม FUNCTION KEY ปุ่มอื่นที่ต้องการอ่านค่าใหม่ สำหรับการเปลี่ยนช่องสัญญาณการแสดงผลให้ให้ทำตามลำดับการกดปุ่มดังแสดงในรูปที่ 4.16



รูปที่ 4.16 แสดงขั้นตอนการกดปุ่มเปลี่ยนการแสดงผลช่องสัญญาณ

การพิมพ์ข้อมูลในโมดนี้ ให้กดปุ่ม PRINT เช่นกัน เครื่องพิมพ์จะพิมพ์ วัน เดือน ปี และเวลาขณะนั้น ข้อมูลที่อ่านได้จากทรานสดิวเซอร์ และหน่วยที่ใช้ เฉพาะช่องสัญญาณที่ใช้จริง

```
HOLD                    TIME  13:31

DATE   03-05-28         TIME  13:31
CH NO. DATA  UNIT   CH NO. DATA  UNIT
001    28.5    C    002    36.3    C
005    36.0    C    015    36.2    C

* UNDER LOW LIMIT
  NOT FOUND

* OVER HIGH LIMIT
  NOT FOUND

* OPEN PROBE
  NOT FOUND
```

รูปที่ 4.17 แสดงการพิมพ์ข้อมูลใน HOLD MODE

เท่านั้น นอกจากนั้นจะพิมพ์ข้อมูลที่ผิดปกติ เช่น ข้อมูลที่เกินค่าขีดกำหนดสูงสุด (Over high limit) ข้อมูลที่ต่ำกว่าขีดกำหนดต่ำสุด (Under Low limit) และข้อมูลที่เกินช่วงการแปลงสัญญาณ (Over range) เป็นต้น ดังแสดงผลการพิมพ์ในรูปที่ 4.17 สำหรับช่องสัญญาณที่ใช้ในการวัดจริงจะเป็นช่องสัญญาณที่ค่าขีดกำหนดสูงสุดและต่ำสุดไม่เท่ากัน และรหัสชนิดทรานสดิวเซอร์ไม่เป็นศูนย์ ดังนั้นถ้าช่องสัญญาณใดไม่ได้ต่อทรานสดิวเซอร์ได้ จะต้องโปรแกรมค่าขีดกำหนดสูงสุดและต่ำสุดให้มีค่าเท่ากัน หรือถ้าไม่ได้ต่อไว้ทั้งแผงวงจรเลือกสัญญาณ ก็ให้โปรแกรมค่าของรหัสชนิดทรานสดิวเซอร์ให้เป็นศูนย์ การเปลี่ยนจาก HOLD MODE ไปยังโมดอื่นทำได้โดยการกดปุ่ม PROG ระบบจะเปลี่ยนไปอยู่ใน PROGRAM MODE หรือถ้ากดปุ่ม HOLD ระบบจะถูกเปลี่ยนไปอยู่ใน RUN MODE การเปลี่ยนโมดทุกครั้งเครื่องพิมพ์จะพิมพ์โมดที่เปลี่ยน และเวลา ขณะเปลี่ยนโมด

ง. RUN MODE เป็นขั้นตอนที่ระบบทำการแสดงผล และพิมพ์รายงานตามเวลาที่กำหนดไว้อย่างอัตโนมัติ โดยการแสดงค่าข้อมูลของแต่ละช่องสัญญาณประมาณ 7 วินาทีแล้วเปลี่ยนช่องสัญญาณต่อไป และเครื่องพิมพ์จะพิมพ์รายงานที่มีลักษณะเช่นเดียวกับใน HOLD MODE ทุก ๆ ช่วงเวลาที่เซทเอาไว้ การทำงานใน RUN MODE ปุ่มกดต่าง ๆ จะไม่มีผล ยกเว้นปุ่ม HOLD และปุ่ม REALM โดยปุ่ม HOLD จะใช้สำหรับเปลี่ยนระบบไปอยู่ใน HOLD MODE และปุ่ม REALM จะใช้สำหรับการหยุดเสียงและหน้าสัมผัสของภาคเตือนภัย ในกรณีที่มีช่องสัญญาณใดอ่านค่าได้เกินขีดจำกัดสูงสุดหรือต่ำสุด

นอกจากการใช้งานตามขั้นตอนดังที่กล่าวมาแล้ว จะต้องทราบรายละเอียดเกี่ยวกับรหัสของชนิดทรานสดิวเซอร์ (RANGE) ซึ่งจะต้องตั้งให้ตรงกับแผงวงจรภาคเลือกสัญญาณ ที่เราปรับอัตราขยายของวงจรขยายสัญญาณว่าใช้สำหรับทรานสดิวเซอร์ชนิดใด ดังนั้นการกำหนดรหัสชนิดของทรานสดิวเซอร์ เพียงช่องสัญญาณเดียว ก็จะเปลี่ยนไปทั้งกลุ่มสวิทช์ที่อยู่บนแผงวงจรเดียวกัน โดยแต่ละกลุ่มจะมี 15 ช่องสัญญาณ และเรียงลำดับเป็นกลุ่ม ๆ ดังนี้คือ (1-15), (16-30),.. (226-240) รหัสที่กำหนดชนิดของทรานสดิวเซอร์จะเป็นตัวกำหนด ชนิดของสัญญาณเข้า ค่าสูงสุดในการวัด (Full scale) และหน่วยที่ใช้วัด ซึ่งมีทั้งหมด 15 ชนิด คือ รหัส 1-15 ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 แสดงรหัสของชนิดทรานสดิวเซอร์หรือสัญญาณเข้า

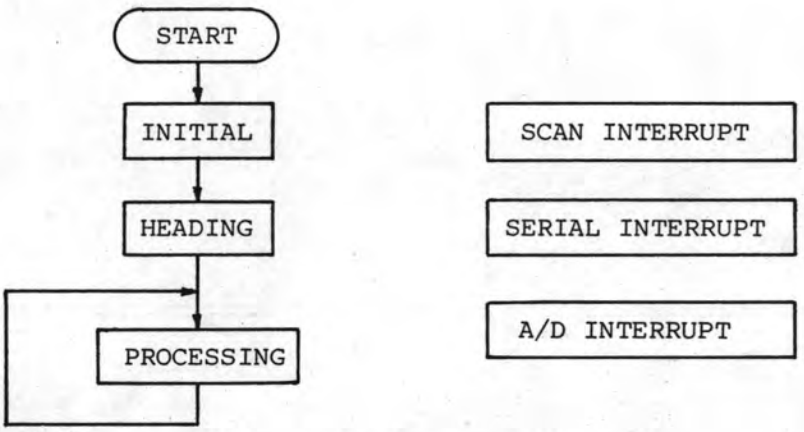
รหัส	ชนิดสัญญาณเข้า	ช่วงการวัดสูงสุด	หน่วย
1	แรงดัน	-2.000 ถึง 2.000	mV
2	แรงดัน	-20.00 ถึง 20.00	mV
3	แรงดัน	-200.0 ถึง 200.0	mV
4	แรงดัน	-2000 ถึง 2000	mV
5	แรงดัน	-20.00 ถึง 20.00	V
6	กระแส	-20.00 ถึง 20.00	mA
7	กระแส	-200.0 ถึง 200.0	mA
8	เทอร์โมคับเบิล	-2000 ถึง 2000	°C
9	เทอร์โมคับเบิล	-200.0 ถึง 200.0	°C
10	เทอร์โมคับเบิล	-2000 ถึง 2000	°F
11	เทอร์โมคับเบิล	-200.0 ถึง 200.0	°F
12	กระเปาะความต้านทาน	-200.0 ถึง 200.0	°C
13	กระเปาะความต้านทาน	-200.0 ถึง 200.0	°F
14	เทอร์มีสเตอร์	-200.0 ถึง 200.0	°C
15	เทอร์มีสเตอร์	-200.0 ถึง 200.0	°F

4.2 โปรแกรมควบคุมการทำงาน

ฟังก์ชันของระบบเก็บข้อมูลแบบอะนาลอก จะถูกกำหนดโดยโปรแกรมควบคุม โดยที่โปรแกรมควบคุมนี้จะเขียนด้วยภาษาเครื่อง และอัดเก็บไว้ใน EPROM ซึ่งเป็นหน่วยความจำที่สำคัญของระบบ โปรแกรมควบคุมที่พัฒนาขึ้นมาสำหรับระบบเก็บข้อมูลแบบอะนาลอกนี้ จะมีโปรแกรมที่สำคัญของระบบเท่าที่จำเป็นเท่านั้นสำหรับในหัวข้อนี้จะอธิบายถึงโฟลว์ชาร์ท MEMORY MAP I/O MAP และหน้าที่ของโปรแกรมย่อยต่าง ๆ

4.2.1 โปรแกรมหลัก (MAIN PROGRAM) เป็นโปรแกรมที่ควบคุมระบบทั้งหมด ประกอบด้วยโปรแกรมย่อย ๆ ดังแสดงโฟลว์ชาร์ทในรูปที่ 4.18 เมื่อเปิดเครื่องจ่ายไฟเข้าระบบ การทำงานของระบบจะเข้าสู่ภาคแรกของโปรแกรม ได้แก่การเซทค่าเริ่มแรกต่าง ๆ ให้กับระบบ (INITIAL) ได้แก่ การเซทค่าเริ่มแรกให้กับหน่วยความจำ การเซทค่าให้อุปกรณ์อินเตอร์เฟสต่าง ๆ เป็นต้น เมื่อผ่านโปรแกรมการเซทค่าให้กับระบบแล้วจะเข้าสู่โปรแกรมการแสดงผลและพิมพ์ข้อความเริ่มแรก (HEADING) ได้แก่การทำงานในขั้นตอน INITIAL MODE ดังที่ได้กล่าวมาแล้ว จากนั้นจะเข้าสู่โปรแกรมการทำงานของระบบ (PROCESSING) ซึ่งการทำงานของโปรแกรมนี้ ได้แก่ การอ่านข้อมูลเวลา การแสดงผล การส่งข้อมูลเข้าเครื่องพิมพ์ และส่งข้อมูลเข้าเครื่องไมโครคอมพิวเตอร์ระบบอื่น เป็นต้น

MAIN PROGRAM



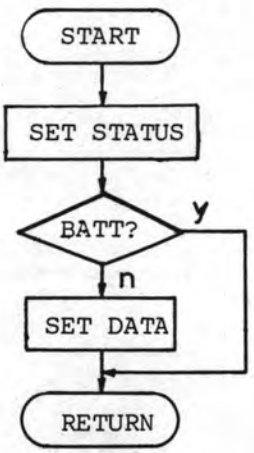
รูปที่ 4.18 แสดงโฟลว์ชาร์ทของ MAIN PROGRAM

นอกจากโปรแกรมย่อยที่กล่าวมาแล้ว ยังมีโปรแกรม ซึ่งทำงานเมื่อมีการอินเตอร์รัพท์จาก CTC อีก 3 โปรแกรม คือ การอินเตอร์รัพท์จากภาคแสดงผลและสแกนแป้นกดข้อมูล (SCAN INTERRUPT) การอินเตอร์รัพท์จากภาครับส่งข้อมูลแบบอนุกรม (SERIAL INTERRUPT) การอินเตอร์รัพท์จากภาคแปลงสัญญาณ (A/D INTERRUPT)

4.2.2 โปรแกรมการเซทค่าเริ่มแรกให้กับระบบ ประกอบด้วยโปรแกรมย่อย 2 โปรแกรมได้แก่

ก. โปรแกรมการเซทค่าให้กับหน่วยความจำ (INITIAL MEMORY) เป็นการเซทค่าเริ่มแรกให้กับหน่วยความจำ โดยข้อมูลที่เก็บไว้ในหน่วยความจำ จะแบ่งออกเป็น 2 ชนิด คือ ข้อมูลที่แสดงสภาวะของระบบ (Status) เป็นข้อมูลที่บอกสภาวะของระบบขณะนั้น หรือเป็นข้อมูลชั่วคราว ข้อมูลเหล่านี้ไม่จำเป็นต้องรักษาค่าเอาไว้เมื่อปิดเครื่อง เนื่องจากจะต้องกำหนดค่าให้ใหม่เมื่อเปิดเครื่องขึ้นมาอีก ข้อมูลเหล่านี้ ได้แก่ ข้อมูลแสดงสภาวะของคำสั่งจากแป้นกดข้อมูล สภาวะการรับส่งข้อมูลกับอุปกรณ์ภายนอก เป็นต้น ข้อมูลอีกชนิดหนึ่งคือ ข้อมูลที่ต้องการการเซทค่าจากแป้นกดข้อมูล หรือข้อมูลที่ได้จากทรานสดิวเซอร์ ข้อมูลเหล่านี้จะต้องรักษาไว้เมื่อดับเครื่อง เพื่อที่จะได้ไม่เสียเวลามาเซทค่าใหม่ ดังนั้นการทำงานของโปรแกรมนี้จะต้องมีการตรวจดูว่าแบตเตอรี่ที่เลี้ยงหน่วยความจำยังทำงานตามปกติหรือไม่ เพราะถ้าแบตเตอรี่หมดหรือขาดหายไป ค่าโปรแกรมที่เซทไว้ให้กับระบบจะผิดพลาดไปจะต้องมีการเซทค่าข้อมูลใหม่ทั้งหมด ดังแสดงโฟลว์-ชาร์ทในรูปที่ 4.19

INITIAL MEMORY



รูปที่ 4.19 แสดงโฟลว์ชาร์ทของโปรแกรมการเซทค่าให้กับหน่วยความจำ

ข. โปรแกรมการเซทค่าให้กับอุปกรณ์ของระบบ (INITIAL INTERFACE) เป็นการเซทค่าให้อุปกรณ์ของระบบให้สามารถทำงานได้ตามที่ออกแบบไว้ ดังแสดงโฟลว์ชาร์ทในรูปที่ 4.20 ได้แก่ การเซทค่าต่าง ๆ ดังนี้ คือ

1) อินเตอร์รัพท์เวคเตอร์ของระบบ (INTERRUPT VECTOR)

2) PIO ทำงานในโมดอินพุท และไม่มีการอินเตอร์รัพท์

3) PIA 8255 NO.1 สำหรับภาคอินเตอร์เฟสกับเครื่องพิมพ์และภาคนาฬิกา โดยที่พอร์ท A และพอร์ท B เป็นโมดเอาท์พุท และพอร์ท C เป็นโมดอินพุท

4) PIA 8255 NO.2 สำหรับภาคเลือกสัญญาณและภาคเตือนภัย โดยที่พอร์ท A พอร์ท B และพอร์ท C เป็นโมดเอาท์พุททั้งหมด

5) PIA 8255 NO.3 สำหรับภาคแสดงผลและแป้นกดข้อมูล โดยที่พอร์ท A เป็นโมดอินพุท พอร์ท B และพอร์ท C เป็นโมดเอาท์พุท

6) CTC มีทั้งหมด 4 ชุดภายในตัวเองการเซทค่าจะต้องทำทีละชุดคือ

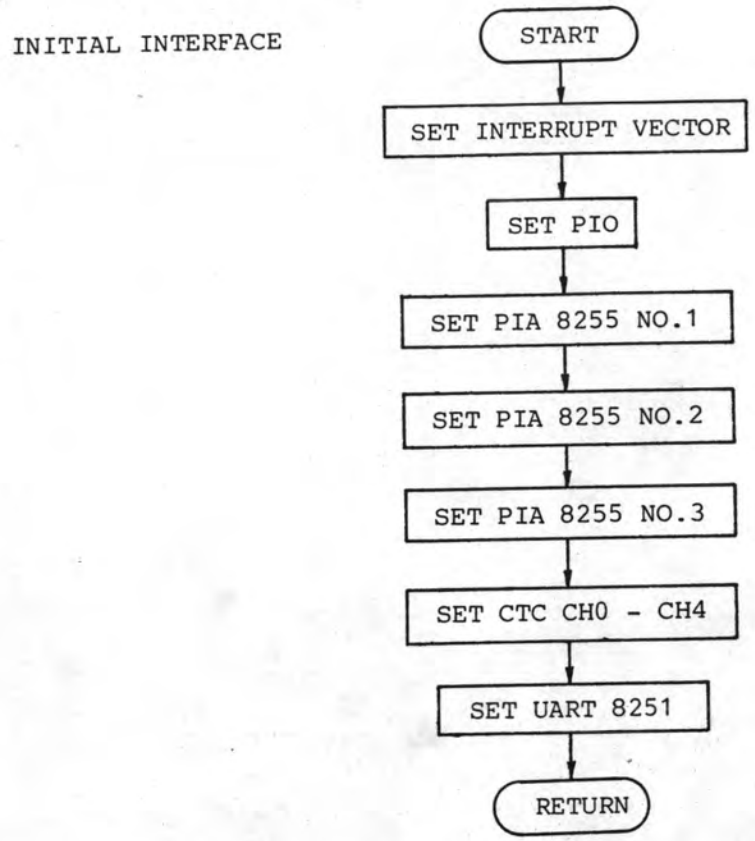
CH 0 ใช้เป็นตัวสร้างสัญญาณนาฬิกาสำหรับการรับส่งข้อมูลแบบอนุกรม เซทให้เป็นดังนี้ คือ DISABLE INTERRUPT, TIMER MODE, PRESCALER VALUE=16 FALLING EDGE TRIGGER, AUTOMATIC TRIGGER, LOAD TIMECONSTANT= 03 และ CONTINUED OPERATION ความถี่ของพัลส์นาฬิกาที่ได้ = 37.3 KHz

CH 1 ใช้สำหรับอินเตอร์รัพท์ เมื่อการแปลงสัญญาณครบหนึ่งรอบ เซทให้เป็นดังนี้ คือ ENABLE INTERRUPT, COUNTER MODE, LOAD TIMECONSTANT=03 และ CONTINUED OPERATION

CH 2 ใช้สำหรับอินเตอร์รัพท์ เมื่อมีความต้องการข้อมูลจากเครื่องไมโครคอมพิวเตอร์ระบบอื่น เซทให้เป็นดังนี้ คือ ENABLE INTERRUPT, TIMER MODE, PRESCALER VALUE=16, RISING EDGE TRIGGER, TRIGGER START, LOAD TIMECONSTANT=02 และ CONTINUED OPERATION

CH 3 ใช้สำหรับอินเตอร์รัพท์ ในการแสดงผลและสแกนแป้นกดข้อมูล เซทให้เป็นดังนี้ คือ ENABLE INTERRUPT, TIMER MODE, PRESCALER VALUE=16 FALLING EDGE TRIGGER, AUTOMATIC TRIGGER, LOAD TIMECONSTANT= 70 และ CONTINUED OPERATION ความถี่ของพัลส์นาฬิกาที่ได้ = 1.6 KHz

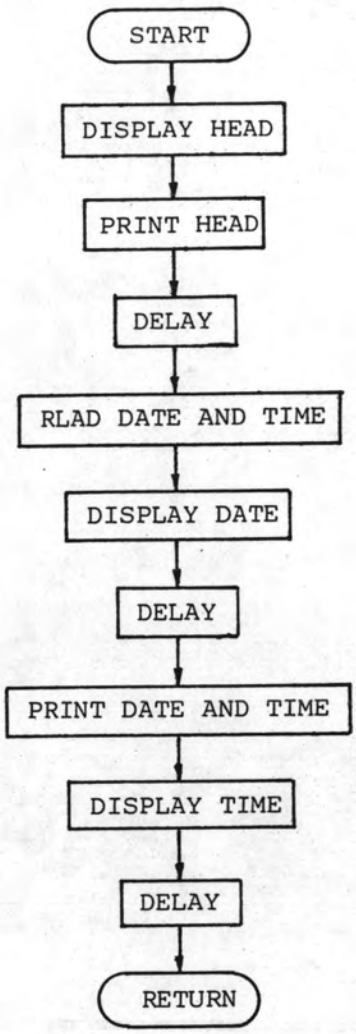
7) UART 8251 สำหรับการรับส่งข้อมูลแบบอนุกรม จะต้องเซทรูปแบบของข้อมูลที่จะใช้ในการรับส่ง โดยเซทให้เป็นดังนี้ คือ 1 STOP BIT, PARITY ENABLE, EVEN PARITY, 8 DATA BITS และอัตราเร็วในการรับส่งข้อมูล = CLOCK/16 ซึ่งจะได้เท่ากับ 2330 บิท/วินาที สำหรับคำสั่งการเซทสภาวะ ใช้เฉพาะ TXRDY และ PARITY ERROR เท่านั้น



รูปที่ 4.20 แสดงโฟลว์ชาร์ทของโปรแกรมการเซทค่าให้กับอุปกรณ์ของระบบ

4.2.3 โปรแกรมการแสดงผลและพิมพ์ข้อความเริ่มแรก เริ่มต้นด้วยการแสดงผลข้อความเริ่มแรก (DISPLAY HEAD) คือ EE-CHULA และการพิมพ์ข้อความเริ่มแรก (PRINT HEAD) คือ CHULALONGKORN UNIVERSITY และ ELECTRICAL ENGINEERING จากนั้นจะเข้าโปรแกรมหน่วงเวลา (DELAY) เพื่อให้การแสดงผลคงค่าอยู่ช่วงเวลาหนึ่ง ในช่วงนี้จะมีการอ่านค่าของเวลา และ วัน เดือน ปี (READ DATE AND TIME) จากภาคนาฬิกา เข้ามาเก็บไว้ในหน่วยความจำ จากนั้นจะเปลี่ยนมาแสดงผลของ วัน เดือน ปี (DISPLAY DATE) ช่วงเวลาหนึ่ง แล้วจึงเปลี่ยนมาแสดงผลของเวลา (DISPLAY TIME) อีกช่วงเวลาหนึ่ง ขณะที่เปลี่ยนจากการแสดงผลของ วัน เดือน ปี มาเป็นการแสดงผลของเวลา ระบบส่งข้อมูลเวลา และวัน เดือน ปี ออกไปที่เครื่องพิมพ์ เมื่อหมดเวลาการแสดงผลของเวลา ก็จะเข้าสู่โปรแกรมอื่นต่อไปดังแสดงโฟลว์ชาร์ทของโปรแกรมการแสดงผลและพิมพ์ข้อความเริ่มแรกในรูปที่ 4.21

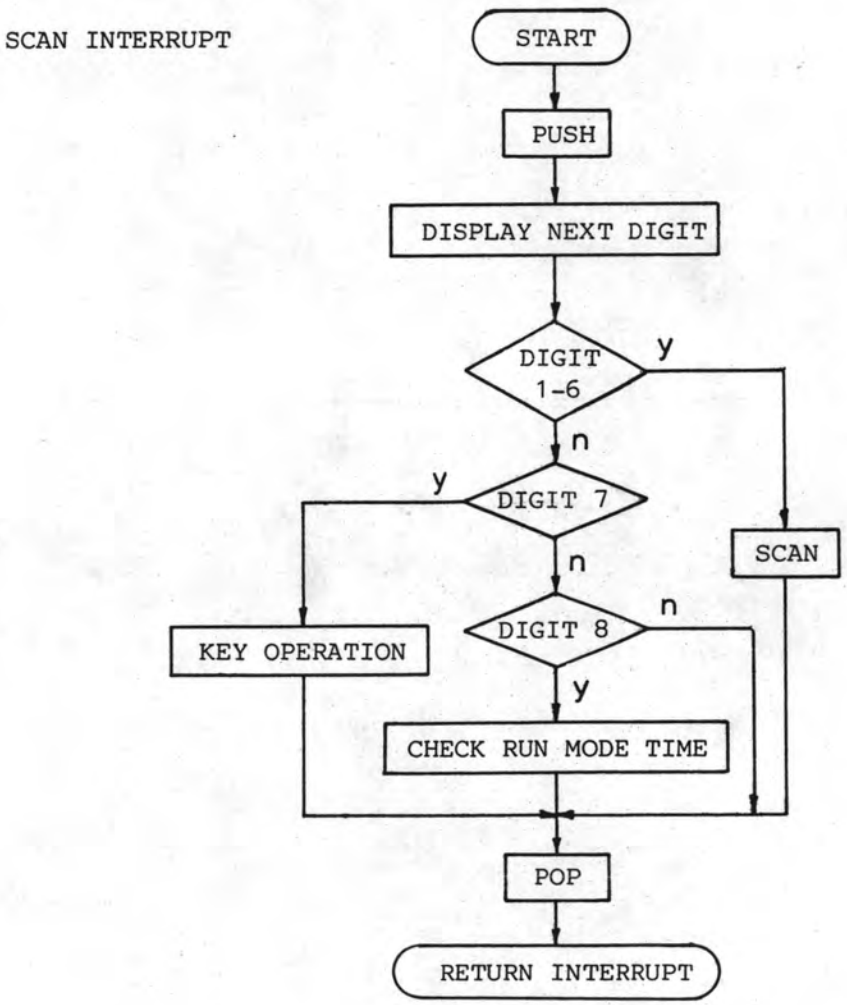
HEADING



รูปที่ 4.21 แสดงโฟล์วชาร์ทของโปรแกรมการแสดงผลและพิมพ์ข้อความเริ่มแรก

4.2.4 โปรแกรมการอินเตอร์รัพท์ เป็นโปรแกรมที่กระทำเมื่อมีการอินเตอร์รัพท์ โดยการอินเตอร์รัพท์จะมาจาก CTC ซึ่งจะมีเวคเตอร์ของการอินเตอร์รัพท์มาด้วยทำให้ทราบว่าการอินเตอร์รัพท์มาจากอุปกรณ์อินเตอร์เฟสใด การทำงานของโปรแกรมโดยมาก จะเป็นการเซทค่าสภาวะ หรือความต้องการของการอินเตอร์รัพท์เอาไว้เท่านั้น เพื่อไม่ให้การทำงานในโปรแกรมหลักต้องหยุดนานเกินไป และจะมีการตรวจความต้องการนั้น ทุกรอบในโปรแกรมการทำงานของระบบ โปรแกรมการอินเตอร์รัพท์มีทั้งหมด 3 โปรแกรม คือ

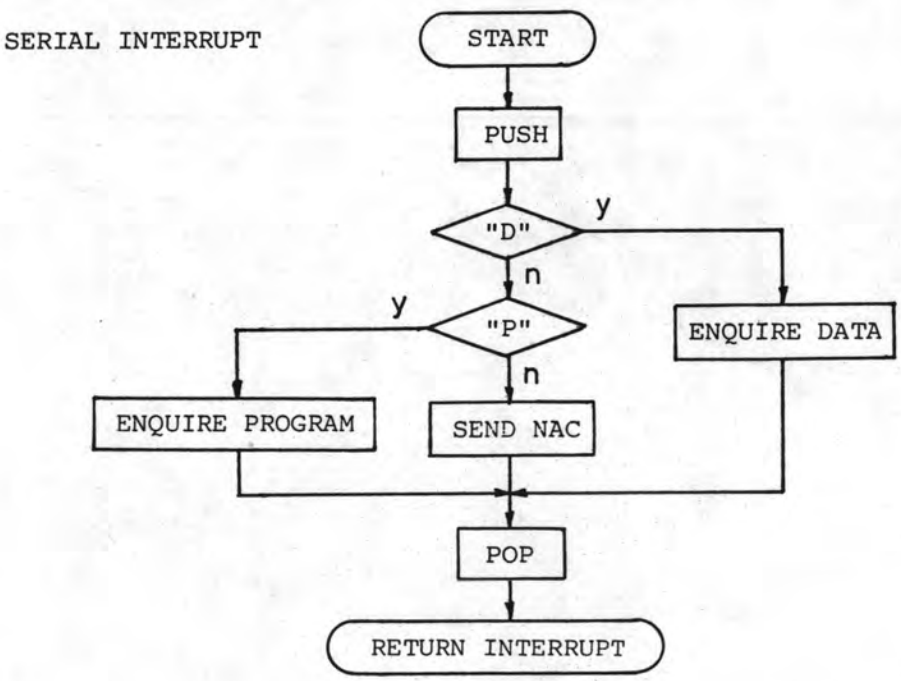
ก. โปรแกรมอินเตอร์รัพท์จากภาคแสดงผลและสแกนแป้นกดข้อมูล เป็นโปรแกรมสำหรับมัลติเพล็กซ์การแสดงผลและสแกนแป้นกดข้อมูล การทำงานในโปรแกรมได้แก่ การแสดงผลหลักถัดไป (DISPLAY NEXT DIGIT) โดยจะทำการเปลี่ยนรหัสหลัก และรหัส 7 ขีด ส่งออกไปที่ภาคแสดงผล การแสดงผลมีทั้งหมด 9 หลักต่อรอบ ในช่วงการแสดงผลหลักที่ 1 ถึง 6 จะทำการสแกนแป้นกดข้อมูล (SCAN) ทั้งหมด 6 แถว ในหลักที่ 7 จะเป็นการทำงานตามแป้นกดข้อมูล (KEY OPERATION) ได้แก่ การกำหนดสภาวะ หรือฟังก์ชันของระบบตามคำสั่งแป้นกดข้อมูล ในหลักที่ 8 จะเป็นการตรวจสอบ และเปรียบเทียบเวลาที่ใช้ในขั้นตอน RUN MODE (CHECK RUN MODE TIME) ได้แก่ การตรวจสอบเวลาการเปลี่ยนช่องสัญญาณการแสดงผล ตรวจสอบเวลาการส่งข้อมูลออกพิมพ์รายงานตามเวลาที่ตั้งไว้ เป็นต้น สำหรับในหลักที่ 9 จะไม่มีการทำโปรแกรมใด



รูปที่ 4.22 แสดงโฟลว์ชาร์ทของโปรแกรมอินเตอร์รัพท์
จากภาคแสดงผล และสแกนแป้นกดข้อมูล

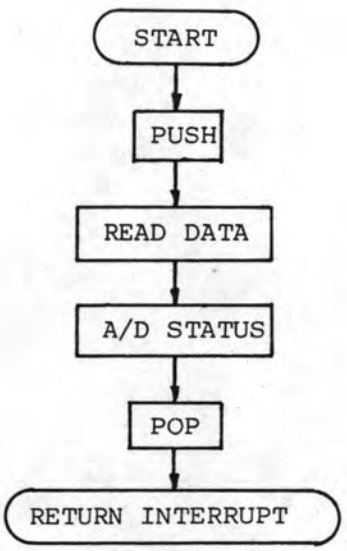
ในโปรแกรมอินเตอร์รัพท์นี้ ก่อนที่จะเข้าโปรแกรมย่อย จะต้องมีการเก็บรักษาค่าในรีจีสเตอร์ต่าง ๆ (PUSH) โดยจะเก็บไว้ในสแตค (Stack) และเมื่อทำงานเสร็จสิ้นโปรแกรมแล้ว จะต้องนำค่าในรีจีสเตอร์นั้นมาเก็บไว้ดังเดิม (POP) นอกจากนี้จะต้องมีคำสั่ง ENABLE INTERRUPT ก่อนที่จะกลับเข้าโปรแกรมเดิมเพื่อให้ระบบสามารถรับการอินเตอร์รัพท์ได้อีก ดังแสดงโฟลว์ชาร์ทของโปรแกรมอินเตอร์รัพท์จากภาคแสดงผลและสแกนแป้นกดข้อมูลในรูปที่ 4.22

ข. โปรแกรมอินเตอร์รัพท์จากการรับส่งข้อมูลแบบอนุกรม เป็นโปรแกรมสำหรับใช้ตรวจความต้องการในการรับส่งข้อมูลแบบอนุกรม โดยตรวจดูว่ารหัสที่ส่งเข้ามาเป็น "D" หรือไม่ ถ้าเป็น "D" แสดงว่าข้อมูลที่ต้องการเป็นข้อมูลที่อ่านจากทรานสดิวเซอร์ โปรแกรมจะไปเซ็ทค่าสภาวะของระบบว่ามีความต้องการข้อมูลที่อ่านจากทรานสดิวเซอร์ (ENQUIRE DATA) ทางบัสอาร์เอส 232 - ซี ถ้ารหัสเป็น "P" แสดงว่าข้อมูลที่ต้องการเป็นข้อมูลที่เซ็ทค่าให้กับระบบจากแป้นกดข้อมูล (ENQUIRE PROGRAM) โปรแกรมจะไปเซ็ทค่าสภาวะของระบบเช่นกัน แต่ถ้าเป็นรหัสอื่นระบบไม่สามารถรับรู้และจะส่งรหัส "N" กลับไป (SEND NAC) ดังแสดงโฟลว์ชาร์ทของโปรแกรมอินเตอร์รัพท์จากการรับส่งข้อมูลแบบอนุกรมในรูปที่ 4.23



รูปที่ 4.23 แสดงโฟลว์ชาร์ทของโปรแกรมอินเตอร์รัพท์
จากการรับส่งข้อมูลแบบอนุกรม

ค. โปรแกรมอินเตอร์รัพท์จากภาคแปลงสัญญาณ เป็นโปรแกรมสำหรับอ่านข้อมูลที่เสร็จจากการแปลงสัญญาณเข้าไปเก็บไว้ในหน่วยความจำชั่วคราว (READ DATA) และเซ็ทค่าสภาวะของระบบว่า การแปลงสัญญาณได้เสร็จสิ้นแล้ว ให้นำค่าที่ได้ในหน่วยความจำชั่วคราวไปทำการประมวลผล (A/D STATUS) ดังแสดงโฟลว์ชาร์ทของโปรแกรมอินเตอร์รัพท์จากภาคแปลงสัญญาณในรูปที่ 4.24

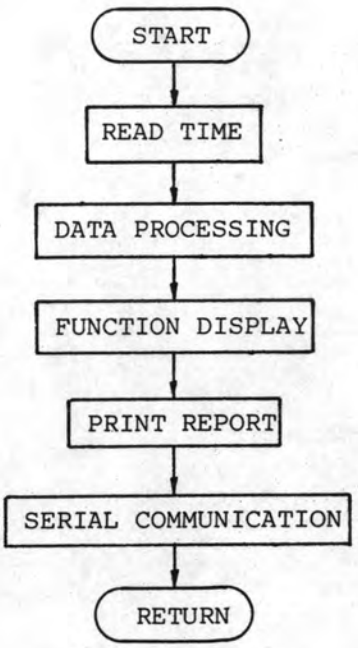


รูปที่ 4.24 แสดงโฟลว์ชาร์ทของโปรแกรมอินเตอร์รัพท์
จากภาคแปลงสัญญาณ

4.2.5 โปรแกรมการทำงานของระบบ เป็นโปรแกรมที่สำคัญของระบบ ใช้สำหรับควบคุมการทำงานทั้งหมดในสภาวะปกติ การทำงานของโปรแกรมส่วนใหญ่จะเป็นการตรวจสอบสภาวะของระบบและทำงานตามสภาวะนั้น ๆ ขั้นตอนการทำงานจะเริ่มจากการอ่านข้อมูลเวลา (READ TIME) โดยจะอ่านข้อมูลเวลาเข้ามาเก็บไว้ในหน่วยความจำเพื่อเป็นการเปลี่ยนข้อมูลเวลาให้ถูกต้องตรงกับเวลาจริง ลำดับต่อมาจะเป็นการตรวจสภาวะการอินเตอร์รัพท์จากภาคแปลงสัญญาณว่ามีการอินเตอร์รัพท์เข้ามาหรือไม่ ถ้ามีการอินเตอร์รัพท์เข้ามาก็จะนำข้อมูลที่ได้ไปทำงานประมวลผล (DATA PROCESSING) การตรวจสอบสภาวะนี้จะต้องกระทำบ่อยครั้ง เพื่อที่จะให้ระบบสามารถทำงานได้ตามข้อกำหนดที่ให้เวลาการอ่านข้อมูลจากทรานสดิวเซอร์ไม่เกิน 0.1 วินาทีต่อหัววัด โดยจะมีการตรวจสอบนี้สลับกับการทำงานของโปรแกรมย่อยอื่น ๆ คือ การแสดงผลตาม

ฟังก์ชันที่ได้รับจากแป้นกดข้อมูล (FUNCTION DISPLAY) ตรวจสอบความต้องการในการพิมพ์รายงานออกที่เครื่องพิมพ์ (PRINT REPORT) ตรวจสอบความต้องการในการรับส่งข้อมูลแบบอนุกรม (SERIAL COMMUNICATION) ดังแสดงโฟลว์ชาร์ทของโปรแกรมการทำงานของระบบในรูปที่ 4.25

PROCESSING



รูปที่ 4.25 แสดงโฟลว์ชาร์ของโปรแกรมการทำงานของระบบ

4.2.6 โปรแกรมการแสดงผลตามฟังก์ชันที่ได้รับจากแป้นกดข้อมูล ฟังก์ชันการทำงานของระบบจะถูกเก็บเป็นรหัสไว้ในหน่วยความจำ ดังนั้นจึงต้องนำมาตรวจสอบดูว่าฟังก์ชันขณะนั้นอยู่ในโมดอะไร มีการแสดงผลเป็นรูปแบบใด จากนั้นจึงกระโดด (Jump) เข้าไปในโปรแกรมตามรหัสนั้น ดังแสดงโฟลว์ชาร์ทของโปรแกรมนี้ในรูปที่ 4.26 โดยมีโปรแกรมย่อยต่าง ๆ คือ

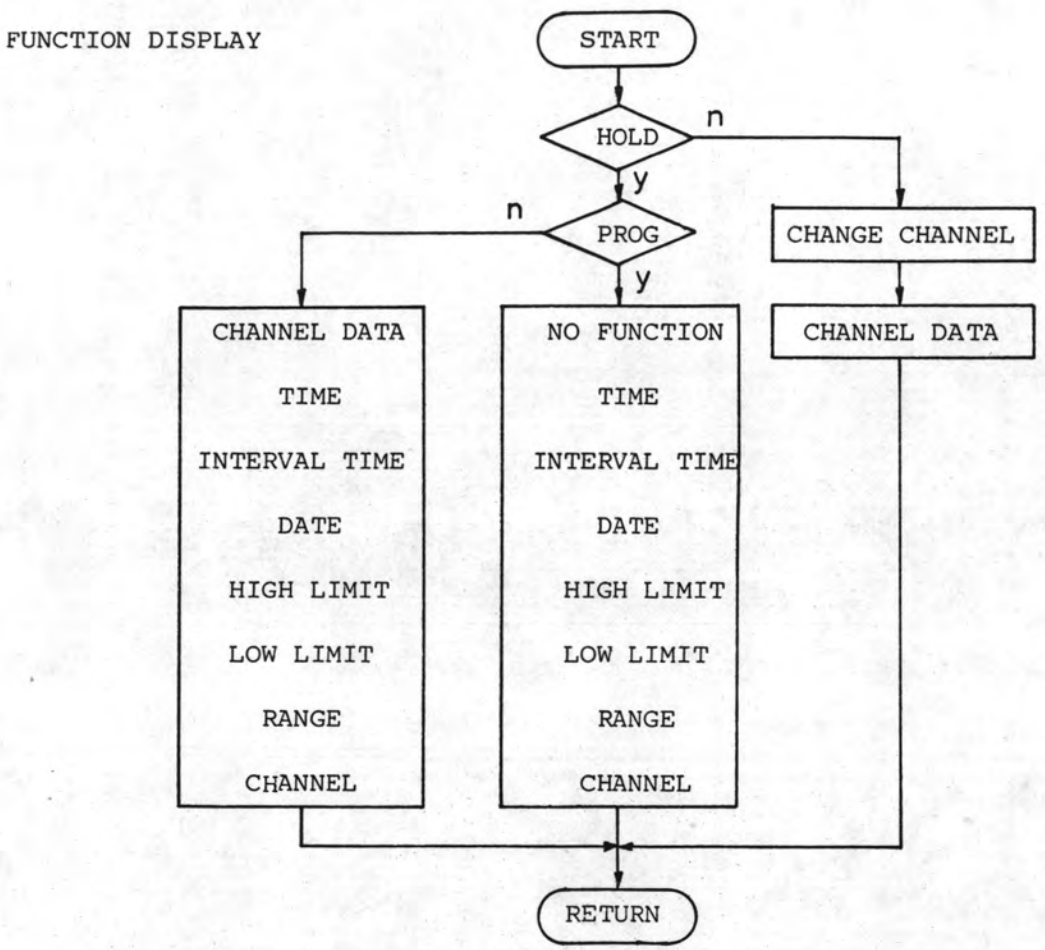
1) แสดงช่องสัญญาณ และข้อมูลที่อ่านจากทรานสดิวเซอร์ (CHANNEL-DATA)
2) แสดงเวลา ชั่วโมง และนาที (TIME)
3) แสดง วัน เดือน ปี (DATE)
4) แสดงช่วงเวลาการพิมพ์ (INTERVAL TIME)
5) แสดงช่องสัญญาณและค่าขีดกำหนดสูงสุด (HIGH LIMIT)
6) แสดงช่องสัญญาณและค่าขีดกำหนดต่ำสุด (LOW LIMIT)

7) แสดงชนิดของทรานสดิวเซอร์ (RANGE)

8) แสดงช่องสัญญาณที่จะเปลี่ยน (CHANNEL)

9) แสดง -Prog- ในสภาวะปกติของ PROGRAM MODE (NO FUNCTION)

10) การเปลี่ยนสัญญาณแบบอัตโนมัติใน RUN MODE (CHANGE CHANNEL)

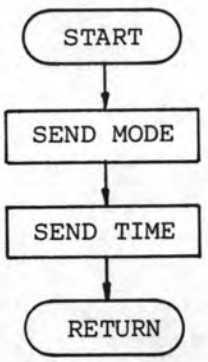


รูปที่ 4.26 แสดงโฟลว์ชาร์ทของโปรแกรมการแสดงผล
ตามฟังก์ชันที่ได้รับจากแป้นกดข้อมูล

4.2.7 โปรแกรมการพิมพ์รายงานออกที่เครื่องพิมพ์ เป็นโปรแกรมสำหรับจัดรูปแบบข้อมูลและส่งข้อมูลออกไปที่เครื่องพิมพ์ รูปแบบหรือชุดของข้อมูลที่ส่งออกไปพิมพ์ สามารถแบ่งตามชนิดของข้อมูลได้ 3 ชุด คือ

ก. พิมพ์โมดและเวลา (PRINT MODE AND TIME) เป็นโปรแกรมที่ส่งข้อมูลโมดและเวลาออกไปที่เครื่องพิมพ์ขณะที่กดปุ่มเปลี่ยนโมดการทำงานของระบบ ประกอบด้วยโปรแกรมย่อยส่งข้อมูลแสดงโมดของระบบ (SEND MODE) และส่งข้อมูลเวลา (SEND TIME) ดังแสดงโฟลว์ชาร์ทในรูปที่ 4.27

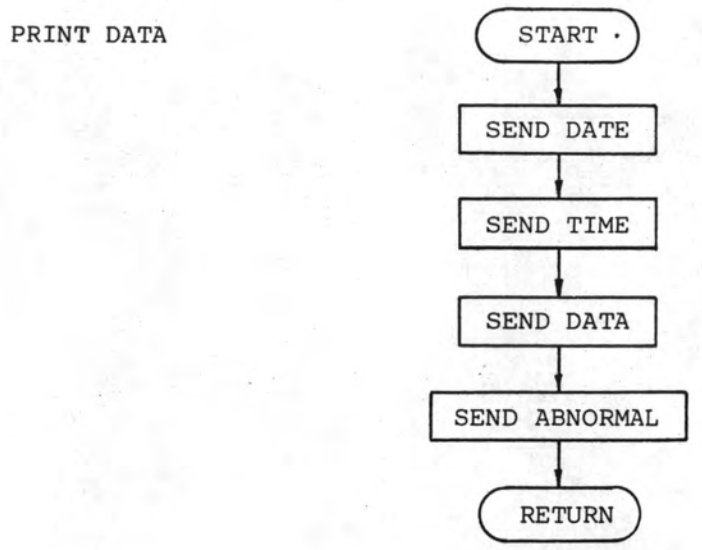
PRINT MODE AND TIME



รูปที่ 4.27 แสดงโฟลว์ชาร์ทของโปรแกรมพิมพ์โมดและเวลา

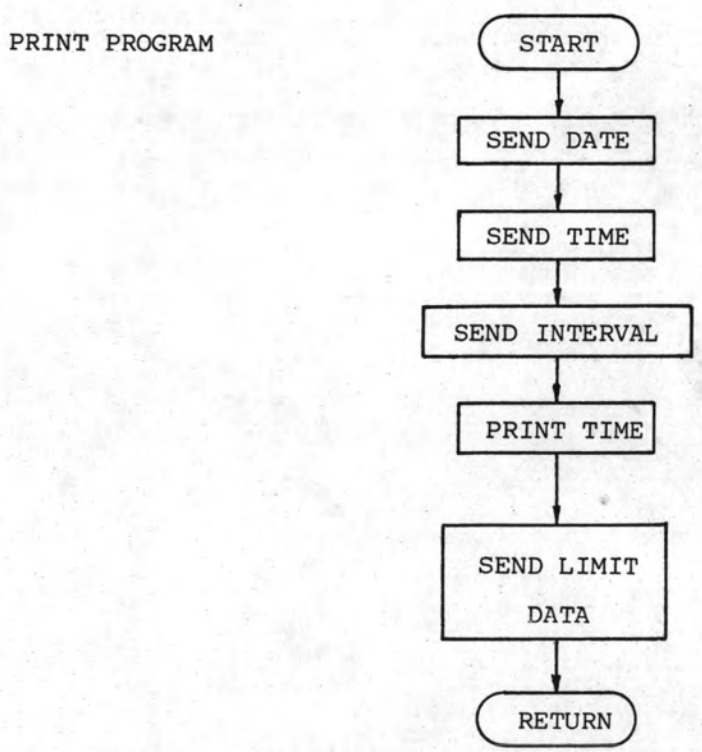
ข. พิมพ์ข้อมูลที่อ่านจากทรานสดิวเซอร์ (PRINT DATA) เป็นโปรแกรมที่ส่งข้อมูล วัน เดือน ปี เวลา ข้อมูลที่อ่านจากทรานสดิวเซอร์และข้อมูลที่ผิดปกติออกไปที่เครื่องพิมพ์ เมื่อได้รับคำสั่งการพิมพ์จากปุ่ม PRINT ใน HOLD MODE หรือเมื่อครบเวลาที่ตั้งไว้ใน RUN MODE ประกอบด้วยโปรแกรมย่อยส่งข้อมูล วัน เดือน ปี (SEND DATE) ส่งข้อมูลเวลา ส่งข้อมูลที่อ่านจากทรานสดิวเซอร์ (SEND DATA) ซึ่งประกอบด้วย หมายเลขช่องสัญญาณ ข้อมูล และหน่วยที่ใช้ ต่อจากนั้นจะเป็นโปรแกรมส่งข้อมูลที่ผิดปกติ (SEND ABNORMAL) ได้แก่ ข้อมูลที่เกินขีดกำหนดสูงหรือต่ำสุด ข้อมูลที่เกินช่วงการแปลงสัญญาณ เป็นต้น ดังแสดงโฟลว์ชาร์ทของโปรแกรมในรูปที่ 4.28

ค. พิมพ์ข้อมูลที่เซ็ทจากแป้นกดข้อมูล (PRINT PROGRAM) เป็นโปรแกรมที่ส่งข้อมูล วัน เดือน ปี เวลา และข้อมูลที่เซ็ทจากแป้นกดข้อมูล ออกไปที่เครื่องพิมพ์ เมื่อได้รับคำสั่งจากปุ่ม PRINT ใน PROGRAM MODE เท่านั้น ประกอบด้วยโปรแกรมย่อยส่งข้อมูล วัน เดือน ปี ส่งข้อมูลเวลา ส่งข้อมูลช่วงเวลาการพิมพ์ (SEND INTERVAL PRINT TIME) และส่งข้อมูลที่เซทค่าขีดกำหนดสูงสุดหรือต่ำสุด (SEND LIMIT DATA) ซึ่งประกอบด้วยหมายเลขช่องสัญญาณ ค่าสูงสุด ค่าต่ำสุด ชนิดของทรานสดิวเซอร์ และหน่วยที่ใช้ ดังแสดงโฟลว์ชาร์ทของโปรแกรมในรูปที่ 4.29



หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 4.28 แสดงโฟลว์ชาร์ทของโปรแกรมพิมพ์ข้อมูล
ที่อ่านจากทรานสดิวเซอร์

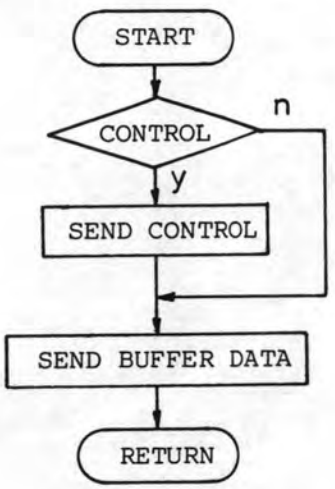


รูปที่ 4.29 แสดงโฟลว์ชาร์ทของโปรแกรมพิมพ์ข้อมูล
ที่เซ็ทจากแป้นกดข้อมูล

การส่งข้อมูลออกไปที่เครื่องพิมพ์ จะจัดข้อมูลออกเป็น 2 ชุด คือ ข้อมูลที่ต้องการพิมพ์ และคำสั่งควบคุมเครื่องพิมพ์ ซึ่งได้แก่การพิมพ์อักษรตัวใหญ่ หรือตัวสีแดง เป็นต้น โดยข้อมูลที่ต้อง

การพิมพ์จะถูกส่งออกไปครั้งละ 20 ไบท์ และคำสั่งควบคุม 3 ไบท์ ก่อนส่งข้อมูลแต่ละชุดจะต้องนำข้อมูลมาเปลี่ยนให้เป็นรหัส ASCII และบรรจุไว้ในหน่วยความจำชั่วคราวซึ่งใช้สำหรับส่งข้อมูล 20 ไบท์ จากนั้นจะบรรจุคำสั่งควบคุมไว้ 3 ไบท์ เมื่อบรรจุข้อมูลต่าง ๆ เรียบร้อยแล้วจะเรียกโปรแกรมส่งข้อมูลออกไปที่เครื่องพิมพ์ (SEND TO PRINTER) ดังแสดงโฟลว์ชาร์ทในรูปที่ 4.30 เมื่อเข้าสู่โปรแกรมนี้จะมีการตรวจดูว่าจะต้องส่งคำสั่งควบคุมหรือไม่ ถ้าต้องการก็จะส่งคำสั่งควบคุมไปก่อน (SEND CONTROL) จากนั้นจะส่งข้อมูลจริงตามไป (SEND BUFFER DATA)

SEND TO PRINTER

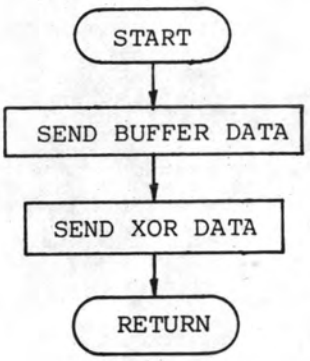


รูปที่ 4.30 แสดงโฟลว์ชาร์ทของโปรแกรมส่งข้อมูลไปที่เครื่องพิมพ์

4.2.8 โปรแกรมการส่งข้อมูลไปที่เครื่องไมโครคอมพิวเตอร์ เป็นโปรแกรมสำหรับจัดรูปแบบข้อมูลและส่งข้อมูลออกไปที่เครื่องไมโครคอมพิวเตอร์ รูปแบบหรือชุดของข้อมูลที่จะส่งไปมีทั้งหมด 2 ชุด คือ การส่งข้อมูลที่อ่านจากทรานสดิวเซอร์ และการส่งข้อมูลที่เซ็ทจากแป้นกดข้อมูล ซึ่งการส่งข้อมูลทั้งสองชนิดจะมีรูปแบบเช่นเดียวกับการส่งข้อมูลออกไปที่เครื่องพิมพ์ แต่ต่างกันที่วิธีการส่งเท่านั้น คือ จะเป็นการส่งข้อมูลแบบอนุกรมทีละบิทจนครบ 20 ไบท์ การส่งข้อมูลนี้จะส่งเฉพาะข้อมูลเท่านั้นไม่มีการส่งคำสั่งควบคุม แต่จะมีการส่งข้อมูลเพื่อตรวจสอบความถูกต้องของข้อมูลที่ส่งไปว่าถูกต้องหรือไม่ (SEND XOR DATA) ดังแสดงโฟลว์ชาร์ทของโปรแกรมในรูปที่ 4.31

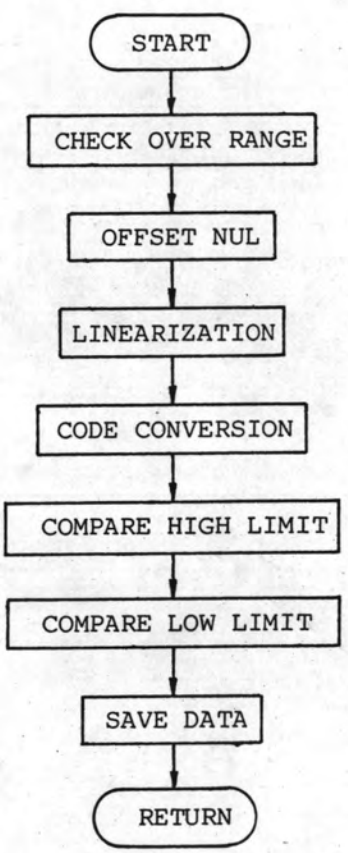
4.2.9 โปรแกรมการประมวลผลข้อมูล เป็นโปรแกรมที่กระทำเมื่อพบสภาวะของการแปลงสัญญาณเสร็จสิ้น โดยจะนำค่าข้อมูลที่ได้จากการแปลงสัญญาณและเก็บไว้ในหน่วยความจำมาทำการประมวลผล ซึ่งประกอบด้วยโปรแกรมย่อย คือ โปรแกรมตรวจสอบข้อมูลเกินช่วงการแปลงสัญญาณ (CHECK OVER RANGE) โปรแกรมการชดเชยออฟเซท (OFFSET NUL) โปรแกรมการ

SEND TO RS 232-C



รูปที่ 4.31 แสดงโฟลว์ชาร์ทของโปรแกรมส่งข้อมูล
ไปที่เครื่องไมโครคอมพิวเตอร์

DATA PROCESSING

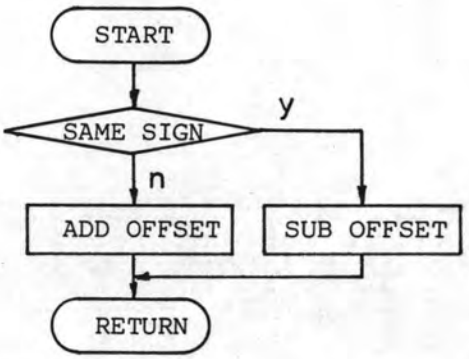


รูปที่ 4.32 แสดงโฟลว์ชาร์ทของโปรแกรมการประมวลข้อมูล

ทำให้เป็นเชิงเส้นตรง (LINEARIZATION) โปรแกรมแปลงค่าข้อมูลที่เป็นรหัสไบนารีให้เป็นรหัส BCD (CODE CONVERSION) โปรแกรมเปรียบเทียบค่าข้อมูลกับค่ากำหนดสูงสุด (COMPARE HIGH LIMIT) และค่ากำหนดต่ำสุด (COMPARE LOW LIMIT) จากนั้นจะนำข้อมูลที่ได้ไปเก็บไว้ในหน่วยความจำ (SAVE DATA) ดังแสดงโฟลว์ชาร์ทของโปรแกรมในรูปที่ 4.32

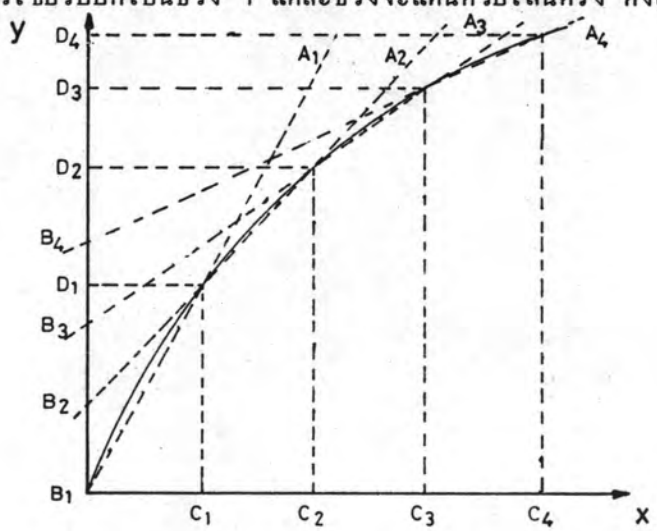
4.2.10 โปรแกรมการชดเชยออฟเซท เป็นโปรแกรมบวกลบข้อมูลอย่างง่าย ๆ โดยในแผงวงจรเลือกสัญญาณแต่ละแผงจะมีช่องสัญญาณหนึ่งต่อสัญญาณเข้าเป็นศูนย์ และจะอ่านข้อมูลนั้นเข้ามาเก็บไว้ ข้อมูลที่ได้จะเป็นค่าออฟเซทของภาคขยายสัญญาณในแผงวงจรนั้น การชดเชยออฟเซทจะทำโดยการนำค่าออฟเซทมาหักลบกับค่าข้อมูลที่อ่านได้ โดยมีการเปรียบเทียบเครื่องหมายของข้อมูลทั้งสองชนิด ถ้าเครื่องหมายเหมือนกันจะต้องนำค่าออฟเซทมาลบออก (SUB OFFSET) แต่ถ้าเครื่องหมายต่างกันจะนำมาบวกกัน (ADD OFFSET) ดังแสดงโฟลว์ชาร์ทในรูปที่ 4.33

OFFSET NUL



รูปที่ 4.33 แสดงโฟลว์ชาร์ทของโปรแกรมการชดเชยออฟเซท

4.2.11 โปรแกรมการทำให้เป็นเชิงเส้นตรง เป็นโปรแกรมที่ใช้แก้ความไม่เป็นเชิงเส้นตรงของทรานสดิวเซอร์ ดังนั้นจะต้องทราบความสัมพันธ์เชิงเส้นของทรานสดิวเซอร์นั้น การทำเป็นเชิงเส้นตรงจะใช้วิธี PIECEWISE LINEAR APPROXIMATION โดยจะแบ่งความสัมพันธ์เชิงเส้นของทรานสดิวเซอร์ออกเป็นช่วง ๆ แต่ละช่วงจะแทนด้วยเส้นตรง ดังแสดงในรูปที่ 4.34

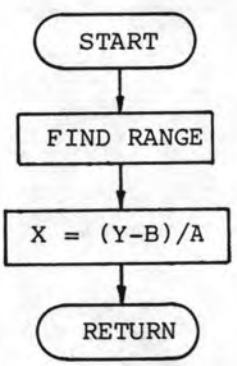


รูปที่ 4.34 แสดงการแทนความสัมพันธ์เชิงเส้นของทรานสดิวเซอร์
ด้วยเส้นตรงเป็นช่วง ๆ

โดยที่เส้นตรงแต่ละเส้นจะมีความชัน (Slope) A = $A_1, A_2 \ldots, A_n$ มีจุดตัดแกนสัญญาณออกหรือแกน Y ที่ B = $B_1, B_2, \ldots B_n$ ช่วงตัดกันของเส้นตรงแต่ละเส้นจะอยู่ที่ C = $C_1, C_2, \ldots C_n$ โดยแต่ละจุดของ C จะมีค่าสัญญาณออกเป็น D = $D_1, D_2, \ldots, D_n$ สมการของแต่ละเส้น คือ Y = AX+B หรือ X = (Y-B)/A ความละเอียดในการแบ่งช่วงการประมาณค่า อย่างน้อยที่สุดค่า Y จะต้องน้อยกว่า 0.1% ของค่าสูงสุดในการวัดเพื่อให้ได้ความแม่นยำตามข้อกำหนด

การสร้างโปรแกรมการทำให้เป็นเชิงเส้นตรง จะต้องนำค่า A B และ D มาเก็บไว้เป็นชุด ๆ เมื่อเข้าสู่การทำงานของโปรแกรม ค่าที่อ่านได้จากทรานสดิวเซอร์ซึ่งคือค่า Y จะถูกนำไปเปรียบเทียบหาช่วงของข้อมูล (FIND RANGE) ว่าตกอยู่ในช่วงใด เมื่อพบช่วงของข้อมูลแล้ว ก็จะทราบค่า A และ B ของช่วงเส้นตรงนั้น จะหาค่าข้อมูลที่ถูกต้องคือ X ได้โดยการนำค่า B ไปลบออกจาก Y และผลลัพธ์ที่ได้หารด้วย A (X = (Y-B)/A) ข้อมูลที่ได้จะเป็นข้อมูลที่ผ่านการทำให้เป็นเชิงเส้นตรงแล้ว ดังแสดงโฟลว์ชาร์ทในรูปที่ 4.35

LINEARIZATION

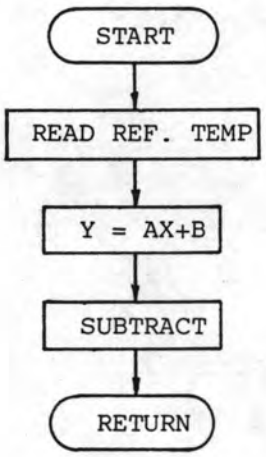


รูปที่ 4.35 แสดงโฟลว์ชาร์ทของโปรแกรมการทำให้เป็นเชิงเส้นตรง

4.2.12 โปรแกรมชดเชยอุณหภูมิที่จุดอ้างอิง ใช้สำหรับทรานสดิวเซอร์ที่เป็นเทอร์โมคับเปิลเท่านั้น เนื่องจากที่จุดอ้างอิงไม่ได้รักษาอุณหภูมิไว้ที่ 0 °C จะมีแรงเคลื่อนไฟฟ้าเกิดขึ้นที่จุดอ้างอิงนั้น ดังนั้นจะต้องทราบความสัมพันธ์เชิงเส้นของแรงดันที่เกิดขึ้นที่จุดอ้างอิงกับอุณหภูมิที่จุดนั้นและใช้วิธีประมาณค่าแบบ PIECEWISE LINEAR APPROXIMATION เช่นเดียวกัน แต่การสร้างโปรแกรมจะต้องนำค่า A B และ C มาเก็บไว้เป็นชุด และจะต้องมีช่องสัญญาณหนึ่งวัดค่าอุณหภูมิที่จุดอ้างอิงไว้โดยใช้ทรานสดิวเซอร์ชนิดอื่น เช่น กระเปาะความต้านทาน หรือ เทอร์มีสเตอร์

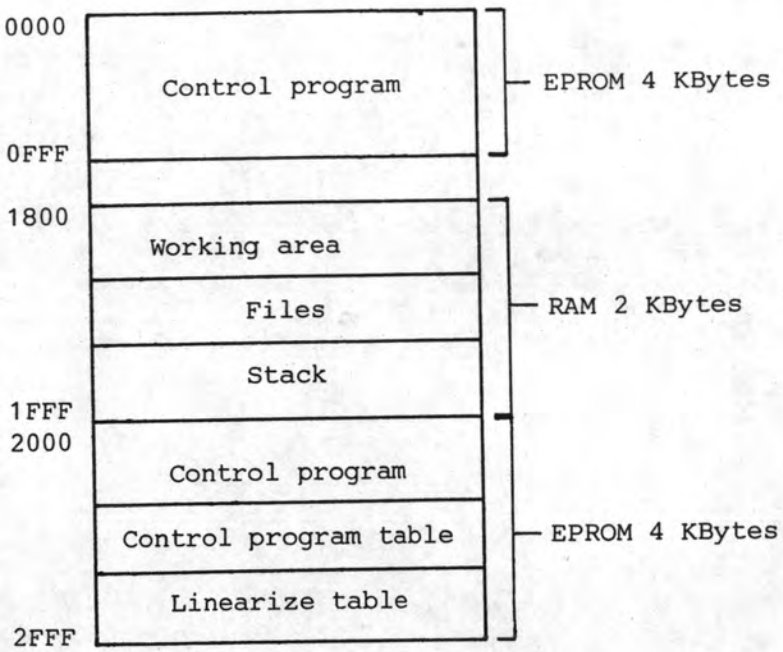
เป็นต้น ค่าอุณหภูมิที่จุดอ้างอิงนี้จะถูกอ่านเข้ามา (READ REF. TEMP) คือค่า X เพื่อจะนำไปคำนวนหาค่าแรงเคลื่อนไฟฟ้าที่เกิดขึ้น โดยการเปรียบเทียบหาช่วงอุณหภูมิ เพื่อจะได้ทราบค่า A และ B ค่าแรงเคลื่นที่ต้องการคือ Y ซึ่งหาได้โดยการนำค่า A ไปคูณกับ X และบวกด้วยค่า B (Y = AX+B) จากนั้นจะนำค่าแรงเคลื่อนที่ได้ ไปลบออกจากข้อมูลที่อ่านได้จากทรานสดิวเซอร์ (SUBTRACT) ข้อมูลที่ได้จะเป็นข้อมูลที่ชดเชยอุณหภูมิที่จุดอ้างอิงแล้ว ดังแสดงโฟลว์ชาร์ทของโปรแกรมในรูปที่ 4.36

COLD JUNCTION COMPENSATION



รูปที่ 4.36 แสดงโฟลว์ชาร์ทของโปรแกรมการชดเชยอุณหภูมิที่จุดอ้างอิง

4.2.13 MEMORY MAP เป็นแผนผังการใช้หน่วยความจำของระบบทั้งหมด ดังแสดงในรูปที่ 4.37 ซึ่งประกอบด้วยหน่วยความจำที่เป็นโปรแกรมควบคุม เก็บไว้ใน EPROM เริ่มต้นจากแอดเดรส 0000H ถึง 0FFFH และ 2000H ถึง 2FFFH ประกอบด้วยโปรแกรม 3 ส่วน คือ โปรแกรมควบคุม (Control program) ตารางสำหรับโปรแกรมควบคุม (Control program table) และตารางสำหรับการทำเป็นเชิงเส้นตรง (Linearize table) ของทรานสดิวเซอร์ชนิดต่าง ๆ สำหรับการเก็บข้อมูลชั่วคราว จะเก็บไว้ใน RAM เริ่มต้นจากแอดเดรส 1800H ถึง 1FFFH ประกอบด้วยเนื้อที่ 3 ส่วน คือ เนื้อที่ใช้งานของโปรแกรมควบคุม (Working area) ได้แก่การเก็บสภาวะต่าง ๆ ของระบบ เนื้อที่สำหรับเก็บข้อมูลที่ได้จากการวัด และการเซทค่าจากแป้นกดข้อมูล (Files) เนื้อที่ที่เหลือ ใช้สำหรับเป็นสแตค (Stack) ของระบบ



รูปที่ 4.37 แสดง MEMORY MAP ของระบบ

4.2.14 I/O MAP เป็นแผนผังของอุปกรณ์การอินเตอร์เฟสต่าง ๆ เช่น PIA 8255 PIO CTC และ UART 8251 เป็นต้น โดยแต่ละตัวจะมีแอดเดรสของตนเอง เมื่อ CPU ต้องการติดต่อรับส่งข้อมูลจะต้องติดต่อโดยการส่งสัญญาณออกมาที่แอดเดรสที่ถูกต้อง ตามแอดเดรสดังแสดงในรูปที่ 4.38 ระบบจึงสามารถทำงานตามข้อกำหนดที่ออกแบบไว้ได้

4.3 การสร้างโปรแกรมควบคุมการทำงาน

โปรแกรมที่เขียนขึ้นเป็นภาษาแอสเซมบลีจากโฟลว์ชาร์ท จะต้องทำการแปลเป็นภาษาเครื่องก่อนที่จะนำไปใช้งาน โดยใช้โปรแกรมการพัฒนาระบบของเครื่องไมโครคอมพิวเตอร์ช่วยในการแปลเป็นภาษาเครื่อง เมื่อได้โปรแกรมที่เป็นภาษาเครื่องแล้ว จะต้องนำไปอัดเก็บไว้ใน EPROM เพื่อใช้ควบคุมระบบ แต่การพัฒนาระบบจะต้องมีการแก้ไขโปรแกรมบ่อยครั้ง การโปรแกรมภาษาเครื่องลง EPROM โดยตรงจะไม่สะดวก ดังนั้นจึงได้ออกแบบสร้างแผงวงจรซึ่งทำหน้าที่แทน ROM สามารถเปลี่ยนแปลงค่าได้ง่าย ทำให้การพัฒนาระบบทำได้รวดเร็วขึ้น

I/O address	detail	device
40	CTC 0 Clock divider for UART	CTC
41	CTC 1 A/D Interrupt	
42	CTC 2 UART Interrupt	
43	CTC 3 Scan Keyboard and display	
80	PIO DA Low byte data from A/D	PIO
81	PIO DB High byte data from A/D	
82	PIO CA	
83	PIO CB	
CO	Start A/D	555
C8	Port A	8255
C9	Port B Data to scanner	
CA	Port C Data to alarm	
CB	Control	
D0	Port A Data to printer	8255
D9	Port B Clock control and Printer control	
D2	Port C Data to clock and Printer control	
D3	Control	
D8	Port A Keyboard data in	8255
D9	Port B 7 segment data out to display	
DA	Port C Digit select and keyboard scan	
DB	Control	
F8	Control UART	8251
F9	Data	

รูปที่ 4.38 แสดง I/O MAP ของระบบ

4.3.1 การพัฒนาระบบ (System development) เป็นการพัฒนาการเขียน ตรวจสอบ และ การเพิ่มความสามารถการทำงานให้กับระบบ โดยเขียนเป็นภาษาแอสเซมบลี แล้วแปลภาษาแอสเซมบลีเป็นภาษาเครื่อง ให้สามารถใช้งานได้ง่าย และรวดเร็วขึ้น การพัฒนาระบบนี้ได้มีผู้เขียนโปรแกรมที่ใช้สำหรับช่วยในการพัฒนาระบบ โดยใช้กับเครื่องไมโครคอมพิวเตอร์ขึ้นเป็นจำนวนมาก ซึ่งแต่ละโปรแกรมจะมีวิธีใช้ที่แตกต่างกัน แต่จะใช้รูปแบบภาษาแอสเซมบลี ที่เป็นมาตรฐานเดียวกัน ในที่นี้จะกล่าวถึงตัวอย่างโปรแกรมที่ใช้ในการทำวิทยานิพนธ์เท่านั้น คือ

ก. ALDS เป็นชุดโปรแกรมที่เขียนขึ้นโดยบริษัท MICROSOFT ซึ่งเขียนขึ้นภายใต้ CP/M โดยใน ALDS จะมีโปรแกรมใช้งาน (Utility program) ต่าง ๆ ที่ช่วยในการพัฒนาระบบคือ

1) M80.COM ใช้ในการแปลภาษาแอสเซมบลีให้เป็นภาษาเครื่อง สามารถใช้กับ CPU ที่เป็น 6502 8080 และ Z80 โปรแกรมภาษาเครื่องที่ได้สามารถจะกำหนดตำแหน่งใด ๆ ในหน่วยความจำ ตามความเหมาะสมของโปรแกรมที่สร้างมาใช้งาน ในภายหลังได้

2) L80.Com ใช้สำหรับกำหนดตำแหน่งในหน่วยความจำต่าง ๆ ของโปรแกรมภาษาเครื่องที่ได้จากการแปลโดย M80.COM นอกจากนั้นยังสามารถต่อ (Link) โปรแกรมย่อยต่าง ๆ เข้าด้วยกันโดยมีตำแหน่งในหน่วยความจำสัมพันธ์กัน

3) CREF80.COM ใช้สำหรับสร้าง CROSS-REFFERENCE LISTING ของสัญญลักษณ์ (Symbols) ต่าง ๆ ในโปรแกรม ซึ่งจะบอกตัวแปรต่าง ๆ ในโปรแกรมว่ามีอะไรบ้าง ถูกเรียกใช้ที่ไหนบ้าง ซึ่งเป็นประโยชน์มากสำหรับการตรวจเช็คโปรแกรม

นอกจากโปรแกรมที่กล่าวมาแล้ว ยังมีโปรแกรมใช้งานอื่นอีก เช่น LIB80.COM DDT65.COM เป็นต้น ซึ่งมีประโยชน์มากสำหรับโปรแกรมที่มีขนาดใหญ่มาก ๆ และยุ่งยากซับซ้อน

ข. SC-MACRO ASSEMBLER เป็นโปรแกรมที่เขียนขึ้นโดยบริษัท SC-SOFTWARE CORPORATION ซึ่งเขียนขึ้นภายใต้ DOS คำสั่งต่าง ๆ ที่ใช้ในโปรแกรมจะคล้ายกับ คำสั่งในโปรแกรมภาษาเบสิค และมีคำสั่งพิเศษเพิ่มขึ้นมาเพื่อช่วยให้สะดวกในการเขียน แก้ไขโปรแกรมและแปลภาษาแอสเซมบลีที่เขียนขึ้นให้เป็นภาษาเครื่องได้ สามารถสรุปคำสั่งต่าง ๆ สำหรับใช้งานได้ 5 กลุ่ม คือ

1) SOURCE COMMAND เป็นคำสั่งที่ใช้กับไฟล์ (File) ได้แก่ NEW LOAD SAVE TEXT HIDE MERGE RESTORE เป็นต้น

2) EDITING COMMAND เป็นคำสั่งที่ใช้สำหรับ เขียน แก้ไข ข้อความต่าง ๆ ในไฟล์ ได้แก่ EDIT COPY LIST FIND REPLACE DELETE RENUMBER เป็นต้น

3) LISTING COMMAND เป็นคำสั่งที่ใช้สำหรับ ควบคุมการตรวจดู ข้อความต่าง ๆ ในไฟล์ ได้แก่ SLOW FAST PRT เป็นต้น

4) OBJECT COMMAND เป็นคำสั่งที่ใช้เกี่ยวกับการแปลเป็นภาษาเครื่อง และเอกซิคิว (Execute) โปรแกรมได้ ได้แก่ ASM MGO VAL SYMBOLS เป็นต้น

5) MISCELLENEOUS เป็นคำสั่งทั่ว ๆ ไป ที่ใช้ในการจัดรูปแบบ หรือ ช่วยในการอำนวยความสะดวกในการเขียนโปรแกรมได้แก่ AUTO MANUAL INCREMENT MEMORY RST MWTR USR เป็นต้น

สำหรับโปรแกรม SC-MACRO ASSEMBLER จะใช้ในการพัฒนาโปรแกรมให้กับเครื่องไมโครคอมพิวเตอร์ ในการรับส่งข้อมูลแบบอนุกรมกับระบบเก็บข้อมูลแบบอะนาลอก โดยที่จะเขียนเป็นโปรแกรมย่อย ๆ ด้วยภาษาแอสเซมบลี และแปลเป็นภาษาเครื่องด้วยโปรแกรม SC-MACRO ASSEMBLER และเก็บไว้ในไฟล์ โปรแกรมย่อยนี้จะเป็นโปรแกรมควบคุมอุปกรณ์การรับส่งข้อมูลของเครื่องไมโครคอมพิวเตอร์ สำหรับโปรแกรมที่ใช้งานจริงจะเขียนเป็นโปรแกรมภาษาเบสิคซึ่งจะเรียกโปรแกรมจากไฟล์เหล่านั้นมาใช้อีกทีหนึ่ง

4.3.2 การพัฒนาโปรแกรมโดยใช้ ALDS โปรแกรมที่เขียนเป็นภาษาแอสเซมบลีจะต้องป้อนเข้าเครื่องไมโครคอมพิวเตอร์ในรูปของไฟล์ (File) แล้วจึงนำไฟล์ที่ได้มาทำการแปลให้เป็นภาษาเครื่อง ขั้นตอนการพัฒนาระบบโดยใช้ ALDS มีดังนี้คือ

ก. สร้างไฟล์โปรแกรมภาษาแอสเซมบลี การสร้างไฟล์ภาษาแอสเซมบลีสำหรับใช้กับ ALDS นั้น จะใช้โปรแกรมที่เป็น EDITOR ที่เป็น CP/M ใด ๆ ก็ได้ แต่ในการทำวิทยานิพนธ์ครั้งนี้จะใช้โปรแกรม WORD STAR ซึ่งเป็นโปรแกรมในการทำ WORD PROCESSING ไฟล์ที่สร้างขึ้นจะมี .EXT เป็น .MAC คือ MACRO FILE การป้อนโปรแกรมแต่ละบรรทัดจะแบ่งเป็นฟิลด์เช่นเดียวกับรูปแบบภาษาแอสเซมบลี แต่ละฟิลด์จะต้องห่างกันไม่น้อยกว่าหนึ่งตัวอักษร และฟิลด์หมายเหตุจะต้องเริ่มต้นด้วย ";" เมื่อเขียนโปรแกรมจบจะต้องปิดท้ายด้วยคำสั่งเทียม END เสมอ

โปรแกรมที่สร้างขึ้นจะสร้างเป็นโมดูล (Module) แต่ละโมดูลจะเป็นโปรแกรมย่อย ๆ ซึ่งสร้างขึ้นและทำการทดสอบแก้ไขจนสามารถทำงานได้ การรวมแต่ละโมดูลเข้าด้วยกัน ทำได้ 2 แบบคือ การใช้คำสั่งเทียม INCLUDE และตามด้วยชื่อของโมดูล เมื่อใช้ M80.COM ทำการแปลเป็นภาษาเครื่อง จะได้ไฟล์ที่ยังไม่ได้กำหนดแอดเดรสแน่นอนเพียงไฟล์เดียว เป็นไฟล์ที่มี .EXT เป็น .REL คือ RELOCATABLE FILE การรวมโมดูลแบบนี้จะใช้รวมโมดูลย่อย ๆ เข้าด้วยกันเป็นโมดูลใหญ่ ชื่อตัวแปรแต่ละโมดูลจะซ้ำกันไม่ได้ การรวมโมดูลอีกแบบหนึ่ง คือ การรวมโมดูลที่เป็น RELOCATABLE FILE เข้าด้วยกัน โดยใช้ L80.COM แบบนี้จะใช้ในการรวมโมดูลใหญ่ ๆ เข้าด้วยกัน ซึ่งชื่อในแต่ละโมดูลสามารถซ้ำกับโมดูลอื่นได้ แต่ถ้าจะใช้ชื่อร่วมกันจะต้องใช้คำสั่งเทียม PUBLIC และ EXTERNAL กำหนดไว้ การเขียนโปรแกรมเป็นโมดูลนี้ ทำให้สะดวกในการตรวจสอบและแก้ไขโปรแกรมได้ง่าย เนื่องจากแต่ละโมดูลจะมีขนาดเล็กและมีฟังก์ชันแน่นอน

ข. แปลภาษาแอสเซมบลีให้เป็นภาษาเครื่อง โปรแกรม M80.COM มีหน้าที่ในการแปล MACRO FILE ให้เป็น RELOCATABLE FILE การใช้โปรแกรม M80.COM ทำการแปลไฟล์ที่มีชื่อ PROGRAM1.MAC ทำได้โดยการป้อนโปรแกรม ดังนี้คือ

```
A> M80 (RETURN)
 * =PROGRAM1/C (RETURN)
```

เมื่อการแปลสิ้นสุดลงแล้ว ถ้ามีข้อบกพร่อง (Error) ภายในโปรแกรม เช่น เขียนผิดรูปแบบภาษา หรือใช้คำสั่งผิด เป็นต้น จะมีการแสดงบนจอ CRT เกี่ยวกับบรรทัดที่มีข้อบกพร่องนั้น รหัสที่บอกข้อบกพร่อง และจำนวนจุดที่มีข้อบกพร่อง แต่ถ้าไม่มีข้อบกพร่อง จะมีการแสดงคำว่า No Fatal Error (s) หลังการแปลจะได้ไฟล์เพิ่มขึ้น 2 ไฟล์ คือ RELOCATABLE FILE และ CROSS REFERENCE FILE ซึ่งมี .EXT เป็น .CRF โดยมีชื่อไฟล์เป็น PROGRAM1 เช่นเดียวกัน โดยที่ CROSS REFERENCE FILE เกิดจากการป้อน /C ตามชื่อไฟล์นั้น ซึ่งไฟล์นี้จะใช้สำหรับสร้าง CROSS REFERENCE LISTING ในโปรแกรม CREF 80.COM

ค. การแปล RELOCATABLE FILE ให้เป็น COMMAND FILE เนื่องจาก RELOCATABLE FILE เป็นภาษาเครื่องที่ยังไม่ระบุตำแหน่งแอดเดรสระหว่างโมดูลให้แน่นอนจะต้องนำ RELOCATABLE FILE หลาย ๆ ไฟล์มารวมกันโดยใช้โปรแกรม L80.COM โดยมีแอดเดรสหรือ

ค่าต่าง ๆ สัมพันธ์กัน ซึ่งสามารถนำไปใช้งานได้ เรียกว่า COMMAND FILE ซึ่งมี .EXT เป็น .COM การใช้โปรแกรม L80.COM ทำการแปลและรวมไฟล์ที่มีชื่อ PROGRAM1.REL PROGRAM2 .REL PROGRAM3.REL ทำได้โดยการป้อนโปรแกรมดังนี้ คือ

```
A> L80 (RETURN)
 * PROGRAM1,PROGRAM2,PROGRAM3,PROGRAM/N/E (RETURN)
```

หลังจากการทำโปรแกรม L80.COM ถ้ามีข้อบกพร่อง จะแสดงออกที่จอ CRT เกี่ยวกับข้อบกพร่องต่าง ๆ ภายในไฟล์ เช่นหาไฟล์ไม่พบหรือแอดเดรสของแต่ละโมดูลซ้อนกัน เป็นต้น ถ้าไฟล์ถูกต้องจะมีการแสดงเฉพาะจำนวนหน่วยความจำเท่านั้น สำหรับไฟล์ที่ได้ใหม่ คือ PROGRAM.COM ซึ่งคือชื่อไฟล์ที่ตามด้วย/Nนั่นเอง สำหรับ/E จะเป็นการกำหนดให้โปรแกรมทำงานแปลและรวมไฟล์ แล้วเก็บไว้ในไฟล์ PROGRAM.COM เท่านั้น จะไม่มีการเอกซิคิวโปรแกรมในไฟล์นั้น

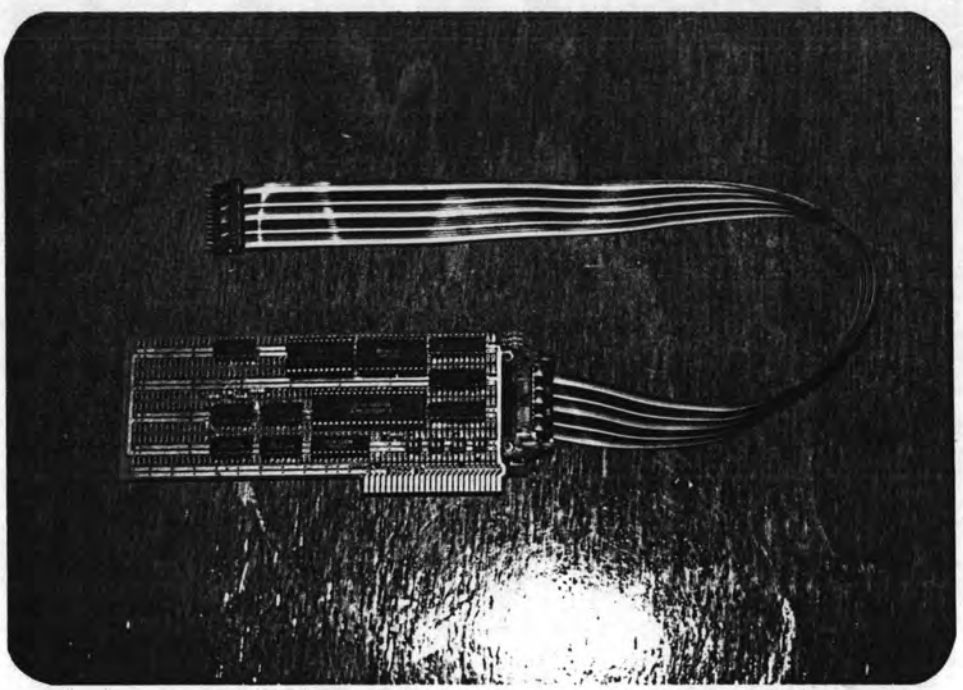
ง. การสร้าง CROSS REFERENCE LISTING เป็นการนำไฟล์ที่เป็น CROSS REFERENCE FILE ซึ่งประกอบด้วยโปรแกรมภาษาแอสเซมบลี รหัสโปรแกรมภาษาเครื่อง และค่าสัญญลักษณ์ต่าง ๆ มาผ่านโปรแกรม CREF80.COM เพื่อสร้าง CROSS REFERENCE LISTING ซึ่งจะบอกค่าตัวแปรต่าง ๆ มีอะไรบ้าง ถูกเรียกใช้ที่ไหนบ้าง และนำออกแสดงทางจอ CRT หรือให้พิมพ์ออกที่เครื่องพิมพ์ ซึ่งเป็นประโยชน์มากในการตรวจสอบและแก้ไขโปรแกรม การใช้โปรแกรม CREF80.COM กับไฟล์ที่มีชื่อ PROGRAM1.CRF สำหรับแสดงผลบนจอ CRT ทำได้โดยการป้อนโปรแกรมดังนี้ คือ

```
A> CREF80 (RETURN)
 * TTY:=PROGRAM1 (RETURN)
```

และให้แสดงผลออกทางเครื่องพิมพ์ทำได้โดยการป้อนโปรแกรมดังนี้คือ

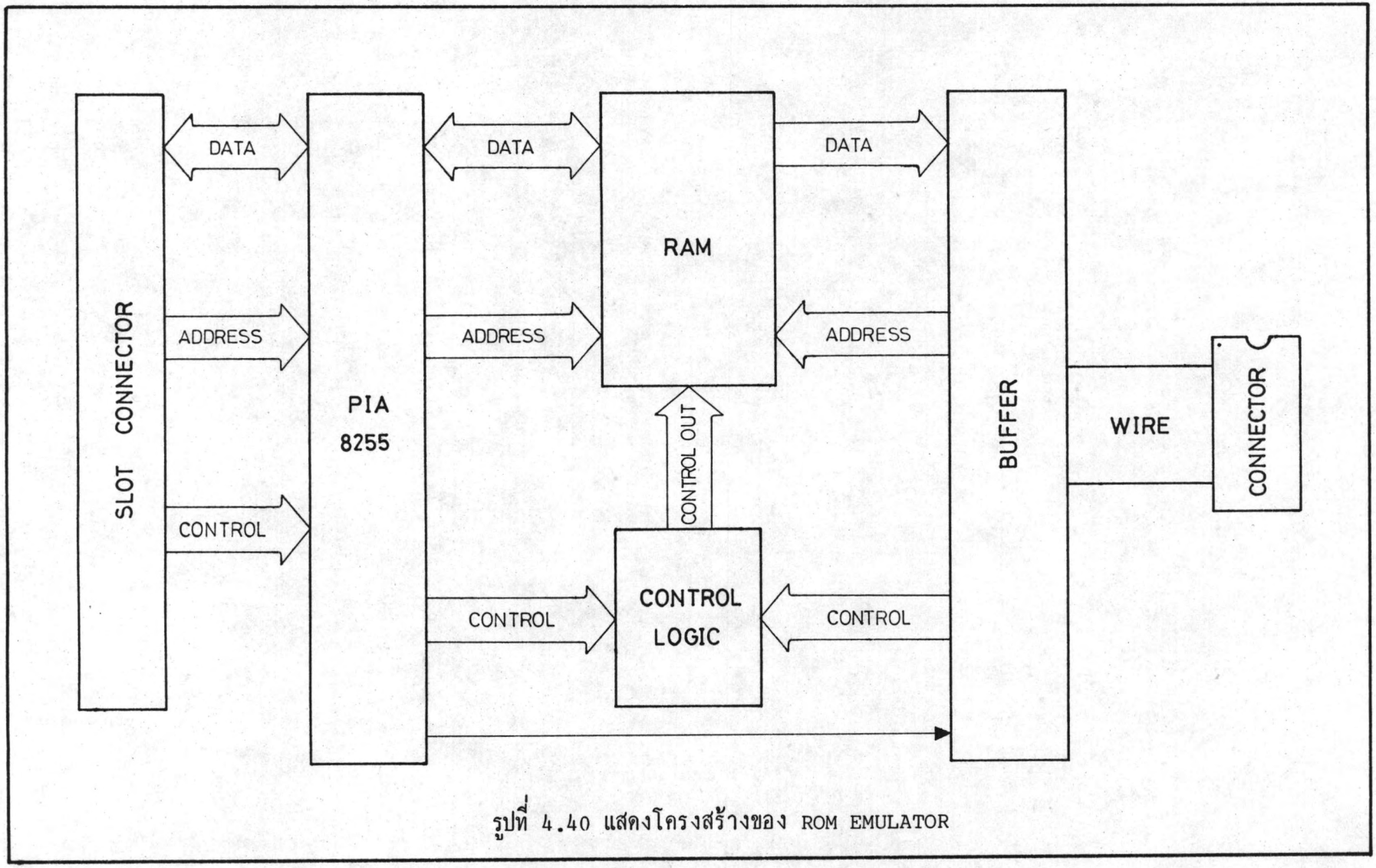
```
A> CREF80 (RETURN)
 * LST:=PROGRAM1 (RETURN)
```

4.3.3 ROM EMULATOR เป็นแผงวงจรดังแสดงในรูปที่ 4.39 ซึ่งมีคอนเนคเตอร์ออกทำหน้าที่เป็น ROM และสามารถโปรแกรมค่าต่าง ๆ ในหน่วยความจำของแผงวงจรโดยต่อเข้ากับสลอตคอนเนคเตอร์ (Slot connector) ของเครื่องไมโครคอมพิวเตอร์ โครงสร้างของ ROM EMULATOR ดังแสดงในรูปที่ 4.40 ประกอบด้วย PIA 8255 ซึ่งรับส่งข้อมูลกับเครื่องไมโครคอมพิวเตอร์ และมีพอร์ทต่าง ๆ ต่อกับหน่วยความจำ RAM ซึ่งพอร์ทเหล่านี้จะถูกแบ่งออกเป็นบัสข้อมูล บัสแอดเดรส บัสควบคุม สำหรับ RAM ในการอ่านและเขียนข้อมูลจากเครื่องไมโครคอมพิวเตอร์ ที่คอนเนคเตอร์ออกของแผงวงจรจะมีบัฟเฟอร์ ซึ่งต่อเข้ากับบัสต่าง ๆ ของ RAM เช่นกัน วงจรภายนอกสามารถอ่านข้อมูลได้อย่างเดียว การอ่านและเขียน RAM จะถูกควบคุมด้วยวงจร CONTROL LOGIC ซึ่งจะควบคุม RAM ให้ติดต่อกับเครื่องไมโครคอมพิวเตอร์ทาง PIA 8255 หรือ ติดต่อกับระบบที่ต้องการใช้ ROM ทางบัฟเฟอร์

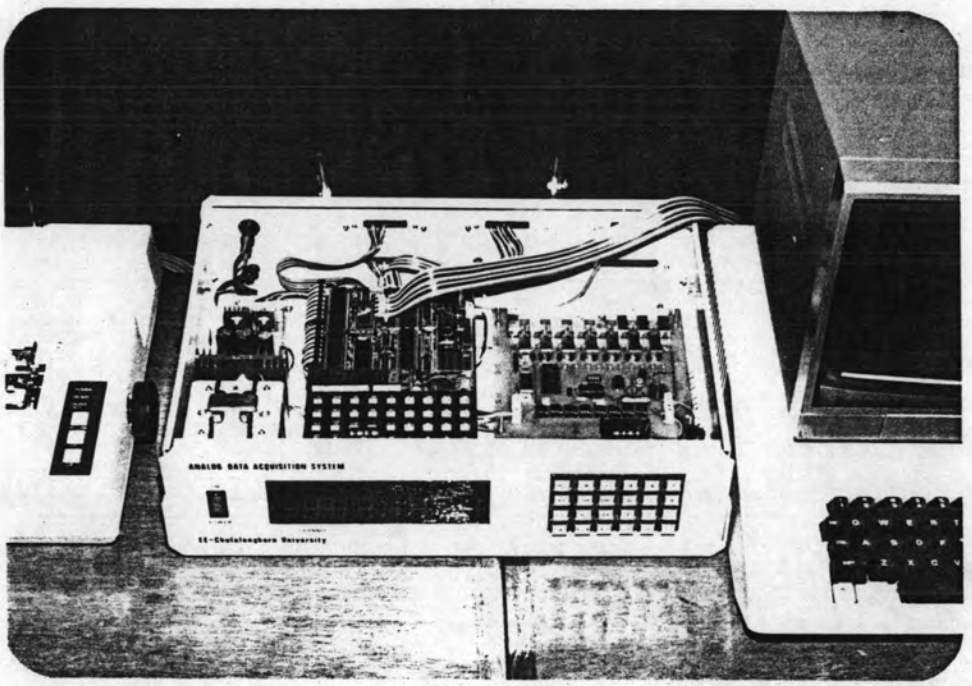


รูปที่ 4.39 แสดงแผงวงจร ROM EMULATOR

การใช้งาน ROM EMULATOR ทำได้โดยการนำแผงวงจรไปเสียบไว้ในเครื่องไมโครคอมพิวเตอร์ APPLE II สำหรับคอนเนคเตอร์ออกไปเสียบไว้แทน ROM ของระบบดังแสดงในรูปที่ 4.41 จากนั้นจะต้องเขียนโปรแกรมให้กับเครื่องไมโครคอมพิวเตอร์ เป็นโปรแกรมง่าย ๆ ใน



รูปที่ 4.40 แสดงโครงสร้างของ ROM EMULATOR



รูปที่ 4.41 แสดงการใช้งานของ ROM EMULATOR

การถ่ายข้อมูลที่อยู่ในหน่วยความจำของเครื่องไมโครคอมพิวเตอร์ลงไปไว้ใน RAM ของแผงวงจรนั้น ระบบเก็บข้อมูลแบบอะนาลอกจะเห็น RAM ในแผงวงจร ROM EMULATOR เป็นเสมือน ROM ของระบบ และสามารถทำงานได้ตามปกติ ถ้าจะแก้ไขโปรแกรมจะต้องแก้ไขจากหน่วยความจำของเครื่องไมโครคอมพิวเตอร์ และถ่ายข้อมูลเข้าไปอีก ทำให้ง่ายต่อการแก้ไขโปรแกรม ไม่ต้องเสียเวลาอัดและลบ EPROM ซึ่งมีขั้นตอนยุ่งยากมากกว่า และทำให้อายุการใช้งานของ EPROM สั้นลง โปรแกรมต่าง ๆ จะถูกแก้ไข และทดสอบบน ROM EMULATOR จนกระทั่งทำงานถูกต้องสมบูรณ์ทั้งระบบ จึงนำโปรแกรมนั้นไปอัดไว้ใน EPROM สำหรับใช้งานจริง