



เครื่องควบคุมการใช้โทรศัพท์โดยใช้การ์ดชนิดเจาะรู
PUNCH-CARD TELEPHONE CONTROLLER



ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง **039023** ใช้

เครื่องควบคุมการใช้โทรศัพท์โดยใช้การ์ดชนิดเจาะรู
PUNCH-CARD TELEPHONE CONTROLLER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาบัตรปีการศึกษา 2540

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมการใช้โทรศัพท์โดยใช้การ์ดชนิดเจาะรู

PUNCH-CARD TELEPHONE CONTROLLER

ผู้จัดทำ

1. นางสาว กฤตยา นาคประสิทธิ์ 37014005
2. นาย เกียรติคุณ วัฒนานุกุลชัย 37014027

ปรโมทย์
(รศ.ดร. ปรโมทย์ วาดเขียน)
อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิชา 01014116 Project II ชั้นปี 4A

โครงการ เครื่องควบคุมการใช้โทรศัพท์โดยใช้การ์ดชนิดเจาะรู

PUNCH - CARD TELEPHONE CONTROLLER

ผู้ร่วมงาน 1. นางสาว กฤตยา นาคประสิทธิ์ 37014005

2. นาย เกียรติคุณ วัฒนานุกุลชัย 37014027

อาจารย์ที่ปรึกษา

(รศ.ดร. ปราโมทย์ วาดเขียน)

บทคัดย่อ

โครงการนี้เป็นการสร้างเครื่องควบคุมการใช้โทรศัพท์ โดยใช้การ์ดชนิดเจาะรูเป็นตัวควบคุมการปล่อยสัญญาณโทรศัพท์เข้าสู่เครื่องโทรศัพท์ ซึ่งจะใช้รังสีอินฟราเรดเป็นตัวตรวจจับสัญญาณจากการ์ด แล้วทำการตรวจสอบรหัสจากข้อมูลที่ได้นบันทึกไว้ในหน่วยความจำ ซึ่งถ้าการตรวจสอบได้ผลถูกต้อง เครื่องนี้จะทำการอนุญาตให้สามารถใช้โทรศัพท์ในการโทรออกได้ โดยการควบคุมของไมโครคอนโทรลเลอร์

เมื่อทำการใช้โทรศัพท์เสร็จเรียบร้อยแล้ว เครื่องนี้จะทำการพิมพ์ค่าใช้บริการ เลขหมายโทรศัพท์ที่ทำการติดต่อ รวมไปถึงเวลาทั้งหมดที่ใช้บริการ โทรศัพท์ ออกทางเครื่องพิมพ์ ซึ่งจะทำให้เราทราบรายละเอียดครบถ้วน

ABSTRACT

This is the telephone controller project. It uses the punch-card to control sending telephone signal to the telephone. Infrared is used to detect the signal from cards , then check for codes , saved in the codes are correct , this controller will allows to call outgoing .

When the call has finished , this controller use the printer to service charge , telephone number and total time used. So we know the complete information.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ทัว ๆ ไปเกี่ยวกับระบบโทรศัพท์	2
2.2 โทรศัพท์ระบบกดปุ่ม (DTMF)	2
2.3 การคิดบริการของโทรศัพท์	3
2.3.1 อัตราค่าบริการการใช้โทรศัพท์ขององค์การโทรศัพท์	3
2.3.2 อัตราค่าบริการโทรศัพท์ในเขตกรุงเทพฯและปริมณฑล	3
2.3.3 อัตราค่าบริการโทรศัพท์ทางไกลต่างจังหวัด	4
2.4 สัญญาณพื้นฐาน	6
2.4.1 สัญญาณในการติดต่อกันระหว่างเครื่องส่งและเครื่องรับโทรศัพท์	7
2.5 MT8870 ไอซีถอดรหัสความถี่โทรศัพท์ (Integrated DTMF Receiver)	7
2.5.1 คุณสมบัติของ MT8870	7
2.5.2 โครงสร้างของ MT8870	8
2.5.3 ฟังก์ชันการทำงานภายในของ MT8870	8
2.5.3.1 ภาคกรองสัญญาณความถี่ (Filter Section)	8
2.5.3.2 ภาคถอดรหัส (Decoder Section)	9
2.5.3.3 ภาคตรวจสอบสัญญาณ (Steering Circuit)	9
2.5.3.4 ภาคขยายสัญญาณความแตกต่าง	11
2.5.3.5 ภาคกำเนิดความถี่ (Oscillator)	11
2.6 วงจรดีเทคสัญญาณสายไม่ว่าง (Busy Tone)	11
2.7 โปรแกรมขับรูทีน	12
2.7.1 คำสั่งกระโดด	13
2.7.2 การกำหนดตำแหน่งแอดเดรส	13
2.7.3 วิธีการอ้างอิงแอดเดรส	13
2.8 ฟลิปฟลอปเจเค	15
2.8.1 รายละเอียดของขาต่อ	15
2.9 การพัฒนาโปรแกรมที่ใช้กับไมโครคอนโทรลเลอร์ด้วยเครื่องมือไมโครคอมพิวเตอร์	17
2.9.1 โครงสร้างภายในโปรแกรมต้นฉบับ	17
2.9.2 การแปลความด้วยโปรแกรมแอสเซมบลอร์	18
2.9.3 การโอนย้ายโปรแกรมสู่ระบบไมโครคอนโทรลเลอร์	19
2.9.4 การใช้งานพื้นฐานของบอร์ดอิพรอมอีมิเตอร์ (EE - 232)	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบและการสร้าง	23
3.1 บล็อกไดอะแกรมของเครื่องควบคุมการใช้โทรศัพท์โดยใช้การ์ดชนิดเจาะรู	23
3.2 การออกแบบการ์ดเจาะรหัส	24
3.3 การออกแบบส่วนอุปกรณ์อ่านรหัสการ์ด	24
3.4 การออกแบบส่วนที่เป็น TONE DECODER	25
3.5 การออกแบบส่วนที่เป็นวงจร DTMF DECODER	26
3.6 การออกแบบส่วนที่เป็นวงจรตรวจสอบการวางหูและยกหูโทรศัพท์	27
3.7 บอร์ดไมโครคอนโทรลเลอร์แอนท์ - 32 (ANT - 32)	28
3.7.1 คุณสมบัติของบอร์ดแอนท์ - 32 (ANT - 32)	28
3.7.2 แนวทางในการพัฒนาโปรแกรม	28
3.7.3 ทึทีแอลไอโอพอร์ต 8255 (TTL I/O PORT)	29
3.7.4 พอร์ตแอลซีดี (LCD PORT)	31
3.7.5 ขาสัญญาณของ LCD MODULE	32
3.8 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์	32
3.8.1 การต่อแบบเมมโมรี่แมพ (Memory map)	32
3.8.2 การต่อแบบพอร์ตไอ / โอ (I/O port)	33
3.8.3 ชุดคำสั่งควบคุมและการแสดงข้อความ	33
3.9 ความเข้าใจพื้นฐาน	34
3.9.1 รายละเอียดของแต่ละคำสั่ง	34
3.9.2 การอ่านและเขียนข้อมูลกับ DDRAM/CGRAM	37
3.9.3 แนวทางการเขียนโปรแกรมควบคุม	38
3.9.4 การไหลของอักขรที่ออกแบบเอง	38
3.10 นาฬิกาเวลาจริง (Real time Clock)	39
3.10.1 หน้าที่และการใช้งานของแต่ละขา	40
3.10.2 การทำงานของ DS 1202	41
3.10.3 การกำหนดแอดเดรสและรูปแบบของคำสั่ง	41
3.10.4 การกำหนดให้เริ่มการทำงานและการควบคุมของ DS 1202	41
3.10.5 การรับส่งข้อมูลกับ DS 1202	41
3.10.6 ขั้นตอนการส่งข้อมูลให้ DS 1202	41
3.10.7 ขั้นตอนการรับข้อมูลให้ DS 1202	42
3.10.8 การจัดแอดเดรสใน DS 1202	43
3.11 แหล่งจ่ายไฟ 5 โวลท์	43

บทที่ 4 ผลการทดลอง	50
4.1 วงจร tone decoder	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 วงจร DTMF decoder	51
4.3 วงจรตรวจสอบขบวนการโทรศัพท์	53
4.4 ส่วนอ่านรหัสการคเจาะรฐ	54
4.5 ส่วนแสดงผลของโปรแกรมทางจอ LCD	54
บทที่ 5 สรุปและวิจารณ์	60
กิตติกรรมประกาศ	61
บรรณานุกรม	62



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูป 2.1 แสดงหน้าปิดปุ่มโทรศัพท์แบบกดและความถี่ที่ใช้	2
รูปที่ 2.2 แสดงรายละเอียดขา MT8870	7
รูปที่ 2.3 แสดงโครงสร้างภายในของ MT8870	8
รูปที่ 2.4 แสดงความถี่ที่ได้จากภาคกรองความถี่	8
รูปที่ 2.5 แสดงแผนภูมิเวลาของ MT8870	10
รูปที่ 2.6 แสดงการต่อวงจรผลิตความถี่	11
รูปที่ 2.7 แสดงการต่อวงจรดีเทคสัญญาณ busy	12
รูปที่ 2.8 แสดงสัญลักษณ์ฟิลิปฟลอปเจคและลักษณะของจริง	15
รูปที่ 2.9 แสดงการเปลี่ยนแปลงของสัญญาณนาฬิกาเมื่อเทียบกับการเปลี่ยนแปลงของเอาต์พุต Q	16
รูปที่ 2.10 แสดงรูปแบบของโปรแกรมภาษาแอสเซมบลี	17
รูปที่ 2.11 แสดงการต่ออิพროมอีเอ็มแอมโมเตอร์เข้ากับเครื่องคอมพิวเตอร์พีซีโดยใช้พอร์ตอนุกรม(RS 232)	19
รูปที่ 2.12 แสดงตำแหน่งต่าง ๆ บนบอร์ดอีอี - 232	22
รูปที่ 3.1 แสดงบล็อกไดอะแกรมของเครื่องควบคุมการใช้โทรศัพท์โดยใช้การ์ดชนิดเจาะรู	23
รูปที่ 3.2 แสดงลักษณะของการ์ดเจาะรูที่ออกแบบ	24
รูปที่ 3.3 แสดงวงจรส่วนอุปกรณ์อ่านรหัสการ์ด 1 ตำแหน่ง	25
รูปที่ 3.4 แสดงวงจร LED ที่ต่อกับตัวต้านทานเพื่อการจำกัดกระแส	25
รูปที่ 3.5 แสดงวงจร TONE DECODER	26
รูปที่ 3.6 แสดงวงจร DTMF DECODER	27
รูปที่ 3.7 แสดงวงจรตรวจสอบการวางหูและยกหูโทรศัพท์	27
รูปที่ 3.9 แสดงบอร์ดไมโครคอนโทรลเลอร์แอนท์ - 32	29
รูปที่ 3.10 แสดง DS1202	40
รูปที่ 3.11 แสดงบล็อกไดอะแกรมส่วนต่าง ๆ DS 1202	40
รูปที่ 3.12 แสดงรูปคลื่นของการรับและส่งข้อมูล	42
รูปที่ 3.13 แสดงวงจรแหล่งจ่ายไฟ 5 โวลต์ 1 แอมแปร์ที่ใช้	44
รูปที่ 4.1 แสดงสัญญาณที่วัดได้เมื่อทำการป้อนสัญญาณอินพุตเป็นรูปไซน์ที่มีความถี่ 400 เฮิร์ต	50
รูปที่ 4.2 แสดงสัญญาณที่วัดได้เมื่อทำการป้อนสัญญาณอินพุต เป็น รูปสี่เหลี่ยม ที่มีความถี่ 400 เฮิร์ต	51
รูปที่ 4.3 แสดงสัญญาณ DTMF เมื่อกดโทรศัพท์หมายเลข 1 และทำการวัดเอาต์พุตที่ขา Q1 จะมีค่า 5 โวลต์	52
รูปที่ 4.4 แสดงสัญญาณ DTMF เมื่อกดโทรศัพท์หมายเลข 2 และทำการวัดเอาต์พุตที่ขา Q1 จะมีค่า 0 โวลต์	52
รูปที่ 4.5 แสดงสัญญาณโทรศัพท์ขณะที่วางหูมีค่า 45.19 โวลต์	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 แสดงสัญญาณโทรศัพท์ขณะที่ยกหูมีค่า 6.401 โวลท์	53
รูปที่ 4.7 แสดงข้อความขณะเครื่องเริ่มทำงาน	54
รูปที่ 4.8 แสดงข้อความขณะทำการสอคการ์ดถูกต้อง	55
รูปที่ 4.9 แสดงข้อความขณะทำการตรวจสอบรหัสจากการ์ด	55
รูปที่ 4.10 แสดงข้อความเมื่อทำการดีเทคการ์ดถูกต้อง	56
รูปที่ 4.11 แสดงข้อความขณะที่การ์ดไม่ถูกต้อง	56
รูปที่ 4.12 แสดงเวลาขณะใช้โทรศัพท์	57
รูปที่ 4.13 แสดงข้อความขณะตรวจสอบสัญญาณโทรศัพท์	57
รูปที่ 4.14 แสดงข้อความขณะสายโทรศัพท์ไม่ว่าง	58
รูปที่ 4.15 แสดงข้อความขณะที่กำลังสนทนา	58
รูปที่ 4.16 แสดงค่าบริการการใช้โทรศัพท์	59
รูปที่ 4.17 แสดงเวลาที่ทำการใช้โทรศัพท์ทั้งหมด	59



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตาราง 2.1 แสดงอัตราค่าบริการโทรศัพท์ทางไกล	4
ตาราง 2.2 แสดงค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ	9
ตารางที่ 2.3 แสดงตารางเท็จจริงของฟิลิปฟลอปเจเค เบอร์ SN 7473	15
ตารางที่ 3.1 คอนโทรลโค้ดที่ใช้ในการกำหนดการทำงานของ 8255 โหมด 0	30
ตารางที่ 3.2 แสดงขาสัญญาณของ LCD MODULE	32
ตารางที่ 3.3 แสดงรายละเอียดของชุดคำสั่งควบคุม	33
ตารางที่ 3.4 แสดงตัวอย่างรูปแบบตัวอักษร	39
ตารางที่ 3.5 แสดงคำสั่งและการจัดแอดเดรส	43
ตารางที่ 4.1 แสดงผลการทดลองการวัดค่าที่ได้จากวงจรตรวจสอบสัญญาณ DTMF	51



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันกล่าวได้ว่าเป็นยุคแห่งเทคโนโลยีการติดต่อสื่อสารจนมีค่าเปรียบเทียบกับระบบการสื่อสารของประเทศว่าเปรียบเสมือนเป็นระบบเส้นประสาทที่รับความรู้สึกจากทุกส่วนของร่างกายให้กับสมอง และนำคำสั่งไปให้ส่วนของร่างกายปฏิบัติ คอบสนองต่อความรู้สึกนั้นโดยฉับพลัน ถ้าส่วนหนึ่งส่วนใดไม่มีระบบสื่อสารก็เหมือนกับว่าเป็นอัมพาต ซึ่งนอกจากจะทำให้ส่วนนั้นไม่มีประโยชน์แล้ว ยังเป็นภาระและอันตรายต่อส่วนอื่นของประเทศอีกด้วย ดังนั้นเราจะเห็นว่าความเจริญทางด้านระบบสื่อสารของประเทศเป็นตัวบอกถึงความเจริญของประเทศได้ ด้วยประโยชน์ของระบบสื่อสารที่ส่งผลต่อการพัฒนาประเทศ รัฐบาลของประเทศไทยก็ได้เล็งเห็นความสำคัญตรงจุดนี้ จึงพยายามมุ่งผลักดันนโยบายส่งเสริมการพัฒนาระบบสื่อสารของประเทศ

โทรศัพท์ก็เป็นอุปกรณ์สื่อสารอีกชนิดหนึ่งที่มีบทบาทในชีวิตประจำวัน ทำให้เราสามารถติดต่อสื่อสารกันได้โดยสะดวกรวดเร็ว ไม่ต้องเสียเวลาในการเดินทางเพื่อจะติดต่อพูดคุยธุระด้วยตนเองเหมือนครั้งก่อน ทำให้มีส่วนทางอ้อมในการแก้ปัญหาจราจรในปัจจุบัน และก็นื่องด้วยค่าบริการที่ไม่แพงจนเกินไปจึงได้รับความนิยมจากผู้ใช้งาน จนถึงได้ว่าโทรศัพท์เป็นอุปกรณ์ที่จำเป็นอย่างหนึ่งที่ทุกที่ต้องมี การใช้โทรศัพท์ของผู้ใช้นั้นจะมีการคิดค่าบริการแตกต่างกันไปตามช่วงเวลาที่ใช้บริการ ระยะทางที่ทำการติดต่อ รวมไปถึงเวลาที่ใช้บริการโทรศัพท์

การคิดค่าบริการนั้น ตามปกติทางองค์กรโทรศัพท์จะเป็นผู้ทำการคิดคำนวณ โดยที่ทางผู้ให้บริการจะไม่มีทางที่จะทราบค่าบริการรวมนี้เลย ถ้าไม่ได้มีการจดบันทึกการใช้โทรศัพท์ไว้โดยละเอียด และก็จะไม่มีทางทราบชื่อผู้ใช้อีกด้วย ถ้าโทรศัพท์เป็นโทรศัพท์ที่ต้องมีการใช้ร่วมกันหลายคน ก็ไม่สะดวกในการจะเก็บเงินกันในทันทีเพราะจะต้องรอใบเสร็จเก็บเงินจากองค์กรโทรศัพท์ในกรณีที่โทรออกนอกเขตพื้นที่ ซึ่งกว่าจะได้ก็เสียเวลานาน โดยโครงการนี้จะเป็นการควบคุมการใช้โทรศัพท์ ดังนั้นจะทำให้เราสามารถทราบชื่อผู้ให้บริการได้ และก็จะทำให้เราทราบเวลาการใช้โทรศัพท์ จำนวนเวลาการใช้ และค่าใช้บริการอีกด้วย

โดยที่ขั้นตอนการทำงานของโครงการนี้นั้น ได้แก่ วงจรส่วนตรวจสอบรหัสจากการ์ดชนิดเจาะรู วงจรตรวจสอบสัญญาณโทรศัพท์ วงจรถอดรหัส DTMF และ วงจรตรวจสอบการยกหู รวมทั้งส่วนที่เป็นไมโครคอนโทรลเลอร์ที่ใช้คอยควบคุมการทำงานของวงจรทั้งหมด และทำการคำนวณค่าบริการการใช้โทรศัพท์ ตรวจสอบเช็คเวลาการใช้และคำนวณเวลาอีกด้วย

ซึ่งการดำเนินงานโครงการในเทอมนี้ที่ได้ทำไปแล้ว ได้ทำการทดลอง การทำงานของวงจรต่าง ๆ อันได้แก่ วงจรตรวจสอบการยกหู วงจรตรวจสอบรหัสจากการ์ดชนิดเจาะรู วงจรตรวจสอบสัญญาณโทรศัพท์ รวมไปถึงการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ไปแล้วเพียงบางส่วน โดยที่ในเทอมหน้า จะทำการทดลองต่อจนสำเร็จ อันจะได้แก่ การเขียนโปรแกรมควบคุมที่เหลือทั้งหมด และการต่อวงจรทุกวงจรเข้ากับไมโครคอนโทรลเลอร์ เพื่อให้โครงการนี้สามารถทำงานได้ตามจุดประสงค์

เมื่อโครงการนี้ได้เสร็จสิ้นแล้วเราก็สามารถที่จะทำการตรวจเช็ครายการผู้ใช้โทรศัพท์และค่าบริการการใช้โทรศัพท์ได้โดยง่ายแล้วเพียงแต่ต่อเครื่องควบคุมการใช้โทรศัพท์นี้ไว้กับสายโทรศัพท์ตามปกติก็จะทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

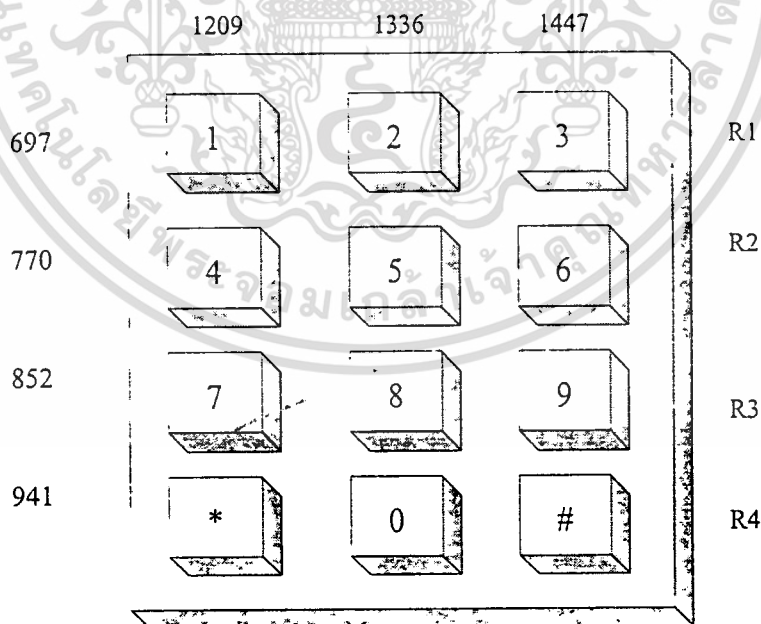
ทฤษฎีและหลักการ

2.1 ทัว ๆ ไปเกี่ยวกับระบบโทรศัพท์

โทรศัพท์ในปัจจุบันจะแบ่งตามลักษณะของการใช้งานออกเป็น 2 แบบ คือโทรศัพท์แบบหมุน และแบบกดปุ่ม ในที่นี้จะขอกล่าวเฉพาะแบบกดปุ่มเพียงอย่างเดียว เมื่อผู้ใช้ยกหูโทรศัพท์ จะได้ยินสัญญาณพร้อมโทรออกหรือเรียกว่าสัญญาณไดอัล โทน (Dial Tone) ซึ่งเป็นสัญญาณเสียงที่มีความถี่ 350 เฮิร์ต กับ 450 เฮิร์ต นำมาอดูเลตกัน หลังจากที่ใช้กดตัวเลขครบทุกตัวแล้ว ถ้าปลายทางสายว่างก็จะมีสัญญาณเรียกกลับมา (Ring Back tone) ซึ่งมีความถี่ 440 เฮิร์ต กับ 480 เฮิร์ต มอดูเลตกัน โดยจะดัง 1 วินาที และ เงียบ 4 วินาที สลับกันไป แต่ถ้าปลายทางสายไม่ว่าง ระบบโทรศัพท์ก็จะส่งสัญญาณสายไม่ว่าง (Busy tone) ซึ่งมีความถี่ 480 เฮิร์ต กับ 620 เฮิร์ต มอดูเลตกันมา โดยจะดัง 0.5 วินาที และ เงียบ 0.5 วินาที โดยปกติทัว ๆ ไป ในสายโทรศัพท์ขององค์การโทรศัพท์ฯ ขณะวางหู จะมีแรงดันไฟฟ้ากระแสตรง 48 โวลต์ เมื่อยกหูจะมีแรงไฟตกลงมาเหลือประมาณ 8-10 โวลต์ และสัญญาณเรียก (Ringing signal) จะเป็นไฟฟ้ากระแสสลับมีค่าประมาณ 100 โวลต์ ที่ความถี่ 25 เฮิร์ต ซึ่งจะดัง 1 วินาที และเงียบ 4 วินาที

2.2 โทรศัพท์ระบบกดปุ่ม (DTMF)

เครื่องใช้โทรศัพท์ระบบกดปุ่มจะใช้กรรมวิธีของไดอัล โทนมัลติฟ्रीควเอนซี (Dial Tone Multi Frequency : DTMF) ในการส่งเลขหมายโทรศัพท์ โดยทัว ๆ ไป หน้าปัดของเครื่องจะมี 12 ปุ่ม แบ่งเป็น 4 แถว และ 3 หลัก และในบางรุ่นจะมี 16 ปุ่ม โดยเพิ่มหลักที่ 4 ขึ้นมา ดังแสดงในรูปที่ 2.1



รูป 2.1 แสดงหน้าปัดปุ่มโทรศัพท์แบบกดและความถี่ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ที่ใช้ในแต่ละแถวและหลักจะมีความถี่ต่างกัน ซึ่งความถี่ของทั้ง 4 แถว เรียกว่า กลุ่มความถี่ต่ำ(Low Group Frequency) และความถี่ของทั้ง 3 หรือ 4 หลัก เรียกว่า กลุ่มความถี่สูง(High Group Frequency) การกดปุ่มที่หมายเลขใด ๆ จะทำให้วงจรภายในของเครื่องผลิตความถี่ออกมา 2 ความถี่ เช่นเมื่อกดเลข 5 ความถี่ที่ผลิตออกมาคือ 770 เฮิร์ต และ 1336 เฮิร์ต เป็นต้น

2.3 การคิดบริการของโทรศัพท์

โครงการนี้จะสามารถบันทึกการใช้โทรศัพท์โดยที่ต้นทุนจะอยู่ภายในเขตกรุงเทพฯ เท่านั้น หากต้องการนำไปใช้ในจังหวัดอื่น ๆ ก็สามารถทำได้โดยการแก้ไขโปรแกรมภายในเครื่อง

2.3.1 อัตราค่าบริการการใช้โทรศัพท์ขององค์การโทรศัพท์ แบ่งออกเป็น

1. ค่าบริการ โทรศัพท์ ภายในจังหวัด
2. ค่าบริการ โทรศัพท์ ทางไกลต่างจังหวัด
3. ค่าบริการ โทรศัพท์ ทางไกลต่างประเทศ

ซึ่งอัตราค่าบริการเหล่านี้ ยังมีส่วนลดตามเวลาที่ใช้ด้วย คือ

1. ในช่วงเวลา 06.00-18.00 น. ใช้อัตราค่าใช้บริการ 100 % ของอัตราเต็ม
2. ในช่วงเวลา 18.00-22.00 น. ใช้อัตราค่าใช้บริการ 50 % ของอัตราเต็ม
3. ในช่วงเวลา 22.00-06.00 น. ใช้อัตราค่าใช้บริการ 1 ใน 3 ของอัตราเต็ม

2.3.2 อัตราค่าบริการโทรศัพท์ในเขตกรุงเทพฯและปริมณฑล

การคิดค่าโทรศัพท์ในเขตกรุงเทพฯและปริมณฑล จะเสียค่าบริการในการใช้บริการ ได้ในราคา 3 บาท ต่อครั้ง แต่จะมีบางหน่วยงานที่ใช้เลขหมายพิเศษ ซึ่งผู้ที่โทรเข้าจะไม่ต้องเสียค่าบริการ ได้แก่

1. 13 บริการสอบถามเบอร์โทรศัพท์ในเขตนครหลวง
2. 183 บริการสอบถามเบอร์โทรศัพท์ในเขตภูมิภาค และเลขหมายของโทรศัพท์เคลื่อนที่
3. 17-xxx แจ้งโทรศัพท์ขัดข้อง
4. 189 ศูนย์รับร้องเรียนเรื่องโทรศัพท์
5. 193 ตำรวจทางหลวง
6. 197 ศูนย์ควบคุมจราจร
7. 191 แจ้งเหตุคว้นเหตุร้าย
8. 123 แจ้งเหตุคว้นเหตุร้าย
9. 199 ดับเพลิงศรีอยุธยา

และมีเลขหมายพิเศษบางชนิดที่ต้องเสียค่าบริการ 3 บาทต่อครั้ง

1. 1144 Pack Link
2. 1500 Easy Call
3. 151,152 Phonelink

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. 161,162 Hutchison Pagephone
5. 141,142 Worldpage
6. 1188 Postel
7. 125 แจ้งท่อประปาแตก
8. 181 เทียบเวลา
9. 182 ศูนย์บริการพยากรณ์อากาศ
10. 184 ศูนย์ร้องเรียน ขสมก.

2.3.3 อัตราค่าบริการโทรศัพท์ทางไกลต่างจังหวัด

อัตราค่าบริการโทรศัพท์ทางไกลต่างจังหวัด จะคิดราคาต่อนาที ซึ่งจะมีราคาค่าบริการโดยเริ่มต้นที่ราคา 3,6,9,12,15 และ 18 บาทต่อ 1 นาที เศษของนาทีคิดเป็น 1 นาที ซึ่งค่าใช้จ่ายต่อนาทีก็ขึ้นอยู่กับจังหวัดนั้น ๆ ซึ่งดูได้จากตาราง 2.1

รหัส	เลขนำ	บาท	จังหวัด
032	2,3	6	ราชบุรี
	4	6	เพชรบุรี
	5,6	12	ประจวบคีรีขันธ์
034	2,3	3	นครปฐม
	4	3	สมุทรสาคร
	5,6	9	กาญจนบุรี
	7	6	สมุทรสงคราม
035	2,3	6	อุรุษยา
	5	6	สุพรรณบุรี
	6	6	อ่างทอง
036	2,3	6	สระบุรี
	4	9	ลพบุรี
	5	9	สิงห์บุรี
037	2	6	ปราจีนบุรี
	3	6	นครนายก
038	2,3,4	6	ชลบุรี
	5	6	ฉะเชิงเทรา
	6	9	ระยอง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ แสดงอัตราค่าบริการโทรศัพท์ทางไกล อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัส	เลขนำ	บาท	จังหวัด
039	3,4	12	จันทบุรี
	5	12	ตราด
042	2,3	15	อุดรธานี
	4	18	หนองคาย
	5	18	นครพนม
	6	18	มุกดาหาร
	7	18	สกลนคร
	8	15	เลย
043	2,3,4	15	ขอนแก่น
	5	15	ร้อยเอ็ด
	7	15	มหาสารคาม
	8	15	กาฬสินธุ์
044	2,3,4	12	นครราชสีมา
	5	12	สุรินทร์
	6	12	บุรีรัมย์
	8	12	ชัยภูมิ
045	2,3,4	15	อุบลราชธานี
	6	15	ศรีสะเกษ
	7	15	ยโสธร
053	2,3	18	เชียงใหม่
	5	18	ลำพูน
	6	18	แม่ฮ่องสอน
	7	18	เชียงราย
054	2	18	ลำปาง
	4	18	พะเยา
	5,6	15	แพร่
	7	18	น่าน

ตาราง 2.1 (ต่อ) แสดงอัตราค่าบริการโทรศัพท์ทางไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัส	เลขหน้า	บาท	จังหวัด
055	2,3	12	พิษณุโลก
	4	15	อุตรดิตถ์
	5	15	ตาก
	6	15	สุโขทัย
	7	12	กำแพงเพชร
056	2,3	12	นครสวรรค์
	4	9	ชัยนาท
	5	9	อุทัยธานี
	6	12	พิจิตร
	7	12	เพชรบูรณ์
073	2	18	ยะลา
	3,4	18	ปัตตานี
	5	18	นราธิวาส
074	2,3,4,5	18	สงขลา
	6	18	พัทลุง
	7	18	สตูล
075	2	18	ตรัง
	3,4,5	18	นครศรีธรรมราช
	6	18	กระบี่
076	2,3	18	ภูเก็ต
	4	18	พังงา
077	2,3,4	18	สุราษฎร์ธานี
	5	15	ชุมพร
	8	15	ระนอง

ตาราง 2.1 (ต่อ) แสดงอัตราค่าบริการโทรศัพท์ทางไกล

2.4 สัญญาณพื้นฐาน

สัญญาณพื้นฐาน คือ สัญญาณที่เครื่องชุมสายโทรศัพท์แจ้งสภาวะต่าง ๆ ว่าควรทำอะไร ประกอบด้วย

1. สัญญาณให้หมุน (Dial Tone) ใช้เพื่อแสดงให้ผู้เรียกหมุนเลขหมายของผู้ที่ต้องการติดต่อได้
2. สัญญาณไม่ว่าง (Busy Tone) ใช้เพื่อเตือนผู้ใช้ว่าสายปลายทางไม่ว่าง ควรวางหูก่อนระยะเวลาหนึ่ง

แล้วจึงโทรไปใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สัญญาณกริ่งเรียก (Ring Tone) ใช้เพื่อเตือนว่ามีผู้เรียกเข้ามา

4. สัญญาณเรียกกลับ (Ring Back Tone) ใช้เพื่อเตือนว่าการติดต่อที่ได้กระทำได้เสร็จสมบูรณ์แล้ว

2.4.1 สัญญาณในการติดต่อกันระหว่างเครื่องส่งและเครื่องรับโทรศัพท์

เครื่องส่ง

1. ขณะที่ไม่มีกรงหูโทรศัพท์ จะมีศักดาตกรวมสายโทรศัพท์เป็นสัญญาณกระแสตรง 48 โวลต์
2. เมื่อมีการกรงหูโทรศัพท์ ศักดาไฟฟ้าจะลดลงเหลือประมาณ 6-8 โวลต์ พร้อมทั้งส่งสัญญาณให้หมุน ซึ่งเมื่อกรงรหัสเลขหมายแล้ว สัญญาณให้หมุนนี้จะหายไป
3. เมื่อกรงรหัสเลขหมายครบทั้ง 7 เลขหมายแล้ว รหัสความถี่ที่ส่งไปจะเป็นสัญญาณผสมระหว่าง 2 ความถี่ เป็นความถี่สูงและความถี่ต่ำผสมกัน โดยในแต่ละเลขหมายจะมี DTMF อยู่หนึ่งคู่
4. ขณะที่รอการรับสาย จะมีสัญญาณตอบกลับ 2 แบบ เพื่อจะบอกว่าขณะนี้สายว่างหรือไม่ ซึ่งก็คือสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง ตามลำดับ
5. เมื่อมีการรับสายแล้ว ศักดาไฟฟ้าจะอยู่ที่ 6-8 โวลต์ โดยมีการกระเพื่อมตามลักษณะของความถี่เสียง ความดังของเสียงพูดตามสาย
6. เมื่อเลิกสนทนา และ วางหู ขนาดของศักดาไฟฟ้าจะกลับ ไปอยู่ที่ 48 โวลต์ดังเดิม

เครื่องรับ

1. ขณะที่วางหูอยู่ จะมีขนาดของศักดาไฟฟ้าตกรวมอยู่ที่ 48 โวลต์
2. เมื่อมีสัญญาณกริ่งเรียก จะมีขนาดประมาณ 100 โวลต์ จังหวะ 1 วินาที หยุด 4 วินาที ซึ่งจะตรงกับสัญญาณเรียกกลับที่เครื่องส่ง
3. จากนั้นเมื่อมีผู้รับสายกรงหูโทรศัพท์ ขนาดของศักดาไฟฟ้าตกรวมจะอยู่ที่ 8 โวลต์ และมีการกระเพื่อมของสัญญาณตามขนาดและความถี่ของเสียงพูด
4. เมื่อวางหูโทรศัพท์ ขนาดของศักดาไฟฟ้าจะกลับ ไปอยู่ที่ 48 โวลต์ตามเดิม

2.5 MT8870 ไอซีถอดรหัสความถี่โทรศัพท์ (Integrated DTMF Receiver)

IN+	1	18	VDD
IN-	2	17	SVGT
GS	3	16	ESI
VRef	4	15	SI0
INH	5	14	O4
PWON	6	13	O3
OSC1	7	12	O2
OSC2	8	11	O1
VSS	9	10	TOE

รูปที่ 2.2 แสดงรายละเอียดขา MT8870

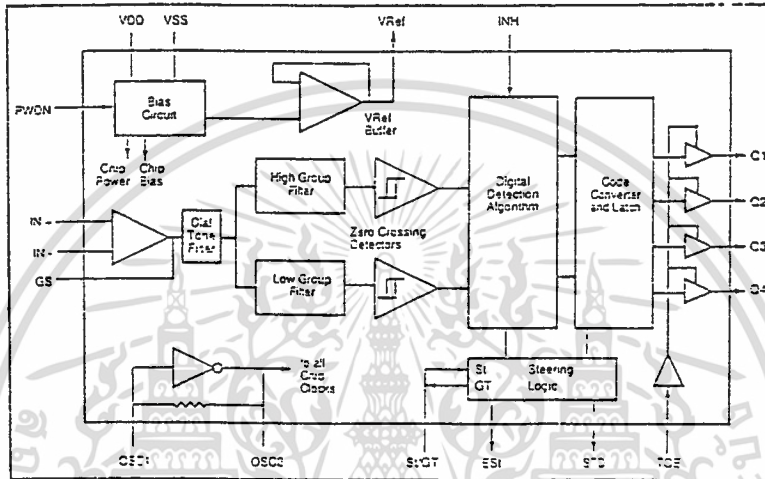
2.5.1 คุณสมบัติของ MT8870

- เป็นตัวรับและถอดรหัสความถี่ (DTMF Receiver)
- กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถปรับการ์ดไทม์ (Guard Time)
- เป็นไอซีคุณภาพดี

2.5.2 โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่ และวงจรถอดรหัสฟังก์ชันทางดิจิตอล เป็นไอซีที่สร้างขึ้นโดยเทคโนโลยี ISO'CMOS ในส่วนวงจรกรองความถี่ใช้เทคนิคสวิทช์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและกรองความถี่ต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอล เพื่อตรวจจับ และถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐาน 2 ขนาด 4 บิต และเช็คช่วงเวลาที่สำคัญเข้ามา ส่วนภาคอินพุทเป็นออปแอมป์ ซึ่งสามารถปรับอัตราขยายได้ โดยต่ออุปกรณ์ภายนอก เอาท์พุทเป็นวงจรแลตซ์ 3 สถานะ



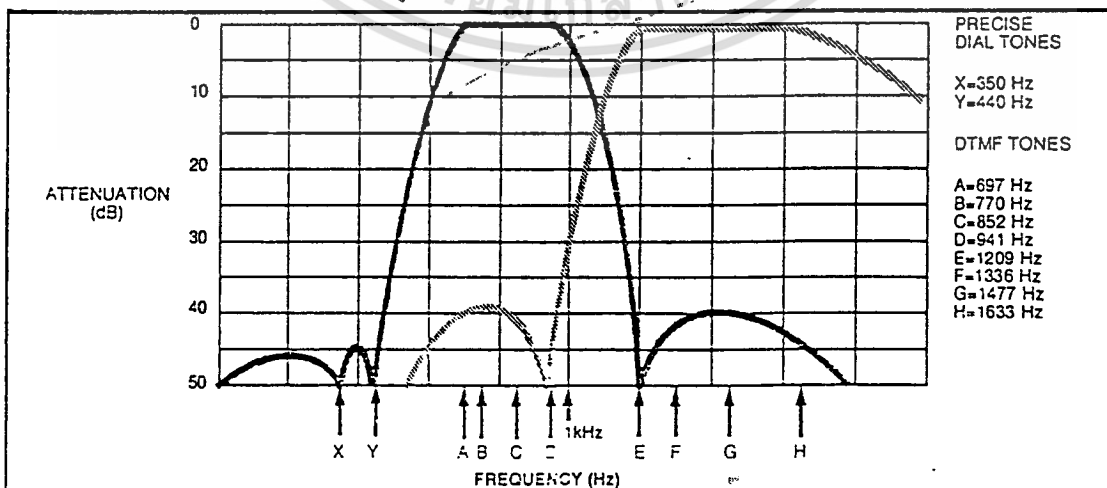
รูปที่ 2.3 แสดง โครงสร้างภายในของ MT8870

2.5.3 ฟังก์ชันการทำงานภายในของ MT8870

ประกอบด้วยส่วนสำคัญ 5 ส่วนคือ

2.5.3.1 ภาคกรองสัญญาณความถี่ (Filter Section)

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือ ช่วงความถี่สูง และช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิด สวิทช์คาปาซิเตอร์ (Six order switched capacitor and pass filter) ซึ่งความถี่ที่แยกได้มี 2 ช่วง คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ



รูปที่ 2.4 แสดงความถี่ที่ได้จากภาคกรองความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.2 ภาคถอดรหัส (Decoder Section)

ความถี่ DTMF ที่ถูกรองเรียบร้อยแล้ว จะผ่านเข้าวงจรถอดรหัสความถี่ ออกเป็นตัวเลข โดยใช้เทคนิคการนับทางดิจิทัล และมีการตรวจสอบความถี่ที่เข้ามาว่า เป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสม เมื่อตรวจสอบความถี่นั้นว่าถูกต้อง สัญญาณที่ขา Est (Early steering) ก็จะมีแอกทีฟ สำหรับค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ นั้นแสดงดังตาราง 2.2

F low	F high	NO	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1447	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1447	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1447	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1447	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

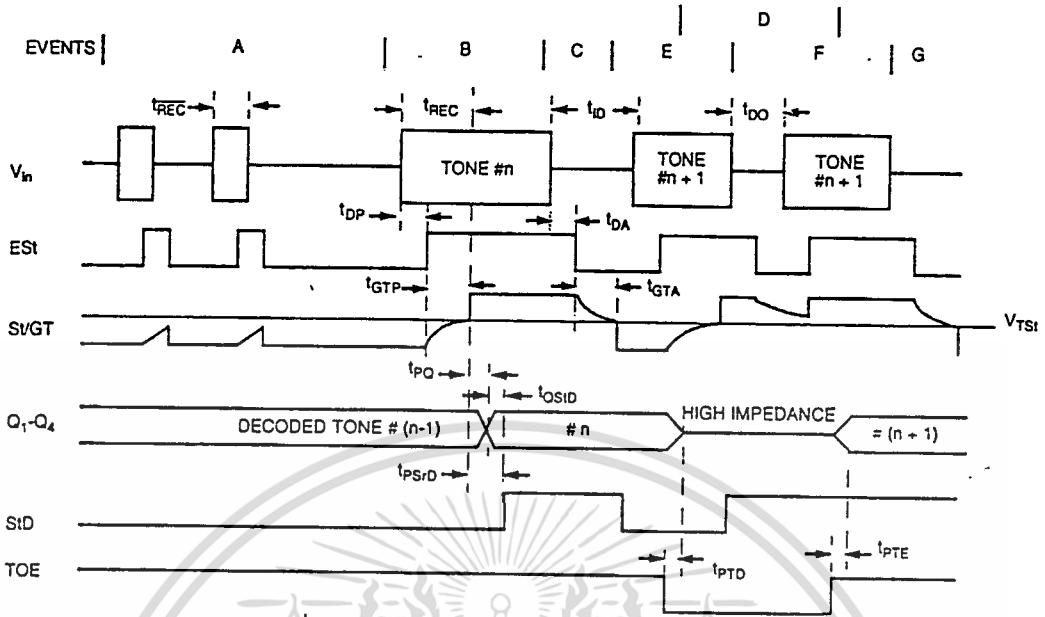
ตาราง 2.2 แสดงค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ

2.5.3.3 ภาคตรวจสอบสัญญาณ (Steering Circuit)

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาท์พุท จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมามีเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลานานเท่าไรนั้น สามารถตั้งโดยใช้ RC ต่อภายนอก สัญญาณที่ขา Est จะเป็น High ทำให้ V_c สูงขึ้น ตัวเก็บประจุ C จะคายประจุ ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดัน V_c สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสออกเป็นตัวเลขขนาด 4 บิตดังแสดงได้ดังแผนภูมิเวลา (Timing diagram) ดังรูปที่ 2.5



รูปที่ 2.5 แสดงแผนภูมิเวลาของ MT8870

ขั้นตอนการทำงาน

- A - ตรวจพบความถี่เข้ามา แต่คาบเวลาที่ถูกต้อง เอาท์พุทไม่เปลี่ยน
- B - ความถี่ #n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และเลขชี้ไว้ที่เอาท์พุท
- C - จบความถี่ #n ช่วงห่างถูกต้อง เอาท์พุทยังคงเลขชี้อยู่นกว่าจะได้รับความถี่ที่ถูกต้องใหม่
- D - เอาท์พุทเปลี่ยนเป็นไฮอิมพีแดนซ์
- E - ความถี่ #n+1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและเลขชี้ไว้
- F - ความถี่ #n+1 หายไป ช่วงห่างไม่ถูกต้อง เอาท์พุทยังเลขชี้อยู่
- G - จบความถี่ #n+1 ช่วงห่างถูกต้อง เอาท์พุทยังเลขชี้อยู่จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายศัพท์ที่ใช้

- V_{in} - สัญญาณความถี่ DTMF ที่เข้ามา
- Est - ใช้แสดงความถี่ที่ถูกต้อง
- St/Gt - Steering input / Guard time output สำหรับต่อกับ RC ภายนอก
- Q_1-Q_4 - เอาท์พุท BCD ขนาด 4 บิต
- Std - Delayed Steering output ใช้แสดงความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อ

แสดงความถูกต้องของสัญญาณ

- TOE - Tone output enable (input) ใช้ควบคุม Q_1-Q_4 ให้เป็นไฮอิมพีแดนซ์
- t_{REC} - คาบเวลานานที่สุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง
- t_{REC} - คาบเวลาสั้นที่สุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง
- t_{ID} - เวลาสั้นที่สุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

t_{DO} - เวลานั้นที่สุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{DA} - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{GTP} - การ์ดใหม่ของการปรากฏความถี่ DTMF

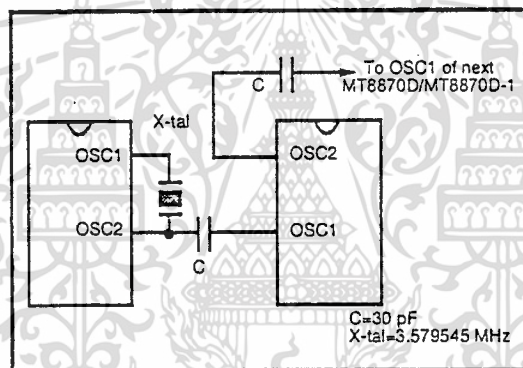
t_{GTA} - การ์ดใหม่ของการหายไปของความถี่ DTMF

2.5.3.4 ภาคขยายสัญญาณความแตกต่าง

วงจรส่วนอินพุทของ MT8870 เป็นภาคขยายออปแอมป์ ที่สามารถปรับอัตราขยายโดยต่อวงจรภายนอกเข้าไปด้วย ซึ่งวงจรที่นำมาตอนนี้แล้วแต่ความต้องการ

2.5.3.5 ภาคกำเนิดความถี่ (Oscillator)

ในภาคนี้ภายในไอซีจะมีวงจรเวลาอยู่ภายใน เพียงแต่ต่อคริสตัลอสซิลเลเตอร์ขนาด 3.58 เมกะเฮิร์ต ก็สามารถใช้งานได้ทันที การต่อวงจรกำเนิดความถี่แสดงดังรูปที่ 2.6



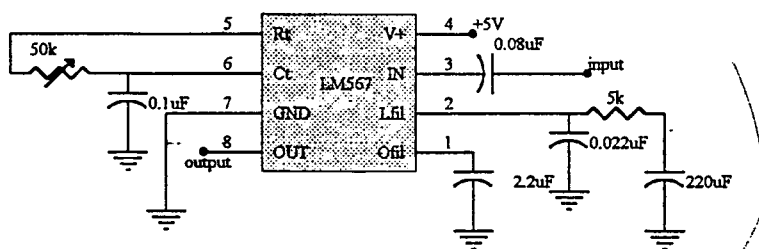
รูปที่ 2.6 แสดงการต่อวงจรผลิตความถี่

2.6 วงจรดีเทคสัญญาณสายไม่ว่าง (Busy Tone)

เราใช้ไอซี โทเนดีเทคเตอร์ (Tone Detector) LM567 ทำการดีเทคสัญญาณบิวซี (Busy) ที่มีลักษณะของสัญญาณเป็นสัญญาณความถี่ ประมาณ 425 เฮิร์ต ส่งมาเป็นช่วง ๆ คือ ส่ง 0.5 วินาที หยุด 0.5 วินาที โดยสัญญาณ busy ที่ใช้จะนำมาจากคู่สายโทรศัพท์ ผ่านวงจรขยายและวงจรบัฟเฟอร์ จากนั้นจึงป้อนสัญญาณนี้เข้าสู่ LM 567 เพื่อทำการดีเทคสัญญาณ โดยลักษณะเอาต์พุตจะเป็นรูปพัลส์ที่แสดงค่า 1 ในช่วงหยุดของสัญญาณ และจะแสดงค่า 0 ในช่วงส่งของสัญญาณ ในการกำหนดค่าความถี่ที่จะทำการดีเทคสัญญาณของ LM 567 ซึ่งในที่นี้ก็คือ ความถี่ของสัญญาณ Busy ความถี่ 425 เฮิร์ต ทำได้โดยการปรับเปลี่ยนค่า R_1 และ C_1 ตามสูตร

$$f_0 = \frac{1}{(1.1)R_1C_1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงการต่อวงจรตีเทคสัญญาณ busy

2.7 โปรแกรมซับรูทีน

โปรแกรมซับรูทีน (subroutine) เป็นชุดคำสั่งของโปรแกรมที่สามารถจะถูกเรียกใช้งานได้หลายครั้งในโปรแกรม ในโปรแกรมแอสเซมเบลอร์ของ MCS-51 สามารถเรียกซับรูทีนได้ 2 วิธี คือ

ACALL addr11

LCALL addr16

คำสั่งทั้งสองมีการทำงานที่คล้ายกันคือให้กระโดดไปยังซับรูทีน ณ ตำแหน่งต่างๆ บนหน่วยความจำ ความหมายของ addr11 คือเป็นการเรียกโดยระบุตำแหน่งแอดเดรสเพียง 11 บิต และความหมายของ addr16 คือเป็นการเรียกโดยระบุตำแหน่งแอดเดรส 16 บิต ซึ่งการอ้างตำแหน่งโดยใช้บิตไม่เท่ากันนี้เป็นข้อแตกต่างของคำสั่ง ACALL และ LCALL หลังจากโปรแกรมทำงานในซับรูทีนเป็นที่เรียบร้อย ไมโครคอนโทรลเลอร์ก็จะกลับมาประมวลผลคำสั่งต่อไปหลังจากที่ผ่านการเรียกแล้ว

ACALL (absolute call) คำสั่งนี้จะถูกนำมาใช้เพื่อเรียกใช้งานซับรูทีนเมื่อแอดเดรสของซับรูทีนนี้อยู่ในช่วงแอดเดรส 2 กิโลไบต์ หรือ ใน 1 เพจ ซึ่งหน่วยความจำทั้งหมด 64 กิโลไบต์ (แอดเดรส 0000H-FFFFH) จะมีการแบ่งออกเป็น 32 เพจ หรือช่วงกว้างเพจละ 800 H โดยกำหนดแอดเดรสของเพจได้ตั้งแต่เพจที่ 00H จนถึง 1FH เห็นได้ว่าการอ้างแอดเดรสของเพจจะใช้บิตอ้างอิงสูงสุดเพียง 5 บิต ดังนั้นคำสั่ง ACALL จึงต้องการบิตแอดเดรสอีกเพียง 11บิต จึงสรุปได้ว่า ACALL มีข้อแม้ในการอ้างแอดเดรสคือ อ้างได้ในตำแหน่งแอดเดรสของเพจที่ตนเองอยู่เท่านั้นในแต่ละช่วงของแอดเดรส 2 กิโลไบต์

ตัวอย่างเช่น ขณะนี้ใช้คำสั่ง ACALL อยู่ที่แอดเดรส 9A12H ดังนั้นการเรียกใช้ซับรูทีนได้ต้องอยู่ในช่วงเพจที่ 13H คือแอดเดรส 9800H-9FFFH เท่านั้น ข้อดีของคำสั่ง ACALL คือประหยัดเนื้อที่ของหน่วยความจำได้ เพราะมันต้องการออปเจ็คโค้ดเพียง 2 ไบต์เท่านั้น ดังนั้นถ้ามีการเรียกใช้คำสั่ง CALL บ่อยครั้งโดยที่สามารถเรียกใช้ด้วยคำสั่ง ACALL มันจะช่วยประหยัดเนื้อที่ได้มากกว่าทีเดียว

LCALL (long call) คำสั่งนี้แตกต่างจากคำสั่ง ACALL คือถ้ามีการเรียกใช้คำสั่งนี้จะสามารถเรียกใช้ซับรูทีนได้ตลอด 64 กิโลไบต์บนหน่วยความจำ ดังนั้นจึงต้องการบิตแอดเดรสทั้งหมด 16 บิต คำสั่ง LCALL นี้มักถูกใช้งานเมื่อโปรแกรมขนาดใหญ่มีการเรียกใช้ซับรูทีนที่อยู่ตำแหน่งที่ห่างมากจนไม่สามารถใช้คำสั่ง ACALL ได้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานคำสั่ง ACALL และ LCALL นี้จะถูกตรวจสอบโดยโปรแกรมแอสเซมเบลอร์ ซึ่งถ้ามีการผิดพลาดเกิดขึ้น แอสเซมเบลอร์จะทำการแจ้งให้ทราบทันที

RET (return) คำสั่งนี้จะถูกใช้งานเมื่อสิ้นสุดโปรแกรมชั่วคราวเพื่อกลับไปสู่บรรทัดที่คำสั่งทำการเรียกรูทีนนั้น

2.7.1 คำสั่งกระโดด

นอกจากการเรียกใช้งานชั่วคราวแล้วยังมีคำสั่งอีกส่วนหนึ่งที่ต้องใช้งานบ่อยครั้งก็คือ คำสั่งกระโดด (jump instructions) ในที่นี้จะขออธิบายคำสั่งกระโดดที่ใช้งานทั่วไป 3 คำสั่งนี้

SJMP rel

AJMP addr11

LJMP addr16

เมื่อไมโครคอนโทรลเลอร์ได้รับคำสั่งเหล่านี้มันจะกระโดดไปทำงานยังตำแหน่งแอดเดรสที่ระบุโดยไม่มีเงื่อนไข การอ้างตำแหน่งแอดเดรสโดยการใช้คำสัมพัทธ์ (rel) เป็นการอ้างตำแหน่งโดยอาศัยแอดเดรสที่ใช้คำสั่งนี้เป็นหลักและกระโดดไปยังตำแหน่งที่อยู่ระหว่าง -128 ถึง +127 แอดเดรสได้ การใช้ตัวอักษร A และ L หน้า JMP ในที่นี้มีความหมายเช่นเดียวกับที่ใช้ในคำสั่ง CALL สำหรับคำสั่ง SJMP (short jump) จะอาศัยคำสัมพัทธ์ในการอ้างแอดเดรสเพื่อกระโดดเข้าไปทำงาน หรือกล่าวได้ว่ากระโดดได้อยู่ในช่วง -128 ถึง +127 ไบต์ในหน่วยความจำ ข้อดีของคำสั่งนี้ก็คือประหยัดออปเจ็คโค้ดที่ใช้งานลงอย่างมากเมื่อต้องมีการกระโดดระยะสั้นๆ บ่อยครั้ง

นอกจากคำสั่งการกระโดดแบบไม่มีเงื่อนไขแล้ว ในชุดคำสั่งเกี่ยวกับการกระโดดนี้ยังมีคำสั่งกระโดดแบบมีเงื่อนไขด้วย ซึ่งทั้งหมดจะเป็นการกระโดดแบบใช้คำสัมพัทธ์เท่านั้น แต่ละคำสั่งก็จะมีเงื่อนไขเฉพาะตนเอง

2.7.2 การกำหนดตำแหน่งแอดเดรส

การกำหนดตำแหน่งแอดเดรสของชั่วคราวทำได้โดยการตั้งชื่อ (LABELS) ให้กับมัน การใช้ชื่อมีความหมายสอดคล้องกับรูทีนที่เรียกใช้ซึ่งทำให้ง่ายในการวิเคราะห์โปรแกรมและไม่สับสน ไฟล์เอกสารที่สร้างแอสเซมเบลอร์จะมีการระบุแอดเดรสที่ถูกต้องของชื่อที่ตั้งขึ้นแต่ละตัวใน ตารางสัญลักษณ์ (symbol table) ด้วย ซึ่งตารางนี้ถูกสร้างขึ้นโดยอัตโนมัติเมื่อทำการสร้างไฟล์เอกสาร โดยโปรแกรมแอสเซมเบลอร์ แอสเซมเบลอร์บางโปรแกรมกำหนดให้ชื่อมีความยาวไม่เกิน 6 ตัวอักษร แต่สำหรับแอสเซมเบลอร์ EASM51 นั้นกำหนดให้ตั้งชื่อได้สูงสุดไม่เกิน 8 ตัวอักษร

2.7.3 วิธีการอ้างอิงแอดเดรส

1. การอ้างอิงแอดเดรสด้วยรีจิสเตอร์ (Register Addressing)

แอกคิวมูลเตอร์ A หรือรีจิสเตอร์ R_0 - R_7 สามารถถูกใช้เป็นได้ทั้งตำแหน่งของปลายทางและต้นทางของตัวโอเปอเรนด์ นอกจากนี้ยังมีรีจิสเตอร์อื่นๆ เช่น B, DPTR เป็นต้น ที่สามารถทำหน้าที่นี้ได้เช่นกัน การอ้างอิงแอดเดรสในโหมดนี้จะทำงานในรีจิสเตอร์เบงก์ที่ถูกเลือกใช้งานเท่านั้น ไม่อนุญาตให้เข้าไปใช้ประโยชน์ด้านการคำนวณเป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น ไม่อนุญาตให้เข้าไปใช้ประโยชน์ด้านการคำนวณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การอ้างแอดเดรสด้วยค่าตำแหน่งแบบทันที (Immediate addressing)

การอ้างแอดเดรสในโหมดนี้ถูกใช้งานเมื่อตัวโอเปอเรนด์เป็นค่าคงที่ค่าหนึ่งซึ่งโปรแกรมแอสเซมเบลอร์จะรู้ว่ามันเป็นค่าคงที่โดยการใส่เครื่องหมาย # นำหน้าค่าคงที่นั้น ค่าคงที่ที่ใช้มีอยู่ 4 รูปแบบ คือ ค่าคงที่เป็นเลขฐานสิบ เป็นเลขฐานสิบหก (เห็นได้ว่าการใส่อักษร H ต่อท้ายไว้ด้วย) เป็นเลขฐานสอง (เห็นได้ว่าการใส่อักษร B ต่อท้ายไว้ด้วย) หรือค่าคงที่ที่อยู่ในรูปของชื่อ

แอสเซมเบลอร์สามารถทำการแยกความแตกต่างระหว่างชื่อและเลขฐานสิบหกได้ด้วยการดูอักษรเริ่มต้น ถ้าเป็นเลขฐานสิบหกทุกครั้งต้องมีเลขศูนย์นำหน้าเสมอ ตัวอย่างเช่น E0H เป็นชื่อ และ 0E0H เป็นค่าคงที่เลขฐานสิบหกหรือค่า 224 ในเลขฐานสิบ ซึ่งในจุดนี้ผู้เขียนโปรแกรมต้องระมัดระวังการใช้งานด้วย

3. การอ้างแอดเดรสโดยตรง (Direct Addressing)

การอ้างแอดเดรสโดยตรงเป็นการอ้างตำแหน่งเพื่อเข้าถึงข้อมูลภายในหน่วยความจำแรมภายในของไมโครคอนโทรลเลอร์โดยการระบุโดยตรง ซึ่งจะรวมถึงรีจิสเตอร์ใช้งานพิเศษ (SFR) ด้วย ในส่วนนี้มีหน่วยความจำขนาด 128 ไบต์ ดังนั้นจึงอ้างได้ทั้งหมด 128 ตำแหน่ง (00H-7FH) และสามารถเข้าถึงรีจิสเตอร์พิเศษตัวใดตัวหนึ่งได้ เช่น แอ็กคิวมูเลเตอร์ซึ่งเป็นรีจิสเตอร์พิเศษตำแหน่งอยู่ที่ตำแหน่งแอดเดรส 0E0H ได้มีการกำหนดค่าคงที่และตั้งชื่อว่า ACC

รีจิสเตอร์อีกตัวหนึ่งที่น่าสนใจก็คือ DPTR (data pointer) ซึ่งมีขนาดถึง 16 บิต มันจึงสามารถอ้างได้เต็มที่ 64 กิโลไบต์ บนหน่วยความจำข้อมูลหรือโปรแกรมได้ DPTR ประกอบด้วยรีจิสเตอร์พิเศษขนาด 8 บิต 2 ตัว ด้านไบต์ต่ำหรือ DPL อยู่ที่ตำแหน่ง 082H และด้านไบต์สูงหรือ DPH ที่ 083H

4. การอ้างแอดเดรสโดยอ้อม (Indirect addressing)

การอ้างแอดเดรสในโหมดนี้เป็นการอ้างแอดเดรสโดยใช้ค่าที่บรรจุอยู่ในรีจิสเตอร์ R_0 และ R_1 ซึ่งจะใช้สัญลักษณ์ @ นำหน้าและไม่สามารถใช้รีจิสเตอร์ R_2 - R_7 มาใช้งานในโหมดนี้ได้ ตัวอย่างเช่นรีจิสเตอร์ R_0 มีข้อมูลค่า 43H เมื่อใช้คำสั่ง @ R_0 จะหมายถึงการอ้างแอดเดรสที่ตำแหน่ง 43H ซึ่งเป็นหน่วยความจำภายใน

5. การอ้างแอดเดรสในหน่วยความจำโปรแกรม (Program memory addressing)

ไมโครคอนโทรลเลอร์สามารถทำการอ่านข้อมูลได้เพียงอย่างเดียวจากหน่วยความจำโปรแกรม ซึ่งโดยปกติจะบันทึกอยู่ในรอมหรืออีพรอม การอ้างแอดเดรสในโหมดนี้ใช้คำสั่ง MOVC ตัวอักษร C ที่ต่อท้ายหมายถึงโค้ดแอดเดรส (code memory) ปลายทางของคำสั่ง MOVC จะเป็นแอ็กคิวมูเลเตอร์ ส่วนด้านต้นทางอ้างตำแหน่งแอดเดรสได้จากตำแหน่งแอดเดรสที่ได้จากตำแหน่งแอดเดรสที่ระบุโดยแอ็กคิวมูเลเตอร์ร่วมกับค่าของรีจิสเตอร์ DPTR โดยมีรูปแบบคำสั่งดังนี้

```
MOVC A, @A+DPTR
```

หรืออาจใช้รีจิสเตอร์ PC (program counter):

```
MOVC A, @A+PC
```

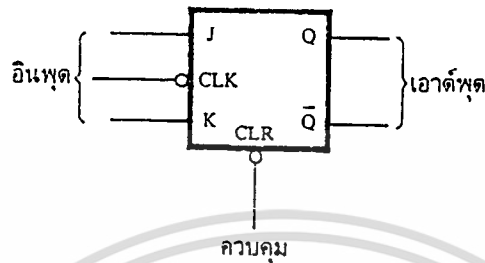
6. การอ้างแอดเดรสในหน่วยความจำข้อมูลภายนอก (External data memory addressing)

คำสั่งที่ใช้งานสำหรับการอ้างแอดเดรสในหน่วยความจำข้อมูลภายนอกคือ MOVX (move external) ซึ่งโดยปกติเป็นหน่วยความจำชนิดแรม การอ้างแอดเดรสจะใช้การระบุตำแหน่งโดยข้อมูลที่บรรจุอยู่ใน DPTR หรือ @DPTR ซึ่งการใช้ DPTR มันจะสามารถอ้างแอดเดรสได้ถึง 16 บิต หรือตลอดย่าน 64 กิโลไบต์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คล้ายกับการอ้างแอดเดรสโดยใช้ @R₀ และ @R₁ แต่ว่ารีจิสเตอร์ทั้งสองมีขนาดเพียง 8 บิต จึงอ้างแอดเดรสได้ในช่วงที่แคบกว่า

2.8 ฟลิปฟลอปजेค

ฟลิปฟลอปजेคที่ใช้ในครั้งนี้เป็นไอซีเบอร์ SN7473 ซึ่งประกอบด้วยฟลิปฟลอปजेคจำนวน 2 ตัว (dual JK - flip flop)



รูปที่ 2.8 แสดงสัญลักษณ์ฟลิปฟลอปजेคและลักษณะของจริง

2.8.1 รายละเอียดของขาต่อ

ขาอินพุต หมายถึง ขาที่ใช้ป้อนสัญญาณหรือแรงดันทางเข้า ซึ่งจะมีด้วยกัน 3 ขา คือ ขา J ขา K และขา CLK ซึ่งเรียกกันว่าขาคล็อก

ขาเอาต์พุต หมายถึง ขาที่ใช้ต่อสัญญาณหรือแรงดันทางออก จะมีด้วยกัน 2 ขา คือ ขา Q และขา \overline{Q}

ขาคววม หมายถึง ขา CLR ซึ่งเรียกกันว่าขาเคลียร์ ใช้ในการยกเลิกการแสดงผลที่ขาเอาต์พุต

หมายเหตุ ที่ขา CLK และขา CLR มีสัญลักษณ์รูป 0 อยู่ตรงขาต่อ สัญลักษณ์นี้แสดงให้ทราบว่า ถ้าสัญญาณหรือแรงดันอินพุตเปลี่ยนแปลงจากบวกเป็นลบจะทำให้ฟลิปฟลอปजेค เกิดการเปลี่ยนแปลงสถานะจากที่เป็นอยู่เดิม อาจกล่าวได้ว่า ลักษณะที่วงจรเกิดการเปลี่ยนแปลงสถานะเมื่อสัญญาณหรือแรงดันเปลี่ยนจากบวกเป็นลบนี้ เรียกว่า โลว์แอคทีฟ (low active)

อินพุต				เอาต์พุต	
CLR	CLK	J	K	Q	\overline{Q}
L	X	X	X	L	H
H		L	L	Q	\overline{Q}
H		H	L	H	L
H		L	H	L	H
H		H	H	TOGGLE	

ตารางที่ 2.3 แสดงตารางเท็จจริงของฟลิปฟลอปजेค เบอร์ SN 7473

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ 1

เมื่อขา CLR อยู่ในสถานะต่ำจะทำให้ Q มีสถานะต่ำ และขา Q มีสถานะสูง ไม่ว่าจะขา CLK ขา J และขา K จะมีสถานะเป็นอะไรก็ตาม (X = DON'T CARE)

กรณีที่ 2

เมื่อขา CLR อยู่ในสถานะสูง ขาเอาต์พุต Q หรือ Q จะมิสถานะเป็นสูง ตามขา J หรือขา K ก่อนที่ขา J และขา K จะมีสถานะเป็นต่ำทั้งคู่ ตัวอย่าง

1. ถ้าสถานะเดิม ขา J มีสถานะต่ำ ขา K มีสถานะสูง เมื่อขา J และขา K เปลี่ยนสถานะเป็นต่ำทั้งคู่ เมื่อป้อนพัลส์หรือสัญญาณนาฬิกาเข้าไปจะทำให้ขาเอาต์พุต Q มีสถานะเป็นต่ำ และ Q มีสถานะเป็นสูง ค้างอยู่เช่นนี้ตลอดไป

2. ถ้าสถานะเดิม ขา J มีสถานะสูง ขา K มีสถานะต่ำ เมื่อขา J และขา K เปลี่ยนสถานะเป็นต่ำทั้งคู่ เมื่อป้อนพัลส์หรือสัญญาณนาฬิกาเข้าไปจะทำให้ขาเอาต์พุต Q มีสถานะเป็นสูง และ Q มีสถานะเป็นต่ำ ค้างอยู่เช่นนี้ตลอดไป

3. ถ้าสถานะเดิม ขา J และขา K มีสถานะสูงทั้งคู่ เมื่อขา J และขา K เปลี่ยนสถานะเป็นต่ำทั้งคู่ เมื่อป้อนพัลส์หรือสัญญาณนาฬิกาเข้าไปจะทำให้ขาเอาต์พุต Q มีสถานะเป็นสูง และ Q มีสถานะเป็นต่ำ ค้างอยู่เช่นนี้ตลอดไป

กรณีที่ 3

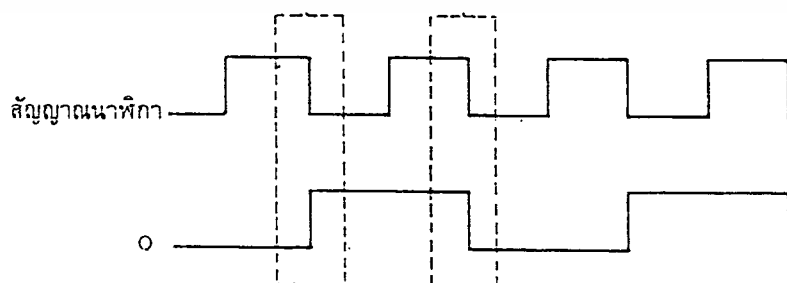
เมื่อขา CLR อยู่ในสถานะสูง ขา J มีสถานะสูง และขา K มีสถานะต่ำ จะทำให้ขา Q มีสถานะเป็นสูง และขา Q มีสถานะเป็นต่ำ ค้างอยู่อย่างนี้ ถึงแม้ว่าจะมีพัลส์หรือสัญญาณนาฬิกาเข้ามาก็ตาม

กรณีที่ 4

เมื่อขา CLR อยู่ในสถานะสูง ขา J มีสถานะต่ำ และขา K มีสถานะเป็นสูง จะทำให้ขา Q มีสถานะเป็นต่ำ และขา Q มีสถานะเป็นสูง ค้างอยู่อย่างนี้ ถึงแม้ว่าจะมีพัลส์หรือสัญญาณนาฬิกาเข้ามาก็ตาม

กรณีที่ 5

เมื่อขา CLR ขา J และขา K มีสถานะเป็นสูง ทุกขาจะทำให้ขา Q และขา Q สลับกันมีสถานะเป็นต่ำ และสถานะสูง ในขณะที่มีพัลส์หรือสัญญาณนาฬิกาป้อนเข้ามา ในกรณีนี้ขอให้สังเกตช่วงการเปลี่ยนแปลงของสัญญาณนาฬิกาเมื่อเปลี่ยนจากสูงเป็นต่ำจะทำให้ฟลิปฟล็อปเปลี่ยนสถานะไปจากเดิม



รูปที่ 2.9 แสดงการเปลี่ยนแปลงของสัญญาณนาฬิกาเมื่อเทียบกับการเปลี่ยนแปลงของเอาต์พุต Q เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

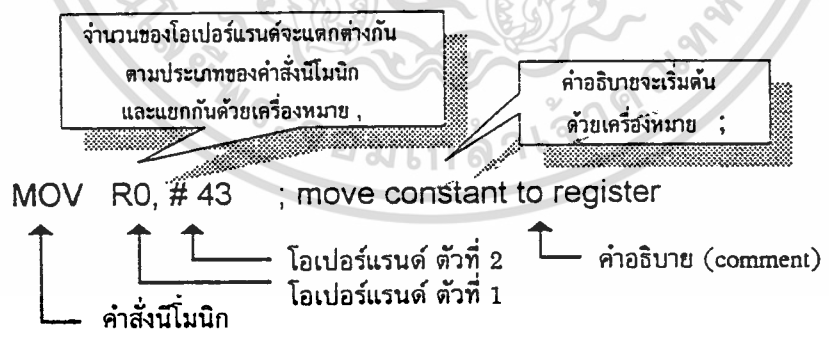
- สัญญาณนาฬิกาเปลี่ยนจากบวกเป็นลบทำให้แรงดันเอาต์พุต Q เปลี่ยนจากต่ำเป็นสูง
- สัญญาณนาฬิกาเปลี่ยนจากสูงเป็นต่ำ ทำให้แรงดันเอาต์พุต Q เปลี่ยนจากสูงเป็นต่ำ

2.9 การพัฒนาโปรแกรมที่ใช้กับไมโครคอนโทรลเลอร์ด้วยเครื่องไมโครคอมพิวเตอร์

โดยทั่วไปโปรแกรมเพื่อการศึกษาทดลองนั้น มักจะเป็นเพียงโปรแกรมขนาดเล็กและไม่มีความซับซ้อนนัก เพื่อที่จะสามารถทำความเข้าใจได้ในขั้นต้น แต่สำหรับการพัฒนาโปรแกรมในการประยุกต์ใช้งานจริงจะมีรายละเอียดและความซับซ้อนขึ้นอีกมาก ดังนั้นผู้พัฒนาโปรแกรมที่มีเครื่องไมโครคอมพิวเตอร์ (หรือที่เรียกกันโดยทั่วไปว่าเครื่องพีซี) จึงมักที่จะนำซอฟต์แวร์อื่น ๆ เข้ามาช่วยในการพัฒนาโปรแกรมเสมอ เช่น โปรแกรมเอดิเตอร์ (Editor Program) สำหรับการพิมพ์ตัวโปรแกรมแอสเซมบลีที่สร้างขึ้นให้จัดเก็บอยู่บนเครื่อง และโปรแกรมแอสเซมเบลอร์ (Assembler Program) สำหรับการแปลความโปรแกรมแอสเซมบลีนั้น ให้เป็นโปรแกรมเลขฐานสิบหกแทนการแปลความจากการเปิดตารางชุดคำสั่งด้วยตนเอง เป็นต้น ทั้งนี้เพื่อต้องการที่จะให้การสร้างโปรแกรมเป็นไปอย่างสะดวกรวดเร็วและถูกต้องมากที่สุดนั่นเอง

2.9.1 โครงสร้างภายในโปรแกรมต้นฉบับ

การพิมพ์ข้อมูล โปรแกรมที่เขียนขึ้นให้จัดเก็บอยู่ในเครื่องคอมพิวเตอร์ โดยใช้โปรแกรมเอดิเตอร์ใด ๆ ก็ตาม ซึ่งเรียกว่าโปรแกรมต้นฉบับ (Source Program) นั้นควรจะต้องมีความเข้าใจพื้นฐานเกี่ยวกับข้อกำหนดและรูปแบบของโปรแกรมภาษาแอสเซมบลีก่อนว่า โครงสร้างของแต่ละบรรทัดภายในโปรแกรมนั้นจัดแบ่งออกเป็นกลุ่ม ๆ โดยใช้ช่องว่างหนึ่งหรือหลายตัวก็ได้ เพื่อที่ตัวแปลความจะได้ทราบว่ากลุ่มใดเป็นคำสั่งนิมोनิก (Mnemonics) โอเปอเรนด์ (Operand) หรือคำอธิบาย (comment)



รูปที่ 2.10 แสดงรูปแบบของโปรแกรมภาษาแอสเซมบลี

นอกเหนือไปจากนี้แล้วควรจะต้องทราบถึงคำสั่งอื่นๆ นอกเหนือไปจากคำสั่งของไมโครคอนโทรลเลอร์เองแล้ว ซึ่งจะต้องพิมพ์แทรกหรือประกอบเข้าไปในโปรแกรมแอสเซมบลี คำสั่งเหล่านี้เรียกว่าคำสั่งเทียม (Pseudo code) หรือคำสั่งสำหรับแอสเซมเบลอร์ (Assembler directive) เพื่อกำหนดให้โปรแกรมเอกสารถือเป็นเอกสารที่สวมนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอสเซมบลีทราบดีกว่าควรจะแปลความหรือดำเนินการอย่างไรบ้าง ซึ่งโดยมากแล้วคำสั่งเหล่านี้มักจะเป็นมาตรฐาน ดังนั้นไม่ว่าโปรแกรมแอสเซมบลีจะเป็นของผู้ผลิตรายใด ก็มักจะใช้คำสั่งที่มีชื่อเหล่านี้เช่นกัน ตัวอย่างคำสั่งเทียมข้างล่างนี้เป็นคำสั่งที่ใช้งานบ่อยในการเขียนโปรแกรม

คำสั่งเทียม	ความหมาย
ORG	กำหนดค่าแอดเดรสให้กับคำสั่งแอสเซมบลี
END	บอกให้ทราบการสิ้นสุดของโปรแกรมแอสเซมบลี
EQU	บอกให้ทราบการใช้ชื่อของกลุ่มตัวอักษร (ลาเบล) แทนการใช้ค่าต่าง ๆ เช่น แอดเดรสหรือตัวเลข โดยตรงภายในโปรแกรม
DB	กำหนดค่าคงที่ขนาดหนึ่งไบต์ให้กับหน่วยความจำ
DW	กำหนดค่าคงที่ขนาดสองไบต์ให้กับหน่วยความจำ
DS	กำหนดหน่วยความจำที่ต้องการจองไว้ล่วงหน้า

การใช้คำสั่งเทียมเหล่านี้เป็นสิ่งที่จำเป็นต้องระบุไว้ภายในโปรแกรมเสมอ เช่น คำสั่ง ORG และ END เป็นต้น มิฉะนั้นโปรแกรมแอสเซมบลีจะไม่สามารถแปลความได้ เนื่องจากบางคำสั่งจะต้องทราบค่าแอดเดรสจึงจะสามารถแปลความได้ เช่น คำสั่งเกี่ยวกับการกระโดด หรือคำสั่งการเรียกโปรแกรมย่อย เป็นต้น ส่วนคำสั่งเทียม DB มักจะนำมาใช้กับการที่ต้องระบุค่าคงที่ลงในบริเวณหน่วยความจำหนึ่ง ๆ เช่น บริเวณหน่วยความจำที่ต้องการให้เก็บค่าของตารางข้อมูล สำหรับโปรแกรมเปิดตาราง (Table Lookup) เป็นต้น โดยไม่สนใจว่าตำแหน่งนี้จะอยู่ที่แอดเดรสใดก็ตาม เพียงแต่เมื่ออ้างอิงก็จะมีค่าคงที่เหล่านั้นจัดเก็บอยู่ภายในเสมอ

2.9.2 การแปลความด้วยโปรแกรมแอสเซมบลี

ภายหลังการเตรียมโปรแกรมแอสเซมบลีต้นฉบับเสร็จสิ้น จะนำมาแปลความด้วยโปรแกรมแอสเซมบลี ซึ่งผลลัพธ์ที่ได้จะได้ไฟล์ที่มีนามสกุลดังนี้

1. ไฟล์ที่มีนามสกุล HEX เป็นไฟล์มาตรฐานรูปแบบหนึ่ง(นิยมใช้แบบ Intel - Hex Format) ซึ่งภายในนอกจากจะมีข้อมูลที่ได้จากการแปลและหมายเลขแอดเดรสแล้ว ยังมีข้อมูลที่ใช้สำหรับการตรวจสอบความถูกต้องของข้อมูลในแต่ละบรรทัดด้วย ดังนั้นจึงนิยมนำไฟล์นี้ไปใช้ในการโอนถ่ายข้อมูลหรือโปรแกรมภายในอีพรอม เนื่องจากจะได้รับความถูกต้องมาก

2. ไฟล์ที่มีนามสกุล LST หรือ PRN เป็นไฟล์ข้อมูลสำหรับผู้ใช้อ้างอิง โดยมีลักษณะเดียวกับโปรแกรมต้นฉบับทุกประการ เพียงแต่เพิ่มการแสดงค่าแอดเดรส และค่าของรหัสคำสั่งในแต่ละบรรทัดเพิ่มขึ้นในส่วนต้นทุกบรรทัด

ตัวอย่างการใช้คำสั่งของโปรแกรมแอสเซมบลีเอสเอ็กซ์เอ 51 (SXA 51)

SXA 51 - L SOURCE.ASM

เป็นการแปลโปรแกรมต้นฉบับชื่อ SOURCE.ASM ซึ่งจะได้ไฟล์นามสกุล HEX ชื่อว่า SOURCE.HEX และกำหนดให้สร้างไฟล์นามสกุล LST ชื่อว่า SOURCE.LST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

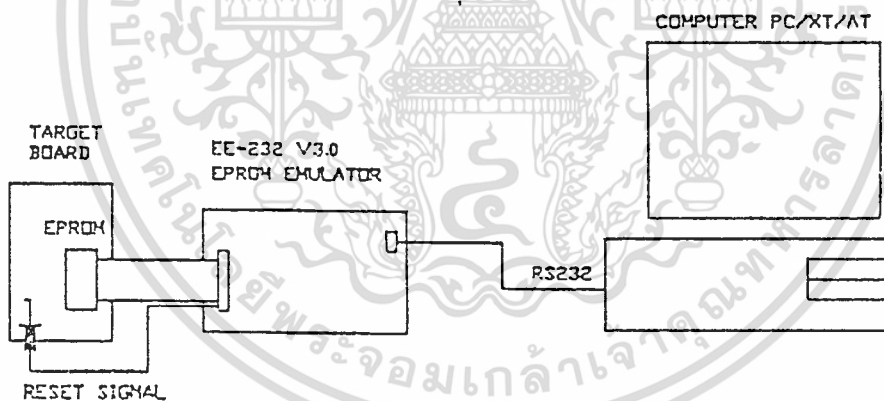
2.9.3 การโอนย้ายโปรแกรมสู่ระบบไมโครคอนโทรลเลอร์

โดยปกติแล้วสำหรับการโอนย้ายข้อมูลสู่ระบบไมโครคอนโทรลเลอร์ สามารถทำได้หลายลักษณะ เช่น

1. บรรจุโปรแกรมลงในไอซีหน่วยความจำอีพรอม (EPROM) โดยใช้เครื่องมืออีพรอมโปรแกรมเมอร์ (EPROM PROGRAMMER) เพื่อนำไอซีนี้ไปเสียบลงในซ็อกเก็ตของบอร์ด กรณีที่ต้องการจะแก้ไขข้อมูลภายในไอซี ก็จะต้องนำไปฉายแสงรังสีอุตราไวโอเลต (UV light) เพื่อล้างข้อมูลก่อน แล้วนำไปบรรจุโปรแกรมใหม่อีกครั้งหนึ่ง

2. กรณีที่มีแผงวงจรจำลองการทำงานของอีพรอม (EPROM EMULATOR) ก็ทำโดยการส่งข้อมูลผ่านช่องสื่อสารแบบอนุกรมหรือแบบขนานของเครื่องไมโครคอมพิวเตอร์ หน่วยทำงานภายในแผงวงจรจำลองการทำงานของอีพรอมจะรับข้อมูลนี้เข้าไปเก็บในไอซีหน่วยความจำแรมบนแผงวงจรเอง ซึ่งหากจะเปรียบเทียบแล้วก็เท่ากับการบรรจุโปรแกรมลงในไอซีอีพรอมนั่นเอง เพียงแต่วิธีการนี้เขียนลงไอซีแรมแทน และสามารถแก้ไขข้อมูลได้โดยการโอนถ่ายข้อมูลชุดใหม่ลงมาแทนที่ข้อมูลเก่าเท่านั้น จึงมีความสะดวกรวดเร็วมากขึ้น

การต่ออีพรอมอิมูเลเตอร์ (EPROM EMULATOR) เข้ากับเครื่องคอมพิวเตอร์พีซีจะใช้พอร์ตอนุกรม (RS 232) ซึ่งสามารถเลือกสายที่มีขั้วแบบ DB25 หรือ DB9 ในด้านที่ต่อเข้ากับเครื่องพีซีได้ตามต้องการ ส่วนทางด้านอีพรอมอิมูเลเตอร์ก็เสียบเข้ากับขั้วแบบ 3 ขาที่อยู่ด้านบนของบอร์ด การต่อใช้งานโดยสรุปจะแสดงได้ดังภาพต่อไปนี้



รูปที่ 2.11 แสดงการต่ออีพรอมอิมูเลเตอร์เข้ากับเครื่องคอมพิวเตอร์พีซีโดยใช้พอร์ตอนุกรม(RS 232)

2.9.4 การใช้งานพื้นฐานของบอร์ดอีพรอมอิมูเลเตอร์ (EE - 232)

เมื่อทำการจ่ายไฟเข้าบอร์ดอีอี - 232 เครื่องจะอยู่ในสภาพที่อิมูเลท (EMULATE) เป็นตัวอีพรอมทันที และจะแสดงที่หลอดแอลอีดีกระพริบเป็นจังหวะช้า ๆ ในขณะที่อีอี - 232 พร้อมทั้งจะใช้งานได้ใน 2 ลักษณะ คือ คอาน์โหลดโหมด (DOWNLOAD MODE) และรีโมทโหมด (REMODE MODE) โดยคอาน์โหลดโหมด คือการโหลดข้อมูลจากเครื่องพีซีได้ทันทีด้วยคำสั่งคอปปี (copy) ในคอส (DOS) และรีโมทโหมดจะเป็นการ

ติดต่อกับบอร์ดอียี - 232 โดยผ่านโปรแกรมสื่อสารเอกซ์ทอล์ค (XTALK) ซึ่งจะทำให้ผู้ใช้สามารถใช้งานคำสั่งต่าง ๆ เกี่ยวกับการอ่านหรือแก้ไขข้อมูลในอียี - 232 ได้

1. คาน์โหลดโหมด คือการนำไฟล์ข้อมูลบนเครื่องพีซีมาโหลดลงอียี - 232 โดยใช้คำสั่งที่อธิบายไว้ในบทนำ แต่อย่างไรก็ตามผู้ใช้จำเป็นต้องกำหนดคุณสมบัติของพอร์ทอนุกรมบนเครื่องพีซีก่อน ด้วยคำสั่งโหมด (เมื่ออยู่บน DOS ทั่วไป) ดังนี้

```
C > MODE COM1 : 96 , N , 8 , 1 , P
```

คอม 1 (COM1) คือพอร์ทอนุกรมหมายเลข 1 ซึ่งบางทีผู้ใช้อาจจะใช้เป็นคอม 2 (COM2) ก็ได้ และตัวเลข 96 คือความเร็วในการสื่อสารที่ 9600 โดยผู้ใช้จะกำหนดเป็นความเร็วอื่น ๆ ได้ตั้งแต่ 2400 ถึง 19200 ในกรณีความเร็วที่ 19200 จะขึ้นอยู่กับเครื่องพีซีแต่ละรุ่นด้วย ซึ่งบางทีตัวฮาร์ดแวร์ของเครื่องพีซีนั้น ๆ อาจจะไม่สามารถใช้ความเร็วที่ 19200 ได้ ส่วนอื่น ๆ ที่เหลือเป็นการกำหนดจำนวนบิตและบิตสตีป (stop) ตามแบบที่อียี - 232 ต้องการ การกำหนดคุณสมบัติด้วยคำสั่งโหมดนี้จะคงอยู่ตลอดไปจนกว่าจะปิดเครื่องพีซี เพราะฉะนั้นให้ทำเพียงครั้งเดียวเมื่อเริ่มต้นเปิดเครื่องก็พอ เมื่อต้องการโหลดข้อมูลลงอียี - 232 ก็ทำได้ด้วยคำสั่งดังนี้

```
C > COPY FILENAME.HEX COM1
```

เมื่อทำคำสั่งนี้แล้ว เครื่องพีซีจะส่ง FILENAME.HEX ออกมาทางพอร์ทอนุกรม และที่อียี - 232 จะรับข้อมูลเข้าบอร์ดทันที โดยสังเกตได้จากหลอดแอลอีดีบนบอร์ดอียี - 232 ซึ่งจะกระพริบตามข้อมูลที่กำลังโหลด ในขณะที่กำลังโหลดนี้ อียี - 232 จะไม่อยู่ในสภาพของกรอิมูเลท และจะส่งสัญญาณรีเซทออกที่ชีพตามแบบที่กำหนดไว้ด้วย จนเมื่อโหลดเสร็จเรียบร้อยแล้ว อียี - 232 จะกลับเข้าสู่สถานะกรอิมูเลทตามเดิม และข้อมูลในตัวอีมูเลทก็จะเปลี่ยนไปตามข้อมูลจาก FILENAME.HEX ที่โหลดเข้ามาใหม่ FILENAME.HEX จะต้องเป็นไปตามมาตรฐานแบบไฟล์อินเทล - เฮกซ์เท่านั้น ในกรณีที่ FILENAME.HEX ไม่ถูกต้องหรือมีความผิดพลาดจากการสื่อสาร อียี - 232 จะแสดงความผิดพลาดให้ทราบด้วยการกระพริบที่แอลอีดีเป็นจังหวะเร็ว ซึ่งผู้ใช้จะต้องกดคีย์รีเซทบนบอร์ดอียี - 232 เพื่อให้เริ่มต้นใช้งานใหม่ได้ ทั้งนี้จะต้องทำการตรวจสอบปัญหาให้ชัดเจน แล้วจึงทำการโหลดข้อมูลอีก อย่างไรก็ตามในบางกรณีการโหลดไฟล์ที่ไม่ถูกต้อง อาจจะไม่มียผลใด ๆ กับอียี - 232 เลยก็ได้ทั้งนี้เนื่องจากอียี - 232 จะไม่สนใจข้อมูลใด ๆ ที่ไม่มีการเริ่มต้นในแบบของไฟล์อินเทล - เฮกซ์

2. รีโมทโหมด จะใช้คำสั่งของโปรแกรมสื่อสารเอกซ์ทอล์ค (XTALK) สำหรับการติดต่อกับ EE - 232 ซึ่งผู้ใช้จะใช้งานในทำนองเดียวกับโปรแกรมดีบั๊ก (DEBUG.COM) บนดอส โดยมีคำสั่งต่าง ๆ ที่สามารถทำการดูและแก้ไขข้อมูลได้อย่างมีประสิทธิภาพ ขั้นตอนการเข้าสู่รีโมทเอดิเตอร์ (Remote Editor) จะทำได้ดังนี้

- ขั้นที่หนึ่ง จัดบอร์ดอียี - 232 ให้อยู่ในสภาพพร้อมเหมือนกับขณะใช้งานในโหมดคาน์โหลด คือต่อสายซีเรียล (serial) เข้ากับเครื่องพีซีและจ่ายไฟเข้าบอร์ดอียี - 232 โดยที่อียี - 232 จะกระพริบแอลอีดีเป็นจังหวะช้า ๆ ทั้งนี้ถ้าไม่เป็นตามนี้ ให้กดรีเซทที่อียี - 232 ก่อน

- ขั้นที่สอง เรียกโปรแกรมเอกซ์ทอล์ค โดยเลือกใช้งานที่พอร์ทคอม 1 หรือคอม 2 ที่เราใช้งาน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A > XTALK

- ขั้นที่สาม โปรแกรมสื่อสารเอกซ์ทอล์คจะเข้าสู่สถานะเตรียมพร้อมสำหรับการติดต่อ ซึ่งจะตั้งความเร็วในการสื่อสารไว้ที่ 9600 โดยจะปรากฏเป็นจอว่าง ๆ และที่บรรทัดสุดท้ายจะมีข้อความดังนี้

^A for Attention, ^F to Switch : Capture Off : Local

- ขั้นที่สี่ ให้กดคีย์สเปซบาร์ (SPACE BAR) ที่เครื่องพีซีหนึ่งครั้ง โดยจะเป็นการเข้าสู่รีโมทดีเตอร์ จะปรากฏข้อความที่จอภาพดังนี้

EE - 232 EPROM EMULATOR V3.0

SILA RESEARCH CO., LTD. BANGKOK, THAILAND

Remote Editor Service... Baud rate : 9600

—

ซึ่งเครื่องหมาย “—” คือพร้อม (prompt) ของรีโมทดีเตอร์ โดยในขณะที่ผู้ใช้จะใช้คำสั่งต่าง ๆ ได้ทันทีที่คีย์คำสั่ง H และกด ENTER ดู (HELP) ซึ่งจะปรากฏข้อความเป็นรายละเอียดของคำสั่งทั้งหมดที่ใช้งานได้

ตัวอย่างคำสั่ง L (DOWN LOAD) มีรูปแบบ L [addr]

—L

Download INTEL - HEX FILE...

Use ^A and SE Command to send file

เครื่องจะแสดงข้อความให้ทราบว่า กำลังรอรับไฟล์จากเครื่องพีซีอยู่ ในระหว่างนี้ผู้ใช้จะคีย์อะไรก็ได้ นอกจากขบวนการส่งไฟล์เท่านั้น เพราะไม่เช่นนั้นอี - 232 จะเข้าใจว่าเป็นความผิดพลาดจากการสื่อสาร และจะแสดงข้อความว่า Download ERROR! ให้ทราบ

ขบวนการส่งไฟล์จากโปรแกรมเอกซ์ทอล์คจะกระทำดังนี้

1. ให้กด ^A เพื่อการเข้าสู่โหมดการใช้คำสั่งของเอกซ์ทอล์ค โดยเครื่องจะแสดงคำว่า Command ? ที่บรรทัดสุดท้ายของจอภาพ และพร้อมที่จะรับคำสั่งต่าง ๆ ของเอกซ์ทอล์ค

2. ใช้คำสั่ง SE และตามด้วยชื่อไฟล์ที่ต้องการจะส่งดังนี้

SE FILENAME.HEX

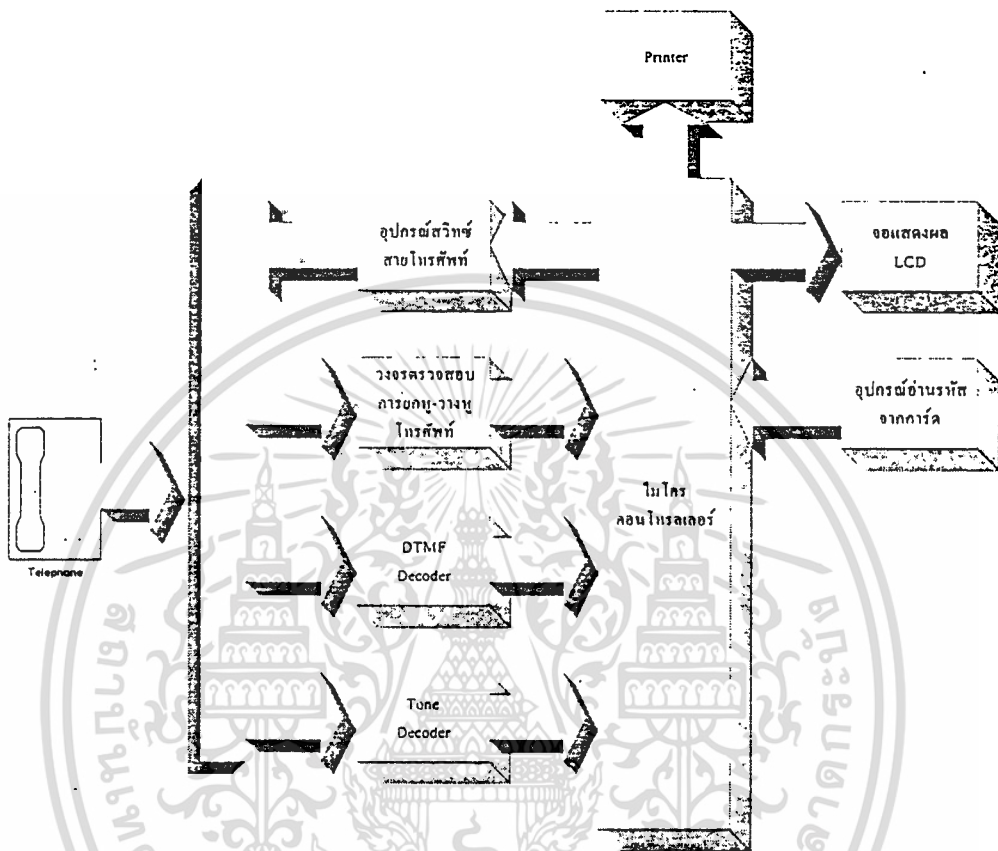
โดยไฟล์ที่จะส่งต้องเป็นไฟล์อินเทล - เฮกซ์ และจะต้องอยู่ในไดเรกทอรีนั้นแต่ถ้าอยู่ที่ไดเรกทอรีอื่น ๆ ผู้ใช้จะสามารถใช้คำสั่ง CD เพื่อการเปลี่ยนไดเรกทอรีได้

3. เครื่องจะแสดงข้อความ Sending file... บนจอภาพและทำการส่งไฟล์ออกไฟทางพอร์ทซีเรียล (Serial Port) เมื่อส่งเรียบร้อยแล้วอีอี - 232 ก็จะกลับมาแสดงเครื่องหมายพร้อม “—” ตามเดิม

บทที่ 3

การออกแบบและการสร้าง

3.1 บล็อกไดอะแกรมของเครื่องควบคุมการใช้โทรศัพท์โดยใช้การ์ดชนิดเจาะรู



รูปที่ 3.1 แสดงบล็อกไดอะแกรมของเครื่องควบคุมการใช้โทรศัพท์โดยใช้การ์ดชนิดเจาะรู

เมื่อผู้ใช้โทรศัพท์ต้องการจะใช้โทรศัพท์เพื่อโทรออกจะต้องทำการสอดการ์ดเจาะรหัสเข้าที่อุปกรณ์อ่านรหัสการ์ด อุปกรณ์อ่านรหัสการ์ดจะส่งข้อมูลที่อ่านได้ทั้ง 16 บิต เข้าไปที่ไมโครคอนโทรลเลอร์เพื่อทำการเปรียบเทียบกับรหัสที่ได้ทำการบันทึกไว้ ถ้าพบว่ารหัสบนการ์ดถูกต้องก็จะทำการสวิตช์สายโทรศัพท์ให้สัญญาณโทรศัพท์เข้าสู่เครื่องโทรศัพท์ สถานะการตรวจสอบรหัสบนการ์ดจะแสดงผลการทำงานบนหน้าจอ LCD เพื่อให้ผู้ใช้ทราบว่าขณะนี้การตรวจสอบรหัสการ์ดถูกต้องหรือผิด สามารถที่จะเข้าใช้โทรศัพท์เพื่อการโทรออกได้หรือไม่

เมื่อผู้ใช้โทรศัพท์กดเลขหมายปลายทางที่ต้องการติดต่อ สัญญาณ DTMF ที่ออกมาจะถูกแปลงให้เป็นรหัสดิจิทัลโดยวงจร DTMF DECODER เข้าสู่ไมโครคอนโทรลเลอร์ เพื่อที่จะให้ไมโครคอนโทรลเลอร์นำข้อมูลนี้ไปทำการบันทึกเลขหมายโทรศัพท์และใช้ในการคำนวณค่าบริการการใช้โทรศัพท์ โดยไมโครคอนโทรลเลอร์จะเริ่มจับเวลาที่เริ่มใช้โทรศัพท์โดยอาศัยการเปลี่ยนแปลงสถานะของเอาท์พุทวงจร TONE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DECODER ที่เปลี่ยนแปลงลอจิกตามสัญญาณโทรศัพท์ และการเลิกจับเวลาการใช้งานโทรศัพท์จะถูกยกเลิกเมื่อผู้ใช้โทรศัพท์ได้ทำการวางหูโทรศัพท์ลง โดยมีวงจรตรวจสอบการวางหูโทรศัพท์และส่งข้อมูลที่ตรวจสอบได้เข้าสู่ไมโครคอนโทรลเลอร์ ข้อมูลบันทึกการใช้งานโทรศัพท์จะทำการบันทึกเก็บไว้ในส่วนที่เป็นหน่วยความจำของไมโครคอนโทรลเลอร์ และสามารถที่จะทำการเรียกดูข้อมูลที่เก็บไว้ได้โดยใช้คำสั่งควบคุมไมโครคอนโทรลเลอร์ให้มีการพิมพ์ผลออกแสดงทางเครื่องพิมพ์

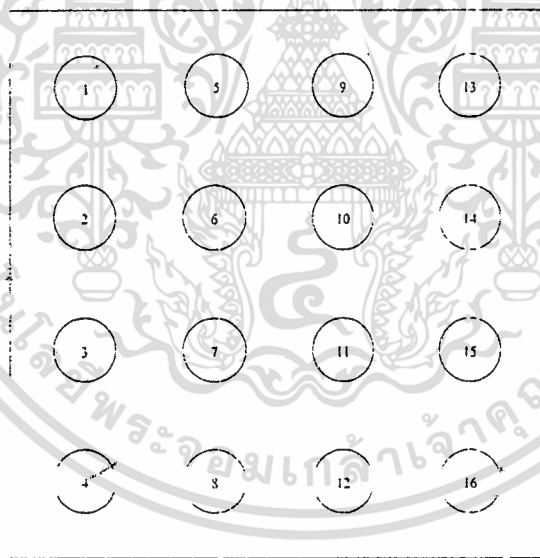
3.2 การออกแบบการ์ดเจาะรหัส

ลักษณะของบัตรจะเป็นสี่เหลี่ยมผืนผ้าพลาสติกแข็งขนาดเท่ากับบัตรโทรศัพท์ขององค์การโทรศัพท์แห่งประเทศไทยเพื่อที่จะได้สามารถพกพาได้ง่าย การเก็บข้อมูลบนการ์ดทำได้โดยการเจาะรูที่บัตร เราสามารถที่จะเจาะรูที่บัตรได้ทั้งหมด 16 รู เพราะว่าบนการ์ดเราออกแบบให้มีการแบ่งเป็นแถวตามแนวนอน 4 รู และตามแนวตั้ง 4 รู ดังนั้นเราจะสามารถที่จะเก็บรหัสได้ 16 บิต โดยแต่ละตำแหน่งจะแทนข้อมูล 1 บิต ทำให้เราสามารถที่จะเก็บข้อมูลได้ถึง 65,536 รหัส

ตำแหน่งที่เราทำการเจาะรูการ์ด ข้อมูลจะเป็นลอจิก "0"

ตำแหน่งที่เราไม่ทำการเจาะรูการ์ด ข้อมูลจะเป็นลอจิก "1"

โดยเราจะให้ตำแหน่งรูที่ 8 เป็นตัวตรวจสอบสถานะว่ามีการเสียบการ์ดหรือไม่ ดังนั้นจึงออกแบบให้ตำแหน่งนี้มีค่าลอจิก "1" เสมอ

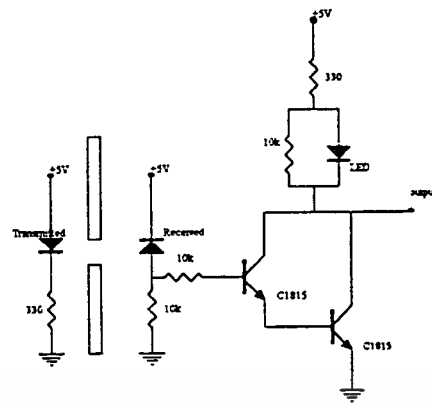


รูปที่ 3.2 แสดงลักษณะของการ์ดเจาะรหัสที่ออกแบบ

3.3 การออกแบบส่วนอุปกรณ์อ่านรหัสการ์ด

การอ่านข้อมูลบนการ์ดเจาะรหัสเราจะต้องสอดการ์ดผ่านอุปกรณ์อ่านรหัสซึ่งประกอบด้วยส่วนที่เป็น INFRARED LED (ตัวส่ง) และ PHOTO DIODE DETECTOR (ตัวรับ) จำนวน 16 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

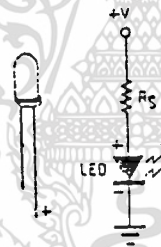


รูปที่ 3.3 แสดงวงจรส่วนอุปกรณ์อ่านรหัสคาร์ด 1 ตำแหน่ง

เราจะต้องมีการต่อตัวต้านทานอนุกรมที่เหมาะสมเข้ากับ LED อินฟราเรด เพื่อจำกัดกระแสที่ไหลผ่าน โดยมีสูตรคำนวณค่าตัวต้านทานดังนี้

$$R_S = (V_{IN} - V_{LED}) / I_{LED}$$

เครื่องหมาย + ที่หลอด LED แสดงถึง แอนโอด และที่เครื่องหมาย - ที่หลอด LED แสดงถึง แคโทด



รูปที่ 3.4 แสดงวงจร LED ที่ต่อกับตัวต้านทานเพื่อการจำกัดกระแส

ส่วนที่ด้านรับมีการออกแบบให้มี LED ไฟสีแดงเป็นตัวแสดงผลค่าลอจิกเอาท์พุทที่ได้ โดย LED จะไฟติดเมื่อตำแหน่งที่อ่านได้จากการ์ดเป็นตำแหน่งที่มีการเจาะรู และในทางกลับกัน LED จะดับเมื่อตำแหน่งที่อ่านได้จากการ์ดเป็นตำแหน่งที่ไม่ได้ทำการเจาะรู ดังนั้นเราจึงสรุปความสัมพันธ์ได้ว่า

เอาท์พุทที่เข้าสู่ไมโครคอนโทรลเลอร์จะเป็นลอจิก "0" LED จะแสดงไฟติด

เอาท์พุทที่เข้าสู่ไมโครคอนโทรลเลอร์จะเป็นลอจิก "1" LED จะดับ

3.4 การออกแบบส่วนที่เป็น TONE DECODER

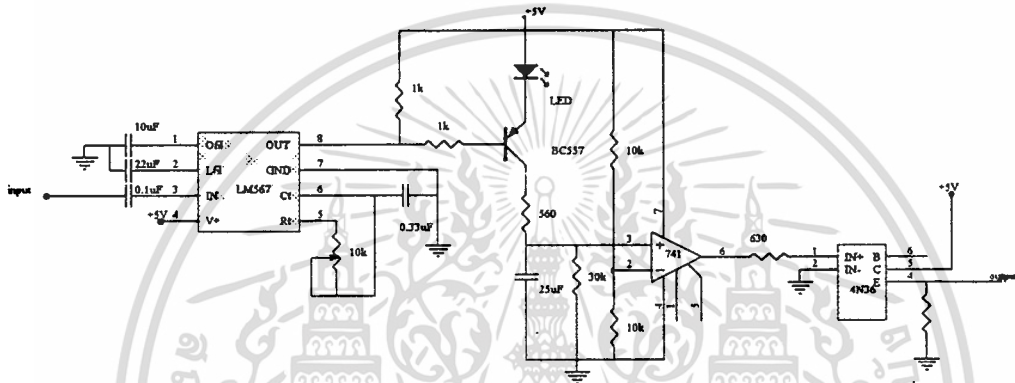
วงจร TONE DECODER จะใช้ ไอซีเบอร์ LM 567 ทำหน้าที่ตรวจจับสัญญาณเฉพาะความถี่ประมาณ 400 เฮิร์ต โดยการกำหนดค่าของตัวต้านทาน R และตัวเก็บประจุ C จากสมการ ความสัมพันธ์ดังนี้

$$f = 1 / 1.1RC$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการขุดโทรศัพท์จะได้ยินสัญญาณ dial tone เอาท์พุทของ LM 567 ที่ขา 8 จะได้ลอจิก “0”

เมื่อมีการกดหมายเลขจะทำให้สัญญาณ dial tone เียบหายไปซึ่งจะทำให้เอาท์พุทของ LM 567 นี้จะเปลี่ยนมาเป็นลอจิก “1” สัญญาณที่จะทำให้อาท์พุทของไอซี LM 567 เปลี่ยนแปลงจากลอจิก “1” เป็น “0” นั้นได้แก่สัญญาณเรียกกลับ (ringback tone) ซึ่งสัญญาณนี้จะมีควมถี่ 440 เฮิรต และ 480 เฮิรต ดัด 1 วินาที ดับ 4 วินาทีสลับกันซึ่งจะทำให้อาท์พุทที่ได้มีค่า “0” และ “1” สลับกัน และอีกสัญญาณหนึ่งคือสัญญาณสายไม่ว่าง (busy tone) ซึ่งสัญญาณนี้จะมีควมถี่ 480 เฮิรต กับ 620 เฮิรต ซึ่งจะดัด 0.5 วินาที ดับ 0.5 วินาที ซึ่งก็จะทำให้ได้เอาท์พุทเป็น “0” และ “1” สลับกัน ซึ่งสัญญาณเอาท์พุทเหล่านี้จะถูกส่งเข้าไมโครคอนโทรลเลอร์เพื่อทำการเปรียบเทียบระดับสัญญาณ โดยจะเลือกเอาเฉพาะสัญญาณที่เป็น “1” นานกว่า 4 วินาทีมาใช้ในการเริ่มต้นการจับเวลาการใช้โทรศัพท์ และจะสิ้นสุดการจับเวลาเมื่อเครื่องโทรศัพท์ถูกวางหูลง

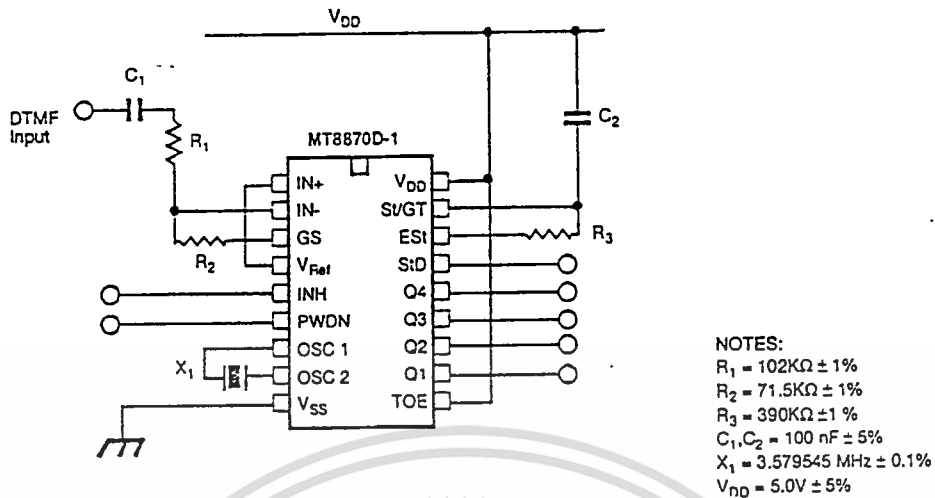


รูปที่ 3.5 แสดงวงจร TONE DECODER

เอาท์พุทที่ได้จากไอซี เบอร์ LM 567 จะถูกส่งเข้าสู่วงจรเปรียบเทียบระดับสัญญาณที่มีระดับสัญญาณแรงดันอ้างอิงที่ขา 2 มีค่า 2.5 โวลต์ เอาท์พุทที่ออกจากวงจรเปรียบเทียบจะถูกต่อไปยังส่วน ที่เป็น opto coupler เบอร์ 4N36 ในช่วงที่มีกระแสไหลผ่าน opto coupler จะทำให้ขาเอาท์พุทของ opto coupler เป็นลอจิก “1” และในทำนองกลับกันในช่วงที่ไม่มีกระแสไหลผ่าน opto coupler จะทำให้ขาเอาท์พุทของ opto coupler เป็นลอจิก “0”

3.5 การออกแบบส่วนที่เป็นวงจร DTMF DECODER

การทำงานของวงจรนี้ จะใช้ IC เบอร์ MT8870 เป็นตัวถอดรหัส DTMF โดยทำงานร่วมกับคริสตัลออสซิลเลเตอร์ความถี่ 3.579 เมกะเฮิรต เมื่อมีการกดเลขหมายป้อนเข้าทางอินพุท ทางเอาท์พุทของวงจรจะถอดรหัสออกมาเป็นเลขไบนารี 4 บิต ที่ขา Q1-Q4 และที่ขา STD จะเป็นตัวบอกให้ซีพียูรับรู้ว่าจะมีการส่งหมายเลขมาแล้ว



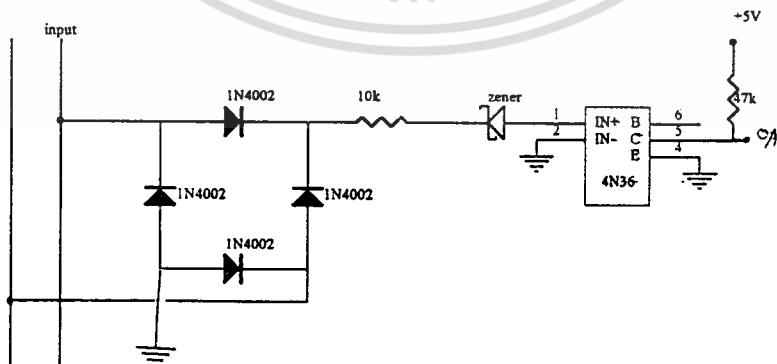
รูปที่ 3.6 แสดงวงจร DTMF DECODER

3.6 การออกแบบส่วนที่เป็นวงจรตรวจสอบการวางหูและยกหูโทรศัพท์

การทำงานของวงจร ในสภาวะปกติสายโทรศัพท์จะมีแรงดันประมาณ 48 โวลต์ เมื่อยกหูจะมีแรงดันลดลงเหลือประมาณ 6-8 โวลต์ เมื่อสัญญาณโทรศัพท์ผ่านวงจรบริดจ์เรกติไฟเออร์ จะได้สัญญาณที่เป็นซีกบวกออกมา ผ่านตัวความต้านทานที่ต่ออนุกรมกับ ไดโอดจะช่วยลิมิตกระแสที่ไหลผ่าน

เมื่อสัญญาณแรงดันขณะยกหูโทรศัพท์ที่มีค่าประมาณ 6 โวลต์ ไดโอดจะไม่เกิดการนำกระแส ทำให้ไม่มีกระแสไปไบอัส ออปโต คัปเปอ์ เอ้าท์พุทได้ที่ขา 5 ของ ไอซี 4N36 จะมีค่าโวลต์เตจออกมา มีสถานะสูงกว่ากรณีวางหูโทรศัพท์

ขณะวางหูโทรศัพท์ สัญญาณจะมีค่าแรงดันประมาณ 48 โวลต์ ไดโอดจะเกิดการนำกระแสทำให้มีกระแสไปไบอัสออปโตคัปเปอ์ เอ้าท์พุทที่ออกมาที่ขา 5 ของ ไอซี 4N36 จะมีค่าโวลต์เตจออกมา มีสถานะต่ำกว่ากรณีแรก



รูปที่ 3.7 แสดงวงจรตรวจสอบการวางหูและยกหูโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 บอร์ดไมโครคอนโทรลเลอร์แอนท์ - 32 (ANT - 32)

แอนท์ - 32 เป็นบอร์ดไมโครคอนโทรลเลอร์นำไปใช้งานในลักษณะเอ็มเบดคอนโทรลเลอร์ (EMBEDDED CONTROLLER) กล่าวคือ เป็นบอร์ดที่ถูกออกแบบมาเพื่องานควบคุมโดยเฉพาะ โดยถูกติดตั้งอยู่ในเครื่องมือ เครื่องจักรกล เครื่องใช้ไฟฟ้า รวมทั้งระบบอัตโนมัติต่าง ๆ บอร์ดนี้สามารถใช้กับซีพียูตระกูลเอ็มซีเอส - 51 (MCS - 51) ของอินเทล ซึ่งประกอบไปด้วยซีพียูเบอร์ต่าง ๆ ได้แก่ 8031 , 8051 , 8032 , 8052 , 8751 , 8752 และ 8052 AHBASIC

3.7.1 คุณสมบัติของบอร์ดแอนท์ - 32 (ANT - 32)

- เป็นบอร์ดคอนโทรลเลอร์ใช้กับไมโครคอนโทรลเลอร์ในตระกูล เอ็มซีเอส - 51 (MCS - 51) (8031/8032) ใช้ ซีพียู เบอร์ 80C32 ทำงานที่ความถี่สัญญาณนาฬิกา 11.0592 เมกกะเฮิร์ต

- ใช้งานหน่วยความจำบนบอร์ดได้ 3 ตำแหน่งด้วยกัน คือ

U2 เป็นหน่วยความจำโปรแกรม (PROGRAM MEMORY) ใช้กับอีพรอมขนาด 8 - 32 กิโลไบต์ เบอร์ 2764 , 27128 หรือ 27256

U3 เป็นหน่วยความจำข้อมูล (DATA MEMORY) ใช้กับแรมขนาด 8 กิโลไบต์ เบอร์ 6264 หรือ 32 กิโลไบต์ เบอร์ 62256 สามารถแบคอัพข้อมูลได้โดยใช้แบตเตอรี่ลิเทียม

U4 เป็นหน่วยความจำโปรแกรมและข้อมูล (PROGRAM AND DATA MEMORY) ใช้กับอีพรอม , แรมหรืออีอีพรอมขนาด 8 - 32 กิโลไบต์ โดยใช้อีพรอมเบอร์ 2764 , 27256 ใช้แรมเบอร์ 6264 , 62256 หรืออีอีพรอม เบอร์ 2864 (A) , 28256 (A)

- มีพอร์ตไอโอ (I/O) เบอร์ 8255 จำนวน 2 ตัว (48 บิต) สำหรับต่อไปใช้งานภายนอก

- มีพอร์ตแอลซีดี (LCD) สำหรับการต่อใช้งานกับแอลซีดีแบบดอทเมทริกซ์ (DOT MATRIX)

- มีวงจร SERIAL INTERFACE DRIVER RS232 ด้วยชิพเบอร์ MAX232 สำหรับการต่อเข้ากับเครื่องไมโครคอมพิวเตอร์

- มีวงจรวอตดอกทาม์เมอร์ (Watchdog Time) , พาวเวอร์อัพ / ดาวน์รีเซท (Powerup / down Reset) ด้วยชิพเบอร์ MAX691

- มีวงจรอาร์ทีซี (Real Time Clock) ใช้ชิพเบอร์ DS1202

- มีคอนเน็คเตอร์สำหรับพอร์ต 1 ของไมโครคอนโทรลเลอร์โดยเฉพาะ

- มีคอนเน็คเตอร์สำหรับซิสเทมบัส (SYSTEM BUS) ทำให้ขยายระบบได้ง่าย และสามารถใช้กับบอร์ดขยายต่าง ๆ ที่มีขึ้นในอนาคต

- สามารถเลือกเบอร์และชนิดหน่วยความจำ หรือกำหนดคุณสมบัติต่าง ๆ ของบอร์ดได้ด้วยจัมพ์เปอร์

- สามารถพัฒนาโปรแกรมได้ทั้งภาษาเบสิกและแอสเซมบลี โดยใช้ซอฟต์แวร์พรอม BASIC32 และ REM31

3.7.2 แนวทางในการพัฒนาโปรแกรม

การใช้งานบอร์ดแอนท์ - 32 ผู้ใช้จำเป็นต้องเขียนโปรแกรมควบคุมที่เรียกกันว่า มอนิเตอร์โปรแกรม ขึ้นมาโดยเฉพาะเพื่อทำให้งานที่ต้องการพัฒนาสำเร็จได้ ในขั้นตอนการพัฒนานี้เองที่เป็นจุดเด่นของแอนท์ - 32 โดยมีโปรแกรมให้เลือก 2 ลักษณะด้วยกัน คือ REM31 และ BASIC32 หลักการของทั้งสองโปรแกรมก็คือเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตสำหรับผู้ใช้ที่ 1 (USER PORT1)

พอร์ต A	ตำแหน่งแอดเดรส	$F800H + 00H = F800H$
พอร์ต B	ตำแหน่งแอดเดรส	$F800H + 01H = F801H$
พอร์ต C	ตำแหน่งแอดเดรส	$F800H + 02H = F802H$
โหมดพอร์ต	ตำแหน่งแอดเดรส	$F800H + 03H = F803H$

พอร์ตสำหรับผู้ใช้ที่ 2 (USER PORT2)

พอร์ต A	ตำแหน่งแอดเดรส	$FC00H + 00H = FC00H$
พอร์ต B	ตำแหน่งแอดเดรส	$FC00H + 01H = FC01H$
พอร์ต C	ตำแหน่งแอดเดรส	$FC00H + 02H = FC02H$
โหมดพอร์ต	ตำแหน่งแอดเดรส	$FC00H + 03H = FC03H$

ก่อนที่จะใช้งานพอร์ต 8255 ผู้ใช้ต้องทำการกำหนดโหมดการทำงาน (configuration) ของพอร์ต A,B และ C ให้เป็นพอร์ตอินพุตหรือเอาต์พุต โดยทำการเขียนค่าคอนโทรลโค้ด (control code) ไปที่ (Mode Port) ซึ่งโหมดพอร์ตนี้สามารถเขียนได้เท่านั้นไม่สามารถอ่านได้ ในที่นี้จะกล่าวเฉพาะการทำงานในโหมด 0 ซึ่งเป็นโหมดที่ใช้งานได้สะดวกและง่ายต่อการทำความเข้าใจ ซึ่งสามารถแสดงค่าคอนโทรลโค้ดดังตารางข้างล่าง

Port A (PA0 - PA7)	Port C บน (PC4 - PC7)	Port B (PB0 - PB7)	Port C ล่าง (PC0 - PC3)	Control code (hex)
output	output	output	output	80H
output	output	output	input	81H
output	output	input	output	82H
output	output	input	input	83H
output	input	output	output	88H
output	input	output	input	89H
output	input	input	output	8AH
output	input	input	input	8BH
input	output	output	output	90H
input	output	output	input	91H
input	output	input	output	92H
input	output	input	input	93H
input	input	output	output	98H
input	input	output	input	99H
input	input	input	output	9AH
input	input	input	input	9BH

ตารางที่ 3.1 คอนโทรลโค้ดที่ใช้ในการกำหนดการทำงานของ 8255 โหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญูญาติให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.4 พอร์ทแอลซีดี (LCD PORT)

บอร์ดแอนด์ - 32 จะมีพอร์ทแอลซีดีให้พร้อมสำหรับการต่อใช้งานโดยสามารถต่อเข้ากับแอลซีดีโมดูล (LCD MODULE) แบบค็อทเมทริกซ์ (DOT MATRIX) ได้ทันที ซึ่งจะใช้ขาสัญญาณทั้งหมด 14 ขา และสำหรับการใช้งานพอร์ทแอลซีดีนั้น จะมีการจัดวงจรในแบบเมโมรี่แมพ (MEMORY MAP) ซึ่งจะช่วยให้การเขียนโปรแกรมทำได้ง่าย โดยจะมองเห็นตำแหน่งต่าง ๆ ที่สรุปได้ดังนี้

แอดเดรส	ลักษณะของพอร์ทที่ติดต่อ
FA00H	สำหรับเขียนคำสั่ง (RS = 0 R / W = 0)
FA01H	สำหรับอ่านค่าไม่ว่าง (BUSY) (RS = 0 R / W = 1)
FA02H	สำหรับเขียนข้อมูล (RS = 1 R / W = 0)
FA03H	สำหรับอ่านข้อมูล (RS = 1 R / W = 1)

การอ่านค่าแอลซีดีแบบค็อทเมทริกซ์นี้ จะสามารถเลือกรุ่นใด ๆ ก็ได้ โดยมีจำนวนตัวอักษรต่อบรรทัดและจำนวนบรรทัดตามที่ต้องการ เพราะสายสัญญาณที่ใช้จะใช้แบบเดียวกันหมดจะแตกต่างกันก็ทีโปรแกรมเท่านั้น การนับหมายเลขขั้วต่อของพอร์ทแอลซีดีจะไม่เหมือนการนับทั่ว ๆ ไป จึงควรดูให้แน่ใจก่อนการต่อใช้งานอีกประการหนึ่ง หมายเลขขั้วต่อที่ด้านแอลซีดีก็มักจะมีหลายแบบ คือ อาจจะเป็นแถวคู่หรือแถวเดี่ยวก็ได้ แต่ทั้งนี้หมายเลข 1 - 14 ของขาสัญญาณก็จะตรงกันหมด กล่าวคือ ต่อหมายเลขให้ตรงเป็นใช้ได้ รายละเอียดของการใช้งานตัวแอลซีดีจะมีรายละเอียดดังนี้

ปัจจุบันแอลซีดีเป็นที่นิยมกันเป็นอย่างมาก สำหรับการแสดงผลในเครื่องมือเครื่องใช้ต่าง ๆ ทั้งนี้เนื่องจากมีความเหมาะสมหลาย ๆ ด้าน เช่นการใช้กระแสดำ สามารถแสดงผลเป็นตัวอักษรและตัวเลข หรือแสดงเป็นกราฟฟิก (เฉพาะรุ่น) จะมีปัญหาที่คือในด้านวงจร ซึ่งมีระบบการทำงานที่ซับซ้อนและหาอุปกรณ์ได้ค่อนข้างยาก แต่ขณะนี้ผู้ผลิตแอลซีดีจะทำรุ่นที่เป็นแอลซีดีจะผลิตเป็นโมดูล (module) ออกมาคือเป็นโมดูลที่มีตัวแอลซีดีและวงจรควบคุมมาให้พร้อม (เรียกว่า LCM) ซึ่งทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโคร ๔ ได้ง่ายและสะดวกสำหรับการเขียนโปรแกรม รวมทั้งมีจำหน่ายกันอย่างกว้างขวางและราคาที่เหมาะสม ทำให้ผู้ใช้ทางด้านไมโคร ๔ หันมาใช้แผงแสดงผลด้วยแอลซีดีโมดูล (LCD module) กันมากขึ้น

แอลซีดีโมดูลมีอยู่หลายรุ่น และคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลักคือ แบบค็อทเมทริกซ์ และ กราฟฟิก โดยแบบค็อทเมทริกซ์จะแสดงผลเป็นตัวอักษรขนาด 5 x 8 และมีจำนวนอักษรและบรรทัดแตกต่างกันไปในแต่ละรุ่น ส่วนแบบกราฟฟิกจะสามารถแสดงผลในแบบบิตแมพ (Bit map) คือจะสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ แนวทางในการใช้งานของทั้ง 2 แบบ จะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบค็อทเมทริกซ์มากกว่าเนื่องจากราคาถูกกว่า และเพียงพอต่องานส่วนใหญ่ คุณสมบัติของค็อทเมทริกซ์แอลซีดีโมดูลสามารถสรุปเป็นข้อ ๆ ดังนี้

1. มีให้เลือกหลายรุ่นตามความต้องการใช้งาน โดยมีจำนวนตัวอักษร และบรรทัดแตกต่างกันไป
 2. ตัวอักษรแสดงด้วยค็อทเมทริกซ์ขนาด 5 x 8
 3. สามารถต่อเข้ากับระบบไมโคร ๔ ได้ 2 ลักษณะคือ แบบเมโมรี่แมพ (Memory map) และแบบผ่าน
 4. พอร์ท 8255 ซึ่งจะใช้ขาสัญญาณทั้งหมด 14 ขา (ขั้วต่อ 16 PIN)
 5. การใช้งานง่ายและสะดวก ระบบไมโคร ๔ เพียงแค่ส่งข้อมูลให้กับแอลซีดีโมดูลเท่านั้น ข้อความก็จะปรากฏบนจอเป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
- ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎบนแผงแสดง และจะค้างไว้ตลอดทำให้ไม่ต้องเสียเวลาของระบบ

6. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น CLEAR , DISPLAY , HOME , CURSOR , ON OFF CURSOR , BLANK CHARACTER และอื่น ๆ อีก

7. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบได้อีก 8 ตัว

8. กินกระแสต่ำและมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์ เท่านั้น

3.7.5 ขาสัญญาณของ LCD MODULE

PIN	SYMBOL	LEVEL	FUNCTION
1.	Vss	-	0 V GND
2.	Vcc	-	+5 V Power Supply
3.	Vee	-	+V For Liquid Crystal Drive
4.	RS	H / L	Register Select H : Data Input L : Instruction Input
5.	R / W	H / L	H : Data Read L : Data Write
6.	E	H	Enable Signal (L -> H)
7.	DB 0	H / L	Data Bus Bit 0
8.	DB 1	H / L	Data Bus Bit 1
9.	DB 2	H / L	Data Bus Bit 2
10.	DB 3	H / L	Data Bus Bit 3
11.	DB 4	H / L	Data Bus Bit 4
12.	DB 5	H / L	Data Bus Bit 5
13.	DB 6	H / L	Data Bus Bit 6
14.	DB 7	H / L	Data Bus Bit 7

ตารางที่ 3.2 แสดงขาสัญญาณของ LCD MODULE

3.8 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์

แอลซีดีโมดูลจะต่อเข้ากับระบบไมโคร ๑ ได้ 2 ลักษณะดังได้กล่าวไปแล้ว ซึ่งทั้งสองแบบนี้จะมีข้อดีและข้อเสียแตกต่างกันไป โดยแต่ละแบบจะมีหลักการดังนี้

3.8.1 การต่อแบบเมโมรี่แมพ (Memory map)

1. สามารถต่อเข้ากับชิพเบอร์ทั่ว ๆ ไปได้ เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโคร ๑ มองเห็นแอลซีดีโมดูลในลักษณะของเมโมรี่ได้ทันที

2. ผู้ใช้สามารถเขียนและอ่านข้อมูลจากแอลซีดีโมดูลได้ ทำให้มองเสมือนว่าเป็นเมโมรี่บัฟเฟอร์ (Memory buffer) ไปในตัว

3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบแฟล็ก (Flag) ความพร้อมในขณะที่แอลซีดีโมดูลกำลังทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ใช้ได้กับบอร์ดที่มีบัสแอลซีดี (LCD bus) มาให้พร้อมเท่านั้น
5. ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่ง และต้องมีการดีโค้ด (Decode) ละเอียดพอสมควร
6. การจัดหาสัญญาณจะต้องเป็นไปตามแบบของชิพ (Chip)

3.8.2 การต่อแบบพอร์ทไอ / โอ (I / O port)

1. สามารถต่อเข้ากับพอร์ทไอ / โอ ใด ๆ ก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมา ให้ตรงกับข้อกำหนดของแอลซีดีโมดูล
2. ผู้ใช้จะเขียนข้อมูลให้แอลซีดีโมดูลได้อย่างเดียว ซึ่งผู้ใช้ควรจะทำหนดเมโมรี่ (memory) ส่วนหนึ่งให้เสมือนบัฟเฟอร์ (Buffer) ให้กับแอลซีดีโมดูล
3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบเองเพื่อรอให้แอลซีดีโมดูลกระทำขบวนการต่าง ๆ
4. ใช้ได้กับบอร์ดทั่ว ๆ ไปที่มีพอร์ท
5. ไม่เปลืองส่วนของเมโมรี่ในการใช้งาน
6. การจัดหาสัญญาณกระทำได้อย่างอิสระ

3.8.3 ชุดคำสั่งควบคุมและการแสดงข้อความ

ขาสัญญาณ VEE มีไว้สำหรับกำหนดความเข้มของตัวอักษร โดยถ้าต่อกับ GND จะมีความเข้มสูงสุด แต่ถ้าต่อกับ VCC จะมีความเข้มต่ำสุด ปกติแอลซีดีรุ่นธรรมดาจะต่อกับ GND ไว้เลยก็ได้ ไม่ต้องใส่ VR ให้สิ้นเปลือง แต่ถ้าเป็นรุ่น STN (มุมมองกว้าง) ให้ใช้ R 2K ต่อลง GND อีกที เพื่อให้ความเข้มมีความเหมาะสม การเขียนหรืออ่านข้อมูลกับแอลซีดีโมดูลก็คือการกำหนดคุณสมบัติต่างๆ ในการใช้งานของแอลซีดีตามชุดคำสั่งควบคุมและรวมไปถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดตามตารางต่อไปนี้

INSTRUCTION	RS	R/W	DATA BIT								EXE. TIME	
			7	6	5	4	3	2	1	0		
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B		40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	*	*		40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*		40
SET CGRAM ADD.	0	0	0	1	CGRAM ADDRESS							40
SET DDRAM ADD.	0	0	1	DDRAM ADDRESS							40	
BUSY, ADD. READ	0	1	BF	ADDRESS							0	
CGRAM, DDRAM WR	1	0	WRITE DATA							40		
CGRAM, DDRAM RD	1	1	READ DATA							40		

ตารางที่ 3.3 แสดงรายละเอียดของชุดคำสั่งควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 ความเข้าใจพื้นฐาน

1. การเขียนข้อมูลให้กับแอลซีดีโมดูลจะแบ่งเป็น 2 ลักษณะคือคำสั่ง (Instruction) และข้อมูล (Data) โดยจะกำหนดด้วยขาสัญญาณ RS คือถ้า $RS = 0$ จะหมายถึงส่งสัญญาณควบคุม (Instruction) หรืออ่านค่าแฟล็กสภาพการทำงานของแอลซีดีโมดูลและถ้า $RS = 1$ จะหมายถึงการเขียนหรืออ่านข้อมูลกับแอลซีดีโมดูล

2. หลักการในการเขียนข้อมูลให้แอลซีดีโมดูลนี้ คือเมื่อมีการเขียนข้อมูลไปแล้ว ตัวแอลซีดีโมดูลจะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง (ตามค่า execute time ในตาราง) ซึ่งระบบไมโคร ๗ สามารถตรวจสอบได้จาก BF (Busy Flag) และถ้าเรียบร้อยแล้ว จึงจะสามารถเขียนข้อมูลอันต่อไปได้ในกรณีที่การต่อวงจรเป็นแบบพอร์ทไอ / โอ คือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโคร ๗ ก็จะต้องใช้วิธีการหน่วงเวลาแทน

3. การเขียนข้อมูลให้กับแอลซีดีโมดูลนี้สามารถทำได้ทั้งแบบ 8 บิต และ 4 บิต โดยกรณี 4 บิตจะใช้สายสัญญาณข้อมูลเพียง 4 เส้น คือ DB4 - DB7 (ใช้สำหรับระบบไมโคร ๗ แบบ 4 บิต หรือเพื่อการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 บิต เพียงแต่ให้เขียน 2 ครั้ง คือ DB4 - DB7 ก่อนแล้วตามด้วย DB0 - DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในคำสั่งกำหนดหน้าที่ (Function set) ด้วย

4. DDRAM (Display Data Ram) คือหน่วยความจำภายในตัวแอลซีดีโมดูลที่เป็นบัพเฟอร์ของข้อมูล โดยถ้าเขียนรหัสแอสกี (ASCII) ใด ๆ ลงไปในหน่วยความจำนี้ ก็จะปรากฏเป็นตัวอักษรที่แสดงบนทันที

5. CGRAM (Character Generator Ram) คือหน่วยความจำภายในตัวแอลซีดีโมดูล สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง (8 ตัว) โดยจะอ้างแอดเดรสได้ทั้งหมด 64 ไบต์ คือ 8 ตัวอักษรคูณกับ 8 แถว

3.9.1 รายละเอียดของแต่ละคำสั่ง

1. CLEAR DISPLAY

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

สำหรับการ Clear Display โดยจะทำการเขียนตัวอักษร Space ลงไปใน DDRAM ทั้งหมด และทำการกำหนดค่าแอดเดรส (DDRAM Address) ให้เป็น 0 พร้อมทั้งเคอร์เซอร์ (Cursor) จะกลับไปตำแหน่งซ้ายบนสุดของจอภาพ

2. CURSOR AT HOME

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

สำหรับกำหนดค่าแอดเดรสของ DDRAM ให้เป็น 0 พร้อมทั้งเคอร์เซอร์จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ENTRY MODE SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

I/D = 0 กำหนดทิศทางของเคอร์เซอร์และ DDRAM ให้เป็นแบบลด (Decrement)

I/D = 1 กำหนดทิศทางของเคอร์เซอร์และ DDRAM ให้เป็นแบบเพิ่ม (Increment)

S = 0 เมื่อเขียนข้อมูลแล้วตัวเคอร์เซอร์จะถูกเลื่อนไปตามทิศทางของค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้วตัวเคอร์เซอร์จะอยู่กับที่ และตัวอักษรจะถูกดันไปตามทิศทางของค่า I/D

การกำหนด I/D และ S นี้ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้วจะต้องไม่ใช่

คำสั่ง Clear display อีก

4. DISPLAY ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

D = 0 กำหนดให้ Off display

D = 1 กำหนดให้ On display

C = 0 กำหนดให้ Off cursor

C = 1 กำหนดให้ On cursor โดยเคอร์เซอร์จะเป็นเส้นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่งเคอร์เซอร์

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่งเคอร์เซอร์ (กระพริบเป็นรูปสี่เหลี่ยม)

5. DISPLAY SHIFT

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

S/C = 0 กำหนดให้เลื่อนเคอร์เซอร์ตามทิศทาง R/L ไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความบนแผงแสดงตามทิศทาง R/L ไป 1 คอลัมน์ (เลื่อนทุกบรรทัด)

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย

R/L = 1 กำหนดให้มีทิศทางไปทางขวา

6. FUNCTION SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้การติดต่อกับแอสซีติโมดูลเป็นแบบ 4 บิต

DL = 1 กำหนดให้การติดต่อกับแอสซีติโมดูลเป็นแบบ 8 บิต จะสังเกตว่า การกำหนดค่า D/L นี้สามารถกระทำได้ที่ DB4 - DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 บิต ตั้งแต่ครั้งแรกหลังจากจ่ายไฟเลี้ยงก็จะทำให้แอสซีติโมดูล มีการรับข้อมูลแบบ 4 บิตทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 คิวตี้ (Duty) และ 1/11 คิวตี้

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 คิวตี้

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5*7 คีอท

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5*10 คีอท

(กรณีที่แอสซีติโมดูลเป็นแบบ 5*7 อยู่แล้ว ก็จะไม่มีผลอะไร)

7. SET CGRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

สำหรับการกำหนดแอดเดรสของ CGRAM เมื่อได้ทำการกำหนดไว้แล้วการอ่านและเขียนข้อมูลทีต่อกันนี้ จะเป็นไปตามแอดเดรสที่กำหนดทันที

8. SET DDRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

สำหรับการกำหนดแอดเดรสของ DDRAM เมื่อได้ทำการกำหนดไว้แล้วการอ่านและเขียนข้อมูลทีต่อกันนี้ จะเป็นไปตามแอดเดรสที่กำหนดทันที ตำแหน่งของแอดเดรสในแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนตัวอักษรต่อบรรทัดไม่เท่ากัน ซึ่งแสดงดังตารางต่อไปนี้

(ตารางนี้จะกำหนดให้บิตที่ 7 เท่ากับ 1 เสมอ เพื่อความสะดวกในการเรียกใช้)

รุ่น DMC161

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่น DMC162

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

รุ่น DMC164

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF

รุ่น DMC202

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	C	CF	D0	D1	D2	D3
														E					

9. BUSY FLAG AND ADDRESS READ

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	BF	ADDRESS							

สำหรับการอ่านค่า BF (Busy Flag) ซึ่งบอกถึงความพร้อมของแอสซีดีโมดูลในการรับข้อมูล ถ้า BF = 0 หมายความว่าพร้อมที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายความว่ายังไม่พร้อม นอกจากนี้ยังเป็น การอ่านค่าแอสซีดีของ CGRAM หรือ DDRAM ด้วย

3.9.2 การอ่านและเขียนข้อมูลกับ DDRAM/CGRAM

1. WRITE DATA TO DDRAM OR CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	DATA							

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเขียนข้อมูลลงหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้วแอดเดรสจะถูกเพิ่มหรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง Entry mode set และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้มีการกำหนดแอดเดรสที่ใด

2. READ DATA FROM DDRAM OR CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	DATA							

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้วแอดเดรสจะถูกเพิ่มหรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง Entry mode set และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้มีการกำหนดแอดเดรสที่ใด

3.9.3 แนวทางการเขียนโปรแกรมควบคุม

1. เมื่อจ่ายไฟเลี้ยงให้กับแอลซีดีโมดูลครั้งแรก ภายในจะมีการรีเซทระบบ โดยอัตโนมัติซึ่งจะใช้เวลา 10 วินาที หลังจากทีระดับแรงไฟขึ้นถึง 4.5 โวลต์ แล้วทั้งนี้ระบบรีเซท ดังกล่าวจะกระทำสิ่งต่าง ๆ ต่อไปดังนี้

- ทำการเคลียร์จอภาพทั้งหมด (Clear Display)
 - กำหนดคุณสมบัติด้วยคำสั่งฟังก์ชันเซทคือ DL = 1 (ติดต่อกับระบบไมโคร ๗ ในแบบ 8 bit) , N = 0 (แสดงข้อมูล 1 บรรทัด) , F = 0 (กำหนดตัวอักษรแบบ 5*7 คีอท)
 - กำหนดคุณสมบัติด้วยคำสั่ง Display ON/OFF คือ D = 0 (ไม่แสดงข้อมูล) , C = 0 (Cursor OFF) , B = 0 (Blank OFF)
 - กำหนดคุณสมบัติด้วยคำสั่ง Entry mode set คือ I/D = 1 (increment) , S = 0 (No shift)
- การใช้งานแอลซีดีโมดูลต้องรอให้ขบวนการรีเซทภายในทำงานเรียบร้อยก่อนซึ่งจะตรวจสอบได้ด้วย BF (Busy Flag) หรืออาจจะใช้การหน่วงเวลาก็ได้

2. การใช้งานแอลซีดีโมดูลจะต้องเกี่ยวข้องกับทางด้านโปรแกรมเป็นส่วนใหญ่ ชุดคำสั่งต่าง ๆ รวมทั้งการอ่านหรือเขียนข้อมูลนั้น จะถูกกำหนดด้วยขาสัญญาณทั้งหมดที่มีอยู่ปกติโปรแกรมจะต้องกำหนดคุณสมบัติต่าง ๆ ที่ต้องการไว้ที่ส่วนต้น จากนั้นก็จะเป็นการอ่านและเขียนข้อมูลลงใน DDRAM ซึ่งก็คือข้อความที่จะให้แสดงผลนั่นเอง

3.9.4 การโหลดอักขรที่ออกแบบเอง

แอลซีดีโมดูลจะสามารถโหลดตัวอักษรที่ออกแบบเองได้ โดยกระทำได้ 8 ตัวอักษรซึ่งหลักการก็คือให้ทำการโหลดรูปแบบที่กำหนดไว้แล้วลงในหน่วยความจำ CGRAM นั่นเอง แอดเดรสที่อ้างถึง CGRAM ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะใช้เพียง 6 บิต โดยอ้างได้ทั้งหมด 64 ไบต์ (คือ 8 ตัวอักษรคูณกับแถว) และการเรียกใช้ตัวอักษรที่ออกแบบไว้จะใช้รหัส 00 - 07 (หรือ 08 - 0F ก็ได้) ขั้นตอนในการโหลดตัวอักษรจะเป็นดังนี้

1. กำหนดแอดเดรสของ CGRAM ด้วยคำสั่งเซต CGRAM ทั้งนี้ถ้าโหลดอักษรตัวแรกก็จะใช้แอดเดรสที่ตำแหน่ง 0
2. เขียนข้อมูลที่เป็นรูปแบบของตัวอักษร ด้วยคำสั่งเขียนข้อมูล (WRITE DATA) โดยรูปแบบนี้จะใช้ 8 ไบต์ต่อหนึ่งตัวอักษร คือเรียงแถวจากแถวบนลงล่างและใช้ bit ที่ 0 - 4 เท่านั้น การโหลดนี้จะทำได้ต่อเนื่องไปเรื่อย ๆ โดยไม่ต้องกำหนดแอดเดรสใหม่
3. ตัวอักษรที่โหลดลงไปแล้ว สามารถเรียกใช้ได้ตลอดไปในขณะที่ยังมีไฟเลี้ยงอยู่ ซึ่งการโหลดนี้ให้กระทำที่ส่วนต้นของโปรแกรมได้เลย รูปแบบของตัวอักษรมีตัวอย่างดังนี้

					ADDRESS	DATA
		*			00	04
	*	*	*		01	0E
*	*	*	*	*	02	1F
*	*	*	*	*	03	1F
	*	*	*		04	0E
		*			05	04
		*			06	04
*	*	*	*	*	07	1F

ตารางที่ 3.4 แสดงตัวอย่างรูปแบบตัวอักษร

3.10 นาฬิกาเวลาจริง (Real time Clock)

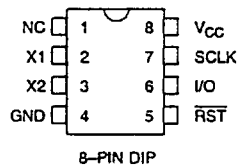
สำหรับการใช้งานระบบไมโครคอนโทรลเลอร์ที่มีเวลามาเกี่ยวข้องกับคัมย จำเป็นต้องมีวงจรในส่วนที่ทำหน้าที่เป็นอาร์ทีซี (Real time Clock) คือนาฬิกาเวลาจริง ซึ่งบอร์ดแอนท์ - 32 ใช้ชิพอาร์ทีซีเบอร์ DS1202 (Serial Timekeeper Chip) ของบริษัท ดัลลาสเซมิคอนดักเตอร์ โดยความร่วมมือกับอุปกรณ์ภายนอกเพียงเล็กน้อย และที่สำคัญคือ ไม่ต้องทำการปรับแต่ง ซึ่งเมื่อจะใช้อาร์ทีซีนี้ต้องมีชิพ DS1202 และ MAX691 รวมทั้งคริสตอล 32, 768 KHz และแบตเตอรี่ลิเธียมบนบอร์ดแอนท์ - 32 ด้วย

DS1202 เป็นชิพไอซีที่ให้เวลาเป็นจริงกับระบบไมโครโปรเซสเซอร์ โดยระบบไมโครโปรเซสเซอร์สามารถเซตค่าเวลาหรืออ่านค่าเวลาออกมาได้เป็นวินาที นาที ชั่วโมง วันและปี และมีการปรับวันสุดท้ายของเดือนรวมทั้งปรับเดือนกุมภาพันธ์ที่มี 29 วันให้โดยอัตโนมัติแล้วยังมีหน่วยความจำแรมให้ใช้งานอีก 24 ไบต์

DS1202 เชื่อมต่อกับระบบไมโครโปรเซสเซอร์หรือต่อโดยตรงกับไอซีทีทีแอล โดยการรับส่งข้อมูลแบบอนุกรม จึงใช้สายเพียง 3 เส้นคือ RESET, SCLK และ I/O ทำให้การอินเตอร์เฟตทำได้โดยง่ายและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

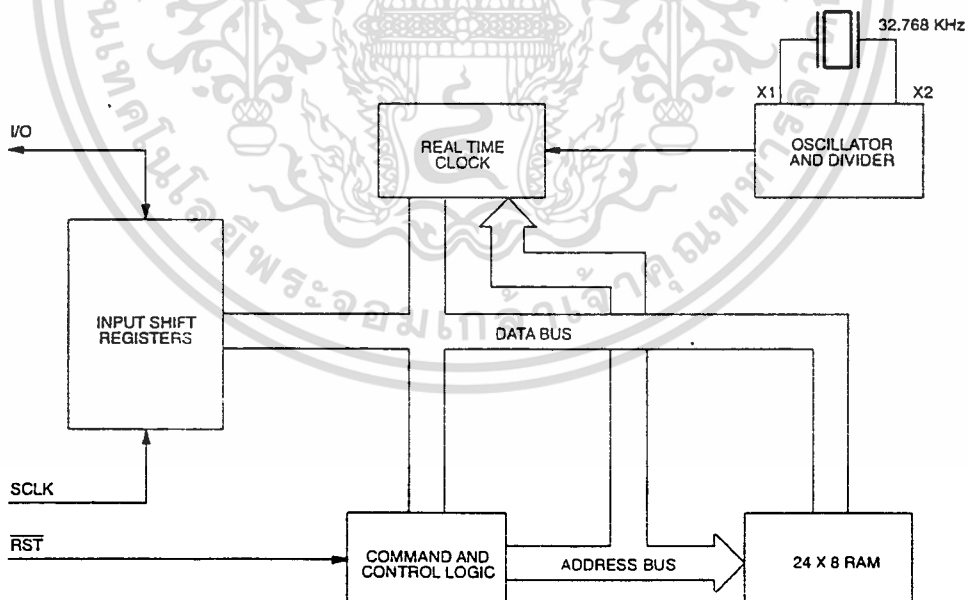
ระบบอาร์ทีซีมีขนาดเล็กมาก DS1202 ได้ถูกออกแบบมาให้ทำงานที่ก่่าลังต่ำ ๆ โดยจะเก็บรักษาข้อมูลนาฬิกา และข้อมูลในแรมได้ที่ไฟเลี้ยง 3 โวลต์ และกินกระแสเพียง 1 ไมโครแอมป์เท่านั้น



รูปที่ 3.10 แสดง DS1202

3. 10.1 หน้าที่และการใช้งานของแต่ละขาคือ

- X₁, X₂ ต่อกับคริสตอลความถี่ 32.768 กิโลเฮิร์ตซ์ เพื่อเป็นฐานเวลาให้กับวงจรรนาฬิกาในตัว DS1202
- RST เป็นขารีเซตเพื่อทำให้วงจรบางส่วนเริ่มต้นทำงานใหม่
- I/O สำหรับการเขียนและอ่านข้อมูลกับระบบไมโครโปรเซสเซอร์
- SCLK คล็อกแบบอนุกรม (Serial Clock) กำหนดการเคลื่อนย้ายข้อมูลแต่ละบิตในการรับและส่งข้อมูล
- Vcc ไฟเลี้ยง 2.0 - 5.5 โวลต์ โดยกินกระแสต่ำกว่า 300 นาโนแอมป์ที่ 2 โวลต์



รูปที่ 3.11 แสดงบล็อกไดอะแกรมส่วนต่าง ๆ DS 1202

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10.2 การทำงานของ DS 1202

รูปที่ 3.11 แสดงส่วนต่าง ๆ ภายใน DS 1202 ที่ประกอบด้วยวงจรถิพรีจิสเตอร์, วงจรกำเนิดความถี่, วงจรนาฬิกา, หน่วยความจำแรมและส่วนควบคุมต่าง ๆ การรับส่งข้อมูลจะทำเมื่อสัญญาณรีเซตเป็น "1" เท่านั้นและข้อมูลไบต์แรกจะกำหนดลักษณะคำสั่ง และแอดเดรสที่ต้องการติดต่อกับ การรับส่งข้อมูลแต่ละบิตจะถูกควบคุมจังหวะด้วยสัญญาณ SCLK (Serial Clock) โดย DS1202 จะรับข้อมูลอินพุตจากขา I/O ที่ขอบขาขึ้น (rising edge) ของสัญญาณ SCLK และจะให้ข้อมูลที่ถูกต้องออกจากขา I/O ที่ขอบขาลง (falling edge) ของสัญญาณ SCLK

3.10.3 การกำหนดแอดเดรสและรูปแบบของคำสั่ง

การติดต่อกับ DS 1202 จะต้องใช้ข้อมูลอย่างน้อย 2 ไบต์โดยไบต์แรกถือว่าเป็นคำสั่งและไบต์ที่สองคือข้อมูลที่จะรับเข้าไปหรือส่งออกมาจากรูปที่ 3 แสดงถึงรูปแบบของคำสั่ง โดยบิตที่ 0 จะบอกว่าเป็นการเขียนหรืออ่านข้อมูล บิตที่ 1 - 5 จะกำหนดแอดเดรสของนาฬิกาหรือแรมที่จะติดต่อกับ บิตที่ 6 จะติดต่อกับนาฬิกาหรือแรม และบิตที่ 7 เป็น "1" แสดงว่าไบต์นี้คือคำสั่ง เมื่อคำสั่งขนาด 1 ไบต์ถูกโหลดเข้าใน DS 1202 แล้ว จึงตามด้วยข้อมูลอีก 1 ไบต์ ในกรณีการเขียนข้อมูลหรือรับข้อมูล 1 ไบต์ออกมาในการอ่าน โดยทั้งคำสั่งและข้อมูลแต่ละบิต จะถูกควบคุมจังหวะด้วยสัญญาณ SCLK

3.10.4 การกำหนดให้เริ่มการทำงานและการควบคุมของ DS 1202

รูปที่ 3.12 แสดงรูปคลื่นของสัญญาณต่าง ๆ เมื่อเริ่มให้ DS 1202 ทำงานจะต้องกำหนดให้ขาเรเซตและ SCLK เป็น "0" เสียก่อนเพื่อให้วงจรบางส่วนเริ่มต้นทำงาน และให้วงจรถิพรีจิสเตอร์รับรู้ว่าข้อมูลไบต์แรกที่รับเข้ามาคือคำสั่ง และเมื่อสัญญาณรีเซตเป็น "0" แล้วจะทำให้ขา I/O เป็นไฮอิมพีแดนซ์ (High Impedance) เมื่อสิ้นสุดการรับส่งข้อมูลแล้วจะต้องทำให้สัญญาณรีเซตเป็น "0" อีกครั้งในขณะที่สัญญาณ SCLK เป็น "1" อยู่

3.10.5 การรับส่งข้อมูลกับ DS 1202

การรับส่งข้อมูลกับ DS 1202 จะกระทำโดยผ่านทางพอร์ทและจะเริ่มต้นรับส่งด้วยบิตที่ 0 จบที่บิตที่ 7 โดยสัญญาณรีเซตจะต้องเป็น "1" ตลอดช่องการรับและส่งข้อมูลในแต่ละครั้ง (ในแต่ละครั้งรับส่งข้อมูลได้ 2 ไบต์หรือมากกว่า)

3.10.6 ขั้นตอนการส่งข้อมูลให้ DS 1202

1. กำหนดให้ขา RESET และ SCLK เป็น "0"
2. ให้ขา RESET เป็น "1" เพื่อเริ่มต้นการส่งข้อมูล
3. กำหนดข้อมูลบิตแรกให้กับขา I/O
4. เปลี่ยนระดับสัญญาณ SCLK จาก "0" เป็น "1" ข้อมูลก็จะถูกโหลดเข้า DS 1202
5. ถ้าสิ้นสุดของข้อมูลทั้งหมดแล้วให้ขา RESET เป็น "0" อีกครั้ง
6. ถ้ายังมีข้อมูลที่จะส่งอีก ให้เปลี่ยนระดับสัญญาณ SCLK จาก "1" เป็น "0" แล้วทำตามขั้นตอนที่ 3

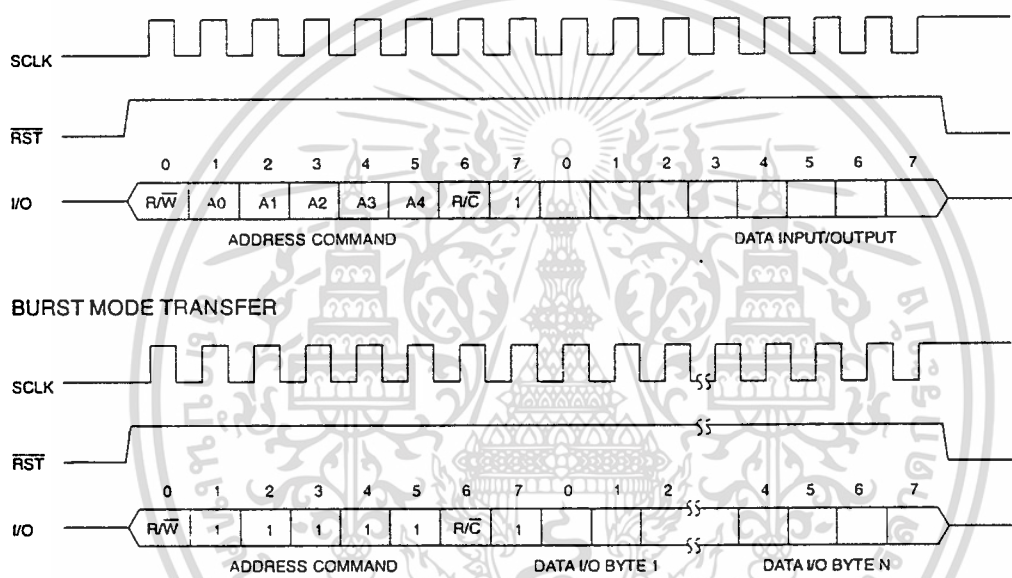
ใหม่จนกระทั่งหมดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10.7 ขั้นตอนการรับข้อมูลให้ DS 1202

1. กำหนดให้ขา RESET และ SCLK เป็น "0"
2. ให้ขา RESET เป็น "1" เพื่อเริ่มต้นการรับข้อมูล
3. ให้ขา SCLK เป็น "1"
4. เปลี่ยนระดับสัญญาณ SCLK จาก "1" เป็น "0"
5. อ่านข้อมูลบิตแรกจากขา I/O
6. เปลี่ยนระดับสัญญาณ SCLK จาก "0" เป็น "1"
7. ถ้าสิ้นสุดของข้อมูลทั้งหมดแล้วให้ขา RESET เป็น "0" อีกครั้ง
8. ถ้ายังมีข้อมูลที่จะรับอีกให้กลับไปทำขั้นตอนที่ 4 ใหม่จนกระทั่งหมดข้อมูล



รูปที่ 3.12 แสดงรูปคลื่นของการรับและส่งข้อมูล

การรับส่งข้อมูลนาฬิกาและแรมสามารถทำได้ 2 แบบคือข้อมูลไบต์เดียว (Single Byte) และแบบข้อมูลเป็นชุด (Burst Mode) ความแตกต่างของทั้งสองแบบมีดังนี้

- ข้อมูลไบต์เดียว เป็นการรับส่งที่จะต้องมีการรับส่งไบต์คำสั่ง (command byte) นำหน้าทุกครั้ง
- ข้อมูลเป็นชุด เป็นการรับส่งที่กำหนดไบต์คำสั่งนำหน้าเพียงครั้งเดียวในการรับส่งข้อมูลแต่ละครั้ง และไบต์ต่อไปจะเป็นข้อมูลทั้งหมด โดยบิตที่ 1 - 5 ในไบต์คำสั่งจะต้องเป็น 1 ทั้งหมด การรับส่งข้อมูลในโหมดนี้ทำการติดต่อกับ DS 1202 สะดวกมากถึงแม้ว่าจะมีลักษณะอนุกรมก็ไม่ใช่ว่าเรื่องยากหรือเกิดความเข้าใจในแง่การใช้งาน

3.10.8 การจัดแอดเดรสใน DS 1202

REGISTER	FUNCTION	COMMAND ADDRESS (HEX)	WRITE=W READ=R	RANGE DATA (BCD)	REGISTER DEFINITION							
					7	6	5	4	3	2	1	0
0	SECONDS	80	W	00-59	CH	10 SEC				SEC		
		81	R									
1	MINUTES	82	W	00-59	0	10 MIN				MIN		
		83	R									
2	12 HRS	84	W	01-12	12V	0	AP	HR	HOU			
	24 HRS	85	R	00-23	24	0	10	HR	R			
3	DATE	86	W	01-31	0	0	10 DATE			DATE		
		87	R									
4	MONTH	88	W	01-12	0	0	0	10 M	MON			
		89	R					TH				
5	DAY	8A	W	01-07	0	0	0	0	DAY			
		8B	R									
6	YEAR	8C	W	00-99	10 YEAR				YEA			
		8D	R		R							
7	WRITE	8E	W	00-80	WP	ALWAYS ZERO						
	PROTECT	8F	R									

ตารางที่ 3.5 แสดงคำสั่งและการจัดแอดเดรส

จากตารางที่ 3.5 แสดงถึงการจัดตำแหน่งของรีจิสเตอร์ที่เก็บค่าเวลาและหน่วยความจำแรม จะเห็นว่าข้อมูลที่เก็บค่าเวลาจะเป็นแบบ BCD และเก็บเท่าที่จำเป็นเท่านั้น เช่นที่แอดเดรส 82 บิตที่ 0-3 จะเป็นหลักหน่วยของนาฬิกา และบิตที่ 4-6 จะเป็นหลักสิบของนาฬิกา (เพราะค่ามากที่สุดคือ 59) ส่วนบิตที่เหลือก็จะ เป็น 0 ไป

ส่วนของรีจิสเตอร์นาฬิกาจะอยู่ที่แอดเดรส 80 - 8D และหน่วยความจำแรมทั้งหมด 24 ไบต์ที่สามารถใช้เก็บข้อมูลได้อย่างอิสระจะอยู่ที่แอดเดรส C0 - EE ส่วนแอดเดรสที่ BE, BF, FE และ FF จะถูกอ้างอิงเมื่อทำงานในลักษณะข้อมูลเป็นชุดเท่านั้น ที่แอดเดรส 8E, 8F เป็นรีจิสเตอร์ป้องกันการเขียนทับ (Write Protect Register) สำหรับควบคุมการเขียนข้อมูลลงในรีจิสเตอร์นาฬิกาหรือแรม โดยถ้าบิตที่ 7 นี้ เป็น "1" จะไม่สามารถเขียนข้อมูลลงในรีจิสเตอร์นาฬิกาหรือแรมได้ และเมื่อจ่ายไฟให้กับ DS 1202 บิตนี้จะถูกเซตให้เป็น "1" อัตโนมัติ ก่อนการเปลี่ยนแปลงข้อมูลใด ๆ จึงต้องทำให้บิตที่ 7 เป็น "0" เสียก่อน

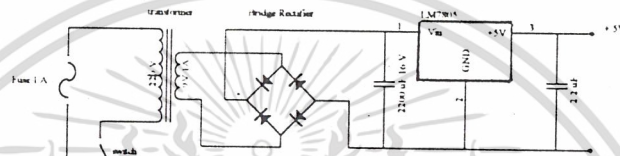
3.11 แหล่งจ่ายไฟ 5 โวลต์

ในโครงการนี้จะต้องใช้แหล่งจ่ายไฟตรง 5 โวลต์ต่อให้กับไอซีทีทีแอล ซึ่งต้องการแหล่งจ่ายไฟ 5 โวลต์ ± 5 เปอร์เซ็นต์ แหล่งจ่ายไฟนี้ถูกออกแบบให้มีความสามารถในการจ่ายกระแสได้ประมาณ 1 แอมแปร์ เพราะต้องใช้ไอซีหลายตัวในวงจร

หลักการทำงานของวงจร คือ ใช้หม้อแปลงแปลงไฟ 220 โวลต์เป็นไฟแรงดันต่ำขนาด 9 โวลต์

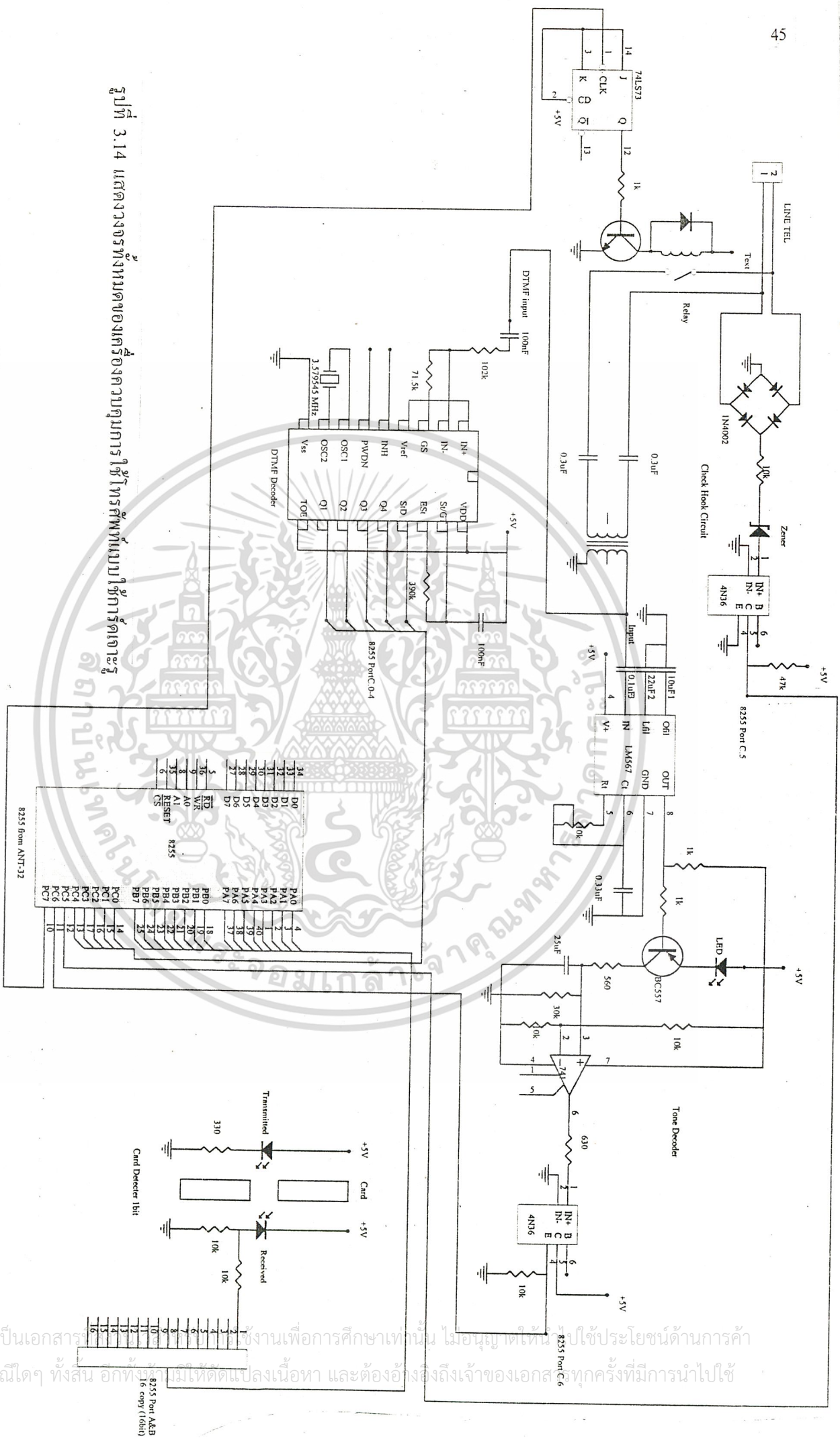
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเรกติไฟไฟสลับนี้ด้วยบริดจ์เรกติไฟ (bridge rectifier) ซึ่งภายในมีไดโอดอยู่ 4 ตัว เมื่อแปลงไฟสลับให้เป็นไฟตรง กรองไฟตรงที่ได้ให้เรียบโดยใช้คอนเดนเซอร์ตัวใหญ่ จากนั้นป้อนไฟตรงที่เรียบนี้ให้อิซีเรกูเลเตอร์ (regulator IC) ซึ่งทำหน้าที่ลดแรงดันไฟลงมาเหลือ 5 โวลต์ และสามารถคงค่าแรงดัน 5 โวลต์นี้ได้ตลอด แม้ว่าจ่ายกระแสไหลออกไปหรือไฟบ้านตกหรือเกินก็ตามแรงดันขาออกจะได้ตามสเป็คคือ 5 โวลต์ \pm 5 เปอร์เซ็นต์อย่างแน่นอน การใช้งานของเรกูเลเตอร์นั้นที่ขาออกของแหล่งจ่ายไฟต่อตัวเก็บประจุแทนทาลัมไว้ เมื่อกำลังสัญญาณรบกวนความถี่สูงที่อาจเกิดจากตัวเรกูเลเตอร์เองหรือจากวงจรไอซีที่ที่แอลที่เร่าต่อ



รูปที่ 3.13 แสดงวงจรแหล่งจ่ายไฟ 5 โวลต์ 1 แอมแปร์ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงวงจรทั้งหมดของเครื่องควบคุมการใช้โทรศัพท์แบบใช้การ์ดเจาะรู

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากทางสถาบันฯ หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง

8255 Port C.4	Q1	TOE	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	Q31	Q32	Q33	Q34
8255 Port C.5	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	Q31	Q32	Q33	Q34	

8255 Port A/B	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA8	PA9	PA10	PA11	PA12	PA13	PA14	PA15	PA16	PA17	PA18	PA19	PA20	PA21	PA22	PA23	PA24	PA25	PA26	PA27	PA28	PA29	PA30	PA31	PA32	PA33	PA34
8255 Port B	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7	PB8	PB9	PB10	PB11	PB12	PB13	PB14	PB15	PB16	PB17	PB18	PB19	PB20	PB21	PB22	PB23	PB24	PB25	PB26	PB27	PB28	PB29	PB30	PB31	PB32	PB33	PB34

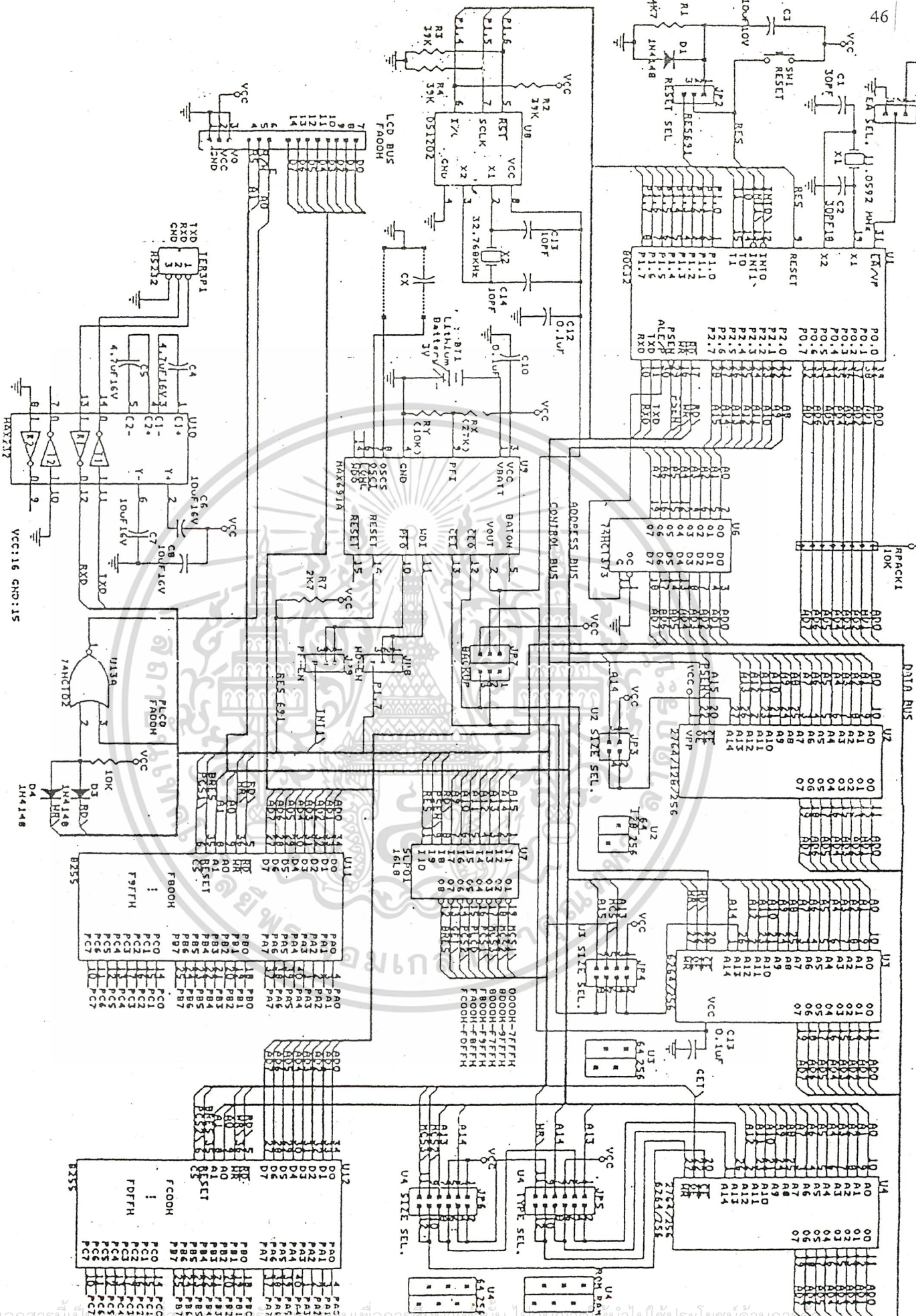
8255 Port C.6	CA0	CA1	CA2	CA3	CA4	CA5	CA6	CA7	CA8	CA9	CA10	CA11	CA12	CA13	CA14	CA15	CA16	CA17	CA18	CA19	CA20	CA21	CA22	CA23	CA24	CA25	CA26	CA27	CA28	CA29	CA30	CA31	CA32	CA33	CA34
8255 Port C.7	CB0	CB1	CB2	CB3	CB4	CB5	CB6	CB7	CB8	CB9	CB10	CB11	CB12	CB13	CB14	CB15	CB16	CB17	CB18	CB19	CB20	CB21	CB22	CB23	CB24	CB25	CB26	CB27	CB28	CB29	CB30	CB31	CB32	CB33	CB34

8255 Port A/B	DA0	DA1	DA2	DA3	DA4	DA5	DA6	DA7	DA8	DA9	DA10	DA11	DA12	DA13	DA14	DA15	DA16	DA17	DA18	DA19	DA20	DA21	DA22	DA23	DA24	DA25	DA26	DA27	DA28	DA29	DA30	DA31	DA32	DA33	DA34
8255 Port B	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8	DB9	DB10	DB11	DB12	DB13	DB14	DB15	DB16	DB17	DB18	DB19	DB20	DB21	DB22	DB23	DB24	DB25	DB26	DB27	DB28	DB29	DB30	DB31	DB32	DB33	DB34

8255 Port C.6	CA0	CA1	CA2	CA3	CA4	CA5	CA6	CA7	CA8	CA9	CA10	CA11	CA12	CA13	CA14	CA15	CA16	CA17	CA18	CA19	CA20	CA21	CA22	CA23	CA24	CA25	CA26	CA27	CA28	CA29	CA30	CA31	CA32	CA33	CA34
8255 Port C.7	CB0	CB1	CB2	CB3	CB4	CB5	CB6	CB7	CB8	CB9	CB10	CB11	CB12	CB13	CB14	CB15	CB16	CB17	CB18	CB19	CB20	CB21	CB22	CB23	CB24	CB25	CB26	CB27	CB28	CB29	CB30	CB31	CB32	CB33	CB34

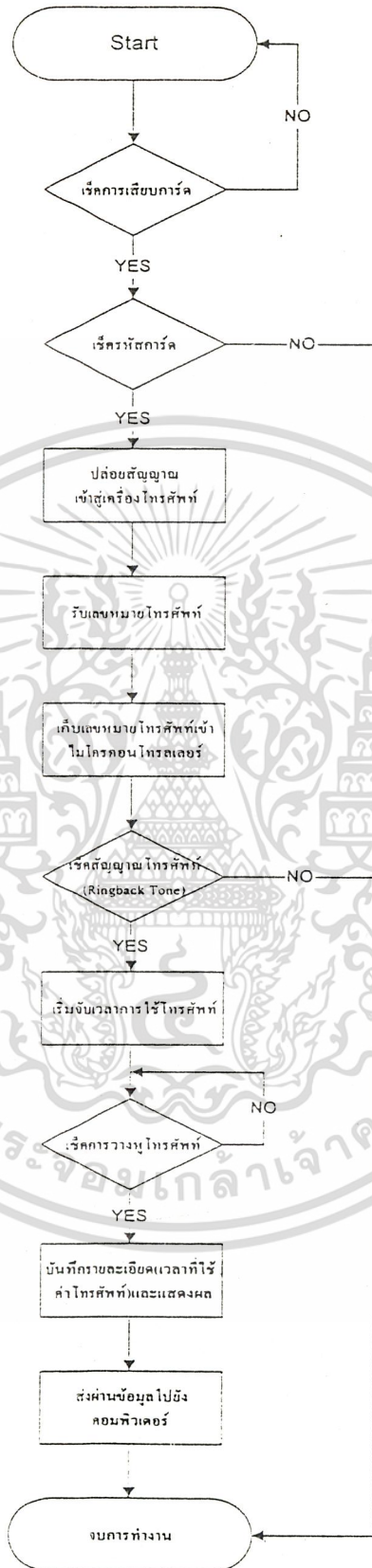
8255 Port A/B	DA0	DA1	DA2	DA3	DA4	DA5	DA6	DA7	DA8	DA9	DA10	DA11	DA12	DA13	DA14	DA15	DA16	DA17	DA18	DA19	DA20	DA21	DA22	DA23	DA24	DA25	DA26	DA27	DA28	DA29	DA30	DA31	DA32	DA33	DA34
8255 Port B	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8	DB9	DB10	DB11	DB12	DB13	DB14	DB15	DB16	DB17	DB18	DB19	DB20	DB21	DB22	DB23	DB24	DB25	DB26	DB27	DB28	DB29	DB30	DB31	DB32	DB33	DB34

8255 from ANT-32
16 copy (16bit)

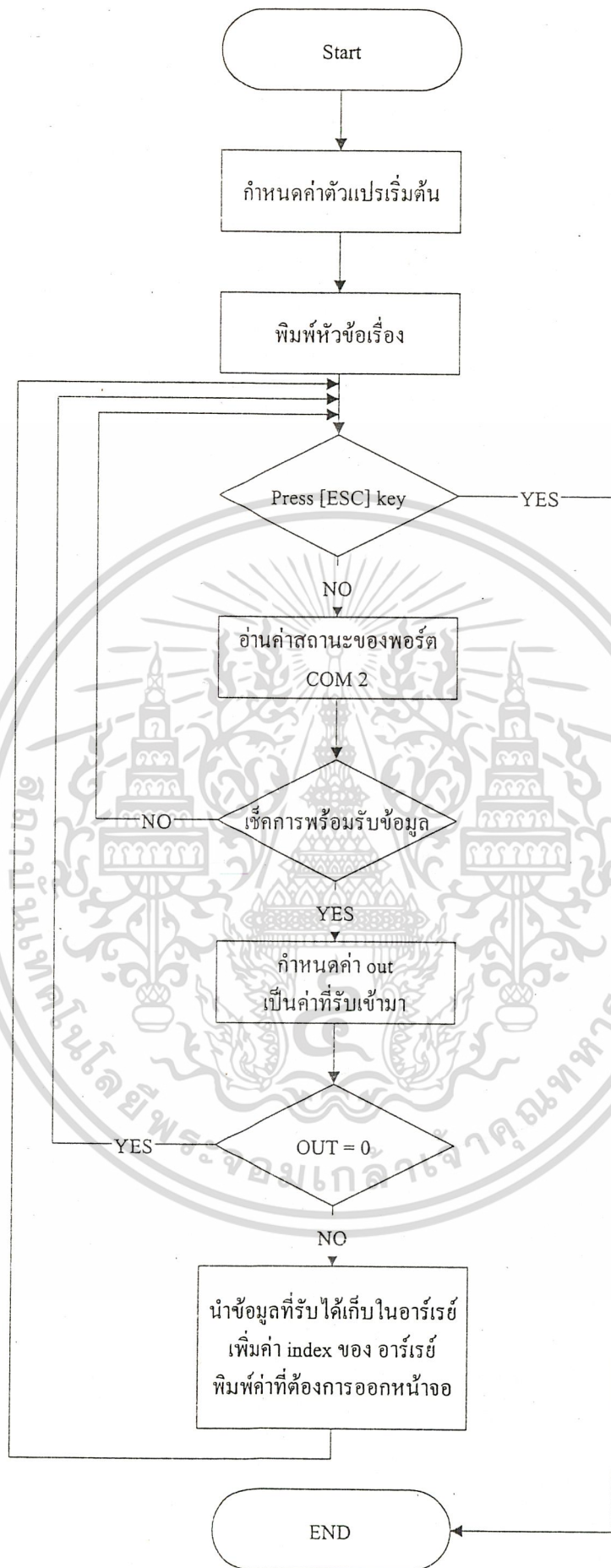


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ ใช้ในเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกา
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งรูปที่ 3.15 แสดงวงจรบอร์ดไมโครคอนโทรลเลอร์ ANT-32 ทุกครั้งที่มีการนำไปใช้

รูปที่ 3.16 โฟลชาร์ตแสดงขั้นตอนการทำงานของเครื่องควบคุมการใช้โทรศัพท์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แสดงโฟลชาร์ตขั้นตอนการทำงานของกรับข้อมูลจากพอร์ตคอมพิวเตอร์ของพีซี เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงข้อความที่ปรากฏบนหน้าจอพีซีขณะทำการรับข้อมูลตามโปรแกรมที่ออกแบบ

(Press [ESC] to exit ...)

LIST OF TELEPHONE

CODE	TELEPHONE NO.	START	END	TOTAL	CHARGE
3333	4725421	15.34	15.56	00.22	3 baht
1111	034258497	17.23	17.31	00.08	24 baht



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

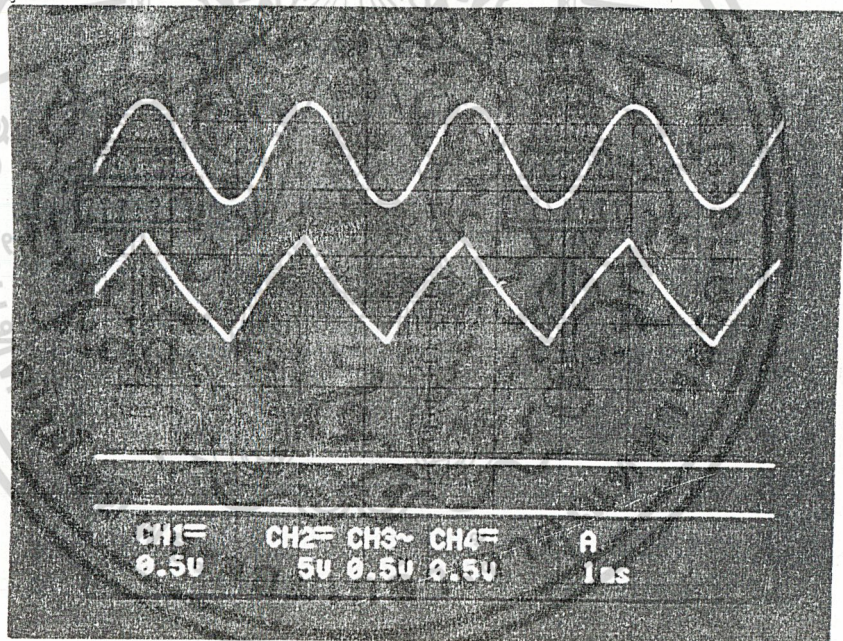
ผลการทดลอง

การทดลองโครงงานในภาคการศึกษานี้ได้ทำการทดลองในส่วนที่เป็น วงจร tone decoder , วงจร DTMF decoder , วงจรตรวจสอบขงหูวางโทรศัพท์ , ส่วนแสดงผลจอ LCD และ ส่วนอ่านรหัสการ์ดเจาะรู

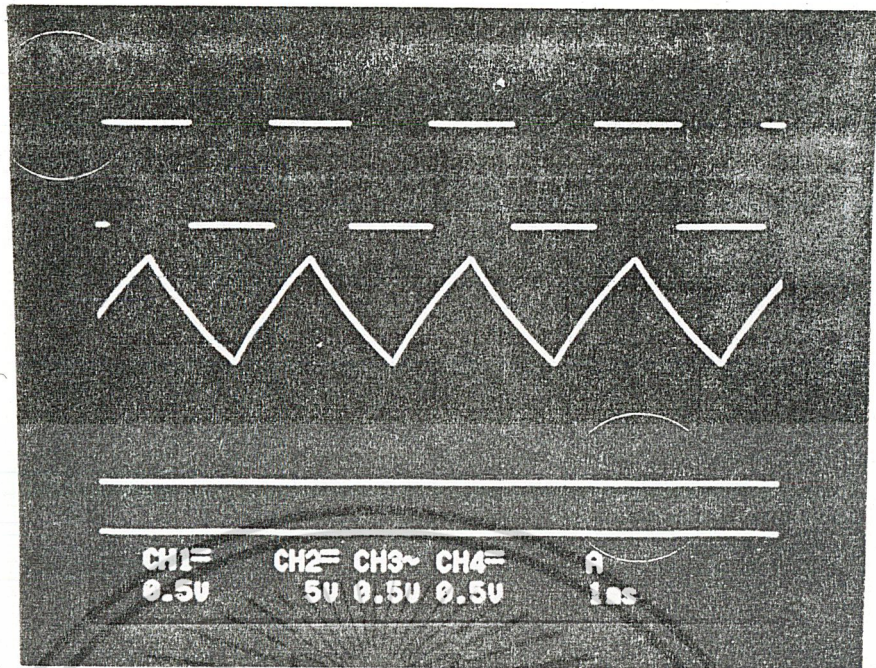
ในส่วนอ่านรหัสการ์ดเจาะรูนั้นจะทำการต่อเข้ากับไมโครคอนโทรลเลอร์ซึ่งทำงานตามโปรแกรมที่เขียนไว้และ ในส่วนแสดงผลจอ LCD ก็จะทำการต่อเข้ากับไมโครคอนโทรลเลอร์เพื่อให้จอ LCD สามารถแสดงตัวอักษรตามที่เรต้องการได้ เพื่อแสดงสถานะการทำงานของไมโครคอนโทรลเลอร์ขณะนั้น

4.1 วงจร tone decoder

เมื่อทำการป้อนสัญญาณอินพุตที่มีความถี่ 400 เฮิร์ต เข้าไปยังไอซีเบอร์ LM 567 ซึ่งเป็น tone decoder ที่เราทำการปรับค่าตัวความต้านทานและตัวเก็บประจุเพื่อให้มีความถี่ที่เราต้องการทดสอบ 400 เฮิร์ต ผลการทดลองที่ได้คือไม่ว่าสัญญาณอินพุตจะมีลักษณะอย่างไรถ้ามีความถี่ตรงกันกับความที่เราต้องการทดสอบ สัญญาณเอาต์พุตที่ออกจากไอซี LM 567 นั้นจะมีสถานะลอจิกต่ำที่ค่า 50 มิลลิโวลต์ และเมื่อนำสัญญาณเอาต์พุตจากไอซี LM 567 ผ่านวงจรเปรียบเทียบสัญญาณแรงดันและ opto coupler จะเปลี่ยนสถานะจากลอจิกต่ำเป็นสูงที่ค่า 4.2 โวลต์



รูปที่ 4.1 แสดงสัญญาณที่วัดได้เมื่อทำการป้อนสัญญาณอินพุตเป็นรูปไซน์ที่มีความถี่ 400 เฮิร์ต



รูปที่ 4.2 แสดงสัญญาณที่วัดได้เมื่อทำการป้อนสัญญาณอินพุต เป็นรูปสี่เหลี่ยมที่มีความถี่ 400 เฮิรต์

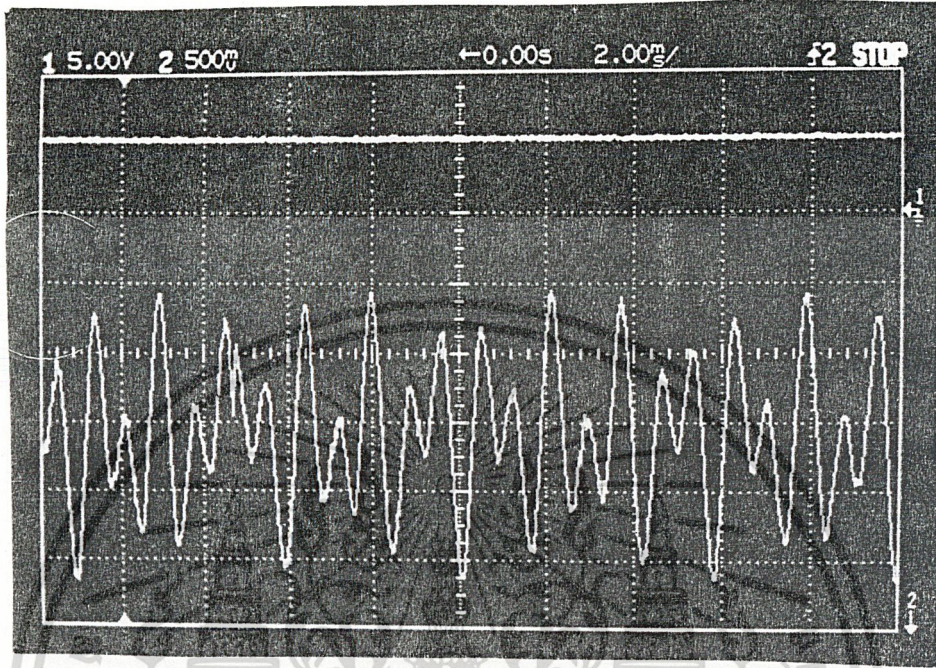
4.2 วงจร DTMF decoder

เมื่อทำการป้อนสัญญาณ DTMF จากโทรศัพท์เข้าไปยังวงจร DTMF Decoder จะได้ผลการทดลองตามตารางดังนี้

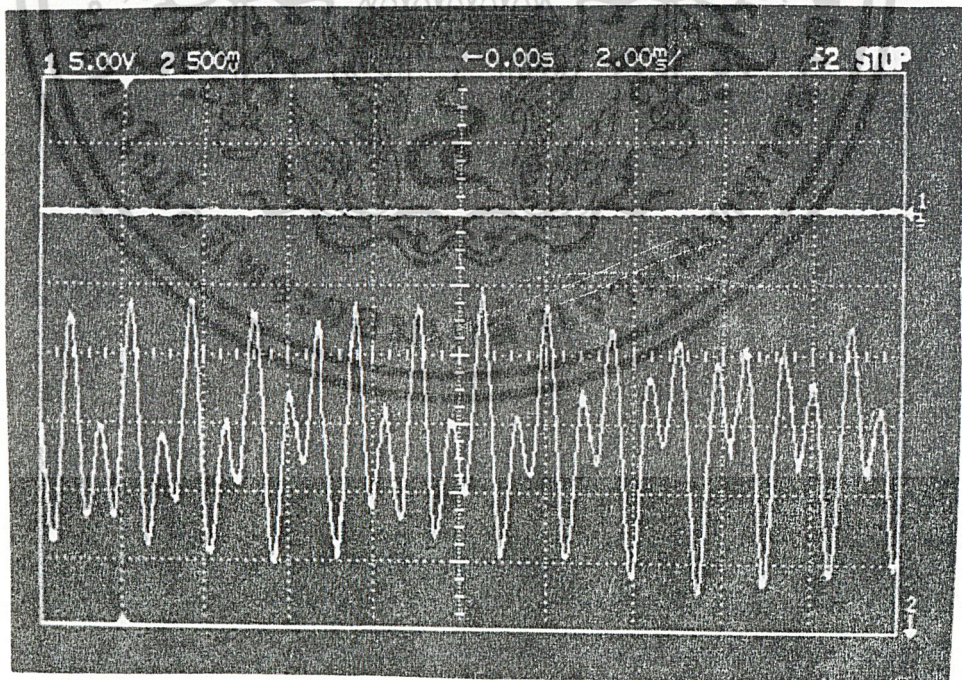
F low	F high	NO	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0V	0V	0V	5V
697	1336	2	H	0V	0V	5V	0V
697	1447	3	H	0V	0V	5V	5V
770	1209	4	H	0V	5V	0V	0V
770	1336	5	H	0V	5V	0V	5V
770	1447	6	H	0V	5V	5V	0V
852	1209	7	H	0V	5V	5V	5V
852	1336	8	H	5V	0V	0V	0V
852	1447	9	H	5V	0V	0V	5V
941	1336	0	H	5V	0V	5V	0V
941	1209	*	H	5V	0V	5V	5V
941	1447	#	H	5V	5V	0V	0V

ตาราง 4.1 แสดงผลการทดลองการวัดค่าที่ได้จากวงจรตรวจสอบสัญญาณ DTMF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงสัญญาณ DTMF เมื่อกดโทรศัพท์หมายเลข 1 และทำการวัดเอาต์พุตที่ขา Q1 จะมีค่า 5 โวลต์

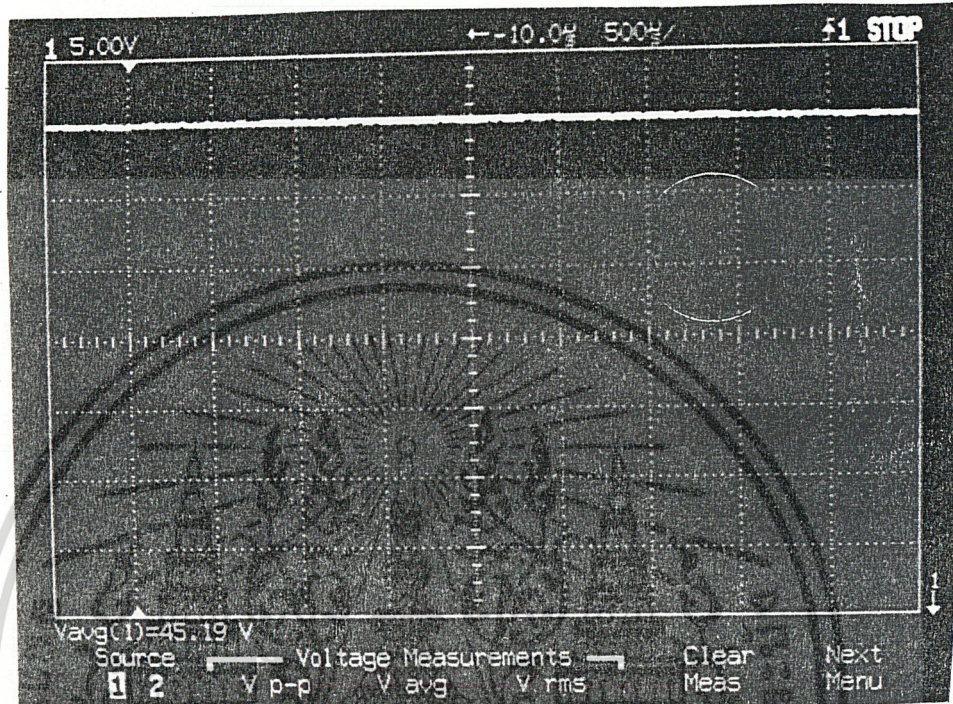


รูปที่ 4.4 แสดงสัญญาณ DTMF เมื่อกดโทรศัพท์หมายเลข 2 และทำการวัดเอาต์พุตที่ขา Q1 จะมีค่า 0 โวลต์

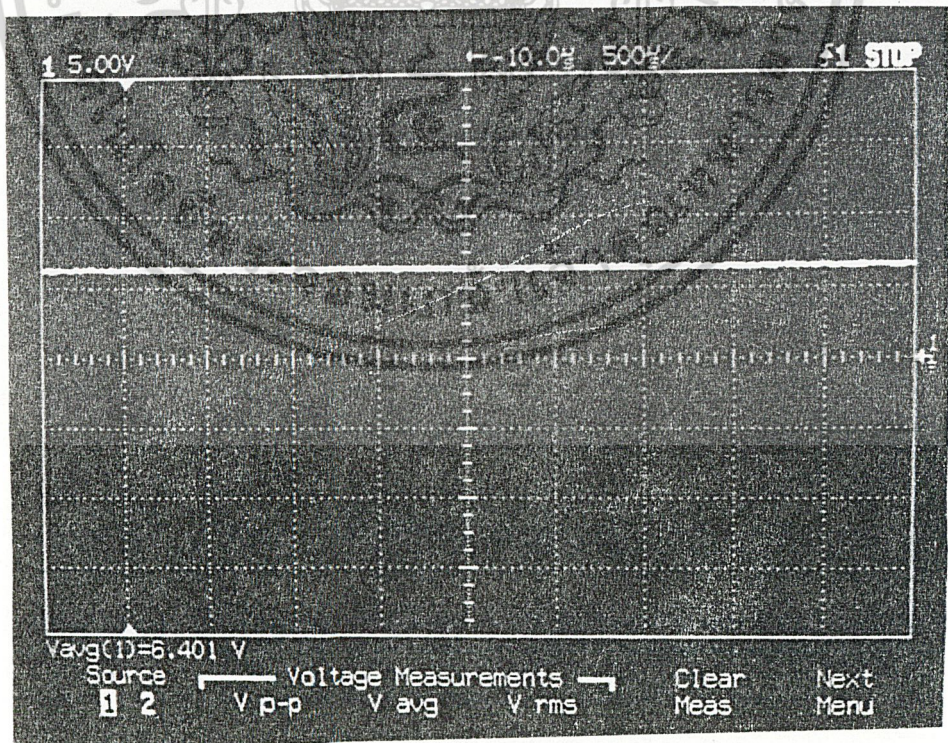
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 วงจรตรวจสอบขงหวางหุโทรศัพท

เมื่อทำการวัดสัญญาณที่คู่สายโทรศัพท์ขณะยกหูและวางหูโทรศัพท์ จะได้สัญญาณขณะที่โทรศัพท์วางหุมือค่า 45.19 โวลท์ และสัญญาณขณะยกหุมือค่า 6.401 โวลท์ ซึ่งเมื่อนำสัญญาณนี้ไปผ่านวงจรตรวจสอบขงหวางหุโทรศัพทจะได้เอาท์พุทขณะที่โทรศัพท์วางหุมือค่าสถานะต่ำที่ 0 โวลท์ และจะได้เอาท์พุทขณะยกหุมือค่าสถานะสูงที่ 4.8 โวลท์



รูปที่ 4.5 แสดงสัญญาณโทรศัพท์ขณะที่วางหุมือค่า 45.19 โวลท์



รูปที่ 4.6 แสดงสัญญาณโทรศัพท์ขณะที่ยกหุมือค่า 6.401 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ส่วนอ่านรหัสการ์ดเจาะรู

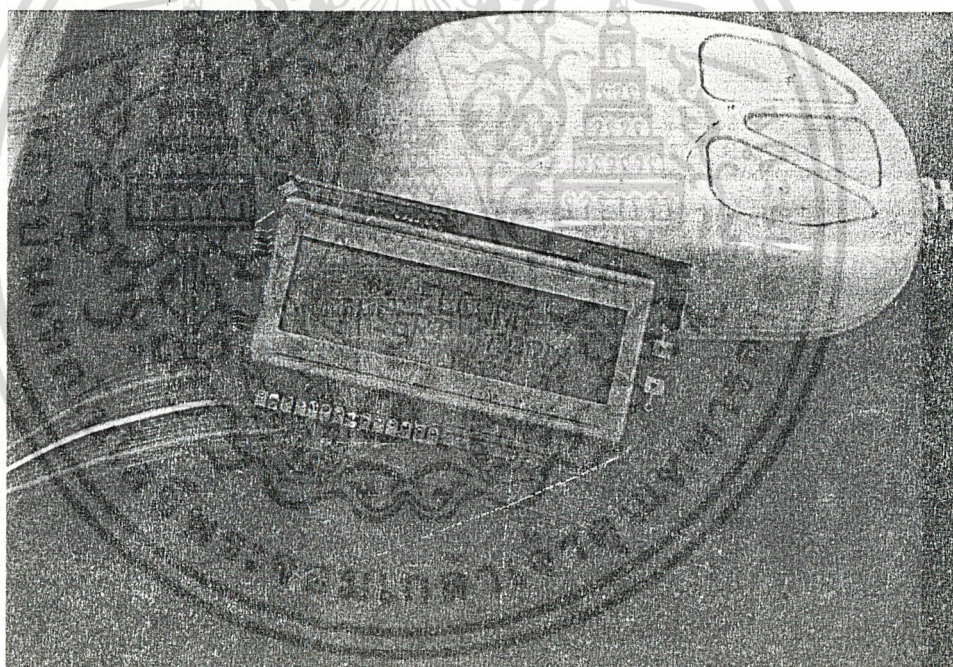
เมื่อทำการวัดสัญญาณเอาต์พุตจากการอ่านสถานะของการ์ด จะได้ผลการทดลองดังนี้

1. ตำแหน่งที่เราตรวจสอบมีการเจาะรู ค่าเอาต์พุตที่ได้มีค่า 0.7 โวลต์ ซึ่งถือเป็นสถานะลอจิกต่ำ
2. ตำแหน่งที่เราตรวจสอบไม่มีการเจาะรู ค่าเอาต์พุตที่ได้มีค่า 4 โวลต์ ซึ่งถือเป็นสถานะลอจิกสูง

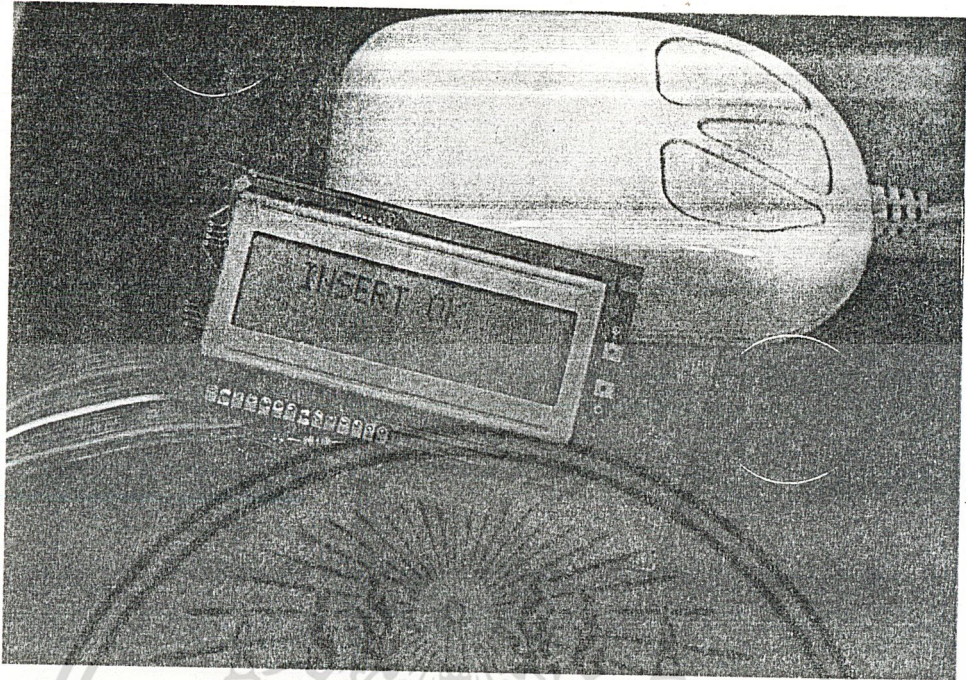
เมื่อนำสัญญาณเอาต์พุตจากการอ่านการ์ดเข้าสู่ไมโครคอนโทรลเลอร์โดยผ่าน 8255 พอร์ต เพื่อให้ไมโครคอนโทรลเลอร์ทำการเปรียบเทียบรหัสที่เข้ามากับรหัสที่ได้ทำการบันทึก โปรแกรมที่เขียนขึ้นสำหรับการตรวจสอบรหัสการ์ดเพื่อให้ไมโครคอนโทรลเลอร์ทำงานมีโปรแกรมดังภาคผนวกที่ 2

4.5 ส่วนแสดงผลของโปรแกรมทางจอ LCD

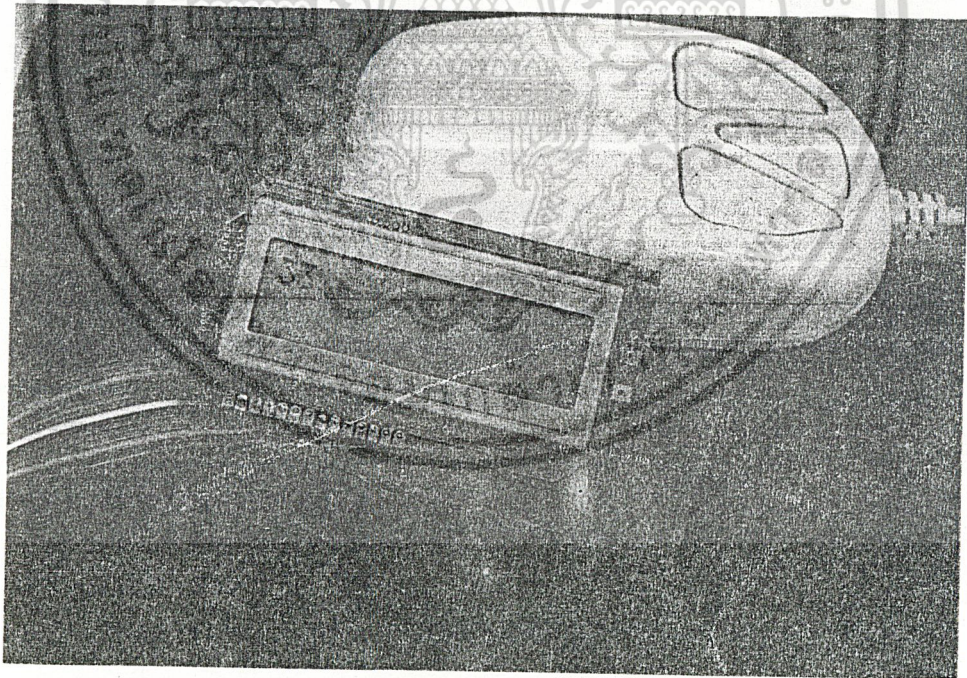
เมื่อทำการเขียน โปรแกรมเพื่อทำการทดสอบการทำงานของเครื่องควบคุมการใช้โทรศัพท์แบบใช้การ์ดเจาะรู ซึ่งได้ทำการเขียนโปรแกรมทดสอบตามขั้นตอนต่าง ๆ (แสดงดังภาคผนวกที่ 2) ได้ผลดังนี้



รูปที่ 4.7 แสดงข้อความขณะเครื่องเริ่มทำงาน

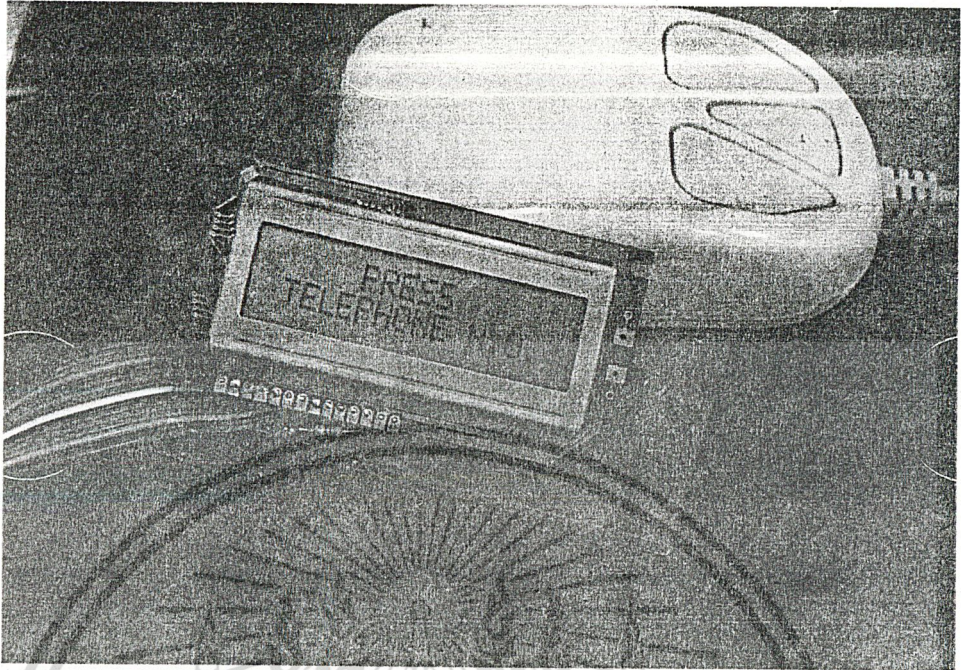


รูปที่ 4.8 แสดงข้อความขณะทำการสอดการ์ดถูกต้อง

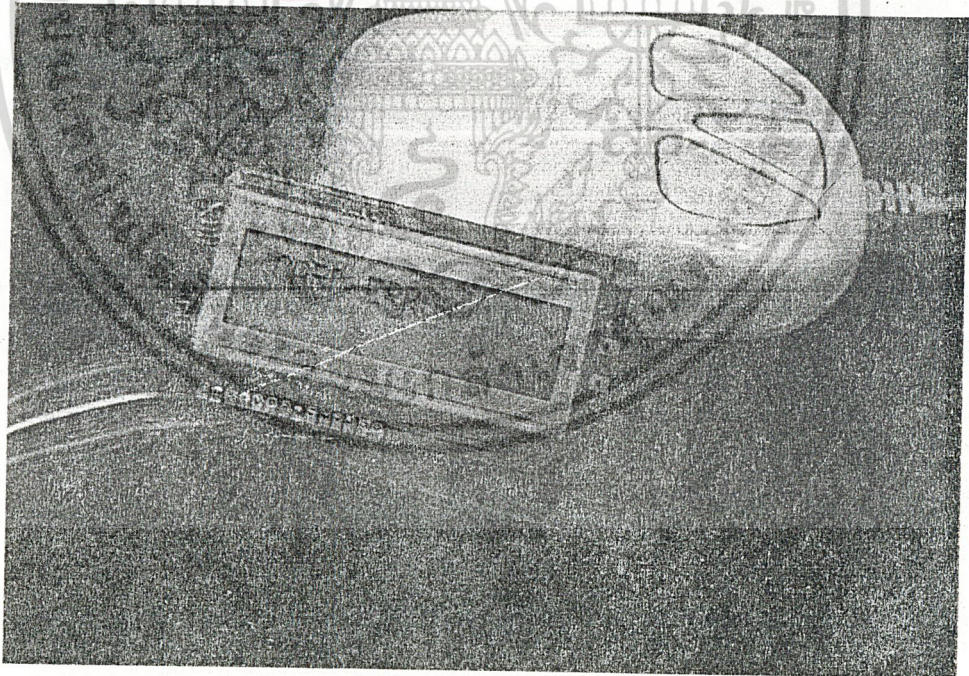


รูปที่ 4.9 แสดงข้อความขณะทำการตรวจสอบรหัสจากการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

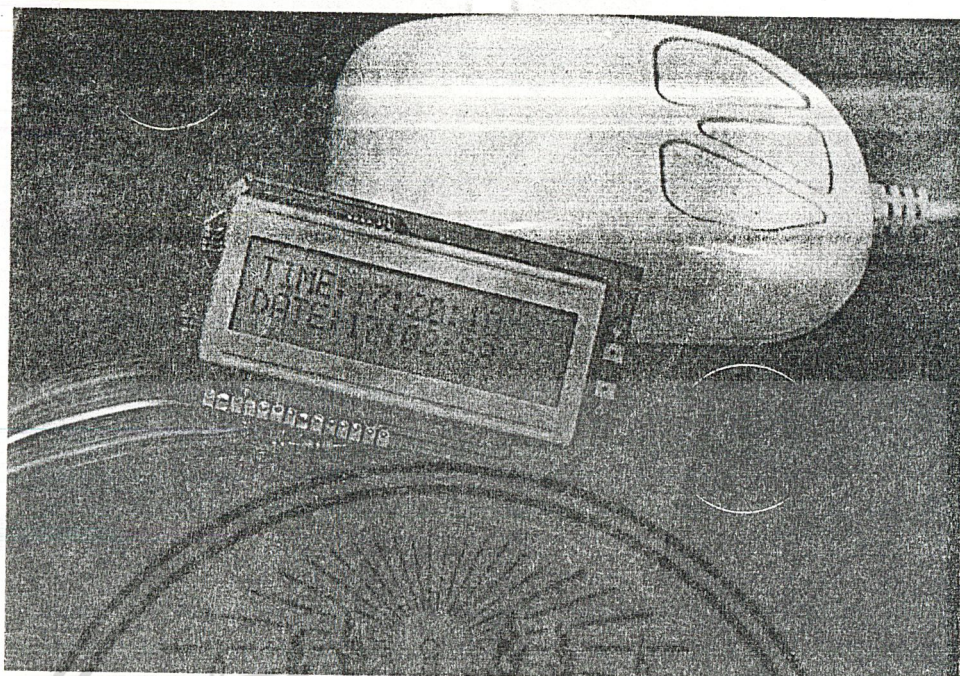


รูปที่ 4.10 แสดงข้อความเมื่อทำการดีเทคการ์ดถูกต้อง

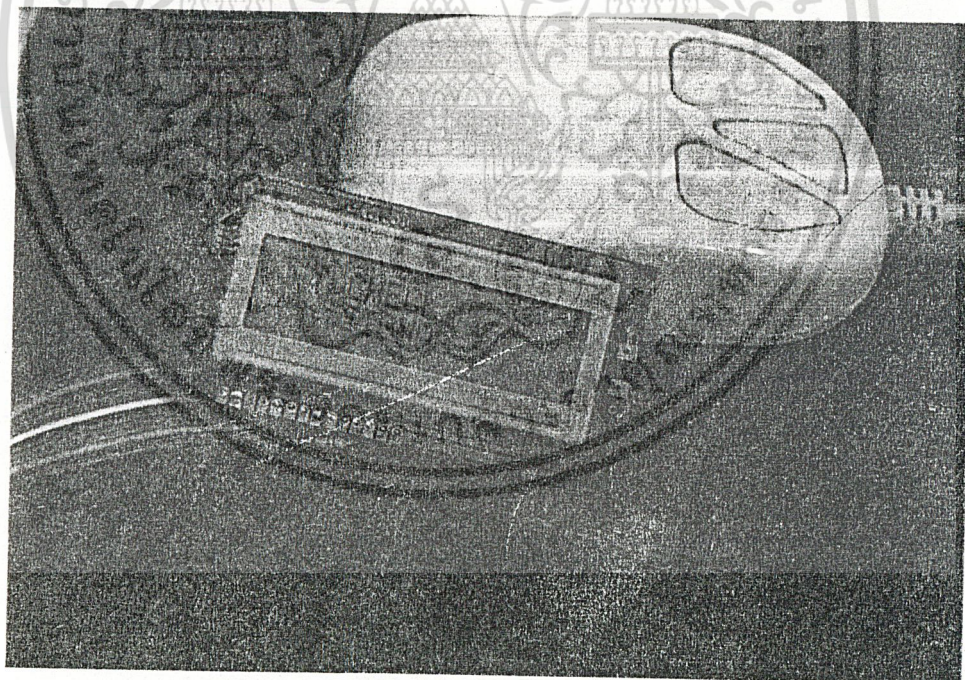


รูปที่ 4.11 แสดงข้อความขณะที่การ์ดไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงเวลาขณะใช้โทรศัพท์

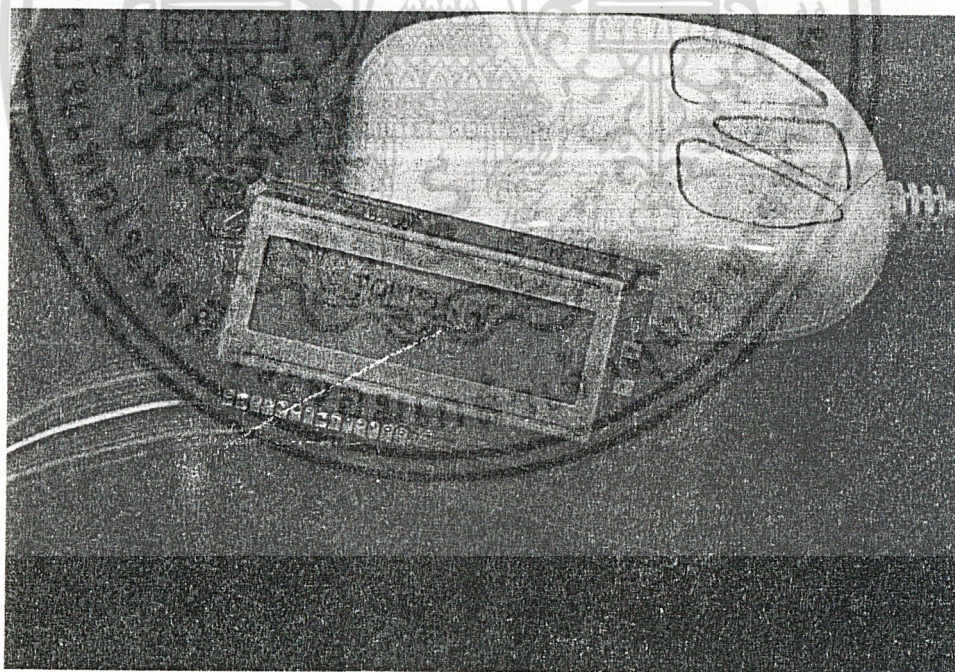


รูปที่ 4.13 แสดงข้อความขณะตรวจสอบสัญญาณ โทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

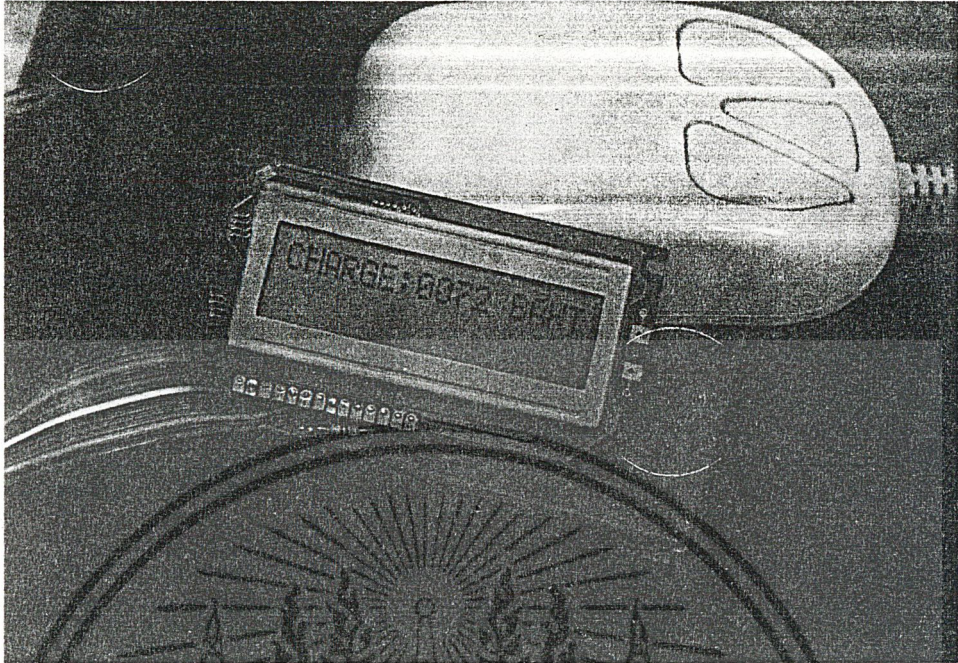


รูปที่ 4.14 แสดงข้อความขณะสายโทรศัพท์ที่ไม่ว่าง

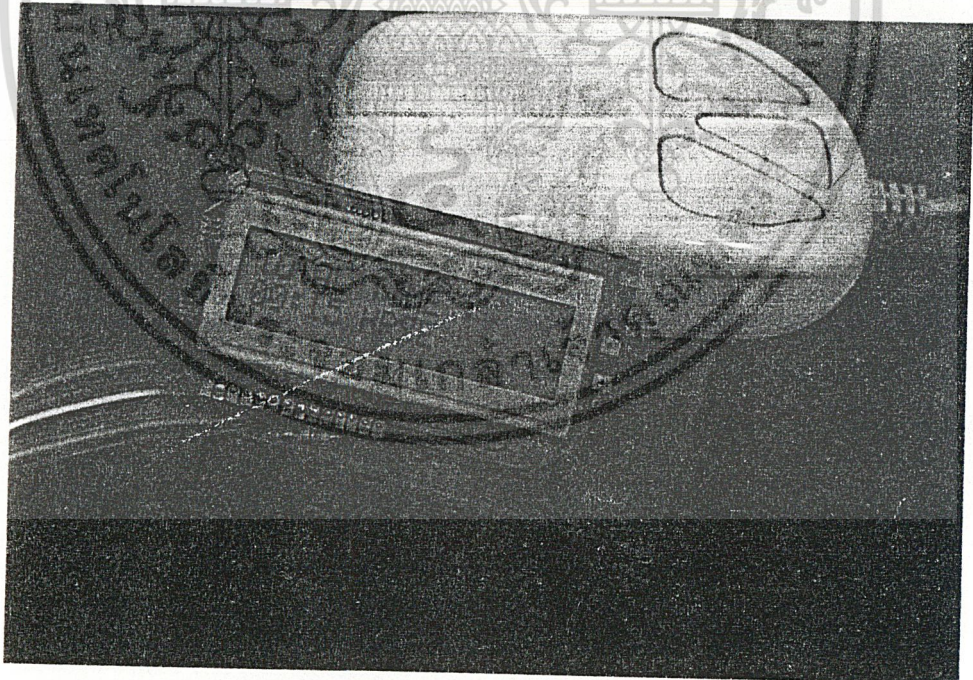


รูปที่ 4.15 แสดงข้อความขณะที่กำลังสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 แสดงค่าบริการการใช้โทรศัพท์



รูปที่ 4.17 แสดงเวลาที่ทำการใช้โทรศัพท์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์

จากการที่ได้ทำการทดลองส่วนต่าง ๆ ของโครงงานนี้ ซึ่งได้ทำการทดลองในส่วนของ วงจรตรวจสอบรหัสจากการ์ดชนิดเจาะรู วงจรตรวจสอบการยกหูวางหู วงจรตรวจสอบสัญญาณเรียกกลับ วงจรสวิทช์ปล่อยสัญญาณโทรศัพท์ และการเขียนโปรแกรมเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ให้ทำงานร่วมกับวงจรส่วนต่าง ๆ ได้อย่างถูกต้องตามที่ต้องการ และแสดงออกทางหน้าจอLCD ซึ่งจากการทดลองนั้นได้ผลเป็นไปตามที่ต้องการ โดยโปรแกรมที่ได้ทำการเขียนและประมวลผลนั้นสามารถทำงานได้ถูกต้อง

ส่วนวงจรที่ใช้ในการทดลองนี้ ที่ได้ทำการทดลองแล้วนั้น ได้ผลการทดลองถูกต้อง ซึ่งจะทำให้อินพุทที่ป้อนเข้าสู่ไมโครคอนโทรลเลอร์มีความถูกต้อง ซึ่งจะทำให้ไมโครคอนโทรลเลอร์สามารถทำงานได้ถูกต้องตามโปรแกรมที่ได้ทำการเขียนไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์จากอาจารย์ที่ปรึกษาคือ รศ.ดร.วิวัฒน์ กิรานนท์ , ผศ.ดร.ปราโมทย์ วาดเขียน และ อาจารย์ วิชา แสงพิสิทธิ์ ที่คอยให้คำแนะนำแนวทางในการทำงานในยามที่มีปัญหา อีกทั้งยังคอยช่วยเหลือในด้านต่าง ๆ เป็นอย่างดี ขอขอบคุณกำลังใจและการสนับสนุนด้านค่าใช้จ่ายในการทำโครงการจากครอบครัววาศประสิทธิ์และครอบครัววัฒนานุกุลชัย และขอขอบคุณเป็นพิเศษสำหรับคุณสวริน รักบัว ที่สละเวลาให้คำปรึกษาทางโทรศัพท์เสมอไม่ว่าเวลาใด

ผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. เปรมจิต วิสุทธิศิริ, "สถาปัตยกรรมของจอแสดงผล", ซีเอ็ดยูเคชั่น, กรุงเทพฯ 2533
2. เปรมจิต วิสุทธิศิริ, "LED อินฟราเรด", ซีเอ็ดยูเคชั่น, กรุงเทพฯ 2534
3. "คู่มือการใช้งาน Jazz-31", ศิลาริเสิร์ท, กรุงเทพฯ
4. "อัตราค่าโทรศัพท์ทางไกล", องค์การโทรศัพท์แห่งประเทศไทย, กรุงเทพฯ 2540
5. PANKORE JI, "Optical process in semiconductor", Dover Publications Inc., New York, 1971
6. SZE SM, "Semiconductor device physics and technology", John Willey & Son Inc., New York, 1985



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS

SEMICONDUCTOR

DS1202, DS1202S

Serial Timekeeping Chip

FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 24 x 8 RAM for scratchpad data storage
- Serial I/O for minimum pin count
- 2.0–5.5 volt full operation
- Uses less than 300 nA at 2 volts
- Single-byte or multiple-byte (burst mode) data transfer for read or write of clock or RAM data
- 8-pin DIP or optional 16-pin SOIC for surface mount
- Simple 3-wire interface
- TTL-compatible ($V_{CC} = 5V$)
- Optional industrial temperature range $-40^{\circ}C$ to $+85^{\circ}C$ (IND)

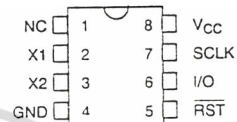
ORDERING INFORMATION

DS1202	8-pin DIP
DS1202S	16-pin SOIC
DS1202S-8	8-pin SOIC
DS1202N	8-pin DIP (IND)
DS1202SN	16-pin SOIC (IND)
DS1202SN-8	8-pin SOIC (IND)

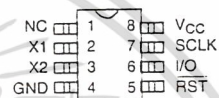
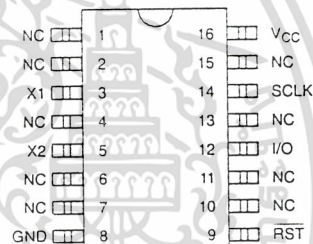
DESCRIPTION

The DS1202 Serial Timekeeping Chip contains a real time clock/calendar and 24 bytes of static RAM. It communicates with a microprocessor via a simple serial interface. The real time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Interfacing the

PIN ASSIGNMENT



8-PIN DIP

8-PIN SOIC
(208 mil)

16-PIN SOIC

PIN DESCRIPTION

NC	- No Connection
X1, X2	- 32.768 KHz Crystal Input
GND	- Ground
RST	- Reset
I/O	- Data Input/Output
SCLK	- Serial Clock
V_{CC}	- Power Supply Pin

DS1202 with a microprocessor is simplified by using synchronous serial communication. Only three wires are required to communicate with the clock/RAM: (1) RST (Reset), (2) I/O (Data line), and (3) SCLK (Serial clock). Data can be transferred to and from the clock/RAM one byte at a time or in a burst of up to 24 bytes. The DS1202 is designed to operate on very low power and retain data and clock information on less than 1 microwatt.

OPERATION

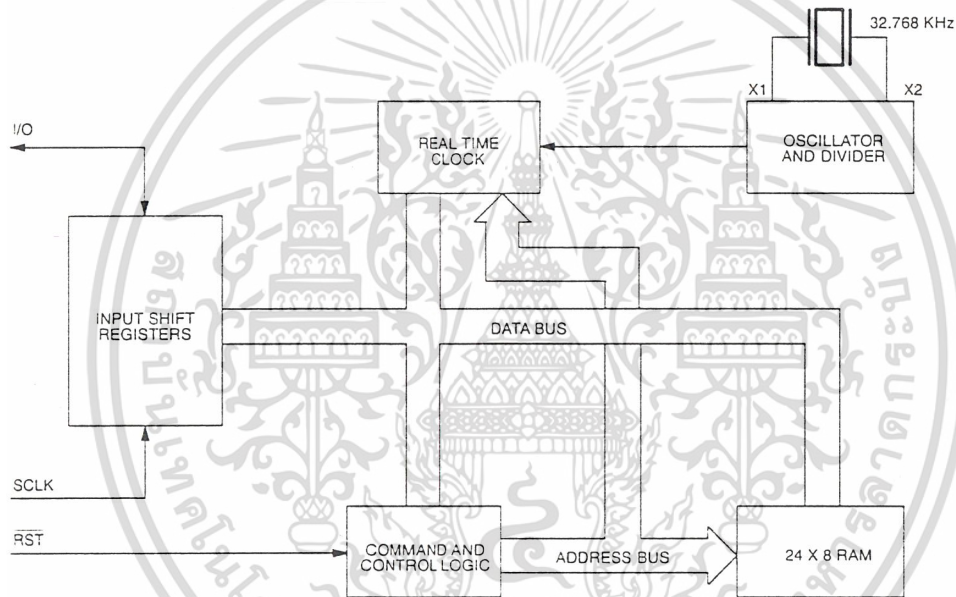
The main elements of the Serial Timekeeper are shown in Figure 1: shift register, control logic, oscillator, real time clock, and RAM. To initiate any transfer of data, \overline{RST} is taken high and eight bits are loaded into the shift register providing both address and command information. Data is serially input on the rising edge of the SCLK. The first eight bits specify which of 32 bytes will be accessed, whether a read or write cycle will take place, and whether a byte or burst mode transfer is to occur. After the first eight clock cycles have occurred which load the command word into the shift register, additional clocks will output data for a read or input data for a write.

The number of clock pulses equals eight plus eight for byte mode or eight plus up to 192 for burst mode.

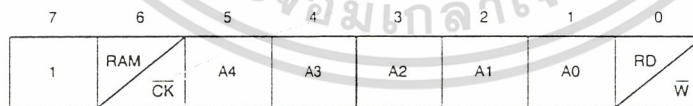
COMMAND BYTE

The command byte is shown in Figure 2. Each data transfer is initiated by a command byte. The MSB (Bit 7) must be a logic 1. If it is zero, further action will be terminated. Bit 6 specifies clock/calendar data if logic 0 or RAM data if logic 1. Bits one through five specify the designated registers to be input or output, and the LSB (Bit 0) specifies a write operation (input) if logic 0 or read operation (output) if logic 1. The command byte is always input starting with the LSB (bit 0).

DS1202 BLOCK DIAGRAM Figure 1



ADDRESS/COMMAND BYTE Figure 2



RESET AND CLOCK CONTROL

All data transfers are initiated by driving the $\overline{\text{RST}}$ input high. The $\overline{\text{RST}}$ input serves two functions. First, $\overline{\text{RST}}$ turns on the control logic which allows access to the shift register for the address/command sequence. Second, the $\overline{\text{RST}}$ signal provides a method of terminating either single byte or multiple byte data transfer. A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, data must be valid during the rising edge of the clock and data bits are output on the falling edge of clock. All data transfer terminates if the $\overline{\text{RST}}$ input is low and the I/O pin goes to a high impedance state. Data transfer is illustrated in Figure 3.

DATA INPUT

Following the eight SCLK cycles that input a write command byte, a data byte is input on the rising edge of the next eight SCLK cycles. Additional SCLK cycles are ignored should they inadvertently occur. Data is input starting with bit 0. Due to the inherent nature of the logic state machine, writing times containing an absolute value of "59" seconds should be avoided.

DATA OUTPUT

Following the eight SCLK cycles that input a read command byte, a data byte is output on the falling edge of the next eight SCLK cycles. Note that the first data bit to be transmitted occurs on the first falling edge after the last bit of the command byte is written. Additional SCLK cycles retransmit the data bytes should they inadvertently occur so long as $\overline{\text{RST}}$ remains high. This operation permits continuous burst mode read capability. Data is output starting with bit 0.

BURST MODE

Burst mode may be specified for either the clock/calendar or the RAM registers by addressing location 31 decimal (address/command bits one through five = logical one). As before, bit six specified clock or RAM and bit 0 specifies read or write. There is no data storage capacity at locations 8 through 31 in the Clock/Calendar Registers or locations 24 through 31 in the RAM registers. When writing to the clock registers in the burst mode, the first eight registers must be written in order for the data to be transferred.

However, when writing to RAM in burst mode it is not necessary to write all 24 bytes for the data to transfer.

Each byte that is written to will be transferred to RAM regardless of whether all 24 bytes are written or not.

CLOCK/CALENDAR

The clock/calendar is contained in eight write/read registers as shown in Figure 4. Data contained in the clock/calendar registers is in binary coded decimal format (BCD).

CLOCK HALT FLAG

Bit 7 of the seconds register is defined as the clock halt flag. When this bit is set to logic 1, the clock oscillator is stopped and the DS1202 is placed into a low-power standby mode with a current drain of not more than 100 nanoamps. When this bit is written to logic 0, the clock will start.

AM-PM/12-24 MODE

Bit 7 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

WRITE PROTECT BIT

Bit 7 of the control register is the write protect bit. The first seven bits (bits 0-6) are forced to zero and will always read a zero when read. Before any write operation to the clock or RAM, bit 7 must be zero. When high, the write protect bit prevents a write operation to any other register.

CLOCK/CALENDAR BURST MODE

The clock/calendar command byte specifies burst mode operation. In this mode the eight clock/calendar registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

RAM

The static RAM is 24 x 8 bytes addressed consecutively in the RAM address space.

RAM BURST MODE

The RAM command byte specifies burst mode operation. In this mode, the 24 RAM registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

REGISTER SUMMARY

A register data format summary is shown in Figure 4.

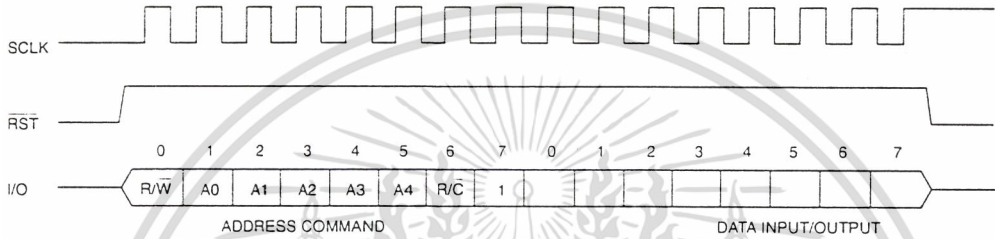
CRYSTAL SELECTION

A 32.768 KHz crystal, can be directly connected to the DS1202 via pins 2 and 3 (X1, X2). The crystal selected for use should have a specified load capacitance (CL) of 6 pF. The crystal is connected directly to the X1 and X2

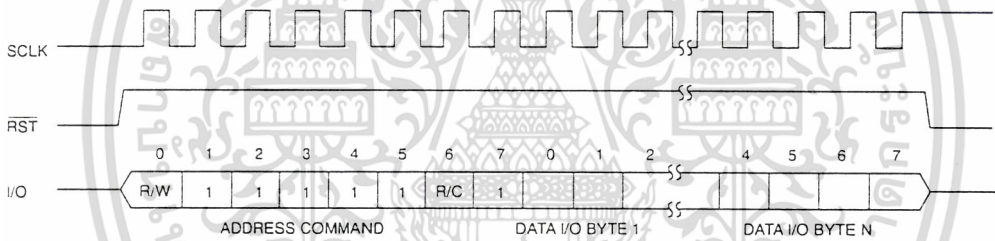
pins. There is no need for external capacitors or resistors. Note: X1 and X2 are very high impedance nodes. It is recommended that they and the crystal be guard-ringed with ground and that high frequency signals be kept away from the crystal area. For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks".

DATA TRANSFER SUMMARY Figure 3

SINGLE BYTE TRANSFER



BURST MODE TRANSFER



FUNCTION	BYTE N	SCLK n
CLOCK	8	72
RAM	24	200

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground
 Operating Temperature
 Storage Temperature
 Soldering Temperature

-0.3V to +7.0V
 0°C to 70°C
 -55°C to +125°C
 260°C for 10 seconds

- * This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

RECOMMENDED DC OPERATING CONDITIONS

(0°C to 70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	2.0		5.5	V	1
Logic 1 Input	V_{IH}	2.0		$V_{CC}+0.3$	V	1
Logic 0 Input	V_{IL}	$V_{CC}=2.0V$	-0.3	+0.3	V	1
		$V_{CC}=5V$	-0.3	+0.8		

DC ELECTRICAL CHARACTERISTICS(0°C to 70°C; $V_{CC} = 2.0$ to 5.5V*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage	I_{LI}			+500	μA	6
I/O Leakage	I_{LO}			+500	μA	6
Logic 1 Output	V_{OH}	$V_{CC}=2V$	1.6		V	2
		$V_{CC}=5V$	2.4			
Logic 0 Output	V_{OL}	$V_{CC}=2V$		0.4	V	3
		$V_{CC}=5V$		0.4		
Active Supply Current	I_{CC}	$V_{CC}=2V$		0.4	mA	5
		$V_{CC}=5V$		1.2		
Timekeeping Current	I_{CC1}	$V_{CC}=2V$		0.3	μA	4
		$V_{CC}=5V$		1		
Leakage Current	I_{CC2}	$V_{CC}=2V$		100	nA	10
		$V_{CC}=5V$		100		

*Unless otherwise noted.

CAPACITANCE(t_A = 25°C)

PARAMETER	SYMBOL	CONDITION	TYP	MAX	UNITS	NOTES
Input Capacitance	C_I		5		pF	
I/O Capacitance	$C_{I/O}$		10		pF	
Crystal Capacitance	C_X		6		pF	

AC ELECTRICAL CHARACTERISTICS

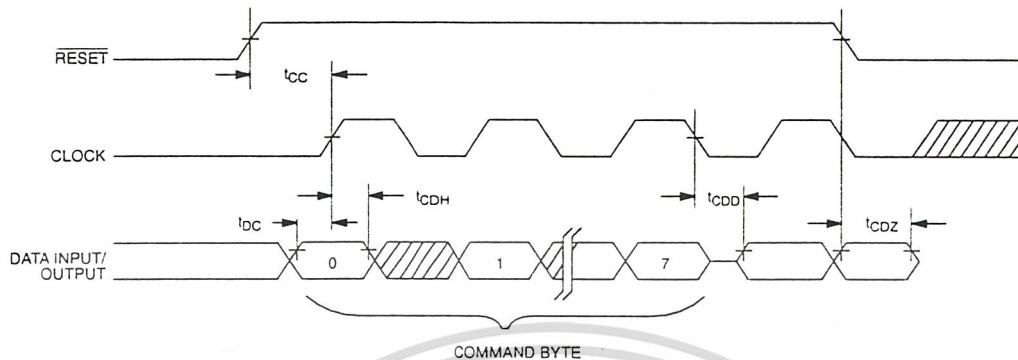
(0°C to 70°C; $V_{CC} = 2.0$ to 5.5V*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Data to CLK Setup	t_{DC}	$V_{CC}=2V$	200			ns 7
		$V_{CC}=5V$	50			
CLK to Data Hold	t_{CDH}	$V_{CC}=2V$	280			ns 7
		$V_{CC}=5V$	70			
CLK to Data Delay	t_{CDD}	$V_{CC}=2V$		800	ns	7, 8, 9
		$V_{CC}=5V$		200		
CLK Low Time	t_{CL}	$V_{CC}=2V$	1000		ns	7
		$V_{CC}=5V$	250			
CLK High Time	t_{CH}	$V_{CC}=2V$	1000		ns	7, 12
		$V_{CC}=5V$	250			
CLK Frequency	f_{CLK}	$V_{CC}=2V$		0.5	MHz	7, 12
		$V_{CC}=5V$	DC	2.0		
CLK Rise and Fall	t_R, t_F	$V_{CC}=2V$		2000	ns	
		$V_{CC}=5V$		500		
\overline{RST} to CLK Setup	t_{CC}	$V_{CC}=2V$	4		μs	7
		$V_{CC}=5V$	1			
CLK to \overline{RST} Hold	t_{CCH}	$V_{CC}=2V$	1000		ns	7
		$V_{CC}=5V$	250			
\overline{RST} Inactive Time	t_{CWH}	$V_{CC}=2V$	4		μs	7
		$V_{CC}=5V$	1			
\overline{RST} to I/O High Z	t_{CDZ}	$V_{CC}=2V$		280	ns	7
		$V_{CC}=5V$		70		

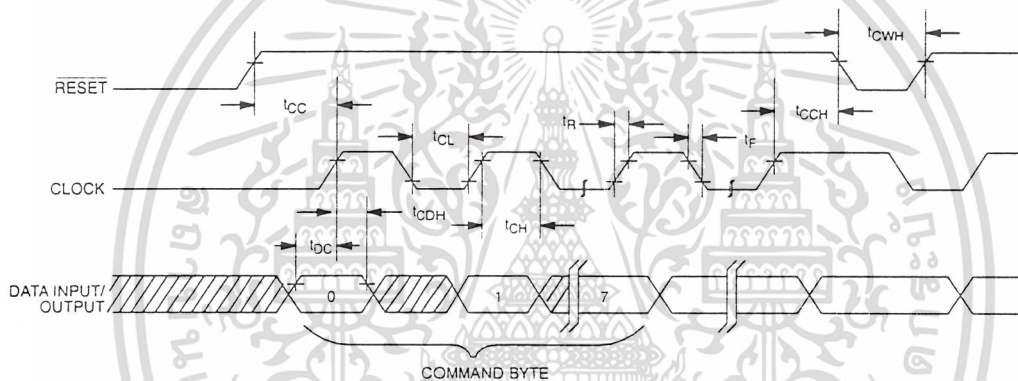
*Unless otherwise noted.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING DIAGRAM: READ DATA TRANSFER Figure 5



TIMING DIAGRAM: WRITE DATA TRANSFER Figure 6

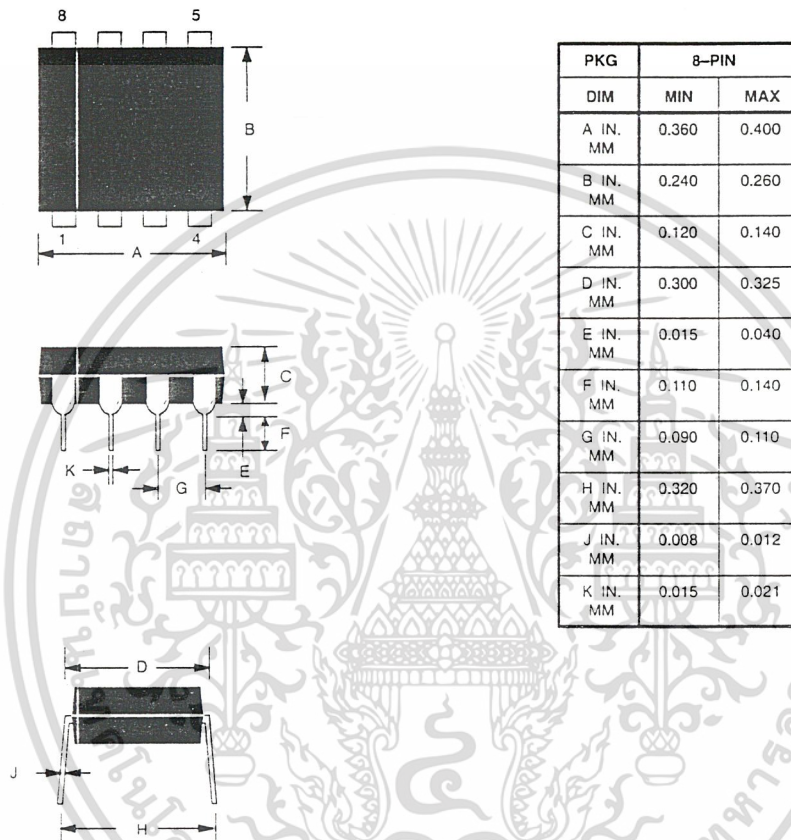


NOTES:

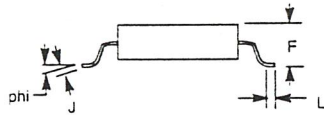
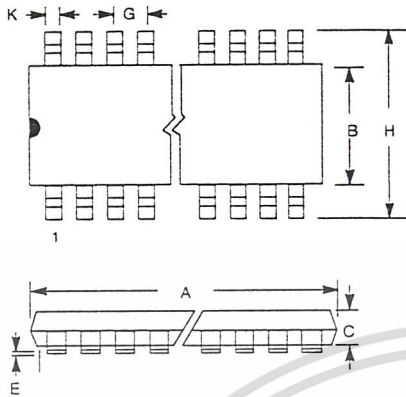
1. All voltages are referenced to ground.
2. Logic one voltages are specified at a source current of 1 mA at $V_{CC}=5V$ and 0.4 mA at $V_{CC}=2V$, $V_{OH}=V_{CC}$ for capacitive loads.
3. Logic zero voltages are specified at a sink current of 4 mA at $V_{CC}=5V$ and 1.5 mA at $V_{CC}=2V$.
4. t_{CC1} is specified with I/O open, \overline{RST} set to a logic 0, and clock halt flag=0 (oscillator enabled).
5. t_{CC} is specified with the I/O pin open, \overline{RST} high, $SCLK=2\text{ MHz}$ at $V_{CC}=5V$; $SCLK=500\text{ KHz}$, $V_{CC}=2V$ and clock halt flag=0 (oscillator enabled).
6. \overline{RST} , $SCLK$, and I/O all have 40K Ω pull-down resistors to ground.
7. Measured at $V_{IH}=2.0V$ or $V_{IL}=0.8V$ and 10 ms maximum rise and fall time.
8. Measured at $V_{OH}=2.4V$ or $V_{OL}=0.4V$.
9. Load capacitance = 50 pF.

10. I_{CC2} is specified with \overline{RST} , I/O, and SCLK open. The clock halt flag must be set to logic one (oscillator disabled).
11. At power-up, \overline{RST} must be at a logic 0 until $V_{CC} \geq 2$ volts. Also, SCLK must be at a logic 0 when \overline{RST} is driven to a logic one state.
12. If t_{CH} exceeds 100 ms with \overline{RST} in a logic one state, then I_{CC} may briefly exceed I_{CC} specification.

DS1202 SERIAL TIMEKEEPER 8-PIN DIP



DS1202S SERIAL TIMEKEEPER 16-PIN SOIC



PKG	16-PIN	
DIM	MIN	MAX
A IN.	0.500	0.511
MM	12.70	12.99
B IN.	0.290	0.300
MM	7.37	7.65
C IN.	0.089	0.095
MM	2.26	2.41
E IN.	0.004	0.012
MM	0.102	0.30
F IN.	0.094	0.105
MM	2.38	2.68
G IN.	0.050 BSC	
MM	1.27 BSC	
H IN.	0.398	0.416
MM	10.11	10.57
J IN.	0.009	0.013
MM	0.229	0.33
K IN.	0.013	0.019
MM	0.33	0.48
L IN.	0.016	0.040
MM	0.406	1.20
phi	0°	8°

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1202S8 8-PIN SOIC 200 MIL



PKG	8-PIN	
	DIM	MIN
A IN.	0.203	0.213
MM	5.16	5.41
B IN.	0.203	0.213
MM	5.16	5.41
C IN.	0.070	0.074
MM	1.78	1.88
E IN.	0.004	0.010
MM	0.102	0.390
F IN.	0.074	0.84
MM	1.88	2.13
G IN.	0.050 BSC	
MM	1.27 BSC	
H IN.	0.302	0.318
MM	7.67	8.07
J IN.	0.006	0.010
MM	0.152	0.254
K IN.	0.013	0.020
MM	0.33	0.508
L IN.	0.19	0.030
MM	4.83	0.762

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;PROGRAM CHECKCODE

```
PORTA EQU 0F800H ;8255 NO.1
PORTB EQU 0F801H
PORTC EQU 0F802H
PORTD EQU 0F803H
PORT_C EQU 0FC02H ;8255 NO.2
PORT_D EQU 0FC03H
LCDWRC EQU 0FA00H ;LCD WRITE CONTROL
LCDRDC EQU 0FA01H ;LCD READ CONTROL (BUSY CHECK)
LCDWRD EQU 0FA02H ;LCD WRITE DATA
LCDRDD EQU 0FA03H ;LCD READ DATA
RTCDAT EQU P1.4 ;SET PORT RTC
RTCCLK EQU P1.5
RTCRST EQU P1.6
```

ORG 0000H

DS 8 ;BANK-0 REGISTER

```
TIMBUF: DS 8 ;TIME BUFFER
FIRSTTM: DS 2 ;FIRST TIME RECORD
LASTTM: DS 2 ;LAST TIME RECORD
HRESUL: DS 1 ;HOUR RESULT
MRESUL: DS 1 ;MIN RESULT
CODE: DS 2 ;USER CODE
BAHTBUF: DS 1
NUMBER: DS 9
CHARGE: DS 2
BUFFER: DS 2
```

ORG 0100H

```
STTWORK: MOV R2, #100 ;FOR FIRST POWER UP
DELA1: MOV R3, #0
DJNZ R3, $
DJNZ R2, DELA1
STTPROG: MOV R2, #100 ;FOR AGAIN BEGIN OPERATION
DELY1: MOV R3, #0
DJNZ R3, $
DJNZ R2, DELY1
```

MOV DPTR, #PORT_D ;SET 8255 NO.2

MOV A, #93H

MOVX @DPTR, A

MOV DPTR, #PORTD

MOV A, #9BH ;SET 8255 NO.1 INPUT

MOVX @DPTR, A

MOV DPTR, #PORT_C ;INPUT RELAY = HI

SETB ACC.7

MOVX @DPTR, A

LCALL LCDINI

INSERT: MOV DPTR, #DATA1

LCALL LCDLD

MOV DPTR, #PORTA

MOVX A, @DPTR

JB ACC.7, INSERT ;BIT 7 OF PORT A IS INSERT CHECK

MOV DPTR, #YYYY

CALL LCDLD

MOV R2, #30

CALL DTSEC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LJMP    XXX

DATA1:  DB    "    *:WELCOME:*    "
        DB    "    INSERT CARD    "
YYYY:   DB    "    INSERT OK    "
        DB    "                    "
;MR. VICHAI    CODE 1111
;MR. NARONG    CODE 5555
;MR. MEECHAI   CODE 7777
;MRS. JUNYA    CODE 3333

XXX:    MOV    DPTR, #PORTB
        MOVX   A, @DPTR
        MOV    CODE, A                ;SAVE CODE 1 BYTE
        PUSH  ACC
        MOV    A, #1
        CALL  LCDWI
        POP   ACC
        PUSH  ACC
        CALL  LCDDPAA
        MOV    R2, #10
        CALL  DTSEC
        POP   ACC
        MOV    R1, A                ;SECOND 8 BITS IN PORTB
        MOV    DPTR, #PORTA
        MOVX   A, @DPTR            ;FIRST 8 BITS IN PORTA
        MOV    CODE+1, A          ;SAVE CODE 1 BYTE
        PUSH  ACC
        MOV    A, #1
        CALL  LCDWI
        POP   ACC
        PUSH  ACC
        CALL  LCDDPAA
        MOV    R2, #10
        CALL  DTSEC
        POP   ACC

CHECK1:  CJNE  A, #55H, CHECK2
        JMP   CHECK11
CHECK2:  CJNE  A, #77H, CHECK3
        JMP   CHECK22
CHECK3:  CJNE  A, #11H, CHECK4
        JMP   CHECK33
CHECK4:  CJNE  A, #33H, ERROR
        JMP   CHECK44

CHECK11: MOV    A, R1
        CJNE  A, #55H, ERROR
        JMP   CORRECT
CHECK22: MOV    A, R1
        CJNE  A, #77H, ERROR
        JMP   CORRECT
CHECK33: MOV    A, R1
        CJNE  A, #11H, ERROR
        JMP   CORRECT
CHECK44: MOV    A, R1
        CJNE  A, #33H, ERROR
        JMP   CORRECT

CORRECT: MOV    DPTR, #DATA2
        LCALL LCDLD
        MOV    DPTR, #PORT_D

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      A, #93H          ;8255 NO.2 port c (bit 4-7) OUTPUT
MOVX     @DPTR, A
MOV      DPTR, #PORT_C   ;BIT 7 OF PORTC =0 SHOW CORRECT CODE
CLR      ACC.7
MOVX     @DPTR, A
MOV      R2, #20
CALL    DTSEC
LJMP     AAA
DATA2:   DB      "      PRESS      "
         DB      " TELEPHONE N.O. "
ERROR:   MOV      DPTR, #DATA3
         LCALL   LCDLD
         MOV      R2, #30
         CALL    DTSEC
         LJMP     LOOP
DATA3:   DB      " CARD ERROR !!! "
         DB      "                "
;
LCALL   INITIAL          ;SET TIME
AAA:    LCALL   LCDINI    ;SET LCD
START:  LCALL   MAIN     ;SHOWTIME
MOV      DPTR, #PORTD
MOV      A, #9BH         ;SET 8255 INPUT
MOVX     @DPTR, A
MOV      DPTR, #PORTC   ;BIT 6 OF PORTC =1 IS RING/BUSY
MOVX     A, @DPTR
JNB      ACC.6, START    ; CHECK RING/BUSY =1
RING:   MOV      DPTR, #DATA6
         LCALL   LCDLD
         MOV      R2, #20
         LCALL   DTSEC
         JMP      RRR
DATA6:   DB      "      CHECK      "
         DB      "      SIGNAL    "
;*****CHECK RINGBACK / BUSY / RECEIVE *****
RRR:    MOV      R2, #07
         LCALL   DTSEC
KK:     MOV      DPTR, #PORTC
         MOVX     A, @DPTR
         JNB      ACC.6, HOOKCH ;0.7
JJ:     MOV      R2, #25
         LCALL   DTSEC
         MOV      DPTR, #PORTC
         MOVX     A, @DPTR
         JB       ACC.6, FF      ;3.2
         MOV      R2, #25
         LCALL   DTSEC
         MOV      DPTR, #PORTC
         MOVX     A, @DPTR
         JNB      ACC.6, CHREC   ;5.7
         JMP      JJ
HOOKCH: MOV      DPTR, #PORTC
         MOVX     A, @DPTR
         JB       ACC.5, JJ      ;HOOK ON
         LJMP     EXIT          ;HOOK OFF
CHREC:  MOV      DPTR, #PORTC
         MOVX     A, @DPTR
         JB       ACC.5, WAIT
         LJMP     EXIT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FF:      MOV     DPTR,#DATA4
        LCALL  LCDLD
        LJMP  EXIT
DATA4:   DB     "   BUSY SIGNAL   "
        DB     "   !!!!!!!!!!!   "
WAIT:    MOV     DPTR,#DATA5
        LCALL  LCDLD
        MOV     R2,#15
        LCALL  DTSEC
        JMP     TTT
DATA5:   DB     "   TALKING   "
        DB     "   TIME     "
TTT:     MOV     DPTR,#PORTD           ;INITIAL PORT 8255
        MOV     A,#9BH
        MOVX   @DPTR,A
        LCALL  LCDINT                 ;INITIAL LCD
        MOV     NUMBER+0,#00H        ;CLEAR #NUMBER BUFFER
        MOV     NUMBER+1,#00H
        MOV     NUMBER+2,#00H
        MOV     NUMBER+3,#00H
        MOV     NUMBER+4,#00H
        MOV     NUMBER+5,#00H
        MOV     NUMBER+6,#00H
        MOV     NUMBER+7,#00H
        MOV     NUMBER+8,#00H
        MOV     A,#80H
        LCALL  LCDWI
        MOV     R1,#7
        MOV     R0,#NUMBER+0
NO1:     MOV     DPTR,#PORTC
        MOVX   A,@DPTR
        MOV     R2,#02
        LCALL  DTSEC
        JNB    ACC.4,NO1
        ANL    A,#0FH
RET:     NOP
        PUSH  ACC
        PUSH  00
        PUSH  02
        CALL  LCDDPB
        POP   02
        POP   00
        POP   ACC
REC1:    MOV     @R0,A
        JB     ACC.4,RET               ;BETWEEN PRESS STD=0
        INC   R0
        DJNZ  R1,NO1
        MOV   R2,#04
        LCALL DTSEC
        MOV   A,#NUMBER+0
        CJNE A,#00H,CASE               ;CALL TO IN-REGION 7 NUMBERS
        MOV   R1,#2                     ;CALL TO OUT-REGION 9 NUMBERS
ADDNO:   MOV     DPTR,#PORTC
        MOVX   A,@DPTR
        MOV   R2,#03
        LCALL DTSEC
        JNB   ACC.4,ADDNO
        ANL   A,#0FH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        PUSH ACC
        PUSH 00
        PUSH 02
        CALL LCDDPB
        POP 02
        POP 00
        POP ACC
        MOV R0,#NUMBER+7
REC2:   MOV @R0,A
        JB ACC.4,RET ;BETWEEN PRESS STD=0
        INC R0
        DJNZ R1,ADDNO
        MOV R2,#05
        LCALL DTSEC
        JMP CASE ;GO TO CHECK COST
;===== CHECK COST FROM TELEPHONE NUMBER =====
CASE:   MOV A,NUMBER+0
        CJNE A,#00H,CASE1
        LJMP CASE2
CASE1:  MOV BAHTBUF,#3H ;CODE 02
        LJMP CALCULATE
CASE2:  MOV A,NUMBER+1
        CJNE A,#3H,BUFF
        LJMP CASE2_1
BUFF:   LJMP CASE3
;===== 03 =====
CASE2_1:MOV A,NUMBER+2 ;CODE 03_
        CJNE A,#2H,CASE2_2
        LJMP CHECK2_1 ;032
CASE2_2:MOV A,NUMBER+2
        CJNE A,#4H,CASE2_3
        LJMP CHECK2_2 ;034
CASE2_3:MOV A,NUMBER+2
        CJNE A,#5H,CASE2_4
        LJMP CHECK2_3 ;035
CASE2_4:MOV A,NUMBER+2
        CJNE A,#6H,CASE2_5
        LJMP CHECK2_4 ;036
CASE2_5:MOV A,NUMBER+2
        CJNE A,#7H,CASE2_6
        LJMP CHECK2_5 ;037
CASE2_6:MOV A,NUMBER+2
        CJNE A,#8H,CASE2_7
        LJMP CHECK2_6 ;038
CASE2_7:MOV A,NUMBER+2
        CJNE A,#9H,BUFF2
        LJMP CHECK2_7 ;039
BUFF2:  LJMP FALSE
;===== 032 =====
CHECK2_1:MOV A,NUMBER+3 ;CHECK SERVICE CHARGE
        CJNE A,#2H,CHK2_1_1
        MOV BAHTBUF,#6H
        LJMP CALCULATE ;032-2____(6)
CHK2_1_1:MOV A,NUMBER+3
        CJNE A,#3H,CHK2_1_2
        MOV BAHTBUF,#6H
        LJMP CALCULATE ;032-3____(6)
CHK2_1_2:MOV A,NUMBER+3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CJNE A, #4H, CHK2_1_3
MOV BAHTBUF, #6H
LJMP CALCULATE ;032-4____(6)
CHK2_1_3: MOV A, NUMBER+3
CJNE A, #5H, CHK2_1_4
MOV BAHTBUF, #0CH
LJMP CALCULATE ;032-5____(12)
CHK2_1_4: MOV A, NUMBER+3
CJNE A, #6H, BUFF3
MOV BAHTBUF, #0CH
LJMP CALCULATE ;032-6____(12)
BUFF3: LJMP FALSE
;===== 034 =====
CHECK2_2: MOV A, NUMBER+3 ;CHECK SERVICE CHARGE
CJNE A, #2H, CHK2_2_1
MOV BAHTBUF, #3H
LJMP CALCULATE ;034-2____(6)
CHK2_2_1: MOV A, NUMBER+3
CJNE A, #3H, CHK2_2_2
MOV BAHTBUF, #3H
LJMP CALCULATE ;034-3____(3)
CHK2_2_2: MOV A, NUMBER+3
CJNE A, #4H, CHK2_2_3
MOV BAHTBUF, #3H
LJMP CALCULATE ;034-4____(3)
CHK2_2_3: MOV A, NUMBER+3
CJNE A, #5H, CHK2_2_4
MOV BAHTBUF, #9H
LJMP CALCULATE ;034-5____(9)
CHK2_2_4: MOV A, NUMBER+3
CJNE A, #6H, CHK2_2_5
MOV BAHTBUF, #9H
LJMP CALCULATE ;034-6____(9)
CHK2_2_5: MOV A, NUMBER+3
CJNE A, #7H, BUFF4
MOV BAHTBUF, #6H
LJMP CALCULATE ;034-7____(6)
BUFF4: LJMP FALSE
;===== 035 =====
CHECK2_3: MOV A, NUMBER+3 ;CHECK SERVICE CHARGE
CJNE A, #2H, CHK2_3_1
MOV BAHTBUF, #6H
LJMP CALCULATE ;035-2____(6)
CHK2_3_1: MOV A, NUMBER+3
CJNE A, #3H, CHK2_3_2
MOV BAHTBUF, #6H
LJMP CALCULATE ;035-3____(6)
CHK2_3_2: MOV A, NUMBER+3
CJNE A, #5H, CHK2_3_3
MOV BAHTBUF, #6H
LJMP CALCULATE ;035-5____(6)
CHK2_3_3: MOV A, NUMBER+3
CJNE A, #6H, BUFF5
MOV BAHTBUF, #6H
LJMP CALCULATE ;035-6____(6)
BUFF5: LJMP FALSE
;===== 036 =====
CHECK2_4: MOV A, NUMBER+3 ;CHECK SERVICE CHARGE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                CJNE  A, #2H, CHK2_4_1
                MOV   BAHTBUF, #6H
                LJMP  CALCULATE                ;036-2_____ (6)
CHK2_4_1:      MOV   A, NUMBER+3
                CJNE  A, #3H, CHK2_4_2
                MOV   BAHTBUF, #6H
                LJMP  CALCULATE                ;036-3_____ (6)
CHK2_4_2:      MOV   A, NUMBER+3
                CJNE  A, #4H, CHK2_4_3
                MOV   BAHTBUF, #9H
                LJMP  CALCULATE                ;036-5_____ (9)
CHK2_4_3:      MOV   A, NUMBER+3
                CJNE  A, #5H, BUFF6
                MOV   BAHTBUF, #9H
                LJMP  CALCULATE                ;036-6_____ (9)
BUFF6:        LJMP  FALSE

;===== 037 =====
CHECK2_5:      MOV   A, NUMBER+3                ;CHECK SERVICE CHARGE
                CJNE  A, #2H, CHK2_5_1
                MOV   BAHTBUF, #6H
                LJMP  CALCULATE                ;037-2_____ (6)
CHK2_5_1:      MOV   A, NUMBER+3
                CJNE  A, #3H, BUFF7
                MOV   BAHTBUF, #6H
                LJMP  CALCULATE                ;037-3_____ (6)
BUFF7:        LJMP  FALSE

;===== 038 =====
CHECK2_6:      MOV   A, NUMBER+3                ;CHECK SERVICE CHARGE
                CJNE  A, #2H, CHK2_6_1
                MOV   BAHTBUF, #6H
                LJMP  CALCULATE                ;038-2_____ (6)
CHK2_6_1:      MOV   A, NUMBER+3
                CJNE  A, #3H, CHK2_6_2
                MOV   BAHTBUF, #6H
                LJMP  CALCULATE                ;038-3_____ (6)
CHK2_6_2:      MOV   A, NUMBER+3
                CJNE  A, #4H, CHK2_6_3
                MOV   BAHTBUF, #6H
                LJMP  CALCULATE                ;038-4_____ (6)
CHK2_6_3:      MOV   A, NUMBER+3
                CJNE  A, #5H, CHK2_6_4
                MOV   BAHTBUF, #6H
                LJMP  CALCULATE                ;038-5_____ (6)
CHK2_6_4:      MOV   A, NUMBER+3
                CJNE  A, #6H, BUFF8
                MOV   BAHTBUF, #9H
                LJMP  CALCULATE                ;038-6_____ (9)
BUFF8:        LJMP  FALSE

;===== 039 =====
CHECK2_7:      MOV   A, NUMBER+3
                CJNE  A, #3H, CHK2_7_1
                MOV   BAHTBUF, #0CH
                LJMP  CALCULATE                ;039-3_____ (12)
CHK2_7_1:      MOV   A, NUMBER+3
                CJNE  A, #4H, CHK2_7_2
                MOV   BAHTBUF, #0CH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                LJMP  CALCULATE                ;039-4_____ (12)
CHK2_7_2: MOV    A,NUMBER+3
                CJNE  A,#5H,BUFF9
                MOV    BAHTBUF,#0CH
                LJMP  CALCULATE                ;039-5_____ (12)
BUFF9:        LJMP  FALSE
;===== 04_ =====

CASE3:        MOV    A,NUMBER+1
                CJNE  A,#4H,BUF1
                LJMP  CASE3_1                ;CODE 04_
BUF1:         LJMP  CASE4
CASE3_1:      MOV    A,NUMBER+2
                CJNE  A,#2H,CASE3_2
                LJMP  CHECK3_1                ;042
CASE3_2:      MOV    A,NUMBER+2
                CJNE  A,#3H,CASE3_3
                LJMP  CHECK3_2                ;043
CASE3_3:      MOV    A,NUMBER+2
                CJNE  A,#4H,CASE3_4
                LJMP  CHECK3_3                ;044
CASE3_4:      MOV    A,NUMBER+2
                CJNE  A,#5H,BUFF10
                LJMP  CHECK3_4                ;045
BUFF10:       LJMP  FALSE
;===== 042 =====
CHECK3_1:     MOV    A,NUMBER+3
                CJNE  A,#2H,CHK3_1_1
                MOV    BAHTBUF,#0FH
                LJMP  CALCULATE                ;042-2 (15)
CHK3_1_1:    MOV    A,NUMBER+3
                CJNE  A,#3H,CHK3_1_2
                MOV    BAHTBUF,#0FH
                LJMP  CALCULATE                ;042-3 (15)
CHK3_1_2:    MOV    A,NUMBER+3
                CJNE  A,#4H,CHK3_1_3
                MOV    BAHTBUF,#13H
                LJMP  CALCULATE                ;042-4 (18)
CHK3_1_3:    MOV    A,NUMBER+3
                CJNE  A,#5H,CHK3_1_4
                MOV    BAHTBUF,#13H
                LJMP  CALCULATE                ;042-5 (18)
CHK3_1_4:    MOV    A,NUMBER+3
                CJNE  A,#6H,CHK3_1_5
                MOV    BAHTBUF,#13H
                LJMP  CALCULATE                ;042-6 (18)
CHK3_1_5:    MOV    A,NUMBER+3
                CJNE  A,#7H,CHK3_1_6
                MOV    BAHTBUF,#13H
                LJMP  CALCULATE                ;042-7 (18)
CHK3_1_6:    MOV    A,NUMBER+3
                CJNE  A,#8H,BUFF11
                MOV    BAHTBUF,#0FH
                LJMP  CALCULATE                ;042-8 (18)
BUFF11:      LJMP  FALSE
;===== 043 =====
CHECK3_2:     MOV    A,NUMBER+3
                CJNE  A,#2H,CHK3_2_1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    BAHTBUF,#0FH
LJMP   CALCULATE           ;043-2(15)
CHK3_2_1: MOV    A,NUMBER+3
        CJNE   A,#3H,CHK3_2_2
        MOV    BAHTBUF,#0FH
        LJMP   CALCULATE           ;043-3(15)
CHK3_2_2: MOV    A,NUMBER+3
        CJNE   A,#4H,CHK3_2_3
        MOV    BAHTBUF,#0FH
        LJMP   CALCULATE           ;043-4(15)
CHK3_2_3: MOV    A,NUMBER+3
        CJNE   A,#5H,CHK3_2_4
        MOV    BAHTBUF,#0FH
        LJMP   CALCULATE           ;043-5(15)
CHK3_2_4: MOV    A,NUMBER+3
        CJNE   A,#7H,CHK3_2_5
        MOV    BAHTBUF,#0FH
        LJMP   CALCULATE           ;043-7(15)
CHK3_2_5: MOV    A,NUMBER+3
        CJNE   A,#8H,BUFF12
        MOV    BAHTBUF,#0FH
        LJMP   CALCULATE           ;043-8(15)
BUFF12: LJMP   FALSE

;===== 044 =====
CHECK3_3: MOV    A,NUMBER+3
        CJNE   A,#2H,CHK3_3_1
        MOV    BAHTBUF,#0CH
        LJMP   CALCULATE           ;044-2(12)
CHK3_3_1: MOV    A,NUMBER+3
        CJNE   A,#3H,CHK3_3_2
        MOV    BAHTBUF,#0CH
        LJMP   CALCULATE           ;044-3(12)
CHK3_3_2: MOV    A,NUMBER+3
        CJNE   A,#4H,CHK3_3_3
        MOV    BAHTBUF,#0CH
        LJMP   CALCULATE           ;044-4(12)
CHK3_3_3: MOV    A,NUMBER+3
        CJNE   A,#5H,CHK3_3_4
        MOV    BAHTBUF,#0CH
        LJMP   CALCULATE           ;044-5(12)
CHK3_3_4: MOV    A,NUMBER+3
        CJNE   A,#6H,CHK3_3_5
        MOV    BAHTBUF,#0CH
        LJMP   CALCULATE           ;044-6(12)
CHK3_3_5: MOV    A,NUMBER+3
        CJNE   A,#8H,BUFF13
        MOV    BAHTBUF,#0CH
        LJMP   CALCULATE           ;044-8(12)
BUFF13: LJMP   FALSE

;===== 045 =====
CHECK3_4: MOV    A,NUMBER+3
        CJNE   A,#2H,CHK3_4_1
        MOV    BAHTBUF,#0FH
        LJMP   CALCULATE           ;045-2(15)
CHK3_4_1: MOV    A,NUMBER+3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CJNE  A, #3H, CHK3_4_2
        MOV   BAHTBUF, #0FH
        LJMP  CALCULATE                ;045-3 (15)
CHK3_4_2: MOV   A, NUMBER+3
        CJNE  A, #4H, CHK3_4_3
        MOV   BAHTBUF, #0FH
        LJMP  CALCULATE                ;045-4 (15)
CHK3_4_3: MOV   A, NUMBER+3
        CJNE  A, #6H, CHK3_4_4
        MOV   BAHTBUF, #0FH
        LJMP  CALCULATE                ;045-6 (15)
CHK3_4_4: MOV   A, NUMBER+3
        CJNE  A, #7H, BUFF14
        MOV   BAHTBUF, #0FH
        LJMP  CALCULATE                ;045-7 (15)
BUFF14:  LJMP  FALSE

```

```

;=====
CASE4:   MOV   A, NUMBER+1
        CJNE  A, #5H, BUF2
        LJMP  CASE4_1                ;CODE 05 _
BUF2:    LJMP  CASE5
CASE4_1: MOV   A, NUMBER+2
        CJNE  A, #3H, CASE4_2
        LJMP  CHECK4_1                ;053
CASE4_2: MOV   A, NUMBER+2
        CJNE  A, #4H, CASE4_3
        LJMP  CHECK4_2                ;054
CASE4_3: MOV   A, NUMBER+2
        CJNE  A, #5H, CASE4_4
        LJMP  CHECK4_3                ;055
CASE4_4: MOV   A, NUMBER+2
        CJNE  A, #6H, BUFF15
        LJMP  CHECK4_4                ;056
BUFF15:  LJMP  FALSE

```

```

;===== 053 =====
CHECK4_1: MOV   A, NUMBER+3
        CJNE  A, #2H, CHK4_1_1
        MOV   BAHTBUF, #13H
        LJMP  CALCULATE                ;053-2 (18)
CHK4_1_1: MOV   A, NUMBER+3
        CJNE  A, #3H, CHK4_1_2
        MOV   BAHTBUF, #13H
        LJMP  CALCULATE                ;053-3 (18)
CHK4_1_2: MOV   A, NUMBER+3
        CJNE  A, #5H, CHK4_1_3
        MOV   BAHTBUF, #13H
        LJMP  CALCULATE                ;053-5 (18)
CHK4_1_3: MOV   A, NUMBER+3
        CJNE  A, #6H, CHK4_1_4
        MOV   BAHTBUF, #13H
        LJMP  CALCULATE                ;053-6 (18)
CHK4_1_4: MOV   A, NUMBER+3
        CJNE  A, #7H, BUFF16
        MOV   BAHTBUF, #13H
        LJMP  CALCULATE                ;053-7 (18)
BUFF16:  LJMP  FALSE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;===== 054 =====
CHECK4_2: MOV    A,NUMBER+3
          CJNE   A,#2H,CHK4_2_1
          MOV    BAHTBUF,#13H
          LJMP   CALCULATE          ;054-2(18)
CHK4_2_1: MOV    A,NUMBER+3
          CJNE   A,#4H,CHK4_2_2
          MOV    BAHTBUF,#13H
          LJMP   CALCULATE          ;054-4(18)
CHK4_2_2: MOV    A,NUMBER+3
          CJNE   A,#5H,CHK4_2_3
          MOV    BAHTBUF,#0FH
          LJMP   CALCULATE          ;054-5(15)
CHK4_2_3: MOV    A,NUMBER+3
          CJNE   A,#6H,CHK4_2_4
          MOV    BAHTBUF,#0FH
          LJMP   CALCULATE          ;054-6(15)
CHK4_2_4: MOV    A,NUMBER+3
          CJNE   A,#7H,BUFF17
          MOV    BAHTBUF,#13H
          LJMP   CALCULATE          ;054-7(18)
BUFF17:  LJMP   FALSE

;===== 055 =====
CHECK4_3: MOV    A,NUMBER+3
          CJNE   A,#2H,CHK4_3_1
          MOV    BAHTBUF,#0CH
          LJMP   CALCULATE          ;055-2(12)
CHK4_3_1: MOV    A,NUMBER+3
          CJNE   A,#3H,CHK4_3_2
          MOV    BAHTBUF,#0CH
          LJMP   CALCULATE          ;055-3(12)
CHK4_3_2: MOV    A,NUMBER+3
          CJNE   A,#4H,CHK4_3_3
          MOV    BAHTBUF,#0FH
          LJMP   CALCULATE          ;055-4(15)
CHK4_3_3: MOV    A,NUMBER+3
          CJNE   A,#5H,CHK4_3_4
          MOV    BAHTBUF,#0FH
          LJMP   CALCULATE          ;055-5(15)
CHK4_3_4: MOV    A,NUMBER+3
          CJNE   A,#6H,CHK4_3_5
          MOV    BAHTBUF,#0FH
          LJMP   CALCULATE          ;055-6(15)
CHK4_3_5: MOV    A,NUMBER+3
          CJNE   A,#7H,BUFF18
          MOV    BAHTBUF,#0CH
          LJMP   CALCULATE          ;055-7(12)
BUFF18:  LJMP   FALSE

;===== 056 =====
CHECK4_4: MOV    A,NUMBER+3
          CJNE   A,#2H,CHK4_4_1
          MOV    BAHTBUF,#0CH
          LJMP   CALCULATE          ;056-2(12)
CHK4_4_1: MOV    A,NUMBER+3
          CJNE   A,#3H,CHK4_4_2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    BAHTBUF,#0CH
LJMP   CALCULATE                ;056-3(12)
CHK4_4_2: MOV    A,NUMBER+3
        CJNE   A,#4H,CHK4_4_3
        MOV    BAHTBUF,#9H
        LJMP   CALCULATE                ;056-4(9)
CHK4_4_3: MOV    A,NUMBER+3
        CJNE   A,#5H,CHK4_4_4
        MOV    BAHTBUF,#9H
        LJMP   CALCULATE                ;056-5(9)
CHK4_4_4: MOV    A,NUMBER+3
        CJNE   A,#6H,CHK4_4_5
        MOV    BAHTBUF,#0CH
        LJMP   CALCULATE                ;056-6(9)
CHK4_4_5: MOV    A,NUMBER+3
        CJNE   A,#7H,BUFF19
        MOV    BAHTBUF,#0CH
        LJMP   CALCULATE                ;056-7(9)
BUFF19: LJMP   FALSE

```

;=====

```

CASE5:  MOV    A,NUMBER+1
        CJNE   A,#7H,BUFF20
        LJMP   CASE5_1                ;CODE 07_
BUFF20: LJMP   FALSE
CASE5_1: MOV    A,NUMBER+2
        CJNE   A,#3H,CASE5_2
        LJMP   CHECK5_1                ;073
CASE5_2: MOV    A,NUMBER+2
        CJNE   A,#4H,CASE5_3
        LJMP   CHECK5_2                ;074
CASE5_3: MOV    A,NUMBER+2
        CJNE   A,#5H,CASE5_4
        LJMP   CHECK5_3                ;075
CASE5_4: MOV    A,NUMBER+2
        CJNE   A,#6H,CASE5_5
        LJMP   CHECK5_4                ;076
CASE5_5: MOV    A,NUMBER+2
        CJNE   A,#7H,BUFF21
        LJMP   CHECK5_5                ;077
BUFF21: LJMP   FALSE

```

;===== 073 =====

```

CHECK5_1: MOV    A,NUMBER+3
        CJNE   A,#2H,CHK5_1_1
        MOV    BAHTBUF,#13H
        LJMP   CALCULATE                ;073-2(18)
CHK5_1_1: MOV    A,NUMBER+3
        CJNE   A,#3H,CHK5_1_2
        MOV    BAHTBUF,#13H
        LJMP   CALCULATE                ;073-3(18)
CHK5_1_2: MOV    A,NUMBER+3
        CJNE   A,#4H,CHK5_1_3
        MOV    BAHTBUF,#13H
        LJMP   CALCULATE                ;073-4(18)
CHK5_1_3: MOV    A,NUMBER+3
        CJNE   A,#5H,BUFF22

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    BAHTBUF, #13H
LJMP   CALCULATE          ;073-5(18)
BUFF22: LJMP   FALSE

```

```

;===== 074 =====

```

```

CHECK5_2: MOV    A, NUMBER+3
        CJNE   A, #2H, CHK5_2_1
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;074-2(18)

```

```

CHK5_2_1: MOV    A, NUMBER+3
        CJNE   A, #3H, CHK5_2_2
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;074-3(18)

```

```

CHK5_2_2: MOV    A, NUMBER+3
        CJNE   A, #4H, CHK5_2_3
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;074-4(18)

```

```

CHK5_2_3: MOV    A, NUMBER+3
        CJNE   A, #5H, CHK5_2_4
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;074-5(18)

```

```

CHK5_2_4: MOV    A, NUMBER+3
        CJNE   A, #6H, CHK5_2_5
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;074-6(18)

```

```

CHK5_2_5: MOV    A, NUMBER+3
        CJNE   A, #7H, BUFF23
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;074-7(18)

```

```

BUFF23: LJMP   FALSE

```

```

;===== 075 =====

```

```

CHECK5_3: MOV    A, NUMBER+3
        CJNE   A, #2H, CHK5_3_1
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;075-2(18)

```

```

CHK5_3_1: MOV    A, NUMBER+3
        CJNE   A, #3H, CHK5_3_2
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;075-3(18)

```

```

CHK5_3_2: MOV    A, NUMBER+3
        CJNE   A, #4H, CHK5_3_3
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;075-4(18)

```

```

CHK5_3_3: MOV    A, NUMBER+3
        CJNE   A, #5H, CHK5_3_4
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;075-5(18)

```

```

CHK5_3_4: MOV    A, NUMBER+3
        CJNE   A, #6H, BUFF24
        MOV    BAHTBUF, #13H
        LJMP   CALCULATE          ;075-6(18)

```

```

BUFF24: LJMP   FALSE

```

```

;===== 076 =====

```

```

CHECK5_4: MOV    A, NUMBER+3
        CJNE   A, #2H, CHK5_4_1
        MOV    BAHTBUF, #13H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                LJMP  CALCULATE                ;076-2(18)
CHK5_4_1: MOV   A,NUMBER+3
                CJNE  A,#3H,CHK5_4_2
                MOV   BAHTBUF,#13H
                LJMP  CALCULATE                ;076-3(18)
CHK5_4_2: MOV   A,NUMBER+3
                CJNE  A,#4H,BUFF25
                MOV   BAHTBUF,#13H
                LJMP  CALCULATE                ;076-4(18)
BUFF25:  LJMP  FALSE

;===== 077 =====
CHECK5_5: MOV   A,NUMBER+3
                CJNE  A,#2H,CHK5_5_1
                MOV   BAHTBUF,#13H
                LJMP  CALCULATE                ;077-2(18)
CHK5_5_1: MOV   A,NUMBER+3
                CJNE  A,#3H,CHK5_5_2
                MOV   BAHTBUF,#13H
                LJMP  CALCULATE                ;077-3(18)
CHK5_5_2: MOV   A,NUMBER+3
                CJNE  A,#4H,CHK5_5_3
                MOV   BAHTBUF,#13H
                LJMP  CALCULATE                ;077-4(18)
CHK5_5_3: MOV   A,NUMBER+3
                CJNE  A,#5H,CHK5_5_4
                MOV   BAHTBUF,#0FH
                LJMP  CALCULATE                ;077-5(15)
CHK5_5_4: MOV   A,NUMBER+3
                CJNE  A,#8H,BUFF26
                MOV   BAHTBUF,#0FH
                LJMP  CALCULATE                ;077-8(15)
BUFF26:  LJMP  FALSE

CALCULATE: NOP
                LCALL USETIME                ;CALCULATE TIME
                LCALL SHOWTI                ;DISPLAY USETIME
                MOV   R2,#15
                LCALL DTSEC
                LCALL SHOWMO                ;DISPLAY TELEPHONE CHARGE
                MOV   R2,#15
                LCALL DTSEC

;PROGRAM FOR SEND DATA TO PC.
                MOV   SCON,#50H                ;INITIAL SERIAL PORT
                MOV   TMOD,#20H
                MOV   PCON,#00H
                MOV   TH1,#0E8H                ;1200 bps
                SETB TR1
STSEND: MOV   SBUF,CODE+0
                LCALL SEND
                MOV   SBUF,CODE+1
                LCALL SEND
                MOV   A,NUMBER
                SWAP A
                ADD  A,NUMBER+1
                MOV   SBUF,A                ;NUMBER+0 & NUMBER+1 044-258080
                LCALL SEND
                MOV   A,NUMBER+2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SWAP A
ADD A, NUMBER+3
MOV SBUF, A ;NUMBER+2 & NUMBER+3
LCALL SEND
MOV A, NUMBER+4
SWAP A
ADD A, NUMBER+5
MOV SBUF, A ;NUMBER+4 & NUMBER+5
LCALL SEND
MOV A, NUMBER+6
SWAP A
ADD A, NUMBER+7
MOV SBUF, A ;NUMBER+6 & NUMBER+7
LCALL SEND
MOV A, NUMBER+8
SWAP A
MOV SBUF, A ;NUMBER+8
LCALL SEND
MOV SBUF, FIRSTTM+0
LCALL SEND
MOV SBUF, FIRSTTM+1
LCALL SEND
MOV SBUF, LASTTM+0
LCALL SEND
MOV SBUF, LASTTM+1
LCALL SEND
MOV SBUF, HRESUL
LCALL SEND
MOV SBUF, MRESUL
LCALL SEND
MOV SBUF, CHARGE+0
LCALL SEND
MOV SBUF, CHARGE+1
LCALL SEND
JMP EXIT
FALSE: MOV DPTR, #FALSE1
LCALL LCDLD
MOV R2, #30
LCALL DTSEC
LJMP EXIT
FALSE1: DB " THIS CODE "
DB " IS FALSE "
EXIT: MOV DPTR, #PORT_C ;BIT 7 OF PORT_C = HIGH
SETB ACC.7
MOVX @DPTR, A
MOV R2, #20
LCALL DTSEC
CLR ACC.7
MOVX @DPTR, A ;BIT 7 OF PORT_C = LOW
MOV R2, #20 ;RELAY OPEN TEL. LINE AT LOW EDGE
LCALL DTSEC
LOOP: LJMP STTPROG ;BEGIN AGAIN
;***** LCDINI SUB *****
;INITAIL FOR LCD 1 LINE 16 CHAR
LCDINT: MOV A, #00111000B ;FUNCTION SET
LCALL LCDWI
MOV A, #00001110B ;DISPLAY ON, CURSOR OFF, NO BLINK
LCALL LCDWI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCDCLR:  MOV    A,#01H          ;CLEAR
         LCALL  LCDWI
         RET

;***** LCDDPA SUB *****
;DISPLAY DATA FROM A TO LCD
LCDDPB:  ACALL  HTOB
         MOV    A,R3
         ACALL  LCDWD
         RET

;***** HTOA.SUB *****
;CONVERT HEXADECIMAL TO ASCII CODE
;IN = A
;OUT = R2,R3
HTOB:    PUSH  ACC
         SWAP  A
         ACALL HTOBS
         MOV   R2,A
         POP   ACC
         ACALL HTOBS
         MOV   R3,A
         RET

HTOBS:   ANL   A,#0FH
         ORL   A,#30H
;*****FREE SBUF SUB*****
SEND:    MOV   SBUF,A
         JNB  TI,$
         CLR  TI
         RET

;*****DISPLAY TIME SUB*****
MAIN:    LCALL  LCDINI
         MOV   A,#80H          ;SET BUFFER LINE 1
         LCALL  LCDWI
         MOV   A,#'T'
         LCALL  LCDWD
         MOV   A,#'I'
         LCALL  LCDWD
         MOV   A,#'M'
         LCALL  LCDWD
         MOV   A,#'E'
         LCALL  LCDWD
         MOV   A,#':'
         LCALL  LCDWD
         LCALL  TIMERD
         MOV   A,TIMBUF+3      ;DISPLAY HOUR
         LCALL  LCDDPA
         MOV   A,#':'
         LCALL  LCDWD
         MOV   A,TIMBUF+4      ;DISPLAY MINUTE
         LCALL  LCDDPA
         MOV   A,#':'
         LCALL  LCDWD
         MOV   A,TIMBUF+5      ;DISPLAY SECONDS
         LCALL  LCDDPA
         MOV   A,#0COH        ;SET BUFFER LINE 2
         LCALL  LCDWI
         MOV   A,#'D'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL LCDWD
MOV A,#'A'
LCALL LCDWD
MOV A,#'T'
LCALL LCDWD
MOV A,#'E'
LCALL LCDWD
MOV A,#':'
LCALL LCDWD
MOV A,TIMBUF+0 ;DISPLAY DATE
LCALL LCDDPA
MOV A,#':'
LCALL LCDWD
MOV A,TIMBUF+1 ;DISPLAY MONTH
LCALL LCDDPA
MOV A,#':'
LCALL LCDWD
MOV A,TIMBUF+2 ;DISPLAY YEAR
LCALL LCDDPA
MOV R2,#30
LCALL DTSEC
RET
;*****
SHOWTI: LCALL LCDINI
MOV A,#80H ;SET BUFFER LINE 1
LCALL LCDWI
MOV A,#'T'
LCALL LCDWD
MOV A,#'O'
LCALL LCDWD
MOV A,#'T'
LCALL LCDWD
MOV A,#'A'
LCALL LCDWD
MOV A,#'L'
LCALL LCDWD
MOV A,#' '
LCALL LCDWD
MOV A,#'T'
LCALL LCDWD
MOV A,#'I'
LCALL LCDWD
MOV A,#'M'
LCALL LCDWD
MOV A,#'E'
LCALL LCDWD
MOV A,#':'
LCALL LCDWD
MOV A,#0C0H ;SET BUFFER LINE 2
LCALL LCDWI
MOV A,HRESUL ;DISPLAY HOUR RESULE TO USE PHONE
LCALL LCDDPA
MOV A,#':'
LCALL LCDWD
MOV A,MRESUL ;DISPLAY MINUTE RESULE TO USE PHONE
LCALL LCDDPA
MOV A,#' '
LCALL LCDWD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    A, #'H'
LCALL LCDWD
MOV    A, #'R'
LCALL LCDWD
MOV    A, #'S'
LCALL LCDWD
MOV    R2, #30
LCALL DTSEC
RET

;*****
SHOWMO: LCALL LCDINI
        MOV    A, #80H           ;SET BUFFER LINE 1
        LCALL LCDWI
        MOV    A, #'C'
        LCALL LCDWD
        MOV    A, #'H'
        LCALL LCDWD
        MOV    A, #'A'
        LCALL LCDWD
        MOV    A, #'R'
        LCALL LCDWD
        MOV    A, #'G'
        LCALL LCDWD
        MOV    A, #'E'
        LCALL LCDWD
        MOV    A, #' '
        LCALL LCDWD
        MOV    A, #'.'
        LCALL LCDWD
        MOV    A, CHARGE+1      ;DISPLAY CHARGE
        LCALL LCDDPA
        MOV    A, CHARGE+0     ;DISPLAY CHARGE
        LCALL LCDDPA
        MOV    A, #' '
        LCALL LCDWD
        MOV    A, #'B'
        LCALL LCDWD
        MOV    A, #'A'
        LCALL LCDWD
        MOV    A, #'H'
        LCALL LCDWD
        MOV    A, #'T'
        LCALL LCDWD
        MOV    A, #0C0H        ;SET BUFFER LINE
        LCALL LCDWI
        MOV    R2, #30
        LCALL DTSEC
        RET

;***** LCDINI SUB *****
;INITAIL FOR LCD 1 LINE 16 CHAR
LCDINI: MOV    A, #00111000B    ;FUNCTION SET
        LCALL LCDWI
        MOV    A, #00001100B   ;DISPLAY ON/OFF
        LCALL LCDWI
        MOV    A, #01H         ;CLEAR
        LCALL LCDWI
        RET

;***** LCDDPA SUB *****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;DISPLAY DATA FROM A TO LCD
LCDDPA: ACALL  HTOA
        MOV    A,R2
        ACALL  LCDWD
        MOV    A,R3
        ACALL  LCDWD
        RET

;***** HTOA.SUB *****
; CONVERT HEXADECIMAL TO ASCII CODE
; IN = A
; OUT = R2,R3
HTOA:  PUSH  ACC
        SWAP  A
        ACALL HTOAS
        MOV   R2,A
        POP   ACC
        ACALL HTOAS
        MOV   R3,A
        RET

HTOAS: ANL   A,#0FH
        ORL  A,#30H
        RET

; ***** LCDWI SUB *****
; LCD WRITE INSTRUCTION (RS=0)
; IN = A
; REG = A
LCDWI: PUSH  DPH
        PUSH  DPL
        MOV   DPTR,#LCDWRC
        MOVX  @DPTR,A
        ACALL LCDBY
        POP   DPL
        POP   DPH
        RET

; ***** LCDWD SUB *****
; LCD WRITE DATA (RS=1)
; IN = A
; REG = A
LCDWD: PUSH  DPH
        PUSH  DPL
        MOV   DPTR,#LCDWRD
        MOVX  @DPTR,A
        ACALL LCDBY
        POP   DPL
        POP   DPH
        RET

; ***** LCDBY SUB *****
; LCD WAIT FOR BUSY
LCDBY: PUSH  DPH
        PUSH  DPL
        MOV   DPTR,#LCDRDC
BUSY:  MOVX  A,@DPTR          ;WAIT FOR BF=0
        JB   ACC.7,BUSY
        POP  DPL
        POP  DPH
        RET

; ***** RTCRD SUB *****
; READ SINGLE BYTE FROM DS1202

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; IN = R2 COMMAND
; OUT = R3 DATA
; REG = A, R2, R3, R4, R5
RTCRD: CLR    RTCCLK                ;CLK=0
        CALL  RTCDL
        SETB  RTCRST                ;RST=1
        CALL  RTCDL
        MOV   A, R2                  ;COMMAND
        CALL  RTCWRC
        MOV   R4, #8                 ;READ DATA
        CLR   A
RTCRD1: CLR    RTCCLK
        CALL  RTCDL
        MOV   C, RTCDAT
        RRC   A
        SETB  RTCCLK
        CALL  RTCDL
        DJNZ  R4, RTCRD1
        MOV   R3, A
        CLR   RTCRST                ;RST=0
        CALL  RTCDL
        RET
; ***** TIMERD SUB *****
; READ RTC -> TIMBUF
; OUT = TIMBUF
; REG = A, R2, R3, R4, R5
TIMERD: MOV   R2, #87H                ;DATE
        CALL  RTCRD
        MOV   TIMBUF, R3
        MOV   R2, #89H                ;MONTH
        CALL  RTCRD
        MOV   TIMBUF+1, R3
        MOV   R2, #8DH                ;YEAR
        CALL  RTCRD
        MOV   TIMBUF+2, R3
        MOV   R2, #8BH                ;WEEK-DAY
        CALL  RTCRD
        MOV   TIMBUF+6, R3
        MOV   R2, #85H                ;HOUR
        CALL  RTCRD
        MOV   TIMBUF+3, R3
        MOV   R2, #83H                ;MINUTE
        CALL  RTCRD
        MOV   TIMBUF+4, R3
        MOV   R2, #81H                ;SECOND
        CALL  RTCRD
        MOV   TIMBUF+5, R3
        RET
; *****INITAIL TIME SUB *****
INITIAL: MOV   TIMBUF+0, #12H          ;SET DATE
        MOV   TIMBUF+1, #01H          ;SET MOUNTH
        MOV   TIMBUF+2, #98H          ;SET YEAR
        MOV   TIMBUF+3, #15H          ;SET HOUR
        MOV   TIMBUF+4, #48H          ;SET MINUTE
        MOV   TIMBUF+5, #00H          ;SET SECOND
        MOV   TIMBUF+6, #02H          ;SET WEEK-DAY
        CLR   RTCCLK
        CLR   RTCRST

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     R2,#0
DJNZ   R2,$
LCALL  TIMEWR
RET

; ***** TIMEWR SUB *****
; WRITE TIMBUF -> RTC
; IN = TIMBUF
; REG = A,R2,R3,R4,R5
TIMEWR: MOV     R2,#8EH           ;WRITE PROTECT = 0
MOV     R3,#0
CALL   RTCWR
MOV     R2,#86H           ;DATE
MOV     R3,TIMBUF
CALL   RTCWR
MOV     R2,#88H           ;MONTH
MOV     R3,TIMBUF+1
CALL   RTCWR
MOV     R2,#8CH           ;YEAR
MOV     R3,TIMBUF+2
CALL   RTCWR
MOV     R2,#8AH           ;WEEK-DAY
MOV     R3,TIMBUF+6
CALL   RTCWR
MOV     R2,#84H           ;HOUR
MOV     R3,TIMBUF+3
CALL   RTCWR
MOV     R2,#82H           ;MINUTE
MOV     R3,TIMBUF+4
CALL   RTCWR
MOV     R2,#80H           ;SECOND
MOV     R3,TIMBUF+5
CALL   RTCWR
MOV     R2,#8EH           ;WRITE PROTECT = 1
MOV     R3,#80H
CALL   RTCWR
RET

; ***** RTCWR SUB *****
; WRITE SINGLE BYTE TO DS1202
; IN = R2 COMMAND
; R3 DATA
; REG = A,R2,R3,R4,R5
RTCWR: CLR     RTCCLK           ;CLK=0
CALL   RTCDL
SETB   RTCRST           ;RST=1
CALL   RTCDL
MOV    A,R2           ;COMMAND
CALL   RTCWRC
MOV    A,R3           ;DATA
CALL   RTCWRC
CLR    RTCRST           ;RST=0
CALL   RTCDL
RET

RTCWRC: MOV     R4,#8           ;WRITE COMMAND/DATA
RTCWRC1: RRC     A
MOV     RTCDAT,C
SETB   RTCCLK           ;RISING EDGE CLOCK
CALL   RTCDL
CLR    RTCCLK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    RTCDL
DJNZ    R4,RTCWRC1
RET
RTCDDL: MOV    R5,#4          ;DELAY
DJNZ    R5,$
RET
;*****
LCDLDS: MOV    A,#80H
LCALL   LCDLDS
MOV     A,#0C0H
LCALL   LCDLDS
RET
LCDLDS: PUSH   DPH
PUSH   DPL
LCALL   LCDWI
POP    DPL
POP    DPH
MOV    R2,#16          ;16 CHARACTER
LCDLDS1: CLR    A
MOV    A,@A+DPTR
PUSH   DPH
PUSH   DPL
LCALL   LCDWD
POP    DPL
POP    DPH
INC    DPTR
DJNZ   R2,LCDLDS1
RET
; ***** DTSEC SUB *****
; DELAY 1/10 SECOND
; IN = R2
; REG = R2,R3,R4
DTSEC:  MOV    R3,#179
DTSEC1: MOV    R4,#0
DJNZ   R4,$
NOP
NOP
DJNZ   R3,DTSEC1
DJNZ   R2,DTSEC
RET
; ***** HTOA SUB *****
; CONVERT HEX TO ASCII
; IN = A
; OUT = R2,R3
; REG = A,R2,R3
HTOAA: PUSH   ACC
SWAP   A
CALL   HTOAAS
MOV    R2,A
POP    ACC
CALL   HTOAAS
MOV    R3,A
RET
HTOAAS: ANL   A,#0FH
CJNE   A,#0AH,$+3
JNC    HTOAAS1
ORL    A,#30H
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HTOAAS1: SUBB   A,#9
          ORL    A,#40H
          RET
LCDDPAA: CALL   HTOAA
          MOV    A,R2
          CALL   LCDWD
          MOV    A,R3
          CALL   LCDWD
          RET
USETIME: NOP
          CLR    RTCCLK
          CLR    RTCRST
          LCALL  TIMERD
          MOV    FIRSTTM+0,TIMBUF+3
          MOV    FIRSTTM+1,TIMBUF+4
          MOV    BUFFER+0,FIRSTTM+0
          MOV    BUFFER+1,FIRSTTM+1
          LCALL  FIND          ;FIND TELEPHONE CHARGE
          RET
; *****
SUBTIME: MOV    A, LASTTM+0
          CJNE   A, FIRSTTM+0, $+3
          JC     HH1          ;LASTTM < FIRSTTM
          MOV    A, LASTTM+0   ;LASTTM > FIRSTTM
          MOV    B, FIRSTTM+0
          SUBB   A,B
          CALL   SUBDA
          MOV    HRESUL,A
          SJMP   HH2
HH1:     MOV    A,#24H
          MOV    B, FIRSTTM+0
          CLR    C
          SUBB   A,B
          MOV    B, LASTTM+0
          ADD    A,B
          DA     A
          MOV    HRESUL,A
HH2:     MOV    A, LASTTM+1
          CJNE   A, FIRSTTM+1, $+3
          JC     MM1          ;LASTTM < FIRSTTM
          MOV    A, LASTTM+1
          MOV    B, FIRSTTM+1
          SUBB   A,B
          CALL   SUBDA
          MOV    MRESUL,A
          SJMP   CAL1
MM1:     MOV    A,#60H
          MOV    B, FIRSTTM+1
          CLR    C
          SUBB   A,B
          MOV    B, LASTTM+1
          ADD    A,B
          DA     A
          INC    A          ;MARGIN 1 MIN.
          MOV    MRESUL,A
CAL1:    NOP
          RET
; ***** SUBDA SUB *****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; BCD ADJUST AFTER SUBB
; IN  = A, CARRY, AUX-CARRY
; OUT = A, CARRY
; REG = A
SUBDA: JC      SUBDA2                ;SUBB BCD ADJUST
        JB      AC, SUBDA1
        RET                                ;C=0, AC=0
SUBDA1: ADD     A, #0FAH              ;C=0, AC=1
        CLR     C
        RET
SUBDA2: JB      AC, SUBDA3
        ADD     A, #0A0H              ;C=1, AC=0
        SETB    C
        RET
SUBDA3: ADD     A, #09AH              ;C=1, AC=1
        SETB    C
        RET

```

;*****FIND TELEPHONE CHARGE*****

```

FIND:  MOV     A, FIRSTTM+0          ;ORIGINAL HOUR
TE06:  CJNE   A, #06H, TE07
        LJMP   CALI
TE07:  CJNE   A, #07H, TE08
        LJMP   CALI
TE08:  CJNE   A, #08H, TE09
        LJMP   CALI
TE09:  CJNE   A, #09H, TE10
        LJMP   CALI
TE10:  CJNE   A, #10H, TE11
        LJMP   CALI
TE11:  CJNE   A, #11H, TE12
        LJMP   CALI
TE12:  CJNE   A, #12H, TE13
        LJMP   CALI
TE13:  CJNE   A, #13H, TE14
        LJMP   CALI
TE14:  CJNE   A, #14H, TE15
        LJMP   CALI
TE15:  CJNE   A, #15H, TE16
        LJMP   CALI
TE16:  CJNE   A, #16H, TE17
        LJMP   CALI
TE17:  CJNE   A, #17H, TE18
        LJMP   CALI
TE18:  CJNE   A, #18H, TE19
        LJMP   CALB
TE19:  CJNE   A, #19H, TE20
        LJMP   CALB
TE20:  CJNE   A, #20H, TE21
        LJMP   CALB
TE21:  CJNE   A, #21H, TE22
        LJMP   CALB
TE22:  CJNE   A, #22H, TE23
        LJMP   CALIII
TE23:  CJNE   A, #23H, TE24
        LJMP   CALIII
TE24:  CJNE   A, #24H, TE00
        LJMP   CALIII
TE00:  CJNE   A, #00H, TE01

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LJMP CALIII
TE01: CJNE A, #01H, TE02
LJMP CALIII
TE02: CJNE A, #02H, TE03
LJMP CALIII
TE03: CJNE A, #03H, TE04
LJMP CALIII
TE04: CJNE A, #04H, TE05
LJMP CALIII
TE05: NOP
LJMP CALIII
CALI: LCALL TIMERD
MOV A, TIMBUF+3
CJNE A, #18H, GOI
MOV LASTTM+0, #18H
MOV LASTTM+1, #00H
LCALL SUBTIME
CHARGEI: MOV A, HRESUL
MOV B, #60
MUL AB
ADD A, MRESUL ;TALK NOT MORE THAN 255 MIN.
MOV B, BAHTBUF
MUL AB
MOV CHARGE+0, A ;MONEY LOW BYTE
MOV CHARGE+1, B ;MONEY HIGH BYTE
MOV A, BAHTBUF
ST0: CJNE A, #06H, ST1
MOV BAHTBUF, #04H
JMP TOTALI
ST1: CJNE A, #09H, ST2
MOV BAHTBUF, #06H
JMP TOTALI
ST2: CJNE A, #12H, ST3
MOV BAHTBUF, #08H
JMP TOTALI
ST3: CJNE A, #15H, ST4
MOV BAHTBUF, #10H
JMP TOTALI
ST4: MOV BAHTBUF, #12H
TOTALI: MOV FIRSTTM+0, #18H
MOV FIRSTTM+1, #00H
LCALL GOI
GOI: MOV DPTR, #PORTC
MOVX A, @DPTR
JB ACC. 5, CALI
LCALL TIMERD
MOV LASTTM+0, TIMBUF+3
MOV LASTTM+1, TIMBUF+4
LCALL SUBTIME
LCALL FEE ;FIND TOTAL MONEY
NETTIM: MOV FIRSTTM+0, BUFFER+0 ;CALCULATE TOTAL TIME
MOV FIRSTTM+1, BUFFER+1
LCALL SUBTIME
LJMP COMPLETE
CALB: MOV A, BAHTBUF
EXB0: CJNE A, #06H, EXB1
MOV BAHTBUF, #04H
JMP TRD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXB1:  CJNE  A, #09H, EXB2
        MOV   BAHTBUF, #06H
        JMP   TRD
EXB2:  CJNE  A, #12H, EXB3
        MOV   BAHTBUF, #08H
        JMP   TRD
EXB3:  CJNE  A, #15H, EXB4
        MOV   BAHTBUF, #10H
        JMP   TRD
EXB4:  MOV   BAHTBUF, #12H
TRD:    LCALL TIMERD
        MOV   A, TIMBUF+3
        CJNE  A, #22H, GOII
        MOV   LASTTM+0, #22H
        MOV   LASTTM+1, #00H
        LCALL SUBTIME
CHARGEII: MOV A, HRESUL
        MOV   B, #60
        MUL  AB
        ADD  A, MRESUL ;TALK NOT MORE THAN 255 MIN.
        MOV  B, BAHTBUF
        MUL  AB
        MOV  CHARGE+0, A ;MONEY LOW BYTE
        MOV  CHARGE+1, B ;MONEY HIGH BYTE
        MOV  A, BAHTBUF
STI0:  CJNE  A, #04H, STI1
        MOV  BAHTBUF, #03H
        JMP  TOTALII
STI1:  CJNE  A, #06H, STI2
        MOV  BAHTBUF, #05H
        JMP  TOTALII
STI2:  CJNE  A, #08H, STI3
        MOV  BAHTBUF, #06H
        JMP  TOTALII
STI3:  CJNE  A, #10H, STI4
        MOV  BAHTBUF, #08H
        JMP  TOTALII
STI4:  MOV  BAHTBUF, #09H
TOTALII: MOV FIRSTTM+0, #22H
        MOV  FIRSTTM+1, #00H
        LCALL GOII
GOII:  MOV  DPTR, #PORTC
        MOVX A, @DPTR
        JNB  ACC.5, LTIR
        LJMP CALB
LTIR:  LCALL TIMERD
        MOV  LASTTM+0, TIMBUF+3
        MOV  LASTTM+1, TIMBUF+4
        LCALL SUBTIME
        LCALL FEE ;FIND TOTAL MONEY
NETTIMII: MOV FIRSTTM+0, BUFFER+0 ;CALCULATE TOTAL TIME
        MOV  FIRSTTM+1, BUFFER+1
        LCALL SUBTIME
        LJMP COMPLETE
CALIII: MOV  A, BAHTBUF
EXOI:  CJNE  A, #06H, EX1I
        MOV  BAHTBUF, #03H
        JMP  TID

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EX1I:   CJNE   A,#09H,EX2I
        MOV    BAHTBUF,#05H
        JMP    TID
EX2I:   CJNE   A,#12H,EX3I
        MOV    BAHTBUF,#06H
        JMP    TID
EX3I:   CJNE   A,#15H,EX4I
        MOV    BAHTBUF,#08H
        JMP    TID
EX4I:   MOV    BAHTBUF,#09H
TID:    LCALL  TIMERD
        MOV    A,TIMBUF+3
        CJNE   A,#06H,GOII
        MOV    LASTTM+0,#06H
        MOV    LASTTM+1,#00H
        LCALL  SUBTIME
CHARGIII:MOV  A,HRESUL
        MOV    B,#60
        MUL   AB
        ADD   A,MRESUL           ;TALK NOT MORE THAN 255 MIN.
        MOV   B,BAHTBUF
        MUL   AB
        MOV   CHARGE+0,A        ;MONEY LOW BYTE
        MOV   CHARGE+1,B        ;MONEY HIGH BYTE
        MOV   A,BAHTBUF
STI0I:  CJNE   A,#03H,STI1I
        MOV    BAHTBUF,#06H
        JMP    TOTALIII
STI1I:  CJNE   A,#05H,STI2I
        MOV    BAHTBUF,#09H
        JMP    TOTALIII
STI2I:  CJNE   A,#06H,STI3I
        MOV    BAHTBUF,#12H
        JMP    TOTALIII
STI3I:  CJNE   A,#08H,STI4I
        MOV    BAHTBUF,#15H
        JMP    TOTALIII
STI4I:  MOV    BAHTBUF,#18H
TOTALIII:MOV  FIRSTTM+0,#06H
        MOV    FIRSTTM+1,#00H
        LCALL  GOIII
GOIII:  MOV    DPTR,#PORTC
        MOVX   A,@DPTR
        JNB   ACC.5,LMRD
        LJMP  CALIII
LMRD:   LCALL  TIMERD
        MOV    LASTTM+0,TIMBUF+3
        MOV    LASTTM+1,TIMBUF+4
        LCALL  SUBTIME
        LCALL  FEE               ;FIND TOTAL MONEY
NETTIB:MOV  FIRSTTM+0,BUFFER+0   ;CALCULATE TOTAL TIME
        MOV    FIRSTTM+1,BUFFER+1
        LCALL  SUBTIME
COMPLETE: NOP
        RET
;*****NET MONEY TO PAY*****
FEE:    MOV    A,HRESUL
        MOV    B,#60

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MUL    AB
ADD    A, MRESUL
MOV    B, BAHTBUF
MUL    AB
MOV    DPL, A
MOV    DPH, B
LCALL  HTOD
MOV    CHARGE+0, R2
MOV    CHARGE+1, R3
RET

; ***** DPADD SUB *****
; DPTR = DPTR + R2, R3
; IN   = DPTR, R2, R3
; OUT  = DPTR, CY
; REG  = A, DPTR
DPADD: MOV    A, DPL
        ADD    A, R3
        MOV    DPL, A
        MOV    A, DPH
        ADDC   A, R2
        MOV    DPH, A
        RET

; ***** HTOD SUB *****
; HEX TO DECIMAL
; IN   = DPTR
; OUT  = R1, R2, R3
; REG  = A, R0, R1, R2, R3, R4, R5, DPTR
HTOD:  CLR    A ; CLEAR OUTPUT
        MOV    R1, A
        MOV    R2, A
        MOV    R3, A
        MOV    R4, #16 ; SHIFT 16 BIT
HTOD1: MOV    A, DPL
        RLC    A
        MOV    DPL, A
        MOV    A, DPH
        RLC    A
        MOV    DPH, A
        MOV    R5, #3 ; ADD DECIMAL
        MOV    R0, #3 ; INDEX TO R3
HTOD2: MOV    A, @R0
        ADDC   A, ACC
        DA    A
        MOV    @R0, A
        DEC    R0
        DJNZ   R5, HTOD2
        DJNZ   R4, HTOD1
        RET

; *****
END

```

Program receive data from serial port

```
#include <bios.h>
#include <conio.h>
#include <stdio.h>

#define COM2          1
#define DATA_READY 0x100
#define TRUE         1
#define FALSE        0

#define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
int code0[100],code1[100];      // all 19 byte
int tel12[100],tel34[100],tel56[100],tel78[100],tel9[100];
int firsthr[100],firstmin[100];
int lasthr[100],lastmin[100];
int totalhr[100],totalmin[100];
int charge0[100],charge1[100];
int countdat;
int alldata;

int main(void)
{
    int in,out,order, status, DONE = FALSE;
    clrscr();
    countdat=1;
    order=1;
    alldata=0;
    _setcursortype(_NOCURSOR);
    bioscom(0, SETTINGS, COM2);
    printf("(Press [ESC] to exit ...)\n");
    gotoxy(23,3);
    printf("      LIST OF TELEPHONE\n");
    gotoxy(9,5);
    printf(" CODE      TELEPHONE NO.      START      END      TOTAL      CHARGE ");
    window(9,7,79,25);
    while (!DONE)
    {
        status = bioscom(3, 0, COM2);
        if (status & DATA_READY)
            if ((out = bioscom(_COM_RECEIVE, 0, COM2) & 0x7F) != 0)
                { textcolor(14);
                    switch (countdat){
                        case 1:code0[order]=out;
                                countdat=countdat+1;
                                printf(" %d",out);
                                break;
                        case 2:code1[order]=out;
                                countdat=countdat+1;
                                printf("%d      ",out);
                                break;
                        case 3:tel12[order]=out;
                                countdat=countdat+1;
                                break;
                        case 4:tel34[order]=out;
                                countdat=countdat+1;
                                break;
                        case 5:tel56[order]=out;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        countdat=countdat+1;
        break;
    case 6:tel78[order]=out;
        countdat=countdat+1;
        break;
    case 7:tel9[order]=out;
        countdat=countdat+1;
        printf("%d%d%d%d%d      ",tel12[order],
            tel34[order],tel56[order],tel78[order],tel9[order]);
        break;
    case 8:firsthr[order]=out;
        countdat=countdat+1;
        printf(" %d:",out);
        break;
    case 9:firstmin[order]=out;
        countdat=countdat+1;
        printf("%d ",out);
        break;
    case 10:lasthr[order]=out;
        countdat=countdat+1;
        printf(" %d:",out);
        break;
    case 11:lastmin[order]=out;
        countdat=countdat+1;
        printf("%d ",out);
        break;
    case 12:totalhr[order]=out;
        countdat=countdat+1;
        printf(" %d:",out);
        break;
    case 13:totalmin[order]=out;
        countdat=countdat+1;
        printf("%d ",out);
        break;
    case 14:charge0[order]=out;
        countdat=countdat+1;
        printf(" %d",out);
        break;
    case 15:charge1[order]=out;
        order=order+1;
        countdat=1;
        alldata=alldata+1;
        printf("%d baht\n",out);
        break;
    } // window
}
if (kbhit())
{
    if ((in = getch()) == '\x1B')
        DONE = TRUE;
}
}
_setcursortype(_NORMALCURSOR);
return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้