

บทที่ ๓

รายละเอียดของเครื่องไมโครคอมพิวเตอร์ที่ใช้ในการวิจัย

การวิจัยนี้ใช้เครื่องไมโครคอมพิวเตอร์ของบริษัทยูนิเทค จำกัด รุ่น CHERRY III ซึ่งเป็นเครื่อง APPLE COMPATIBLE มีหลักการทำงานเช่นเดียวกับเครื่อง APPLE II ภาษาที่ใช้ในการเขียนโปรแกรม คือ APPLESOFT BASIC

ส่วนประกอบและรายละเอียดของเครื่องที่ใช้ในการวิจัย

ส่วนประกอบของเครื่องไมโครคอมพิวเตอร์ รุ่น CHERRY III และอุปกรณ์อื่น ๆ คือ

1. เครื่องไมโครคอมพิวเตอร์ คือหน่วยประมวลผลกลาง (Central Processing Unit) หรือ CPU ใช้ไมโครโพรเซสเซอร์ เบอร์ 6502 และ Z - 80 มีหน่วยความจำทั้งสิ้น 64 K bytes 8 bits/word และมีความเร็วนาฬิกา 1.023 MHz

2. แป้นกดข้อมูล (Keyboard) มีตัวกด (keys) ต่าง ๆ รวม 95 คีย์ เป็นฟังก์ชันคีย์ 12 ฟังก์ชันคีย์ มี Auto Repeat ในตัว caps lock เลือกตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก คีย์ภาษาไทย และมี Buzzer ในตัว เพื่อเน้นการกดคีย์ต่าง ๆ และมีฟังก์ชันพิเศษ 12 ฟังก์ชัน ทำหน้าที่ต่าง ๆ ดังนี้

F ₁	=	CATALOG
F ₂	=	RUN
F ₃	=	CALL - 151
F ₄	=	LIST
F ₅	=	PRINT
F ₆	=	PR #

F ₇	=	LOAD
F ₈	=	SAVE
F ₉	=	POKE
F ₁₀	=	PEEK
F ₁₁	=	GOTO
F ₁₂	=	TEXT

3. เครื่องขับจานแม่เหล็ก (Disk Drive) เป็นชนิดคู่ จานแม่เหล็กแต่ละตัวมีความจุ 163 K bytes โดยใช้ระบบ DISK OPERATION SYSTEM (DOS) VERSION 3.3 ซึ่งเป็นระบบที่ใช้ควบคุมการทำงานระหว่างไมโครคอมพิวเตอร์และเครื่องขับจานแม่เหล็ก (Disk Drive) เพื่อให้มีการจัดเรียงลำดับข้อมูล และเก็บรักษาลำดับข้อมูลลงบนแผ่นจานข้อมูล (diskette) ซึ่งเรียกว่า disk format ทำให้สามารถเรียกข้อมูลที่เก็บไว้ออกมาใช้ในภายหลังอย่างมีประสิทธิภาพ

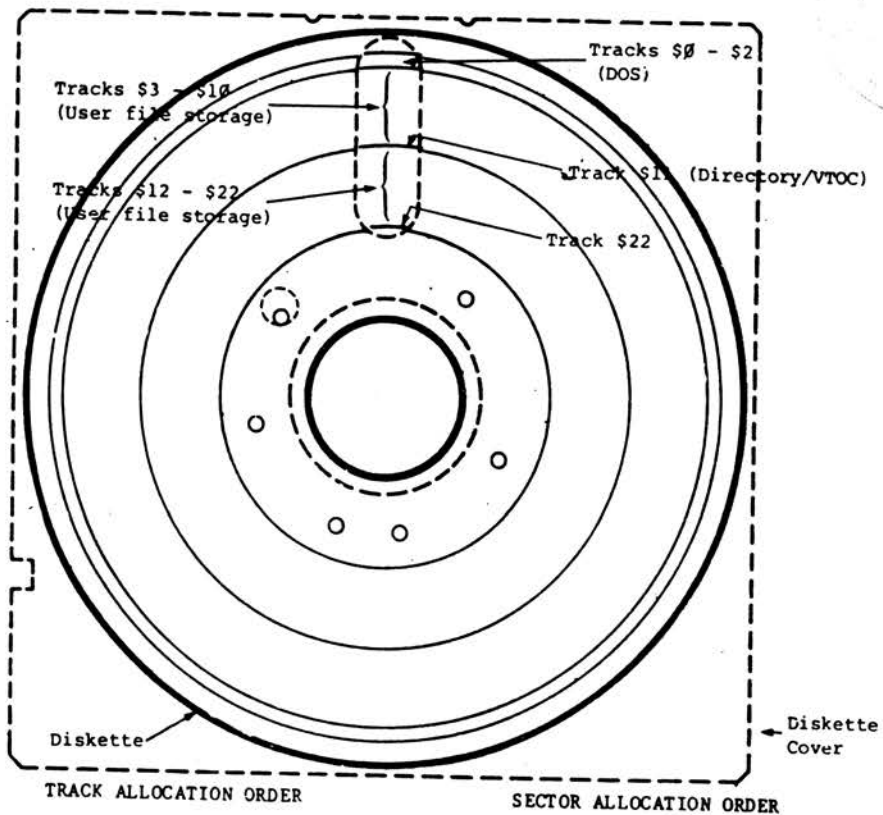
ปัจจุบันระบบ APPLESOFT ใช้ DOS 3.3 ซึ่งแต่เดิมใช้ DOS 3.2.1 หรือ 3.2 หรือ DOS ที่มี Version ต่ำกว่า 3.3 ส่วนสำคัญที่แตกต่างกันระหว่าง DOS 3.3 กับ DOS แบบเก่า คือจำนวนเซกเตอร์ที่แบ่งใน diskette ระบบ DOS 3.3 จะแบ่ง diskette เป็น 16 เซกเตอร์ แต่รุ่นก่อน ๆ จะแบ่ง diskette เป็น 13 เซกเตอร์ แต่ถ้านำ diskette ที่แบ่งเป็น 13 เซกเตอร์ มาใช้กับระบบ DOS 3.3 จะปรากฏคำว่า Unable To Read / write บนจอภาพ

4. จอภาพ (Monitor) เป็นชนิด Monochrome nonglasing คือ Green Monitor อักษรที่แสดงบนจอภาพมีขนาด 5 × 7 dot matrix 24 บรรทัด บรรทัดละ 40 ตัวอักษร สำหรับภาษาไทยจะแสดงผลบนจอภาพ 12 บรรทัด บรรทัดละ 40 ตัวอักษร

ลักษณะทั่วไปของ diskette

diskette หรือ floppy disk ถือเป็นหน่วยความจำสำรองที่ใช้เก็บข้อมูลในลักษณะของสารแม่เหล็ก และสามารถส่งข้อมูลที่เก็บไปยังเครื่องคอมพิวเตอร์ได้ มีลักษณะคล้ายแผ่นเสียง diskette ที่ใช้กับเครื่องไมโครคอมพิวเตอร์มีขนาด $5\frac{1}{4}$ นิ้ว ผิวของ diskette จะฉาบด้วย

สารแม่เหล็ก ซึ่งใช้เป็นที่เก็บข้อมูล และเพื่อป้องกันมิให้ผิวแม่เหล็กได้รับการขูดขีด ซึ่งจะทำให้ข้อมูลเปลี่ยนแปลง และทำให้แผ่น disk เสื่อมสภาพลง ดังนั้น diskette จึงต้องมีช่องกระดาษแข็งหุ้มอยู่ โดยเว้นช่องสำหรับให้หัว อ่าน/เขียน ทำการอ่าน/เขียนบนแผ่น disk ได้



รูปที่ 3.1 ลักษณะของฟลอปปีดิสก์

คำสั่งเกี่ยวกับ disk

การให้เครื่องบันทึก (disk drive) ทำงาน ขึ้นอยู่กับคำสั่งที่ใช้ ซึ่งมีหลายลักษณะ คือ

1. คำสั่ง PR # 6 (F_6) และกดปุ่ม RETURN เครื่องบันทึกจะหมุน
2. คำสั่ง CATALOG (F_1) และกดปุ่ม RETURN เครื่องบันทึกจะหมุนสักครู่จะดับ ซึ่งเป็นการ LIST รายชื่อไฟล์ทั้งหมดที่มีอยู่ใน diskette ให้ปรากฏบนจอภาพ อักษรและตัวเลขที่

ปรากฏ จะมีลักษณะและความหมายดังตัวอย่างต่อไปนี้

A	010	MENU	
*	B	018	PHYSICS
*	I	050	FRUITS
T	029	NAMES	

* แสดงว่ามีการ LOCK โปรแกรมไว้ ไม่สามารถจะ LIST โปรแกรมได้

A เป็นการบอกชนิดของไฟล์

010 บอกจำนวนเซกเตอร์ที่ใช้

MENU ชื่อของไฟล์

ชนิดของไฟล์ที่ใช้มี 4 จำพวก คือ

A คือไฟล์ที่เขียนด้วยภาษา APPLESOFT BASIC

B คือ BINARY FILE เขียนด้วย MACHINE LANGUAGE

I คือ ไฟล์ที่เขียนด้วยภาษา INTEGER BASIC

T คือ TEXT FILE หรือ DATA FILE

3. คำสั่ง RUN หรือ LOAD ตามด้วยชื่อไฟล์ และกดปุ่ม RETURN

4. คำสั่ง INIT เป็นคำสั่งที่ใช้กับ diskette แผ่นใหม่ เป็นการจัด FORMAT ให้กับ diskette แผ่นใหม่ เพื่อจะได้นำไปบันทึกโปรแกรมต่าง ๆ ได้ วิธีการ INITIALIZE ให้ทำดังนี้

4.1 สอดแผ่น DOS 3.3 (SYSTEM MASTER) เข้าเครื่องปัดฝา และกด PR # 6 (F₆) และกดปุ่ม RETURN เครื่องบันทึกหมุนสักครู่จะดับ

4.2 พิมพ์คำว่า NEW และกดปุ่ม RETURN เพื่อลงโปรแกรมทั้งหมดที่มีอยู่ในหน่วยความจำของเครื่อง

4.3 พิมพ์โปรแกรมที่ต้องการหรือโปรแกรมน้อย ๆ แล้วลอง RUN ดู เมื่อได้ผลให้ถอดแผ่น DOS 3.3 ออก ใส่ diskette แผ่นใหม่แทนที่

4.4 ใช้คำสั่ง INIT ตามด้วยชื่อโปรแกรม (ตามที่ตั้งไว้) กดปุ่ม RETURN เครื่องบันทึกจะทำงาน สักครู่จะดับ diskette ที่ทำการ INIT แล้วจะสามารถจะนำไปบันทึกโปรแกรม

อื่น ๆ ต่อไปนี้







5. คำสั่ง SAVE ตามด้วยชื่อไฟล์ และกดปุ่ม RETURN เครื่องบันทึกจะทำงาน ทำการบันทึกโปรแกรมที่ต้องการไว้ในแผ่น diskette

6. คำสั่ง DELETE ตามด้วยชื่อไฟล์ และกดปุ่ม RETURN เครื่องบันทึกจะหมุนและทำการลบโปรแกรมตามชื่อไฟล์ที่พิมพ์ไว้

การแก้ไขโปรแกรม







เครื่องไมโครคอมพิวเตอร์ที่ใช้ภาษา APPLESOFT BASIC จะมีเครื่องหมาย PROMPT และ CURSOR ปรากฏอยู่ด้านซ้ายมือของจอภาพ เครื่องหมาย PROMPT คือ] ส่วนเครื่องหมาย CURSOR คือ █ เป็นสี่เหลี่ยมกระพริบ CURSOR นี้จะเป็นเครื่องหมายที่ชี้บนจอภาพว่า เมื่อกดอักษรใดลงบนจอภาพ อักษรนั้นจะไปแทนที่ CURSOR และเครื่องหมาย CURSOR จะถูกเลื่อนไปทางขวามือ

เครื่อง CHERRY III มีคีย์สำหรับเลื่อน CURSOR อยู่ 6 แบบ คือ

	เลื่อน CURSOR ไปทางซ้าย
	เลื่อน CURSOR ไปทางขวา
	เลื่อน CURSOR ไปทางซ้าย
	เลื่อน CURSOR ไปทางขวา
	เลื่อน CURSOR ขึ้นบน
	เลื่อน CURSOR ลงล่าง

วิธีการใช้คีย์ทั้ง 6 แบบ มีดังนี้

- คีย์  และ  ใช้ในกรณีที่ต้องการแก้ไขตัวอักษรที่อยู่ในบรรทัด โดยยังไม่ได้กดปุ่ม RETURN

2. กรณีที่ต้องการแก้ไขข้อมูลที่อยู่ในหมายเลขบรรทัดต่าง ๆ ที่ผ่านมาแล้ว ให้กดคีย์  หรือ  เพื่อเลื่อน CURSOR ไปยังหมายเลขบรรทัดที่ต้องการแก้ไข แล้วกดคีย์  เพื่อเลื่อน CURSOR ไปยังหมายเลขบรรทัดตัวแรก จากนั้นให้กดคีย์  เพื่อเลื่อน CURSOR ไปยังอักษรที่ต้องการแก้ไข (ค่อจากนี้ให้ใช้คีย์  และ  เท่านั้น) เมื่อแก้ไขเรียบร้อยแล้ว ให้เลื่อน CURSOR ไปทางขวามือให้พ้นอักษรตัวสุดท้ายของหมายเลขบรรทัดนั้น แล้วกดปุ่ม RETURN

การใช้ตัวแปร

ตัวแปร (variables) เป็นอักษรที่ใช้แทนตัวเลข หรือค่าที่ต้องการเก็บไว้ในหน่วยความจำของเครื่อง เช่น $A = 10$ A จะเป็นชื่อตัวแปรที่มีค่าเท่ากับ 10

ตัวแปรมี 2 แบบคือ ตัวแปรที่บรรจุตัวเลข เรียกว่า NUMERICAL VARIABLES และตัวแปรที่บรรจุอักษรเป็นกลุ่ม เรียกว่า STRING VARIABLES ใช้ตัว § ตามหลังอักษร เช่น $A§ = \text{"HELLO"}$

หลักเกณฑ์การตั้งชื่อตัวแปรมีดังนี้

1. ชื่อตัวแปรต้องขึ้นต้นด้วยตัวอักษร
2. ชื่อตัวแปรจะเป็นอักษรตัวเดียวใด ๆ เช่น A B C ... ถึง Z
3. ชื่อตัวแปรอาจมีอักษรตามหลังได้ เช่น AA BC AD เป็นต้น
4. ชื่อตัวแปรอาจใช้ตัวเลขตามหลังอักษรได้ เช่น A1 A2 เป็นต้น
5. การเขียนชื่อตัวแปรอาจใช้ตัวอักษรนำแล้วตามด้วยอักษรหรือตัวเลขปนกันได้ แต่ความยาวต้องไม่เกิน 238 ตัว ซึ่งตัวที่มีความหมายจะใช้แค่ 2 ตัวแรกเท่านั้น (การตั้งชื่อตัวแปรต้องไม่ให้อักษร 2 ตัวแรกเหมือนกัน)
6. ชื่อตัวแปรต้องไม่เป็นคำเดียวกับคำสงวน (RESERVE WORDS)

เครื่องหมายที่ใช้ในการคำนวณ

เครื่องหมายที่ใช้ในการคำนวณของไมโครคอมพิวเตอร์ มีดังนี้

+	แทนการบวก
-	แทนการลบ
*	แทนการคูณ
/	แทนการหาร
Λ	แทนการยกกำลัง

เครื่องจะทำงานตามลำดับขั้น ดังนี้

ยกกำลัง คูณ หาร บวก ลบ

ดังนั้นเพื่อป้องกันการสับสนในการคำนวณ ให้ใช้เครื่องหมายวงเล็บเพื่อแบ่งการทำงานเป็นขั้นตอน การใช้วงเล็บอาจใช้ซ้อนกันหลาย ๆ วงเล็บได้ โดยเครื่องจะทำงานจากวงเล็บในออกมาวงเล็บนอก

คำสั่งในระบบ APLIESOFT BASIC

เครื่องไมโครคอมพิวเตอร์ที่ใช้ในการวิจัยนี้ จะมีแป้นกดคำสั่ง คือ

ESC ย่อจาก ESCAPE มีความหมายเพื่อใช้ร่วมกับแป้นกดข้อมูลตัวอื่น ๆ ใช้โดยการกดคีย์นี้ 1 ครั้ง แล้วปล่อย และกดคีย์อื่น ๆ ตาม เช่น ESC, @ เป็นการล้างจอภาพทั้งหมด และ cursor จะไปอยู่มุมบนซ้ายมือ

ESC, E เป็น การลบอักษรจนถึงอักษรสุดท้ายที่ปรากฏบนหมายเลขบรรทัดนั้น

CTRL ย่อจาก CONTROL ใช้ร่วมกับคีย์อื่น ๆ โดยกดคีย์นี้ค้างไว้แล้วกดคีย์อื่น ๆ ตาม เช่น CTRL - C เป็นการหยุดการทำงานของเครื่องชั่วคราวหนึ่ง

CTRL - S เป็นการหยุดการ LIST โปรแกรม เมื่อต้องการให้ทำงานต่อ ต้องกด CTRL - S อีกครั้ง

SHIFT ใช้เมื่อต้องการพิมพ์อักษรหรือเครื่องหมายที่อยู่ส่วนบนของแป้นกดข้อมูล (keys) มีความหมายเช่นเดียวกับการยกแคร่ของพิมพ์ดีด

- RESET ใช้ร่วมกับ CTRL คือ CTRL - RESET จะเป็นการย้อนกลับไปยังต้น
โปรแกรม
- RETURN เป็นคีย์ที่ใช้บ่อยที่สุด เมื่อพิมพ์จบบรรทัดแล้วต้องการขึ้นทนายเลขบรรทัดใหม่
ต้องกดคีย์นี้ เพื่อให้เครื่องรับรู้ว่าจะบรรทัดที่พิมพ์
- BACK SPACE ใช้เมื่อต้องการเลื่อน cursor ย้อนกลับมาทางซ้ายหนึ่งตัวอักษร
- ← เป็นการเลื่อน cursor ย้อนกลับไปทางซ้ายมือ
 - เป็นการเลื่อน cursor ไปทางขวามือ
 - ↑ เป็นการเลื่อน cursor ขึ้นบน
 - ↓ เป็นการเลื่อน cursor ลงล่าง
- CAPS LOCK เป็นคีย์ที่อยู่ล่างสุดซ้ายมือติดกับ space bar ใช้เมื่อต้องการพิมพ์อักษรตัว
พิมพ์เล็ก (ปกติเป็นอักษรตัวพิมพ์ใหญ่) และถ้าจะกลับมาใช้ตัวพิมพ์ใหญ่ ให้กด
CAPS LOCK อีกครั้ง
- SPACE BAR เป็นแป้นยาวอยู่ล่างสุด ทำหน้าที่เว้นช่องไฟในขณะพิมพ์อักษร

คำสั่ง (COMMAND) ที่ใช้กับเครื่องไมโครคอมพิวเตอร์ มี 2 แบบ คือ

1. System Command คือคำสั่งให้เครื่องปฏิบัติตามโดยตรง System Command
ที่ใช้ในภาษา APPLESOFT BASIC มีดังนี้

- NEW เป็นคำสั่งลบสิ่งต่าง ๆ ที่มีอยู่ในหน่วยความจำทั้งหมด เพื่อมิให้โปรแกรมที่
เขียนใหม่ปนกับโปรแกรมเก่าที่มีอยู่ในหน่วยความจำ
- LIST เป็นคำสั่งให้เครื่องแสดงโปรแกรมที่เขียนไว้ตั้งแต่ต้นโปรแกรมจนจบโปรแกรม
โดยแสดงผลออกทางจอภาพ ตัวอย่างการใช้ LIST เช่น
- LIST 10 เครื่องจะแสดงโปรแกรมเฉพาะบรรทัดที่ 10
- LIST 20 - 50 เครื่องจะแสดงโปรแกรมตั้งแต่บรรทัดที่ 20 ถึงบรรทัดที่ 50

- LIST - 50 หรือ LIST, 50 เครื่องจะแสดงโปรแกรมตั้งแต่ต้นจนถึงบรรทัดที่ 50
- LIST 50 - หรือ LIST 50, เครื่องจะแสดงโปรแกรมตั้งแต่บรรทัดที่ 50 จนถึงจบโปรแกรม
- RUN เป็นคำสั่งให้เครื่องปฏิบัติตามโปรแกรมที่วางไว้ตั้งแต่ต้นจนจบโปรแกรม ใช้คู่กับ RETURN
- HOME เป็นคำสั่งให้เครื่องลบอักษรหรือภาพทั้งหมดที่มีอยู่บนจอภาพ ใช้คู่กับ RETURN
- LOAD เป็นคำสั่งที่ให้เครื่องรับโปรแกรมที่ส่งมาจากเทปบันทึกหรือรับโปรแกรมจาก diskette ถ้าใช้ diskette ต้องระบุชื่อโปรแกรมด้วย คำสั่ง LOAD ที่ใช้คู่กับ RETURN
- SAVE เป็นคำสั่งให้เครื่องส่งโปรแกรมที่มีอยู่ในหน่วยความจำไปเก็บไว้ในเทปบันทึกหรือส่งไปเก็บไว้ใน diskette ซึ่งต้องระบุชื่อโปรแกรมด้วย คำสั่งนี้ใช้คู่กับ RETURN
- CTRL - C เป็นการสั่งให้เครื่องหยุดการ LIST หรือ RUN
- CONT เป็นการสั่งให้เครื่องทำงานต่อจากการที่กด CTRL - C ไว้
- CTRL - S เป็นการสั่งให้เครื่องหยุดการ LISTชั่วคราว เมื่อจะให้เครื่องทำงานต่อไปให้กด CTRL - S อีกครั้ง
- TRACE เมื่อใช้คำสั่งนี้ก่อน RUN หรือดำเนินการตามโปรแกรม เครื่องจะแสดงออกทางจอภาพว่า เครื่องกำลังดำเนินการถึงโปรแกรมบรรทัดไหน เพื่อสะดวกต่อการตรวจสอบ
- NOTRACE เป็นคำสั่งยกเลิก TRACE
- DEL เป็นคำสั่งให้ลบโปรแกรม ต้องตามด้วยหมายเลขบรรทัด เช่น

DEL 50, 50 ให้ลบโปรแกรมบรรทัดที่ 50

DEL 50 - 200 ให้ลบโปรแกรมบรรทัดที่ 50 ถึงบรรทัดที่ 200

PRINT เป็นคำสั่งให้เครื่องแสดงค่าหรืออักษรที่ต้องการบนจอภาพ ถ้ามีเครื่องพิมพ์ ก็ให้พิมพ์ออกมาทางเครื่องพิมพ์

2. Program Command คือคำสั่งที่บรรจุไว้ในโปรแกรมเพื่อให้เครื่องปฏิบัติตาม โปรแกรมที่เขียนไว้ตามลำดับ เช่น GOTO GOSUB IF---THEN FOR---NEXT INPUT GET เป็นต้น

คำสั่งต่าง ๆ ที่ใช้สำหรับเขียนโปรแกรม มีดังนี้

PRINT เป็นคำสั่งให้เครื่องแสดงผลเป็นตัวเลขหรือตัวอักษร บนจอภาพ คำสั่ง PRINT นี้ อาจใช้เครื่องหมาย ? แทนได้ เช่น 10 PRINT "ABC"

INPUT เป็นคำสั่งที่ใส่ในโปรแกรม เพื่อให้เครื่องหยุดการป้อนข้อมูลเข้าทางแป้นกด ข้อมูล คำสั่งนี้ต้องใช้ RETURN ด้วยทุกครั้งที่ป้อนข้อมูลหมด

GET เป็นคำสั่งเช่นเดียวกับ INPUT แต่ข้อมูลที่ป้อนเข้าจะใช้ได้เพียงหนึ่งตัวอักษร เท่านั้น และไม่ต้องใช้ RETURN

GOTO เป็นคำสั่งให้เครื่องกระโดดไปทำงานที่หมายเลขบรรทัดอื่น ตามที่ระบุไว้ หลังคำสั่ง GOTO คำสั่ง GOTO ต้องตามด้วยหมายเลขบรรทัด

GOSUB เป็นคำสั่งให้เครื่องกระโดดไปทำงานที่โปรแกรมย่อยที่ระบุไว้หลัง GOSUB เมื่อจบโปรแกรมย่อยเครื่องจะกลับมาทำงานต่อจากคำสั่ง GOSUB (คำสั่ง GOSUB ต้องตามด้วยหมายเลขบรรทัดของโปรแกรมย่อย)

IF---THEN--- เป็นคำสั่งของการเปรียบเทียบ เครื่องหมายแสดงการเปรียบเทียบจะอยู่ หลัง IF ถ้าการเปรียบเทียบ เป็นจริง เครื่องจะปฏิบัติตามคำสั่งหลัง THEN แต่ถ้าไม่เป็นจริงจะผ่านไปปฏิบัติตามคำสั่งในบรรทัดถัดไป

เครื่องหมายแสดงการเปรียบเทียบที่ใช้กับ IF--THEN-- คือ

- = มีความหมาย เท่ากับ
- <> มีความหมาย ไม่เท่ากับ
- < มีความหมาย น้อยกว่า
- > มีความหมาย มากกว่า
- <= มีความหมาย น้อยกว่าหรือเท่ากับ
- >= มีความหมาย มากกว่าหรือเท่ากับ

FOR---NEXT-- เป็นคำสั่งที่ให้โปรแกรมปฏิบัติตามเป็นวงรอบ อาจจะมีวงรอบเดียวหรือหลาย ๆ วงรอบ และเมื่อเครื่องปฏิบัติตามโปรแกรมที่วางไว้แล้วจะหลุดออกมาเองโดยอัตโนมัติ

ON---GOTO-- เป็นคำสั่งที่ระบุให้ตัวแปรซึ่งตามหลัง ON มีค่าเป็นตัวเลขจำนวนเต็มตั้งแต่เลข 1 ขึ้นไป จะต้องกระโดดไปปฏิบัติตามโปรแกรมในหมายเลขบรรทัดที่อยู่หลัง GOTO โดยปฏิบัติเรียงตามลำดับ ดังตัวอย่างต่อไปนี้

```
50 ON A GOTO 100, 150, 200
```

ความหมายของโปรแกรมที่ 50 มีดังนี้

ถ้า A = 1 ให้กระโดดไปทำงานที่บรรทัดที่ 100

ถ้า A = 2 ให้กระโดดไปทำงานที่บรรทัดที่ 150

ถ้า A = 3 ให้กระโดดไปทำงานที่บรรทัดที่ 200

แต่ถ้า A มากกว่า 3 (มากกว่าจำนวนหมายเลขบรรทัดที่ตามหลัง GOTO)

โปรแกรมจะผ่านไปยังบรรทัดถัดไป

- ON---GOSUB--- วิธีการใช้เช่นเดียวกับ ON---GOTO--- ผิดกันตรงที่สั่งให้เครื่องกระโดดไปทำงานที่โปรแกรมย่อยตามหมายเลขบรรทัดที่ระบุไว้
- TAB คือคำสั่งที่กำหนดให้อักขรที่ต้องการให้ปรากฏบนจอภาพอยู่ในตำแหน่งใดตำแหน่งหนึ่งของจอภาพ
TAB เป็น PROGRAM COMMAND ต้องใช้ตามหลัง PRINT และมีหมายเลขในวงเล็บกำกับไว้ ซึ่งหมายเลขนี้คือตำแหน่งที่จะให้อักขรปรากฏว่าเป็นตำแหน่งตัวอักขรที่เท่าใดในบรรทัดของจอภาพ
- HTAB มีความหมายเช่นเดียวกับ TAB แต่มีข้อแตกต่างคือ
-HTAB ต้องเขียนอยู่ในโปรแกรมก่อนบรรทัดที่มีคำสั่ง
-ตัวเลขที่ระบุตำแหน่งจะอยู่หลัง HTAB โดยไม่ต้องมีวงเล็บ
-สามารถจะสั่งให้อักขรปรากฏที่ปลายบรรทัดก่อนแล้วค่อยมาปรากฏที่ต้นบรรทัดได้ โดยไม่จำเป็นต้องเรียงตามลำดับ
- VTAB คือคำสั่งให้อักขรปรากฏในแต่ละบรรทัดตามหมายเลขที่ระบุไว้หลัง VTAB ถ้าจอภาพสามารถแบ่งได้ 24 บรรทัด ค่าตัวเลขที่ตามหลัง VTAB ต้องไม่เกินหมายเลข 24 การใช้ VTAB เช่นเดียวกับ HTAB คือหมายเลขที่ตามหลัง VTAB ไม่ต้องมีวงเล็บ การเขียนต้องเขียนในโปรแกรมก่อนบรรทัดที่มีคำสั่ง PRINT และนอกจากนี้ยังสามารถสั่งให้แสดงที่บรรทัดใดบนจอภาพก่อนได้ ไม่จำเป็นต้องเรียงตามลำดับ
- READ---DATA--- คำสั่ง READ เป็นคำสั่งให้เครื่องอ่านและเก็บข้อมูลที่อยู่ในบรรทัดที่มีคำว่า DATA โดยมีหลักเกณฑ์ ดังนี้
-ตัวแปรที่เก็บข้อมูลจะต้องเป็นตัวแปรชนิดเดียวกับข้อมูล
-ตัวแปรในคำสั่ง READ จะมากกว่าข้อมูลใน DATA ไม่ได้ แต่ข้อมูลใน DATA จะมากกว่าตัวแปรได้
-ข้อมูลจะอยู่ส่วนใดของโปรแกรมก็ได้

I1588580X

- DIM** เป็นคำสั่งในการจองที่ว่างในหน่วยความจำ สำหรับ ARRAY หรือ STRING ซึ่งหลัง DIM จะต้องเป็นตัวเลขอยู่ในวงเล็บ
array คือ ระบบการเรียกตัวแปรที่เป็นหมวดหมู่ซึ่งมีอยู่ 2 แบบที่ใช้กันมากคือ
- ARRAY มิติเดียว เช่น A(5) B\$ (11) AB (20) เป็นต้น
- ARRAY สองมิติ เช่น A(5, 11) A\$ (12, 6) AB(6, 30) เป็นต้น
- SPEED** เป็นคำสั่งให้เครื่องปฏิบัติงานช้าลง SPEED ต้องตามหลังด้วยหมายเลข ซึ่งหมายเลขที่แสดงการปฏิบัติงานของเครื่องตามปกติ คือหมายเลข 255 ถ้าหมายเลขต่ำกว่า 255 เครื่องจะทำงานช้าลง
- ONERR GOTO--** เป็นคำสั่งที่ใช้ในกรณีที่มีข้อผิดพลาด (ERROR) ในโปรแกรม
หรือ
ONERR GOSUB-- เครื่องจะหยุดทำงาน แต่ถ้าใส่คำสั่งนี้ก่อนบรรทัดที่จะมีการคำนวณที่จะทำให้เกิด ERROR ขึ้น เครื่องก็จะดำเนินการต่อไปได้เอง ถึงแม้ว่าจะมี ERROR เกิดขึ้นก็ตาม
- END** เป็นคำสั่งที่แสดงถึงการสิ้นสุดโปรแกรม ปกติจะใช้เฉพาะก่อนหน้าโปรแกรมย่อยเท่านั้น
- PRINT CHR\$(4)** คือคำสั่งที่บรรจุอยู่ในโปรแกรมแล้วสามารถที่จะสั่งให้เครื่องดำเนินการ LOAD หรือ RUN โปรแกรมอื่น ๆ ต่อเนื่องกันไปได้เช่น
30 PRINT CHR\$(4) ; "RUN PHYSICS"
- CLEAR** เป็นคำสั่งให้ตัวแปร และจำนวนใด ๆ ของตัวแปรมีค่าเป็นศูนย์
- X = FRE (0)** เป็นคำสั่งให้ค่าของ String เริ่มต้นที่จุดเริ่มต้นใน String Space
- POKE - 16368, 0** หรือ **POKE 49168, 0** เป็นการ Reset Keyboard strobe ที่อาจจะมีอักษรถูกอ่านเข้ามา
- K=PEEK(-16384)** หรือ **K = PEEK (49152)** เป็นการอ่านค่า Keyboard ถ้า X > 127 (X เป็นค่า ASCII ของคีย์ที่ถูกกด) แสดงว่ามีการกดคีย์

POKE 216, 0 เป็นการ clear ข้อผิดพลาดที่อาจจะเกิดขึ้นได้ในโปรแกรม

SCIENTIFIC NOTATION

ในภาษา APPLESOFT BASIC เมื่อมีการแสดงค่าตัวเลขที่มีค่ามาก ๆ หรือน้อย ๆ จะแสดงออกมาในรูปของ SCIENTIFIC NOTATION

ถ้าค่าตัวเลข น้อยกว่า 0.01 หรือมากกว่า 99999999.2 จะแสดงค่าเป็น SCIENTIFIC NOTATION

ถ้าค่าตัวเลขอยู่ในช่วง 0.01 คือ 99999999.2 จะแสดงค่าเป็นตัวเลขธรรมดา

ตัวอย่างของ SCIENTIFIC NOTATION ในภาษา APPLESOFT BASIC คือ

$$5E + 09 = 5 \times 10^9$$

$$5E - 06 = 5 \times 10^{-6}$$

$$5E + 16 = 5 \times 10^{16}$$

FUNCTION

FUNCTION ที่ใช้ในไมโครคอมพิวเตอร์ มีอยู่ 2 แบบ คือ

1. BUILT - IN FUNCTION เป็น FUNCTION ที่สร้างบรรจุไว้ในเครื่อง ผู้ใช้สามารถเรียนมาใช้ได้ทันที FUNCTION เหล่านี้มีดังนี้

SIN(X) คือค่า sin ของมุม X มีหน่วยเป็น Radian

COS(X) คือค่า cos ของมุม X มีหน่วยเป็น Radian

TAN(X) คือค่า tan ของมุม X มีหน่วยเป็น Radian

ATN(X) คือค่าของ Arctangent เป็นมุม Radian ซึ่งมุมนี้จะมีค่าระหว่าง $-\frac{\pi}{2}$ ถึง $+\frac{\pi}{2}$

INT(X) จะให้ค่าเลขจำนวนเต็มของ X ซึ่งก็คือการตัดเศษทิ้งไป

RND(X) จะให้ค่าสุ่มของตัวเลข (RANDOM NUMBER) ตัวเลขสุ่มนี้จะมีค่าระหว่าง

0 ถึง 1

SGN(X) จะให้เครื่องหมายของ X

ถ้า $x > 0$ SGN(X) = 1

ถ้า $x < 0$ SGN(X) = -1

ถ้า $x = 0$ SGN(X) = 0

ABS(X) จะให้ค่าสมบูรณ์ (ABSOLUTE) ของ X คือไม่คิดเครื่องหมายลบ

SQR(X) จะให้ค่ารากที่สอง (SQUARE ROOT) ของ X (ที่เป็นบวก)

EXP(X) จะให้ค่า e^x ($e = 2.718289$)

LOG(X) จะให้ค่า \log ฐาน e ($\log_e(x)$)

ASC("") จะให้รหัสแอสกี (ASCII CODE) ของอักษรที่อยู่ในเครื่องหมายคำพูด

VAL(X\$) เป็นการทำให้ตัวแปรพันระที่หมดสภาพ เป็นตัวเลขไปแล้วกลับมา มีสภาพ เป็นตัวเลขอีก เช่น

```
10 A$ = "15"
```

```
20 B$ = "10"
```

```
30 PRINT VAL (A$) + VAL (B$)
```

ผลที่ได้คือการนำจำนวนตัวเลข 15 บวกกับ 10

STR\$(X) เป็นการทำให้ตัวเลขหมดสภาพการเป็นตัวเลข เช่น

```
10 A = 15
```

```
20 B = 10
```

```
30 PRINT STR $(A) + STR $(B)
```

ผลที่ได้คือ ตัวเลข 15 และตัวเลข 10 จะเรียงต่อกันเป็น 1510

LEN(X\$) จะให้ค่าเป็นจำนวนตัวอักษรที่ประกอบเป็น X\$

- LEFT\$(X\$, X) จะแสดงตัวอักษรที่อยู่ซ้ายมือสุดของ X\$ เป็นจำนวน X ตัวอักษร
- MID\$(X\$, X, Y) จะแสดงตัวอักษรที่อยู่ตรงกลางของ X\$ นับตั้งแต่ X ตัวอักษรจากซ้ายมือไป เป็นจำนวน Y ตัวอักษร
- RIGHT\$(X\$, X) จะแสดงตัวอักษรที่อยู่ขวามือสุดของ X\$ เป็นจำนวน X ตัวอักษร (นับจากขวามือมา)

2. DERIVED FUNCTION หรือ USER - DEFINED FUNCTION เป็น FUNCTION ที่ผู้ใช้สร้างขึ้นมา และสามารถจะเรียกมาใช้ได้ มีประโยชน์ในกรณีที่เป็นสูตรที่ต้องใช้มากกว่า 1 ครั้ง ตัวอย่างเช่น

$$\text{DEF FN A(R)} = 3.14159 \times R \wedge 2$$

A คือ ชื่อของ FUNCTION

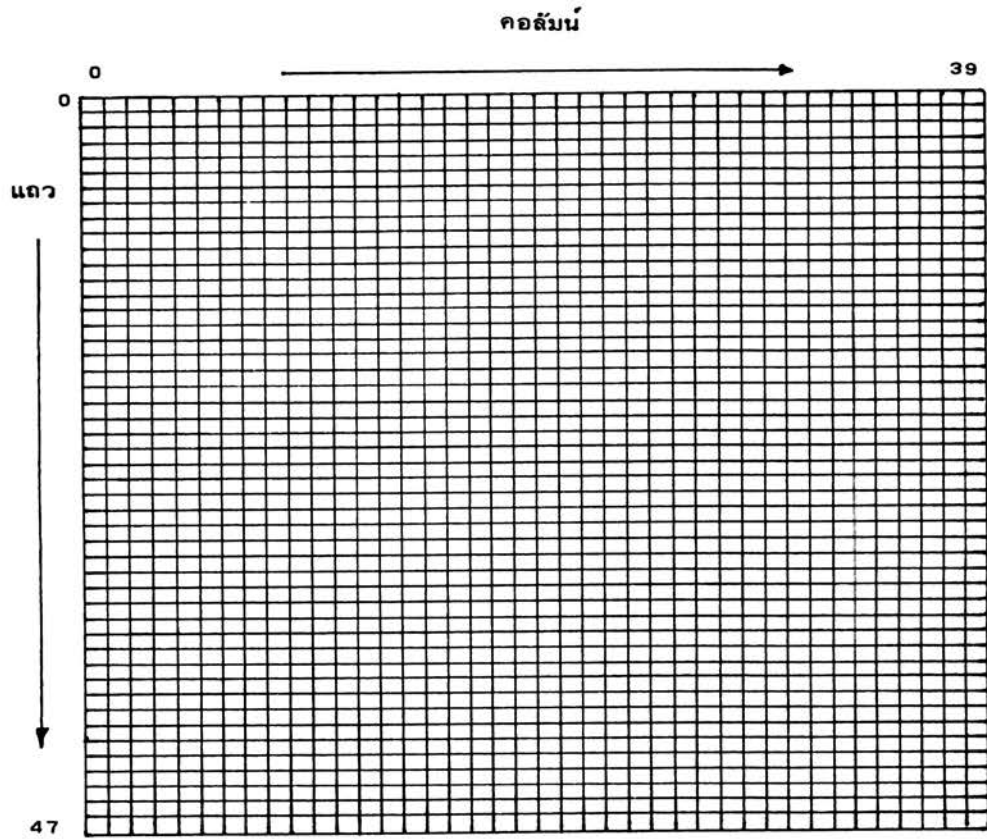
R คือ ตัวอ้างอิง ซึ่งตัวที่ตามหลังเครื่องหมายเท่ากับจะขึ้นอยู่กับตัวอ้างอิงตัวนี้

กราฟฟิก

ภาพที่แสดงออกทางจอภาพจะมีอยู่ 2 แบบคือ เป็นตัวอักษรหรือตัวเลขหรือเครื่องหมายที่อยู่บนแป้นพิมพ์ แบบนี้เรียกว่า TEXT MODE ส่วนการแสดงผลทางจอภาพอีกแบบหนึ่ง เรียกว่า GRAPHICS MODE ซึ่งแบ่งเป็น 2 แบบคือ กราฟฟิกแบบเส้นหยาบ (LOW RESOLUTION GRAPHICS) และกราฟฟิกแบบเส้นละเอียด (HIGH RESOLUTION GRAPHICS)

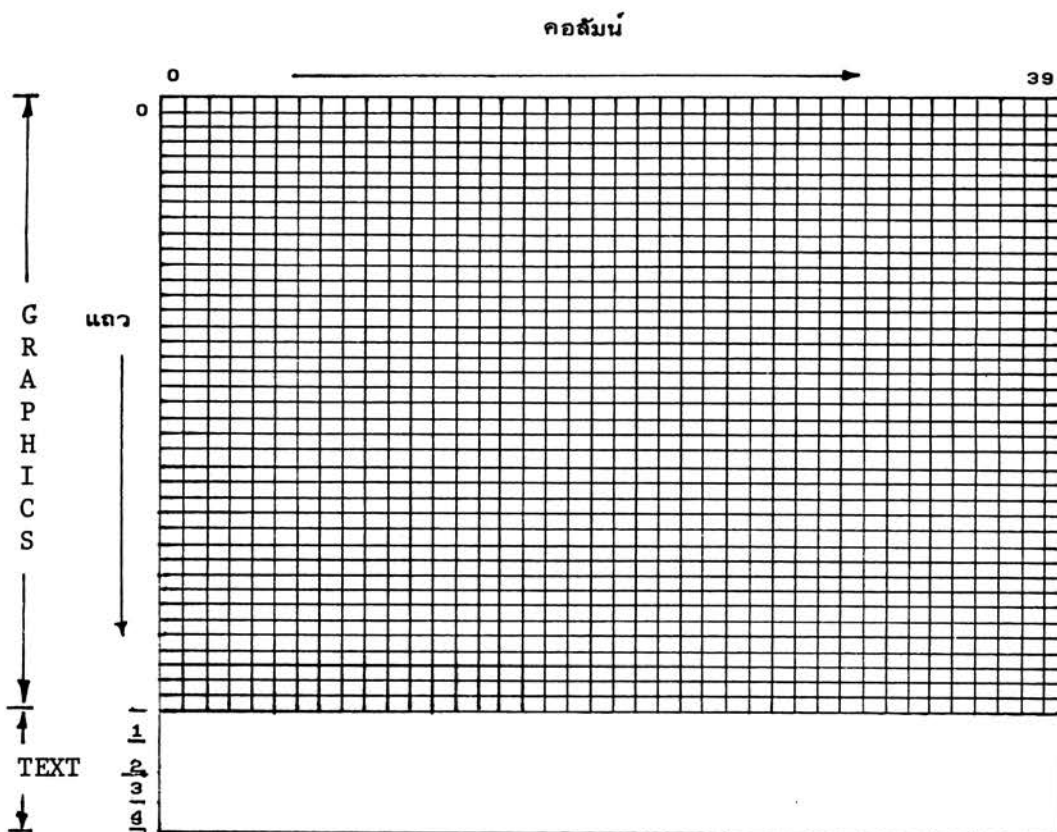
1. กราฟฟิกเส้นหยาบ (LOW RESOLUTION GRAPHICS) หรือที่เรียกว่า LO - RES GRAPHICS แบ่งเป็น 2 แบบคือ SPLIT SCREEN GRAPHICS กับ FULL SCREEN GRAPHICS

ใน FULL SCREEN GRAPHICS จอภาพจะแบ่งเป็นตาราง จากซ้ายไปขวา 40 ช่อง หรือ 40 คอลัมน์ (เหมือน TEXT MODE) และจากบนลงล่างเป็น 48 ช่อง ในแต่ละตารางซึ่งมีทั้งหมด 1920 ตาราง จะสามารถกำหนดสีต่าง ๆ ลงไปได้ สีทั้งหมดมี 16 สี (กรณีที่ใช้จอภาพเป็นสี)



รูปที่ 3.2 FULL SCREEN LO - RES GRAPHICS

ใน SPLIT SCREEN GRAPHICS จะมีการแบ่งจอภาพคล้าย FULL SCREEN GRAPHICS ผิดกันตรงจำนวนแถวจากบนลงล่าง ซึ่งใน SPLIT SCREEN GRAPHICS จะแสดงเพียง 40 แถว คือตั้งแต่ 0 ถึง 39 ส่วนอีก 8 แถวข้างล่างจะไว้เป็นช่องสำหรับแสดงอักษร ตัวเลข หรือเครื่องหมาย ส่วนล่างนี้เรียกว่า TEXT WINDOW มีจำนวน 4 บรรทัด



รูปที่ ๓.๓ SPLIT SCREEN GRAPHICS

หมายเลข	สี
0	ดำ (BLACK)
1	ม่วง (MAGENTA)
2	น้ำเงินแก่ (DARK BLUE)
3	ม่วงแดง (PURPLE)
4	เขียวแก่ (DARK GREEN)
5	เทาเบอร์ 1 (GRAY # 1)
6	น้ำเงินเข้ม (MEDIUM BLUE)
7	น้ำเงินอ่อน (LIGHT BLUE)
8	น้ำตาล (BROWN)
9	ส้ม (ORANGE)
10	เทาเบอร์ 2 (GRAY # 2)
11	ชมพู (PINK)
12	เขียวอ่อน (LIGHT GREEN)
13	เหลือง (YELLOW)
14	AQUA
15	ขาว (WHITE)

รูปที่ 3.4 สีของ LO - RES GRAPHICS

จาก TEXT MODE จะเข้าสู่ LO - RES GRAPHICS MODE ให้พิมพ์คำว่า GR และกดปุ่ม RETURN เครื่องจะเข้าสู่ LO - RES GRAPHICS MODE ซึ่งจะเป็นชนิด SPLIT SCREEN GRAPHICS ที่มี TEXT WINDOW

ถ้าจะให้เครื่องอยู่ใน FULL SCREEN GRAPHICS ให้พิมพ์คำว่า POKE - 16302, 0 และกดปุ่ม RETURN แต่เมื่อต้องการกลับเข้าสู่ SPLIT SCREEN GRAPHICS อีก ก็พิมพ์คำว่า

GR และกดปุ่ม RETURN

หลังจากกด GR แล้ว ต้องกำหนดสีที่มันที่ มิฉะนั้นจะไม่เห็นภาพ เช่น

10 GR

20 COLOR = 9

คำสั่งในการวาดภาพใน LO - RES GRAPHICS MODE มี 3 คำสั่ง คือ

PLOT เป็นคำสั่งให้แสดงภาพ เป็นจุดตามตำแหน่งที่กำหนด หลังคำสั่ง PLOT ต้องเป็นตัวเลขบอก แถว และคอลัมน์ เช่น

10 GR

20 COLOR = 9

30 PLOT 24, 19

HLIN ย่อจาก HORIZONTAL LINE คือเส้นในแนวนอน เป็นคำสั่งให้ลากเส้นตามแนวระดับ ตามตำแหน่งที่กำหนดตามหลังคำสั่ง HLIN ถ้ากำหนดเป็น HLIN A, B AT C

A : คือหมายเลขคอลัมน์ที่เป็นตำแหน่งตั้งต้นของเส้น และต้องมีค่าน้อยกว่า ๐

B : คือหมายเลขคอลัมน์ที่เป็นตำแหน่งปลายของเส้น และต้องมีค่ามากกว่า A และไม่เกิน ๓๑

C : คือหมายเลขแถวที่เป็นตำแหน่งของเส้นระดับ

VLIN ย่อจาก VERTICAL LINE หรือเส้นในแนวตั้ง เป็นคำสั่งให้ลากเส้นตามแนวตั้งตามตำแหน่งที่กำหนด

ถ้ากำหนดเป็น VLIN A, B AT C

A : คือหมายเลขแถวที่เป็นตำแหน่งเริ่มต้นของเส้น จะต้องมีค่าน้อยกว่า ๐

B : คือหมายเลขแถวที่เป็นตำแหน่งปลายของเส้น ซึ่งจะต้องมีค่ามากกว่า A

C : คือหมายเลขคอลัมน์ที่เป็นตำแหน่งของเส้นแนวตั้ง มีค่าอยู่ระหว่าง ๐ ถึง

39 (กรณีของ SPLIT SCREEN GRAPHICS)

2. กราฟฟิกเส้นละเอียด (HIGH RESOLUTION GRAPHICS) หรือเรียกย่อ ๆ ว่า HI - RES GRAPHICS แบ่งเป็น 2 แบบเช่นกัน คือ แบบ SPLIT SCREEN (HGR) และแบบ FULL SCREEN (HGR 2)

การเข้าสู่ HI - RES SPLIT SCREEN GRAPHICS MODE ทำได้โดยการพิมพ์ HGR และกด RETURN จากนั้นจอภาพจะถูกแบ่งเป็นตารางชนิดมองไม่เห็น จากซ้ายไปขวา 280 คอลัมน์ (จาก 0 ถึง 279) และจากบนลงล่าง 160 แถว (จาก 0 ถึง 159) ด้านล่างของจอภาพจะเป็น TEXT WINDOW

แต่ถ้าพิมพ์ HGR 2 จอภาพจะถูกแบ่งเป็นตารางจากซ้ายไปขวา 280 คอลัมน์ และจากบนลงล่าง 192 แถว (0 ถึง 191) โดยไม่มี TEXT WINDOW บนจอภาพ

ใน HI - RES GRAPHICS จะสามารถแสดงสีได้เพียง 8 สี โดยแบ่งเป็นสีคู่และสีคี่ดังนี้

สีคู่	สีคี่
0 ดำ	1 เขียว (ขึ้นอยู่กับชนิดของจอ)
2 น้ำเงิน (ขึ้นอยู่กับชนิดของจอ)	3 ขาว 1
4 ดำ 2	5 (ขึ้นอยู่กับชนิดของจอ)
6 (ขึ้นอยู่กับชนิดของจอ)	7 ขาว 2

รูปที่ 3.5 สีใน HI - RES GRAPHICS

การกำหนดสีให้ใช้คำสั่ง HCOLOR แล้วตามด้วยเครื่องหมายเท่ากับและหมายเลขสี

เช่น

HCOLOR = 3

คำสั่งการวาดภาพใน HI - RESOLUTION GRAPHICS มี 3 แบบ คือ

(1) HPLOT A, B : เป็นการสั่งให้แสดงเป็นจุด

A : คือหมายเลขคอลัมน์

B : คือหมายเลขแถว

(2) HPLOT A, B TO C, D : เป็นการสั่งให้ลากเส้น

A : คือหมายเลขคอลัมน์ที่เป็นจุดเริ่มต้นของเส้น

B : คือหมายเลขแถวที่เป็นจุดเริ่มต้นของเส้น

C : คือหมายเลขคอลัมน์ที่เป็นจุดสุดท้ายของเส้น

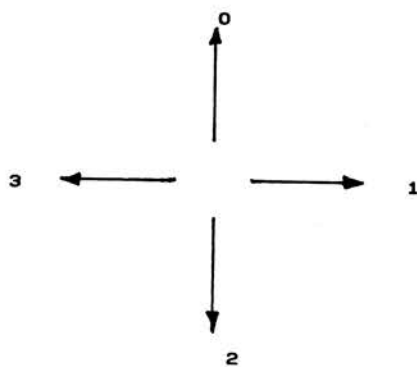
D : คือหมายเลขแถวที่เป็นจุดสุดท้ายของเส้น

(3) HPLOT TO A, B : เป็นคำสั่งที่ต่อเนื่องจากคำสั่ง HPLOT อื่น ๆ

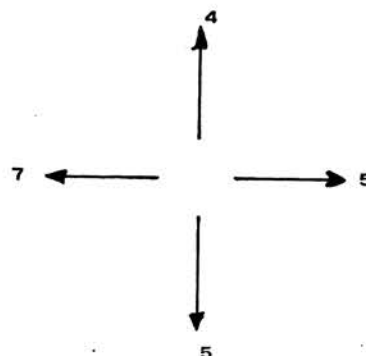
สำหรับการใช้ GRAPHICS MODE นี้ ถ้าต้องการกลับเข้าสู่ TEXT MODE ให้กด TEXT และกดปุ่ม RETURN

SHAPE TABLE

การสร้างรูปร่าง (SHAPE) จะต้องมีการกำหนดจุดศูนย์กลางหรือจุดเริ่มต้นของรูป และทำการลากเส้นออกจากจุดดังกล่าวไปตามรอบรูปของรูปร่างจนมาบรรจบกับจุดเดิม การลากเส้นมี 2 แบบ คือ แบบไม่ให้ปรากฏเส้น บนจอภาพ และแบบให้ปรากฏเส้นบนจอภาพ



รูปที่ 3.6 แสดงรหัสและทิศทางการ
เคลื่อนที่แบบไม่ปรากฏเส้น



รูปที่ 3.7 แสดงรหัสและทิศทางการเคลื่อนที่
แบบปรากฏเส้น

ความหมายของรหัสมีดังนี้

- 0 หมายถึง การเคลื่อนที่ขึ้นบนโดยไม่ปรากฏเส้น
- 1 หมายถึง การเคลื่อนที่ไปทางขวามือโดยไม่ปรากฏเส้น
- 2 หมายถึง การเคลื่อนที่ลงล่างโดยไม่ปรากฏเส้น
- 3 หมายถึง การเคลื่อนที่ไปทางซ้ายมือโดยไม่ปรากฏเส้น
- 4 หมายถึง การลากเส้นขึ้นบน
- 5 หมายถึง การลากเส้นไปทางขวามือ
- 6 หมายถึง การลากเส้นลงล่าง
- 7 หมายถึง การลากเส้นไปทางซ้ายมือ
- 8 หมายถึง รหัสปิดท้ายที่ต้องการให้เครื่องรับรู้ว่าหมด รูปร่างแล้ว

วิธีการใช้ ให้นำรหัสมาเขียนเรียงกัน และปิดท้ายด้วย รหัสเลข 8 ดังตัวอย่าง

1000 DATA 2, 2, 2, 7, 4, 4, 7, 7, 4, 4, 5, 5, 4, 4, 5, 5, 6, 6,
5, 5, 6, 6, 7, 7, 6, 6, 7, 8

รหัสที่กล่าวนี้จะถูกแปลงเป็นเลขฐานสิบหก และเก็บไว้ในหน่วยความจำที่กำหนดเป็นพิเศษ
ซึ่งจะอยู่ในรูปของ SUBROUTINE ดังตัวอย่าง

```
500 REM PROGRAM SUBROUTINE FOR SHAPE TABLE
510 POKE 768, 01 : POKE 769, 00 : POKE 770, 04 : POKE 771, 00
    : M = 772
520 POKE 232, 00 : POKE 233, 03
530 READ A
540 IF A = 8 THEN 610
550 READ B
560 IF B = 8 THEN 600
```


POKE 768, 01 : 01 คือจำนวนของ SHAPE TABLE
 POKE 769, 00 : 00 คือส่วนที่ไม่ใช่ต้องเป็นศูนย์เสมอ
 POKE 770, 04 : 04 คือค่า offset ของ SHAPE TABLE
 เป็น LOW BYTE
 POKE 771, 00 : 00 คือ offset ของ SHAPE TABLE เป็น HIGH BYTE

เมื่อใส่ SHAPE TABLE ลงไปยังตำแหน่งของหน่วยความจำ แล้วจะต้องบอกให้ เครื่องรู้ว่ามันอยู่ที่ไหน โดยการใส่หน่วยความจำตำแหน่ง 232 และ 233 ซึ่งสำรองไว้ คำสั่ง POKE ต่อไปนี้จะเป็นการบอกให้ APPLESOFT ว่า SHAPE TABLE เริ่มต้นที่หน่วยความจำที่ 768 (§ 300)

POKE 233, 03 : 03 HIGH BYTE
 POKE 232, 00 : 00 LOW BYTE

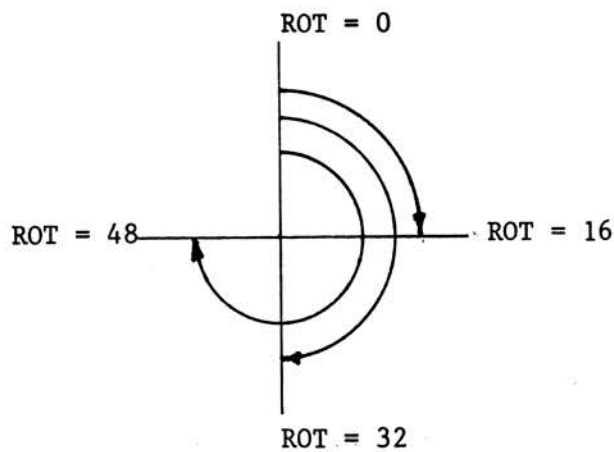
ค่า HIGH BYTE ของ 233 คำนวณได้จาก decimal address ของ SHAPE TABLE โดยใช้สูตรต่อไปนี้

HIGH BYTE = INT (decimal address / 256)
 = INT (768/256) = 3
 LOW BYTE = decimal address - 256 * INT (decimal address/256)
 = 768 - 256 * INT(768/256)
 = 0

เมื่อสร้างรูปร่างและตารางรูปร่างเก็บไว้ในหน่วยความจำแล้ว การจะนำเอารูปร่างที่สร้างออกมาแสดงใน HI - RES GRAPHICS MODE ต้องใช้คำสั่งต่อไปนี้

SCALE เป็นการบอกอัตราส่วนที่จะนำมาแสดงบนจอภาพ ถ้า SCALE = 1 รูปร่างที่ปรากฏบนจอภาพจะเท่ากับที่เขียนไว้ ถ้า SCALE = 5 รูปร่างที่ปรากฏจะเป็น 5 เท่า

ROT ย่อจาก ROTATE เป็นคำสั่งหมุนภาพ โดยจุดที่เริ่มต้นเขียนจะเป็นแกนกลาง ในการหมุน ค่าของ ROT จะอยู่ระหว่าง 0 ถึง 255 ค่าของ ROT แบ่งเป็น 0 16 32 หรือ 48 รูปร่างจะหมุนทวนเข็มนาฬิกา 0 90 180 หรือ 270 องศา (ตามเข็มนาฬิกา)



รูปที่ 3.9 การหมุนของรูป

ถ้ากำหนดว่า ROT อยู่ระหว่าง 2 ค่าใด ค่า ROT จะถูกปัดไปค่าที่มีตัวเลขน้อย เมื่อให้ SCALE = 1 ค่าของ ROT จะหมุน 4 ค่า แต่ถ้า SCALE = 2 ค่าที่ เครื่องหมุนจะมีค่า 8 ค่า

DRAW คือคำสั่งให้วาดภาพ รูปแบบของคำสั่งคือ

DRAW N AT A, B

N : คือหมายเลขของรูปร่างโปรแกรม กรณีที่มีรูปร่างเดียว N = 1

A : คือหมายเลขตำแหน่งของจุดเริ่มต้นในแนวนอน (คือหมายเลขคอลัมน์)

B : คือหมายเลขตำแหน่งของจุดเริ่มต้นในแนวตั้ง (คือหมายเลขแถว)

X DRAW คือคำสั่งให้รูปร่างที่ปรากฏอยู่ในตำแหน่งที่ระบุไว้ภายใน รูปแบบของคำสั่ง คือ

```
X DRAW N AT A, B
```

การสร้างรูปร่างให้ปรากฏบนจอภาพต้องใช้คำสั่งการเข้าสู่ HI-RES GRAPHICS MODE คือ HGR และต้องกำหนดสีของภาพ ก่อนการใช้คำสั่งวาดภาพที่กล่าวมาแล้วเสมอ

เสียง

เสียงที่สร้างอยู่ภายในเครื่อง APPLESOFT มี 2 แบบ คือเสียงกระดิ่ง และเสียงคลิก

1. เสียงกระดิ่ง เกิดจากการใช้ CTRL-G หรือโดยการพิมพ์ CHR\$(7) ตัวอย่างคือ

```
10 PRINT CHR$(7)
```

เสียงที่เกิดขึ้นสามารถทำให้ช้าลง และทำให้ยาวนานได้ โดยการใช้ SPEED และ

FOR--NEXT--LOOP ตัวอย่างคือ

```
10 G$ = CHR$(7) : SPEED = 50
```

```
20 FOR I = 1 TO 10
```

```
30 PRINT G$
```

```
40 NEXT I
```

```
50 SPEED = 255
```

2. เสียงคลิก โดยการใช้คำสั่ง PEEK ไปยังตำแหน่งในหน่วยความจำ คือ

```
10 X = PEEK (-16336)
```

เสียงที่เกิดขึ้นนี้เบามาก การทำให้เสียงดังขึ้นต้องใช้ FOR--NEXT LOOP ช่วย คือ

```
10 S = -16336
```

```
20 FOR I = 1 TO 10
```

```
30 X = PEEK(S)
```

```
40 NEXT I
```

คำสงวน (RESERVED WORDS)

คำที่จะนำมาใช้เป็นตัวแปรไม่ได้มีดังนี้

ABS

ABD

ASC

AT

ATN

CALL

CHR\$

CLEAR

COLOR =

CONT

COS

DATA

DEF

DEL

DIM

END

EXP

FLASH

FN

FOR

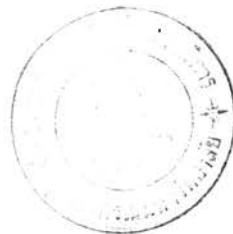
FRE

GET

GOSUB

GOTO

GR



HCOLOR

HGR

HGR2

HIMEM :

HLIN

HOME

HLOT

HTAB

IF

IN#

INPUT

INT

INVERSE

LEFT\$

LEN

LET

LIST

LOAD

LOG

LOMEM :

MID\$

NEW

NEXT

NORMAL

NOT

NOTRACE

ON

ONERR

OR
PDL
PEEK
PLOT
POKE
POP
POS
PRINT
PR#
READ
RECALL
REM
RESTORE
RESUME
RETURN
RIGHT\$
RND
ROT =
RUN

ASCII CODE

รหัสแอสกี	ตัวที่แสดง บนจอภาพ	ตัวบน เป็นพิมพ์
0		CTRL - @
1		CTRL - A
2		CTRL - B
3		CTRL - C
4		CTRL - D
5		CTRL - E
6		CTRL - F
7		CTRL - G
8		CTRL - H หรือ ←
9		CTRL - I
10		CTRL - J
11		CTRL - K
12		CTRL - L
13		CTRL - M หรือ RETURN
14		CTRL - N
15		CTRL - O
16		CTRL - P
17		CTRL - Q
18		CTRL - R
19		CTRL - S
20		CTRL - T

รหัสแอสกี	ตัวที่แสดง บนจอภาพ	ตัวบน เป็นพิมพ์
21		CTRL - U หรือ →
22		CTRL - V
23		CTRL - W
24		CTRL - X
25		CTRL - Y
26		CTRL - Z
27		ESC
28		
29		CTRL - SHIFT - M
30		CTRL - Λ
31		
32	SPACE	SPACE
33	!	!
34	"	"
35	#	#
36	\$	\$
37	%	%
38	&	&
39	'	'
40	((
41))
42	*	*
43	+	+
44	,	,

รหัสแอสกี	ตัวที่แสดง บนจอภาพ	ตัวบนแป้นพิมพ์
45	-	-
46	.	.
47	/	/
48	0	0
49	1	1
50	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7
56	8	8
57	9	9
58	:	:
59	;	;
60	<	<
61	=	=
62	>	>
63	?	?
64	@	@
65	A	A
66	B	B
67	C	C
68	D	D
69	E	E
70	F	F

รหัสแอสกี	ตัวที่แสดง บนจอภาพ	ตัวบนแป้นพิมพ์
71	G	G
72	H	H
73	I	I
74	J	J
75	K	K
76	L	L
77	M	M
78	N	N
79	O	O
80	P	P
81	Q	Q
82	R	R
83	S	S
84	T	T
85	U	U
86	V	V
87	W	W
88	X	X
89	Y	Y
90	Z	Z
91	[
92	\	
93]] (SHIFT-M)
94	^	^ (SHIFT-N)
95	-	

ระบบภาษาไทย

ปัจจุบันการใช้ภาษาในเครื่องไมโครคอมพิวเตอร์ทั่ว ๆ ไป ไม่มีมาตรฐานของรูปแบบภาษาไทยที่แน่นอน ภาษาอังกฤษมีรหัสที่ถือว่าเป็นมาตรฐาน คือรหัสแอสกี (ASCII CODE) รหัสที่เป็นมาตรฐานที่ทำให้การทำงานของเครื่องไมโครคอมพิวเตอร์ทั่ว ๆ ไป เป็นมาตรฐาน และรับรู้ข้อมูลและโปรแกรมระหว่างกันและกันได้โดยง่าย แต่ภาษาไทยไม่มีการกำหนดรหัสมาตรฐาน ขึ้นอยู่กับความพอใจของผู้ทำภาษาไทย ดังนั้นผู้ทำภาษาไทยจึงต้องบัญญัติรหัสขึ้นมาเอง ส่วนใหญ่จะออกแบบให้ง่ายต่อการประมวลผล เช่นถ้าอักษรใด ตรงกับคีย์ตัวใดก็ให้มีรหัสตรงกับรหัสแอสกีของอักษรอังกฤษตัวนั้น การสร้างรหัสที่ไม่เป็นมาตรฐานนี้ ทำให้โปรแกรมภาษาไทยต่าง ๆ ไม่สามารถใช้ร่วมกันได้

การใช้ภาษาไทยของเครื่องไมโครคอมพิวเตอร์ที่ใช้ในการวิจัยนี้มีข้อยกเว้นและการเปลี่ยนแปลง ดังนี้

1. GRAPHICS MODE ใช้ได้แต่ HI-RES GRAPHICS PAGE 1 การสร้างรูปในระบบภาษาไทย สามารถสร้างได้เลย โดยใช้คำสั่ง H PLOT ไม่ต้องพิมพ์คำว่า HGR และสามารถสร้างทั้งรูปภาพและพิมพ์อักษรให้อยู่ในจอภาพเดียวกันได้

2. คำสั่งที่ใช้ในระบบภาษาไทย บางคำสั่งต้องมีการเปลี่ยนแปลงและยกเลิกดังนี้

คำสั่งที่ไม่ใช้

คำสั่งที่ใช้แทน

HOME	PRINT CHR\$(16) และ CTRL-P
HGR และ GR	-
TEXT	-
INVERSE	CTRL-I
FLASH	-
NORMAL	CTRL-N
CALL-868	CTRL-O+CTRL-F
CALL-958	CTRL-O+CTRL-E

3. ความหมายของ CONTROL ทั้งหมดที่ใช้ในระบบภาษาไทย มีดังนี้

CTRL - E	ทำหน้าที่เปลี่ยนจาก Mode ภาษาไทยเป็นภาษาอังกฤษ หรือจากภาษาอังกฤษเป็นภาษาไทย
CTRL - I	INVERSE
CTRL - N	NORMAL
CTRL - P	ล้างจอภาพ (มีความหมายเท่ากับ HOME)
CTRL - O + CTRL - E	CLEAR ตั้งแต่ CURSOR จนถึงสุดบรรทัด
CTRL - O + CTRL - F	CLEAR ตั้งแต่ CURSOR จนถึงสุดจอ
CTRL - B	LOCK TAB (ใช้เฉพาะร่วมกับ CTRL - Q)
CTRL - Q	ให้ CURSOR กลับไปยังที่ CTRL - B LOCK TAB ไว้
CTRL - J	LINEFEED
CTRL - K	ใช้เมื่อพิมพ์ภาษาไทยผิด และต้องการย้อนกลับมาแก้อักษร ที่พิมพ์ผิด (เมื่อยังไม่ได้ออกจาก Mode ภาษาไทย)
CTRL - L	กดเมื่อต้องการนำตัวอักษรที่ถูกเก็บไว้ใน BUFFER CTRL-K ออกมา
CTRL - W	LOCK WINDOW ตั้งแต่ขวามาล่างใช้ CURSOR เป็นหลัก
CTRL - V	LOCK WINDOW ตั้งแต่ซ้ายมาล่างใช้ CURSOR เป็นหลัก
CTRL - Y	ยกเลิก CTRL - W และยกเลิก CTRL - V
CTRL - Z	ยกเลิกการใช้ CTRL ทั้งหมด

การใช้ระบบภาษาไทย ใช้ได้เฉพาะข้อมูลที่ต้องการให้บันทึกหรือแสดงผลบนจอภาพเท่านั้น
คำสั่งต่าง ๆ ที่ใช้ในการเขียนโปรแกรมเพื่อให้เครื่องรับรู้ยังคงเป็นภาษาอังกฤษ

การใช้ภาษาไทยให้กด CTRL - E และกดปุ่ม CAPS LOCK (ไฟสีแดงติด) เพื่อให้
อักษรไทยที่ปรากฏบนจอภาพตรงกับอักษรไทยที่อยู่บนแป้นกด เมื่อใดที่ต้องการออกจากระบบภาษาไทย
ให้กด CTRL-E และ CAPS LOCK กลับคืน

ในระบบภาษาไทยที่ใช้ CURSOR ที่ปรากฏบนจอภาพ จะมีลักษณะและความหมายดังนี้

- หมายถึง อยู่ในระบบภาษาอังกฤษ
- หมายถึง อยู่ในระบบภาษาไทย

การแก้คำที่พิมพ์ผิดในบรรทัดที่พิมพ์อยู่ ทำได้ต่อเมื่อยังไม่ได้ออกจาก Mode ภาษาไทย โดยการกด CTRL - K CURSOR จะเลื่อนไปทางซ้ายมือ จนถึงตำแหน่งที่ต้องการแก้ไข การแก้คำที่พิมพ์ผิดทำได้โดยการพิมพ์อักษรทับลงไป และจะต้องพิมพ์อักษรเรื่อยไปจนกว่าจะจบบรรทัด เพราะการเลื่อน CURSOR ทำได้ เฉพาะจากขวาไปซ้ายเท่านั้น