



ภาควิชาวิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาโท

ชื่อหัวข้อ บอร์ดอักษรไฟวิ่ง แอลอีดีเมทริกซ์
LED Matrix Board

ชื่อนักศึกษา 1. นายชาญชัย พนมเชื้อ รหัสประจำตัว 42035332
2. นายวิจักษ์ ปราโมทย์ รหัสประจำตัว 42035349
3. นายหัตสนัย กุลทัศน์ รหัสประจำตัว 42035360

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์พีระวุฒิ สุวรรณจันทร์

อาจารย์ที่ปรึกษาร่วม อาจารย์สุชิน อาจหาญ

คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. อาจารย์พีระวุฒิ สุวรรณจันทร์	
2. อาจารย์สุชิน อาจหาญ	
3. อาจารย์ปิยะ จิตธรรมมาภิรมย์	
4. อาจารย์อำพล ทองระอา	
5. อาจารย์ไพบูลย์ พวงวงศ์ตระกูล	

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 9 ธันวาคม พ.ศ. 2543 เวลา 13.15 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว
ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 4 เดือน พค. พ.ศ. 2543



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

บอร์ดตัวอักษรไฟวิ่ง แอลอีดีเมตริกซ์

LED MATRIX BOARD



นายชาญชัย พนมเชื้อ
นายวิจักษ์ ปราโมทย์
นายหัตสนัย กุลทีสน์

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต

ปี ๒๕๔๓

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

๒๕๔๓

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

๒๕๔๓

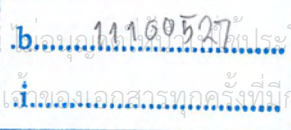
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขที่.....

ปีการศึกษา 2543

เลขทะเบียน 40162

วัน, เดือน, ปี 17 ส.ค. 2544



ฉบับนี้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
หากมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงแหล่งเอกสารทุกครั้งเพื่อการนำไปใช้

ปริญญานิพนธ์

เรื่อง บอร์ดตัวอักษรไฟวิ่ง แอลอีดีเมตริกซ์

LED Matrix Board

วัตถุประสงค์

1. เพื่อศึกษาการใช้งานของไมโคร โพรเซสเซอร์เบอร์ Z80180
2. เพื่อพัฒนาโปรแกรมการใช้งาน LED Matrix Board ให้สามารถติดต่ออุปกรณ์เอาต์พุตและอินพุตได้
3. เพื่อพัฒนาโปรแกรมให้สามารถควบคุมอุปกรณ์แสดงผลได้
4. เพื่อนำอุปกรณ์แสดงผลแสดงข่าวสารรูปแบบต่างๆ ได้จริง

ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจถึงวิธีการใช้งานของไมโคร โพรเซสเซอร์
2. สามารถใช้ภาษาแอสเซมบลีควบคุมการติดต่อระหว่างอุปกรณ์อินพุตและเอาต์พุตได้
3. นำความรู้ต่างๆ ไปประยุกต์ใช้งานจริงได้
4. สามารถนำ LED Matrix Board มาใช้งานในการแสดงผลได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	บอร์ดตัวอักษรไฟวิ่ง แอลอีดีเมตริกซ์
นักศึกษา	นายชาญชัย พนมเชื้อ นายวิจักษ์ ปราโมทย์ นายหัสนัย กุลทัศน์
อาจารย์ที่ปรึกษา	อาจารย์พีระวุฒิ สุวรรณจันทร์
อาจารย์ที่ปรึกษาร่วม	อาจารย์สุชิน อาจหาญ
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์
ปีการศึกษา	2543

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ได้กล่าวถึงการสร้าง เครื่องแสดงตัวอักษรแบบต่อเนื่อง โดยใช้ ไมโครโปรเซสเซอร์เบอร์ Z80180 เป็นตัวควบคุมชุดแสดงผล ที่ประกอบด้วยวงจรเพิ่มแรงดันให้ไดโอดเปล่งแสง ทั้งแนวตั้งและแนวนอน ซึ่งใช้ไดโอดเปล่งแสงทั้งหมด 24×144 ตัว โดยสามารถแสดงข้อความรวม ทั้งยังเปลี่ยนข้อความทันทีที่ต้องการ โดยการป้อนข้อมูลคีย์บอร์ดแสดงภาษาอังกฤษและภาษาไทยได้ นอกจากนี้ยังสามารถควบคุมการแสดงผลและป้อนข้อมูลผ่านทางเครื่องคอมพิวเตอร์ PC ได้ แผงการแสดงผลนี้ยังสามารถแสดงผลตัวอักษรอย่างต่อเนื่อง และสามารถหยุดกะพริบ ขึ้นลงและควบคุมการเลื่อนของข่าวสาร

Thesis Title	LED Matrix Board
Students	Mr. Chanchai Panomchua Mr. Wijk Pramote Mr. Hatsanai Kunlathat
Advisor	Mr. Peerawut Suwanjan
Co-Advisor	Mr. Suchin Adhan
Education Level	Bachelor of Science in Industrial Education
Program in	Electronics and Computer
Academic Year	2000

ABSTRACT

This Thesis is the Digital alphabets display which is controlled by the Z80180 microprocessors, microprocessors is the circuit with increase the light of diode in both vertical and horizontal This display consists of LED matrix board and program. In Display there are 24×144 piece of LED and the program is Visual Basic 6. It can be change message as soon as you input the new message in both English and Thai language.

Moreover it has many design, can display continuously and can be control the moving of the message.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปด้วยดีเนื่องจาก ความอนุเคราะห์ของอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ตลอดจนเพื่อนๆ ที่คอยให้คำแนะนำและให้กำลังใจตลอดมาจนโครงการนี้เสร็จสมบูรณ์ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ คอยให้กำลังใจและทุนสนับสนุนในการทำโครงการครั้งนี้เป็นอย่างดีเยี่ยม และขอขอบพระคุณ ท่านอาจารย์สุชิน ออาจหาญ, อาจารย์ พิระวุฒิ สุวรรณจันทร์ และอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน เป็นอย่างสูง ที่คอยให้การช่วยเหลือ รวมทั้งให้คำแนะนำในการจัดทำและพัฒนาโครงการเป็นอย่างดี ขอขอบคุณ คุณ สุทธิพงษ์ จันทะรัง ที่ช่วยสอนเทคนิคการเขียนโปรแกรม Visual Basic และตลอดจนเพื่อนๆ ทุกคนที่ได้ให้ความช่วยเหลือในการจัดทำโครงการครั้งนี้ คุณงามความดีครั้งนี้ขอมอบแด่ คุณพ่อ คุณแม่ และครูอาจารย์ทุกท่านที่ได้ให้ความรู้ และทุกสิ่งทุกอย่างมาตั้งแต่ต้นจนถึงปัจจุบัน

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ชี้ความสามารถของโครงการ	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ไมโครโปรเซสเซอร์ เบอร์ Z80180	3
2.1.1 โครงสร้างภายในของ Z80180	3
2.1.2 สถาปัตยกรรมของ Z80180	7
2.1.3 ส่วนของอุปกรณ์ I/O	9
2.1.4 คำสั่งเพิ่มเติม 12 คำสั่ง	9
2.1.5 โหมดการทำงาน	11
2.2 ส่วนของการอินเตอร์เฟส	15
2.2.1 อินเตอร์เฟสเป็นพิมพ์	15
2.3 หลักการทำงานของวงจร	17
2.3.1 หลักการทำงานของวงจรติดต่อเป็นพิมพ์	17
2.3.2 หลักการทำงานของวงจรถอดรหัส	19
2.3.3 หลักการทำงานของวงจรถอดรหัสทางด้านหลัก	20
2.3.4 หลักการทำงานของวงจรถอดรหัสทางด้านแถว	21
2.3.5 หน่วยแสดงผล	22
2.3.6 ส่วนป้อนข้อมูล	23
2.4 หลักในการทำให้เกิดภาพในแผงแสดงผล	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4.1 Latch	24
2.4.2 Scan	24
2.5 การเคลื่อนที่ของตัวอักษร	26
2.5.1 การเลื่อนซ้าย	26
2.5.2 การเลื่อนขวา	27
2.6 คอนโทรลภายในของ Visual Basic	28
2.6.1 คอนโทรลภายในทั่วไป	28
2.6.2 คอนโทรลภายในด้านระบบไฟล์	34
2.6.3 คอนโทรลภายในด้านกราฟิก	36
2.6.4 คอนโทรลภายในด้านเวลา	39
2.7 คอนโทรลอาร์เรย์ (Control Array)	39
2.8 การสร้างเมนู	40
2.9 ฟังก์ชันที่เกี่ยวข้องกับ สตริง	41
2.9.1 การแปลงตัวเลขให้เป็น สตริง	42
2.9.2 การตัดคำใน สตริง	43
2.9.3 การทำงานกับรหัส แอสกี ของ สตริง	44
2.9.4 การตัดช่องว่างในสตริงทิ้ง	44
2.10 การใช้ CommonDialog Control	44
2.11 การจัดการข้อมูล	46
2.11.1 การเพิ่มข้อมูล	46
2.11.2 แก้ไขข้อมูล	46
2.11.3 ลบข้อมูล	46
2.11.3 ค้นหาข้อมูล	46
บทที่ 3 การออกแบบการสร้างและการทำงาน	48
3.1 ขั้นตอนในการออกแบบ	48
3.2 ฮาร์ดแวร์	48
3.2.1 วงจรนับเวลาและวันที่	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.3 ส่วนของซอฟต์แวร์	49
3.3.1 ส่วนควบคุมการแสดงผลที่คอมพิวเตอร์	49
3.3.2 การทำงานของ โปรแกรมหลักที่ บอร์ดอักษรไฟวิ่ง	56
บทที่ 4 การทดลองและผลการทดลอง	61
4.1 การทดสอบแผงแสดงผลข้อความ แอลอีดี	61
4.1.1 ตรวจสอบเช็ค ซ็อกเก็ต (Socket) ไมโครโปรเซสเซอร์ เบอร์ Z8180	61
4.1.2 ทดสอบใช้คำสั่ง สแกน ให้ บอร์ดอักษรไฟวิ่ง ทางแนวตั้ง	61
4.1.3 ทดสอบเขียนคำสั่งให้ บอร์ดอักษรไฟวิ่ง ติดหมดทุกดวง	61
4.1.4 ตรวจสอบ สายแพ ที่เชื่อมต่อในวงจร	61
4.1.5 ตรวจสอบแหล่งจ่ายไฟในวงจร	61
4.1.6 ตรวจสอบ ทรานซิสเตอร์ เบอร์ TIP127	61
4.2 การทดสอบโปรแกรมควบคุมแผงแสดงผล แอลอีดี	62
4.2.1 เปิดโปรแกรมให้อยู่ในสภาวะพร้อมใช้งาน	62
4.2.2 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 08 (Font 8 × 08)	64
4.2.3 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 10 (Font 8 × 10)	64
4.2.4 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 12 (Font 8 × 12)	65
4.2.5 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 14 (Font 8 × 14)	66
4.2.6 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 16 (Font 8 × 16)	66
4.2.7 การทดลองการเคลื่อนที่ของตัวอักษรจากขวาไปซ้าย ที่บอร์ดจำลองการแสดงผล	67
4.2.8 การทดลองการเคลื่อนที่ของตัวอักษรจากซ้ายไปขวา ที่บอร์ดจำลองการแสดงผล	68
4.2.9 การทดลองการเคลื่อนที่ของตัวอักษรจากบนลงล่าง ที่บอร์ดจำลองการแสดงผล	68
4.2.10 การทดลองการเคลื่อนที่ของตัวอักษรจากล่างขึ้นบน ที่บอร์ดจำลองการแสดงผล	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.2.11 การทดลองการเคลื่อนที่ของตัวอักษรหลายลักษณะ (Pattern Style)	70
4.2.12 การทดลองแก้ไขข้อความที่อยู่ใน ช่องกำหนดการแสดงผลข้อความ (Text Preview)	72
4.2.13 การทดลองแทรกข้อความที่อยู่ภายในช่องกำหนดข้อความ (Text Preview)	73
4.2.14 การทดลองลบข้อความ (Remove Text)	74
4.2.15 การทดลองบันทึกชุดข้อความ (Save)	75
4.2.16 การทดลองเรียกชุดข้อความที่บันทึกมาใช้งาน (Open)	76
4.2.17 การทดลองสร้างชุดข้อความใหม่ (New)	77
4.3 การทดลองการใช้งานของแผงแสดงผล LED	78
4.3.1 การทดลองให้แผงแสดงผล LED พร้อมใช้งาน	78
4.3.2 การทดลองป้อนข้อมูลเป็นตัวอักษรภาษาไทยและภาษาอังกฤษ	78
4.3.3 การทดลองการแสดงผลการเคลื่อนที่ของตัวอักษร	82
4.3.4 การทดลองการควบคุมการแสดงผลผ่านทาง โปรแกรมควบคุมที่เครื่อง คอมพิวเตอร์	84
บทที่ 5 บทสรุปปัญหา แนวทางการแก้ไขปัญหา และพัฒนา	88
5.1 บทสรุป	88
5.2 ปัญหาและอุปสรรค	88
5.3 การแก้ไข	89
5.4 แนวทางในการพัฒนา	89
ภาคผนวก ก รูปเครื่องต้นแบบ	90
ภาคผนวก ข โปรแกรมการทำงาน	92
ภาคผนวก ค คู่มือการใช้งาน	129
บรรณานุกรม	142
ประวัติผู้แต่ง	143

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 สถานะการทำงานของ ซีพียู	5
ตารางที่ 2.2 ฟังก์ชันที่เกี่ยวข้องกับ สตริง	41



สารบัญรูป

รูป	หน้า
รูปที่ 2.1 โครงสร้างภายในของ Z80180	8
รูปที่ 2.2 รีจิสเตอร์ที่ควบคุมโหมดการทำงาน	11
รูปที่ 2.3 สัญญาณ M1 แอคทีฟ ระหว่างการทำคำสั่ง RETI และเมื่อ M1E = 0	12
รูปที่ 2.4 ไซเคิลการอ่านและเขียนอุปกรณ์ภายนอก เมื่อ IOC = 1	12
รูปที่ 2.5 ไซเคิลการอ่านและเขียนอุปกรณ์ภายนอก เมื่อ IOC = 0	13
รูปที่ 2.6 ลักษณะของขา Z80180	14
รูปที่ 2.7 ขาสัญญาณของแป้นพิมพ์	15
รูปที่ 2.8 สตริมข้อมูลของแป้นพิมพ์	16
รูปที่ 2.9 วงจรติดต่อแป้นพิมพ์	18
รูปที่ 2.10 วงจรถอดรหัสหลัก	20
รูปที่ 2.11 วงจรถอดรหัสทางด้านหลัก	21
รูปที่ 2.12 วงจรถอดรหัสทางด้านแถว	22
รูปที่ 2.13 วงจรส่วนแสดงผล	23
รูปที่ 2.14 การสแกนทางแนวตั้ง	24
รูปที่ 2.15 การสแกนทางแนวนอน	25
รูปที่ 2.16 การค้นข้อมูลในหน่วยความจำแสดงผลไปทางซ้าย	27
รูปที่ 2.17 การค้นข้อมูลในหน่วยแสดงผลไปทางซ้าย	27
รูปที่ 2.18 คอนโทรล CheckBox ในขณะออกแบบ	28
รูปที่ 2.19 คอนโทรล ComboBox ในขณะออกแบบ	29
รูปที่ 2.20 คอนโทรล CommandButton ในขณะออกแบบ	29
รูปที่ 2.21 ผลของการจัดกลุ่มด้วยตัวบรรจุ Frame ที่มีต่อคอนโทรล OptionButton	30
รูปที่ 2.22 คอนโทรล HScrollBar และ VScrollBar ที่ถูกวางลงในฟอร์มในขณะออกแบบ	31
รูปที่ 2.23 คอนโทรล Label ที่ถูกวางลงในฟอร์มในขณะออกแบบ	31
รูปที่ 2.24 คอนโทรล ListBox ในขณะออกแบบ	32
รูปที่ 2.25 คอนโทรล OptionButton ในขณะออกแบบ	33
รูปที่ 2.26 คอนโทรล TextBox ในขณะออกแบบ	33
รูปที่ 2.27 คอนโทรล DriveListBox ในขณะออกแบบและทำงาน	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.28 คอนโทรล DirListBox ในขณะออกแบบ	35
รูปที่ 2.29 คอนโทรล FileListBox ในขณะออกแบบ	36
รูปที่ 2.30 คอนโทรล Image ในขณะออกแบบ	36
รูปที่ 2.31 คอนโทรล Line ในขณะออกแบบ	37
รูปที่ 2.32 คอนโทรล PictureBox ในขณะออกแบบ	38
รูปที่ 2.33 คอนโทรล Shape ในขณะออกแบบ	38
รูปที่ 2.34 คอนโทรล Timer ในขณะออกแบบ	39
รูปที่ 2.35 จอภาพ Menu Editor	40
รูปที่ 2.36 การเลือก Microsoft Common Dialog Control 6.0	45
รูปที่ 3.1 หน้าต่าง วินโดวส์ ของโปรแกรม	49
รูปที่ 3.2 ผังการทำงานของโปรแกรมหลักที่โปรแกรมจำลองการแสดงผล	51
รูปที่ 3.3 ผังการทำงานของโปรแกรมรับข้อมูลที่ต้องการจะแสดงผล	52
รูปที่ 3.4 ผังการทำงานของโปรแกรมเลือกฟอนต์ตัวหนังสือ	53
รูปที่ 3.5 ผังการทำงานของโปรแกรมเลือกรูปแบบการแสดงผล	54
รูปที่ 3.6 ผังการทำงานของโปรแกรมหลัก	56
รูปที่ 3.7 ผังการทำงานของโปรแกรม Runtime	57
รูปที่ 3.8 ผังการทำงานของฟังก์ชัน S๒๑	59
รูปที่ 3.9 ผังการทำงานของฟังก์ชัน S๒๑	60
รูปที่ 4.1 หน้าต่างของโปรแกรมควบคุมการทำงาน บอร์ดอักษรไฟวิ่ง	62
รูปที่ 4.2 ข้อความที่ปรากฏใน Text Input	63
รูปที่ 4.3 ข้อความที่อยู่ใน Text Preview	63
รูปที่ 4.4 ข้อความขนาด Font 8×08	64
รูปที่ 4.5 ข้อความขนาด Font 8×10	65
รูปที่ 4.6 ข้อความขนาด Font 8×12	65
รูปที่ 4.7 ข้อความขนาด Font 8×14	66
รูปที่ 4.8 ข้อความขนาด Font 8×16	67

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.9 การเคลื่อนที่ของข้อความจากขวาไปซ้ายที่บอร์ดจำลองการแสดงผล	67
รูปที่ 4.10 การเคลื่อนที่ของข้อความจากซ้ายไปขวาที่บอร์ดจำลองการแสดงผล	68
รูปที่ 4.11 การเคลื่อนที่ของข้อความจากบนลงล่าง ที่บอร์ดจำลองการแสดงผล	69
รูปที่ 4.12 การเคลื่อนที่ของข้อความจากล่างขึ้นบน ที่บอร์ดจำลองการแสดงผล	69
รูปที่ 4.13 ลักษณะการป้อนข้อความและเลือก Pattern Style หลากหลายแบบ	70
รูปที่ 4.14 เมื่อแสดงข้อความมาจนถึงข้อความสุดท้าย	71
รูปที่ 4.15 การย้อนกลับมาแสดงที่ข้อความแรก	71
รูปที่ 4.16 วิธีการเปลี่ยนข้อความที่อยู่ภายใน Text Preview	72
รูปที่ 4.17 การแก้ไขข้อความที่อยู่ภายใน Text Preview	72
รูปที่ 4.18 การใช้เมาส์คลิกข้อความให้เป็นแถบสี	73
รูปที่ 4.19 ข้อความที่แทรกภายใน Text Preview	73
รูปที่ 4.20 การใช้เมาส์คลิกที่ข้อความให้เป็นแถบสี	74
รูปที่ 4.21 การกดที่ปุ่ม Remove Text	74
รูปที่ 4.22 ใช้เมาส์คลิกที่ เมนู File	75
รูปที่ 4.23 ใช้เมาส์เลือกคลิกที่ บันทึก (Save)	76
รูปที่ 4.24 การเลือกเปิดไฟล์ .led	76
รูปที่ 4.25 เมื่อเปิดไฟล์ .led	77
รูปที่ 4.26 พร้อมป้อนชุดข้อความใหม่	77
รูปที่ 4.27 สภาวะเริ่มต้นการใช้งานในครั้งแรก	78
รูปที่ 4.28 การแสดงผล ฟอนต์ขนาด 8×8	79
รูปที่ 4.29 การแสดงผล ฟอนต์ขนาด 8×10	79
รูปที่ 4.30 การแสดงผล ฟอนต์ขนาด 8×12	80
รูปที่ 4.31 การแสดงผล ฟอนต์ขนาด 8×14	81
รูปที่ 4.32 การแสดงผล ฟอนต์ขนาด 8×16	81
รูปที่ 4.33 การเคลื่อนที่ของข้อความจากทางขวาไปซ้าย	82
รูปที่ 4.34 การเคลื่อนที่ของข้อความจากทางซ้ายไปขวา	83

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.35 การเคลื่อนที่ของข้อความจากบนลงล่าง	83
รูปที่ 4.36 การเคลื่อนที่ของข้อความจากล่างขึ้นบน	84
รูปที่ 4.37 การเคลื่อนที่ของข้อความเมื่อใช้คำสั่ง Right to Left	85
รูปที่ 4.38 การเคลื่อนที่ของข้อความเมื่อใช้คำสั่ง Left to Right	85
รูปที่ 4.39 การเคลื่อนที่ของข้อความเมื่อใช้คำสั่ง Up to Down	86
รูปที่ 4.40 การเคลื่อนที่ของข้อความเมื่อใช้คำสั่ง Down to Up	87



5. สามารถ BACK UP ข้อมูลในกรณีไฟฟ้าดับได้

6. กำหนดความเร็วในการรับส่งข้อมูลกับเครื่องคอมพิวเตอร์ พีซี ได้

1.1 เนื้อหาโดยสังเขป

เนื้อหาในปฏิญญาฉบับนี้แบ่งออกเป็นบทต่างๆทั้งนี้เพื่อความสะดวกต่อการศึกษา และทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาสำคัญดังต่อไปนี้

บทที่ 2 กล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องกับโครงการ จะประกอบด้วยหลักการ ทำงานพื้นฐานของไมโครโปรเซสเซอร์ Z80180 รวมทั้งโครงสร้างภายใน หลักการทำให้เกิดภาพ บนจอแสดงผล การใช้เครื่องคอมพิวเตอร์ พีซี มากำหนดรูปแบบและข้อความที่ต้องการจะนำมา แสดงผลที่จอแสดงผล บอร์ดอักษرفิวิ่ง การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ PC และ บอร์ดอักษرفิวิ่ง ซึ่งจะทำให้ผู้อ่านมีความเข้าใจ อันจะเป็นประโยชน์ต่อการทำความเข้าใจกับ วงจรใช้งานจริง

บทที่ 3 การออกแบบการสร้างและการทำงาน โดยกล่าวถึง หลักการสร้าง และการออกแบบ ทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ การนำเครื่องคอมพิวเตอร์ PC มาทำการโปรแกรมส่วน ควบคุมรูปแบบการแสดงผลรวมทั้งหลักการทำงานในส่วนต่างๆ ซึ่งจะทำให้ผู้อ่านมีความเข้าใจการ ทำงานโดยรวมของโครงการนี้

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการทดลองวงจรต่างๆ การทดลองการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ พีซี และ บอร์ดอักษرفิวิ่ง รวมทั้งในส่วนของการโปรแกรมด้วย เพื่อหาผลการทดสอบประสิทธิภาพการทำงานว่าเป็นไปตามวัตถุประสงค์หรือไม่

บทที่ 5 บทสรุปปัญหาแนวทางแก้ไขและพัฒนา ได้สรุปปัญหาที่เกิดขึ้นและวิธีการแก้ไข และการพัฒนา เพื่อให้โครงการนี้มีประสิทธิภาพและสามารถใช้งานได้อย่างกว้างขวางมากขึ้น

ในภาคผนวกได้แสดงรายละเอียดในส่วนต่างๆดังต่อไปนี้

ภาคผนวก ก รูปเครื่องต้นแบบ

ภาคผนวก ข โปรแกรมการทำงาน

ภาคผนวก ค คู่มือการใช้งาน

บทที่ 2

ทฤษฎีและหลักการ

2.1 ไมโครโปรเซสเซอร์ เบอร์ Z80180

2.1.1 โครงสร้างภายในของ Z80180

Z80180 เป็น ซีพียู ที่มีความสามารถสูงที่ได้รวม ชิพ (CHIP) สำคัญอื่นๆไว้ใน ซีพียู เดียว จึงทำให้มีลักษณะคล้ายกับ ซีพียู ที่ใช้งานคอนโทรล ในจำพวก ชิพเดี่ยว (Single Chip) มีข้อดีคือ เป็นระบบเล็กราคาถูก แต่ข้อเสีย คือ การโปรแกรม คอนโทรล ก่อนข้างยากในตอนเริ่มต้นและกับระบบงานที่ใหญ่ขึ้นจะเขียนโปรแกรมได้ยาก แต่ Z80180 ทางด้านโปรแกรมจะสะดวกอย่างมาก เพราะคำสั่งที่ใช้มีมาก และตรงไปตรงมา ทั้งคู่มือภาษาไทยและตัวอย่างการใช้งานอย่างมากมาย เพราะ ซีพียู Z80180 นี้เป็น Supper Compat Z80 คือคำสั่งทั้งหมดยังเป็น Z80 และได้เพิ่มชุดคำสั่งขึ้นมาเพื่อเพิ่มความสะดวกในการใช้งานขึ้นอีก

เมื่อมองดูระบบ ไมโครคอนโทรลเลอร์ ชิพเดี่ยว แล้ว Z80180 จะดีกว่าตรงที่ไม่มี Rom Ram และ พอร์ต (ใน Z80180 รุ่นใหม่จะมี ROM RAM หรือ พอร์ต แล้ว) แต่ถ้าในระดับงานอุตสาหกรรมระบบของ Z80180 กับ Chip Micro Controller แล้วจะไม่ต่างกันเลยเพราะความต้องการเนื้อที่ในการเก็บข้อมูลมากและ พอร์ต มากตามมาจึงทำให้ต้องต่อวงจรภายนอกขึ้น จึงทำให้ Z80180 ในระดับงาน คอนโทรล อุตสาหกรรม คล่องตัวมากกว่าเพราะภายใน Z80180 ประกอบเป็น CMOS , OSCILATOR ในตัว ทำงานที่ 10 เมกะเฮิร์ตซ์ – 20 เมกะเฮิร์ตซ์, MMU ชิพ อ้าง หน่วยความจำ (Memory) ได้ 1 เมกะไบต์ (MBYTE), DMA 2 Channel , Port สื่อสาร UART 2 Channel , Clock Serial I/O , 16 Bit Time Counter และเกี่ยวกับ พอร์ต สื่อสาร สามารถทำ Multi Processor Communication

1) ขาการใช้งาน

A0-A19

แอดเดรสบัส (Address Bus) 20 เส้น ระหว่าง รีเซต (Reset) จะเป็น High Impedance

BUSAK

BUS Acknowledge เป็นขา Output Active Low ทำงาน ก็ต่อเมื่อ Z80180 ตอบสนองต่อการขอบัส (Bus) ของ BUSRQ และจะทำให้ บัสข้อมูล บัสแอดเดรส (Bus Address) และ สัญญาณคอนโทรล บางเส้นเป็น ไฮอิมพีแดนซ์ (High Impedance)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CKA0 , CKA1

Asynchronous Clock 0 และ 1 เป็นขาสัญญาณนาฬิกา แบบ 2 ทิศทาง คือ จะใช้เป็นขาอินพุต หรือ เอาต์พุต ก็ได้

CKS

Serial Clock เป็นขา Clock 2 ทิศทางของ CSI/O

CLOCK

เป็นขา เอาต์พุต โดยจะเป็นครึ่งหนึ่งของ คริสตอล (X'TAL) หรือ Clock Out เช่น คริสตอล 12 เมกะเฮิร์ตซ์ Z80180 จะ ทำงานที่ 6 เมกะเฮิร์ตซ์

CTS0-CTS1

Clear to send 0 และ 1 เป็นขา Input Active Low ใช้ในการควบคุม โมเด็ม(Modem)

D0-D7

Data Bus เป็นแบบ 2 ทิศทาง

DCDO

Data Carrier Detect 0 เป็นขา Input Active Low ใช้ควบคุมในการติดต่อกับ โมเด็ม ของ Ascii Channel 0

DREQ0-DREQ1

DMA Request 0 และ 1 เป็นขา Input Active Low ใช้ในการขอ DMA และขานี้จะโปรแกรมได้ว่าให้ตรวจสัญญาณที่ขอบหรือระดับได้

E

Enable Clock เป็นขา Output Active Low ซึ่งใช้ซิงค์ การทำงานกับอุปกรณ์ อุปกรณ์ภายนอก ระหว่างการทำงานเกี่ยวกับ บัส และใช้เชื่อมต่อกับอุปกรณ์ในตระกูล 68XX และ 80XX

HALT

เป็นขา Output Active Low จะทำงานเมื่อคำสั่ง HALT หรือ SLP

INT1 , INT2

เช่นเดียวกับ Into แต่มีระดับความสำคัญ รองลงมาตามลำดับ

IORQ

เป็นขา เอาต์พุต เพื่อบอกว่ากำลังติดต่อกับ I/O หรือขา IOE ใน 64180

M1

Machine Cycle 1 เป็นขา Output Active Low จะทำงาน เมื่อ FETCH OP-CODE 64180

LIR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NMI

Non Maskable Interrupt เป็นขา Input Active Low ขานี้จะตอบรับการอินเทอร์รัพท์ เสมอ โดยไม่สามารถหยุดด้วย ซอฟต์แวร์

RD

เป็นขาที่ใช้ทำการอ่านข้อมูลจาก หน่วยความจำ หรือ I/O

RFSH

เป็นขาที่ให้ Address Low (A0-A7) ไป Refresh Dynamic Ram หรือขา REF ของ 64180

RTS0

Request to Send เป็นขา Output Active Low ขานี้ใช้โปรแกรมสัญญาณควบคุมโมเด็มของ Ascii Cannel 0

RXA0 , RXA 1

Receive Data 0 และ 1 เป็นขารับสัญญาณจาก Serial ของ Ascii

RXS

Clock Serial Receive Data เป็นขารับสัญญาณ Serial ของ CSIO

ST

Status เป็นขา Output Active Low ใช้แสดงสถานะการทำงาน ซีพียู โดยร่วมกับ M1 และ Halt ดังตาราง

ตารางที่ 2.1 สถานะการทำงานของ ซีพียู

ST	HALT	M1	Operation
0	1	0	CPU operation (1 st op-code fetch)
1	1	0	CPU operation (2 nd op-code and 3 rd op-code fetch)
1	1	1	CPU operation (MC except for op-code fetch)
0	X	1	DMA operation
0	0	0	HALT mode
1	0	1	SLEEP mode (including SYSTEM STOP MODE)

หมายเหตุ X: Don't care

MC: Machine cycle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEND0-TEND1

Transfer End 0 และ 1 เป็นขา Output Active Low ใช้แสดงถึงว่าทำ DMA ลีนสุดลงแล้ว

TOUT

Timer out ใช้กำเนิดพัลส์จาก PRT Channel 1

TXS

Clock Serial Transmit Data เป็นขาส่งข้อมูล Serial ของ CSIO

WAIT

Input Active Low จะถูกตรวจที่ขอบขาของสัญญาณนาฬิกา ลูกที่ 2 ของทุกๆ Machine เพื่อเป็นการรอให้อุปกรณ์ภายนอกทำงานให้ทันกับการทำงานของ ซีพียู

WR

ใช้สำหรับการส่งข้อมูลไปยัง I/O หรือหน่วยความจำ

X'TAL

เป็นขาที่ใช้ต่อกับคริสตัล

2) ขาที่ Multilex**A18/TOUT**

ระหว่าง รีเซต จะเป็น A18 แต่ถ้ามีการเลือก เซต (Set) BIT TOC1 หรือ TOC0 ใน Timer Control Register (TCR) ก็จะทำหน้าที่เป็น TOUT

CKA0/DREQ0

ระหว่าง รีเซต ขานี้จะเป็น CKA0 แต่ถ้า DM1 หรือ SM1 ใน DMA Mode Register (DMODE) ถูก เซต เป็น 1 จะเป็นขา DREQ0

CKA1/TEND0

ระหว่าง รีเซต จะเป็นขา CKA1 แต่ถ้า BIT CKA1D ใน Ascii ถูก เซต จะเป็นขา TEND0

RXS/CTS1

ระหว่าง รีเซต ขานี้จะเป็นขา RXS ถ้า BIT CTS1E ใน Ascii ถูก เซต จะเป็นขา CTS1

3) คุณสมบัติทั่วไปของ Z80180

1. ทำงานด้วยสัญญาณนาฬิกาสูงถึง 10 เมกะเฮิร์ตซ์
2. มีหน่วยจัดการหน่วยความจำที่เรียกว่า MMU ทำให้มีขีดความสามารถอ้างแอดเดรสได้ถึง 1 เมกะไบต์ หรือ 512 กิโลไบต์ และยังใช้ในขอบเขตเดิมได้คือ 64 กิโลไบต์ ได้
3. มีหน่วยควบคุม DMA ใน ชิพ ที่ทำให้ควบคุม DMA ได้ 2 ช่อง เพื่อให้เกิดกลไกในการถ่ายเทข้อมูลระหว่าง อินพุต เอาต์พุต กับหน่วยความจำ และหน่วยความจำ กับหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สร้าง Wait สำหรับเชื่อมโยงกับ หน่วยความจำ หรืออุปกรณ์ อินพุต เอาต์พุต ที่ซ้ำ
5. โปรแกรมการทำงานสำหรับการ Refresh Ram ได้
6. มีพอร์ตอนุกรมแบบ Asynchronous เต็มรูปแบบ 2 พอร์ต พร้อมวงจรการสร้างอัตราบ็อต
7. มีพอร์ต สร้างสัญญาณ Timing ที่ทำให้เกิดสัญญาณได้สูงถึง 400 กิโลไบต์/วินาที ที่ความเร็ว 8 เมกะเฮิร์ตซ์
8. มีวงจรควบคุม อินเทอร์รัพท์ภายนอกได้ 4 สัญญาณ และ อินเทอร์รัพท์ภายในได้ 8 สัญญาณ
9. มีวงจรการสร้างความพิลาจากคริสตอล
10. มีคำสั่งพิเศษเพิ่มเข้ามา

2.1.2 สถาปัตยกรรมของ Z80180

Z80180 ประกอบด้วยหน่วยประมวลผลกลางที่มีการควบคุมการทำงานสูงพร้อมกับระบบที่มีความปลอดภัย หน่วยประมวลผลกลางจะประกอบด้วย 5 ฟังก์ชัน

1) การกำเนิดสัญญาณนาฬิกา (Clock)

การกำเนิดสัญญาณนาฬิกา ได้โดยวงจรกำเนิดสัญญาณนาฬิกาจากคริสตอลภายนอกและสัญญาณนาฬิกาภายนอก

2) การควบคุม Bus State

ทำได้โดยการใช้การควบคุมสถานะของ Bus (Bus State Wait) ใช้เป็นวงจรควบคุมและสร้างสัญญาณที่เป็นกิจกรรมต่างๆ ของ บัส โดยกำเนิด ไชเคิล ของ บัส ที่เชื่อมต่อกับ ภายนอก รวมทั้งการสร้างสถานะรอ (Wait) Reset การรีเฟรช DRAM และการจัดการเรื่องสัญญาณควบคุม บัส ขณะทำการ DMA

3) การควบคุมอินเทอร์รัพท์

ใช้วงจรควบคุมอินเทอร์รัพท์ คอยตรวจสอบสัญญาณอินเทอร์รัพท์ และจัดลำดับความสำคัญของอินเทอร์รัพท์

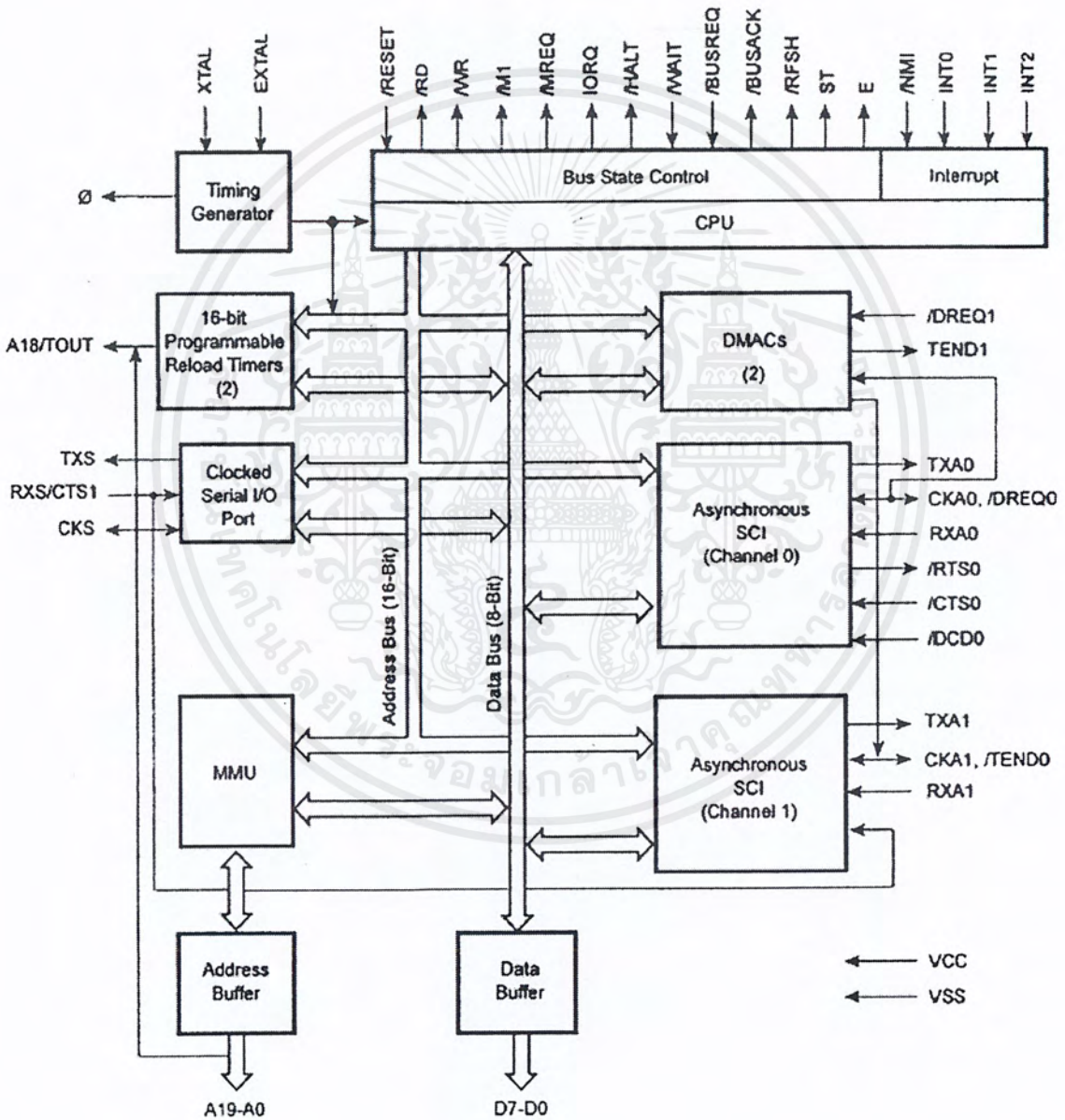
4) การจัดการหน่วยความจำ (หน่วยความจำ Management Unit)

หน่วยจัดการหน่วยความจำ (MMU) ทำหน้าที่สร้างช่องหน้าต่างเพื่อ Map Memory 64 กิโลไบต์ ไปอยู่ใน 1 เมกะไบต์ หรือ 512 กิโลไบต์ ของหน่วยความจำ ฟิสิคัล การจัดการหน่วยความจำนี้ ใช้วิธีการสร้างหน่วยความจำนี้เป็นส่วนๆ โดยมีโหมดจัดการหน่วยความจำหลายโหมด

ใช้ซอฟต์แวร์จัดการ โดยมีคำสั่งที่สั่งการทำงานของ MMU ที่สร้างขึ้นเป็นพิเศษ นอกเหนือคำสั่ง Z80

5) หน่วยประมวลผลกลาง (Control Processing Unit)

เป็นส่วนการทำงานเหมือน หน่วยประมวลผลกลางเดิม คือ สามารถใช้ ซอร์ฟแวร์ ของ Z80



รูปที่ 2.1 โครงสร้างภายในของ Z80180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ส่วนของอุปกรณ์ I/O จะประกอบด้วย 4 ฟังก์ชัน ดังต่อไปนี้

1) การจัดการหน่วยความจำโดยตรง (Direct Access หน่วยความจำ)

ใช้ควบคุมการทำงาน DMA 2 ช่อง เพื่อควบคุมการถ่ายเทข้อมูลระหว่างหน่วยความจำกับหน่วยความจำ หรือจากหน่วยความจำกับอุปกรณ์ อินพุต/เอาต์พุต ส่วนของการถ่ายเทข้อมูล กับหน่วยความจำนี้ทำได้เต็มที่ ขนาดของหน่วยความจำ 1 เมกะไบต์ หรือ 512 กิโลไบต์ ตามค่าแอดเดรสเพื่อส่งถ่ายข้อมูลคร่อมขอบเขตของ 64 กิโลไบต์

2) การติดต่อสื่อสารอนุกรมแบบ Asynchronous

ใช้วงจรควบคุมการสื่อสารแบบอนุกรม เป็นวงจรที่มีพอร์ตอนุกรมแบบ Full Duplex 2 พอร์ตแบบ UART รวมทั้งส่วนของการโปรแกรมวงจรสร้างอัตราความถี่ที่กำหนดบิตต่อเรต สัญญาณการควบคุม โมเด็ม สัญญาณเพื่อใช้ในการส่งถ่ายข้อมูลแบบอัตรา ความเร็วสูงเพื่อใช้การทำงานแบบหลายหน่วยประมวลผลกลาง โดยมีการนส่งถ่ายข้อมูลร่วมกัน นอกจากนี้ยังทำงานร่วมกับ DMA เพื่อส่งถ่ายข้อมูลความเร็วสูงอีกด้วย

3) การโปรแกรมเวลา

ใช้วงจร Time ที่โปรแกรมได้มี 2 Channel ที่แยกจากกันได้สามารถกำหนดค่าตัวเลขการนับ โดยมีรีจิสเตอร์รับค่าโปรแกรมแบบ 16 บิต

4) Clock Serial I/O Channel (CS I/O)

วงจรการสร้างสัญญาณนาฬิกา พอร์ต อินพุต/เอาต์พุต หรือ CS I/O เป็นวงจรแบบ Half duplex เพื่อใช้สร้างสัญญาณนาฬิกา รูปแบบและความถี่ที่แตกต่างกันตามการ โปรแกรมที่ควบคุม

2.1.4 คำสั่งเพิ่มเติม 12 คำสั่ง

นอกเหนือจากคำสั่งเดิมของไมโคร โปรเซสเซอร์เบอร์ Z80 แล้ว ไมโครโปรเซสเซอร์เบอร์ Z80180 ยังมีคำสั่งเพิ่มเติมอีก 12 คำสั่ง ดังต่อไปนี้

SLP

เมื่อใช้คำสั่งนี้ ซีพียู จะหยุดทำงานบางอย่างทำให้ใช้กำลังงานต่ำ

MLT

MULTPLY ใช้สำหรับคูณเลข 8 บิต (BIT) โดยผลลัพธ์จะเป็น 16 บิต โดย รีจิสเตอร์ (REGISTOR) ที่ใช้ในการคูณอาจจะเป็น BC, DE, HL หรือ SP โดยผลลัพธ์จะได้ที่ รีจิสเตอร์ คู่ นั้น

OTIM, OTIMR, OTDMR, OTDMR-BLOCK I/O

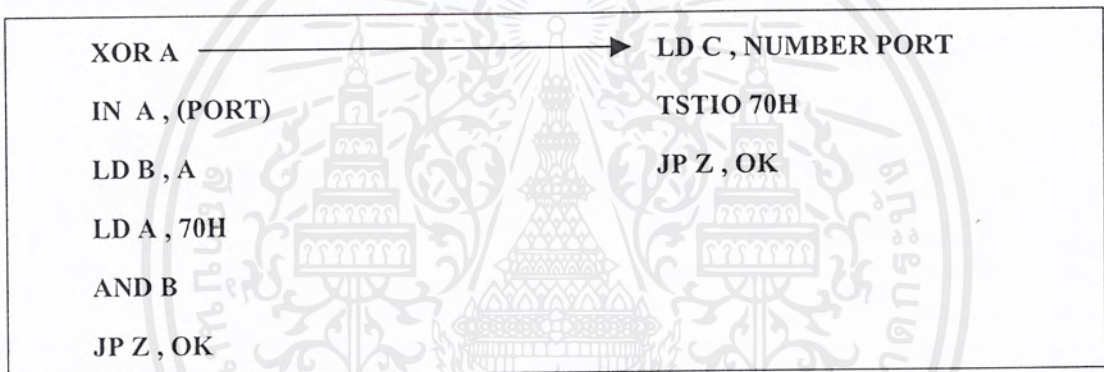
เป็นคำสั่ง เอาต์พอร์ต (Out Port) เป็น Block ของ พอร์ต แอดเดรส (Address) คำ A0- A7 เท่านั้น Out ข้อมูลเป็น Block โดยที่ พอร์ต เพิ่มหรือลดตามจำนวนข้อมูล โดยใช้ HL เป็นตัวชี้ ข้อมูลที่จะ Out ออกไป และ C เป็น หมายเลข พอร์ต (Number Port) ในคำสั่ง OTIM และ OTDM ก็คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเพิ่มค่า HL ที่ชี้ขึ้นเป็นหนึ่งหรือลดลง 1 ตามด้วย พอร์ต เพิ่มขึ้นหรือลดลงด้วยและค่า B จะลดลง 1 ซึ่ง B จะเป็น Counter ในการส่ง Data ส่วน OTIMR และ OTDMR จะมีลักษณะเช่นเดียวกับ OTIM และ OTDM เพียงแต่จะทำการส่งข้อมูลเพิ่มขึ้นหรือลงและ พอร์ต Number เพิ่มขึ้นหรือลดลงตามค่า B จนกระทั่ง B = 0

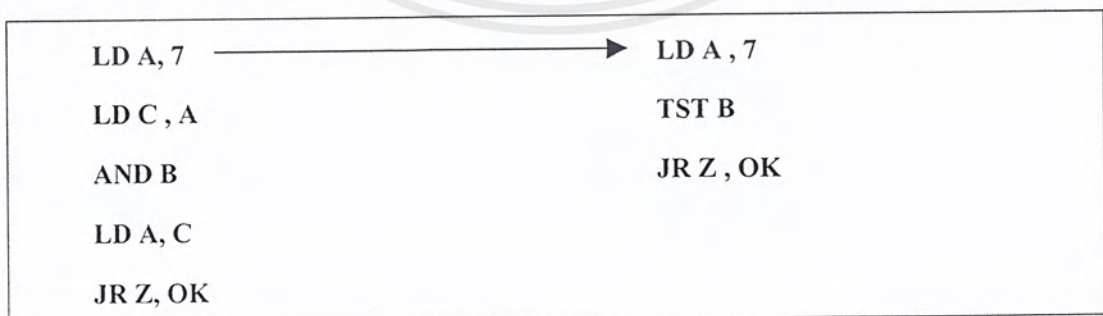
TSTIO m

ใช้สำหรับ TEST I/O พอร์ต คือ จะทำการอ่านค่า พอร์ต ที่กำหนดโดย รีจิสเตอร์ C เข้ามา แล้วทำการ And กับ Data 8 บิต ที่ต้องการ โดยค่าข้อมูลที่ IN เข้ามานั้น แต่จะให้ผลที่ FLAG และ พอร์ต ที่ IN เข้ามาจะเป็นเฉพาะ แอดเดรส ต่ำ A0-A7 เท่านั้นสามารถเปรียบเทียบเป็นโปรแกรมได้ ดังนี้



TST g – TEST REGISTOR

โดยที่ค่ากำหนดใน รีจิสเตอร์ จะ And กับ Accumulator ซึ่งจะให้มีผลต่อ Flag ตามคำสั่ง And แต่ค่าใน Accumulator และ รีจิสเตอร์ ไม่เปลี่ยนแปลง เช่น ตัวอย่าง



TST m - TEST IMMEDIATE

เช่นเดียวกับ รีจิสเตอร์ เพียงแต่ข้อมูลเป็น Data โดย ตรงที่ And กับ Accumulator

TST (HL) - TEST IMMEDIATE

คือ จะนำค่าใน หน่วยความจำ ที่ถูกชี้โดย HL AND กับ ACCUMULATOR โดยค่าทั้ง 2 ไม่เปลี่ยนแปลงแต่ผลการกระทำที่ FLAG

IN0 g (m) – INPUT , IMMEDIATE I/O

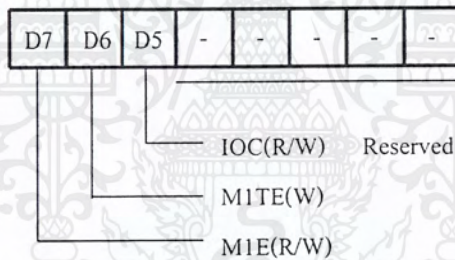
IN ค่าจาก พอร์ต 8 บิต (A0-A7) มายัง รีจิสเตอร์ ใดๆก็ได้ A, BC , DE , HL

OUTO (m) , OUTPUT , IMMEDIATE T/O

OUT ค่าจาก รีจิสเตอร์ ใดๆไปยัง พอร์ต 8 บิต (A0-A7) รีจิสเตอร์ ก็มี A , BC , DE , HL

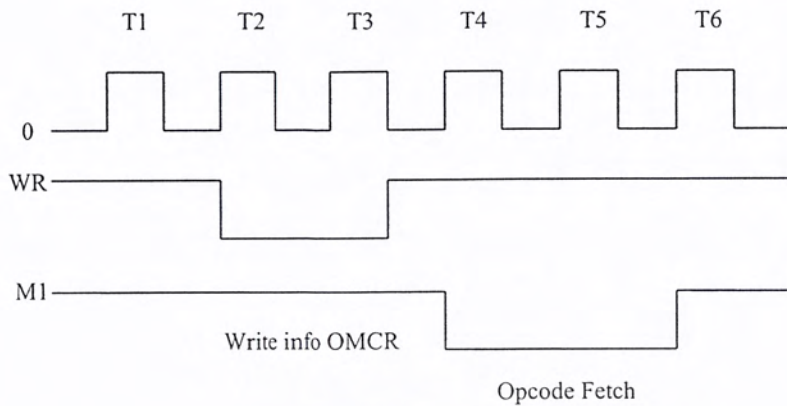
2.1.5 โหมดการทำงาน

Z80180 สามารถกำหนดการทำงานเหมือนกับ 64180 โหมดการทำงานนี้สามารถที่จะกำหนดได้โดยควบคุมขาสัญญาณของ M1 , IORQ , WR และ RD ส่วนรีจิสเตอร์ที่ควบคุมโหมดการทำงานสามารถควบคุม M1 , Timing ของ IORQ , RD และ WR และการทำงาน REIT ดังรูปที่ 2.2



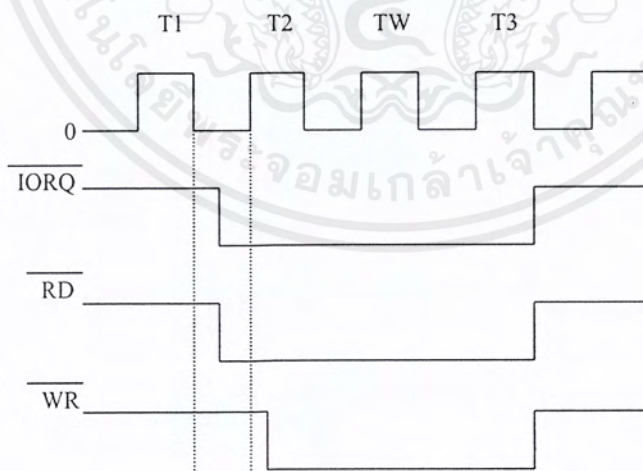
รูปที่ 2.2 รีจิสเตอร์ที่ควบคุมโหมดการทำงาน

(Mienble) บิตนี้จะควบคุมให้เอาต์พุตของ M1 และกำหนดเป็น 1 ในระหว่างเซตเมื่อ MTE = 1 เอาต์พุตของ M1 จะเป็น 0 ในระหว่างการไหลเคลิการเฟสคำสั่งการตอบรับการอินเตอร์รัพท์แบบ INTO และเมซซึนไหลเคลิแรกของการรับอินเตอร์รัพท์แบบ NMI การเกิดแบบนี้จะเป็นสาเหตุทำให้สัญญาณ M1 แอคทีฟ ระหว่างการทำคำสั่ง RETI และเมื่อ M1E = 0 , เอาต์พุตของ M1 แอคทีฟ ตามปกติ ดังรูปที่ 2.3



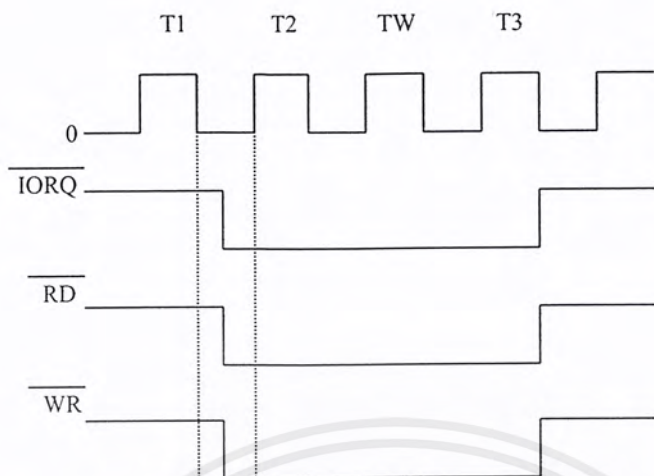
รูปที่ 2.3 สัญญาณ M1 แอคทีฟ ระหว่างการทำคำสั่ง RETI และเมื่อ M1E = 0

MITE (M1 Temporary Enable) บิตนี้จะควบคุมการทำงานของสัญญาณ M1 โดยจะเซตเป็น 1 เสมอในระหว่างการรีเซต หน้าที่จะถูกใช้ในการอินเตอร์รัพต์ภายในของ Z80180 เมื่อคำสั่งควบคุมเขียนถึง Z80PIO เพื่อทำการอินเทอร์พิว แต่จะไม่อินาเบิลสัญญาณ M1 จะแอคทีฟเมื่อ MITE = 1 จะไม่มีการเปลี่ยนแปลงการทำงานของสัญญาณ M1 เมื่อ MITE = 0 , เอาต์พุตของ M1 จะแอคทีฟที่ไซเคิลของการเฟสคำสั่งต่อไป IOC : บิตนี้จะควบคุม timing ของ IORQ และ RD และจะเป็น 1 เมื่อรีเซตเมื่อ IOC = 1 หน้าที่ของสัญญาณ IORQ และ RD จะเหมือนกับ Z64108 ดังรูปที่ 2.4



รูปที่ 2.4 ไซเคิลการอ่านและเขียนอุปกรณ์ภายนอก เมื่อ IOC = 1

เมื่อ IOC = 0 , หน้าที่ของสัญญาณ IORQ และ RD จะเหมือนกับ Z 80 ดังรูปที่ 2.5 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ไช้กิจการอ่านและเขียนอุปกรณ์ภายนอก เมื่อ $IOC = 0$

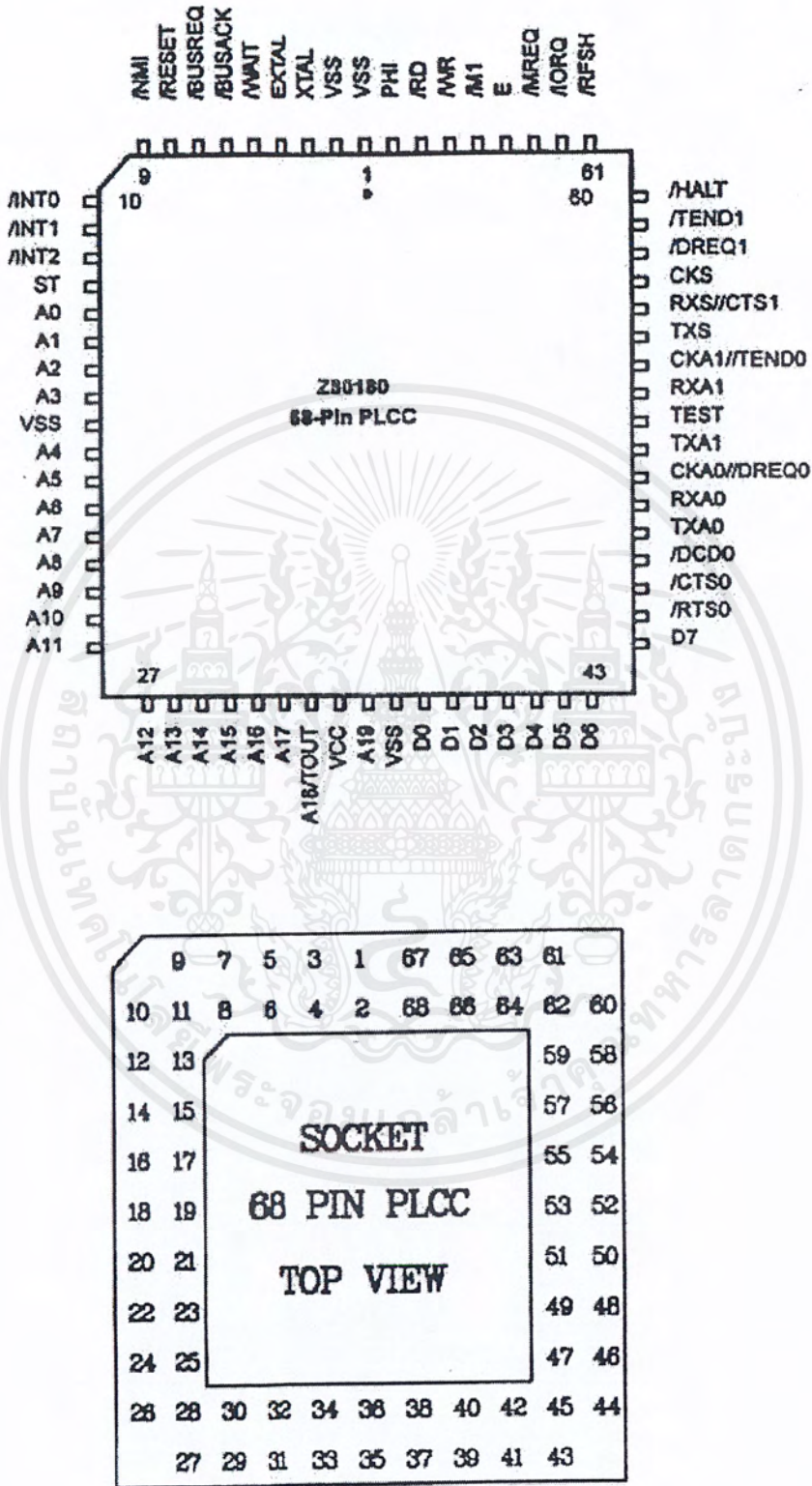
ในโหมดการทำงานปกติ Z80180 จะใช้กระแสเพียงเล็กน้อย ส่วนในโหมด Sleep และ SYSTEM STOP ยังคงลดกำลังในการทำงานด้วย

โหมด Sleep

Z80180 หรือ 64180 มีคำสั่ง SLP เพิ่มเข้ามาคำสั่งนี้มีขนาด 2 ไบต์ เมื่อหน่วยประมวลผลกลาง ทำงานตามคำสั่งนี้จะมีผลในเรื่องการหยุดทำงานของ หน่วยประมวลผลกลาง โดยสัญญาณนาฬิกา จะหยุด เพื่อลดค่าพลังงานที่ใช้ใน หน่วยประมวลผลกลาง ส่วนวงจรที่ใช้ในการสร้างสัญญาณนาฬิกาที่คริสตอลยังคงทำงานอยู่ หน่วยประมวลผลกลางยังคงรับสัญญาณอินเตอร์รัพท์ทั้งภายในและภายนอกได้ ไช้เคิล การรีเฟรชหน่วยความจำก็จะหยุดหมดแม้กระทั่ง DMAC ก็ไม่ทำงาน ส่วนสัญญาณ BUSRQ นั้น หน่วยประมวลผลกลาง ยังคงรับได้ แอดเดรสของ หน่วยประมวลผลกลาง จะมีลอจิกเป็น โดยที่ อุปกรณ์ อินพุต เอาต์พุต ในชิพไม่หยุดการเรียกเข้ามาทำงานใหม่มีวิธีการได้ 2 วิธี

1. การรีเซต จะทำให้ หน่วยประมวลผลกลาง กลับมาเริ่มทำงานใหม่ที่แอดเดรส 0000H
2. การอินเตอร์รัพท์ ซึ่งรับอินเตอร์รัพท์ทั้งภายในและภายนอกโหมด IOSTOP

เป็นโหมดที่หยุดการทำงาน เฉพาะส่วนอุปกรณ์ อินพุต เอาต์พุต ที่อยู่ภายใน ชิพ โดยที่ หน่วยประมวลผลกลาง ยังคงทำงานอยู่ การหยุดทำงานส่วนนี้ทำได้โดยการควบคุมโปรแกรมโดยสั่งเข้ามา ควบคุมบิต IOSTOP ที่อยู่ในรีจิสเตอร์ควบคุม อินพุต เอาต์พุต (ICR) ด้วยค่า "1" ส่วนนี้ จะทำให้วงจร Asynchronous วงจร CSI/0 และ PRT หยุดวางกลับเข้ามาทำงานใหม่ ด้วยการโปรแกรมบิต ICR เป็น "0"



รูปที่ 2.6 ลักษณะของขา Z80180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด System Stop

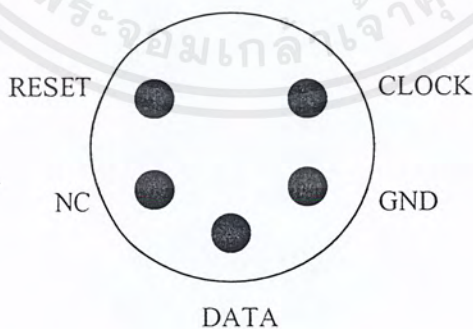
เป็นโหมดที่หยุดการทำงานทั้งหมด โดยรวบรวมโหมด Sleep และโหมด IOSTOP เข้าด้วยกัน การเข้าสู่ SYSTEM ทำได้โดยการกำหนดบิต ICR เป็น “1” และทำคำสั่ง SLP

รีจิสเตอร์คำสั่งและส่วนควบคุม (Instruction Register and Control) ในการทำการเฟสหน่วยประมวลผลกลาง จะอ่านคำสั่งจากหน่วยความจำที่เป็นส่วนของโปรแกรมโดยรอคำสั่งนั้นมาเก็บไว้ใน IR เพื่อทำการถอดรหัสคำสั่ง และส่งสัญญาณควบคุมการทำงานภายใน หน่วยประมวลผลกลาง หรือ ควบคุมการทำงานของระบบสัญญาณควบคุมเหล่านี้ จะออกมาในจังหวะต่างๆ กัน เพื่อใช้ควบคุมระบบในการทำงานต่อไป

2.2 ส่วนของการอินเตอร์เฟส

2.2.1 อินเตอร์เฟสเป็นพินท์

เป็นพินท์จัดเป็นอุปกรณ์สื่อสาร 2 ทิศทาง ที่สามารถส่งสัญญาณหรือข้อมูลระหว่างเป็นพินท์หรือระบบงาน เป็นพินท์มีทั้งหมด 101 คีย์ โดยส่วนกลางของเป็นพินท์นั้นจะมีลักษณะเหมือนกับเป็นพินท์ของเครื่องพิมพ์ดีด สำหรับเป็นพินท์ด้านข้างของเป็นพินท์นั้น จะมีลักษณะพิเศษ ต่างจากเป็นพินท์ของเครื่องพิมพ์ดีด ทางด้านขวาจะเป็นคีย์ที่ใช้อ่านตัวเลขและใช้ควบคุมการเลื่อนตำแหน่งของเคอร์เซอร์พร้อมกันไปด้วย สำหรับเป็นพินท์ทางด้านซ้ายจะเป็นคีย์ที่ฟังก์ชันคีย์ซึ่งสามารถใช้ฟังก์ชันการทำงานให้แก่ฟังก์ชันคีย์แต่ละคีย์ได้ การติดต่อเป็นพินท์เข้าระบบจะทำผ่านทางสายสัญญาณ จำนวน 4 เส้นซึ่งใช้ส่งสัญญาณและจ่ายไฟให้แก่เป็นพินท์ ดังแสดงในรูปที่ 2.7

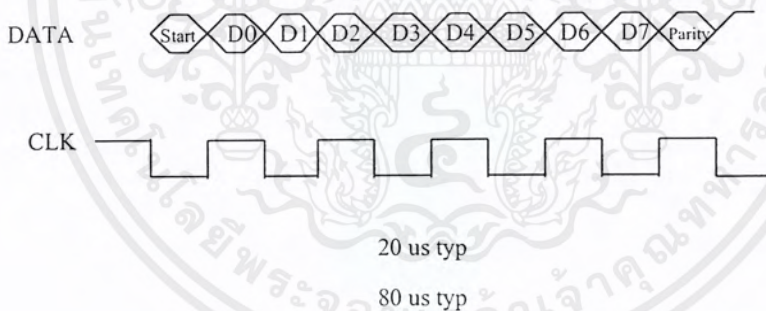


รูปที่ 2.7 ขาสัญญาณของเป็นพินท์

1) ขานาฬิกาและขา DATA

เป็นพินท์และระบบสื่อสารจะต้องประกอบไปด้วยเส้นสัญญาณนาฬิกา และข้อมูลซึ่งเมื่อต้องการติดต่อหรือทำการส่งข้อมูล ขาสัญญาณนาฬิกาและข้อมูลจะมีสถานะเป็น Negative และถ้าไม่มีการติดต่อสื่อสารหรือส่งข้อมูลจะมีสถานะเป็น Positive เป็นพินท์จะส่งสัญญาณขนาด 8 บิต จำนวนหนึ่งรหัสทุกครั้งทีก็ยู่กกด และทุกครั้งทีปล่อยคีย์ ถ้ากดคีย์ใดค่างไว้เป็นพินท์จะส่งรหัสเมื่อคีย์ถูกกดจำนวนหนึ่งแล้วหยุดชั่วขณะแล้วส่งรหัสตัวเดิมซ้ำต่อไปอีกตามจังหวะของการพินท์จนกว่าคีย์นั้นจะถูกปล่อย เป็นพินท์จะส่งรหัสรหัสอีกตัวหนึ่งที่ต่างออกกันไป ลักษณะการทำงานแบบนี้ ช่วยให้ระบบทราบถึงการทำงานของเป็นพินท์ สามารถระบุได้ว่าขณะใดที่ใช้ Shift Key อยู่รวมกันทั้งการกดคีย์ค่าง

ในการรับและส่งข้อมูลจากเป็นพินท์จะประกอบด้วยบิตจำนวน 11บิต ซึ่งในการส่งจะทำการส่งข้อมูลแบบอนุกรม เมื่อระบบต้องการส่งข้อมูลไปยังเป็นพินท์ ขาสัญญาณข้อมูลจะอยู่ในสถานะ Negative ส่วนขาสัญญาณนาฬิกาจะอยู่ในสถานะ Positive แลเมื่อเป็นพินท์ต้องการส่งหรือรับข้อมูลจากระบบขาสัญญาณนาฬิกาจะอยู่ในสถานะ Negative ส่วนขาสัญญาณข้อมูลจะอยู่ในสถานะ High หรือ Low ตามลำดับ ดังแสดงสตรึมของข้อมูลได้ในรูปที่ 2.8



รูปที่ 2.8 สตรึมข้อมูลของเป็นพินท์

หน้าที่ของบิตต่างๆ

1 st bit	start bit
2 nd bit	ข้อมูล bit 0(last – significant)
3 rd bit	ข้อมูล bit 1
4 th bit	ข้อมูล bit 2
5 th bit	ข้อมูล bit 3
6 th bit	ข้อมูล bit 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 th bit	ข้อมูล bit 5
8 th bit	ข้อมูล bit 6
9 th bit	ข้อมูล bit 7(most – significant)
10 th bit	ข้อมูล bit 1(odd parity)
11 th bit	Stop bit

2) ขา VCC และ GND

เป็นขาสัญญาณไฟเลี้ยงเป็นพินท์

3) เป็นพินท์ข้อมูลแสดงผล

เมื่อเป็นพินท์พร้อมที่จะทำการส่งข้อมูลระบบ สิ่งแรกที่จะทำการเช็คคือสถานะของ ขา สัญญาณข้อมูล และขาสัญญาณนาฬิกาว่าพร้อมที่จะทำการส่งข้อมูลหรือไม่ ถ้าขาสัญญาณนาฬิกา อยู่ในสถานะ Low ข้อมูลจะถูกเก็บไว้ในเป็นพินท์บัฟเฟอร์ เมื่อต้องการส่งข้อมูลขาสัญญาณ นาฬิกา จะอยู่ในสถานะ High และขาสัญญาณข้อมูลจะอยู่ในสถานะ Low ข้อมูลจะถูกเก็บไว้ใน เป็นพินท์บัฟเฟอร์ จากนั้นเป็นพินท์จะทำการรับข้อมูลจากระบบ

เมื่อระบบพร้อมที่จะส่งข้อมูลไปยังเป็นพินท์ ระบบจะทำการตรวจก่อนว่าถ้าเป็นพินท์ ต้องการส่งข้อมูล ถ้าเป็นพินท์ต้องการส่งระบบก็จะไม่ทำการสร้างนาฬิกา ส่วนขาสัญญาณนาฬิกา ก็ จะอยู่ในสถานะ Negative แต่ถ้าเมื่อใดที่เป็นพินท์ต้องการส่งข้อมูลระบบก็จะเป็นผู้รับข้อมูล แต่ ถ้าเป็นพินท์ไม่ทำการส่งข้อมูลและสตาร์ท ขาสัญญาณจะอยู่ในสถานะ Low ส่วนขาสัญญาณ นาฬิกาจะอยู่ในสถานะ Positive

2.3 หลักการทำงานของวงจรถอด

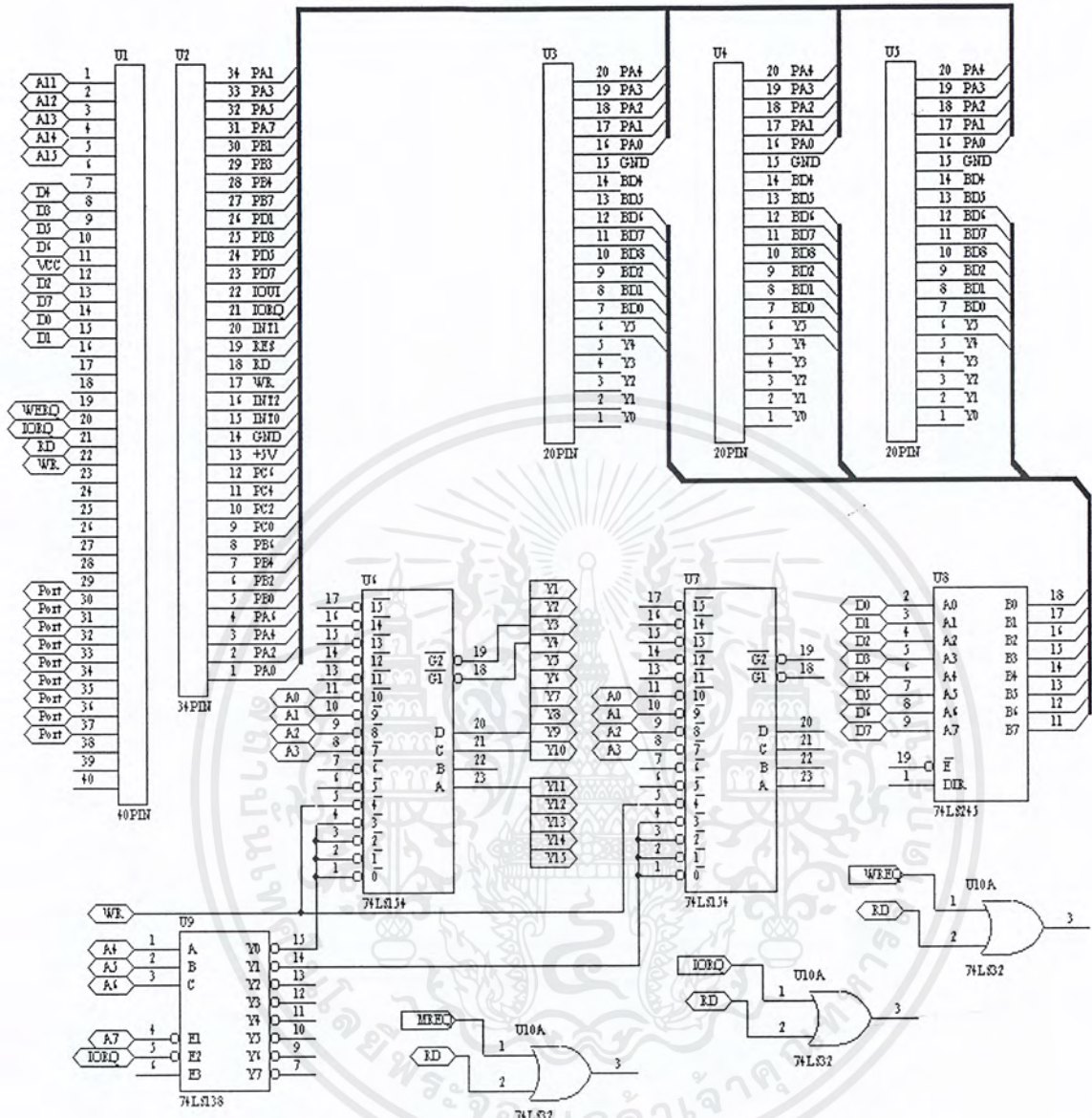
2.3.1 หลักการทำงานของวงจรถอดคือเป็นพินท์

จากหลักการข้างต้นจึงได้ทำการประยุกต์เอาไมโครคอนโทรลเลอร์เบอร์ 8051 มาเป็นตัว ควบคุมในการติดต่อกับเป็นพินท์ของชุดควบคุมหลักโดยให้ไมโครคอนโทรลเลอร์เบอร์ 8051 คอยตรวจสอบว่ามีการกดเป็นพินท์หรือไม่ เป็นแบบธรรมดาหรือ Shift Key และเป็นภาษาไทย หรือภาษาอังกฤษ ซึ่งข้อมูลที่ได้มานี้จะเป็นแบบอนุกรมต้องทำการแปลงเป็นข้อมูล แบบขนาด 8 บิตโดยใช้โปรแกรมที่เขียนด้วยภาษา C แล้วบันทึกลงในหน่วยความจำถาวรของไมโคร คอนโทรลเลอร์เบอร์ 8051 ซึ่งมีข้อมูลในเป็นพินท์เข้ามา ไมโครคอนโทรลเลอร์เบอร์ 8051 ก็จะทำ การประมวลผลและทำการแปลงข้อมูลให้โดยอัตโนมัติ ข้อมูลจะถูกส่งออกไปที่ขา P0.0 – P0.7

ACTIVE LOW ที่ขา INTO จะเกิดการอินเวอร์รต์ ให้ไปทำการอ่านข้อมูลจากขา P1.7 ทุกครั้งจนได้ข้อมูลที่ ได้จากการกด Keyboard ครบและนำไปเปรียบเทียบให้ได้ค่ารหัสที่ต้องการส่งไปที่ขา P 0.0 – P 0.7 ในการทำงานสามารถกำหนดค่าที่ ขา P 0.0 – P 0.7 ให้เป็นค่า 0 ได้โดยการกำหนดขา สัญญาณ PC7 ให้เป็น 1 ซึ่ง ต่ออยู่กับขา P 1.1 และสามารถรีเซ็ตการทำงานของ ไมโครคอนโทรลเลอร์เบอร์ 8051 ได้เมื่อขาสัญญาณ PC 0 ซึ่งต่ออยู่กับขา RESET เป็น 1

2.3.2 หลักการทำงานของวงจรถอดรหัส

จะมีวงจรถอดรหัสดังรูปที่ 2.4 ซึ่งจะรับเอาข้อมูลจากชุดประมวลผลกลาง มาถอดรหัสแล้วส่งไปให้แผงวงจรแสดงผลอีกทั้ง 3 ชุด เพื่อทำการถอดรหัสทางด้านหลักและทางด้านแถวชุดแสดงผลอีกครั้งดังรูปที่ 2.10 โดยวงจรใช้ IC เบอร์ 74LS154 สองตัวเพื่อให้สามารถควบคุม IC ได้ทั้ง 18 ตัวร่วมกับ IC 74LS138 ถอดรหัสที่ได้จากชุดประมวลผลควบคุมการแสดงผลทางด้านหลักโดยผ่านสายสัญญาณแอดเดรส A0 – A7 และสัญญาณ WR โดยการทำงานจะเริ่มดำเนินการทำงานเมื่อมีการใช้คำสั่ง OUT DATA จากชุดประมวลผลเป็นคำสั่งที่ต้องการติดต่อกับอุปกรณ์ภายนอกจะทำให้สัญญาณ WR มีค่าเป็น LOW ทำให้ IC 74LS154 ทำงาน

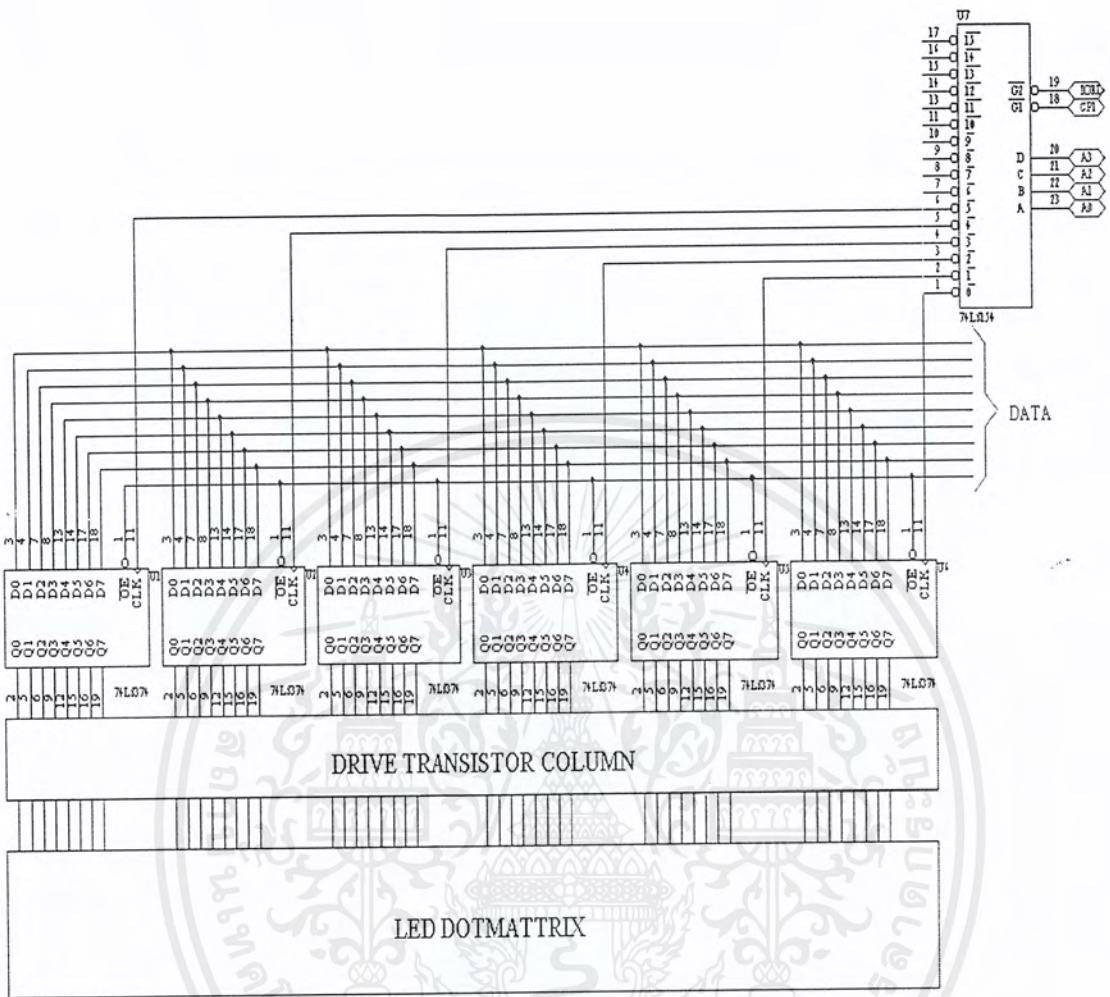


รูปที่ 2.10 วงจรถอดรหัสหลัก

2.3.3 หลักการทำงานของวงจรถอดรหัสทางด้านหลัก

กำหนดให้ A0 – A7 เป็นสัญญาณ แอดเดรส ในการเลือกชิพของไอซี 74LS374 จะถูกถอดรหัสบนบอร์ดอินเตอร์เฟสออกเป็นสัญญาณในการเลือกชิพไอซี 74LS374 จำนวน 18 ตัว เพื่อให้ IC 74LS374 เป็นตัวบัพเฟอร์ข้อมูลที่แสดงผลและใช้ทรานซิสเตอร์เป็นวงจร ขับหลอดไดโอดเปล่งแสง แสงผลดังกล่าวนี้เราจึงสามารถออกแบบได้ดังวงจรที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

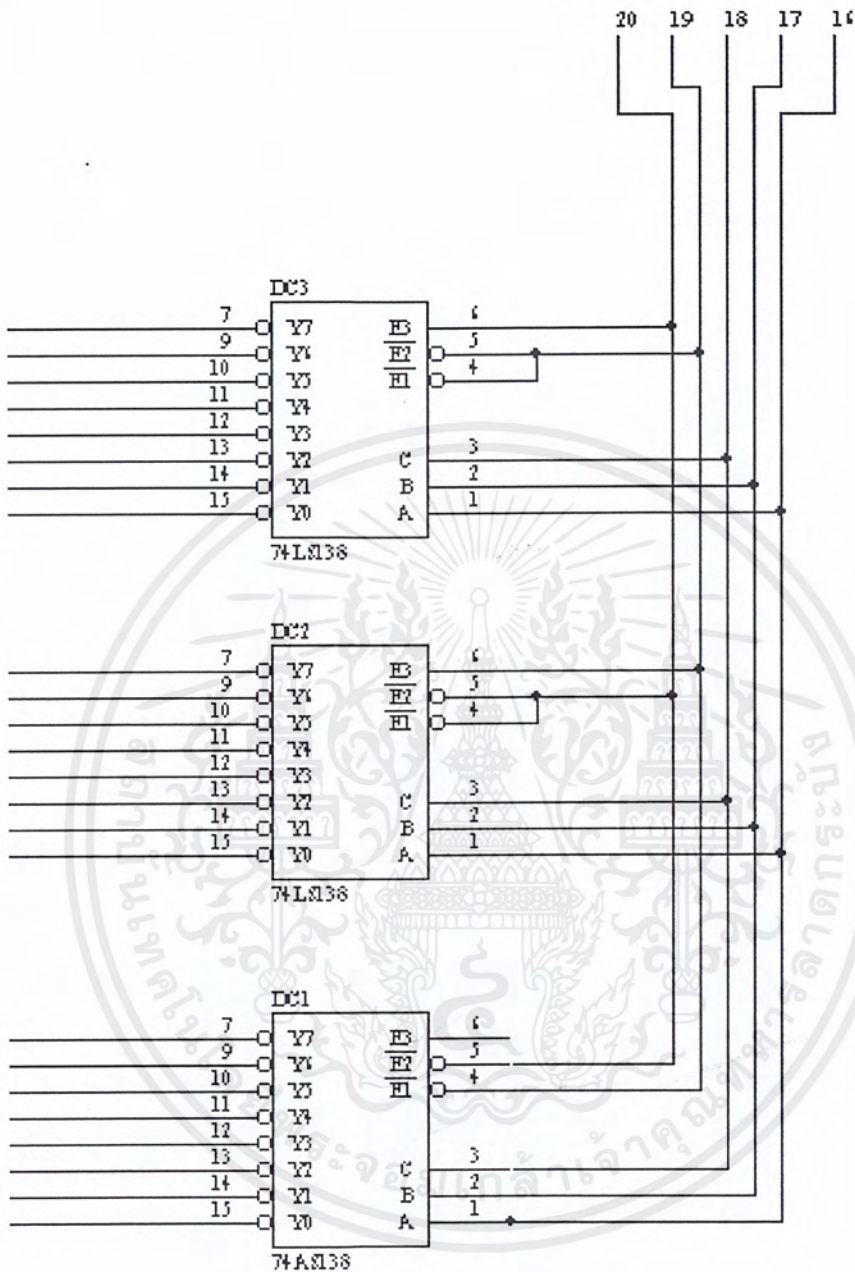


รูปที่ 2.11 วงจรถอดรหัสทางด้านหลัก

2.3.4 หลักการทำงานของวงจรถอดรหัสทางด้านแถว

การออกแบบการสแกนจะต้องใช้เวลาในการสแกนเล็กน้อยที่สุด และสัมพันธ์กับ Duty Cycle ของไดโอดเปล่งแสง เพื่อให้เกิดภาพที่ดีที่สุด ดังนั้นการออกแบบส่วนแสดงผลที่ทำนึ่งมีการสแกน 24 x 144 ตัว ดังนั้นจะเห็นว่าการสแกนทางแนวตั้งจะต้องใช้เวลานานมาก ดังนั้นเพื่อที่จะใช้ชุดสแกนสามารถออกแบบได้ง่าย จึงออกแบบให้สแกนทางแนวนอนโดยใช้ IC 74LS138 ถอดรหัสคือ U1 U2 U3 ใช้ในการสแกนครั้งละ 1 แถว ซึ่งจำนวนแถวทั้งหมดมี 24 แถว โดยเราสมมุติเปรียบเสมือนเป็น Memory MAP ที่หน่วยประมวลผลกลางมองเห็นแล้วสามารถจำลองเป็นหน่วยความจำได้ด้วย ดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



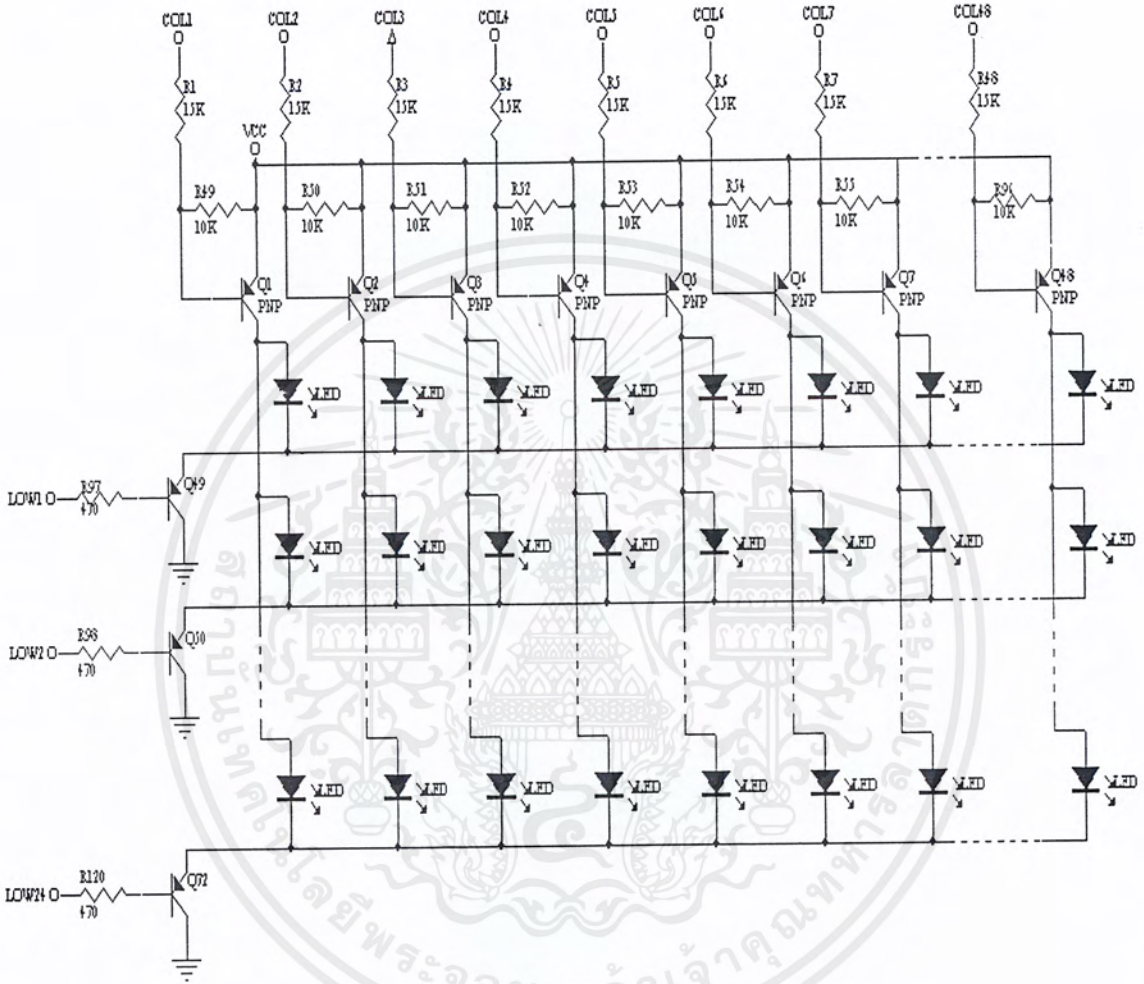
รูปที่ 2.12 วงจรถอดรหัสทางด้านแฉว

2.3.5 หน่วยแสดงผล

ใช้หลอดไดโอดเปล่งแสง ต่อกันในลักษณะเมตริกจำนวน 144 หลัก 24 แฉวการต่อของหลอดไดโอดเปล่งแสง เป็นแบบคอมมอนคาโอด ทางแนวแฉวใช้ทรานซิสเตอร์เป็นสวิทซ์ซึ่งในการแอกทีฟทางด้านแฉว ส่วนในทางด้านหลักใช้ไอซี 74LS374 เป็นตัว LATCH ข้อมูลและมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานซิสเตอร์เป็นบัฟเฟอร์ ซึ่งใช้หลักการสแกนตามหลักการที่ทำให้เกิดภาพ ซึ่งมีหลักการทำงานจริงตามรูปที่ 2.13



รูปที่ 2.13 วงจรส่วนแสดงผล

2.3.6 ส่วนป้อนข้อมูล

ใช้คีย์บอร์ดแบบมาตรฐานของเครื่องคอมพิวเตอร์พีซี AT เป็นคีย์บอร์ดแบบ 101 คีย์ ลักษณะการใช้งานเหมือนของเครื่องคอมพิวเตอร์ทั่วไปสามารถป้อนข้อมูลได้ทั้งภาษาไทยและภาษาอังกฤษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 หลักในการทำให้เกิดภาพในแผงแสดงผล

การทำงานให้เกิดตัวอักษรแสดงผลสามารถทำได้ 2 ลักษณะใหญ่ๆ คือ

1. Latch
2. Scan

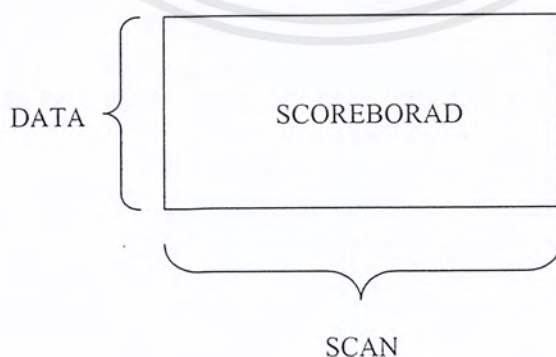
2.4.1 Latch

วิธีนี้ในการเขียนโปรแกรมแสดงผลจะง่าย และแผงแสดงผลจะมีความสว่าง แต่จะมีข้อเสีย คือ ต้นทุนในการผลิตมีราคาสูงมากเพราะ 1 ดวงของไดโอดเปล่งแสง จะเท่ากับชุด DRIVE 1 ชุด ดังนั้นถ้าแผงแสดงผล 8×24 นี้ จะหมายถึงถ้าใช้พอร์ต IC เบอร์ 74LS374 จะต้องใช้ IC 74LS374 ถึง 24 ตัวจึงจะได้ตัวอักษรขึ้นมา 1 ตัว การเขียนโปรแกรมเขียนเหมือนไฟวิ่งแบบ แถว เดียว เช่น ตัวอักษร A 1 ตัว ประกอบไปด้วยข้อมูล 01H,02H,03H,04H,.....14H เมื่อจะทำให้เกิด ตัวอักษรก็ส่ง 01H ไปที่ 74LS374 ตัวที่ 1 ต่อไปก็นำข้อมูล 02H ส่งไปที่ 74LS374 ตัวที่ 2 และส่ง เรื่อยไปจนถึงตัวที่ 24 ก็ให้เห็นเป็นตัวอักษรขึ้นมา 1 ตัว และตัวอักษรก็จะติดอยู่อย่างนั้นจนกว่าจะ ส่งข้อมูลไป Clear 74LS374 แล้วข้อมูลก็จะถูกค้างไว้มันเอง

2.4.2 Scan

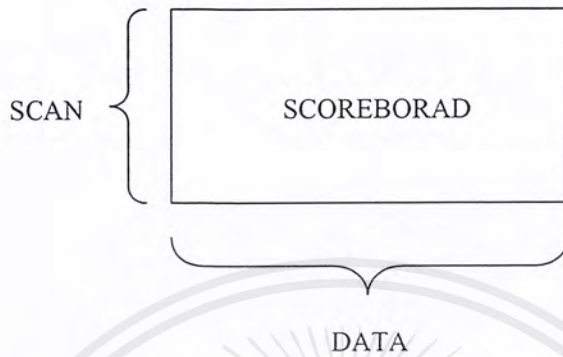
วิธีนี้ใช้การเขียนแสดงผลจะมีความยุ่งยากมาก แต่ข้อดีในการประหยัดทางด้าน ฮาร์ดแวร์ (Hard Ware) ได้เป็นจำนวนมาก โดยวิธีการ สแกน (Scan) ยังแบ่งได้เป็น 2 ลักษณะ เนื่องจากการ สแกน ควบคุมการติดดับของหลอดไฟ จะต้องกระทำใน 2 ทิศทาง คือ แนวทางแถวและแนวทาง หลัก ซึ่งเป็นลักษณะของMatrix ดังนั้นเราสามารถกำหนดการ สแกน ได้ 2 รูปแบบ คือ

- 1) แบบสแกนตามแนวตั้ง (Vertical) หรือหลัก ดังรูปที่ 2.14



รูปที่ 2.14 การสแกนทางแนวตั้ง

2) แบบสแกนทางแนวนอน (Horizontal) หรือแถว ดังรูปที่ 2.15



รูปที่ 2.15 การสแกนทางแนวนอน

เช่นเราใช้วิธีการ สแกน ทางแถว โดยเริ่มต้นเราส่งข้อมูลตัวที่ 1 ออกไปจากรูปส่ง 18H จากนั้นก็ Active แถวที่ 1 ถัดไปก็ส่งข้อมูลตัวที่ 2 ทำลักษณะเช่นนี้เรื่อยไปจนถึงแถวที่ 8 ก็จะได้ตัวอักษร A มาช่วงขณะหนึ่ง แต่ความเป็นจริงแล้ว ถ้าเราพิจารณาการทำงานการติดของไดโอดเปล่งแสง จะติดได้ครั้งละแถวเท่านั้น เมื่อแถวไหนทำงานแถวอื่นที่นอกเหนือจากนั้นก็จะไม่ทำงาน แต่ที่เราเห็นเป็นตัวอักษรก็เพราะดวงตาของเรา ไม่สามารถจับการเปลี่ยนแปลงของแต่ละแถวได้เร็ว เราก็จะมองเห็นการติดของแต่ละแถวเป็นภาพรวม ดังตัวอย่าง เช่นหลอดไฟที่ใช้ในบ้านเรามีอัตราการติดดับ 50 ครั้งใน 1 วินาที แต่ดวงตาของเราจะเห็นเป็นหลอดไฟติดตลอด หรืออย่างเช่นพัดลม จะเห็นว่าใบพัดแยกเป็น 3 ใบ ที่มีระยะห่างกันพอสมควร แต่เมื่อใบพัดหมุนเราจะมองไม่เห็นช่องว่างของระยะห่างกันนั้น เช่นกันกับการติดของตัวอักษรในแบบ สแกน แต่เมื่อตัวอักษรเกิดขึ้นในลักษณะนั้นสายตาของเรายังอาจจะจับภาพไม่ได้เพราะการเกิดเร็วมาก เราจำเป็นต้องทำการส่งข้อมูลด้วยขบวนการเดิมอีกสักประมาณ 7-8 ครั้ง ก็จะทำให้สายตาของเราเห็นเป็นตัวอักษรได้

ในส่วนการ สแกน ให้เกิดตัวอักษรไม่ว่าจะเป็นการ สแกน ทางแถว หรือ สแกน ทางหลัก จะไม่มีผลกระทบต่อส่วนอื่นมากนัก แต่เมื่อให้ตัวอักษรนั้นมีการเคลื่อนที่ได้ การ สแกน ทางด้านใดๆนั้นจะมีผลกระทบกับการเขียนโปรแกรม เช่น การเคลื่อนที่ของตัวอักษรจากขวาไปซ้าย หรือ ซ้ายไปขวา ถ้าเราทำโปรแกรมหรือ ฮาร์ดแวร์ ให้ สแกนทางหลัก จะทำให้เขียนโปรแกรมได้ง่าย แต่การ สแกนทางหลัก ก็จะมีผลเสียทางด้านความสว่างของการแสดงผล ถ้าจำนวนของหลักมีจำนวนมากขึ้น นั่นก็คือความสว่างของการแสดงผลขึ้นอยู่กับความยาวของ จอแสดงผล แต่ถ้าใช้การ สแกน ทางด้านแถว คือความสว่างของการแสดงผลจะไม่กระทบกับความยาวของแผงแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากขึ้น ทำให้ความสว่างของการแสดงผลดีกว่าการ สแกน ทางหลัก เพราะในแนวแถว จะถูก กำหนดคงที่ไม่มีเปลี่ยนแปลง เช่น แถวเท่ากับ 24 แต่เมื่อการขยายตัวอักษรคือ เพิ่มหลักมากขึ้นแต่ สแกน ให้เกิดภาพอย่างไร ก็ยังเป็น 24 นั้นเอง

2.5 การเคลื่อนที่ของตัวอักษร

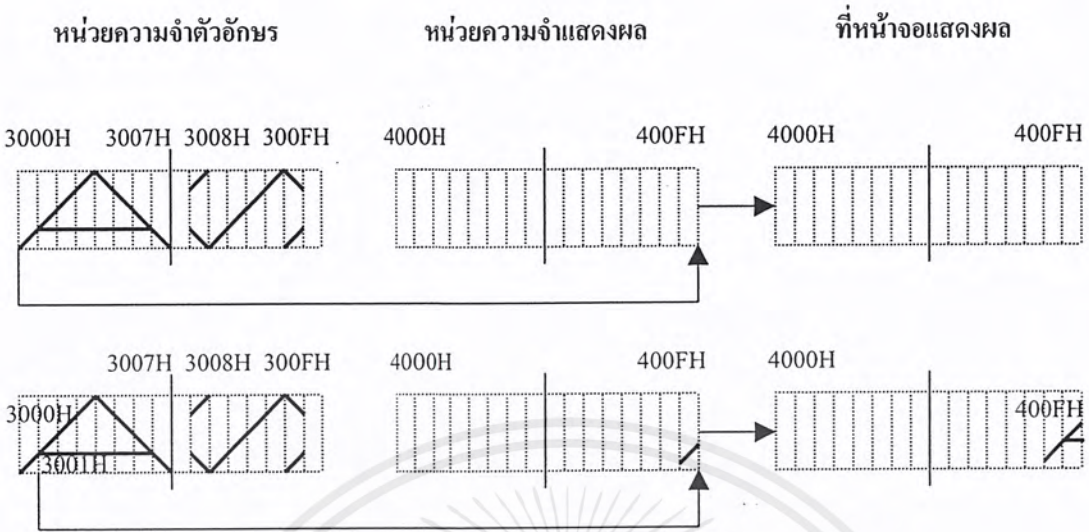
พิจารณาการ สแกน ทางหลักกลับมาที่รูปตัว A เมื่อคิดข้อมูลของตัวอักษรแต่ละ Byte ก็มอง ทางหลัก ซึ่งจะมีข้อมูลที่ได้ดังนี้

เมื่อเราจะทำให้เกิดภาพ เราก็จะนำข้อมูลจากหลัก 0 ถึง 7 โดยได้ Active ไปทีละหลักเมื่อ ต้องการเลื่อนตัวอักษร เราก็เพิ่มตำแหน่งที่ชี้ข้อมูลตัวอักษรขึ้นไป 1 ตำแหน่ง แล้วก็กลับมาทำ โปรแกรมการ สแกน ใหม่ซึ่งในขณะนี้ตำแหน่ง 3001H จะเป็นตำแหน่งข้อมูลเริ่มต้นของการ สแกน เมื่อทำโปรแกรม สแกน ก็จะทำให้เห็นว่าข้อมูลตัว A ถูกเลื่อนไปทางซ้ายโดยขอบของตัว A จะมาชิดจอและขอบขวามือจะมีข้อมูลตัวใหม่เข้ามาแทนที่

ในกรณีจะให้เคลื่อนที่ไปทางขวาบ้าง ก็จะกำหนดตำแหน่งที่ชี้ตัวอักษรตัวสุดท้าย และ ตำแหน่งข้อมูล Byte สุดท้ายของตัวอักษรตัวนั้นแต่เวลา สแกน ก็ให้เริ่มที่หลักที่ 7 ก่อนเมื่อจะเลื่อน ก็ให้ลดตำแหน่งตัวชี้ข้อมูลตัวอักษรลง 1 ตำแหน่ง ถ้าสังเกตจะเห็นว่าการกระทำทุกอย่างกลับกับ แบบแรกทั้งหมด และเมื่อมองที่จุดนี้จะเห็นว่าเมื่อเปลี่ยนรูปแบบการวิ่ง โปรแกรมที่ใช้ สแกน ตัว อักษรก็จะถูกเปลี่ยนไปด้วย ซึ่งเมื่อเราทำไฟวิ่งให้มีรูปแบบหลายๆอย่างก็จะทำให้มีโปรแกรม สแกน แยกเป็นหลายชุด กลับจะทำให้โปรแกรมที่เขียนขึ้นมีความซับซ้อนมากขึ้น และทำให้การ คิดตัวอักษรที่ใช้ในการแสดงผลเป็นไปด้วยความลำบากอีกด้วย

2.5.1 การเลื่อนซ้าย

เริ่มด้วยการนำข้อมูลตัวอักษรตำแหน่งแรกไปไว้ที่หน่วยความจำแสดงผลตำแหน่งสุดท้าย แล้วดันข้อมูลในหน่วยความจำแสดงผลไปทางซ้ายโดยสามารถแสดง ได้ดังรูปที่ 2.16

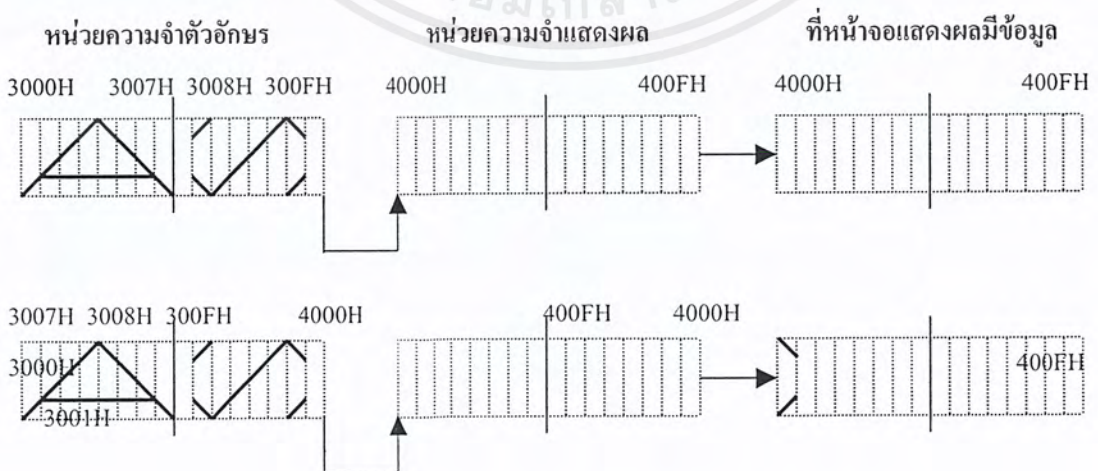


รูปที่ 2.16 การค้นข้อมูลในหน่วยความจำแสดงผลไปทางซ้าย

แล้วเลื่อนตำแหน่งชี้ข้อมูลถัดไป แล้วทำขบวนการอย่างเดิมจนกว่าจะหมดข้อมูล จากนั้นก็จำเป็นต้องนำข้อมูล 0 เป็นจำนวนเท่ากับหน้าจอแสดงผล ดันเข้าไปที่หน่วยความจำแสดงผล เพื่อให้ข้อมูลที่แสดงได้เต็มหน้าจอแล้ว ถูกดันหายไปทางซ้าย

2.5.2 การเลื่อนขวา

เช่นเดียวกับการเลื่อนซ้ายต่างกันตรงที่ นำข้อมูลตัวอักษรเข้าไปที่หน่วยแสดงผล จะเริ่มนำเข้าไปที่หน่วยความจำแสดงผล ตำแหน่งแรกสุดแล้วดันไปทางขวา ส่วนข้อมูลตัวอักษรที่นำเข้าไปยังหน่วยแสดงผลจะเป็นตัวอักษรตัวสุดท้ายดังรูปที่ 2.17



รูปที่ 2.17 การค้นข้อมูลในหน่วยแสดงผลไปทางซ้าย

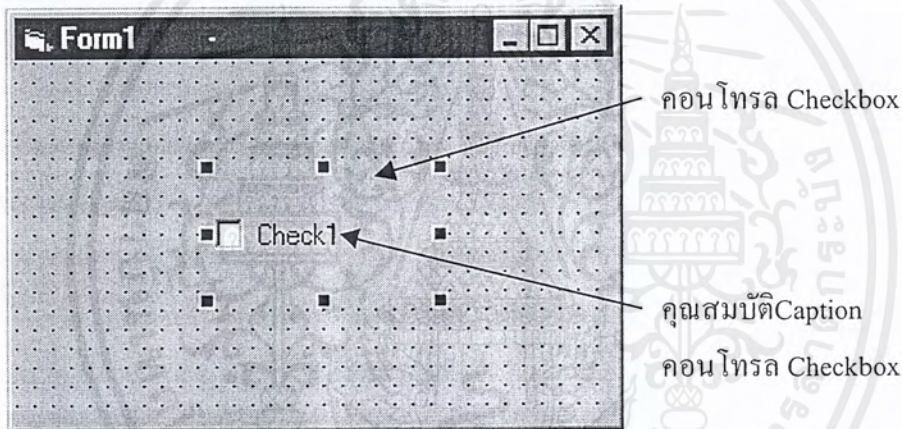
2.6 คอนโทรลภายในของ Visual Basic

คอนโทรลภายในจะประกอบด้วยคอนโทรลต่างๆดังต่อไปนี้

2.6.1 คอนโทรลภายในทั่วไป

1) คอนโทรล CheckBox

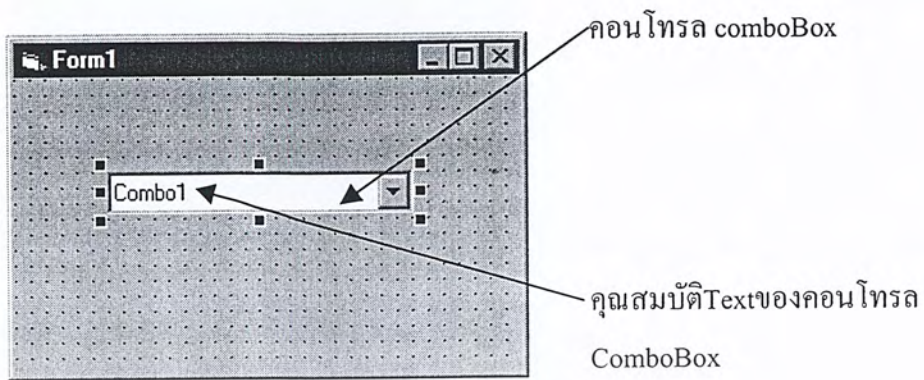
คอนโทรล CheckBox เป็นคอนโทรลที่จะถูกนำไปใช้ในลักษณะของการกำหนดหรือแสดงผล สถานะเป็นจริง (True) หรือเท็จ (False) เหมือนกับคอนโทรล OptionButton แต่ข้อแตกต่างในด้านของพฤติกรรมจากคอนโทรล OptionButton ตรงที่คอนโทรล CheckBox ไม่มีลักษณะการรวมกลุ่ม (grouping) ดังนั้นคอนโทรล CheckBox แต่ละตัวจึงสามารถถูกกำหนดหรือแสดงผลสถานะเท่ากับ True ได้หลายๆตัวพร้อมๆกัน



รูปที่ 2.18 คอนโทรล CheckBox ในขณะออกแบบ

2) คอนโทรล ComboBox

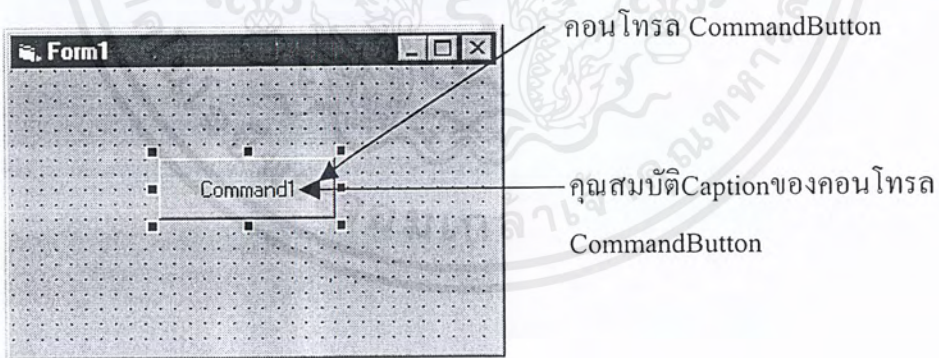
คอนโทรล ComboBox เป็นคอนโทรลที่รวมเอาความสามารถของคอนโทรล ListBox และ TextBox เข้าไว้ด้วยกัน ซึ่งสามารถใช้คอนโทรลนี้ในการแสดงผลรายการที่ใช้พื้นที่แสดงผลน้อยกว่าคอนโทรล ListBox หรือให้ผู้ใช้สามารถเพิ่มรายการใหม่เข้ามาในคอนโทรลได้อีกด้วย นอกจากนี้ยังสามารถกำหนดค่าให้กับคุณสมบัติ Style ของ ComboBox เพื่อให้คอนโทรล ComboBox มีความสามารถตามที่กำหนดเอาไว้ ในขณะออกแบบคอนโทรล ComboBox ที่วางลงบนฟอร์มจะมีลักษณะดังรูปที่ 2.19



รูปที่ 2.19 คอนโทรล ComboBox ในขณะออกแบบ

3) คอนโทรล CommandButton

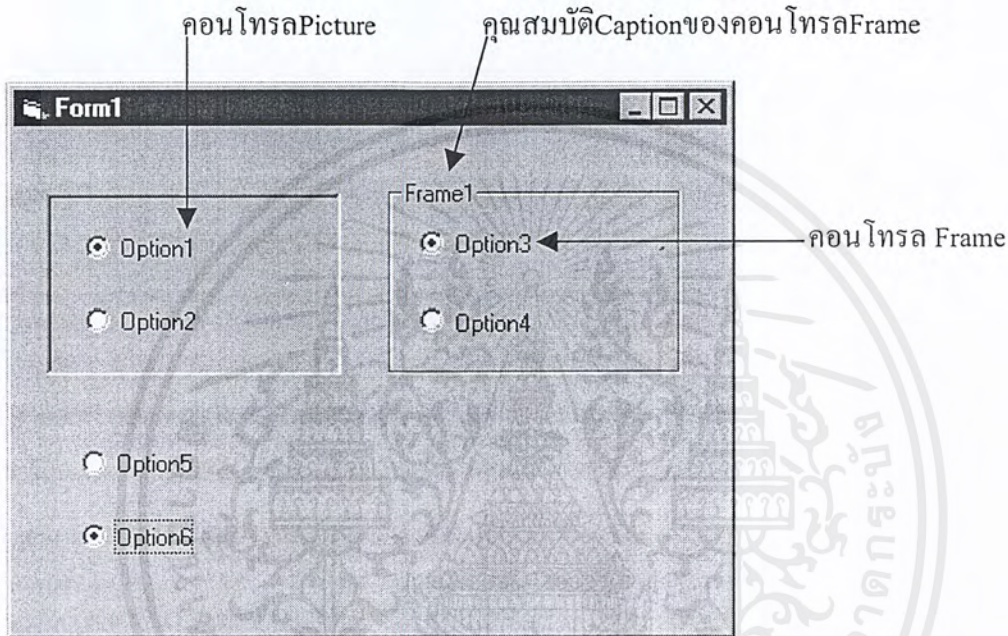
คอนโทรล CommandButton จะเป็นคอนโทรลที่จะถูกนำไปใช้งานมากที่สุด เพราะในการกำหนดให้ผู้ใช้งานเลือกที่จะตอบ OK หรือ Cancel นั้นเรามักจะใช้คอนโทรล CommandButton เป็นส่วนใหญ่ ดังนั้นถือได้ว่าเป็นคอนโทรลที่พื้นฐานที่สุดของ Visual Basic เนื่องจากคอนโทรลนี้เป็นปุ่มคำสั่งที่ใช้งานในรูปแบบของการคลิกเพื่อยืนยัน ในขณะที่การออกแบบคอนโทรล CommandButton ที่วางลงบนฟอร์มจะมีลักษณะดังตัวอย่างในรูปที่ 2.20



รูปที่ 2.20 คอนโทรล CommandButton ในขณะออกแบบ

4) คอนโทรล Frame

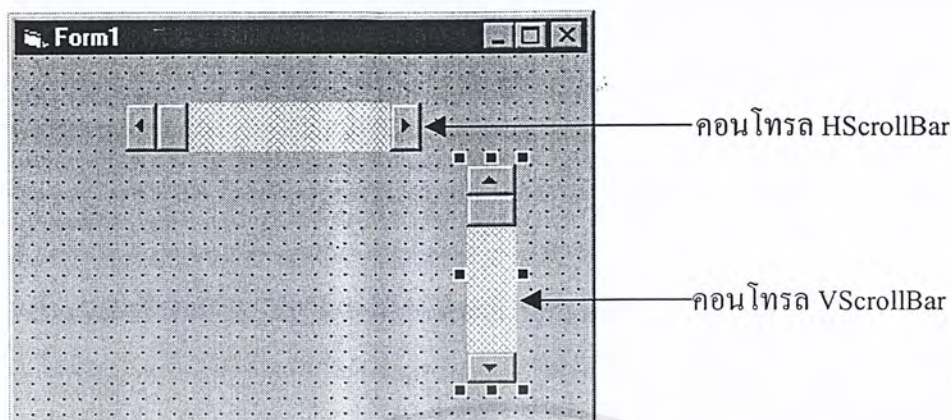
คอนโทรล Frame เป็นคอนโทรลประเภทตัวบรรจุที่สามารถใช้ในการจัดกลุ่ม (grouping) คอนโทรลชนิดปุ่มเลือก เช่น คอนโทรล OptionButton เป็นต้น และนอกจากนี้ตัวบรรจุโดยทั่วไปยังสามารถจำกัดของเขตการแสดงผลและการวางตำแหน่งของคอนโทรลอื่นๆ ภายในตัวบรรจุได้อีกด้วย



รูปที่ 2.21 ผลของการจัดกลุ่มด้วยตัวบรรจุ Frame ที่มีต่อคอนโทรล OptionButton

5) คอนโทรล HScrollBar และ VScrollBar

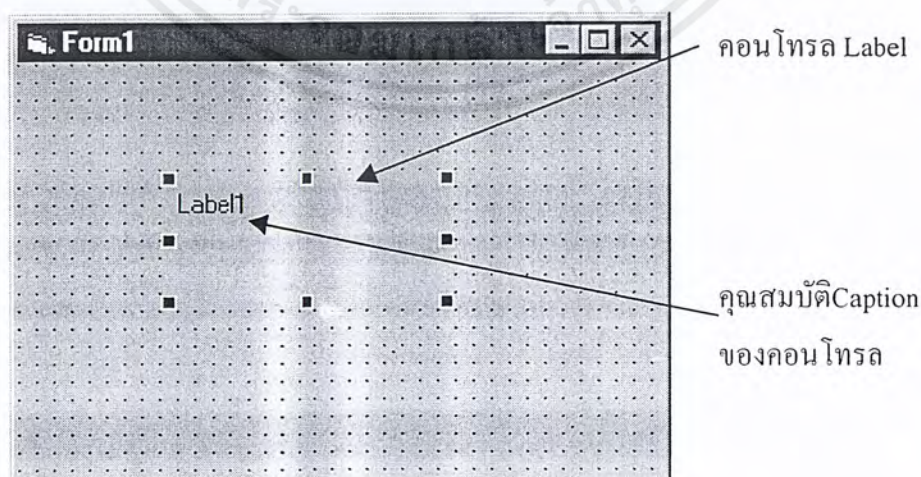
คอนโทรล HScrollBar และ VScrollBar โดยทั่วไปจะเป็นคอนโทรลที่จะถูกนำไปใช้งานร่วมกับคอนโทรลอื่นๆ เพื่อช่วยในการกำหนดช่วงของการเลื่อนตำแหน่งส่วนที่บรรจุภายในคอนโทรลอื่นๆ ในแนวนตั้งและแนวนอนมาลำดับ ซึ่งบางคอนโทรล เช่น ListBox หรือ FileListBox ก็จะมีคุณสมบัติของคอนโทรล scroll bar เหล่านี้เพื่อช่วยในการเลื่อนรายการอีกต่อไป แต่สำหรับบางคอนโทรลหรือในการทำงานบางประเภทเราอาจจะต้องการเฉพาะส่วนที่สามารถบอกถึงลำดับของการเลื่อนรายการหรือกลุ่มคอนโทรลได้ เช่น การสร้างแอปพลิเคชันการแสดงผลบางพื้นที่เฉพาะบางส่วนของภาพกราฟิกที่ต้องการแสดงผลก็ได้



รูปที่ 2.22 คอนโทรล HScrollBar และ VScrollBar ที่ถูกวางลงในฟอร์มในขณะออกแบบ

6) คอนโทรล Label

คอนโทรล Label เป็นคอนโทรลในลักษณะของกราฟิกที่ถูกใช้งานด้านการแสดงผลข้อความบนฟอร์ม เหมือนกับผู้อ่านได้นำป้ายข้อความอย่างหนึ่งวางไว้บนฟอร์ม เพื่อใช้ในการสื่อข้อความกับผู้ใช้และคอนโทรลนี้ผู้ใช้ไม่สามารถแก้ไขได้โดยตรงด้วยวิธีการคีย์หรือการใช้เมาส์ในขณะรันแอปพลิเคชัน นอกจากการใช้แอปพลิเคชันจะมีการเขียนโค้ดสำหรับแก้ไขข้อความในคอนโทรลโดยการแก้ไขค่าของคุณสมบัติ Caption เท่านั้น และนอกจากนี้ Label ยังเป็นคอนโทรลที่มีความสามารถด้าน CCE (Dynamic Data) อีกด้วย ในขณะการออกแบบสามารถเพิ่มคอนโทรลลงในฟอร์มหรือตัวบรรจ้อื่นๆ ก็จะปรากฏหน้าต่างของคอนโทรลดังในรูปที่ 2.23

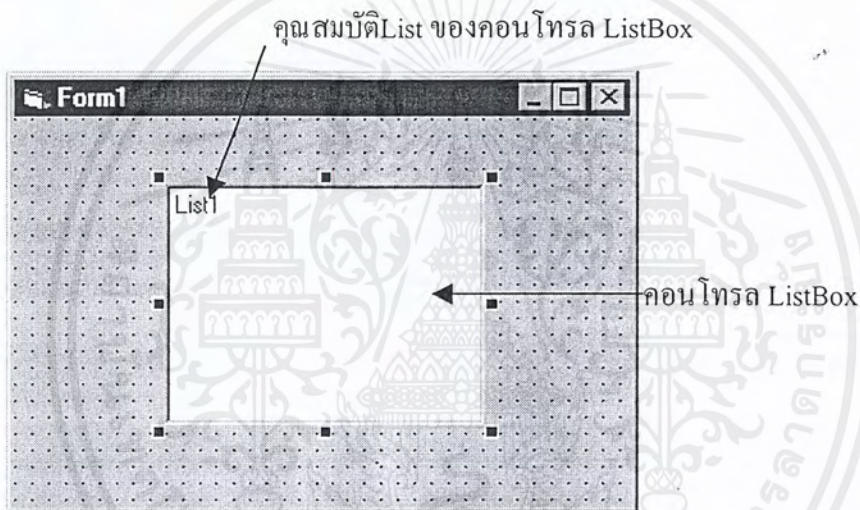


รูปที่ 2.23 คอนโทรล Label ที่ถูกวางลงในฟอร์มในขณะออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) คอนโทรล ListBox

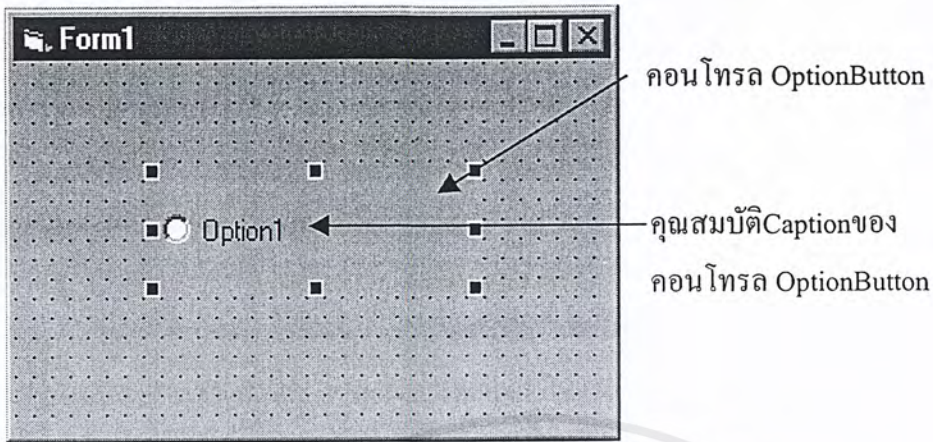
คอนโทรล ListBox เป็นคอนโทรลที่มีลักษณะการใช้งานที่แตกต่างไปจากคอนโทรล CheckBox และ OptionButton โดยที่ลักษณะการใช้งานจะเป็นในรูปแบบการแสดงข้อความหลายๆบรรทัดในคอนโทรล ListBox และถ้าหากจำนวนรายการที่มีในคอนโทรล ListBox มีจำนวนมากจนไม่สามารถแสดงผลได้ภายในหน้าต่างเดียว Visual Basic ก็จะเพิ่มแถบเลื่อน (scroll bar) ให้กับคอนโทรลโดยอัตโนมัติ โดยแถบเลื่อนนี้สามารถเพิ่มได้ทั้งแนวดิ่งและแนวนอน ทั้งนี้ก็ขึ้นอยู่กับว่ามีการกำหนดวิธีการแสดงผลรายการของ ListBox อย่างไร โดยในขณะที่ออกแบบคอนโทรล ListBox ที่วางลงในฟอร์ม ก็จะมีลักษณะดังในรูปที่ 2.24



รูปที่ 2.24 คอนโทรล ListBox ในขณะออกแบบ

8) คอนโทรล OptionButton

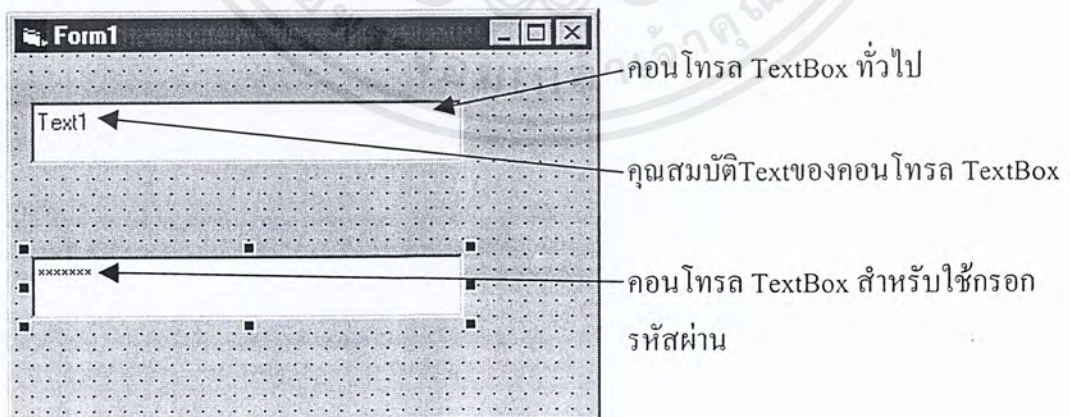
คอนโทรล OptionButton เป็นคอนโทรลที่ถูกออกแบบมาเพื่อนำมาใช้ในลักษณะของการเลือกอย่างใดอย่างหนึ่ง และคอนโทรลนี้ยังสามารถนำมาใช้โดยแยกออกเป็นกลุ่มๆ ได้ โดยอาศัยคอนโทรลชนิดตัวบรรจุ เช่น Form, PictureBox หรือ Frame เป็นต้น คอนโทรล OptionButton มีความแตกต่างกับคอนโทรล CheckBox คือการเลือกคอนโทรล OptionButton ภายในกลุ่มเดียวกันคือ สามารถเลือกได้ตัวเดียว แต่สำหรับคอนโทรล CheckBox สามารถเลือกพร้อมๆ กันได้คราวละหลายๆตัว เพราะคอนโทรล CheckBox แต่ละตัวได้ทำหน้าที่เป็นตัวเลือกภายในตัวเองอยู่แล้ว ในขณะที่ออกแบบคอนโทรล OptionButton ที่วางลงบนฟอร์มหรือตัวบรรจุใดๆ ก็จะมีลักษณะดังตัวอย่างในรูปที่ 2.25



รูปที่ 2.25 คอนโทรล OptionButton ในขณะออกแบบ

9) คอนโทรล Textbox

คอนโทรล Textbox มักจะถูกนำไปใช้ในหลายๆ ฟอรั่มที่มีการรับกรอกข้อความจากผู้ใช้ เนื่องจากคอนโทรลนี้ทำหน้าที่แสดงข้อมูล (โดยผ่านทางคุณสมบัติ Text) ในคอนโทรลและยังอนุญาตให้ผู้ใช้สามารถแก้ไขตัวอักษรต่างๆของคุณสมบัตินี้ได้ด้วยเช่นเดียวกันนอกจากนี้แล้วคอนโทรล TextBox ยังได้รวมเอาความสามารถหลายๆ ด้านของคอนโทรล Label มาใช้ เช่นสามารถแสดงข้อมูลได้มากกว่า 1บรรทัด ความสามารถด้าน DDE(Dynamic Data Exchange) และนอกจากนี้ยังสามารถถูกนำไปใช้ในลักษณะของการกรอกรหัสผ่าน(password)

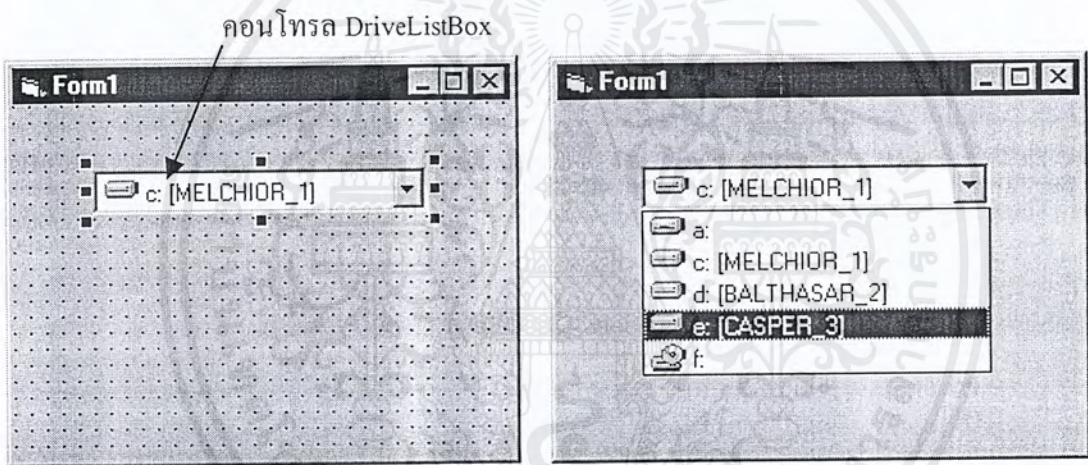


รูปที่ 2.26 คอนโทรล TextBox ในขณะออกแบบ

2.6.2 คอนโทรลภายในด้านระบบไฟล์

1) คอนโทรล DriveListBox

คอนโทรล DriveListBox เป็นคอนโทรลที่ช่วยงานด้านการแสดงผลไดรฟ์ที่ติดตั้งจริงๆ ของระบบ ซึ่งสามารถเป็นไปได้ทั้งไดรฟ์กายภาพ (physical drive) ไดรฟ์ตรรกะ (logical drive) ซึ่งเป็นไดรฟ์ที่สร้างขึ้นภายในหน่วยความจำ โดยอาศัยไดรเวอร์ในการจำลองหน่วยความจำให้เป็นดิสก์ไดรฟ์จริงๆ หรือไดรฟ์ในระบบเครือข่าย ในขณะที่ออกแบบคอนโทรล DriveListBox ที่วางลงบนฟอร์มจะแสดงผลเฉพาะไดรฟ์ที่ปัจจุบันดังในรูปที่ 2.27 ทางซ้ายและในขณะที่รันแอปพลิเคชันก็สามารถคลิกปุ่ม เพื่อแสดงรายการดิสก์ไดรฟ์ของระบบในรูปแบบ drop list ซึ่งสามารถเลือกไดรฟ์ที่ต้องการได้ ดังในรูปที่ 2.27 ทางขวาจะแสดงคอนโทรลในการทำงาน



รูปที่ 2.27 คอนโทรล DriveListBox ในขณะออกแบบและทำงาน

2) คอนโทรล Dirlistbox

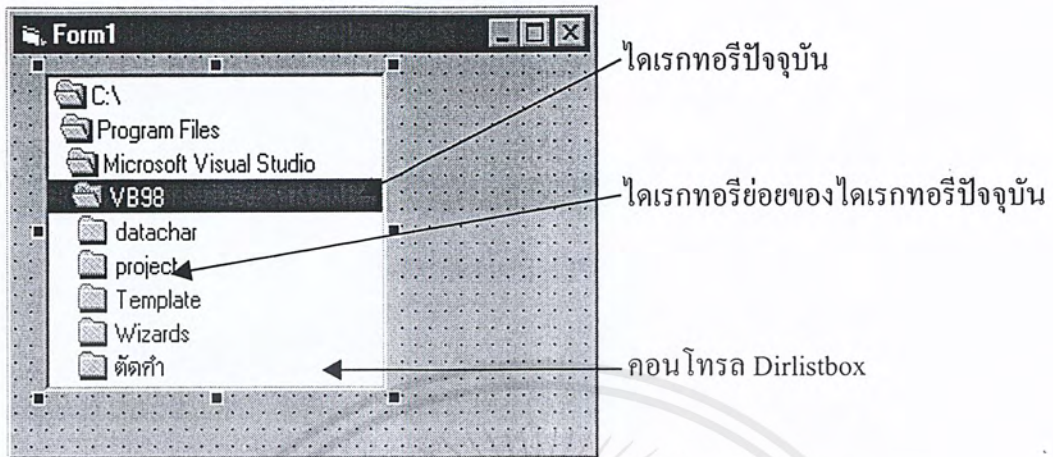
คอนโทรล Dirlistbox เป็นคอนโทรลที่แสดงผลไดเรกทอรีของไดรฟ์ที่ปัจจุบัน โดยในการแสดงผลไดเรกทอรีภายในคอนโทรลนี้ Visual Basic จะแสดงผล ไดเรกทอรีด้วยไอคอนด้านหน้าชื่อของไดเรกทอรีดังนี้

สำหรับไดเรกทอรีย่อยของไดเรกทอรีปัจจุบันที่ถูกเปิด

สำหรับไดเรกทอรีที่ยังไม่ถูกเปิดหรือไม่ใช่ไดเรกทอรีปัจจุบัน

สำหรับไดเรกทอรีย่อยเป็นไดเรกทอรีลูกก็จะแสดงผลในลักษณะย่อหน้าเข้าไปดังรูปที่ 2.28 ซึ่งในขณะที่ออกแบบคอนโทรลที่ถูกวางลงบนฟอร์ม คอนโทรลก็จะแสดงผลไดเรกทอรีเฉพาะไดรฟ์ปัจจุบันดังในรูปที่ 2.28

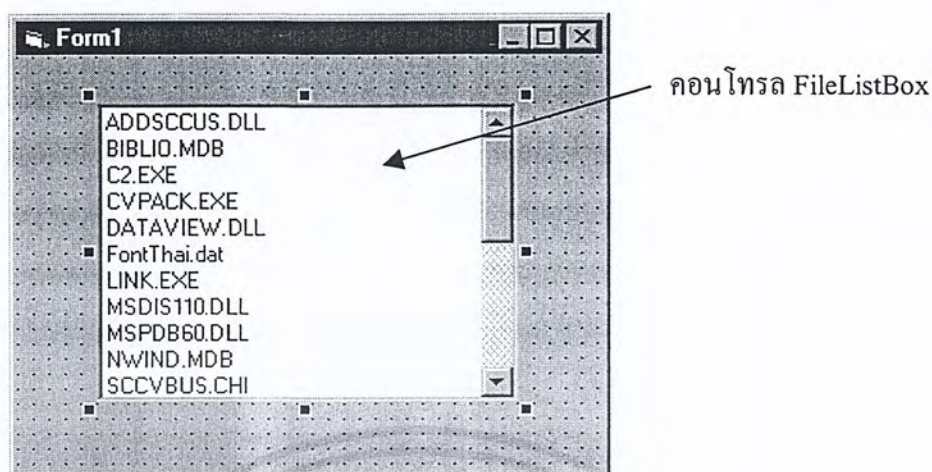
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 คอนโทรล DirListBox ในขณะออกแบบ

3) คอนโทรล FileListBox

คอนโทรล FileListBox เป็นคอนโทรลที่ช่วยในการแสดงผลไฟล์ต่างๆ ในไดเรกทอรีปัจจุบัน ซึ่งเป็นไดเรกทอรีที่ถูกกำหนดให้กับคุณสมบัติ Path โดยที่ผู้อ่านสามารถกำหนดเป็นไดเรกทอรีใดๆ ก็ได้ในขณะรันแอปพลิเคชัน โดยการกำหนดไดเรกทอรีนั้นๆ ให้กับคุณสมบัติ Path แต่เนื่องจากผู้อ่านสามารถกำหนดแอตทริบิวต์ได้หลายค่า ดังนั้นคอนโทรลตัวนี้จึงอนุญาตให้กำหนดชนิดของแอตทริบิวต์ของไฟล์ที่ต้องการแสดงผลได้ด้วย และคอนโทรลตัวนี้ก็มีคุณสมบัติหลายๆ ประการที่คล้ายกับคอนโทรล ListBox ได้แก่การอนุญาตให้สามารถเลือกไฟล์ได้คราวละหลายๆ ไฟล์โดยการกำหนดค่าให้กับคุณสมบัติ MultiSelect เป็นต้น ซึ่งในขณะออกแบบคอนโทรลที่ถูกวางลงบนฟอร์มก็จะแสดงผลไฟล์ในไดเรกทอรีปัจจุบันของไดรฟ์ปัจจุบันดังในรูปที่ 2.29

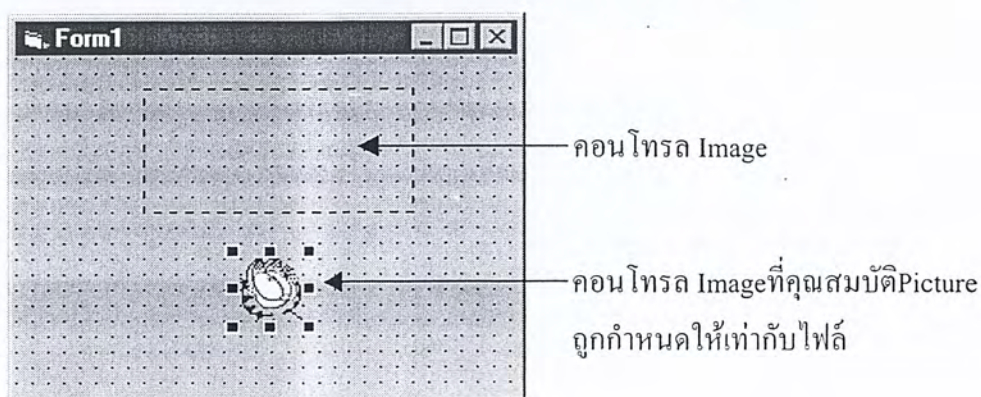


รูปที่ 2.29 คอนโทรล FileListBox ในขณะออกแบบ

2.6.3 คอนโทรลภายในด้านกราฟิก

1) คอนโทรล Image

คอนโทรล Image เป็นคอนโทรลที่ช่วยในการแสดงผลกราฟิกหรือไฟล์กราฟิกต่างๆ เช่นเดียวกับคอนโทรล PictureBox แต่คอนโทรล Image จะสามารถทำงานหรือวาดกราฟิกภายในคอนโทรลได้อย่างรวดเร็วกว่า และที่สำคัญมีการใช้ทรัพยากรน้อยกว่าระบบของคอนโทรล PictureBox อีกด้วย แต่เนื่องจากคอนโทรล Image ใช้ทรัพยากรค่อนข้างน้อย ดังนั้นจึงสนับสนุนคุณสมบัติ เหตุการณ์และวิธีคอนโทรล PictureBox เช่นกัน ดังนั้นในการเลือกใช้คอนโทรลทั้งสองต้องคำนึงถึงความต้องการที่จะให้คอนโทรลตอบสนองต่อผู้ใช้ และพิจารณาถึงความสามารถของคอนโทรลเหมาะสมกับงานหรือไม่ ทั้งให้แอปพลิเคชันสามารถทำงานได้รวดเร็วยิ่งขึ้น สำหรับในขณะออกแบบคอนโทรลที่ถูกวางลงบนฟอร์ม ก็จะมีลักษณะดังตัวอย่างในรูปที่ 2.30

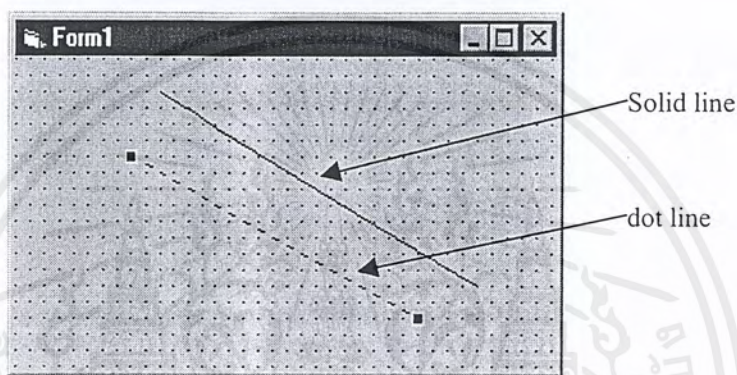


รูปที่ 2.30 คอนโทรล Image ในขณะออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) คอนโทรล Line

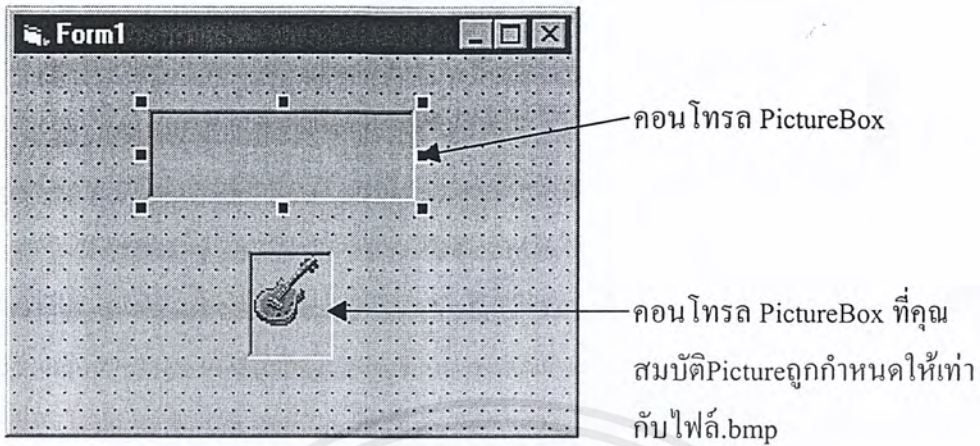
คอนโทรล Line เป็นคอนโทรลที่ค่อนข้างจะพิเศษซักหน่อยตรงที่เป็นคอนโทรลที่ไม่มีเหตุการณ์ ดังนั้นจึงเป็นคอนโทรลที่ไม่ตอบสนองต่อทุกเหตุการณ์ที่เกิดขึ้น ทั้งนี้เนื่องมาจากคอนโทรล Line ได้รับการออกแบบสำหรับเป็นเครื่องมือในการวาดกราฟพื้นฐานคือ เส้นตรง ให้กับฟอร์มเท่านั้น ดังนั้นจึงไม่มีความจำเป็นที่จะต้องตอบสนองต่อเหตุการณ์ที่เกิดขึ้น ซึ่งในขณะที่ออกแบบคอนโทรล Line ที่ถูกวางลงบนฟอร์มก็จะมีลักษณะดังตัวอย่างในรูปที่ 2.31



รูปที่ 2.31 คอนโทรล Line ในขณะออกแบบ

3) คอนโทรล PictureBox

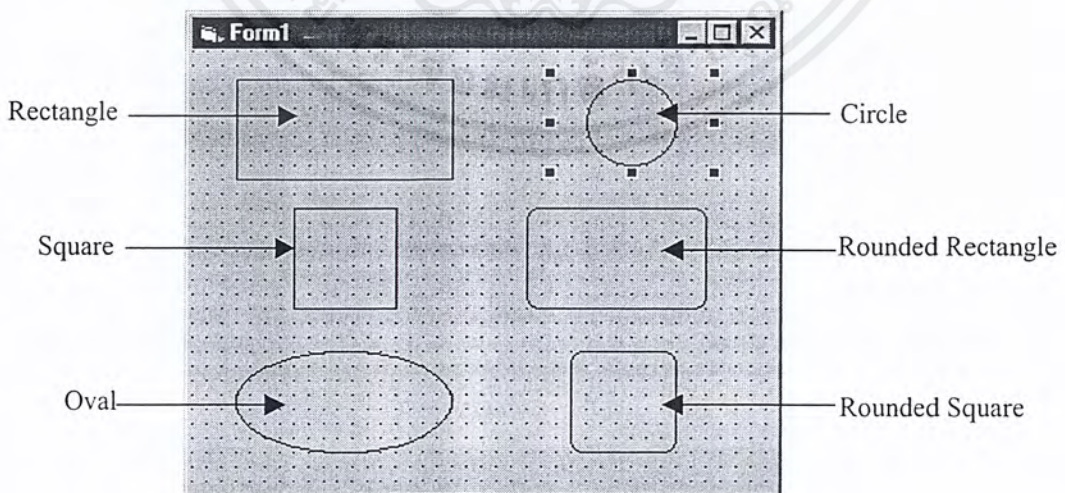
คอนโทรล PictureBox เป็นคอนโทรลที่ใช้ในการแสดงผลกราฟิก ได้เช่นเดียวกับคอนโทรล Image แต่คอนโทรล PictureBox เป็นคอนโทรลที่มีความสามารถเหนือกว่าในหลายๆ ด้าน เช่น การใช้ตัวบรรจุ (container) การแสดงผลข้อความแบบกราฟิก DDE การจัดการฟีกด้วยวิธีด้านกราฟิก ฯลฯ ด้วยความสามารถที่เหนือกว่าหลายๆ ด้านจึงทำให้คอนโทรล PictureBox มีความจำเป็นต้องใช้ทรัพยากรระบบมากกว่า และใช้เวลาในการประมวลผลเพื่อแสดงผลข้อมูลมากกว่าด้วยเช่นกัน ดังนั้นการนำคอนโทรล PictureBox ไปใช้ในงานแอปพลิเคชันก็ควรที่จะเลือกนำไปใช้ในส่วนที่จำเป็นจริงๆ และคอนโทรล Image ไม่สามารถรองรับความต้องการนั้นๆ ทั้งนี้เพื่อให้แอปพลิเคชันที่ได้สามารถทำงานได้รวดเร็วยิ่งขึ้น สำหรับในขณะที่ออกแบบคอนโทรล PictureBox ที่ถูกวางลงบนฟอร์มก็จะมีลักษณะดังตัวอย่างในรูปที่ 2.32



รูปที่ 2.32 คอนโทรล PictureBox ในขณะออกแบบ

4) คอนโทรล Shape

คอนโทรล Shape เป็นคอนโทรลที่ค่อนข้างจะพิเศษคล้ายกับคอนโทรล Line ดังที่กล่าวมาแล้ว ตรงที่เป็นคอนโทรลที่ไม่มีเหตุการณ์เช่นเดียวกัน ดังนั้นจึงเป็นคอนโทรลที่ไม่ตอบสนองต่อทุกเหตุการณ์ที่เกิดขึ้น ทั้งนี้เนื่องจากคอนโทรล Shape ได้รับการออกแบบสำหรับเป็นเครื่องมือในการวาดการพิกัดสี่เหลี่ยมพื้นฐานต่างๆ เช่น สี่เหลี่ยม รูปวงรีหรือรูปวงกลม เป็นต้น ให้กับฟอร์มเท่านั้น ดังนั้นจึงไม่มีความจำเป็นที่จะต้องมีการตอบสนองต่อเหตุการณ์ที่เกิดขึ้น ซึ่งในขณะออกแบบคอนโทรล Shape ที่ถูกนำมาวางลงบนฟอร์ม ก็จะมีลักษณะดังตัวอย่างในรูปที่ 2.33



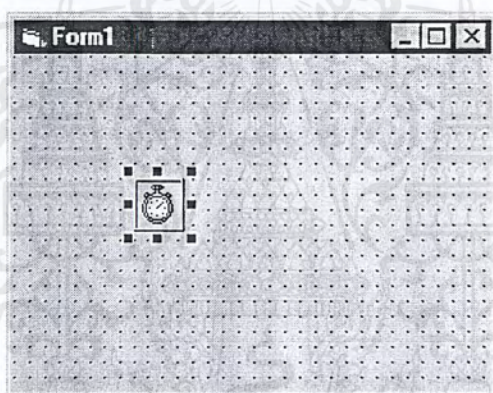
รูปที่ 2.33 คอนโทรล Shape ในขณะออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4 คอนโทรลภายในด้านเวลา

คอนโทรล Timer

คอนโทรล Timer เป็นคอนโทรลที่ใช้ในการควบคุมและจัดการเหตุการณ์ด้านเวลา ซึ่งเทียบได้กับประโยค ON TIME GOTO ของ QuickBASIC โดยผู้อ่านสามารถเขียนโค้ดเพื่อทำงานใดๆ เมื่อช่วงเวลาผ่านไปตามค่าที่กำหนด เช่น ทำการปรับปรุงการแสดงผลทุกๆ 1 นาที เป็นต้น โดยที่คอนโทรลนี้จะตอบสนองเหตุการณ์เดียวเท่านั้น แต่สามารถกำหนดให้แต่ละฟอร์มมีคอนโทรล Timer มากกว่า 1 คอนโทรล เนื่องจากคอนโทรล Timer เป็นคอนโทรลที่ทำงานตามนาฬิกาของระบบ ดังนั้นมันจึงถูกควบคุมโดยตัวของระบบเอง สำหรับวินโดวส์ 95, วินโดวส์ 98 และวินโดวส์ NT ในทางปฏิบัติจะไม่มีกัการจำกัดจำนวนของคอนโทรล Timer ในแต่ละฟอร์ม ดังนั้นจึงสามารถใช้งานคอนโทรล Timer พร้อมๆกันครั้งละหลายๆ คอนโทรลได้อย่างไม่จำกัด



รูปที่ 2.34 แสดงคอนโทรล Timer ในขณะออกแบบ

2.7 คอนโทรลอาร์เรย์ (Control Array)

คอนโทรลที่ถูกกำหนดลงในฟอร์มเดียวกัน สามารถที่จะมีชื่อ (คุณสมบัติ Name) เดียวกันได้ และคอนโทรลเหล่านี้ก็จะถูกอ้างอิงในรูปของอาร์เรย์ เช่น ถ้าหากกำหนดให้คอนโทรล Label จำนวน 6 คอนโทรลภายใต้ชื่อ Label1 ถูกกำหนดลงในฟอร์ม Visual Basic ก็จะกำหนดให้คอนโทรล Label1 เป็นคอนโทรลอาร์เรย์ลำดับที่ 0,1,2,3,4 และ 5 ตามลำดับ ซึ่งในกรณีของคอนโทรลอาร์เรย์นี้ สามารถที่จะทราบถึงจำนวนของสมาชิกหรือหมายเลขสมาชิกลำดับต่ำสุด (lower bound) หรือบนสุด (upper bound) ได้โดยการใช้ฟังก์ชัน LBound หรือ UBound ร่วมกับชื่อของคอนโทรลอาร์เรย์ที่ต้องการและก็เช่นเดียวกัน ก็สามารถที่จะอ่านจำนวนสมาชิกทั้งหมดของคอนโทรลอาร์เรย์ได้ด้วยฟังก์ชัน Count ดังรายละเอียดต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน Count LBound และ UBound

รายงานจำนวนของสมาชิก (Count) ของคอนโทรลอาร์เรย์ และค่าขอบเขตล่าง (LBound) ค่าขอบเขตบน (UBound) ของคอนโทรลอาร์เรย์

รูปแบบการใช้งาน

`ctrlArray().Count`

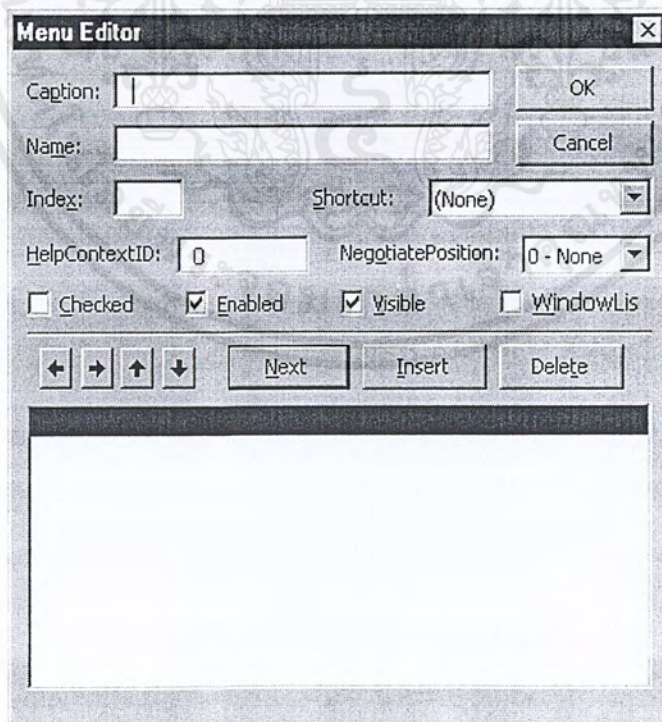
`ctrlArray().LBound`

`ctrlArray().UBound`

`ctrlArray()` หมายถึง ชื่อคอนโทรลอาร์เรย์

2.8 การสร้างเมนู

ในการสร้างเมนูบน Form จะต้องระบุ Form ที่จัดทำเมนูก่อน โดยการเรียก Form ที่จะจัดทำเมนูขึ้นมาบนจอภาพ จากนั้นให้เลือกเมนู Tools และ Menu Editor ตามลำดับ หรือไม่ถ้าไม่ต้องการคลิกหลายครั้ง ก็ให้คลิกที่ Icon “Menu Editor” ใน Toolbar แทน จะปรากฏจอภาพ Menu Editor ดังรูปที่ 2.35



รูปที่ 2.35 จอภาพ Menu Editor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน Menu Editor นี้จะมีส่วนที่น่าสนใจดังนี้

- 1) ข้อความที่ใส่ถ้าต้องการให้อักษรใดมีการขีดเส้นใต้ก็ให้ใส่ & ไว้ข้างหน้าตัวอักษรนั้น เช่น E&xit เป็นต้น จะปรากฏเป็นExit ซึ่งตัวอักษรที่ขีดเส้นใต้นี้จะใช้ร่วมกับคีย์ Alt
- 2) Caption ใช้สำหรับใส่ข้อความที่จะให้เป็นเมนู เช่น ข้อความ File,Edit,View ในจอภาพของ Visual Basic
- 3) Name ใช้สำหรับกำหนดชื่อให้กับแต่ละเมนู เพราะ Visual Basic จะถือว่าแต่ละเมนูก็คือ 1 Object
- 4) Shortcut จะอยู่ในรูปแบบของ Drop-Down List ซึ่งกำหนดให้ Hot Key ให้กับเมนู เช่น Alt-X
- 5) Checked ใช้สำหรับกำหนดให้เมนูนั้นเป็นแบบ On/Off (จะมีเครื่องหมายถูกอยู่หน้าเมนู) เช่น การ Set Default Printer ในเมนูของ Setting Printer ใน Windows 95
- 6) Enabled ใช้กำหนดว่าต้องการให้เมนูนั้นทำงานหรือไม่ ซึ่งถ้าไม่เลือก Enabled เมื่อทำการ Run เมื่อนั้นจะปรากฏเป็นข้อความสีเทาในเมนู ซึ่งไม่สามารถเลือกเพื่อใช้งานได้
- 7) Visible ใช้กำหนดให้เมนูนั้นปรากฏบนจอภาพหรือไม่
- 8) Window List ใช้กำหนดว่า เมื่อดังกล่าวจะมีเมนูย่อยหรือไม่

2.9 ฟังก์ชันที่เกี่ยวข้องกับ Sting

ฟังก์ชันที่เกี่ยวข้องกับ สตริง (Sting) จะปรากฏอยู่ในตาราง ดังนี้

ตารางที่ 2.2 ฟังก์ชันที่เกี่ยวข้องกับ Sting

ชื่อ	ประเภท	ใช้สำหรับ
Asc	ฟังก์ชัน	แสดงค่ารหัส แอสกี (ASCII) ของตัวอักษร
Chr\$	ฟังก์ชัน	แปลงค่ารหัส แอสกี ไปเป็นตัวอักษร
Format\$	ฟังก์ชัน	กำหนดรูปแบบของ สตริง
InStr	ฟังก์ชัน	หาตำแหน่งของตัวอักษรในคำ
LCase\$	ฟังก์ชัน	แปลงตัวอักษรใน สตริง ให้อยู่ในรูปอักษรตัวเล็ก
Left\$	ฟังก์ชัน	ตัดคำของ สตริง จากซ้ายไปขวา
Len	ฟังก์ชัน	แสดงค่าความยาวของ สตริง
LSet	ฟังก์ชัน	จัดตัวอักษรให้ชิดซ้าย

ตารางที่ 2.2 (ต่อ) ฟังก์ชันที่เกี่ยวข้องกับ Sting

ชื่อ	ประเภท	ใช้สำหรับ
LTrim\$	ฟังก์ชัน	ตัดช่องว่างของคำจากซ้ายไปขวา
Mid\$	ฟังก์ชัน	ตัดคำจากตำแหน่งเริ่มต้นและความยาวตามที่กำหนด
Mid\$	คำสั่ง	กำหนดค่าให้กับ สตริง ในตำแหน่งที่กำหนด
Right\$	ฟังก์ชัน	ตัดคำของ สตริง จากขวาไปซ้าย
RSet	ฟังก์ชัน	จัดตัวอักษรให้ชิดขวา
RTrim\$	ฟังก์ชัน	ตัดช่องว่างของคำจากทางขวาไปซ้าย
Str\$	ฟังก์ชัน	แปลงค่าของตัวเลขให้เป็น สตริง
Space\$	ฟังก์ชัน	แสดงช่องว่างตามที่กำหนด
Sting\$	ฟังก์ชัน	แสดงตัวอักษรซ้ำตามจำนวนที่กำหนด
Trim\$	ฟังก์ชัน	ตัดช่องว่างออกจากคำ
UCase\$	ฟังก์ชัน	แปลงตัวอักษรใน สตริง ให้อยู่ในรูปตัวอักษรใหญ่

2.9.1 การแปลงตัวเลขให้เป็น Sting

เราสามารถแปลง (Convert) ตัวเลขให้อยู่ในรูปของ สตริง ได้โดยใช้ฟังก์ชัน Str\$ เช่น

```
X% = 100
Y% = -100
XNum$ = Str$(x%)
YNum$ = Str$(y%)
```

ค่าของ XNum\$ จะมีค่าเท่ากับ "100" และค่าของ YNum\$ จะมีค่าเท่ากับ "-100"

แต่ทั้งนี้ฟังก์ชัน Str\$ จะไม่สามารถกำหนดรูปแบบของผลลัพธ์จากการแปลงได้ ดังนั้นจึงอาจนำฟังก์ชัน Format\$ มาใช้แทนได้ เช่น

```
X% = 234.9
Num1$ = Str$(X!)
Num2$ = Str$(X!, "###.00")
```

ค่าของ Num1\$ จะมีค่าเป็น "234.9" ในขณะที่ค่าของ Num2\$ จะมีค่าเป็น "234.90"

2.9.2 การตัดคำใน Sting

มีอยู่ 3 ฟังก์ชัน ด้วยกันที่ใช้ในการตัดคำ คือ Left\$, Mid\$ และ Right\$

```
T$ = "This car is rad"
A$ = Left$(T$, 4)
B$ = Right$(T$, 3)
C$ = Mid$(T$, 6, 3)
```

A\$ จะมีค่าเป็น "This" เพราะตัดคำจากทางซ้ายไป 4 ตัวอักษร B\$ จะมีค่าเป็น "red" เพราะตัดคำจากทางขวา 3 ตัวอักษร และ C\$ จะมีค่าเป็น "car" เพราะตัดคำเริ่มจากตำแหน่งที่ 6 ไป 3 ตัวอักษร

นอกจากนี้เรายังสามารถนำฟังก์ชันเหล่านี้ไปเขียนเป็นประโยคคำสั่ง (Statement) ซึ่งจะให้ผลลัพธ์ที่แตกต่างกันออกไป เช่น

```
T$ = "This car is red"
Mid$(T$, 6, 3) = "bat"
```

T\$ จะมีค่าเปลี่ยนไปเป็น "This car is red" เนื่องจากคำว่า "car" จะถูกแทนด้วย "bat"

2.9.3 การทำงานกับรหัส ASCII ของ Sting

ในบางครั้งเราอาจต้องการดูค่ารหัส แอสกี ของ สตริง ซึ่งสามารถทำได้โดยใช้ฟังก์ชัน Asc เช่น

```
A% = Asc("H")
```

A% จะมีค่าเป็น 72 ซึ่งเป็นรหัส แอสกี ให้อยู่ในรูปตัวอักษรได้เช่นกัน โดยฟังก์ชัน Chr\$

```
B$ = Chr$(72)
```

B\$ จะมีค่าเป็น “H” เพราะค่า 72 ในรหัส แอสกี จะหมายถึงตัวอักษร “H”

2.9.4 การตัดช่องว่างใน Sting ทิ้ง

ฟังก์ชันที่ใช้ในการตัดช่องว่างใน สตริง ทิ้งประกอบด้วย Ltrim\$,Trim\$ และ Rtrim\$

```
A$ = " Hello "
```

```
B$ = Ltrim$(A$) ...
```

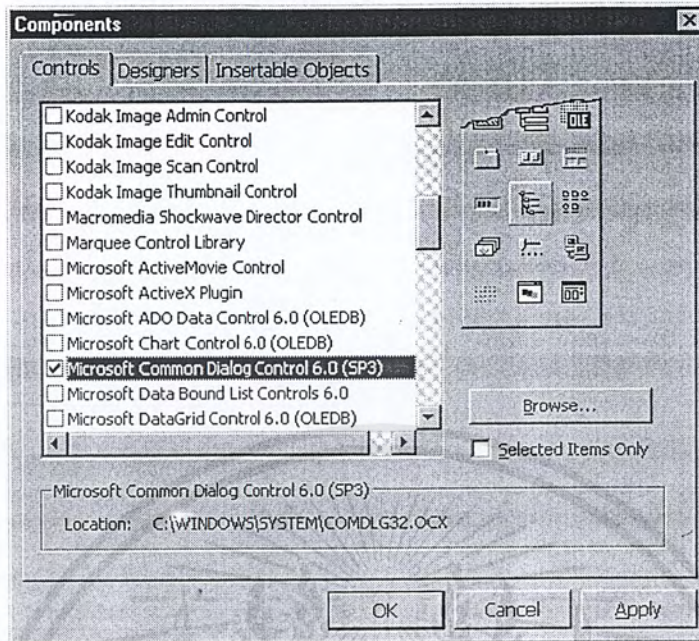
```
C$ = Rtrim$(A$)
```

```
D$ = Trim$(A$)
```

B\$ จะมีค่าเป็น “Hello ” (ช่องว่างทางด้านซ้ายหายไป) CS จะมีค่าเป็น “Hello ” (ช่องว่างทางด้านขวาหายไป) และD\$ จะมีค่าเป็น “Hello ” (ช่องว่างทางด้านซ้ายและขวาหายไป)

2.10 การใช้ CommonDialog Control

CommonDialog Control เป็น Control ที่ใช้สำหรับเรียกจอกภาพ Dialog Box ของ Windows มาใช้งาน แต่เนื่องจากไม่ใช่ Control มาตรฐานของ Visual Basic ดังนั้นเราต้องเพิ่ม Control ก่อน โดยเลือก “Microsoft Common Dialog Control 6.0” ในจอกภาพ Components ดังรูปที่ 2.36



รูปที่ 2.36 แสดงการเลือก Microsoft Common Dialog Control 6.0

คลิกเมาส์ที่ปุ่ม OK จะปรากฏ Control Common Dialog ใน Toolbox มาให้ Dialog Box ของ Windows แยกออกเป็น

Input Box เป็นจอภาพที่ใช้สำหรับรับข้อมูลที่ป้อนเข้ามา ซึ่งเป็นจอภาพเดียวที่ไม่ต้องใช้ Control Common Dialog แต่จะเรียกใช้โดยคำสั่ง

```
InputBox(prompt[, default] [, xpos] [, ypos] [, helpfile, context])
```

โดยที่	prompt	หมายถึง ข้อความที่ต้องการให้ปรากฏในจอภาพ
	title	หมายถึง ข้อความที่ต้องการให้ปรากฏในส่วน Title ของจอภาพ
	default	หมายถึงค่า Default ที่ต้องการให้ปรากฏใน TextBox
	xpos และ ypos	หมายถึงตำแหน่งเริ่มต้นในแนวแกน X และ Y ที่ต้องการให้จอภาพปรากฏ
	helpfile	หมายถึงชื่อของ Help File ที่ต้องการใช้ ซึ่งต้องระบุคู่กับ context ซึ่งเป็นหมายเลขของ Help File นั้น

File เป็นจอภาพที่ใช้สำหรับเปิด ไฟล์ (File) ในการเรียกจะใช้ Method showOpen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Font	เป็นจอภาพที่ใช้สำหรับเปิด ฟอนต์ (Font) ในการเรียกใช้จะใช้ Method showFont
Color	เป็นจอภาพที่ใช้กำหนดสี ในการเรียกจะใช้ Method showColor
Print	เป็นจอภาพที่ใช้สำหรับเลือก Printer และตั้งค่ากระดาษ ในการเลือกใช้จะใช้ Method showPrinter

2.11 การจัดการข้อมูล

ในการทำงานกับ Database โดยทั่วไปจะเกี่ยวข้องกับการเพิ่ม แก้ไข ลบและค้นหาข้อมูล ซึ่งแต่ละการทำงานจะอาศัย Method ที่แตกต่างกันไป ดังนั้นเราจำเป็นต้องทราบการใช้ Method ต่างๆ ในการจัดการกับข้อมูล Database

2.11.1 การเพิ่มข้อมูล

ในการเพิ่มข้อมูลเข้าไปยัง ตาราง (Table) ต่างๆ ใน ดาต้าเบส (Database) จะอาศัย Method AddNew เพื่อสร้าง เรคคอร์ด (Record) ว่างต่อเพิ่มเข้าไปใน เรคคอร์ดเซต (Recordset)

2.11.2 แก้ไขข้อมูล

Method ที่ใช้ในการบันทึกข้อมูลก็คือ Method Update โดยจะบันทึกข้อมูลที่ปรากฏอยู่ใน Object ที่เป็น Bound Control กลับลงไปยัง เรคคอร์ด ปัจจุบันใน เรคคอร์ดเซต

2.11.3 ลบข้อมูล

Method ที่ใช้สำหรับลบข้อมูล เรคคอร์ด ปัจจุบันออก เรคคอร์ดเซต ก็คือ Method Delete ในการลบ เรคคอร์ด ทุกครั้งหลังจากทำการลบ ควรจะมีการขยับ Pointer เนื่องจาก เรคคอร์ด ปัจจุบัน ได้ถูกลบไปเพื่อป้องกันการสับสนว่า Pointer ชี้ไปที่ เรคคอร์ด

2.11.4 ค้นหาข้อมูล

Method ที่ใช้ในการค้นหาข้อมูลใน เรคคอร์ดเซต จะประกอบไปด้วย

- 1) **FindFirst** ใช้สำหรับการค้นหาเรคคอร์ด โดยทิศทางในการหาจะเริ่มจาก เรคคอร์ด แรก ไปยัง เรคคอร์ด สุดท้ายใน เรคคอร์ดเซต จนกระทั่งพบ เรคคอร์ด ซึ่งตรงกับเงื่อนไขในการค้นหา
- 2) **FindLast** ใช้สำหรับการค้นหาเรคคอร์ด โดยทิศทางในการหาจะเริ่มจาก เรคคอร์ด สุดท้ายไปยัง เรคคอร์ด แรกใน เรคคอร์ดเซต จนกระทั่งพบ เรคคอร์ด ซึ่งตรงกับเงื่อนไขในการค้นหา

3) **FindNext** ใช้สำหรับการค้นหาเรคคอร์ด โดยทิศทางในการหาจะเริ่มจาก เรคคอร์ดปัจจุบันไปยัง เรคคอร์ด สุดท้ายใน เรคคอร์ดเซต จนกระทั่งพบ เรคคอร์ด ซึ่งตรงกับเงื่อนไขในการค้นหา

4) **FindPrevious** ใช้สำหรับการค้นหาเรคคอร์ด โดยทิศทางในการหาจะเริ่มจาก เรคคอร์ดปัจจุบันไปยัง เรคคอร์ด แรกใน เรคคอร์ดเซต จนกระทั่งพบ เรคคอร์ด ซึ่งตรงกับเงื่อนไขในการค้นหา



บทที่ 3

การออกแบบการสร้างและการทำงาน

3.1 ขั้นตอนในการออกแบบ

โครงการนี้เป็นโครงการที่ได้มีการพัฒนามาจากโครงการเดิมที่มีอยู่ ประกอบด้วยส่วนของบอร์ดอักษรไฟวิ่ง ซึ่งเป็นส่วนของฮาร์ดแวร์ที่ได้มีการสร้างไว้ก่อนแล้ว และยังมีส่วนของโปรแกรมหรือซอฟต์แวร์ซึ่งยังไม่มีคุณสมบัติเพียงพอที่จะนำมาใช้งานได้จริง ทางคณะผู้จัดทำโครงการจึงได้นำมาทำการพัฒนาด้านโปรแกรมหรือซอฟต์แวร์ให้ใช้งานได้ตรงตามวัตถุประสงค์ที่ได้ตั้งไว้ นอกจากนี้ยังได้พัฒนาในส่วนของฮาร์ดแวร์เพิ่มขึ้นอีก เพื่อให้โครงการนี้มีประสิทธิภาพยิ่งขึ้น ในส่วนของฮาร์ดแวร์นั้นผู้ที่ต้องการการออกแบบและการสร้าง สามารถศึกษาได้จากปริชญานิพนธ์ชื่อ บอร์ดอักษรไฟวิ่ง แอลอีดี เมตริกซ์ (LED Matrix Board) ฉบับปี พ.ศ.2541 ของนักศึกษาสาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์-อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

3.2 ฮาร์ดแวร์

3.2.1 วงจรนับเวลาและวันที่

ในส่วนของโครงการนี้ได้มีการสร้างในส่วนของฮาร์ดแวร์มาก่อนแล้ว แต่ยังคงขาดอยู่ในส่วนของการแสดงผลวันที่และเวลาปัจจุบัน ซึ่งในบอร์ดอักษรไฟวิ่ง ทั่วไปที่เราเห็นมักจะต้องแสดงข้อความในส่วนนี้ด้วย โดยจะใช้แสดงผลในกรณีที่ไม่ต้องการที่จะแสดงข้อความอะไรหรืออาจใช้ในกรณีไฟฟ้าดับ และเมื่อมีไฟบอร์ดก็จะทำการแสดงผลในส่วนของวันที่และเวลาปัจจุบันขึ้นมาเพื่อรอคอยการป้อนข้อความใหม่และยังแสดงผลในช่วงที่กำลังทำการการโปรแกรมข้อความใหม่ด้วย

ดังนั้นจึงจำเป็นต้องเพิ่มในส่วนของฮาร์ดแวร์ที่เรียกว่าวงจร “วงจรถับเวลาและวันที่” ซึ่งเป็นวงจรที่ทำหน้าที่ นับวันที่และเวลาปัจจุบัน แล้วนำไปแสดงผลที่ บอร์ดอักษรไฟวิ่ง ซึ่งจะต้องใช้ในส่วนของโปรแกรมในการนำข้อมูลจากวงจร วงจรถับเวลาและวันที่ไปแสดงผลที่ บอร์ดอักษรไฟวิ่ง ต่อไป

3.3 ส่วนของซอฟต์แวร์

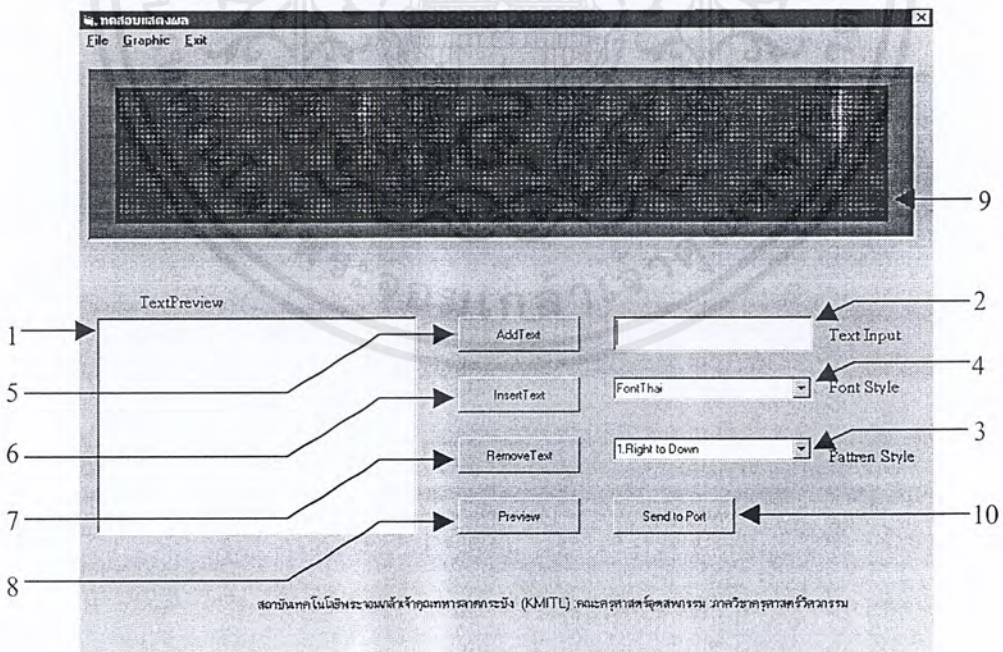
การทำงานของ ซอฟต์แวร์ สามารถแบ่งออกได้เป็น 2 ส่วนหลักๆ อันได้แก่ ซอฟต์แวร์ของ ส่วนควบคุมแสดงผลที่ คอมพิวเตอร์ เป็นส่วนควบคุมการแสดงผลที่ถูกส่งออกมาจาก พอร์ต คอม (Com) ของเครื่องคอมพิวเตอร์ ซึ่งในที่นี้จะเลือกใช้โปรแกรม Visual Basic 6 และยังมีซอฟต์แวร์ ในส่วนของ บอร์ดอักษรไฟวิ่ง ซึ่งจะไดกล่าวในส่วนของรายละเอียดตามหัวข้อต่อไป

3.3.1 ส่วนควบคุมการแสดงผลที่เครื่องคอมพิวเตอร์

ในการควบคุมการแสดงผลที่ บอร์ดอักษรไฟวิ่ง เราสามารถที่จะควบคุมการแสดงผลผ่าน พอร์ตคอม ของเครื่องคอมพิวเตอร์ ได้ทั้งนี้เพื่อความสะดวกในกรแก้ไขข้อมูลเมื่อนำเครื่อง คอมพิวเตอร์ ไปติดตั้งเรียบร้อยแล้ว ในที่นี้เราจะใช้ โปรแกรม Visual Basic 6 ในการเขียน โปรแกรมเพราะมีความง่ายในการเขียนและยังสามารถพัฒนาโปรแกรมได้ง่ายอีกด้วย ซึ่งมีลักษณะ การสร้างโดยสรุปดังนี้

1) ขั้นตอนการสร้าง หน้าต่าง Window ของโปรแกรม

จะมีการกำหนดหน้าที่การทำงานของปุ่ม ควบคุมการแสดงผล และ หน้าจอแสดงผลดังต่อไปนี้



รูปที่ 3.1 หน้าต่าง Window ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 เราสามารถอธิบายการกำหนดหน้าที่การทำงานของส่วนต่างๆ ของหน้าต่าง WINDOWS ของโปรแกรม ดังต่อไปนี้

ส่วนที่ 1 TEXT BOX1 (Text Preview)

เป็นส่วนการแสดงผลที่บอกให้เราเห็นว่า ในขณะที่เราทำการกำหนดข้อความที่จะนำไปแสดงผลที่ LED MATRIX BOARD ว่ามีข้อความอะไรอยู่บ้าง และมีลักษณะการแสดงผลลักษณะใด เช่น

kmitl ## left to right

หมายถึง ข้อความ kmitl แสดงการเลื่อนจาก ทางด้าน ซ้าย มาทางด้าน ขวา

ส่วนที่ 2 TEXT BOX2 (Text Input)

ใช้ในการกรอกข้อความที่ต้องการใช้แสดงผลที่ บอร์ดอักษรไฟวิ่ง ซึ่งจะใช้งานร่วมกับปุ่ม Add text ในส่วนที่ 5(Command button 1) ส่วนที่ 3 (List box 1 Pattern) และ การเลือก Font ในส่วนที่ 4 (List box 2 Font)

ส่วนที่ 3 LIST BOX 1 (Pattern Style)

ใช้สำหรับการเลือกลักษณะการแสดงผลของ บอร์ดอักษรไฟวิ่ง ของแต่ละข้อความ เช่น left to right (แสดงการเลื่อนจากทางด้าน ซ้าย มาทางด้าน ขวา)

ส่วนที่ 4 LIST BOX 2 (Font Style)

ใช้ในการเลือก Font ที่ต้องการจะแสดงใน บอร์ดอักษรไฟวิ่ง ซึ่งจะประกอบด้วย Font ทั้ง Font ภาษาไทย และ Font ภาษาอังกฤษ

ส่วนที่ 5 COMMAND BUTTON 1 (Add Text)

เป็นปุ่มที่ใช้ในการ Add ข้อความที่ต้องการแสดงใน บอร์ดอักษรไฟวิ่ง โดยข้อความที่เราต้องการจะแสดงจะกรอกข้อความเราจะกรอกลงใน text box 2(text)

ส่วนที่ 6 COMMAND BUTTON 2 (Insert Text)

ปุ่มที่ใช้ในการบรรจุข้อความลงใน TEXT BOX1

ส่วนที่ 7 COMMAND BUTTON 3 (Remove Text)

เป็นปุ่มที่ใช้ในการลบข้อความที่จะนำมาแสดงที่ บอร์ดอักษรไฟวิ่ง โดยข้อความเหล่านี้จะอยู่ใน TEXT BOX1 ในการลบเราจะลบที่ละข้อความเท่านั้น

ส่วนที่ 8 COMMAND BUTTON 4 (Preview)

ใช้ในการแสดงผลการทำงานเมื่อเรากำหนดการทำงานของผลการแสดงผลเสร็จเรียบร้อยแล้ว โดยจะมีตัวอย่างแสดงใน ส่วนที่ 9

ส่วนที่ 9 PICTURE BOX

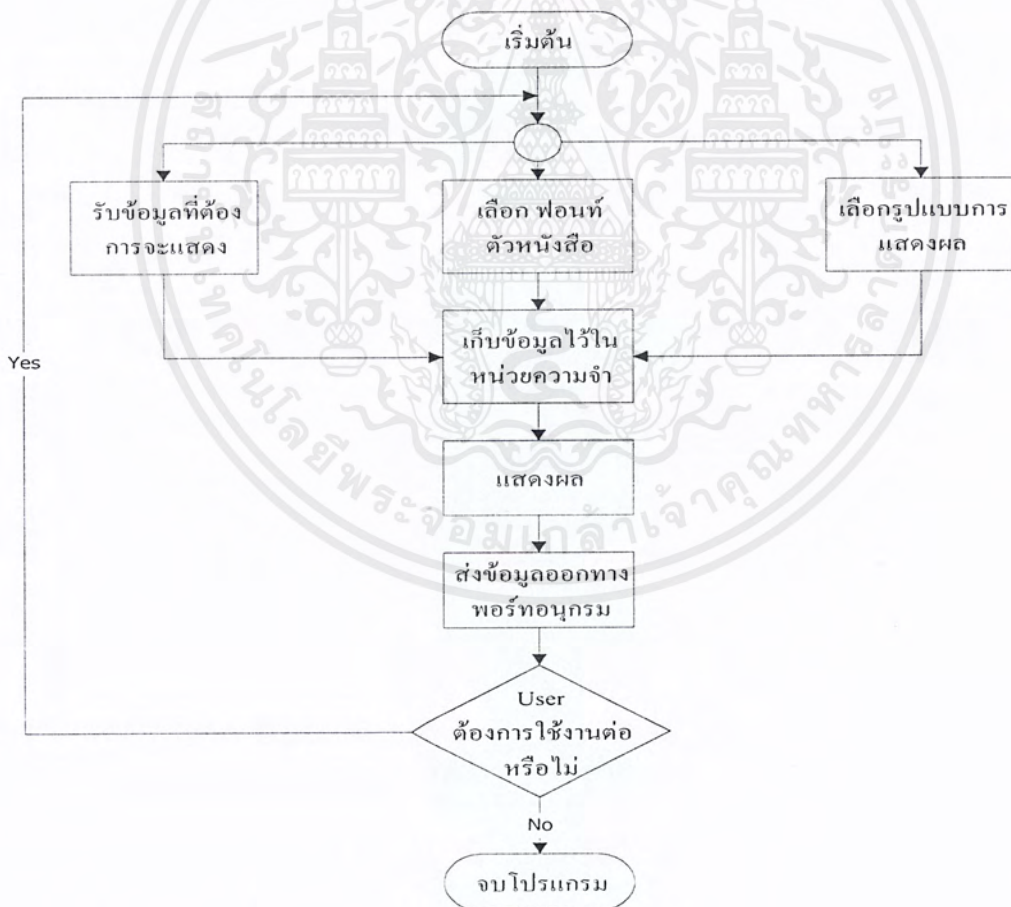
เป็นหน้าต่างแสดงตัวอย่างการแสดงผลที่ บอร์ดอักษรไฟวิ่ง ซึ่งจะแสดงตัวอย่างการเลื่อนของตัวอักษรใน บอร์ดอักษรไฟวิ่ง ลักษณะต่างๆเพื่อตรวจสอบลักษณะการแสดงผลจริงที่ บอร์ดอักษรไฟวิ่ง

ส่วนที่ 10 COMMAND BUTTON 5 (Send Port)

ปุ่มที่ใช้ในการส่งข้อมูลออกไปแสดงผลที่ บอร์ดอักษรไฟวิ่ง เพื่อใช้ในการแสดงผลจริงๆ โดยจะต่อเข้าทาง Port อนุกรมของตัวเครื่อง

2) การทำงานของโปรแกรมจำลองการแสดงผล

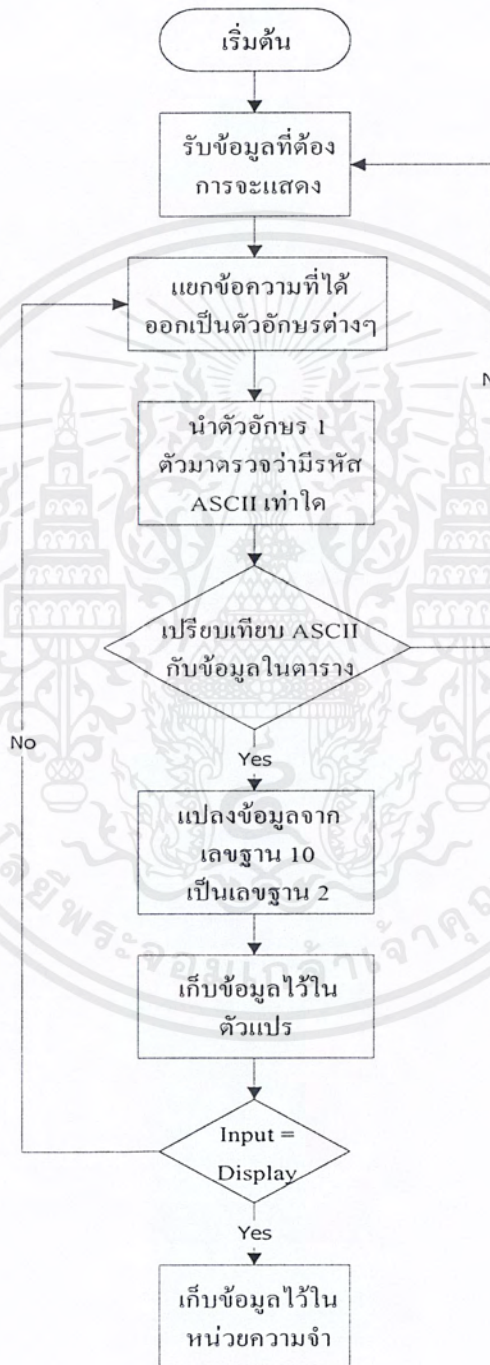
ในการใช้งานโปรแกรมจำลองการแสดงผลของ บอร์ดอักษรไฟวิ่ง ด้วยโปรแกรมภาษา Visual Basic สามารถอธิบายได้ดังรูปที่ 3.2



รูปที่ 3.2 ผังการทำงานของโปรแกรมหลักที่โปรแกรมจำลองการแสดงผล

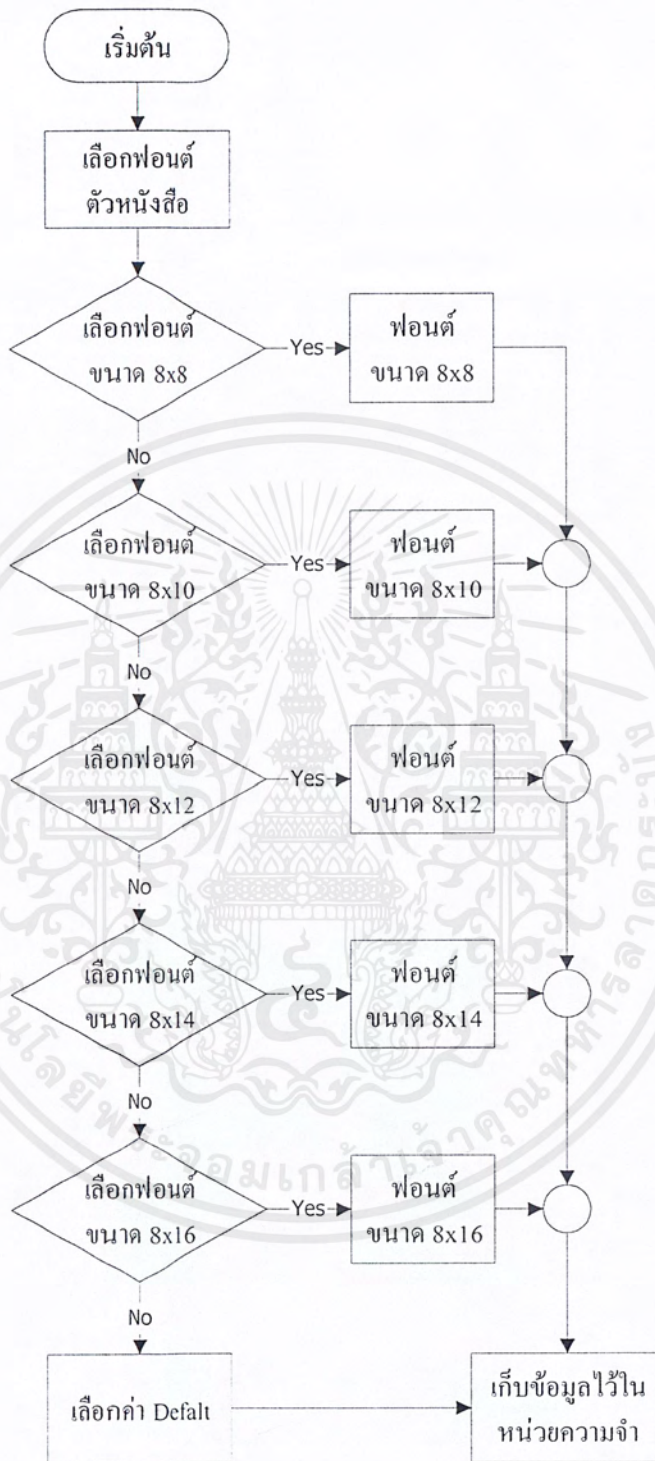
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในโปรแกรมจำลองการแสดงผลหลักยังประกอบด้วย โปรแกรมย่อย อีก 3 โปรแกรม คือ ส่วนรับข้อมูล ดังรูปที่ 3.3 ส่วนเลือกฟอนต์ตัวหนังสือ ดังรูปที่ 3.4 ส่วนเลือกรูปแบบการแสดงผล ดังรูปที่ 3.5



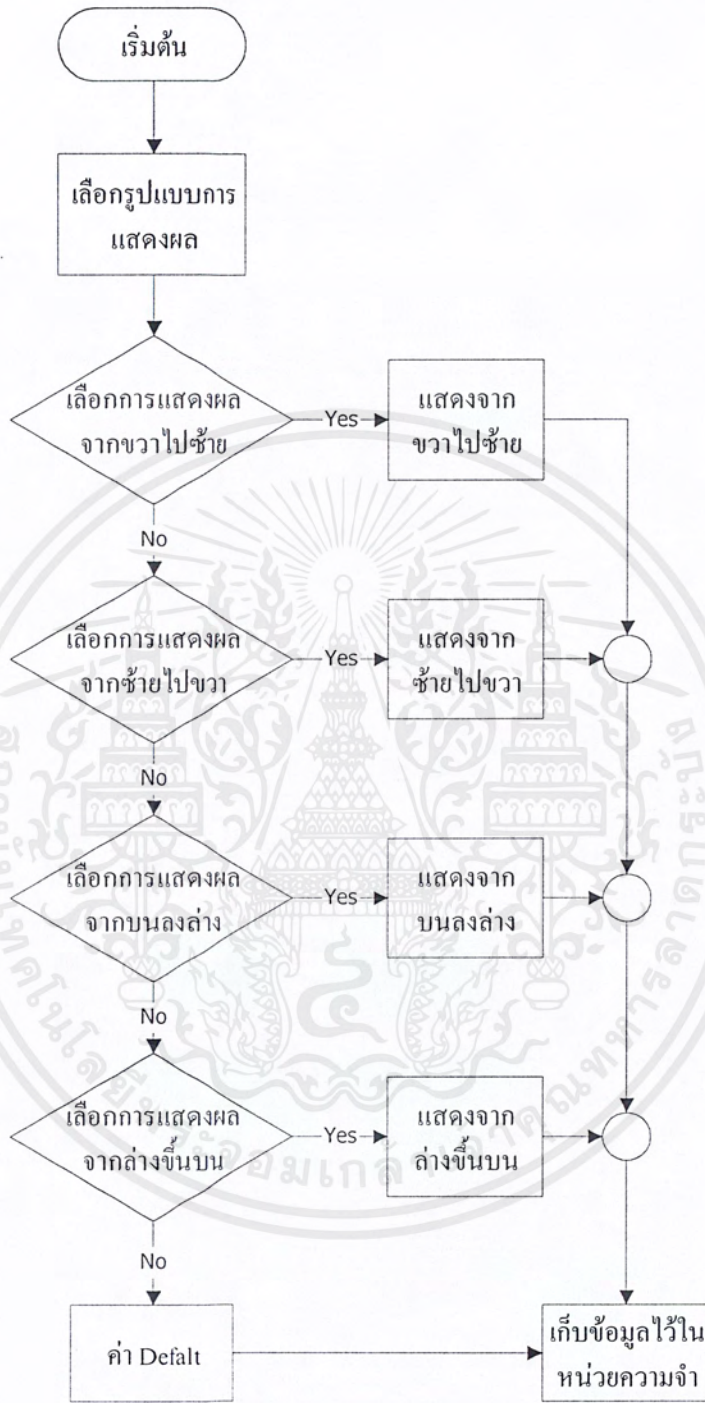
รูปที่ 3.3 ผังการทำงานของโปรแกรมรับข้อมูลที่ต้องการจะแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ผังการทำงานของโปรแกรมเลือกฟอนต์ตัวหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ผังการทำงานของโปรแกรมเลือกรูปแบบการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรมจำลองการแสดงผลของ บอร์ดอักษรไฟวิ่ง ด้วยโปรแกรมภาษา วิชาลเบสิก สามารถอธิบายได้ดังต่อไปนี้

1. เริ่มจากผู้ใช้ป้อนข้อมูลที่ต้องการแสดงผลด้วยคีย์บอร์ด ในช่องของ Text Input ที่กำหนดให้หลังจากนั้น โปรแกรมก็จะทำการตรวจสอบจำนวนตัวอักษรที่รับเข้ามาว่ามีกี่ตัว หลังจากนั้นจะทำการแยกข้อความที่รับเข้ามาทีละหนึ่งตัวอักษรจนครบทุกตัวในข้อความ เพื่อใช้ในการตรวจสอบว่าตัวอักษรนั้นมีรหัส แอสกี เท่าใด แล้วก็จะไปเรียกข้อมูลที่เป็นตัวอักษร(แยกออกไปจัดเก็บต่างหากจะกล่าวถึงในหัวข้อการจัดเก็บข้อมูล)ในไฟล์ของข้อมูลโดยนำ แอสกี ที่ได้ไปอ้างอิงกับเรคคอร์ดของไฟล์ข้อมูล เช่น

Text Input A

เมื่อรับข้อมูล A เข้ามา ก็จะตรวจสอบว่า A มีรหัส แอสกี เท่ากับ 65 หลังจากนั้นก็จะไปเปิดไฟล์ที่จัดเก็บข้อมูลไว้โดยอ้างอิงเรคคอร์ดที่ 65 ก็จะได้ข้อมูลตัวอักษร A ในรูปของเลขฐาน 16

หลังจากได้ข้อมูลของข้อความที่ต้องการแสดงผลในรูปของเลขฐาน 16 แล้วก็จะนำข้อมูลเลขฐาน 16 มาเป็นข้อมูลเลขฐาน 2 (ตัวอักษรหนึ่งตัวจะประกอบด้วยกลุ่มของข้อมูลที่เป็นเลขฐาน 16 จำนวน 24 ชุด) และข้อมูลที่ได้จากเลขฐาน 2 ก็จะนำมาพล็อตตามตำแหน่งต่างๆเพื่อให้เกิดตัวอักษร

หลังจากทำตัวอักษรตัวแรกเสร็จก็จะไปทำตัวอักษรตัวถัดไปจนกว่าจะจบข้อความ

2. ผู้ใช้เลือกฟอนต์ที่ต้องการ โดยมีให้เลือกดังนี้

ฟอนต์ขนาด 8x8

ฟอนต์ขนาด 8x10

ฟอนต์ขนาด 8x12

ฟอนท์ขนาด 8x14

ฟอนท์ขนาด 8x16

3. ผู้ใช้เลือกรูปแบบในการแสดงผล ซึ่งสามารถเลือกได้ดังนี้

การแสดงผลข้อความจากขวาไปซ้าย

การแสดงผลข้อความจากซ้ายไปขวา

การแสดงผลข้อความจากบนลงล่าง

การแสดงผลข้อความจากล่างขึ้นบน

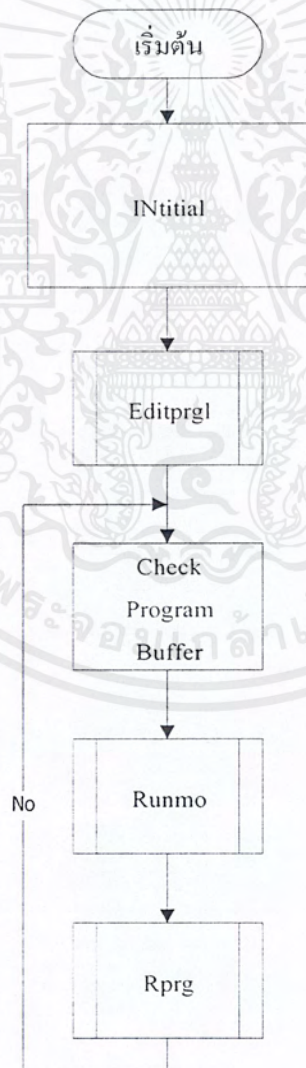
4. กดปุ่ม Add Text เพื่อนำข้อมูล,แบบตัวอักษร,รูปแบบการแสดงผลไปเก็บไว้ใน Text Preview ซึ่งจะเป็นตัวบอกว่าข้อความกลุ่มนี้ ใช้สำหรับการแสดงผล

5. กดปุ่มเพื่อจำลองการแสดงผลบนคอมพิวเตอร์

6. เมื่อได้รูปแบบการแสดงผลเรียบร้อยแล้วก็กดปุ่ม Send Port ซึ่งเป็นการส่งข้อมูลผ่านทางพอร์ตอนุกรมไปยังตัวบอร์ดอักษรไฟวิ่งและที่ถูกส่งไปจะประกอบไปด้วย ข้อความที่ต้องแสดงผล,แบบตัวอักษรและรูปแบบในการแสดงผล โดยข้อมูลทั้งหมดจะส่งไปจัดเก็บในหน่วยความจำของตัวบอร์ดอักษรไฟวิ่ง

3.3.2 การทำงานของโปรแกรมหลักที่ บอร์ดอักษรไฟวิ่ง

สามารถอธิบายได้ดังรูปที่ 3.6



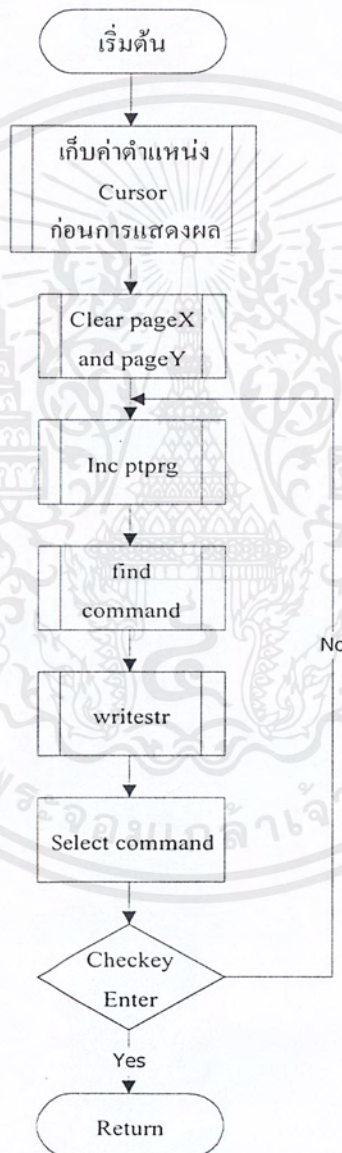
รูปที่ 3.6 ผังการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 เป็นการแสดงการทำงานของโปรแกรมหลักโดยมีโปรแกรมย่อยคือ

1. โปรแกรมย่อย Initial คือส่วนที่ทำหน้าที่กำหนดค่าเริ่มต้นในการทำงานให้กับโปรแกรม และกำหนดค่าควบคุม Hardware ให้สามารถทำงานได้ตามต้องการ

2. โปรแกรมย่อย EditPrg คือส่วนที่ทำหน้าที่รับข้อความและคำสั่งจากผู้ใช้งานและนำค่าที่ได้เก็บใน Program Buffer (Mprg)

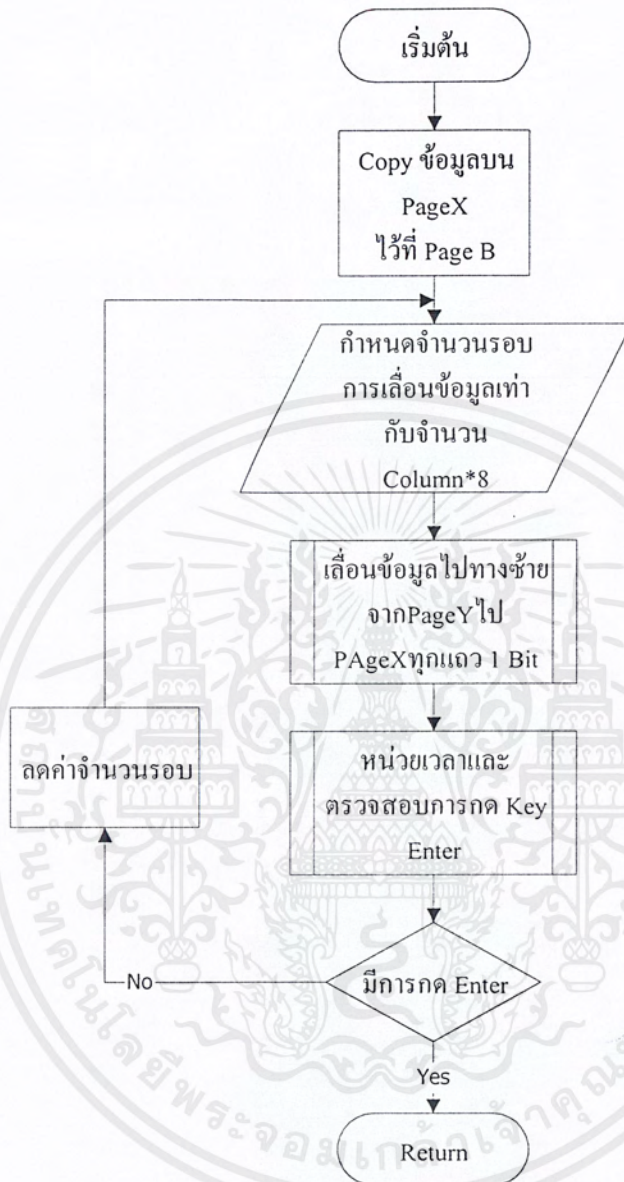


รูปที่ 3.7 ผังการทำงานของโปรแกรม Runtime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

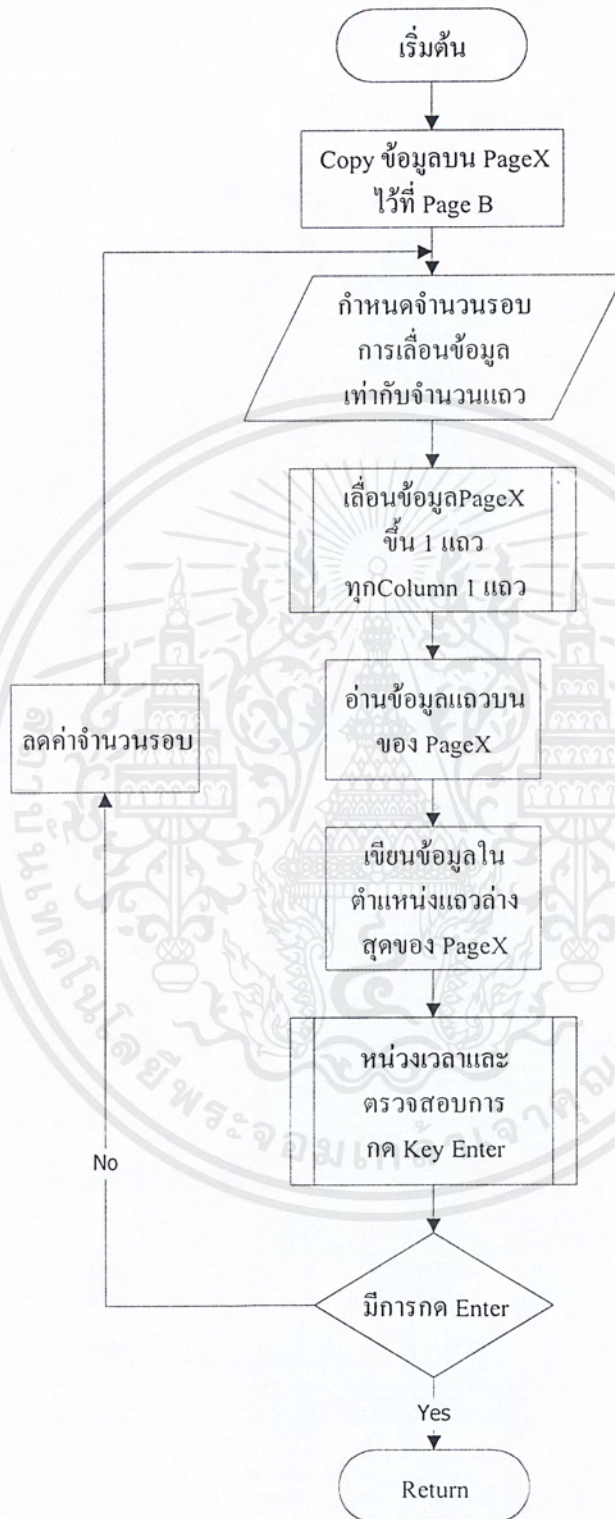
จากรูปที่ 3.8 เป็นแผนผังการทำงานของโปรแกรม Runtime สามารถอธิบายการทำงานแต่ละส่วนได้ดังนี้

1. โปรแกรมย่อย Svecursor เป็นการเก็บค่าตำแหน่ง Cursor ก่อนที่จะแสดงข้อความเพื่อจะสามารถคืนค่าตำแหน่ง Cursor เดิมเมื่อเข้าสู่โปรแกรม Editprg
2. โปรแกรมย่อย Incptprg ทำการเพิ่มค่าตำแหน่ง ตัวชี้ตำแหน่งใน โปรแกรม Buffer
3. โปรแกรมย่อย Clear pageX and pageY ทำการลบข้อมูลบน PAGE X และ PAGE Y เพื่อที่จะได้เขียนข้อความที่ต้องการ
4. Final Command ค้นหาชุดคำสั่งที่ต้องการแสดงผล
5. WriteStr เป็นการอ่าน Code Font จากโปรแกรม Buffer เพื่อเปรียบเทียบนำค่าจาก Font Buffer เขียนบน PAGE Y
6. Select Command ทำหน้าที่อ่านค่า Code Command มาเปรียบเทียบกับโปรแกรมคำสั่งมากระทำต่อข้อความที่อยู่ PAGE Y ให้แสดงผลบน PAGE X โดยสามารถแสดงการกระทำของโปรแกรมคำสั่งย่อยต่างๆ ดังรูปที่ 3.8 และรูปที่ 3.9



รูปที่ 3.8 ผังการทำงานของฟังก์ชัน S ๒๑

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ผังการทำงานฟังก์ชัน SCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดสอบแสดงผลข้อความ แอลอีดี

การทดลองการแสดงผลเพื่อทดสอบการใช้งานของแผงแสดงผลว่าเป็นปรกติหรือไม่ เพื่อในเวลากการเขียนโปรแกรมควบคุมการแสดงผลบอร์ดอักษรไฟวิ่ง จะทำงานไม่ผิดพลาด โดยจะมีขั้นตอนการทดสอบเครื่องดังนี้

4.1.1 ตรวจสอบซ็อกเก็ต (Socket) ไมโครโปรเซสเซอร์ เบอร์ Z80180

เนื่องจากเมื่อทำการป้อนโปรแกรมทดสอบในครั้งแรก บอร์ดอักษรไฟวิ่ง จะไม่ทำงาน ทำการถอดไมโครโปรเซสเซอร์ เบอร์ Z8180 ออกจาก ซ็อกเก็ต แล้วใส่ใหม่ จึงสามารถใช้งานได้

4.1.2 ทดสอบใช้คำสั่ง สแกนให้ บอร์ดอักษรไฟวิ่ง ทางแนวตั้ง

เพื่อทดสอบการใช้งานของ แผงแสดงผล แอลอีดี สามารถใช้งานได้เป็นปรกติหรือไม่ โดยใช้คำสั่งให้ บอร์ดอักษรไฟวิ่ง แสดงผลทางตั้ง แผงการทดลอง แอลอีดี แผงที่ 3 เกิดความบกพร่องติดเพียงบางแถว

4.1.3 ทดสอบเขียนคำสั่งให้ บอร์ดอักษรไฟวิ่ง ติดหมดทุกดวง

เพื่อทดสอบว่า ไดโอดเปล่งแสง ติดทุกดวงหรือไม่ ผลการทดลอง มีไดโอดเปล่งแสง บางดวงไม่ติด ทำการย้ายรอยบัดกรีบางจุดที่ไดโอดเปล่งไม่ติด

4.1.4 ตรวจสอบ สายแพ ที่เชื่อมต่อในวงจร

จากการทดสอบเขียนโปรแกรมทดสอบให้ ไดโอดเปล่ง ติดทุกดวงปรากฏว่า สายแพ ที่เชื่อมต่อกับแผงแสดงผล แอลอีดี แผงที่ 3 เกิดการเสียหายทำการเปลี่ยนสายแพ และเปลี่ยน จัมเปอร์ Jumper ใหม่

4.1.5 ตรวจสอบแหล่งจ่ายไฟในวงจร

ผลการทดสอบ บอร์ดอักษรไฟวิ่ง ในบางครั้ง ไดโอดเปล่ง จะเกิดการ กระตุก จึงทำการตรวจสอบระบบไฟที่ แหล่งจ่ายไฟ (Supply) ให้ใช้งานได้ตามปรกติ

4.1.6 ตรวจสอบ ทรานซิสเตอร์ เบอร์ TIP127

ทรานซิสเตอร์ เบอร์ TIP127 ทำหน้าที่ขับไดโอดเปล่งแสงทางแนวนอน บอร์ดอักษรไฟวิ่ง บางแถวทางแนวนอนบางแถวจะไม่ติดทำการเปลี่ยนทรานซิสเตอร์ เบอร์ TIP127 ใหม่ให้สามารถแสดงผลได้

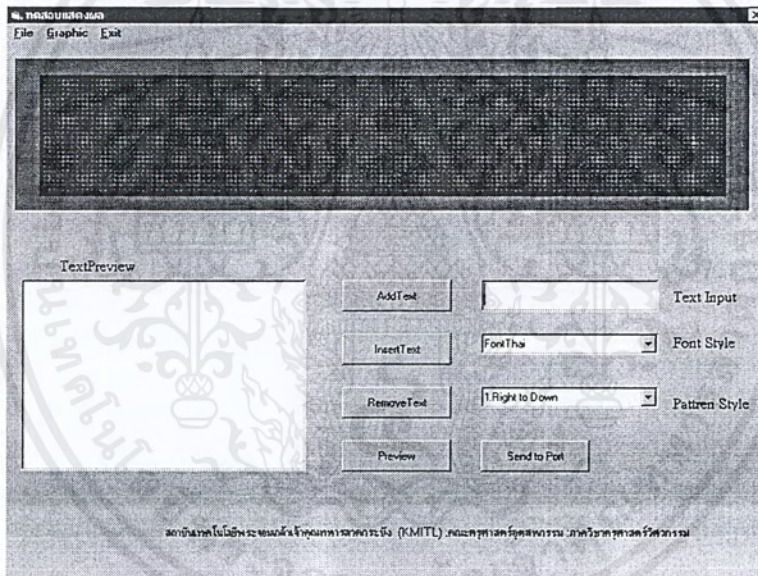
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบโปรแกรมควบคุมแผงแสดงผล แอลอีดี

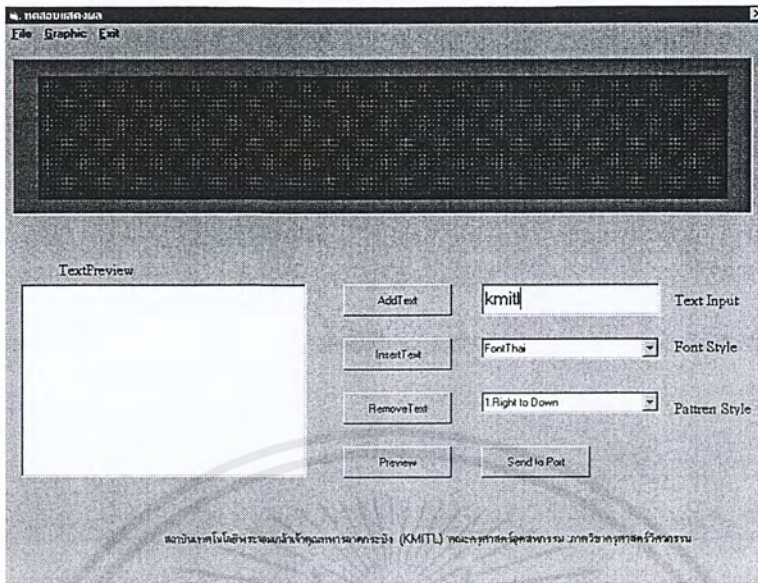
โปรแกรมควบคุมการแสดงผลที่เครื่องคอมพิวเตอร์ จะมีลักษณะการแสดงผลตัวอย่างในโปรแกรมเพื่อตรวจสอบสภาวะการทำงานของข้อความว่ามีการเคลื่อนที่ของข้อความลักษณะไหน

4.2.1 เปิดโปรแกรมให้อยู่ในสภาวะพร้อมใช้งาน

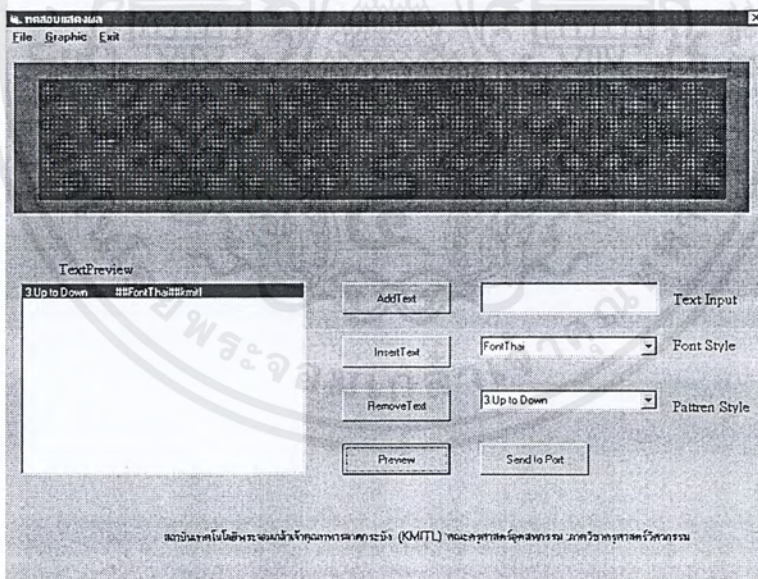
เรียกโปรแกรมโดยดับเบิลคลิกเมาท์ที่ไอคอน LED Show จะปรากฏหน้าต่างของโปรแกรม ดังรูปที่ 4.1 ป้อนข้อความที่ช่องรับข้อความ (Text Input) จะปรากฏตัวอักษรที่ช่องรับข้อความ Text Input ดังรูปที่ 4.2 ดึงข้อความเขาไปไว้ใน ช่องแสดงข้อความ (Text Input) โดยปุ่มใส่ข้อความ (Add Text) จะได้ผลการทดลองดังรูปที่ 4.3



รูปที่ 4.1 หน้าต่างของ โปรแกรมควบคุมการทำงาน บอร์ดอักษรไฟวิ่ง



รูปที่ 4.2 ข้อความที่ปรากฏใน Text Input

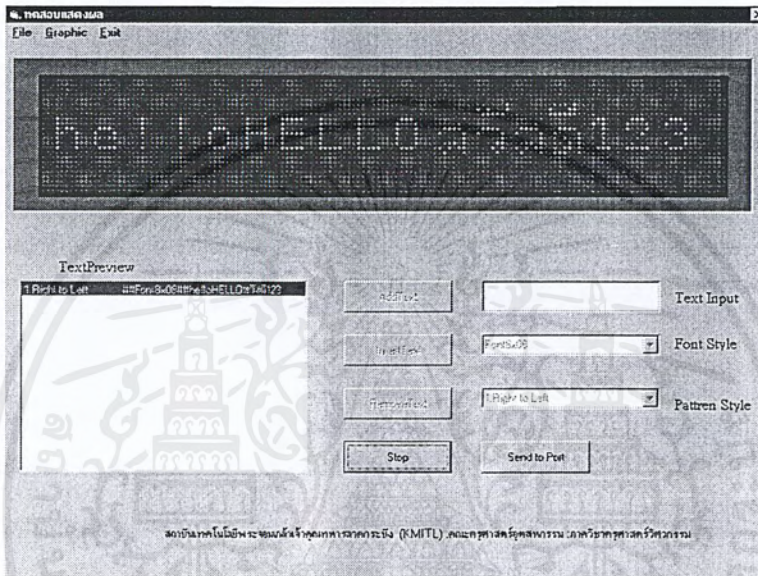


รูปที่ 4.3 ข้อความที่อยู่ใน Text Preview

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 08 (Font 8 × 08)

ป้อนข้อความที่ข้อความเป็นภาษาอังกฤษตัวพิมพ์เล็ก ภาษาอังกฤษตัวพิมพ์ใหญ่ ข้อความภาษาไทย ตัวเลข โดยเมื่อป้อนข้อความแล้วกดที่ปุ่ม กำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะได้ผลดังรูปที่ 4.4



รูปที่ 4.4 ข้อความขนาด Font 8×08

4.2.3 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 10 (Font 8 × 10)

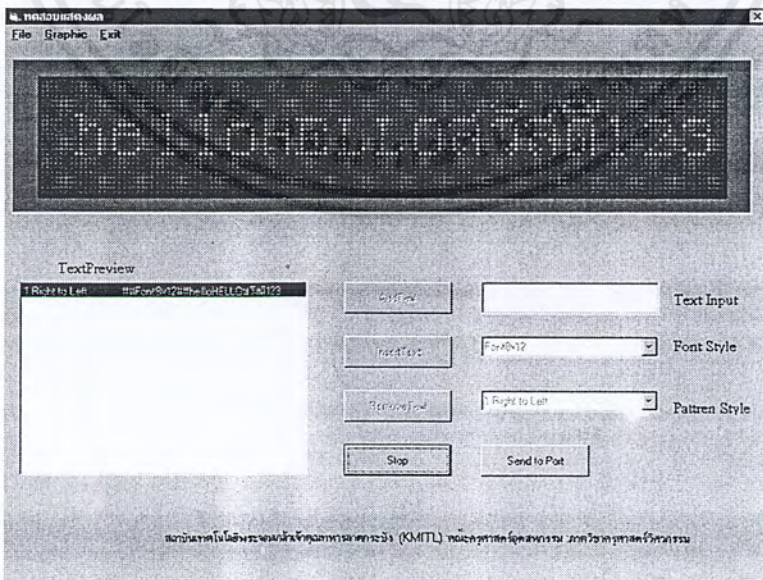
ป้อนข้อความที่ข้อความเป็นภาษาอังกฤษตัวพิมพ์เล็ก ภาษาอังกฤษตัวพิมพ์ใหญ่ ข้อความภาษาไทย ตัวเลข โดยเมื่อป้อนข้อความแล้วกดที่ปุ่ม กำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะได้ผลดังรูปที่ 4.5



รูปที่ 4.5 ข้อความขนาด Font 8×10

4.2.4 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 12 (Font 8 × 12)

ป้อนข้อความที่ข้อความ เป็นภาษาอังกฤษตัวพิมพ์เล็ก ภาษาอังกฤษตัวพิมพ์ใหญ่ ข้อความภาษาไทย ตัวเลข โดยเมื่อป้อนข้อความแล้วกดที่ปุ่ม กำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะ ได้ผลดังรูปที่ 4.6

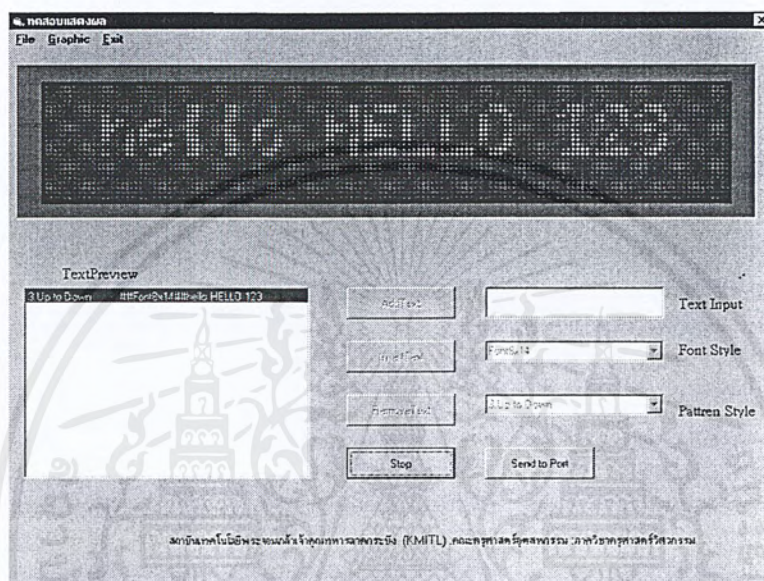


รูปที่ 4.6 ข้อความขนาด Font 8×12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 14 (Font 8 × 14)

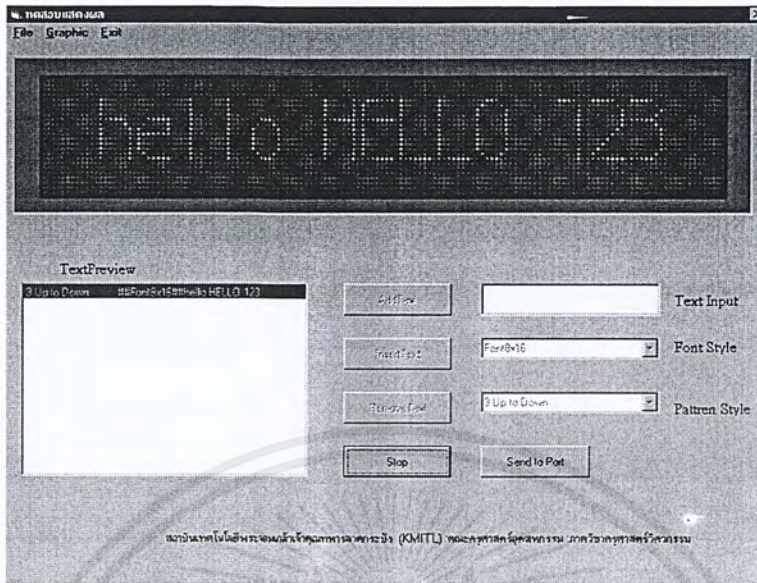
ป้อนข้อความที่ข้อความเป็นภาษาอังกฤษตัวพิมพ์เล็ก ภาษาอังกฤษตัวพิมพ์ใหญ่ ตัวเลข โดยเมื่อป้อนข้อความแล้วกดที่ปุ่ม กำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะได้ผลดังรูปที่ 4.7



รูปที่ 4.7 ข้อความขนาด Font 8×14

4.2.6 ทดลองป้อนข้อมูลแสดงผล ขนาดฟอนต์ 8 × 16 (Font 8 × 16)

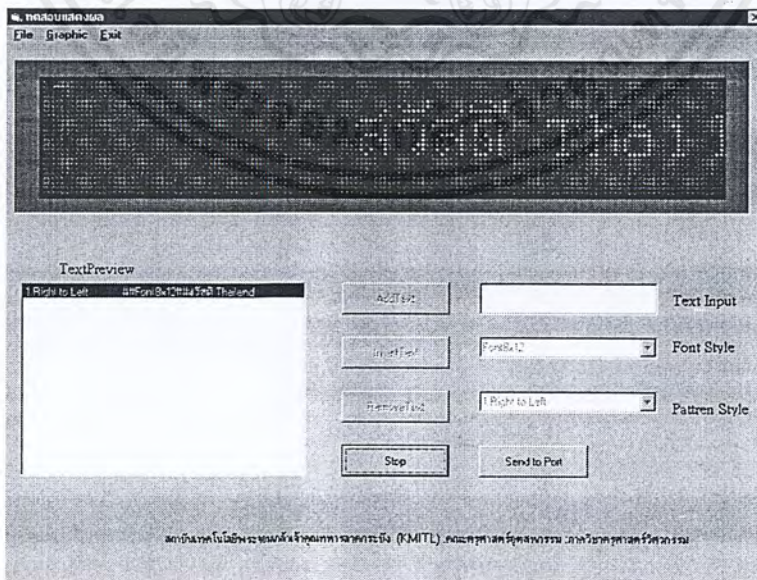
ป้อนข้อความที่ข้อความเป็นภาษาอังกฤษตัวพิมพ์เล็ก ภาษาอังกฤษตัวพิมพ์ใหญ่ ตัวเลข โดยเมื่อป้อนข้อความแล้วกดที่ปุ่ม กำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะได้ผลดังรูปที่ 4.8



รูปที่ 4.8 ข้อความขนาด Font 8×16

4.2.7 การทดลองการเคลื่อนที่ของตัวอักษรจากขวาไปซ้ายที่บอร์ดจำลองการแสดงผล

เมื่อกำหนดข้อความ (Text Input) ลักษณะของตัวอักษร (Font Style) เลือกลักษณะการแสดงผลหัวข้อ ขวาไปซ้าย (Right to Left) ที่ช่องเลือกลักษณะการแสดงผล (Pattern Style) กดที่ปุ่มกำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะ ได้การแสดงผลดังรูปที่ 4.9

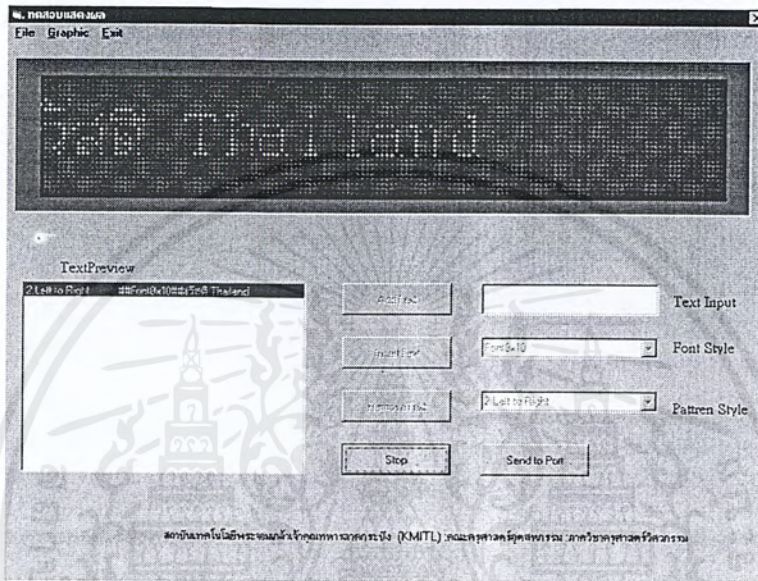


รูปที่ 4.9 การเคลื่อนที่ของข้อความจากขวาไปซ้ายที่บอร์ดจำลองการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.8 การทดลองการเคลื่อนที่ของตัวอักษรจากซ้ายไปขวาที่บอร์ดจำลองการแสดงผล

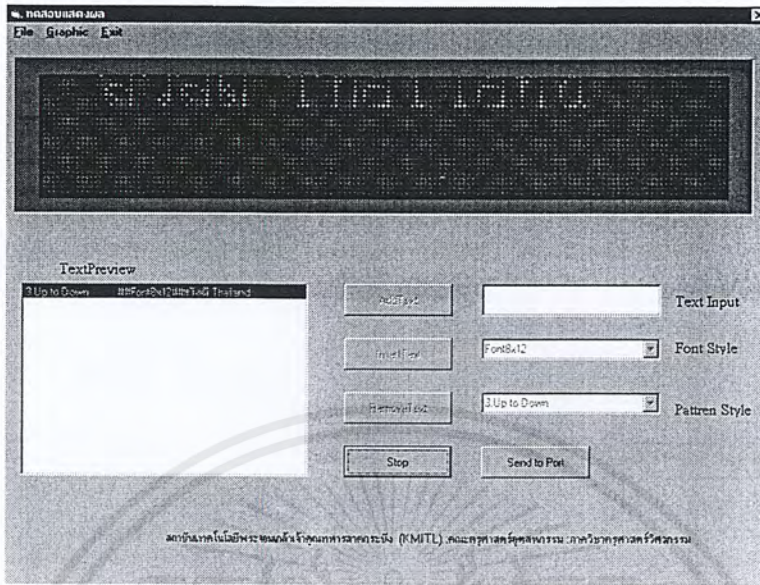
เมื่อกำหนดข้อความ (Text Input) ลักษณะของตัวอักษร (Font Style) เลือกลักษณะการแสดงผล หัวข้อ ซ้ายขวาไป (Left to Right) ที่ช่องเลือกลักษณะการแสดงผล (Pattern Style) กดที่ปุ่ม กำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะได้รับการแสดงผลดังรูปที่ 4.10



รูปที่ 4.10 การเคลื่อนที่ของข้อความจากซ้ายไปขวาที่บอร์ดจำลองการแสดงผล

4.2.9 การทดลองการเคลื่อนที่ของตัวอักษรจากบนลงล่างที่บอร์ดจำลองการแสดงผล

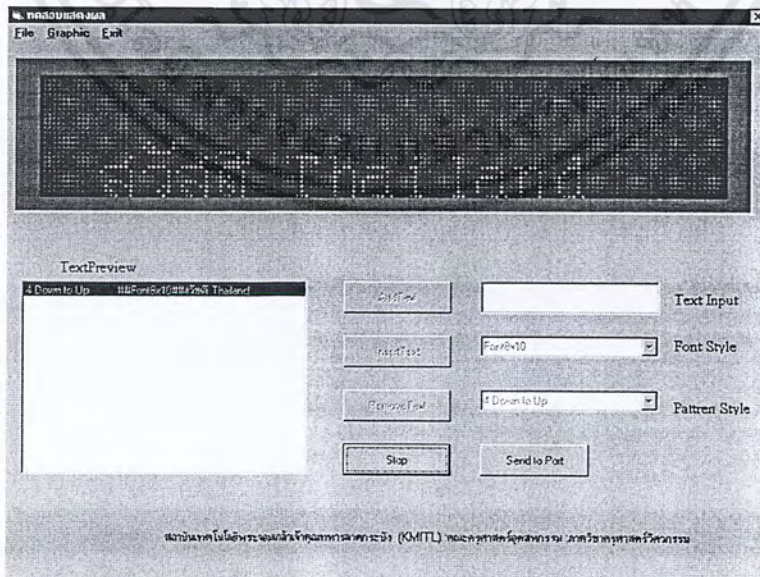
เมื่อกำหนดข้อความ (Text Input) ลักษณะของตัวอักษร (Font Style) เลือกลักษณะการแสดงผล หัวข้อ บนลงล่าง (Up to Down) ที่ช่องเลือกลักษณะการแสดงผล (Pattern Style) กดที่ปุ่ม กำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะได้รับการแสดงผลดังรูปที่ 4.11



รูปที่ 4.11 การเคลื่อนที่ของข้อความจากบนลงล่าง ที่บอร์ดจำลองการแสดงผล

4.2.10 การทดลองการเคลื่อนที่ของตัวอักษรจากล่างขึ้นบนที่บอร์ดจำลองการแสดงผล

เมื่อกำหนดข้อความ (Text Input) ลักษณะของตัวอักษร (Font Style) เลือกลักษณะการแสดงผลหัวข้อ ล่างขึ้นบน (Down to Up) ที่ช่องเลือกลักษณะการแสดงผล (Pattern Style) กดที่ปุ่มกำหนดข้อความ (Add Text) และกดที่ปุ่ม แสดงผล (Preview) จะได้การแสดงผลดังรูปที่ 4.12

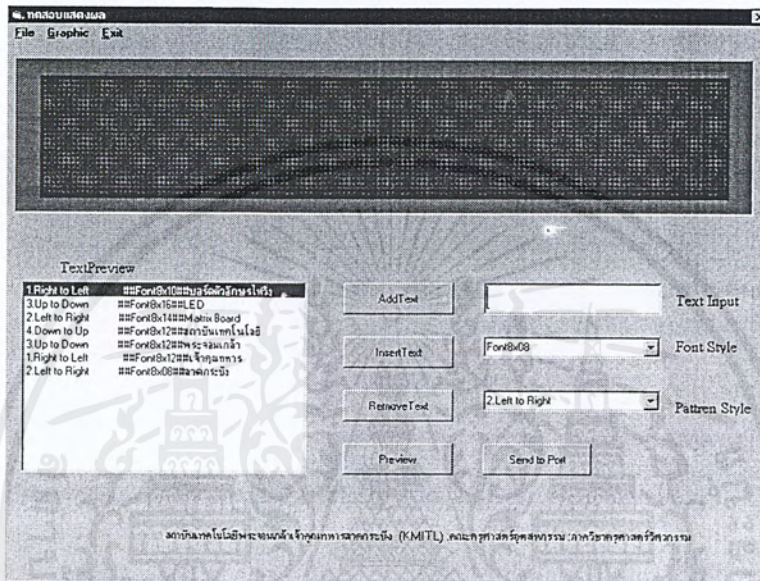


รูปที่ 4.12 การเคลื่อนที่ของข้อความจากล่างขึ้นบน ที่บอร์ดจำลองการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

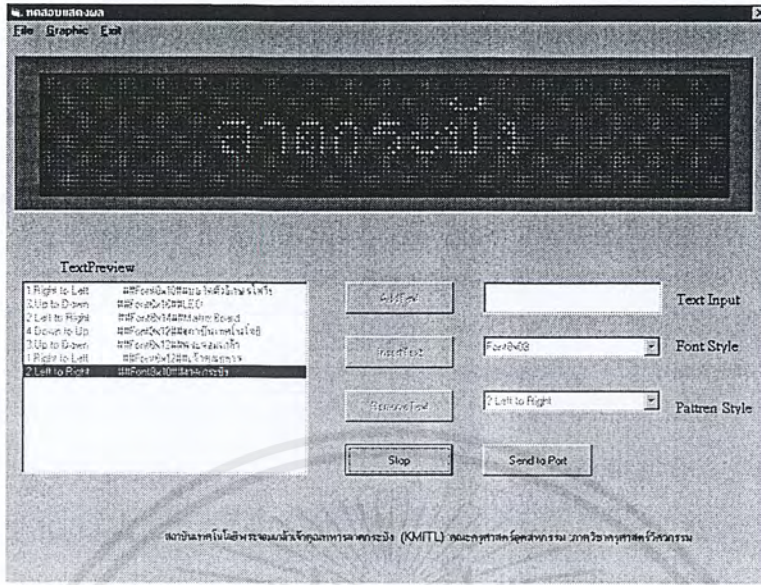
4.2.11 การทดลองการเคลื่อนที่ของตัวอักษรหลายลักษณะ(Pattern Style)

ทำการป้อนข้อความและเลือกลักษณะการแสดงผลในหลายๆ ลักษณะ และหลายๆ ข้อความ ติดต่อกันดังรูปที่ 4.13 เมื่อสุดที่ข้อความสุดท้ายก็ให้กลับมาแสดงที่ข้อความแรกใหม่ เป็นเช่นนี้เรื่อยๆ ไปจนกว่าจะหยุดทำงานหรือกำหนดข้อความ และลักษณะการแสดงผลของข้อความใหม่



รูปที่ 4.13 ลักษณะการป้อนข้อความและเลือก Pattern Style หลายๆแบบ

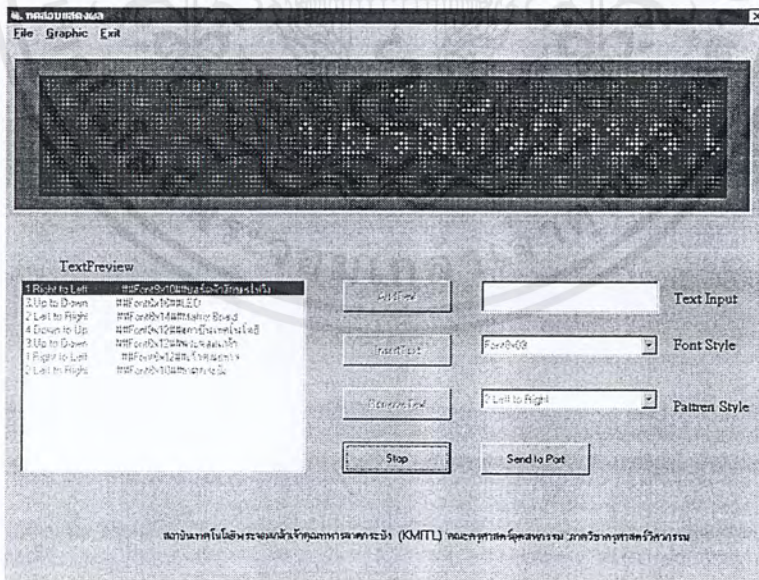
เมื่อกดที่ปุ่ม Preview ข้อความก็จะเริ่มแสดงที่ข้อความแรกต่อกันไปจนถึงข้อความสุดท้าย ดังรูปที่ 4.14



รูปที่ 4.14 เมื่อแสดงข้อความจนถึงข้อความสุดท้าย

เมื่อแสดงข้อความจนถึงข้อความสุดท้ายก็จะย้อนกลับมาแสดงที่ข้อความแรกใหม่ ดังรูป

รูปที่ 4.15

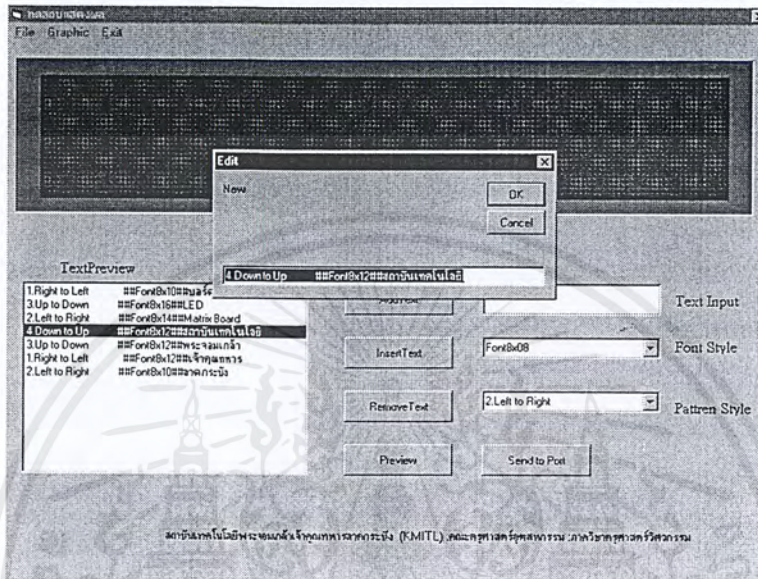


รูปที่ 4.15 การย้อนกลับมาแสดงที่ข้อความแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

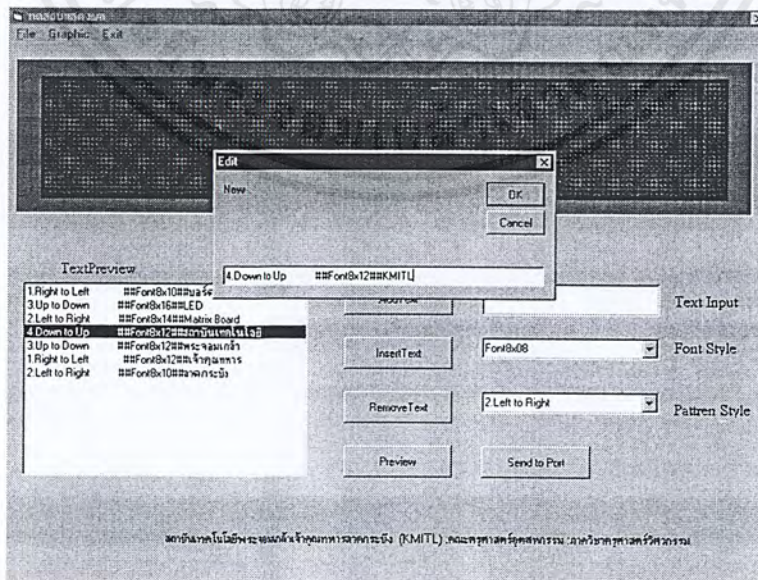
4.2.12 การทดลองแก้ไขข้อความที่อยู่ใน ช่องกำหนดการแสดงผลข้อความ (Text Preview)

การแก้ไขข้อความที่อยู่ภายใน Text Preview ทำได้โดยการดับเบิลคลิก เมาส์ (Mouse) ที่ข้อความที่ต้องการเปลี่ยน ดังรูปที่ 4.16



รูปที่ 4.16 วิธีการเปลี่ยนข้อความที่อยู่ภายใน Text Preview

เปลี่ยนข้อความโดยการแก้ไขข้อความแล้วกดที่ปุ่ม OK ดังรูปที่ 4.17

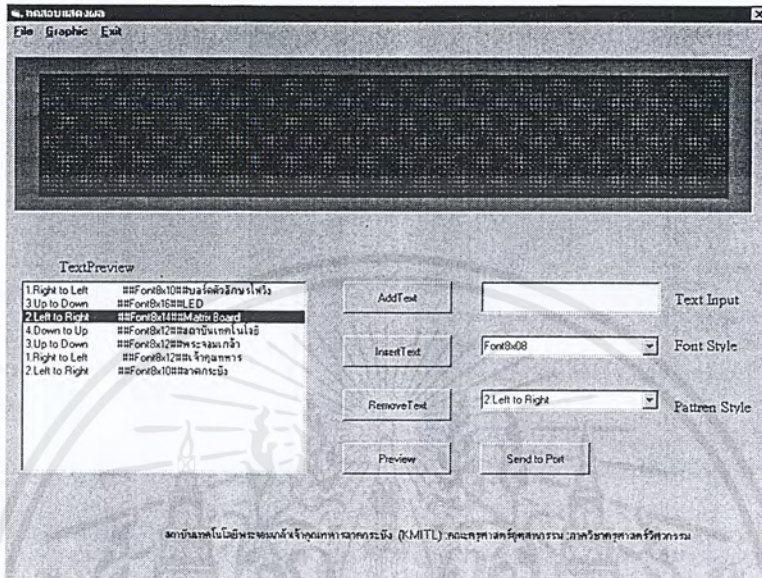


รูปที่ 4.17 การแก้ไขข้อความที่อยู่ภายใน Text Preview

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

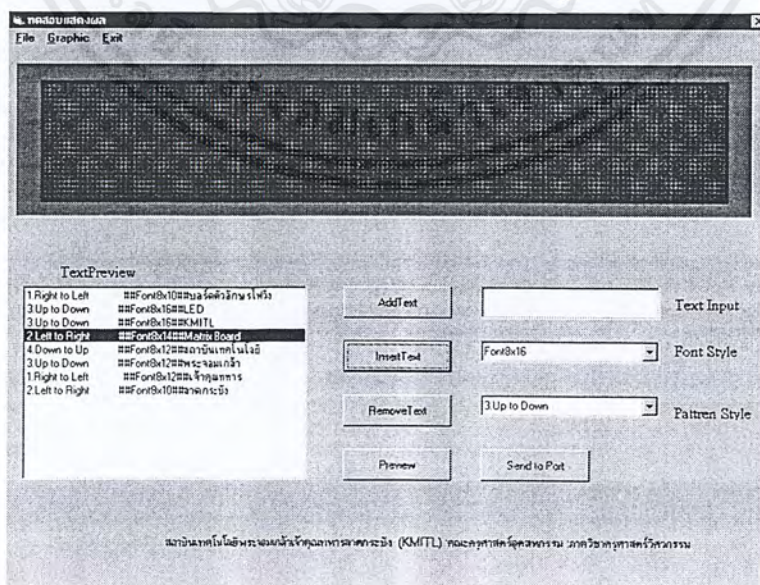
4.2.13 การทดลองแทรกข้อความที่อยู่ภายในช่องกำหนดข้อความ(Text Preview)

ใช้เมาส์คลิกที่ข้อความหลังข้อความที่เราต้องการจะแทรกให้เป็นแถบสี ดังรูปที่ 4.18



รูปที่ 4.18 การใช้เมาส์คลิกข้อความให้เป็นแถบสี

ป้อนข้อความที่ต้องการแทรก แล้วกดที่ปุ่ม Insert Text ข้อความก็จะเข้าไปปรากฏแทรกอยู่ใน Text Preview ดังรูปที่ 4.19

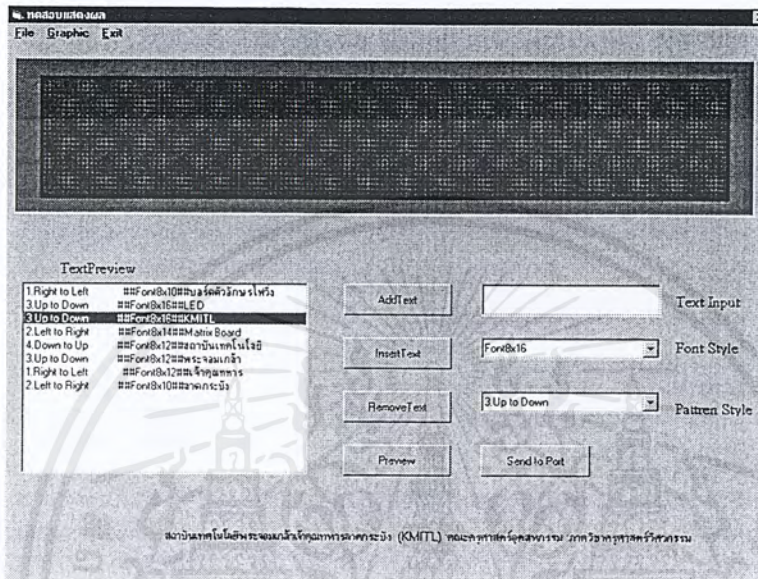


รูปที่ 4.19 ข้อความที่แทรกภายใน Text Preview

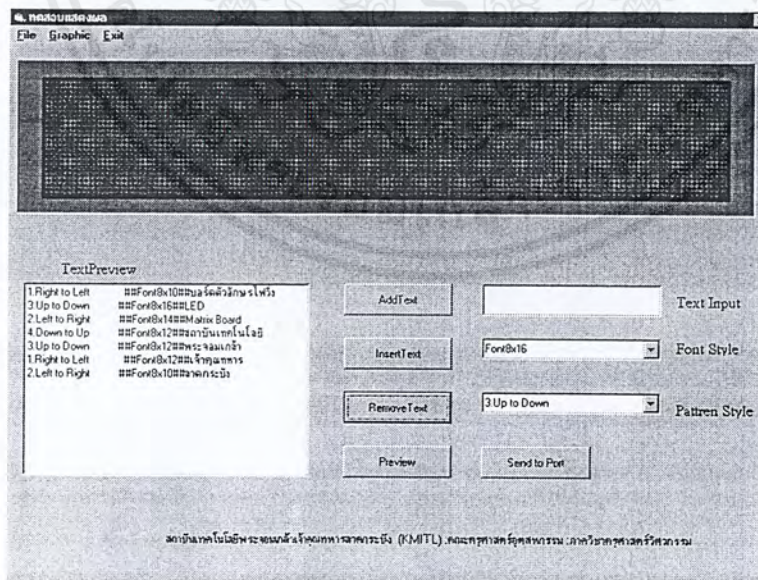
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.14 การทดลองลบข้อความ (Remove Text)

ใช้เมาส์คลิกที่ข้อความที่ต้องการจะลบให้เป็นแถบสี และกดที่ปุ่ม Remove Text ดังรูปที่ 4.20 และรูปที่ 4.21



รูปที่ 4.20 การใช้เมาส์คลิกที่ข้อความให้เป็นแถบสี

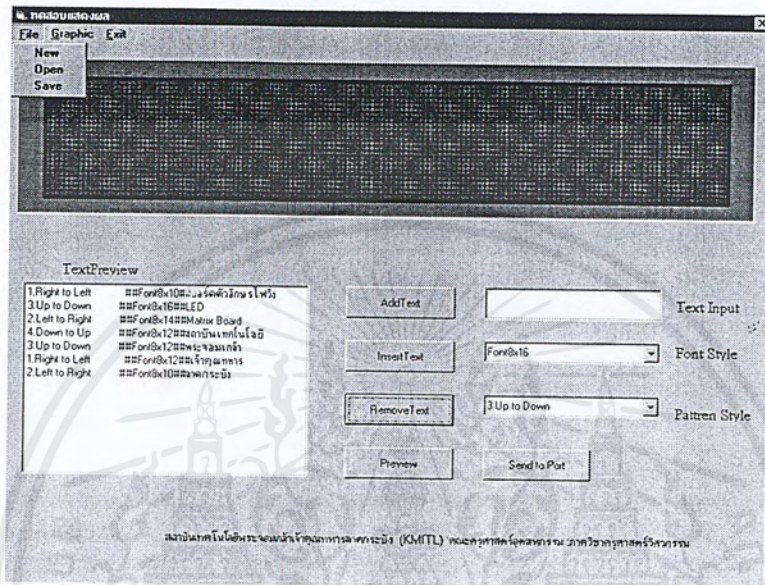


รูปที่ 4.21 การกดที่ปุ่ม Remove Text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

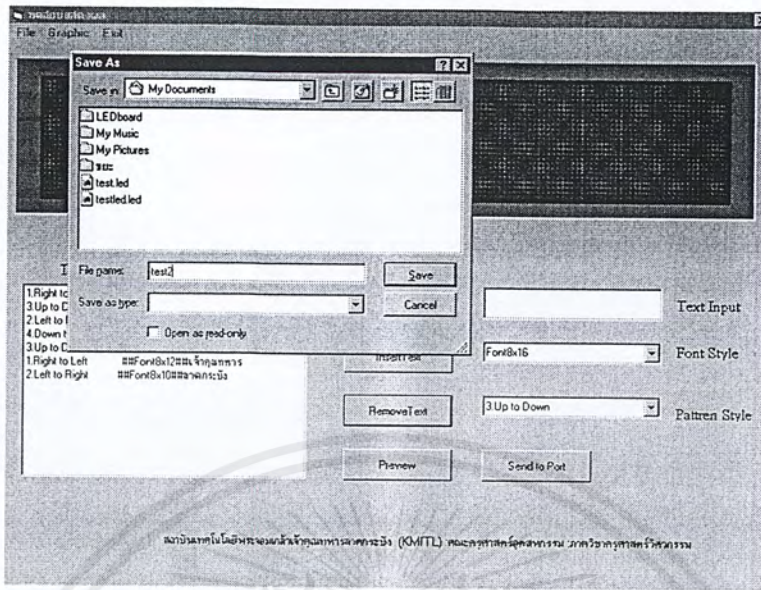
4.2.15 การทดลองบันทึกชุดข้อความ (Save)

เมื่อกำหนดชุดข้อความให้อยู่ภายใน Text Preview เรียบร้อยแล้วใช้เมาส์คลิกที่ เมนู File ดังรูปที่ 4.22



รูปที่ 4.22 ใช้เมาส์คลิกที่ เมนู File

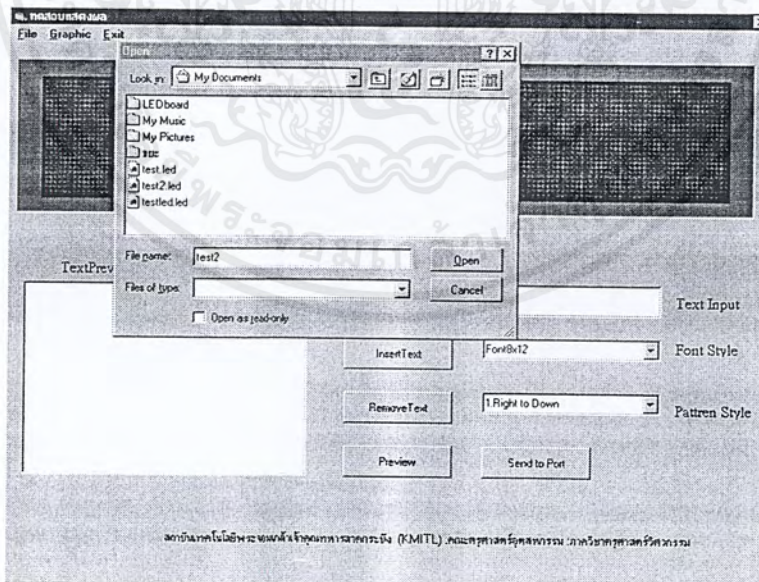
ใช้เมาส์เลือกคลิกที่ บันทึก (Save) เพื่อบันทึกชุดข้อความ ทำการบันทึกชุดข้อความ จะได้ไฟล์ (File) .led ดังรูปที่ 4.23



รูปที่ 4.23 ใช้เมาส์เลือกคลิกที่ บันทึก (Save)

4.2.16 การทดลองเรียกดูข้อความที่บันทึกมาใช้งาน (Open)

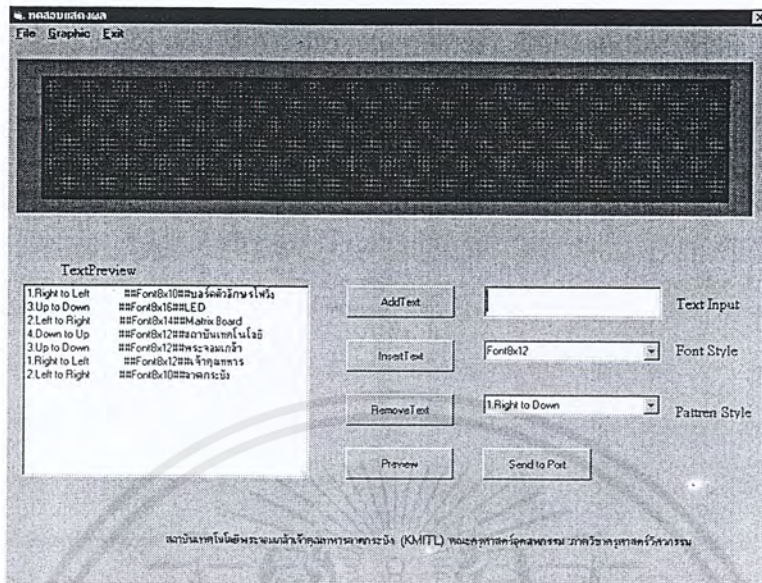
ใช้เมาส์คลิกเลือกที่ เมนู File เลือกที่ เปิด (Open) ทำการเลือกเปิด ไฟล์ .led ดังรูปที่ 4.24



รูปที่ 4.24 การเลือกเปิดไฟล์ .led

จะปรากฏข้อความเข้ามาอยู่ภายใน Text Preview ดังรูปที่ 4.25

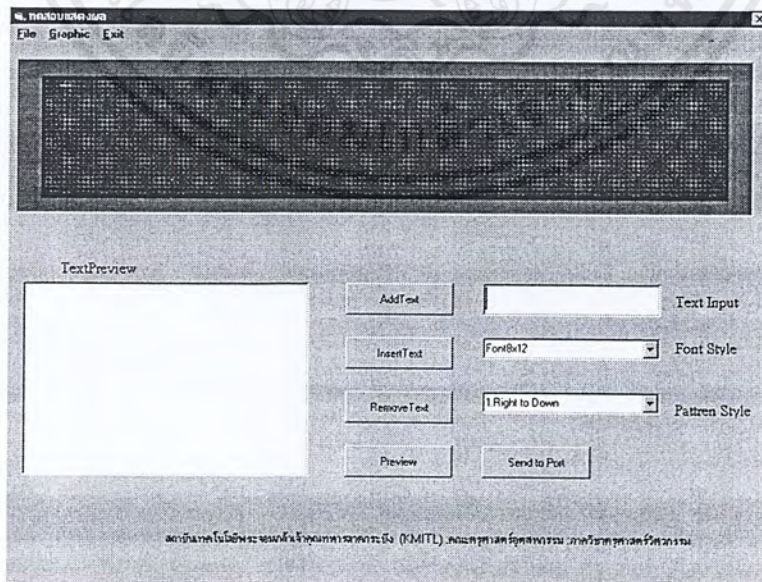
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 เมื่อเปิดไฟล์ .led

4.2.17 การทดลองสร้างชุดข้อความใหม่ (New)

ใช้เมาส์คลิกเลือกที่เมนู File เลือกที่ สร้างชุดข้อความใหม่ (New) จะได้ผลดังรูปที่ 4.26 คือชุดข้อความทั้งหมดที่อยู่ภายใน Text Preview จะหายไปพร้อมที่จะป้อนชุดข้อความใหม่



รูปที่ 4.26 พร้อมป้อนชุดข้อความใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองการใช้งานของแผงแสดงผล LED

การทดลองการใช้งานของแผงแสดงผลให้เป็นไปตามที่จะมีลักษณะและสภาวะการใช้งานในหลายๆ รูปแบบตามวิธีในการใช้งาน

4.3.1 การทดลองให้แผงแสดงผล LED พร้อมใช้งาน

จากรูปที่ 4.27 จะเป็นการแสดงสภาวะการเริ่มต้นการใช้งานในครั้งแรกของแผงแสดงผล LED เมื่อเราทำการเปิดสวิตช์ไฟ ในส่วนของหน่วยประมวลผลกลางจะทำการ Clear Display ก่อน แล้วจึงรอการรับการกดของ Keyboard



รูปที่ 4.27 สภาวะเริ่มต้นการใช้งานในครั้งแรก

4.3.2 การทดลองป้อนข้อมูลเป็นตัวอักษรภาษาไทยและภาษาอังกฤษ

การแสดงผลของตัวอักษรภาษาอังกฤษและภาษาไทยจะมีลักษณะและจำนวนของฟอนต์เหมือนกับภาคควบคุมการแสดงผลที่ ภาคโปรแกรมควบคุมการแสดงผลที่เครื่อง คอมพิวเตอร์

จากรูปที่ 4.28 แสดงลักษณะการแสดงผล ฟอนต์ขนาด 8x8 ซึ่งจะแสดงผลได้ทั้งข้อมูลที่เป็นภาษาอังกฤษและภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 การแสดงผล ฟอนต์ขนาด 8x8

จากรูปที่ 4.29 แสดงลักษณะการแสดงผล ฟอนต์ขนาด 8x10 ซึ่งจะแสดงผลได้ทั้งข้อมูลที่เป็นภาษาอังกฤษและภาษาไทย



รูปที่ 4.29 การแสดงผล ฟอนต์ขนาด 8x10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.30 แสดงลักษณะการแสดงผล ฟอนต์ขนาด 8×12 ซึ่งจะแสดงผลได้ทั้งข้อมูลที่เป็นภาษาอังกฤษและภาษาไทย



รูปที่ 4.30 การแสดงผล ฟอนต์ขนาด 8×12

จากรูปที่ 4.31 แสดงลักษณะการแสดงผล ฟอนต์ขนาด 8×14 ซึ่งจะแสดงผลได้ทั้งข้อมูลที่เป็นภาษาอังกฤษเพียงอย่างเดียว



รูปที่ 4.31 การแสดงผล ฟอนต์ขนาด 8×14

จากรูปที่ 4.32 แสดงลักษณะการแสดงผล ฟอนต์ขนาด 8×16 ซึ่งจะแสดงผลได้ทั้งข้อมูลที่เป็นภาษาอังกฤษเพียงอย่างเดียว



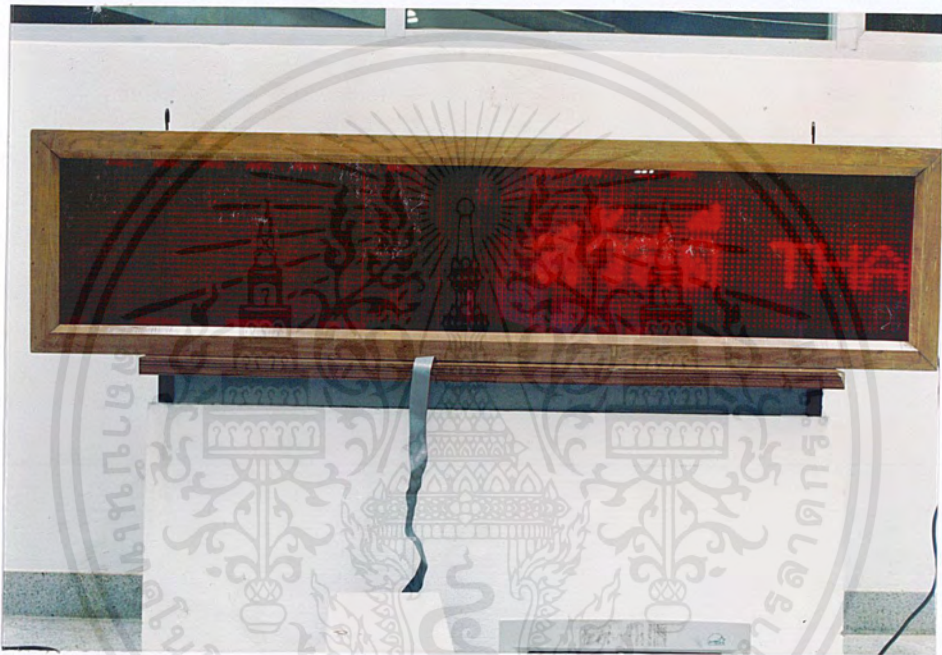
รูปที่ 4.32 การแสดงผล ฟอนต์ขนาด 8×16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การทดลองการแสดงผลการเคลื่อนที่ของตัวอักษร

เมื่อทำการป้อนข้อมูลและกำหนดทิศทางการเคลื่อนที่ของตัวอักษรแล้ว เมื่อต้องการให้แสดงผลให้กดที่ปุ่ม ENTER ซึ่งสามารถแบ่งเป็นหมวดลักษณะการแสดงผลได้ 4 รูปแบบ

จากรูปที่ 4.33 แสดงลักษณะการแสดงผลของข้อความที่เคลื่อนจากทางขวาไปซ้ายของบอร์ดอักษรไฟวิ่ง



รูปที่ 4.33 การเคลื่อนที่ของข้อความจากทางขวาไปซ้าย

จากรูปที่ 4.34 แสดงลักษณะการแสดงผลของข้อความที่เคลื่อนจากทางซ้ายไปขวาของบอร์ดอักษรไฟวิ่ง



รูปที่ 4.34 การเคลื่อนที่ของข้อความจากทางซ้ายไปขวา

จากรูปที่ 4.35 แสดงลักษณะการแสดงผลของข้อความที่เคลื่อนจากทางบนลงล่างของบอร์ด
อักษรไฟวิ่ง



รูปที่ 4.35 การเคลื่อนที่ของข้อความจากบนลงล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.36 แสดงลักษณะการแสดงผลของข้อความที่เคลื่อนจากทางล่างขึ้นบนของ บอร์ดอักษรไฟวิ่ง

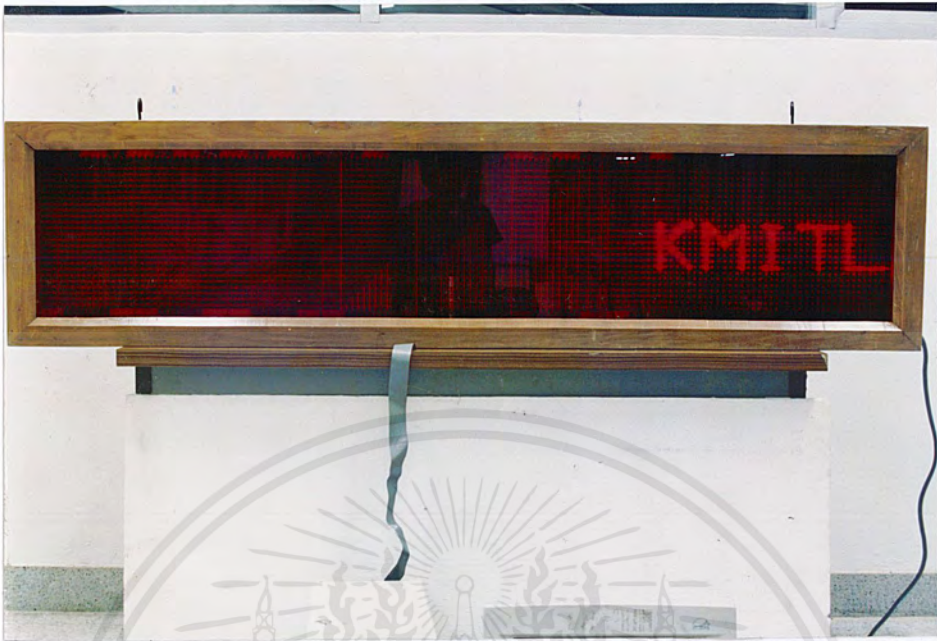


รูปที่ 4.36 การเคลื่อนที่ของข้อความจากล่างขึ้นบน

4.3.4 การทดลองการควบคุมการแสดงผลผ่านทางโปรแกรมควบคุมที่เครื่องคอมพิวเตอร์

เมื่อกำหนดขนาดของ ฟอนต์และกำหนดลักษณะการแสดงผล ที่แสดงทางโปรแกรมจำลอง การแสดงผลที่เครื่อง คอมพิวเตอร์เสร็จแล้ว ใช้เมาส์คลิกที่ปุ่ม “Sent to Port” ที่อยู่ในหน้าต่าง โปรแกรมโปรแกรมก็จะทำการส่งข้อมูล มาให้ที่ตัว บอร์ดอักษรไฟวิ่ง บอร์ดอักษรไฟวิ่งก็จะแสดงผล

จากรูปที่ 3.37 แสดงลักษณะการแสดงผลเมื่อใช้คำสั่ง Right to Left ตัวอักษรจะมีการเลื่อน จากทางด้าน ขวามาทางด้านซ้ายของ บอร์ดอักษรไฟวิ่ง



รูปที่ 4.37 การเคลื่อนที่ของข้อความเมื่อใช้คำสั่ง Right to Left

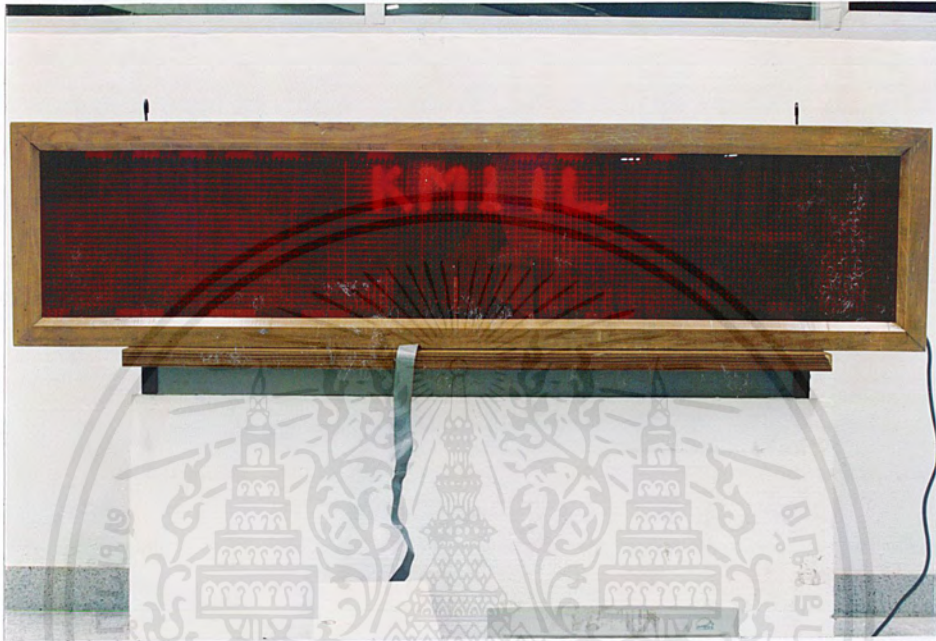
จากรูปที่ 3.38 แสดงลักษณะการแสดงผลเมื่อใช้คำสั่ง Left to Right ตัวอักษรจะมีการเคลื่อนจากทางด้านซ้ายมาทางด้านขวาของ บอร์ดอักษรไฟวิ่ง



รูปที่ 4.38 การเคลื่อนที่ของข้อความเมื่อใช้คำสั่ง Left to Right

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.39 แสดงลักษณะการแสดงผลเมื่อใช้คำสั่ง Up to Down ตัวอักษรจะมีการเลื่อนจากด้านบนลงด้านล่าง ของ บอร์ดอักษรไฟวิ่ง



รูปที่ 4.39 การเคลื่อนที่ของข้อความเมื่อใช้คำสั่ง Up to Down

จากรูปที่ 4.40 แสดงลักษณะการแสดงผลเมื่อใช้คำสั่ง Down to Up ตัวอักษรจะมีการเลื่อนจากด้านล่างขึ้นด้านบน ของ บอร์ดอักษรไฟวิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.40 การเคลื่อนที่ของข้อความเมื่อใช้คำสั่ง Down to Up

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปปัญหาและแนวทางการแก้ไขปัญหา

5.1 บทสรุป

การสร้างปริญญาณิพนธ์ฉบับนี้จะได้รับประโยชน์ดังต่อไปนี้

1. จากการทดลองของโครงการนี้ ทำให้เราทราบถึง เทคนิคในการเขียนโปรแกรมภาษาแอสเซมบลีและสามารถประยุกต์คำสั่ง ของโปรแกรมต่างๆ ให้สามารถแสดงผลได้หลายรูปแบบทำให้สามารถ นำความรู้ไปใช้ในการเรียนการสอนหรืองานที่เกี่ยวข้องกับไมโคร โปรเซสเซอร์
2. การติดต่อสื่อสารกับเครื่อง คอมพิวเตอร์พีซี โดยโปรแกรม Visual Basic ซึ่งมีประโยชน์ในการนำไปประยุกต์ใช้งานจริงได้
3. ทราบถึงคุณสมบัติและลักษณะพิเศษของ ไคโอคเปล่งแสง, ไมโครคอนโทรลเลอร์ ซึ่งสามารถนำไปประยุกต์ใช้งานต่อไปได้
4. สามารถนำโครงการนี้ ไปใช้งานได้จริงเพื่อประชาสัมพันธ์และบอกข่าวสาร ประกาศ เรื่องต่างๆให้นักศึกษาในคณะทราบ ซึ่งเป็นประโยชน์ต่อ คณะครุศาสตร์อุตสาหกรรม ด้วยรูปแบบการแสดงผลทั้งภาษาไทยและภาษาอังกฤษรวมถึงรูปภาพกราฟิกลักษณะต่างๆได้

5.2 ปัญหาและอุปสรรค

1. ปัญหาการเขียนโปรแกรมทดสอบให้ไคโอคเปล่งแสง คิดเรียงแถวทางแนวตั้งแล้วไม่แสดงผลตามที่ได้เขียนโปรแกรมสั่ง
2. ในการเขียนและพัฒนาโปรแกรม อีพรอม อีมูเลเตอร์ (EPROM Emulator) แทน อีพรอม (EPROM) จริงที่ใช้ในแผ่นวงจรพิมพ์ ทำให้การเครื่องย้ายต้องถอด อีพรอม อีมูเลเตอร์ ออกเสียก่อนเป็นผลทำให้ ซ็อกเก็ต (Socket) ของ อีพรอม ในแผ่นวงจรหลวมทำให้การสื่อสารข้อมูลผิดพลาด
3. การเขียนโปรแกรมทดสอบให้ ไคโอคเปล่งแสงคิดเรียงแถวแนวตั้ง จะมี ไคโอคเปล่งแสง บางแถวไม่ติด หรือมีการกระพริบและกระตุก ทำให้การแสดงผลไม่สามารถทำได้อย่างสมบูรณ์
4. เมื่อเขียนโปรแกรมคำสั่งให้ไคโอคเปล่งติดแถวเดียวทางแนวนอนเป็นเวลานานๆ ทรานซิสเตอร์จะร้อนมากทำให้เกิดการเสียหายได้
5. การเขียนโปรแกรมจำลองผลการทำงานของ บอร์ดอักษรไฟวิ่ง โดยโปรแกรม Visual Basic เมื่อทดสอบ ป้อนข้อความยาวๆแล้วสั่งให้ข้อความเคลื่อนที่ ข้อความจะวิ่งช้าลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การแก้ไข

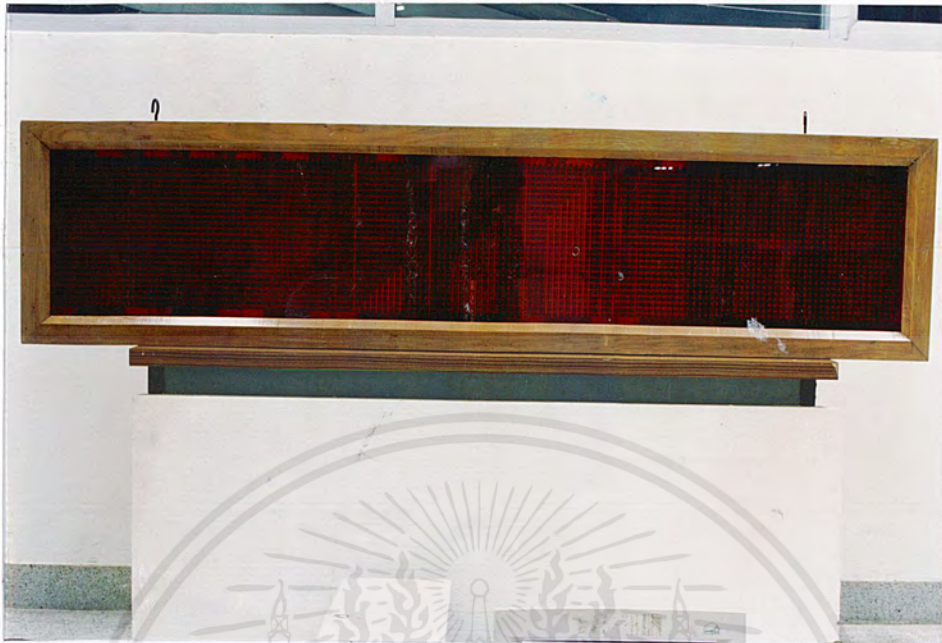
1. ตรวจสอบวงจรและตัวอุปกรณ์แล้วพบว่า ไมโครโปรเซสเซอร์เบอร์ Z80180 ที่เสียบใน ซ็อกเก็ต เกิดการหลวม จึงทำการถอดไมโครโปรเซสเซอร์เบอร์ Z80180 ออกจาก ซ็อกเก็ต แล้วทำการใส่เข้าไปใหม่จึงสามารถแสดงผลได้ตามปรกติ
2. ตรวจสอบเช็คสายของ อีพროม อิมูเลเตอร์ และ ซ็อกเก็ต ของ อีพროม ในแผ่นวงจรให้แน่นยิ่งขึ้นเพื่อให้การสื่อสารข้อมูลถูกต้อง
3. เช็คสาย คอนเน็คต์ (Connect) ที่ต่ออยู่กับบอร์ดของไดโอดเปล่งแสง ปรากฏว่าขาเสียบ Connect เกิดการเสียหาย ทำการบัดกรี ขาเสียบ Connect ใหม่และทำการเปลี่ยนสาย คอนเน็คต์ บางส่วน
4. หลีกเลี่ยงการเขียน โปรแกรมสั่งให้ ไดโอดเปล่งแสง ติดทางแนวนอนเป็นเวลานานๆ
5. การใช้งาน โปรแกรมควรถัดตั้งในเครื่องคอมพิวเตอร์ที่มีความเร็ว ซีพียู ตั้งแต่ 300 MHz ขึ้นไป

5.4 แนวทางในการพัฒนา

1. ควรมีการพัฒนาโปรแกรมให้มีการใช้ อินเตอร์รัพท์ (Interrupt) ภายในได้ เพื่อลดปัญหาที่อาจเกิดขึ้นจากการ อินเตอร์รัพท์ ภายนอก เพราะจะทำให้การแสดงผลเป็นไปอย่างสมบูรณ์
2. รูปแบบของกล่องควรวางตัวที่มีน้ำหนักเบา เพื่อง่ายต่อการขนย้ายและทำการติดตั้ง
3. ควรมี ฟอนต์ (Font) ตัวอักษรหลายๆรูปแบบให้เลือกใช้งานมากกว่าที่มีอยู่
4. การจำแสดงผลการจำลองการแสดงผลโดยโปรแกรมที่เครื่อง คอมพิวเตอร์พีซี การหาวิธีการแสดงผลที่เมื่อพิมพ์ข้อความยาวๆแล้ว ข้อความไม่ถูกหน้าต่างให้ช้าลง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 รูปเครื่องต้นแบบด้านหน้า



รูปที่ ก.2 รูปเครื่องต้นแบบด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการทำงานทางด้านวิซวลเบสิก

```

Dim FontFile As String, PointStart As Integer      'ตัวแปรในการเรียกใช้ฟรอนท์
Dim Char As String * 1                          'เก็บข้อมูลสำหรับเช็ค ASCII
Dim CharTemp As String * 1
Dim TextNowShow As String                       'ตัวแปรสำหรับเก็บข้อความที่กำลังแสดง
Dim CountNowShow, CountChar As Integer         'ตัวแปรใช้ในการนับข้อความที่ต้องการแสดง
ผล
Dim Mix As Boolean
Dim PatType As Integer
Dim FontStye As String
Dim TableSelect, UndoIndex, TextIndex, RanLED As Integer
*****เรียกใช้
font*****
Private Sub Combo1_Click()
    FontFile = Trim(Combo1.Text) & ".dat"
End Sub
Private Sub Combo2_Click()
    PatType = Val(Combo2.ListIndex) + 1
End Sub

Private Sub FontLoad()
    FontFile = Trim(FontStye) & ".dat"
End Sub

Private Sub Command1_Click()
    Dim A As Integer, Counter As Integer
    If Timer1.Enabled = False Then
        PatType = Left$(list1.List(0), 1)
        Select Case PatType
            Case 1
                PointStart = 144
            Case 2
                TextNowShow = list1.List(list1.ListIndex)
                PointStart = (144 - (Len(TextNowShow) * 8))
                PointStart = 0
            Case 3
                PointStart = -23
            Case 4
                PointStart = 23
        End Select
        CountNowShow = 0
        list1.ListIndex = CountNowShow
        TextNowShow = list1.List(list1.ListIndex)
        Timer1.Enabled = True
        Command1.Caption = "Stop"      'เปลี่ยนสถานะการกดปุ่ม Preview
        Text1.Enabled = False          'ยกเลิกสถานะของ TextInput
        Combo1.Enabled = False        'ยกเลิกสถานะของการเลือกฟรอนท์
        Combo2.Enabled = False        'ยกเลิกสถานะของpattren
    
```

```

Command2.Enabled = False
Command3.Enabled = False
Command4.Enabled = False
list1.Enabled = False
Else
Timer1.Enabled = False
Command1.Caption = "Preview"
Text1.Enabled = True
Combo1.Enabled = True
Combo2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
Command2.Enabled = True
list1.Enabled = True
Picture1.Cls
End If
End Sub

```

ได้

เมื่อมีการกดปุ่ม preview จะคืนสถานะเดิม
 'TextInput,Fontstyle,pattren ให้สามารถใช้งานได้

ฟังก์ชันสำหรับตรวจคำที่รับเข้ามาแล้วตัดคำครั้งละ 1 ตัวอักษร เพื่อนำไปเปรียบเทียบกับข้อมูลใน
 File ตัวอักษร

Sub ShowLED(Char As String, X As Integer, Y As Integer, Index As Integer)

```

Dim Temp(0 To 23) As Integer, I, j, k As Integer
Dim DataVal As Employee
Dim DataValTemp As Employee
Dim Temp2(0 To 23) As String * 8, CH As String * 1
Open (App.Path & "\" & FontFile) For Random As #1 Len = Len(DataVal)
Get #1, Asc(Char), DataVal
For I = 0 To 23
If Val("&H" & DataVal.Num(I)) = 0 Then
Temp(I) = "&H00"
Else
Temp(I) = "&H" & Hex((Val("&H" & DataVal.Num(I))))
End If
Next I
Close #1
'แปลงเลขฐาน 16 เป็นฐาน 2
For I = 0 To 23
For j = 0 To 7
k = Int(Temp(I) Mod 2)
Temp(I) = Int(Temp(I) / 2)
Temp2(I) = k & Temp2(I)
Next j
Next I
For I = 0 To 23
For j = 1 To 8
CH = Mid(Trim(Temp2(I)), j, 1)
If CH = 1 Then
LED (X + ((Index - 1) * 8) + j) * 5, (Y + I) * 5, 10
End If

```

```

    Next j
  Next I
End Sub
Function LED(X As Double, Y As Integer, C As Integer)
  Picture1.Circle (X, Y), 1, QBColor(C)
End Function

Private Sub Command2_Click() 'AddText
If Len(Trim(Text1.Text)) <> "" Then
  list1.AddItem (Combo2.List(Combo2.ListIndex) & " ###" &
(Combo1.List(Combo1.ListIndex) &
  "###" & Text1.Text))
End If
Text1.Text = ""
End Sub

Private Sub Command3_Click() 'Insert
TableSelect = list1.ListIndex
If Text1.Text <> "" Then
  If TableSelect = -1 Then
    list1.AddItem (Combo2.List(Combo2.ListIndex) & " ###" & (Combo1.List
(Combo1.ListIndex) &
    "###" & Text1.Text)), TableSelect + 1
  Else
    list1.AddItem (Combo2.List(Combo2.ListIndex) & " ###" & (Combo1.List
(Combo1.ListIndex) &
    "###" & Text1.Text)), TableSelect
  End If
End If
Text1.Text = ""
End Sub

Private Sub Command4_Click() 'Remove
  TableSelect = list1.ListIndex
  UndoText = list1.List(TableSelect)
  UndoIndex = TableSelect
If list1.ListCount > 0 And TableSelect >= 0 Then
  list1.RemoveItem (TableSelect)
  TableSelect = -1
End If
End Sub

Private Sub Form_Load()
FontFile = Trim(FontStyle) & ".dat"
End Sub

Private Sub list1_Db1Click() 'dubble Click Edit text in textprevice
TableSelect = list1.ListIndex
newtext = (InputBox("New", "Edit", list1.Text))
If newtext <> "" Then
  list1.AddItem (newtext), list1.ListIndex

```

```

list1.RemoveItem (list1.ListIndex)
End If
End Sub

Private Sub mExit_Click()
End
End Sub

Private Sub mNew_Click()
list1.Clear
End Sub

Private Sub mNewGraphic_Click()
Form1.Enabled = False
NewGraph.Show
End Sub

Private Sub mOpen_Click()
Dim DataOpen As Employee2
Dim I As Integer
CommonDialog1.ShowOpen
If Not Cancel Then
Open CommonDialog1.FileName For Random As #5
I = 1
list1.ListIndex = -1
Do
Get #5, I, DataOpen
list1.List(list1.ListIndex) = Trim(DataOpen.DataString)
I = I + 1
Loop While Not EOF(5)
Close #5
End If
End Sub

Private Sub mSave_Click()
Dim DataSave As Employee2
Dim I As Integer
CommonDialog1.ShowSave
If Not Cancel Then
Kill CommonDialog1.FileName
Open CommonDialog1.FileName For Random As #5
For I = 1 To list1.ListCount
DataSave.DataString = list1.List(I - 1)
Put #5, I, DataSave
Next I
Close #5
End If
End Sub

Private Sub SendPort_Click()
Dim I As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    I = I + 1
  Else
    ShowLED Char, (144 - (Len(TextNowShow) * 8)) / 2, PointStart, A - I
  End If
  A = A + 1
Loop
If PointStart = 1 Then
  For I = 1 To 2000
    Picture1.Refresh
  Next I
End If
PointStart = PointStart - 1
If PointStart < -24 Then
  PointStart = 23
  CountNowShow = CountNowShow + 1
  If CountNowShow >= list1.ListCount Then
    CountNowShow = 0
  End If
  list1.ListIndex = CountNowShow
  TextNowShow = list1.List(list1.ListIndex)
  PatType = Left$(list1.List(list1.ListIndex), 1)
  CheckPatType PatType
End If
  Case Else
End Select
End Function

Function CheckPatType(pat As Integer)
Select Case pat
Case 1
  PointStart = 144
Case 2
  PointStart = 0
Case 3
  PointStart = -23
Case 4
  PointStart = 23
Case Else
End Select
End Function

```

รูปที่ ข.1 โปรแกรม วิชาลเบสิก

โปรแกรมควบคุมการทำงานหลัก

000001 0000	
000002 0001	.EQU CNTLAA0,1
000003 0003	.EQU CNTLB0,3
000004 0005	.EQU STAT0,5
000005 0007	.EQU TDR0,7
000006 0009	.EQU RDR0,9
000007 000C	.EQU NDSP,12
000008 0000	
000009 0010	.EQU TCR,10H
000010 000C	.EQU TMDROL,0CH
000011 000D	.EQU TMDROH,0DH
000012 000E	.EQU RLDROL,0EH
000013 000F	.EQU RLDROH,0FH
000014 00FF	.EQU DELAYSCAN,0FFH
000015 0000	.EQU DINTSCAN,0
000016 0011	.EQU EINTSCAN,11H
000017 0000	
000018 80FF	.EQU SYSSTACK,80FFH
000019 0000	
000020 0014	.EQU DELAYMIN,20
000021 03E8	.EQU DELAYMID,1000
000022 0000	.EQU DELAYMAX,0
000023 008C	.EQU PORTROW,8CH
000024 009D	.EQU INKEY,9DH
000025 009E	.EQU CLRKEY,9EH
000026 008F	.EQU CTPORT1,8FH
000027 009F	.EQU CTPORT2,9FH
000028 0000	
000029 0003	.EQU OVER_COL,3
000030 0018	.EQU MAXROW,24
000031 0012	.EQU MAXCOL,144/8
000032 01B0	.EQU PAGESIZE,MAXCOL*MAXROW
000033 0018	.EQU CHARSIZE,24
000034 0012	.EQU ROWBYTE,18
000035 0000	
000036 0038	.EQU CBR,38H
000037 0039	.EQU BBR,39H
000038 003A	.EQU CBAR,3AH
000039 0000	
000040 0001	.EQU FONTCOMM,01
000041 0002	.EQU FONTCHAR,02
000042 0001	.EQU MODEMOVE,01
000043 0002	.EQU MODEEDIT,02
000044 0000	
000045 001B	.EQU ESC,27
000046 0004	.EQU DEL,04
000047 000D	.EQU ENTER,13

```

000048 001E      .EQU KLEFT,30
000049 001A      .EQU KRIGHT,26
000050 0018      .EQU KUP,24
000051 0019      .EQU KDN,25
000052 0002      .EQU KHOME,2
000053 0005      .EQU KEND,5
000054 0003      .EQU KPGUP,3
000055 0006      .EQU KPGDN,6
000056 0008      .EQU KBKSP,8
000057 0000
000058 0000
000059 0000
000060 0000
000061 0000      ;POWER UP
000062 0000 210000      START: ld hl,0
000063 0003 00      START1: nop
000064 0004 2B      dec hl
000065 0005 7C      ld a,h
000066 0006 B5      or l
000067 0007 20FA      jr nz,start1
000068 0009      INIT:
000069 0009 F3      di
000070 000A 3E00      ld a,0
000071 000C ED3938      out0 (38h),a
000072 000F ED3939      out0 (39h),a
000073 0012 ED393A      out0 (3ah),a
000074 0015 31FF80      ld sp,80ffh
000075 0018 3E80      ld a,80h
000076 001A D38F      out (8fh),a
000077 001C 3E82      ld a,82h
000078 001E D39F      out (9fh),a
000079 0020 C39900      jp main
000080 0023
000081 0038      .ORG 0038H
000082 0038      SCANDISP:
000083 0038 F3      di
000084 0039 F5      push af
000085 003A C5      push bc
000086 003B D5      push de
000087 003C E5      push hl
000088 003D DDE5      push ix
000089 003F FDE5      push iy
000090 0041 C36800      jp conti
000091 0066      .ORG 0066h
000092 0066 ED45      retn
000093 0068 2AB189      CONTI: ld hl,(addscan)
000094 006B 0612      ld b,18
000095 006D 0E11      ld c,17
000096 006F 3EFF      ld a,0ffh
000097 0071 D38C      out (portrow),a
000098 0073 7E      SCAN1: ld a,(hl)

```

000099 0074 2F	cpl
000100 0075 ED79	out (c),a
000101 0077 23	inc hl
000102 0078 0D	dec c
000103 0079 10F8	djnz scan1
000104 007B 3AB089	ld a,(rowscan)
000105 007E 3D	dec a
000106 007F D38C	out (portrow),a
000107 0081 2005	jr nz,scan2
000108 0083 3E18	ld a,24
000109 0085 214081	ld hl,pagex
000110 0088 32B089	SCAN2: ld (rowscan),a
000111 008B 22B189	ld (addscan),hl
000112 008E	ENDSCAN:
000113 008E FDE1	pop iy
000114 0090 DDE1	pop ix
000115 0092 E1	pop hl
000116 0093 D1	pop de
000117 0094 C1	pop bc
000118 0095 F1	pop af
000119 0096 FB	ei
000120 0097 ED4D	reti
000121 0099	MAIN:
000122 0099 3E18	ld a,24
000123 009B 32B089	ld (rowscan),a
000124 009E 214081	ld hl,pagex
000125 00A1 22B189	ld (addscan),hl
000126 00A4 3EFF	ld a,0ffh
000127 00A6 D38C	out (portrow),a
000128 00A8 ED56	im 1
000129 00AA FB	ei
000130 00AB CDDA04	call clearall
000131 00AE 21DC05	ld hl,1500
000132 00B1 22BF89	ld (crmprg),hl
000133 00B4 CDFA00	call resetkey
000134 00B7 DD21EA05	ld ix,prg1+4
000135 00BB 214081	ld hl,pagex
000136 00BE 22C389	ld (pageput),hl
000137 00C1 CDFC04	call writestr
000138 00C4	;ld ix,prg2+4
000139 00C4 21F082	ld hl,pagey
000140 00C7 22C389	ld (pageput),hl
000141 00CA CDFC04	call writestr
000142 00CD CDE502	call writemes
000143 00D0 3AE98C	ld a,(boot)
000144 00D3 CD1E02	STILL: call editprg
000145 00D6 CD3303	RUNRUN: call runmode
000146 00D9 CDA205	call delaym
000147 00DC 18F5	jr still
000148 00DE	
000149 00DE	INITPRG:

```

000150 00DE 21CD05      ld hl,orgprg
000151 00E1 11C589      ld de,mprg
000152 00E4 010C00      ld bc,12
000153 00E7 EDB0        ldir
000154 00E9 110800      ld de,8
000155 00EC 21C589      ld hl,mprg
000156 00EF 22BB89      ld (stprg),hl
000157 00F2 22B989      ld (ptprg),hl
000158 00F5 19          add hl,de
000159 00F6 22BD89      ld (endprg),hl
000160 00F9 C9          ret
000161 00FA          RESETKEY:
000162 00FA 3E01          ld a,01h
000163 00FC D39E          out (clrkey),a
000164 00FE 00          nop
000165 00FF 00          nop
000166 0100 00          nop
000167 0101 00          nop
000168 0102 00          nop
000169 0103 CD9205      call dela
000170 0106 CD0A01      call clearkey
000171 0109 C9          ret
000172 010A          CLEARKEY:
000173 010A 3E00          ld a,0
000174 010C D39E          out (clrkey),a
000175 010E 00          nop
000176 010F 00          nop
000177 0110 00          nop
000178 0111 3E80          ld a,80h
000179 0113 D39E          out (clrkey),a
000180 0115 C9          ret
000181 0116          GETKEY:
000182 0116 CD0A01      call clearkey
000183 0119          LPGETKEY:
000184 0119 DB9D          in a,(inkey)
000185 011B FE00          cp 0
000186 011D 28FA          jr z,lpgetkey
000187 011F FEF8          cp 0f8h
000188 0121 28F6          jr z,lpgetkey
000189 0123 C9          ret
000190 0124          FLEFT:
000191 0124 2AB989      ld hl,(ptprg)
000192 0127 7E          ld a,(hl)
000193 0128 FE01          cp 1
000194 012A          ;jr z,lpedit
000195 012A 110400      ld de,4
000196 012D ED52          sbc hl,de
000197 012F 22B989      ld (ptprg),hl
000198 0132 C31E02          jp lpedit
000199 0135          FRIGHT:
000200 0135 2AB989      ld hl,(ptprg)

```

000201 0138 7E	ld a,(hl)
000202 0139 FE02	cp 2
000203 013B CA1E02	jp z,lpedit
000204 013E 110400	ld de,4
000205 0141 19	add hl,de
000206 0142 22B989	ld (ptprg),hl
000207 0145 C31E02	jp lpedit
000208 0148	FEND:
000209 0148 2ABD89	ld hl,(endprg)
000210 014B 110400	ld de,4
000211 014E ED52	sbc hl,de
000212 0150 22B989	ld (ptprg),hl
000213 0153 D1	pop de
000214 0154 E1	pop hl
000215 0155 C9	ret
000216 0156	INCPTPRG:
000217 0156 E5	push hl
000218 0157 D5	push de
000219 0158 2AB989	ld hl,(ptprg)
000220 015B 110400	ld de,4
000221 015E 19	add hl,de
000222 015F 22B989	ld (ptprg),hl
000223 0162 D1	pop de
000224 0163 E1	pop hl
000225 0164 C9	ret
000226 0165	DECPTPRG:
000227 0165 E5	push hl
000228 0166 D5	push de
000229 0167 2AB989	ld hl,(ptprg)
000230 016A 110400	ld de,4
000231 016D ED52	sbc hl,de
000232 016F 22B989	ld (ptprg),hl
000233 0172 D1	pop de
000234 0173 E1	pop hl
000235 0174 C9	ret
000236 0175	FBKSP:
000237 0175 2AB989	ld hl,(ptprg)
000238 0178 7E	ld a,(hl)
000239 0179 FE01	cp 01
000240 017B CA1E02	jp z,lpedit
000241 017E CD8C02	call deletechar
000242 0181 CD6501	call decptprg
000243 0184 C31E02	jp lpedit
000244 0187	PUTCOMM:
000245 0187 3E03	ld a,03
000246 0189 32E78C	ld (ascii),a
000247 018C 3E01	ld a,fontcomm
000248 018E 32E88C	ld (font),a
000249 0191 2AB989	ld hl,(ptprg)
000250 0194 7E	ld a,(hl)
000251 0195 FE02	cp 2

000252 0197 CA1502	jp z,endputcomm
000253 019A 110400	ld de,4
000254 019D 19	add hl,de
000255 019E 22B989	ld (ptprg),hl
000256 01A1 CDBA02	call insertchar
000257 01A4 2ABD89	ld hl,(endprg)
000258 01A7 19	add hl,de
000259 01A8 22BD89	ld (endprg),hl
000260 01AB 214081	ld hl,pagex
000261 01AE CD5B05	call clearpg
000262 01B1 CDE502	call writemes
000263 01B4 CD1601	call getkey
000264 01B7 FE18	cp kup
000265 01B9 2008	jr nz,noup
000266 01BB 3E04	ld a,4
000267 01BD 32E78C	ld (ascii),a
000268 01C0 C3E401	jp putc1
000269 01C3	NOUNP:
000270 01C3 FE19	cp kdn
000271 01C5 2008	jr nz,nodn
000272 01C7 3E05	ld a,5
000273 01C9 32E78C	ld (ascii),a
000274 01CC C3E401	jp putc1
000275 01CF	NODN:
000276 01CF FE1E	cp kleft
000277 01D1 20FC	jr nz,nodn
000278 01D3 3E06	ld a,6
000279 01D5 32E78C	ld (ascii),a
000280 01D8 C3E401	jp putc1
000281 01DB	NOLF:
000282 01DB FE1A	cp kright
000283 01DD 2015	jr nz,nort
000284 01DF 3E07	ld a,7
000285 01E1 32E78C	ld (ascii),a
000286 01E4	PUTC1:
000287 01E4 DD2AB989	ld ix,(ptprg)
000288 01E8 DD7700	ld (ix+0),a
000289 01EB 3AE88C	ld a,(font)
000290 01EE DD7701	ld (ix+1),a
000291 01F1 C31502	jp endputcomm
000292 01F4	NORT:
000293 01F4 CD8C02	call deletechar
000294 01F7 2AB989	ld hl,(ptprg)
000295 01FA 110400	ld de,4
000296 01FD ED52	sbc hl,de
000297 01FF 22B989	ld (ptprg),hl
000298 0202 2ABD89	ld hl,(endprg)
000299 0205 110400	ld de,4
000300 0208 ED52	sbc hl,de
000301 020A 22B989	ld (ptprg),hl
000302 020D 2ABD89	ld hl,(endprg)

```

000303 0210 ED52          sbc hl,de
000304 0212 22BD89        ld (endprg),hl
000305 0215                ENDPUTCOMM:
000306 0215 C31E02        jp lpedit
000307 0218                FPGDN:
000308 0218 CDDE00        call initprg
000309 021B C31E02        jp lpedit
000310 021E
000311 021E                ;EDIT/INPUT PROGRAM
000312 021E                EDITPRG:
000313 021E                LPEDIT:
000314 021E 214081        ld hl,pagex
000315 0221 CD5B05        call clearpg
000316 0224 CDE502        call writemes
000317 0227 CD1601        call getkey
000318 022A FE1B          cp esc
000319 022C CA8701        jp z,putcomm
000320 022F FE04          cp del
000321 0231 CA8602        jp z,delchar
000322 0234 FE0D          cp enter
000323 0236 CA5C02        jp z,endedit
000324 0239 FE1E          cp kleft
000325 023B CA2401        jp z,fleft
000326 023E FE1A          cp kright
000327 0240 CA3501        jp z,fright
000328 0243 FE02          cp khome
000329 0245                ;jp z,fhome
000330 0245 FE05          cp kend
000331 0247 CA4801        jp z,fend
000332 024A FE03          cp kpgup
000333 024C                ;jp z,fpgup
000334 024C FE06          cp kpgdn
000335 024E CA1802        jp z,fpgdn
000336 0251 FE08          cp kbkspace
000337 0253 CA7501        jp z,fbkspace
000338 0256 CD6102        call insert
000339 0259 C31E02        jp lpedit
000340 025C                ENDEDIT:
000341 025C E1            pop hl
000342 025D C3D600        jp runrun
000343 0260 C9            ret
000344 0261                INSERT:
000345 0261 32E78C        ld (ascii),a
000346 0264 3E02          ld a,fontchar
000347 0266 32E88C        ld (font),a
000348 0269 2AB989        ld hl,(ptprg)
000349 026C 7E            ld a,(hl)
000350 026D FE02          cp 2
000351 026F 2814          jr z,endinst
000352 0271 110400        ld de,4
000353 0274 19            add hl,de

```

```

000354 0275 22B989      ld (ptprg),hl
000355 0278 CD8A02      call insertchar
000356 027B 2ABD89      ld hl,(endprg)
000357 027E 19          add hl,de
000358 027F 22BD89      ld (endprg),hl
000359 0282 C31E02      jp lpedit
000360 0285              ENDINST:
000361 0285 C9          ret
000362 0286              DELCHAR:
000363 0286 CD8C02      call deletechar
000364 0289 C31E02      jp lpedit
000365 028C              ;DELETE CHARACTER FROM MPRG
000366 028C              ;INPUT :PTPRG
000367 028C              ;OUTPUT :MPRG
000368 028C              DELETECHAR:
000369 028C F5          push af
000370 028D C5          push bc
000371 028E D5          push de
000372 028F E5          push hl
000373 0290 2ABD89      ld hl,(endprg)
000374 0293 ED5BB989     ld de,(ptprg)
000375 0297 1A          ld a,(de)
000376 0298 FE01          cp 01
000377 029A 2819        jr z,delch1
000378 029C FE02          cp 02
000379 029E 2815        jr z,delch1
000380 02A0 ED52          sbc hl,de
000381 02A2 4D          ld c,l
000382 02A3 44          ld b,h
000383 02A4 210400      ld hl,4
000384 02A7 19          add hl,de
000385 02A8 EDB0        ldir
000386 02AA 2ABD89      ld hl,(endprg)
000387 02AD 110400      ld de,4
000388 02B0 ED52          sbc hl,de
000389 02B2 22BD89      ld (endprg),hl
000390 02B5              DELCH1:
000391 02B5 E1          pop hl
000392 02B6 D1          pop de
000393 02B7 C1          pop bc
000394 02B8 F1          pop af
000395 02B9 C9          ret
000396 02BA              ;INSERT CHARACTER TO MPRG
000397 02BA              ;INPUT :PTPRG,ASCII,FONT
000398 02BA              ;OUTPUT :MPRG
000399 02BA              INSERTCHAR:
000400 02BA F5          push af
000401 02BB C5          push bc
000402 02BC D5          push de
000403 02BD E5          push hl
000404 02BE ED5BBD89     ld de,(endprg)

```

000405 02C2 D5	push de
000406 02C3 210400	ld hl,4
000407 02C6 19	add hl,de
000408 02C7 EB	ex de,hl
000409 02C8 ED4BB989	ld bc,(ptprg)
000410 02CC ED42	sbc hl,bc
000411 02CE 4D	ld c,l
000412 02CF 44	ld b,h
000413 02D0 E1	pop hl
000414 02D1 EDB8	lddr
000415 02D3 3AE78C	ld a,(ascii)
000416 02D6 77	ld (hl),a
000417 02D7 23	inc hl
000418 02D8 3AE88C	ld a,(font)
000419 02DB 77	ld (hl),a
000420 02DC 23	inc hl
000421 02DD 3E00	ld a,0
000422 02DF 77	ld (hl),a
000423 02E0 E1	pop hl
000424 02E1 D1	pop de
000425 02E2 C1	pop bc
000426 02E3 F1	pop af
000427 02E4 C9	ret
000428 02E5	:WRITE STRING TO PAGE ON EDIT/INPUT MODE
000429 02E5	:INPUT :PTPRG
000430 02E5	:OUTPUT :PAGEX
000431 02E5	WRITEMES:
000432 02E5 F5	push af
000433 02E6 C5	push bc
000434 02E7 D5	push de
000435 02E8 E5	push hl
000436 02E9 CD8905	call testscan
000437 02EC DD2AB989	ld ix.(ptprg)
000438 02F0 0E11	ld c,17
000439 02F2	LPWRM:
000440 02F2 DD7E01	ld a,(ix+1)
000441 02F5 32E88C	ld (font),a
000442 02F8 DD5600	ld d,(ix+0)
000443 02FB CD3405	call putch
000444 02FE DD7E00	ld a,(ix+0)
000445 0301 FE01	cp fontcomm
000446 0303 2811	jr z,stopwr
000447 0305 0D	dec c
000448 0306 3EFF	ld a,0ffh
000449 0308 B9	cp c
000450 0309 280B	jr z,stopwr
000451 030B DD2B	dec ix
000452 030D DD2B	dec ix
000453 030F DD2B	dec ix
000454 0311 DD2B	dec ix
000455 0313 C3F202	jp lpwrm

```

000456 0316          STOPWR:
000457 0316 FB              ei
000458 0317 E1             pop hl
000459 0318 D1             pop de
000460 0319 C1             pop bc
000461 031A F1             pop af
000462 031B C9             ret
000463 031C
000464 031C          ;*****MOVING FUNCTION*****
000465 031C          FINDCOMM:
000466 031C 110400         ld de,4
000467 031F DD2AB989      ld ix,(ptprg)
000468 0323          LPFCR:
000469 0323 DD7E01         ld a,(ix+1)
000470 0326 FE01         cp fontcomm
000471 0328 2804         jr z,endfcr
000472 032A DD19         add ix,de
000473 032C 18F5         jr lpfcr
000474 032E          ENDFCR:
000475 032E DD22B989      ld (ptprg),ix
000476 0332 C9             ret
000477 0333          RUNMODE:
000478 0333 CDDA04         call clearall
000479 0336 21C589      ld hl,mprg
000480 0339 22B989      ld (ptprg),hl
000481 033C          NEXT:
000482 033C 3AE58C         ld a,(mode)
000483 033F FE00         cp 0
000484 0341 C0             ret nz
000485 0342 CD5601         call incptprg
000486 0345 2AB989      ld hl,(ptprg)
000487 0348 22B789      ld (runprg),hl
000488 034B          :call findcomr
000489 034B DD2AB789      ld ix,(runprg)
000490 034F 21F082      ld hl,pagey
000491 0352 22C389      ld (pageput),hl
000492 0355 CDFC04         call writestr
000493 0358 DD2AB989      ld ix,(ptprg)
000494 035C DD7E01         ld a,(ix+1)
000495 035F FE01         cp fontcomm
000496 0361 C23C03         jp nz,next
000497 0364 DD7E00         ld a,(ix+0)
000498 0367 FE02         cp 02
000499 0369 CA3303         jp z,runmode
000500 036C FE04         cp 04
000501 036E 2006         jr nz,no04
000502 0370 CDE903         call shiftup
000503 0373 C33C03         jp next
000504 0376          no04:
000505 0376 FE05         cp 05
000506 0378 2006         jr nz,no05

```

```

000507 037A CD9703          call shiftdn
000508 037D C33C03          jp next
000509 0380                no05:
000510 0380 FE06              cp 06
000511 0382 2006            jr nz,no06
000512 0384 CD3404          call shiftleft
000513 0387 C33C03          jp next
000514 038A                no06:
000515 038A FE07            cp 07
000516 038C 2006            jr nz,no07
000517 038E CD4804          call shiftright
000518 0391 C33C03          jp next
000519 0394                no07:
000520 0394                NEXT2:
000521 0394 C33C03          jp next
000522 0397
000523 0397                ;SHIFT DOWN
000524 0397                shiftdn:
000525 0397 CD8905          call testscan
000526 039A 0618            ld b,maxrow
000527 039C                lpdwn:
000528 039C CDAC03          call rtdn1l
000529 039F CD9205          call dela
000530 03A2 3AE58C          ld a,(mode)
000531 03A5 FE00            cp 0
000532 03A7 C0              ret nz
000533 03A8                :djnz lpdwn
000534 03A8 CDA205          call delaym
000535 03AB C9              ret
000536 03AC                ;ROTATE DOWN 1 LINE
000537 03AC                rtdn1l:
000538 03AC F5              push af
000539 03AD C5              push bc
000540 03AE D5              push de
000541 03AF E5              push hl
000542 03B0 21F082          ld hl,pagex+pagesize
000543 03B3 112281          ld de,linebuff+rowbyte
000544 03B6 011200          ld bc,rowbyte
000545 03B9 EDB8            lddr
000546 03BB 11F082          ld de,pagex+pagesize
000547 03BE 21DE82          ld hl,pagex+pagesize-rowbyte
000548 03C1 019E01          ld bc,pagesize-rowbyte
000549 03C4 EDB8            lddr
000550 03C6 115281          ld de,pagex+rowbyte
000551 03C9 21A084          ld hl,pagey+pagesize
000552 03CC 011200          ld bc,rowbyte
000553 03CF EDB8            lddr
000554 03D1 11A084          ld de,pagey+pagesize
000555 03D4 218E84          ld hl,pagey+pagesize-rowbyte
000556 03D7 019E01          ld bc,pagesize-rowbyte
000557 03DA EDB8            lddr

```

```

000558 03DC 212281      ld hl,linebuff+rowbyte
000559 03DF 011200      ld bc,rowbyte
000560 03E2 EDB8         lddr
000561 03E4 E1           pop hl
000562 03E5 D1           pop de
000563 03E6 C1           pop bc
000564 03E7 F1           pop af
000565 03E8 C9           ret
000566 03E9             ;SHIFT UP
000567 03E9             shiftup:
000568 03E9 CD8905        call testscan
000569 03EC 0618         ld b,maxrow
000570 03EE             lpup:
000571 03EE CDFA03        call rtup1l
000572 03F1 CD9205        call dela
000573 03F4 10F8        djnz lpup
000574 03F6 CDA205        call delaym
000575 03F9 C9           ret
000576 03FA             ;ROTATE UP 1 LINE
000577 03FA             rtup1l:
000578 03FA F5           push af
000579 03FB C5           push bc
000580 03FC D5           push de
000581 03FD E5           push hl
000582 03FE 214081        ld hl,pagex
000583 0401 111081        ld de,linebuff
000584 0404 011200        ld bc,rowbyte
000585 0407 EDB0         ldir
000586 0409 114081        ld de,pagex
000587 040C 215281        ld hl,pagex+rowbyte
000588 040F 019E01        ld bc,pagesize-rowbyte
000589 0412 EDB0         ldir
000590 0414 21F082        ld hl,pagey
000591 0417 011200        ld bc,rowbyte
000592 041A EDB0         ldir
000593 041C 11F082        ld de,pagey
000594 041F 210283        ld hl,pagey+rowbyte
000595 0422 019E01        ld bc,pagesize-rowbyte
000596 0425 EDB0         ldir
000597 0427 211081        ld hl,linebuff
000598 042A 011200        ld bc,rowbyte
000599 042D EDB0         ldir
000600 042F E1           pop hl
000601 0430 D1           pop de
000602 0431 C1           pop bc
000603 0432 F1           pop af
000604 0433 C9           ret
000605 0434
000606 0434             ;SHIFT LEFT
000607 0434             SHIFTLLEFT:
000608 0434 0690         ld b,rowbyte*8

```

```

000609 0436          LPSHLT:
000610 0436 CD9B04          call rtlal1b
000611 0439 CD9205          call dela
000612 043C 3AE58C          ld a,(mode)
000613 043F FE00          cp 0
000614 0441 C0          ret nz
000615 0442 10F2          djnz lpsHLT
000616 0444 CDA205          call delaym
000617 0447 C9          ret
000618 0448
000619 0448          ;SHIFT RIGHT
000620 0448          SHIFTRIGHT:
000621 0448 0690          ld b,rowbyte*8
000622 044A          LPSHRT:
000623 044A CD5C04          call rtral1b
000624 044D CD9205          call dela
000625 0450 3AE58C          ld a,(mode)
000626 0453 FE00          cp 0
000627 0455 C0          ret nz
000628 0456 10F2          djnz lpsHRT
000629 0458 CDA205          call delaym
000630 045B C9          ret
000631 045C
000632 045C          ;rotate right all line 1 bit
000633 045C          rtral1b:
000634 045C C5          push bc
000635 045D CD8905          call testscan
000636 0460 0618          ld b,maxrow
000637 0462 0E00          ld c,0
000638 0464          lprralb1:
000639 0464 CD6D04          call rtr111b
000640 0467 0C          inc c
000641 0468 10FA          djnz lprralb1
000642 046A C1          pop bc
000643 046B FB          ei
000644 046C C9          ret
000645 046D
000646 046D          ;rotate right 1 line 1 bit
000647 046D          ;input c=line
000648 046D          rtr111b:
000649 046D F5          push af
000650 046E C5          push bc
000651 046F D5          push de
000652 0470 E5          push hl
000653 0471 51          ld d,c
000654 0472 1E12          ld e,rowbyte
000655 0474 ED5C          mlt de
000656 0476 214081          ld hl,pagex
000657 0479 19          add hl,de
000658 047A E5          push hl
000659 047B 21F082          ld hl,pagey

```

000660 047E 19	add hl,de
000661 047F E5	push hl
000662 0480 215181	ld hl,pagex+rowbyte-1
000663 0483 19	add hl,de
000664 0484 0612	ld b,rowbyte
000665 0486 48	ld c,b
000666 0487 7E	ld a,(hl)
000667 0488 1F	rra
000668 0489 E1	pop hl
000669 048A	shiftr1:
000670 048A CB1E	rr (hl)
000671 048C 23	inc hl
000672 048D 10FB	djnz shiftr1
000673 048F E1	pop hl
000674 0490 41	ld b,c
000675 0491	shiftr2:
000676 0491 CB1E	rr (hl)
000677 0493 23	inc hl
000678 0494 10FB	djnz shiftr2
000679 0496 E1	pop hl
000680 0497 D1	pop de
000681 0498 C1	pop bc
000682 0499 F1	pop af
000683 049A C9	ret
000684 049B	
000685 049B	;rotate left all line 1 bit
000686 049B	rtl1b:
000687 049B C5	push bc
000688 049C CD8905	call testscan
000689 049F 0618	ld b,maxrow
000690 04A1 0E00	ld c,0
000691 04A3	lprlab1:
000692 04A3 CDAC04	call rtl11b
000693 04A6 0C	inc c
000694 04A7 10FA	djnz lprlab1
000695 04A9 C1	pop bc
000696 04AA FB	ei
000697 04AB C9	ret
000698 04AC	
000699 04AC	
000700 04AC	;rotate left 1 line 1 bit
000701 04AC	;input c=line
000702 04AC	rtl11b:
000703 04AC F5	push af
000704 04AD C5	push bc
000705 04AE D5	push de
000706 04AF E5	push hl
000707 04B0 51	ld d,c
000708 04B1 1E12	ld e,rowbyte
000709 04B3 ED5C	mlt de
000710 04B5 215181	ld hl,pagex+rowbyte-1

```

000711 04B8 19      add hl,de
000712 04B9 E5      push hl
000713 04BA 210183   ld hl,pagey+rowbyte-1
000714 04BD 19      add hl,de
000715 04BE E5      push hl
000716 04BF 214081   ld hl,pagex
000717 04C2 19      add hl,de
000718 04C3 0612    ld b,rowbyte
000719 04C5 48      ld c,b
000720 04C6 7E      ld a,(hl)
000721 04C7 17      rla
000722 04C8 E1      pop hl
000723 04C9          shiftl1:
000724 04C9 CB16    rl (hl)
000725 04CB 2B      dec hl
000726 04CC 10FB    djnz shiftl1
000727 04CE E1      pop hl
000728 04CF 41      ld b,c
000729 04D0          shiftl2:
000730 04D0 CB16    rl (hl)
000731 04D2 2B      dec hl
000732 04D3 10FB    djnz shiftl2
000733 04D5 E1      pop hl
000734 04D6 D1      pop de
000735 04D7 C1      pop bc
000736 04D8 F1      pop af
000737 04D9 C9      ret
000738 04DA          CLEARALL:
000739 04DA 214081   ld hl,pagex
000740 04DD          ;call clearpage
000741 04DD 21F082   ld hl,pagey
000742 04E0 CD5B05   call clearpg
000743 04E3 21A084   ld hl,pagez
000744 04E6 CD5B05   call clearpg
000745 04E9 215086   ld hl,pagea
000746 04EC CD5B05   call clearpg
000747 04EF 210088   ld hl,pagem
000748 04F2 CD5B05   call clearpg
000749 04F5 211081   ld hl,linebuff
000750 04F8 CD5B05   call clearpg
000751 04FB C9      ret
000752 04FC          ;WRITE STRING TO PAGE
000753 04FC          ;INPUT :IX = PRG
000754 04FC          ;OUTPUT :PAGE
000755 04FC          WRITESTR:
000756 04FC F5      push af
000757 04FD C5      push bc
000758 04FE D5      push de
000759 04FF E5      push hl
000760 0500 CD8905   call testscan
000761 0503 3E02    ld a,fontchar

```

000762 0505 32E88C	ld (font),a
000763 0508 0E00	ld c,0
000764 050A	LPWRST:
000765 050A DD7E01	ld a,(ix+1)
000766 050D FE01	cp fontcomm
000767 050F CA2E05	jp z,stopwrite
000768 0512 3E12	ld a,18
000769 0514 B9	cp c
000770 0515 380B	jr c,fullwrite
000771 0517 DD5600	ld d,(ix+0)
000772 051A 3E00	ld a,0
000773 051C 32C189	ld (crlocal),a
000774 051F CD3405	call putch
000775 0522	FULLWRITE:
000776 0522 DD23	inc ix
000777 0524 DD23	inc ix
000778 0526 DD23	inc ix
000779 0528 DD23	inc ix
000780 052A 0C	inc c
000781 052B C30A05	jp lpwrst
000782 052E	STOPWRITE:
000783 052E FB	ei
000784 052F E1	pop hl
000785 0530 D1	pop de
000786 0531 C1	pop bc
000787 0532 F1	pop af
000788 0533 C9	ret
000789 0534	
000790 0534	;PUT CHARACTER TO PAGE ON MOVE/RUN MODE
000791 0534	;INPUT :D=ASCII,C=LOCATION(COL),(PAGEPUT)
000792 0534	;OUTPUT :PAGEX
000793 0534	PUTCH:
000794 0534 F5	push af
000795 0535 C5	push bc
000796 0536 D5	push de
000797 0537 E5	push hl
000798 0538 DDE5	push ix
000799 053A CD6805	call getcharadd
000800 053D DD2AC389	ld ix,(pageput)
000801 0541 0600	ld b,0
000802 0543 DD09	add ix,bc
000803 0545 011200	ld bc,rowbyte
000804 0548 1E18	ld e,24
000805 054A	LPPUT:
000806 054A 7E	ld a,(hl)
000807 054B DD7700	ld (ix+0),a
000808 054E 23	inc hl
000809 054F DD09	add ix,bc
000810 0551 1D	dec e
000811 0552 20F6	jr nz,lpput
000812 0554 DDE1	pop ix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000813 0556 E1          pop hl
000814 0557 D1          pop de
000815 0558 C1          pop bc
000816 0559 F1          pop af
000817 055A C9          ret
000818 055B
000819 055B          ;CLEAR PAGE
000820 055B          ;INPUT :HL=PAGE ADDRESS
000821 055B          ;OUTPUT :PAGE
000822 055B          CLEARPG:
000823 055B 01B001      ld bc,pagesize
000824 055E          LPCLR:
000825 055E 3E00      ld a,0
000826 0560 77          ld (hl),a
000827 0561 23          inc hl
000828 0562 0B          dec bc
000829 0563 78          ld a,b
000830 0564 B1          or c
000831 0565 20F7       jr nz,lpclr
000832 0567 C9          ret
000833 0568
000834 0568          ;GET CHARACTER ADDRESS
000835 0568          ;INPUT :D = ASCII
000836 0568          ;OUTPUT :HL = ADDRESS OF FONT
000837 0568          GETCHARADD:
000838 0568 F5          push af
000839 0569 D5          push de
000840 056A 3AE88C       ld a,(font)
000841 056D FE01       cp fontcomm
000842 056F 2005       jr nz,gcafc
000843 0571 210100     ld hl,fontcomm
000844 0574 1808       jr gca
000845 0576          GCAFC:
000846 0576 214840     ld hl,fontte
000847 0579 7A          ld a,d
000848 057A 1620       ld d,32
000849 057C 92          sub d
000850 057D 57          ld d,a
000851 057E          GCA:
000852 057E 1E18       ld e,24
000853 0580 ED5C       mlt de
000854 0582 214840     ld hl,fontte
000855 0585 19          add hl,de
000856 0586 D1          pop de
000857 0587 F1          pop af
000858 0588 C9          ret
000859 0589
000860 0589          TESTSCAN:
000861 0589 3AB089     ld a,(rowscan)
000862 058C FE18       cp 18h
000863 058E 20F9       jr nz,testscan

```

000864 0590 F3	di
000865 0591 C9	ret
000866 0592	DELA:
000867 0592 F5	push af
000868 0593 E5	push hl
000869 0594 211400	ld hl,delaymin
000870 0597 C3A705	jp lpdelm1
000871 059A	DELAY:
000872 059A F5	push af
000873 059B E5	push hl
000874 059C 21E803	ld hl,delaymid
000875 059F C3A705	jp lpdelm1
000876 05A2	DELA:M:
000877 05A2 F5	push af
000878 05A3 E5	push hl
000879 05A4 210000	ld hl,delaymax
000880 05A7	LPDELM1:
000881 05A7 FB	ei
000882 05A8	LPDELM:
000883 05A8 CDB905	call checkkey
000884 05AB 3AE58C	ld a,(mode)
000885 05AE FE00	cp 0
000886 05B0 C0	ret nz
000887 05B1 2B	dec hl
000888 05B2 7D	ld a,l
000889 05B3 B4	or h
000890 05B4 20F2	jr nz,lpdelm
000891 05B6 E1	pop hl
000892 05B7 F1	pop af
000893 05B8 C9	ret
000894 05B9	CHECKKEY:
000895 05B9 F5	push af
000896 05BA CD0A01	call clearkey
000897 05BD DB9D	in a,(inkey)
000898 05BF FE0D	cp 13
000899 05C1 2804	jr z,key13
000900 05C3 3E00	ld a,0
000901 05C5 1800	jr endcheck
000902 05C7	KEY13:
000903 05C7	ENDCHECK:
000904 05C7 32E58C	ld (mode),a
000905 05CA F1	pop af
000906 05CB C9	ret
000907 05CC	WATCHDOG:
000908 05CC C9	ret
000909 05CD	
000910 05CD	ORGRP:
000911 05CD 0101000053020000	.DB 01,01,0,0,83,02,0,0,85,02,0,0,67
05D5 5502000043	
000912 05DA 0301000004010005	.DB 03,01,0,0,04,01,0,05,01,0,0,06
05E2 01000006	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000913	05E6	PRG1:	
000914	05E6 0101000004015302		.DB 01,01,0,0,04,01,83,02,0,0,73,06
	05EE 00004906		
000915	05F2 02010000000000		.DB 02,01,0,0,0,0,0
000916	05F9		
000917	4000	.ORG 4000h	
000918	4000	FONTCOM:	
000919	4000 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h ;SP
000920	4008 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000921	4010 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000922	4018		
000923	4018 FEFE82829E9E9E9E		.DB 0feh,0feh,82h,82h,9eh,9eh,9eh,9eh ;[
000924	4020 9E9E9E9E9E9E9E9E		.DB 9eh,9eh,9eh,9eh,9eh,9eh,9eh,9eh
000925	4028 9E9E9E9E8282FEFE		.DB 9eh,9eh,9eh,9eh,82h,82h,0feh,0feh
000926	4030		
000927	4030 FEFE8282F2F2F2F2		.DB 0feh,0feh,82h,82h,0f2h,0f2h,0f2h,0f2h ;]
000928	4038 F2F2F2F2F2F2F2F2		.DB 0f2h,0f2h,0f2h,0f2h,0f2h,0f2h,0f2h,0f2h
000929	4040 F2F2F2F28282FEFE		.DB 0f2h,0f2h,0f2h,0f2h,82h,82h,0feh,0feh
000930	4048		
000931	4048	;FONT 8X24	
000932	4048	FONTTE:	
000933	4048		
000934	4048 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h ;SP
000935	4050 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000936	4058 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000937	4060		
000938	4060 0000000000000010		.DB 00h,00h,00h,00h,00h,00h,00h,10h ;!
000939	4068 3838383810100010		.DB 38h,38h,38h,38h,10h,10h,00h,10h
000940	4070 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000941	4078		
000942	4078 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h ;"
000943	4080 2828280000000000		.DB 28h,28h,28h,00h,00h,00h,00h,00h
000944	4088 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000945	4090		
000946	4090 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h ;#
000947	4098 0024247E247E2424		.DB 00h,24h,24h,7eh,24h,7eh,24h,24h
000948	40A0 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000949	40A8		
000950	40A8 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h ;\$
000951	40B0 00107C907C127C10		.DB 00h,10h,7ch,90h,7ch,12h,7ch,10h
000952	40B8 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000953	40C0		
000954	40C0 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h ;%
000955	40C8 00C2C40810204686		.DB 00h,0c2h,0c4h,08h,10h,20h,46h,86h
000956	40D0 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000957	40D8		
000958	40D8 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h ;&
000959	40E0 00386C38528A8C73		.DB 00h,38h,6ch,38h,52h,8ah,8ch,73h
000960	40E8 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h
000961	40F0		
000962	40F0 0000000000000000		.DB 00h,00h,00h,00h,00h,00h,00h,00h ;'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ควรแก้ไขใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000963 40F8 1808100000000000	.DB 18h,08h,10h,00h,00h,00h,00h,00h
000964 4100 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00
000965 4108	
000966 4108 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;(
000967 4110 1020202020201008	.DB 10h,20h,20h,20h,20h,20h,10h,08h
000968 4118 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
000969 4120	
000970 4120 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;)
000971 4128 1008080808081020	.DB 10h,08h,08h,08h,08h,08h,10h,20h
000972 4130 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
000973 4138	
000974 4138 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;*
000975 4140 000024187E182400	.DB 00h,00h,24h,18h,7eh,18h,24h,00h
000976 4148 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
000977 4150	
000978 4150 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;+
000979 4158 000018187E181800	.DB 00h,00h,18h,18h,7eh,18h,18h,00h
000980 4160 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
000981 4168	
000982 4168 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;.
000983 4170 0000000000001818	.DB 00h,00h,00h,00h,00h,00h,18h,18h
000984 4178 0830000000000000	.DB 08h.30h,00h,00h,00h,00h,00h,00h,00h
000985 4180	
000986 4180 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;-
000987 4188 000000007E000000	.DB 00h,00h,00h,00h,7eh,00h,00h,00h
000988 4190 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
000989 4198	
000990 4198 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;.
000991 41A0 0000000000001818	.DB 00h,00h,00h,00h,00h,00h,18h,18h
000992 41A8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
000993 41B0	
000994 41B0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;/
000995 41B8 0408081020204040	.DB 04h,08h,08h,10h,20h,20h,40h,40h
000996 41C0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
000997 41C8	
000998 41C8 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;0
000999 41D0 42424E5A7242423C	.DB 42h,42h,4eh,5ah,72h,42h,42h,3ch
001000 41D8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001001 41E0	
001002 41E0 0000000000000010	.DB 00h,00h,00h,00h,00h,00h,00h,10h ;1
001003 41E8 3010101010101038	.DB 30h,10h,10h,10h,10h,10h,10h,38h
001004 41F0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001005 41F8	
001006 41F8 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;2
001007 4200 420202040810207E	.DB 42h,02h,02h,04h,08h,10h,20h,7eh
001008 4208 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001009 4210	
001010 4210 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;3
001011 4218 4202021C0202423C	.DB 42h,02h,02h,1ch,02h,02h,42h,3ch
001012 4220 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001013 4228	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001014 4228 0000000000000004	.DB 00h,00h,00h,00h,00h,00h,00h,04h ;4
001015 4230 0C14244484FE0404	.DB 0ch,14h,24h,44h,84h,0feh,04h,04h
001016 4238 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001017 4240	
001018 4240 00000000000000FC	.DB 00h,00h,00h,00h,00h,00h,00h,0fch ;5
001019 4248 8080F88402028478	.DB 80h,80h,0f8h,84h,02h,02h,84h,78h
001020 4250 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001021 4258	
001022 4258 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;6
001023 4260 2040407C4242423C	.DB 20h,40h,40h,7ch,42h,42h,42h,3ch
001024 4268 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001025 4270	
001026 4270 000000000000007E	.DB 00h,00h,00h,00h,00h,00h,00h,7eh ;7
001027 4278 4202040408081010	.DB 42h,02h,04h,04h,08h,08h,10h,10h
001028 4280 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001029 4288	
001030 4288 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;8
001031 4290 4242423C4242423C	.DB 42h,42h,42h,3ch,42h,42h,42h,3ch
001032 4298 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001033 42A0	
001034 42A0 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;9
001035 42A8 4242423E02040870	.DB 42h,42h,42h,3eh,02h,04h,08h,70h
001036 42B0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001037 42B8	
001038 42B8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ::
001039 42C0 0000181800181800	.DB 00h,00h,18h,18h,00h,18h,18h,00h
001040 42C8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001041 42D0	
001042 42D0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ::
001043 42D8 0000181800181830	.DB 00h,00h,18h,18h,00h,18h,18h,30h
001044 42E0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001045 42E8	
001046 42E8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;<
001047 42F0 0004081020100804	.DB 00h,04h,08h,10h,20h,10h,08h,04h
001048 42F8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001049 4300	
001050 4300 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;=
001051 4308 0000007C007C0000	.DB 00h,00h,00h,7ch,00h,7ch,00h,00h
001052 4310 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001053 4318	
001054 4318 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;>
001055 4320 0020100804081020	.DB 00h,20h,10h,08h,04h,08h,10h,20h
001056 4328 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001057 4330	
001058 4330 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;?
001059 4338 4242320408080008	.DB 42h,42h,32h,04h,08h,08h,00h,08h
001060 4340 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001061 4348	
001062 4348 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;@
001063 4350 1C224D55554A201E	.DB 1ch,22h,4dh,55h,55h,4ah,20h,1eh
001064 4358 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001065 4360	
001066 4360 0000000000000010	.DB 00h,00h,00h,00h,00h,00h,00h,10h ;A
001067 4368 28448282FE828282	.DB 28h,44h,82h,82h,0feh,82h,82h,82h
001068 4370 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001069 4378	
001070 4378 00000000000000FC	.DB 00h,00h,00h,00h,00h,00h,00h,0fch ;B
001071 4380 4242427C424242FC	.DB 42h,42h,42h,7ch,42h,42h,42h,0fch
001072 4388 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001073 4390	
001074 4390 000000000000001C	.DB 00h,00h,00h,00h,00h,00h,00h,1ch ;C
001075 4398 22404040404040221C	.DB 22h,40h,40h,40h,40h,40h,22h,1ch
001076 43A0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001077 43A8	
001078 43A8 00000000000000F8	.DB 00h,00h,00h,00h,00h,00h,00h,0f8h ;D
001079 43B0 4442424242424244F8	.DB 44h,42h,42h,42h,42h,42h,44h,0f8h
001080 43B8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001081 43C0	
001082 43C0 00000000000000FE	.DB 00h,00h,00h,00h,00h,00h,00h,0feh ;E
001083 43C8 42424878484042FE	.DB 42h,42h,48h,78h,48h,40h,42h,0feh
001084 43D0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001085 43D8	
001086 43D8 00000000000000FE	.DB 00h,00h,00h,00h,00h,00h,00h,0feh ;F
001087 43E0 42404878484040E0	.DB 42h,40h,48h,78h,48h,40h,40h,0e0h
001088 43E8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001089 43F0	
001090 43F0 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;G
001091 43F8 424240404E42423A	.DB 42h,42h,40h,40h,4eh,42h,42h,3ah
001092 4400 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001093 4408	
001094 4408 0000000000000042	.DB 00h,00h,00h,00h,00h,00h,00h,42h ;H
001095 4410 4242427E42424242	.DB 42h,42h,42h,7eh,42h,42h,42h,42h
001096 4418 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001097 4420	
001098 4420 0000000000000038	.DB 00h,00h,00h,00h,00h,00h,00h,38h ;I
001099 4428 1010101010101038	.DB 10h,10h,10h,10h,10h,10h,10h,38h
001100 4430 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001101 4438	
001102 4438 00000000000000E	.DB 00h,00h,00h,00h,00h,00h,00h,0eh ;J
001103 4440 0404040404044438	.DB 04h,04h,04h,04h,04h,04h,44h,38h
001104 4448 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001105 4450	
001106 4450 000000000000004E	.DB 00h,00h,00h,00h,00h,00h,00h,4eh ;K
001107 4458 44485060504844E7	.DB 44h,48h,50h,60h,50h,48h,44h,0e7h
001108 4460 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001109 4468	
001110 4468 0000000000000070	.DB 00h,00h,00h,00h,00h,00h,00h,70h ;L
001111 4470 202020202020227E	.DB 20h,20h,20h,20h,20h,20h,22h,7eh
001112 4478 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001113 4480	
001114 4480 0000000000000082	.DB 00h,00h,00h,00h,00h,00h,00h,82h ;M
001115 4488 C6AA929282828282	.DB 0c6h,0aah,92h,92h,82h,82h,82h,82h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญฎาตหนาไปไซประโยชนดานการคา
ไมวารณิตุๆ ทั้งสิ้น อิกทั้งหามมิใหัดดแปลงเนือหาและตองอางอิงถึงเจาของเอกสารทุกครั้งที่มีการนำไปไซ

001116 4490 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001117 4498	
001118 4498 00000000000000E7	.DB 00h,00h,00h,00h,00h,00h,00h,0e7h ;N
001119 44A0 625252524A4A46E6	.DB 62h,52h,52h,52h,4ah,4ah,46h,0e6h
001120 44A8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001121 44B0	
001122 44B0 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;O
001123 44B8 424242424242423C	.DB 42h,42h,42h,42h,42h,42h,42h,3ch
001124 44C0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001125 44C8	
001126 44C8 000000000000005C	.DB 00h,00h,00h,00h,00h,00h,00h,5ch ;P
001127 44D0 2222223C20202070	.DB 22h,22h,22h,3ch,20h,20h,20h,70h
001128 44D8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001129 44E0	
001130 44E0 000000000000003C	.DB 00h,00h,00h,00h,00h,00h,00h,3ch ;Q
001131 44E8 42424242524A463D	.DB 42h,42h,42h,42h,52h,4ah,46h,3dh
001132 44F0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001133 44F8	
001134 44F8 00000000000000FC	.DB 00h,00h,00h,00h,00h,00h,00h,0Fch ;R
001135 4500 4242427C504844E7	.DB 42h,42h,42h,7ch,50h,48h,44h,0e7h
001136 4508 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001137 4510	
001138 4510 000000000000003A	.DB 00h,00h,00h,00h,00h,00h,00h,3ah ;S
001139 4518 464240300C42625C	.DB 46h,42h,40h,30h,0ch,42h,62h,5ch
001140 4520 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001141 4528	
001142 4528 00000000000000FE	.DB 00h,00h,00h,00h,00h,00h,00h,0feh ;T
001143 4530 9210101010101038	.DB 92h,10h,10h,10h,10h,10h,10h,38h
001144 4538 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001145 4540	
001146 4540 00000000000000E7	.DB 00h,00h,00h,00h,00h,00h,00h,0e7h ;U
001147 4548 424242424242423C	.DB 42h,42h,42h,42h,42h,42h,42h,3ch
001148 4550 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001149 4558	
001150 4558 0000000000000082	.DB 00h,00h,00h,00h,00h,00h,00h,82h ;V
001151 4560 8282824444282810	.DB 82h,82h,82h,44h,44h,28h,28h,10h
001152 4568 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001153 4570	
001154 4570 0000000000000082	.DB 00h,00h,00h,00h,00h,00h,00h,82h ;W
001155 4578 8282828292AA4444	.DB 82h,82h,82h,82h,92h,0aah,44h,44h
001156 4580 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001157 4588	
001158 4588 0000000000000082	.DB 00h,00h,00h,00h,00h,00h,00h,82h ;X
001159 4590 8244281028448282	.DB 82h,44h,28h,10h,28h,44h,82h,82h
001160 4598 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001161 45A0	
001162 45A0 00000000000000C6	.DB 00h,00h,00h,00h,00h,00h,00h,0c6h ;Y
001163 45A8 4444281010101038	.DB 44h,44h,28h,10h,10h,10h,10h,38h
001164 45B0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001165 45B8	
001166 45B8 00000000000000FE	.DB 00h,00h,00h,00h,00h,00h,00h,0feh ;Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001167 45C0 82040810204082FE	.DB 82h,04h,08h,10h,20h,40h,82h,0feh
001168 45C8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001169 45D0	
001170 45D0 0000000000000038	.DB 00h,00h,00h,00h,00h,00h,00h,38H ;[
001171 45D8 2020202020202038	.DB 20h,20h,20h,20h,20h,20h,20h,38h
001172 45E0 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001173 45E8	
001174 45E8 0000000000000040	.DB 00h,00h,00h,00h,00h,00h,00h,40h ;\
001175 45F0 4020201008080404	.DB 40h,20h,20h,10h,08h,08h,04h,04h
001176 45F8 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001177 4600	
001178 4600 0000000000000038	.DB 00h,00h,00h,00h,00h,00h,00h,38h ;]
001179 4608 0808080808080838	.DB 08h,08h,08h,08h,08h,08h,08h,38h
001180 4610 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001181 4618	
001182 4618 000000000102844	.DB 00h,00h,00h,00h,00h,10h,28h,44h ;^
001183 4620 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001184 4628 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h,
4630 00	
001185 4631	
001186 4631 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;_
001187 4639 000000000000007E	.DB 00h,00h,00h,00h,00h,00h,00h,7eh
001188 4641 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001189 4649	
001190 4649 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;:
001191 4651 3020100000000000	.DB 30h,20h,10h,00h,00h,00h,00h,00h
001192 4659 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001193 4661	
001194 4661 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;:
001195 4669 007804047C84847A	.DB 00h,78h,04h,04h,7ch,84h,84h,7ah
001196 4671 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001197 4679	
001198 4679 0000000000000060	.DB 00h,00h,00h,00h,00h,00h,00h,60h ;:
001199 4681 203824222222225C	.DB 20h,38h,24h,22h,22h,22h,22h,5ch
001200 4689 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001201 4691	
001202 4691 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;C
001203 4699 003C42404040423C	.DB 00h,3ch,42h,40h,40h,40h,42h,3ch
001204 46A1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001205 46A9	
001206 46A9 000000000000000C	.DB 00h,00h,00h,00h,00h,00h,00h,0ch ;D
001207 46B1 041C2444444443A	.DB 04h,1ch,24h,44h,44h,44h,44h,3ah
001208 46B9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001209 46C1	
001210 46C1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;E
001211 46C9 003C427E4040423C	.DB 00h,3ch,42h,7eh,40h,40h,42h,3ch
001212 46D1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001213 46D9	
001214 46D9 0000000000000018	.DB 00h,00h,00h,00h,00h,00h,00h,18h ;F
001215 46E1 2420207020202070	.DB 24h,20h,20h,70h,20h,20h,20h,70h
001216 46E9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001217 46F1	
001218 46F1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;G
001219 46F9 003A4444443C0404	.DB 00h,3ah,44h,44h,44h,3ch,04h,04h
001220 4701 4438000000000000	.DB 44h,38h,00h,00h,00h,00h,00h,00h
001221 4709	
001222 4709 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;H
001223 4711 202C322222222262	.DB 20h,2ch,32h,22h,22h,22h,22h,62h
001224 4719 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001225 4721	
001226 4721 0000000000000010	.DB 00h,00h,00h,00h,00h,00h,00h,10h ;I
001227 4729 0030101010101038	.DB 00h,30h,10h,10h,10h,10h,10h,38h
001228 4731 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001229 4739	
001230 4739 0000000000000004	.DB 00h,00h,00h,00h,00h,00h,00h,04h ;J
001231 4741 000C040404040424	.DB 00h,0ch,04h,04h,04h,04h,04h,24h
001232 4749 1800000000000000	.DB 18h,00h,00h,00h,00h,00h,00h,00h
001233 4751	
001234 4751 0000000000000060	.DB 00h,00h,00h,00h,00h,00h,00h,60h ;K
001235 4759 2024283030282464	.DB 20h,24h,28h,30h,30h,28h,24h,64h
001236 4761 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001237 4769	
001238 4769 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;L
001239 4771 080808080808081C	.DB 08h,08h,08h,08h,08h,08h,08h,1ch
001240 4779 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001241 4781	
001242 4781 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;M
001243 4789 000E929292929292	.DB 00h,0eh,92h,92h,92h,92h,92h,92h
001244 4791 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001245 4799	
001246 4799 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;N
001247 47A1 00D8644444444444	.DB 00h,0d8h,64h,44h,44h,44h,44h,44h
001248 47A9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001249 47B1	
001250 47B1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;O
001251 47B9 003C42424242423C	.DB 00h,3ch,42h,42h,42h,42h,42h,3ch
001252 47C1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001253 47C9	
001254 47C9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;P
001255 47D1 005C222222223C20	.DB 00h,5ch,22h,22h,22h,22h,3ch,20h
001256 47D9 2070000000000000	.DB 20h,70h,00h,00h,00h,00h,00h,00h
001257 47E1	
001258 47E1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;Q
001259 47E9 003A444444443C04	.DB 00h,3ah,44h,44h,44h,44h,3ch,04h
001260 47F1 040E000000000000	.DB 04h,0eh,00h,00h,00h,00h,00h,00h
001261 47F9	
001262 47F9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;R
001263 4801 0058242420202070	.DB 00h,58h,24h,24h,20h,20h,20h,70h
001264 4809 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001265 4811	
001266 4811 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;S
001267 4819 003C42403C02423C	.DB 00h,3ch,42h,40h,3ch,02h,42h,3ch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญฎาตหน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001268 4821 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001269 4829	
001270 4829 0000000000000020	.DB 00h,00h,00h,00h,00h,00h,00h,20h ;T
001271 4831 2070202020202418	.DB 20h,70h,20h,20h,20h,20h,24h,18h
001272 4839 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001273 4841	
001274 4841 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;U
001275 4849 004444444444443A	.DB 00h,44h,44h,44h,44h,44h,44h,3ah
001276 4851 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001277 4859	
001278 4859 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;V
001279 4861 0082828244442810	.DB 00h,82h,82h,82h,44h,44h,28h,10h
001280 4869 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001281 4871	
001282 4871 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;W
001283 4879 0082828292AA4444	.DB 00h,82h,82h,82h,92h,0aah,44h,44h
001284 4881 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001285 4889	
001286 4889 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;X
001287 4891 0082442810284482	.DB 00h,82h,44h,28h,10h,28h,44h,82h
001288 4899 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001289 48A1	
001290 48A1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;Y
001291 48A9 00424242423E0202	.DB 00h,42h,42h,42h,42h,3eh,02h,02h
001292 48B1 0438000000000000	.DB 04h,38h,00h,00h,00h,00h,00h,00h
001293 48B9	
001294 48B9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;Z
001295 48C1 007E42040810227E	.DB 00h,7eh,42h,04h,08h,10h,22h,7eh
001296 48C9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001297 48D1	
001298 48D1 0000000000000000E	.DB 00h,00h,00h,00h,00h,00h,00h,0eh ;{
001299 48D9 101020402010100E	.DB 10h,10h,20h,40h,20h,10h,10h,0eh
001300 48E1 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001301 48E9	
001302 48E9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;
001303 48F1 0010101000101010	.DB 00h,10h,10h,10h,00h,10h,10h,10h
001304 48F9 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001305 4901	
001306 4901 0000000000000070	.DB 00h,00h,00h,00h,00h,00h,00h,70h ;}
001307 4909 0808040204080870	.DB 08h,08h,04h,02h,04h,08h,08h,70h
001308 4911 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001309 4919	
001310 4919 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;~
001311 4921 000060920C000000	.DB 00h,00h,60h,92h,0ch,00h,00h,00h
001312 4929 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001313 4931	
001314 4931 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h ;
001315 4939 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001316 4941 0000000000000000	.DB 00h,00h,00h,00h,00h,00h,00h,00h
001317 4949	
001318 4949	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001319 4949          DISBUF:
001320 8110          .ORG 8110h
001321 8110
001322 8110          linebuff: .rs 48
001323 8140          pagex: .rs PAGESIZE;
001324 82F0          pagey: .rs PAGESIZE;
001325 84A0          pagez: .rs PAGESIZE
001326 8650          pagea: .rs PAGESIZE
001327 8800          pagem: .rs PAGESIZE
001328 89B0
001329 89B0 00          rowscan: .db 0
001330 89B1 0000          addscan: .dw 0
001331 89B3 0000          crcol: .dw 0
001332 89B5 0000          crrow: .dw 0
001333 89B7
001334 89B7 0000          runprg: .dw 0
001335 89B9 0000          ptprg: .dw 0
001336 89BB 0000          stprg: .dw 0
001337 89BD 0000          endprg: .dw 0
001338 89BF 0000          crmprg: .dw 0
001339 89C1
001340 89C1 0000          crlocal: .dw 0
001341 89C3 0000          pageput: .dw 0
001342 89C5          mprg: .rs 200*4
001343 8CE5 0000          mode: .dw 0
001344 8CE7
001345 8CE7 00          ascii: .db 0
001346 8CE8 00          font: .db 0
001347 8CE9 0000          boot: .dw 0
001348 8CEB          .END

```

```

ADDSCAN =89B1 INKEY =009D RLDROL =000E
ASCII =8CE7 INSERT =0261 ROWBYTE =0012
BBR =0039 INSERTCH=02BA ROWSCAN =89B0
BOOT =8CE9 KBKSP =0008 RTDN1L =03AC
CBAR =003A KDN =0019 RTL1L1B =04AC
CBR =0038 KEND =0005 RTLAL1B =049B
CHARSIZE=0018 KEY13 =05C7 RTR1L1B =046D
CHECKKEY =05B9 KHOME =0002 RTRAL1B =045C
CLEARALL=04DA KLEFT =001E RTUP1L =03FA
CLEARKEY=010A KPGDN =0006 RUNMODE =0333
CLEARPG =055B KPGUP =0003 RUNPRG =89B7
CLRKEY =009E KRIGHT =001A RUNRUN =00D6
CNTLAO0 =0001 KUP =0018 SCAN1 =0073
CNTLB0 =0003 LINEBUFF=8110 SCAN2 =0088
CONTI =0068 LPCLR =055E SCANDISP=0038
CRCOL =89B3 LPDELM =05A8 SHIFTDN =0397
CRLOCAL =89C1 LPDELM1 =05A7 SHIFTL1 =04C9
CRMPRG =89BF LPDOWN =039C SHIFTL2 =04D0
CRROW =89B5 LPEDIT =021E SHIFTLF=0434
CTPORT1 =008F LPFCR =0323 SHIFTR1 =048A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CTPORT2 =009F LPGETKEY=0119 SHIFTR2 =0491
DECPTPRG=0165 LPPUT =054A SHIFTRIG=0448
DEL =0004 LPRALB1=04A3 SHIFTUP =03E9
DELA =0592 LPRRALB1=0464 START =0000
DELAY =059A LPSHLT =0436 START1 =0003
DELAYM =05A2 LPSHRT =044A STAT0 =0005
DELAYMAX=0000 LPUP =03EE STILL =00D3
DELAYMID=03E8 LPWRM =02F2 STOPWR =0316
DELAYMIN=0014 LPWRST =050A STOPWRIT=052E
DELAYSCA=00FF MAIN =0099 STPRG =89BB
DELCH1 =02B5 MAXCOL =0012 SYSSTACK=80FF
DELCHAR =0286 MAXROW =0018 TCR =0010
DELETECH=028C MODE =8CE5 TDR0 =0007
DINTSCAN=0000 MODEEDIT=0002 TESTSCAN=0589
DISBUF =4949 MODEMOVE=0001 TMDROH =000D
EDITPRG =021E MPRG =89C5 TMDROL =000C
EINTSCAN=0011 NDSP =000C WATCHDOG=05CC
ENDCHECK=05C7 NEXT =033C WRITEMES=02E5
ENEDIT =025C NEXT2 =0394 WRITESTR=04FC
ENDFCR =032E NO04 =0376
ENDINST =0285 NO05 =0380
ENDPRG =89BD NO06 =038A
ENDPUTCO=0215 NO07 =0394
ENDSCAN =008E NODN =01CF
ENTER =000D NOLF =01DB
ESC =001B NORT =01F4
FBKSP =0175 NOUP =01C3
FEND =0148 ORGPRG =05CD
FINDCOMM=031C OVER_COL=0003
FLEFT =0124 PAGEA =8650
FONT =8CE8 PAGEM =8800
FONTCHAR=0002 PAGEPUT =89C3
FONTCOM =4000 PAGESIZE=01B0
FONTCOMM=0001 PAGEX =8140
FONTTE =4048 PAGEY =82F0
FPGDN =0218 PAGEZ =84A0
FRIGHT =0135 PORTROW =008C
FULLWRIT=0522 PRG1 =05E6
GCA =057E PTPRG =89B9
GCAFC =0576 PUTC1 =01E4
GETCHARA=0568 PUTCH =0534
GETKEY =0116 PUTCOMM =0187
INCPTPRG=0156 RDR0 =0009
INIT =0009 RESETKEY=00FA
INITPRG =00DE RLDROH =000F

```

รูปที่ ข.2 โปรแกรมการทำงานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



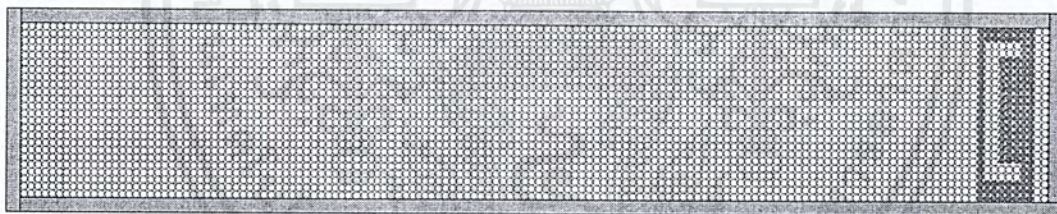
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน LED Matrix Board

ในการควบคุมการแสดงผลที่ บอร์ดอักษรไฟวิ่ง เราสามารถควบคุมการแสดงผลได้ทั้ง 2 ทาง คือในส่วน ควบคุมโดยตรงที่ตัวเครื่องโดยนำ คีย์บอร์ด (Keyboard) มาต่อที่ พอร์ต ต่อ คีย์บอร์ด ที่ด้านข้างของ ตัวเครื่อง นอกจากนี้ยังสามารถควบคุมการแสดงผลที่ คอมพิวเตอร์ พีซี โดยนำ สาย ส่งข้อมูลมาต่อที่ พอร์ต อนุกรม และทำการควบคุมการแสดงผลโดยโปรแกรม ซึ่งเขียนจาก โปรแกรม Visual Basic จะทำการกล่าวถึงวิธีการใช้งานในหัวข้อต่อไป

การใช้งาน LED Matrix Board โดยป้อนข้อมูลโดยตรง

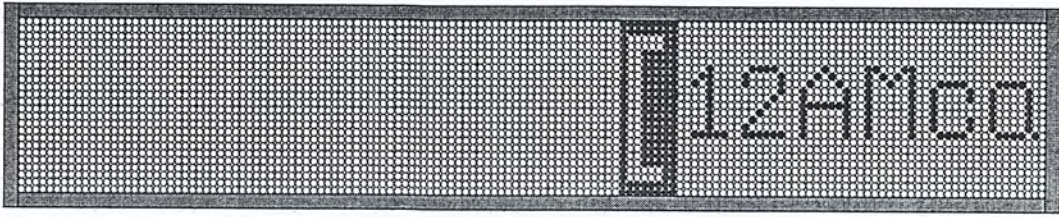
เมื่อเริ่มการทำงานหรือทำการเปิดสวิตช์ ในส่วนของหน่วยประมวลผลกลางจะทำการ ตั้งสถานะเริ่มต้นการแสดงผลก่อนแล้วจึงจะรอการกดแป้นพิมพ์ (แบบ PS/IT) ถ้าไม่มีการกดแป้นพิมพ์ หน่วยประมวลผลกลางจะทำการวนลูปรอรับคีย์ไปเรื่อยๆ จนกว่าจะมีการกดแป้นพิมพ์แล้ว จอแสดงผลจะปรากฏเครื่องหมายวงเล็บเปิดไว้



รูปที่ ค.1 พร้อมที่จะกดแป้นพิมพ์

1. การป้อนข้อความผ่านแป้นพิมพ์

1. สามารถป้อนข้อมูลเป็นตัวเลขและตัวอักษรทั้งภาษาไทยและภาษาอังกฤษได้ ในกรณีที่มีการกดแป้นพิมพ์ จะปรากฏตัวอักษรและตัวเลขที่จอแสดงผล เช่น

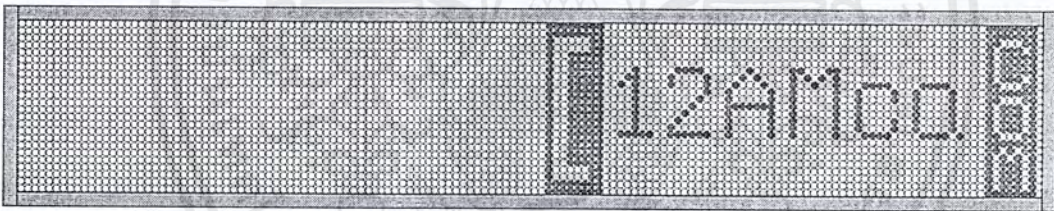


รูปที่ ค.2 การป้อนข้อมูลผ่านเป็นพิมพ์

แต่ถ้ามีการกดแป้นพิมพ์ที่เป็น Shift พร้อมกับมีการกดตัวอักษร หรือตัวเลขจอแสดงผลจะแสดงตัวอักษรขนาดใหญ่

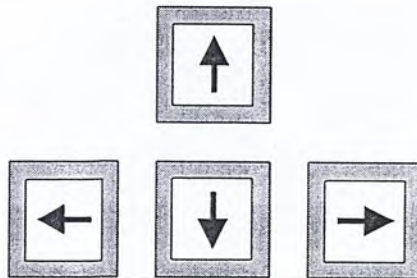
2. การใช้คำสั่งกำหนดรูปแบบการแสดงผล

หลังจากป้อนข้อมูลเป็นตัวอักษรภาษาไทยหรือภาษาอังกฤษและตัวเลขเรียบร้อยแล้ว ในการที่จะกำหนดทิศทางการเคลื่อนที่หรือรูปแบบแสดงข้อมูลของตัวอักษรนั้น จะต้องทำการเลือกแป้นพิมพ์ที่เป็น Esc แล้วจอแสดงผล จะแสดงดังรูปที่ ค.3



รูปที่ ค.3 การใช้คำสั่งกำหนดรูปแบบ

เป็นการแสดงว่าออร์เดอร์รับคำสั่ง A-Z ตามฟังก์ชันการทำงานดังตารางที่ 1 และรอกการกำหนดการเคลื่อนที่ ในการเลือกทิศทางการเคลื่อนที่ให้กดแป้นพิมพ์ที่เป็นลูกศร



รูปที่ ค.4 ลูกศรกำหนดทิศทางแสดงผล


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

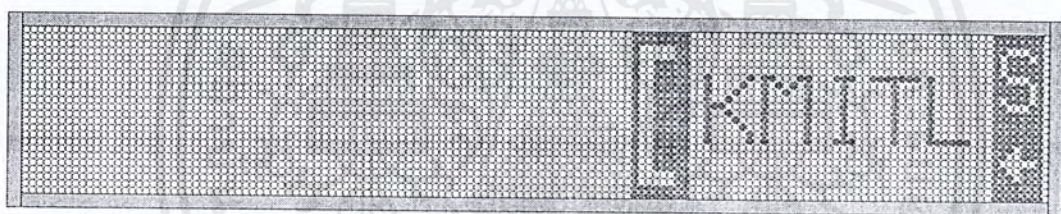
3. ตัวอย่างการใช้คำสั่งและการแสดงผล

การใช้คำสั่งให้ข้อความแสดงผลตามรูปแบบที่มีอยู่นั้น มีวิธีการปฏิบัติก็คือ พิมพ์ข้อความที่เป็นพิมพ์ เมื่อได้ข้อความที่ต้องการแล้ว กดปุ่ม ESC เพื่อเลือกฟังก์ชันการแสดงผลและกด ตัวอักษรตามฟังก์ชันที่กำหนด ดังตารางที่ 1. ตามด้วยลูกศรเพื่อกำหนดทิศทางการเล่นของข้อความ ตัวอย่าง

3.1 การใช้คำสั่ง S “การใช้คำสั่งให้ข้อความเลื่อนจากขวาไปซ้ายของจอแสดงผล”

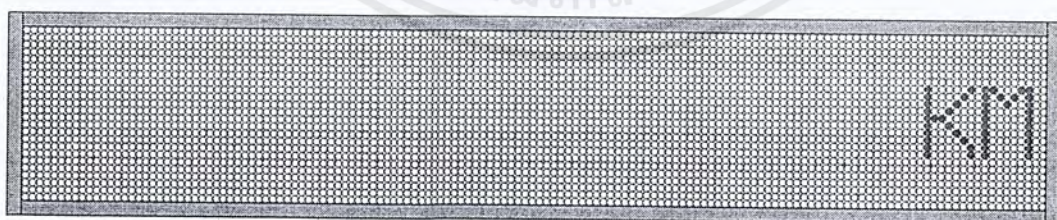
ขั้นตอน

1. พิมพ์ข้อความต่างๆ ที่ต้องการให้แสดง
2. กดปุ่ม ESC
3. กด S
4. กด 




รูปที่ ค.5 การใช้คำสั่งเลื่อนขวา

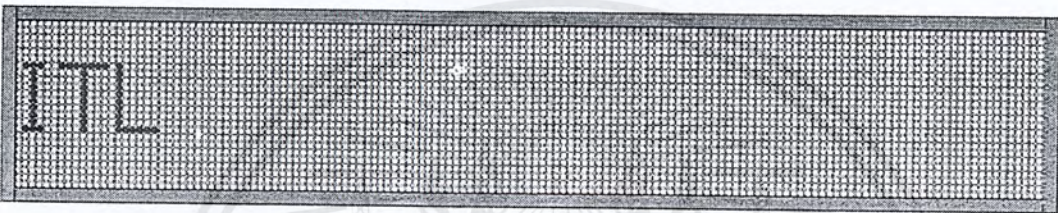
เมื่อกดปุ่ม ENTER ข้อความ “KMITL” จะเลื่อนจากขวาไปซ้ายของจอแสดงผลดังรูปที่ ค.6



รูปที่ ค.6 การเลื่อนจากขวามาซ้ายของข้อความ


3.2 การใช้คำสั่ง S “การใช้คำสั่งให้ข้อความเลื่อนจากซ้ายไปขวาของจอแสดงผล” ขั้นตอน

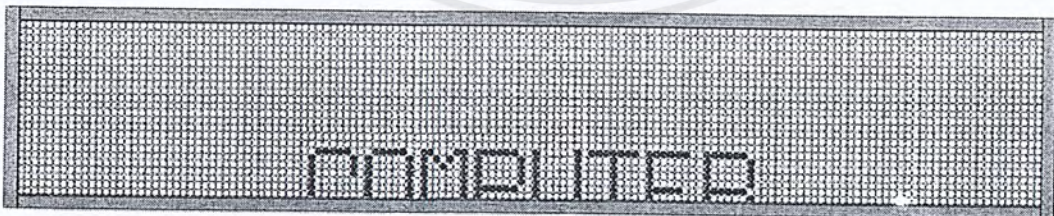
1. พิมพ์ข้อความต่างๆ ที่ต้องการให้แสดง
2. กดปุ่ม ESC
3. กด S
4. กด 



รูปที่ ค.7 การเลื่อนจากซ้ายมาขวาของข้อความ


3.3 การใช้คำสั่ง S “การใช้คำสั่งให้ข้อความเลื่อนจากด้านล่างไปด้านบนของจอแสดงผล” ขั้นตอน

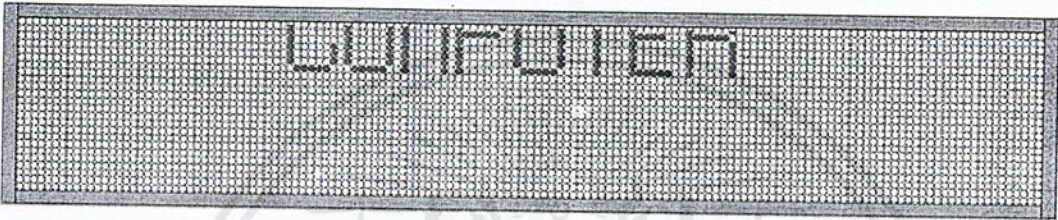
1. พิมพ์ข้อความต่างๆ ที่ต้องการให้แสดง
2. กดปุ่ม ESC
3. กด S
4. กด 



รูปที่ ค.8 การเลื่อนจากด้านล่างไปด้านบนของข้อความ

3.4 การใช้คำสั่ง S “การใช้คำสั่งให้ข้อความเลื่อนจากด้านบนไปด้านล่างของจอแสดงผล” ขั้นตอน

1. พิมพ์ข้อความต่างๆ ที่ต้องการให้แสดง
2. กดปุ่ม ESC
3. กด S
4. กด 



รูปที่ ค.9 การเลื่อนจากด้านบนไปด้านล่างของข้อความ

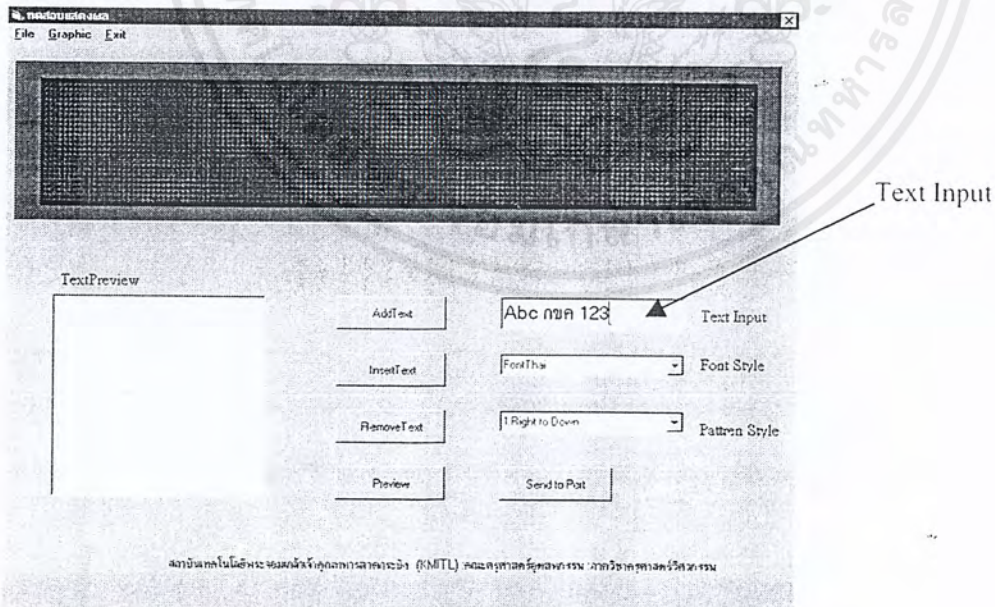
การใช้งาน LED MATRIX BOARD โดยป้อนข้อมูลโดยผ่านทางเครื่อง PC Computer

การป้อนข้อมูลวิธีนี้เป็นการป้อนข้อมูลและควบคุมการแสดงผล โดยผ่านทางเครื่องคอมพิวเตอร์พีซี ซึ่งเราจะทำการติดตั้งโปรแกรมที่ใช้ในการควบคุมการแสดงผล ไว้ในเครื่องคอมพิวเตอร์พีซี เสียก่อน โดยโปรแกรมที่ได้ทำการสร้างขึ้น จะมีความสามารถทางด้านควบคุมการแสดงผล และทดสอบดูตัวอย่างก่อนที่จะนำไปแสดงผลจริงที่ บอร์ดอักษรไฟวิ่ง ซึ่งทำให้มีความง่ายในการใช้งานและจะสามารถใช้งานสะดวกขึ้น เมื่อ บอร์ดอักษรไฟวิ่ง ถูกติดตั้งในสถานที่ที่เราทำการป้อนข้อมูลได้ลำบาก เช่นในสถานที่ที่อยู่สูง แต่ถ้าเราใช้ เครื่อง คอมพิวเตอร์พีซี มาใช้ในการควบคุมเราเพียงแต่นำสายสัญญาณ มาทำการต่อที่ พอร์ตรับข้อมูลแบบอนุกรมที่ บอร์ดอักษรไฟวิ่ง และนำมาต่อที่ พอร์ต คอม (Com) ของเครื่องคอมพิวเตอร์พีซี เราก็จะสามารถใช้โปรแกรมที่สร้างขึ้นควบคุมการแสดงผลได้

วิธีการใช้งานโปรแกรมควบคุมการแสดงผล LED MATRIX BOARD ที่เครื่อง Computer

1. การป้อนข้อความ

สำหรับวิธีการป้อนข้อความเราสามารถป้อนได้ที่ ช่อง Text Input โดยเราสามารถที่จะป้อนข้อความได้ทั้งภาษาอังกฤษและภาษาไทย ดังรูปที่ ก.10



รูปที่ ก.10 แสดงการกรอกข้อความที่ Text Input

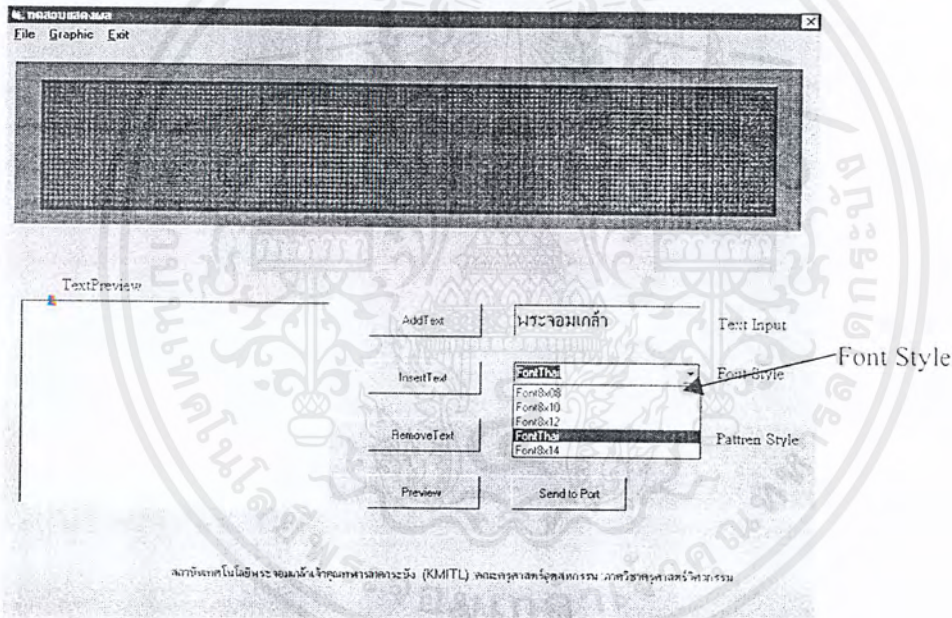
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การป้อนข้อความที่ Text Input เราสามารถที่จะกำหนดข้อความให้มีความยาวเท่าใดก็ได้ตามความพึงพอใจ แต่ไม่ควรจะให้ความยาวมากนักเพราะข้อความจะมีความยาวมากกว่าความยาวของ บอร์ด (Board) จะทำให้ไม่สามารถแสดงผลข้อความได้สมบูรณ์แบบ

2. วิธีการเลือก Font

เมื่อเราทำการป้อนข้อความที่ Text Input เสร็จเรียบร้อยแล้วต่อมาเราก็จะทำการเลือก Font จะใช้ในการแสดงผลซึ่งเราสามารถกำหนด Font ที่จะใช้ในการแสดงผลได้หลายรูปแบบ โดยจะมีภาษาทั้ง ภาษาอังกฤษและภาษาไทย ในขนาดและรูปแบบแตกต่างกัน

การเลือกFont นั้นเราจะทำการเลือกที่ เมื่อเราใช้ Mouse คลิกที่ช่องนี้จะปรากฏ Font ให้เลือกในหลายๆรูปแบบและหลายๆขนาด ทั้ง ภาษาอังกฤษและภาษาไทยดังรูปที่ ค.11



รูปที่ ค.11 การเลือก Font Style

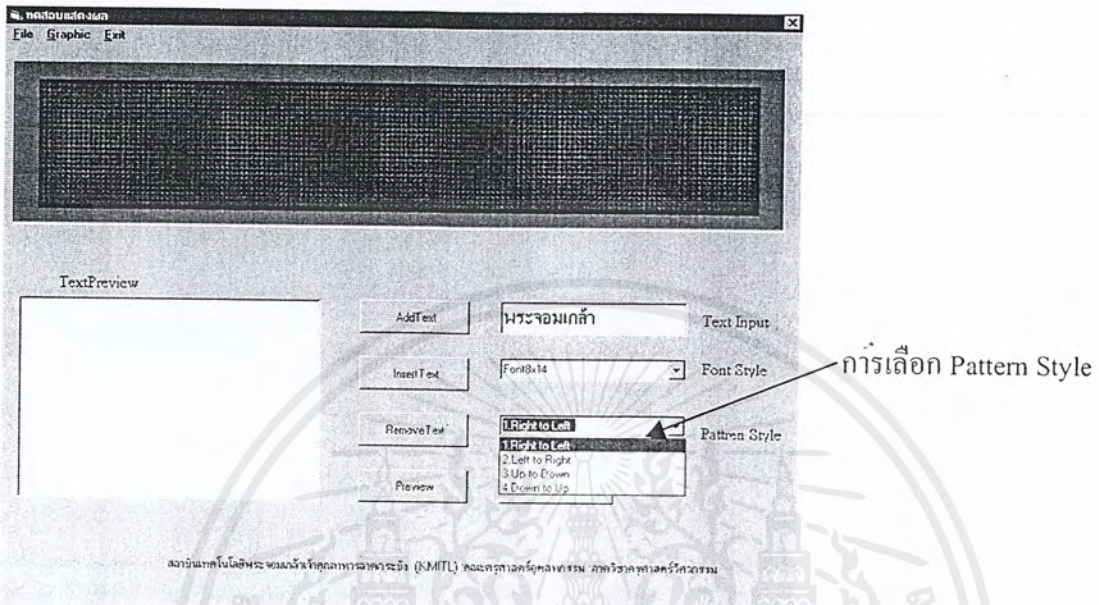
การพิมพ์ข้อความเราสามารถที่จะพิมพ์ได้ทั้งข้อความที่เป็นภาษาไทยและภาษาอังกฤษได้ในข้อความเดียว

3. การเลือกรูปแบบการแสดงผล

การเลือกรูปแบบการแสดงผล เป็นการเลือกลักษณะการแสดงผลที่ บอร์ด ว่าจะให้มีการแสดงผลข้อความในลักษณะไหน เช่น เคลื่อนที่จากทางด้านซ้ายไปทางด้านขวา จากทางด้านบนลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางด้านล่าง โดยเราจะทำการเลือกลักษณะการแสดงผลที่ช่อง Pattern Style เมื่อเรา ใช้ เมาส์ (Mouse) คลิกที่ช่องนี้ ก็จะปรากฏรูปแบบการแสดงผลให้เราเลือกในหลายๆ รูปแบบ ดังรูปที่ ค.12

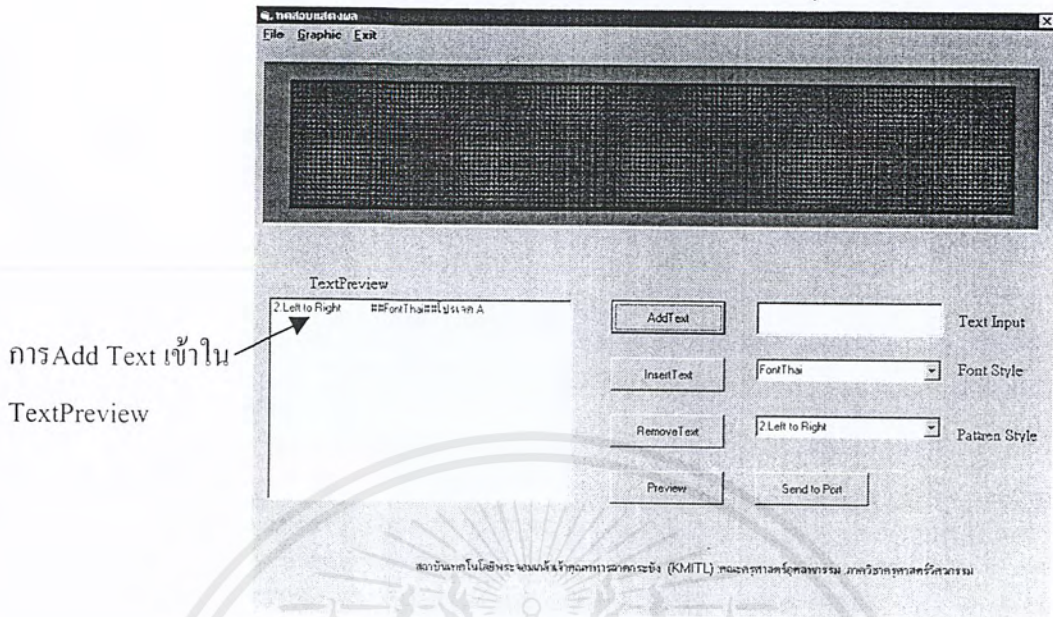


รูปที่ ค.12 ลักษณะการเลือกลักษณะการแสดงผลที่ Pattern Style

ในการเลือกลักษณะการแสดงผลเราสามารถที่จะเลือกลักษณะการแสดงผลได้หลายๆรูปแบบ ในหนึ่งข้อความ

4. Add ข้อมูล

การ Add ข้อมูลที่เราได้เลือก Font . Pattern เสร็จเรียบร้อยแล้ว เป็นการเลือกข้อความที่ต้องการให้แสดงผลย้ายเข้ามาใน TextPreview เพื่อยืนยันการแสดงผลว่าเราเลือก Font เป็นอะไร มีลักษณะการแสดงผลเป็นอย่างไร ทั้งนี้จะมีข้อควรระวัง คือ เมื่อทำการกำหนดข้อความที่ Text Input เสร็จเรียบร้อยแล้ว ต้องทำการเลือก Font Style และ Pattern Style ไม่เช่นนั้น จะไม่แสดงผลอะไรที่ บอร์ด เมื่อเราทำการAdd Text อย่างถูกวิธีแล้วจะได้ลักษณะ Text Preview ดังรูปที่ ค.13



รูปที่ ค.13 ลักษณะ Text Preview

การใช้งานของAdd Text จะใช้งานคู่กับปุ่มRemoveText คือปุ่มที่ใช้ในการลบข้อความที่ต้องการลบออกจาก Text Preview บางข้อความ ซึ่งจะลบออกทั้ง ข้อความ,Font,และรูปแบบการแสดงผล นอกจากนี้ก็ยังมีปุ่ม InsertText ที่ใช้ในการเพิ่มข้อความใหม่ๆเข้าไป

ตัวอย่าง "3. Up to down ##Font8x10##ลาดกระบัง"

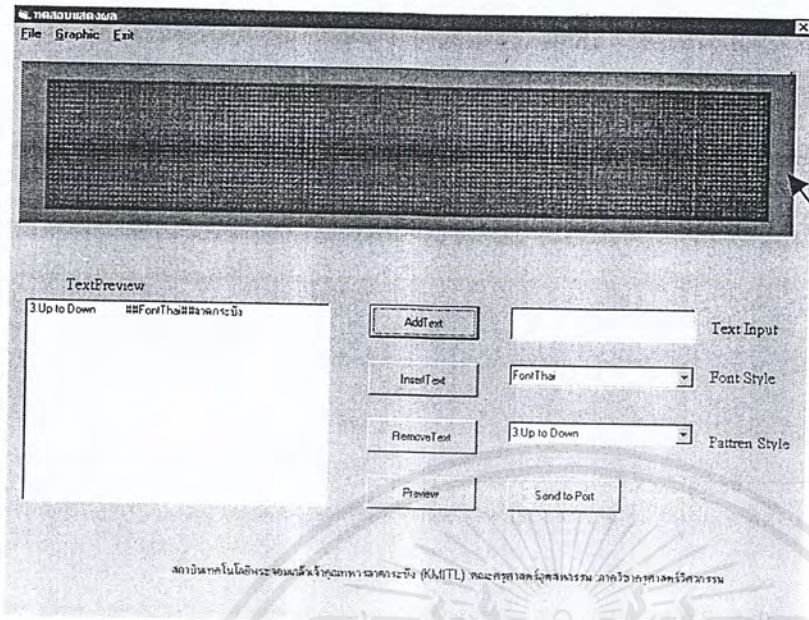
3. Up to down คือ ลักษณะการแสดงผล

##Font8x10## คือ Font ตัวหนังสือ

ลาดกระบัง คือ ข้อความที่ต้องการแสดงผล

5. การแสดงผล

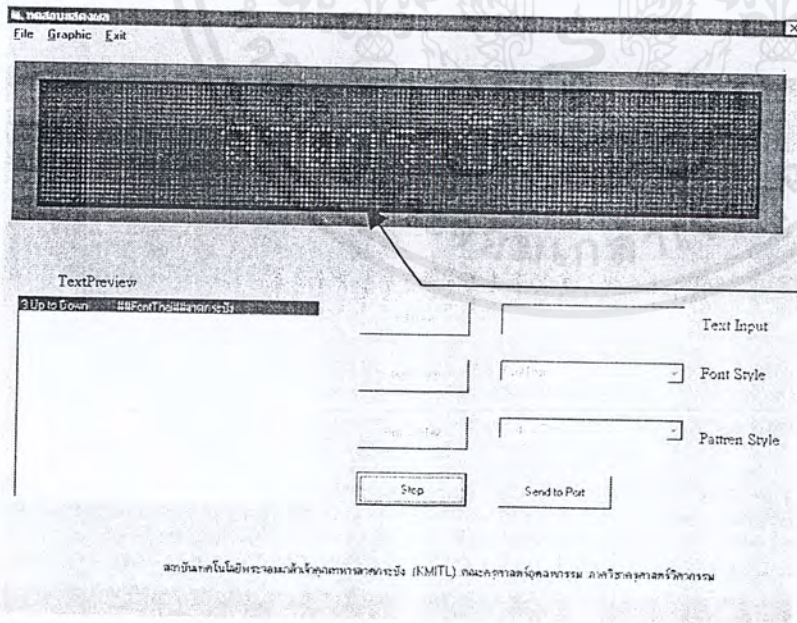
การแสดงผลของข้อความ คือ ทดสอบและตรวจลักษณะการแสดงผลของข้อความ ว่าเป็นไปตามที่เราต้องการหรือไม่ โดยจะแสดงที่ บอร์ด จำลองการแสดงผลในตัวโปรแกรม ดังรูปที่ ค.14 ซึ่งจะแสดง บอร์ด จำลองการแสดงผล



Board จำลอง
การแสดงผล

รูปที่ ก.14 ลักษณะบอร์ดจำลองการแสดงผล

ลักษณะการแสดงผลจะทำงานก็ต่อเมื่อเรากดปุ่ม Preview โดยจะมีข้อความแสดงผลที่บอร์ดจำลองการแสดงผล ซึ่งจะมีลักษณะการแสดงผลดังรูปที่ ก.15



ลักษณะการแสดงผลคำ
ว่า "ลาคระบัง"

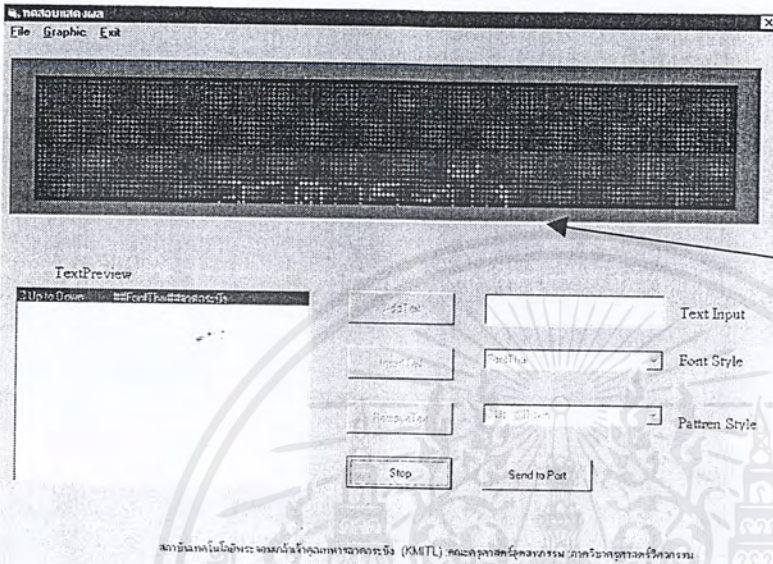
รูป ก.15 ลักษณะการแสดงผลที่ บอร์ดจำลองการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลการเคลื่อนที่ของตัวอักษร จะมีลักษณะการแสดงผลตามคำสั่งที่เราเลือกใน

Pattern Style

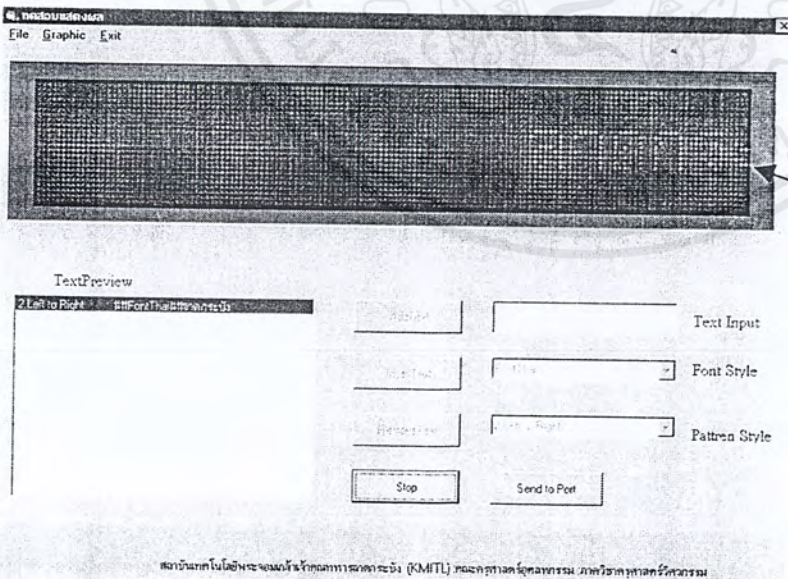
ตัวอย่างที่ 1 คำสั่ง “Up to Down” คือการแสดงผลในลักษณะเลื่อนจากด้านบนสู่ด้านล่าง



การแสดงผลในลักษณะเลื่อนจากด้านบนสู่ด้านล่าง

รูปที่ ค.16 การใช้คำสั่ง “Up to Down”

ตัวอย่างที่ 2 คำสั่ง “Left to Right” คือการแสดงผลในลักษณะการเคลื่อนจากทางด้านซ้ายไปสู่ด้านขวา



การแสดงผลในลักษณะเลื่อนจากซ้ายสู่ด้านขวา

รูปที่ ค.17 การใช้คำสั่ง “Left to Right”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การส่งข้อมูลแสดงผลที่ LED MATRIX BOARD

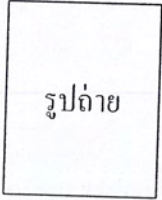
การตรวจสอบลักษณะการทำงานในหัวข้อที่ 5 จะยังไม่แสดงผลที่ บอร์ดอักษรไฟวิ่ง ในทันที เป็นแต่เพียงการตรวจสอบลักษณะการแสดงผลในตัวโปรแกรมเท่านั้นถ้า ถ้าเราต้องการให้ข้อความแสดงที่ บอร์ดอักษรไฟวิ่ง จริงๆ เราจะต้องกดที่ปุ่ม Send to Port คือการที่ข้อมูลการแสดงผลจะถูกส่งออกที่ พอร์ตอนุกรม ไปที่ ตัว บอร์ดอักษรไฟวิ่ง ข้อความที่ได้จะแสดงผลที่ บอร์ดอักษรไฟวิ่ง ในทันที



บรรณานุกรม

- สุทธิศักดิ์ พงษ์นาพานิชย์. **Visual Basic 5.0 Professional**. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น ,2541
- กิตติ ภัคดีวัฒนะกุล และคณะ **Visual Basic 5 ฉบับ โปรแกรมเมอร์**. กรุงเทพมหานคร :
 เคทีพีคอมพิวเตอร์ แอนด์ คอนซัลท์ ,2542
- สำราญ คำยิ่ง และคณะ “**บอร์ดตัวอักษรไฟวิ่งแอลอีดี**.” ปรินิพนยานิพนธ์ครุศาสตร์อุตสาหกรรม-
 บัณฑิต สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์. คณะครุศาสตร์อุตสาหกรรม
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. ,2541
- นพดล ผันเผาะ และคณะ “**บอร์ดตัวอักษรไฟวิ่ง แอลอีดีเมตริกซ์**.” ปรินิพนยานิพนธ์ครุศาสตร์-
 อุตสาหกรรมบัณฑิต สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์. คณะครุศาสตร์-
 อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. ,2542

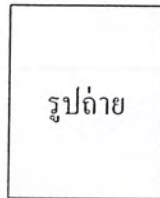
ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานិพนธ์	นายชาญชัย พนมเชื้อ
วันเดือนปีเกิด	20 กุมภาพันธ์ 2522
สถานที่เกิด	จังหวัดสระแก้ว
ภูมิลำเนาเดิม	191 หมู่ 4 ต.ท่าแยก อ.เมือง จ.สระแก้ว
ที่อยู่ปัจจุบัน	191 หมู่ 4 ต.ท่าแยก อ.เมือง จ.สระแก้ว
โทรศัพท์	037 - 437037
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนบ้านหนองเตียน
มัธยมศึกษาตอนต้น	โรงเรียนสระแก้ว
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคปราจีนบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคปราจีนบุรี
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คดีพจน์	คิดก่อนทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญาณพันธ์ นายวิจักษ์ ปราโมทย์
วันเดือนปีเกิด 9 กันยายน 2521
สถานที่เกิด จังหวัดระยอง
ภูมิลำเนาเดิม 1/1 หมู่ 8 ต.ปากน้ำประแสร์ อ.แกลง จ.ระยอง
ที่อยู่ปัจจุบัน 1/1 หมู่ 8 ต.ปากน้ำประแสร์ อ.แกลง จ.ระยอง
โทรศัพท์ 038 - 661286

ประวัติการศึกษา
ประถมศึกษา โรงเรียนชุมชนวัดตะเคียนงาม
มัธยมศึกษาตอนต้น โรงเรียนชำนานูสามัคคีวิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.) วิทยาลัยเทคนิคระยอง
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) วิทยาลัยเทคนิคระยอง
ปริญญาตรี สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล -
ทุนการศึกษา -
คติพจน์ ความอิสระเสรีคือความสุขที่แท้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

รูปถ่าย

ชื่อผู้ทำปริญญาบัตร นายหัตถ์นัย กุลทัศน์
วันเดือนปีเกิด 16 พฤศจิกายน 2522
สถานที่เกิด จังหวัดนครนายก
ภูมิลำเนาเดิม 24 หมู่ 3 ต.บ้านพร้าว อ.บ้านนาจ จ.นครนายก
ที่อยู่ปัจจุบัน 24 หมู่ 3 ต.บ้านพร้าว อ.บ้านนาจ จ.นครนายก
โทรศัพท์ 037-381864

ประวัติการศึกษา

ประถมศึกษา โรงเรียนไทยรัฐวิทยา 73 (วัดบ้านพร้าว)
มัธยมศึกษาตอนต้น โรงเรียนนครนายกวิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.) วิทยาลัยเทคนิคนครนายก
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) วิทยาลัยเทคนิคนครนายก
ปริญญาตรี สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล -
ทุนการศึกษา -
คดิพจน์ ชื่อกินไม่หมด คคกินไม่นาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้