

เครื่องับการใช้งานโทรศัพท์



นางสาวจินตนา แสนศรี	42015679
นายณัฐพงษ์ บุญศรี	42015683
นายอัครวัฒน์ สุขสวัสดิ์	42015715

2544

เลขที่.....
 เลขทะเบียน..... 42348
 วัน, เดือน, ปี 17 พ.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต
 ภาควิชาเทคนิคอุตสาหกรรม
 คณะวิศวกรรมศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ปีการศึกษา 2544

หัวข้อปริญญานิพนธ์ เครื่องตรวจนับการใช้โทรศัพท์

ชื่อนักศึกษา นางสาวจินตนา แสนศรี 42015679

นายณัฐพงษ์ บุญศรี 42015683

นายอัครวัฒน์ สุขสวัสดิ์ 42015715

อาจารย์ที่ปรึกษา อาจารย์ณภพินท์ อนันตรศิริชัย

ภาควิชา เทคโนโลยีอุตสาหกรรม

ปีการศึกษา 2544

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

ประธานกรรมการ

()

กรรมการ

()

กรรมการ

()

กรรมการ

()

กรรมการ

()

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องตรวจนับการใช้โทรศัพท์
ชื่อนักศึกษา	นางสาวจินตนา แสนศรี 42015679 นายณัฐพงษ์ บุญศรี 42015683 นายอัศววัฒน์ สุขสวัสดิ์ 42015715
อาจารย์ที่ปรึกษา	อาจารย์นภพินท์ อนันตรศิริชัย
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2544

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาหลักการทำงานของเครื่องโทรศัพท์และมาตรฐานของสัญญาณในระบบโทรศัพท์ เพื่อนำไปประยุกต์สร้างเครื่องนับการใช้งานโทรศัพท์ โดยจำนวนครั้งที่ตรวจสอบได้ มีความแม่นยำ น่าเชื่อถือ และสามารถเปรียบเทียบกับข้อมูลกับที่องค์กรโทรศัพท์แจ้งมาได้

เครื่องนับการใช้งานโทรศัพท์นี้ จะมีไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวควบคุมการทำงาน โดยจะรับสัญญาณสถานะขहु วงหุ, สัญญาณให้หมุม, สัญญาณเรียกกลับ, สัญญาณสายไม่ว่าง, สัญญาณถอดรหัส DTMF และสัญญาณถอดรหัสสี่คีย์ มาประมวลผลแล้วแสดงผลออกทางจอ LCD

ภาคแสดงผลด้วยจอ LCD จะแสดงวันที่, เวลา และจำนวนครั้งในการเรียกออกภายในรหัสพื้นที่เดียวกัน นอกจากนี้ยังสามารถนำเครื่องนับการใช้งานโทรศัพท์นี้ไปต่อกับเครื่องคอมพิวเตอร์ เพื่อพิมพ์ข้อมูลออกทางเครื่องพิมพ์ได้

Project Title	Telephone counter	
Name	Ms. Jintana Sansri	42015679
	Mr. Nattapong Boonsri	42015683
	Mr. Akkarawat Suksawat	42015715
Project Advisor	Ms.Noppin Anantrasirichai	
Department	Industrial Technology	
Academic Year	2001	

ABSTRACT

This project is studying in principle of telephone subscriber and standard signal in telephone system for applied in this Telephone Counter. The checking number of calling that can check will be corrected, trusted and can be compared with information from Telephone Organization.

This Telephone Counter used microcontroller MCS-51 to control the operation system. It will get handcheck signal, dial tone, ringing tone, DTMF decoder signal and key decoder signal for processing and then display on LCD monitor.

LCD display section will display date, time and number of calling in the same area code. The data output from Telephone Counter can be recorded in personal computer and print out.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์ เสนอแนะข้อมูล และให้คำปรึกษาที่ดียิ่งจากอาจารย์ที่ปรึกษาปริญญาบัตร คือ อาจารย์ณภินท์ อนันตรศิริชัย และพนักงานองค์การโทรศัพท์แห่งประเทศไทย ชุมสายโทรศัพท์ตำราญราษฎร์ จึงได้ขอขอบพระคุณทุกท่านไว้ ณ.ที่นี้ ที่ได้ให้ความช่วยเหลือและอำนวยความสะดวกในการจัดทำโครงการให้สำเร็จได้ด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูปภาพ	ฉ
สารบัญตาราง	ช
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	2
2.1 ระบบโทรศัพท์	2
2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	9
2.3 ภาคตรวจจับสัญญาณเสียง	21
2.4 วงจรถอดรหัสเลขหมาย (DTMF Decode) โดยใช้ไอซี MT8870	23
2.5 วงจรถอดรหัสการกดคีย์ (keypad decoder) โดยใช้ไอซี 74C922	30
2.6 การจับโมดูลแสดงผลแบบผลึกเหลว (LCD MODULE)	32
บทที่ 3 การออกแบบและการสร้าง	44
การออกแบบส่วนที่เป็น HARD WARE	44
3.1 วงจรตรวจจับสัญญาณ โทรศัพท์	44
3.2 การออกแบบวงจรตรวจจับสัญญาณเสียง	45
3.3 การออกแบบวงจรถอดรหัสสัญญาณ DTMF	47
3.4 การออกแบบวงจรถอดรหัสการกดคีย์	47
3.5 การทำงานรวมของวงจร	48
การออกแบบส่วนที่เป็น SOFT WARE	48
บทที่ 4 การทดลองและผลการทดลอง	51
4.1 วงจรตรวจการยกหู	51
4.2 วงจรตรวจจับสัญญาณเสียง	51
4.3 วงจรตรวจสอบสัญญาณ DTMF	54
4.4 วงจรถอดรหัสการกดคีย์	55

สารบัญ (ต่อ)

	หน้า
บทที่ 5 บทสรุป ปัญหาและข้อเสนอแนะ	56
บรรณานุกรม	58
ภาคผนวก	
ภาคผนวก ก วงจรรวมและลายแผ่นปริ้นท์	59
ภาคผนวก ข DATA SHEET ของไอซีที่ใช้ในโครงการ	66
ภาคผนวก ค SOURCE CODE ของโปรแกรมส่วนควบคุม	81

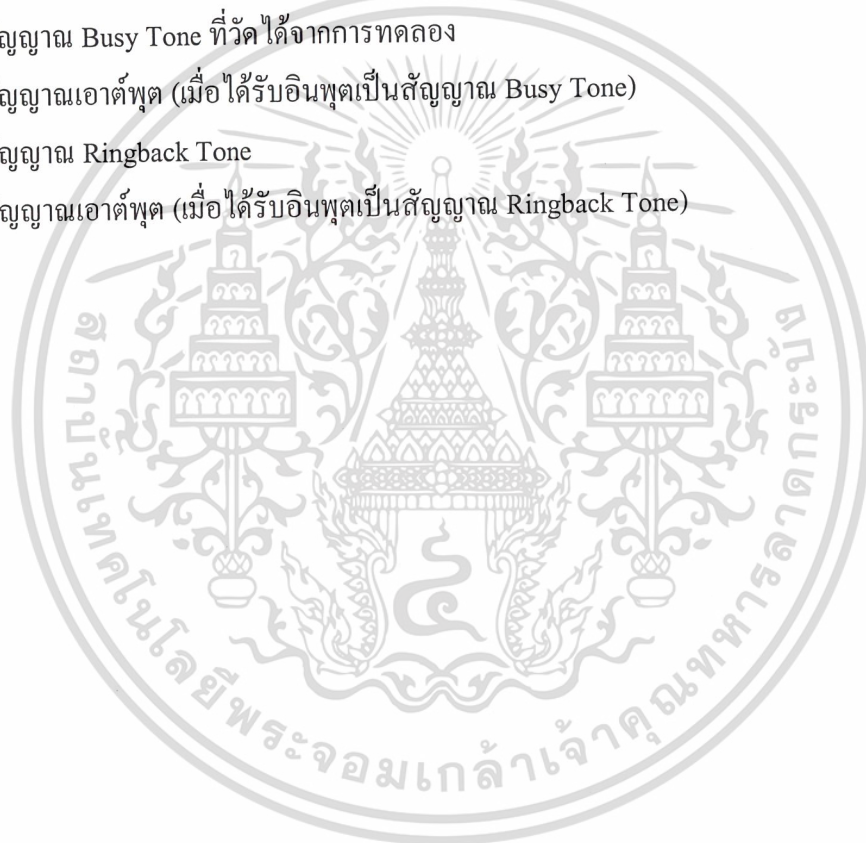


สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 การเชื่อมต่อระหว่างชุมสายโทรศัพท์	2
รูปที่ 2.2 บล็อกไดอะแกรมของเครื่องรับโทรศัพท์	4
รูปที่ 2.3 แป้นกดหมายเลขของเครื่องโทรศัพท์แบบกดปุ่มและค่าความถี่ที่ใช้	6
รูปที่ 2.4 การติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ที่อยู่ในชุมสายโทรศัพท์เดียวกัน	8
รูปที่ 2.5 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051	10
รูปที่ 2.6 โครงสร้างขาของ 8051	11
รูปที่ 2.7 โครงสร้างภายในพอร์ตอินพุตเอาต์พุตของ 8051	13
รูปที่ 2.8 หน่วยความจำโปรแกรมของ 8051	14
รูปที่ 2.9 หน่วยความจำข้อมูลของ 8051	15
รูปที่ 2.10 หน่วยความจำข้อมูลภายใน	16
รูปที่ 2.11 รายละเอียดของรีจิสเตอร์ฟังก์ชันพิเศษ	16
รูปที่ 2.12 ขาสัญญาณต่างๆและโครงสร้างภายในของ 8255	19
รูปที่ 2.13 ความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ 8255	22
รูปที่ 2.14 ลักษณะของสัญญาณโทรศัพท์	22
รูปที่ 2.15 โครงสร้างภายใน IC XR-2211	23
รูปที่ 2.16 วงจรถอดรหัสเลขหมาย	24
รูปที่ 2.17 ความถี่ที่ได้จากภาคกรองความถี่	26
รูปที่ 2.18 การต่อวงจรควบคุมตรวจสอบสัญญาณเบื้องต้นของไอซี MT8870	28
รูปที่ 2.19 การต่อวงจรภาคอินพุตของ ไอซี MT8870	28
รูปที่ 2.20 การต่อวงจรผลิตความถี่	29
รูปที่ 2.21 ไดอะแกรมสัญญาณต่างๆของไอซี MT8870	30
รูปที่ 2.22 โครงสร้างภายในของไอซี 74C922	31
รูปที่ 2.23 รูปร่างและการจัดขาสัญญาณของแอลซีดีโมดูล	33
รูปที่ 2.24 ไดอะแกรมการทำงานของแอลซีดีโมดูลแบบอักษร	35
รูปที่ 3.1 วงจรตรวจสอบการยกหูวางหู	44
รูปที่ 3.2 วงจรตรวจจับสัญญาณเสียง	46
รูปที่ 3.3 วงจรตรวจจับสัญญาณ DTMF	47

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 3.4 วงจร scan keypads	47
รูปที่ 3.5 วงจรรวมส่วนฮาร์ดแวร์	49
รูปที่ 3.6 Flow chart รวมของโปรแกรมในส่วนควบคุม	50
รูปที่ 4.1 สัญญาณ Dial Tone ที่วัดได้จากการทดลอง	51
รูปที่ 4.2 สัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Dial tone)	52
รูปที่ 4.3 สัญญาณ Busy Tone ที่วัดได้จากการทดลอง	52
รูปที่ 4.4 สัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Busy Tone)	53
รูปที่ 4.5 สัญญาณ Ringback Tone	53
รูปที่ 4.6 สัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Ringback Tone)	54



สารบัญตาราง

	หน้า
ตารางที่ 1 หน้าที่พิเศษของพอร์ต 3	12
ตารางที่ 2 การกำหนดรูปแบบการทำงานของพอร์ต 8255	18
ตารางที่ 3 หน้าที่การทำงานของขาสัญญาณ ไอซี 8255	18
ตารางที่ 4 ระบบรีจิสเตอร์ของ 8255	20
ตารางที่ 5 ข้อมูลทางขาสัญญาณ	20
ตารางที่ 6 ตำแหน่งของรีจิสเตอร์	21
ตารางที่ 7 คำรหัสที่ได้จากความถี่ต่างๆ	26
ตารางที่ 8 ขาสัญญาณของแอลซีดี โมดูล	34
ตารางที่ 9 ชุดคำสั่ง	37
ตารางที่ 10 การกำหนดค่า clear display	38
ตารางที่ 11 การกำหนดค่าตำแหน่ง DDRAM	39
ตารางที่ 12 การกำหนดค่า mode set	39
ตารางที่ 13 การกำหนดค่า ON/OFF	39
ตารางที่ 14 การกำหนดค่าการเลื่อน	40
ตารางที่ 15 การกำหนดค่าการเซตฟังก์ชัน	40
ตารางที่ 16 การกำหนดค่าตำแหน่งของ CGRAM	41
ตารางที่ 17 การกำหนดค่าตำแหน่งของ DDRAM	41
ตารางที่ 18 การกำหนดค่าตำแหน่งของ DDRAM ที่ใช้งาน	41
ตารางที่ 19 การกำหนดค่า	41
ตารางที่ 20 การเขียนข้อมูลลง DDRAM หรือ CGRAM	42
ตารางที่ 21 การอ่านข้อมูลลง DDRAM หรือ CGRAM	42
ตารางที่ 22 การทดสอบวงจรส่วนการถอดรหัสเลขหมาย	54
ตารางที่ 23 เอต์พุดที่ได้จากการทดสอบวงจร Keypad decode	55

บทที่ 1

บทนำ

ในการติดต่อสื่อสารทางโทรศัพท์หรือการใช้งานโทรศัพท์นั้น สามารถแบ่งลักษณะการคิดคำนวณค่าใช้จ่ายการให้บริการแต่ละครั้งตามพื้นที่ใช้งาน กล่าวคือถ้าเป็นการให้บริการโดยการเรียกออกไปยังเลขหมายนอกพื้นที่หรือเรียกไปยังโทรศัพท์เคลื่อนที่ ก็จะมีการแจ้งให้ผู้ให้บริการทราบถึงวันที่ เวลา ระยะเวลาใช้งาน และจำนวนครั้งในการใช้งานให้กับผู้ใช้บริการได้ทราบ แต่ถ้าเป็นการเรียกไปยังเลขหมายภายในพื้นที่เดียวกัน ผู้ใช้จะไม่สามารถทราบรายละเอียดในการให้บริการเหมือนกับ การเรียกไปยังเลขหมายนอกพื้นที่หรือเรียกไปยังโทรศัพท์มือถือได้เลย ซึ่งในกรณีที่เป็นการเรียกไปยังเลขหมายภายในพื้นที่เดียวกันนี้ จะไม่สามารถตรวจสอบความผิดพลาด ที่อาจเกิดขึ้นจากการนับจำนวนครั้งการใช้โทรศัพท์ได้เลย

ด้วยเหตุผลดังกล่าว จึงเป็นเหตุจูงใจในการที่จะทำการสร้างเครื่องนับการใช้งานโทรศัพท์ (Telephone counter) ให้สามารถนับจำนวนการเรียกออกไปยังเลขหมายภายในพื้นที่เดียวกันได้อย่างถูกต้อง กล่าวคือ ขณะที่ผู้เรียกทำการเรียกออกนั้น ผู้เรียกออกซึ่งเรียกว่า “ผู้เรียก” จะทำการเรียกออกไปยังเลขหมายปลายทางเลขหมายหนึ่ง ซึ่งเป็นเลขหมายของผู้ที่ถูกเรียกเข้าซึ่งเรียกว่า “ผู้ถูกเรียก” เมื่อผู้ถูกเรียกได้ทำการยกหูโทรศัพท์ขึ้น เครื่องนับการใช้งาน โทรศัพท์จะทำการนับจำนวนครั้งการใช้งานเป็นหนึ่งครั้ง ในกรณีที่ทางด้านผู้ถูกเรียกไม่มีผู้รับสายจนกระทั่งทางชุมสายโทรศัพท์ได้ทำการตัดสัญญาณเรียกหรือผู้ถูกเรียกมีการใช้งานคู่สายอยู่ ก็จะไม่ถือว่าเป็นนับจำนวนครั้งการใช้งาน

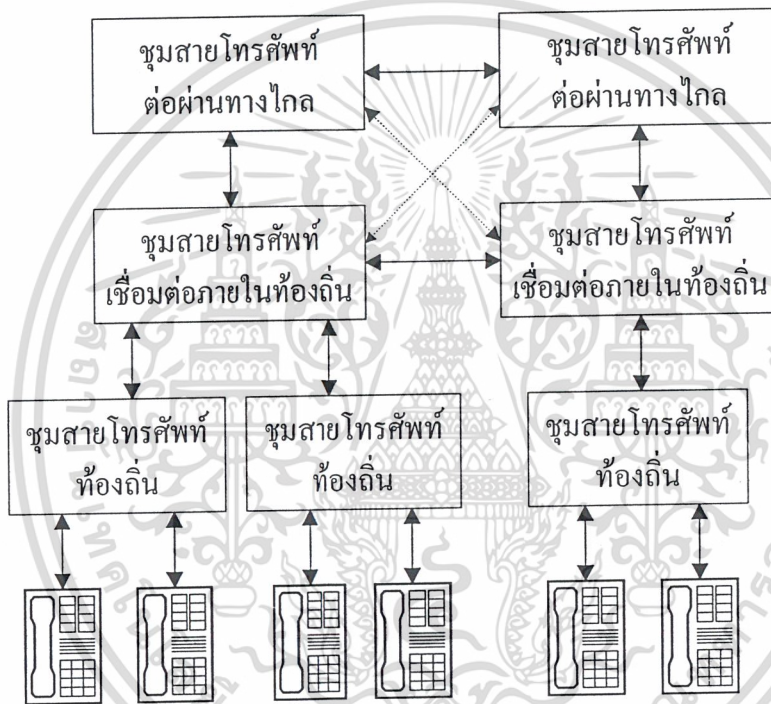
จากที่ได้กล่าวมาถ้าเครื่องนับการใช้งานโทรศัพท์นี้สามารถใช้งานได้จริง มีความถูกต้องและน่าเชื่อถือ ก็จะเป็นประโยชน์ต่อผู้ใช้บริการโทรศัพท์ตามบ้านเป็นอย่างมาก เพราะสามารถตรวจสอบข้อมูลที่ทางองค์การ โทรศัพท์ได้แจ้งมาว่าตรงกับข้อมูลที่ตรวจสอบได้จากเครื่องนับการใช้งานโทรศัพท์หรือไม่

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบโทรศัพท์

ชุมสายโทรศัพท์

การใช้งานโทรศัพท์ในการติดต่อกัน ระหว่างเครื่องโทรศัพท์ ผู้ใช้ จะกระทำผ่านชุมสายโทรศัพท์ต่าง ๆ ที่องค์การโทรศัพท์มีให้บริการ มีรายละเอียดดังรูปที่ 2.1



รูปที่ 2.1 การเชื่อมต่อระหว่างชุมสายโทรศัพท์ที่ใช้ในการติดต่อระหว่างเครื่องโทรศัพท์ผู้ใช้

โดยทั่วไปสามารถแบ่งชุมสายโทรศัพท์ออกเป็น 2 ประเภทใหญ่ ๆ ได้แก่

ชุมสายโทรศัพท์ที่มีเครื่องโทรศัพท์ผู้ใช้ของผู้เช่าต่อเข้าโดยตรง

ชุมสายโทรศัพท์ที่มีเครื่องโทรศัพท์ผู้ใช้ของผู้เช่าต่อเข้าโดยตรง ได้แก่ ชุมสายโทรศัพท์ท้องถิ่น และชุมสายโทรศัพท์ปลายทางอัตโนมัติ

1. ชุมสายโทรศัพท์ท้องถิ่น (Local Exchange)

เป็นชุมสายโทรศัพท์ที่มีเครื่องโทรศัพท์ของผู้เช่าโดยตรง ชุมสายแบบนี้มีขนาดตั้งแต่เป็นร้อย ๆ หมายเลขจนถึงเป็นหมื่นหมายเลขหรือมากกว่านั้น

2. ขุมสายโทรศัพท์ปลายทางอัตโนมัติ (PABX)

ขุมสายโทรศัพท์ปลายทางอัตโนมัติ (PABX) ได้ถูกพัฒนามาจากขุมสายโทรศัพท์ปลายทาง (PBX) ซึ่งเป็นอุปกรณ์ที่ให้บริการสำหรับผู้เช่า (Customer Premise Equipment : CPE) ที่ทำการติดตั้งในอาคารสำนักงาน บริษัท เป็นต้น เพื่อที่จะให้บริการเรียกสำหรับคู่สายสมาชิกได้ทั้งภายในและภายนอกของสำนักงานนั้น ๆ (ภายในขุมสาย) โดยไม่ต้องผ่านขุมสายท้องถิ่น

ขุมสายโทรศัพท์ที่ไม่มีเครื่องโทรศัพท์ผู้เช่าต่อเข้าโดยตรง

ขุมสายโทรศัพท์ที่ไม่มีเครื่องโทรศัพท์ผู้เช่าต่อเข้าโดยตรง เป็นขุมสายต่อผ่านที่ให้บริการเรียกระหว่างขุมสายโทรศัพท์ท้องถิ่นกับขุมสายโทรศัพท์ท้องถิ่นด้วยกัน การเรียกระหว่างผู้เช่าโทรศัพท์สองเครื่องอาจเรียกผ่านไปยังขุมสายต่อผ่านหลาย ๆ ขุมสายได้ ขุมสายต่อผ่านยังแบ่งออกเป็น 2 แบบคือ

1. ขุมสายโทรศัพท์เชื่อมต่อภายในท้องถิ่น (Tandem Exchange)

เป็นขุมสายโทรศัพท์ต่อผ่านที่ให้บริการเรียกระหว่างขุมสายโทรศัพท์กับขุมสายโทรศัพท์ท้องถิ่นภายในท้องถิ่น

2. ขุมสายโทรศัพท์ต่อผ่านทางไกล (Transit Exchange)

เป็นขุมสายโทรศัพท์ที่ให้บริการระหว่างขุมสายโทรศัพท์เชื่อมต่อภายในท้องถิ่นกับโทรศัพท์เชื่อมต่อภายในท้องถิ่น

เครื่องโทรศัพท์ผู้ใช้ (Subscriber)

1. โครงสร้างบล็อกโคอะแกรมของเครื่องโทรศัพท์

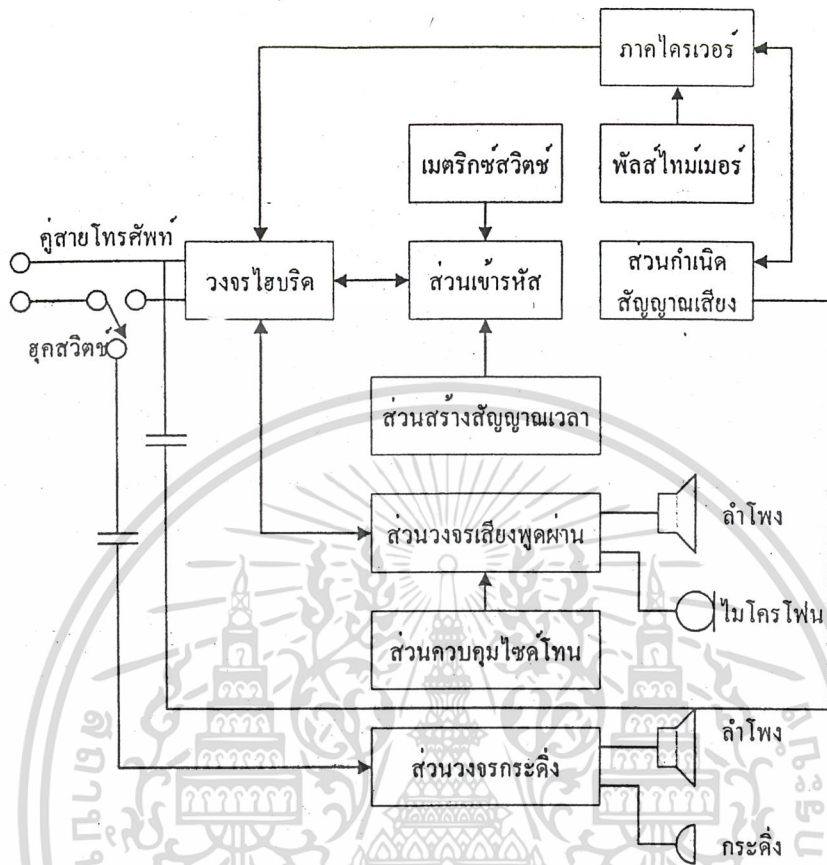
เครื่องโทรศัพท์จะมีรายละเอียด โครงสร้างตามบล็อกโคอะแกรมดังรูปที่ 2.2 จากรูปสามารถแบ่งโครงสร้างเครื่องโทรศัพท์ออกเป็น 3 ส่วนใหญ่ ๆ ด้วยกันคือ

ส่วนวงจรกระดิ่ง (Ringer)

ส่วนวงจรกระดิ่งนี้จะทำหน้าที่เรียกเป็นเสียงกระดิ่งทุกครั้งที่มีการเรียกเข้ามา เพื่อให้เจ้าของเครื่องทราบว่ามีการเรียก วงจรกระดิ่งอาจใช้กระดิ่งหรือไอซีโทนริงเกอร์ก็ได้

หากใช้กระดิ่ง เมื่อมีสัญญาณเรียกเข้ามาเป็นกระแสไฟสลับประมาณ 100-120 โวลต์ ส่งเข้ามายังโซลินอยด์ ทำให้เกิดสนามแม่เหล็กขึ้นเหมือนกระดิ่งหรือออด (Bell) ทั่ว ๆ ไป

หากเป็นระบบที่เป็นไอซี จะมีการนำเอาพัลส์ดังกล่าวเข้าสู่ระบบเร็คติไฟเพื่อจ่ายให้หน่วยผลิตความถี่ ส่งความถี่ออกถ้าโพงเป็นสัญญาณเรียกที่ใช้วิธีการสังเคราะห์เสียงในรูปแบบอิเล็กทรอนิกส์



รูปที่ 2.2 แสดงบล็อกไดอะแกรมของเครื่องรับโทรศัพท์

ส่วนวงจรปากพูดและหูฟัง (Speech Network)

ส่วนวงจรปากพูดและหูฟังนี้จะทำหน้าที่เสมือนวงจรเครื่องรับและเครื่องส่งที่ใช้ในการติดต่อกันระหว่างเครื่องโทรศัพท์ผู้ใช้ เมื่อมีการเรียกจากอีกฝ่ายหนึ่งสัญญาณเรียกจะถูกส่งผ่านระบบชุมสายจากชุมสายจะมีการเรียกด้วยระบบมัลติเพล็กซ์เพื่อแยกคู่สายไปยังเลขหมายที่ผู้เรียกต้องการติดต่อชุมสายจะส่งสัญญาณเรียกเป็นแรงดันไฟฟ้ากระแสสลับ 100-120 โวลต์ มายังเครื่องผู้รับ ถ้าเครื่องรับที่ออกแบบอย่างถูกต้องจะมีกระแสไหลในวงจรด้วยระบบเคอร์เรนท์เรกูเลเตอร์ (Current Regulator) รักษากระแสคงที่ไว้ที่ 23 มิลลิแอมป์ เมื่อผู้ถูกเรียกยกหูรับสาย ทั้งผู้เรียกและผู้ถูกเรียกสามารถพูดคุยสนทนาด้วยวงจรอินเตอร์คอมหรือวงจรเสียงพูดผ่าน เพียงแต่วงจรเสียงพูดผ่านต้องมีไซด์โทน (Side Tone) ไม่มากจนเกินไป

วงจรรหัสตัวเลขคีย์กด (Encoder)

เมื่อต้องการเรียกสาย คู่สนทนาต้องยกหูโทรศัพท์แล้วกดหมายเลขที่ต้องการ จากแป้นกด

(key) วงจรเข้ารหัสข้อมูลความถี่ผสมส่งความถี่สองความถี่ (สูง - ต่ำ) ไปยังระบบชุมสาย แล้วชุมสายจะส่งข้อมูลไปตามเครือข่าย หากส่วนนี้ไม่ทำงานเครื่องโทรศัพท์ที่ยังคงรับได้แต่ไม่สามารถโทรออกได้

ระบบของเครื่องโทรศัพท์

เครื่องโทรศัพท์ที่ใช้ในชุมสายโทรศัพท์มีอยู่ 2 ระบบด้วยกัน คือระบบโทน (Tone) หรือระบบดีทีเอ็มเอฟ (DTMF) กับระบบพัลส์ (Pulse) ซึ่งเป็นระบบดั้งเดิม โดยตัวเครื่องจะมีลักษณะหน้าปัดแบบหมุน จะใช้งานร่วมกับชุมสายท้องถิ่นระบบเก่า หากบริเวณใดเอาเครื่องระบบหมุนไปต่อแล้วไม่สามารถติดต่อกันได้ แสดงว่าชุมสายดังกล่าวเป็นระบบใหม่ ซึ่งจะต้องใช้เครื่องโทรศัพท์แบบกดปุ่มเท่านั้น แต่มีใจว่าบริเวณที่ใช้เครื่องแบบกดปุ่มจะต้องเป็นชุมสายระบบใหม่ทั้งหมดเสมอไป เพราะเมื่อหลายปีที่ผ่านมาบริษัทเอทีแอนด์ทีได้คิดค้นเครื่องโทรศัพท์ระบบกดปุ่มที่สามารถให้กำเนิดสัญญาณอิมพัลส์ เพื่อให้สามารถใช้กับชุมสายท้องถิ่นระบบเก่าได้ โดยสามารถสร้างพัลส์แบบเดียวกันกับเครื่องแบบหน้าปัดหมุนได้ ฉะนั้นเครื่องโทรศัพท์แบบกดในปัจจุบันสามารถที่จะใช้งานกับชุมสายโทรศัพท์ทั้งสองแบบได้ โดยจะมีสวิตช์เลือกให้เครื่องทำงานแบบทัชพัลส์ (Touch Pulse) หรือแบบกดปุ่ม (Touch Tone)

1. แบบหมุน (Rotary dial)

เครื่องโทรศัพท์ชนิดนี้จะสร้างสัญญาณจากกระแสลูปโดยต่อเข้ากับอุปกรณ์สวิตช์ และอุปกรณ์สวิตช์นี้จะทำหน้าที่ “เปิด” และ “ปิด” เข้ากับกลไกการหมุนเลขหมายในเครื่อง ทำให้กระแสพัลส์ตอบสนองกับหมายเลขที่หมุน โดยจำนวนพัลส์ที่เกิดขึ้นก็จะมีจำนวนเท่ากับหมายเลขที่หมุน

2. แบบกดปุ่ม (Touch tone)

เครื่องโทรศัพท์แบบนี้จะส่งความถี่สองความถี่ที่มอดูเลตกันของหมายเลขที่กดออกไป ความถี่ที่ส่งออกไปจะอยู่ในย่านของความถี่เสียงพูดโดยมีค่าประมาณ 0 – 4 กิโลเฮิร์ต โดยค่าความถี่ต่าง ๆ ในการกดเลขหมายจะแสดงได้ดังรูปที่ 2.3

ค่าความถี่ที่ใช้ในแต่ละแถวแต่ละคอลัมน์จะมีความถี่ต่างกัน ความถี่ของแต่ละแถวจะเป็นกลุ่มความถี่ต่ำ และความถี่ของแต่ละคอลัมน์จะเป็นกลุ่มความถี่สูง ความถี่ที่ได้จากการกดเลขหมายนี้ก็จะได้จากกรรมอดูเลตของความถี่สูงและความถี่ต่ำของแต่ละเลขหมาย เช่น การกดหมายเลข 1 จะค่าการมอดูเลตของความถี่ 597 เฮิร์ต และ 1209 เฮิร์ต เป็นต้น

ข้อดีของเครื่องโทรศัพท์ที่ใช้ระบบดีทีเอ็มเอฟ

1. สามารถลดเวลาในการหมุนเลขหมายลง ทำให้ติดต่อกับชุมสายได้เร็วขึ้น
2. มีความแม่นยำในการส่งเลขหมาย
3. ประหยัดอุปกรณ์หน่วยความจำที่ใช้ภายในชุมสาย
4. สามารถนำไปเชื่อมต่อกับอุปกรณ์ชุมสายได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		กลุ่มความถี่สูง (เฮิร์ต)				
		1209	1336	1477	1633	
กลุ่มความถี่ต่ำ (เฮิร์ต)	697	1	2	3	A	R1
	770	4	5	6	B	R2
	852	7	8	9	C	R3
	941	*	0	#	D	R4
		C1	C2	C3	C4	R : แถว C : คอลัมน์

รูปที่ 2.3 แสดงเป็นกคหมายเลขของเครื่องโทรศัพท์แบบกดปุ่มและค่าความถี่ที่ใช้

มาตรฐานสัญญาณโทรศัพท์พื้นฐานขององค์การโทรศัพท์

1. สัญญาณให้หมุน (Dial Tone)

เป็นสัญญาณที่ชุมสายโทรศัพท์ส่งไปให้เครื่องโทรศัพท์จะมีความถี่ต่อเนื่องประมาณ 400 เฮิร์ต เพื่อแสดงให้ผู้เรียกรู้ว่า พร้อมรับหมายเลขโทรศัพท์ที่ปลายทางแล้ว

2. สัญญาณหมายเลข

เป็นสัญญาณหมายเลขที่ต้องการติดต่อ โดยเครื่องโทรศัพท์ส่งสัญญาณจะส่งสัญญาณนี้ไปยังชุมสาย ซึ่งมีอยู่ 2 แบบคือ สัญญาณแบบพัลส์ (สำหรับเครื่องโทรศัพท์แบบหมุน) และสัญญาณความถี่ผสม (สำหรับเครื่องโทรศัพท์แบบกดปุ่ม)

สัญญาณพัลส์ (Pulse)

เครื่องโทรศัพท์ระบบหมุน จะผลิตสัญญาณพัลส์เป็นช่วง ๆ ตามจำนวนหมายเลขที่ต้องการเรียก โดยการผลิตนั้นอาจทำได้โดยตัดต่อสวิตช์ทางกล เช่น ในระบบงานหมุน (Dial) หรือใช้การผลิตความถี่ด้วยออสซิลเลเตอร์ ในกรณีที่เป็นเครื่องแบบกดปุ่ม

สัญญาณความถี่ผสม (Dual Tone Multi Frequency dialing:DTMF)

เครื่องโทรศัพท์แบบกดปุ่มจะผลิตสัญญาณความถี่แบบผสม หรือในบางครั้งเรียกว่าระบบทัชโทน (Touch Tone) โดยการส่งหมายเลขใดหมายเลขหนึ่งจะประกอบด้วยโทนเสียง 2 ความถี่ผสมกัน โดยเราพิจารณาปุ่มที่กดหรือคีย์บอร์ด การแบ่งกลุ่มความถี่ตามแถวหรือหลัก หากกดหมายเลข 9 ซึ่งอยู่แถวที่ 3 ความถี่ 825 เฮิร์ต และหลักที่ 3 ความถี่ 1447 เฮิร์ต วงจรภายในเครื่องจะทำการเข้ารหัสเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ได้ แล้วส่งให้วงจรผลิตสัญญาณผสมระหว่างสัญญาณความถี่แฉวและหลัก แล้วส่งออกไปยังคู่สายโทรศัพท์

3. สัญญาณไม่ว่าง (Busy Tone)

เป็นสัญญาณที่ชุมสายส่งไปยังเครื่องโทรศัพท์เพื่อบอกให้ผู้เรียกทราบว่า สายไม่ว่าง ผู้เรียกควรวางหูสักรุ่นแล้วค่อยโทรเรียกใหม่ สัญญาณไม่ว่างจะมีความถี่ 400 เฮิร์ต เป็นเสียงดังเป็นเวลาประมาณ 0.5 วินาที เสียงเงียบ 0.5 วินาที

4. สัญญาณกริ่งเรียกหรือสัญญาณกระดิ่ง (Ringing Tone)

เป็นสัญญาณที่ชุมสายแจ้งให้ผู้ถูกเรียกทราบว่า มีคนโทรเรียกเข้ามา ให้ไปรับสายสัญญาณกริ่งเรียกนี้เป็นสัญญาณ 25 เฮิร์ต 100 โวลต์พีคทูพีค เสียงดังประมาณ 1 วินาที เสียงเงียบประมาณ 4 วินาที โดยชุมสายโทรศัพท์ทั่วไปจะส่งสัญญาณกริ่งนี้ไปประมาณ 15 ครั้ง

5. สัญญาณเรียกกลับ (Ring Back Tone)

เป็นสัญญาณที่ชุมสายแจ้งให้ผู้เรียกทราบว่า การเรียกเป็นผลสำเร็จ ให้รอการรับสายของผู้รับได้ จะเป็นสัญญาณ 400 เฮิร์ต ส่งเสียงดัง 4 วินาที เสียงเงียบ 1 วินาที

รายละเอียดการติดต่อระหว่างเครื่องโทรศัพท์ผู้ใช้ผ่านชุมสายโทรศัพท์

ในการติดต่อกันระหว่างเครื่องโทรศัพท์ผู้ใช้จะมีอยู่ 2 กรณีด้วยกัน คือ การติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ที่อยู่ในชุมสายโทรศัพท์เดียวกันและการติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ต่างชุมสายกัน

1. การติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ที่อยู่ในชุมสายโทรศัพท์เดียวกัน

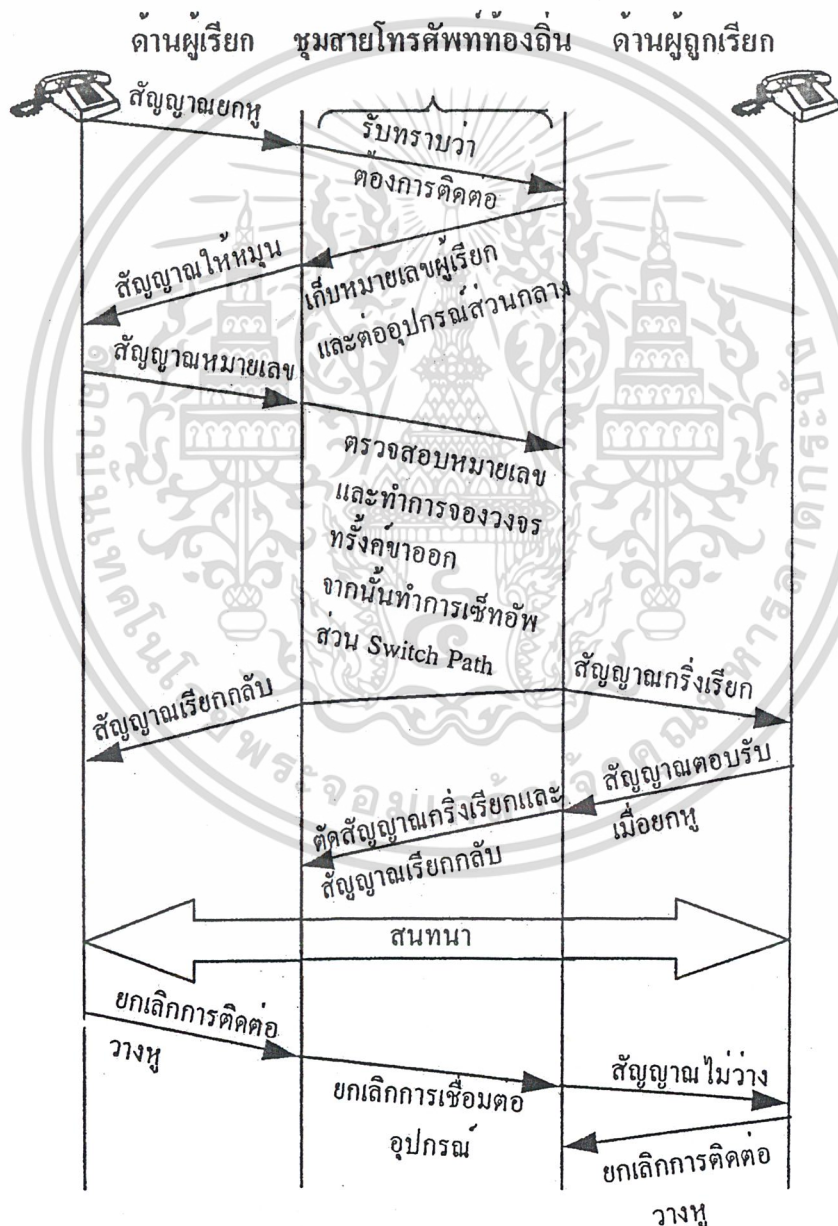
กรณีที่ในการติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ที่อยู่ในชุมสายโทรศัพท์เดียวกัน ด้านของผู้เรียก (Calling Subscriber) เมื่อยังไม่ยกหูจะมีแรงไฟตรงที่คู่สายประมาณ 48 โวลต์ เมื่อผู้เรียกยกหู (Off-Hook) เพื่อทำการโทรเรียกไปยังผู้รับจะทำให้ระดับแรงดันไฟตรงที่คู่สายโทรศัพท์เปลี่ยนแปลงระดับไปจาก 48 โวลต์ ไปเป็น 10 โวลต์ ชุมสายโทรศัพท์จะรับรู้ว่าเป็นการเริ่มต้นการเรียก ฉะนั้นชุมสายโทรศัพท์จะส่ง สัญญาณให้หมუნ ไปยังผู้เรียก (Called Subscriber) จากนั้นวงจรภายในเครื่องโทรศัพท์ก็จะสร้างสัญญาณรหัสหมายเลขส่งไปยังชุมสายโทรศัพท์ ชุมสายโทรศัพท์จะตัดสัญญาณให้หมუნออกทันทีที่ได้รับหมายเลขตัวแรก

เมื่อชุมสายโทรศัพท์รับหมายเลขผู้ถูกเรียก ก็จะทำการจองทางผ่านระหว่างผู้เรียกและผู้ถูกเรียก รวมทั้งส่ง สัญญาณกริ่ง ไปยังผู้ถูกเรียกพร้อมทั้งส่ง สัญญาณเรียกกลับ ไปยังผู้เรียก หากคู่สายทางด้านผู้ถูกเรียกไม่ว่าง ชุมสายจะส่ง สัญญาณไม่ว่าง ไปให้ผู้เรียกทราบ หากคู่สายด้านผู้รับว่าง ชุมสายจะส่ง สัญญาณกริ่ง เรียกไปยังผู้ถูกเรียก และ สัญญาณเรียกกลับ ไปยังด้านผู้เรียก เมื่อผู้เรียกได้ยินเสียงกริ่งแล้วยกหูรับสายจะทำให้ระดับไฟตรงที่คู่สายของผู้ถูกเรียกเปลี่ยนจาก 48 โวลต์ เป็น 10 โวลต์ จะเกิด สัญญาณตอบรับ ส่งไปยังชุมสาย ชุมสายก็จะตัด สัญญาณกริ่ง เรียกด้านผู้ถูกเรียก และ

ยกเลิก สัญญาณเรียกกลับ ด้านผู้เรียก ทำให้ทางผ่านระหว่างทรีงขาออก (Outing Trunk) ของผู้เรียก และทรีงขาเข้า (Incoming Trunk) ของผู้รับว่าง การสนทนาจึงจะสามารถเริ่มต้นได้

เมื่อการสนทนาสิ้นสุดลง คู่สนทนาทั้งสองจะทำการวางหูทำให้ชุมสายรับรู้ว่าการติดต่อ ด้านผู้รับเมื่อมีการเรียกจากภายนอกชุมสายจะส่ง สัญญาณกริ่งเรียก ขนาด 100 โวลต์พีคทูพีค ไปยังผู้รับ ถ้าผู้รับยังไม่มารับสาย สัญญาณกริ่งเรียก ก็จะดำเนินอยู่ประมาณ 15 ครั้ง จากนั้นชุมสายจะทำการตัดสัญญาณเองหากไม่มีการตอบรับ และส่ง สัญญาณไม่ว่าง ไปยังผู้เรียกเพื่อให้ทำการเรียกใหม่

โครงสร้างการทำงานแสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 แสดงการติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้อยู่ภายในชุมสายโทรศัพท์เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ต่างชุมสายกัน

ในการติดต่อเมื่อเครื่องโทรศัพท์ผู้ใช้ต่างชุมสายกันนั้น การติดต่อด้านผู้เรียกจากชุมสายท้องถิ่นไปยังชุมสายเชื่อมต่อภายในท้องถิ่น ด้านผู้เรียกจะแตกต่างจากการติดต่อภายในชุมสายเดียวกัน คือ เมื่อชุมสายท้องถิ่นได้รับสัญญาณเลขหมายปลายทาง จะส่ง สัญญาณยกหู ไปยังชุมสายเชื่อมต่อภายในท้องถิ่น จากนั้นชุมสายเชื่อมต่อภายในท้องถิ่นจะส่ง สัญญาณให้หมุน เพื่อให้ชุมสายท้องถิ่นส่งหมายเลขปลายทางมาให้ ส่วนขั้นตอนอื่น ๆ จะเหมือนการติดต่อภายในชุมสายเดียวกัน

2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ในโครงการที่จัดทำขึ้นใช้ไมโครคอนโทรลเลอร์ 8052 แต่เนื่องจากไมโครคอนโทรลเลอร์เบอร์ 8052 และไมโครคอนโทรลเลอร์เบอร์ 8051 เป็นไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เช่นเดียวกัน จึงมีคุณสมบัติ, โครงสร้าง, จำนวนพอร์ตและรายละเอียดการใช้งานของพอร์ตเหมือนกัน ดังนั้นในการนำเสนอรายละเอียดของไมโครคอนโทรลเลอร์ 8052 จะนำเสนอรายละเอียดของไมโครคอนโทรลเลอร์ 8051 แทน

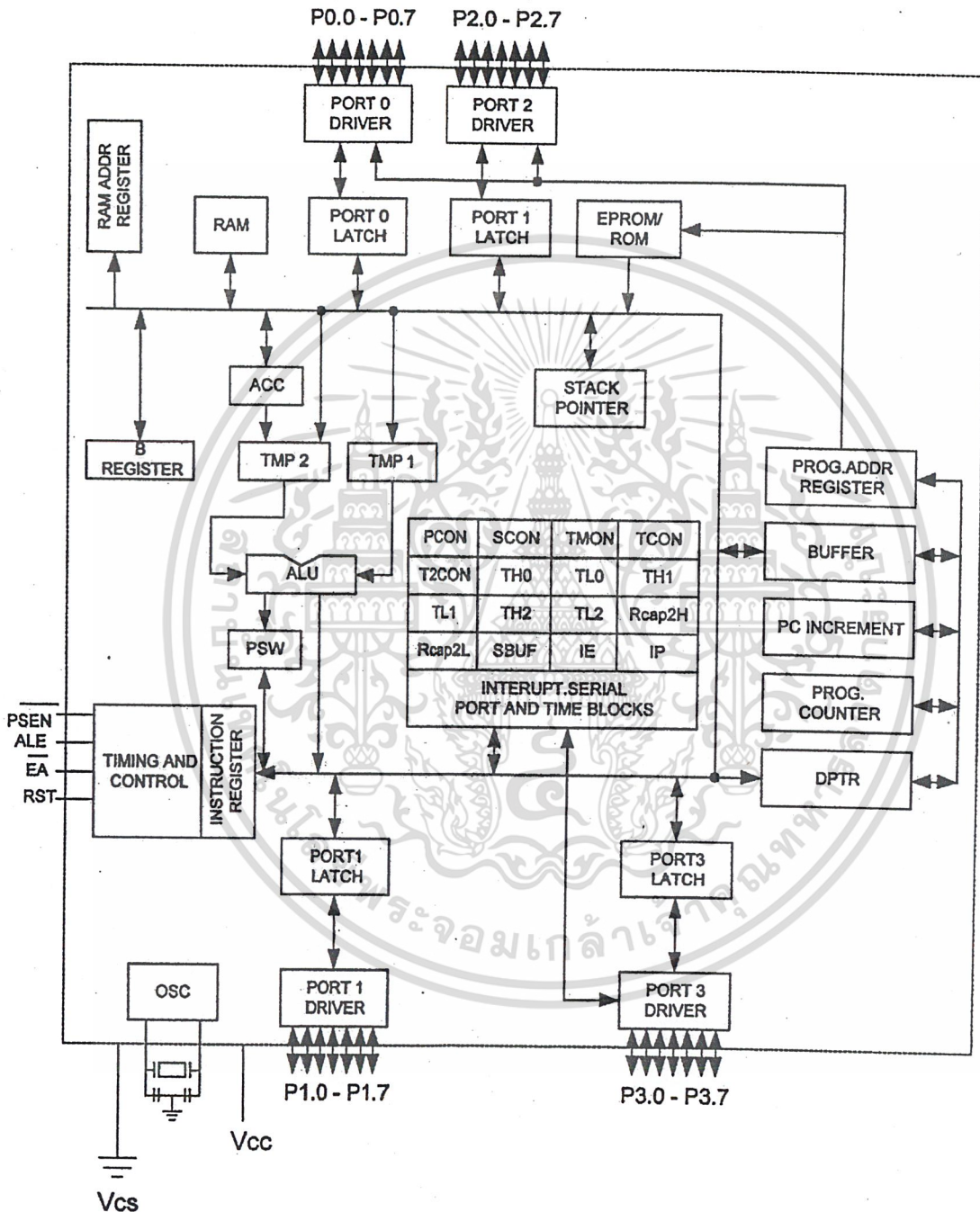
ไมโครคอนโทรลเลอร์ 8051 เป็นไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ของบริษัทอินเทล ถูกผลิตขึ้นมาให้มีคุณสมบัติเหมาะสมกับงานควบคุมระบบต่าง ๆ ในลักษณะที่เรียกว่า ไมโครคอนโทรลเลอร์ชิปเดี่ยว (Single Chip Microcontroller)

คุณสมบัติของ 8051

- เป็นไมโครโปรเซสเซอร์ขนาด 8 บิต สำหรับงานควบคุมระบบต่าง ๆ
- มีความสามารถประมวลผลของลอจิกระดับบิต
- มีขนาดหน่วยความจำสำหรับ โปรแกรมทำงาน ได้ถึง 64 กิโลไบต์ (Program Memory)
- มีขนาดหน่วยความจำสำหรับเก็บข้อมูลทำงาน ได้ถึง 64 กิโลไบต์ (Data Memory)
- มีหน่วยความจำสำหรับโปรแกรมภายในขนาด 4 กิโลไบต์
- มีหน่วยความจำข้อมูลภายในขนาด 128 กิโลไบต์
- มีพอร์ตสำหรับควบคุม 4 พอร์ต สามารถอ้างอิงพอร์ตได้ระดับบิตต่อบิต
- มีชุดไทม์เมอร์ (Timer) / ตัวนับ (Counter) ขนาด 16 บิต 2 ชุด
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ (full duplex UART)
- มีโครงสร้างอินเตอร์รัพท์จาก 6 แหล่งกำเนิดสัญญาณ และ 5 ตำแหน่งโปรแกรม
- มีวงจรออสซิลเลเตอร์ภายใน

โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051

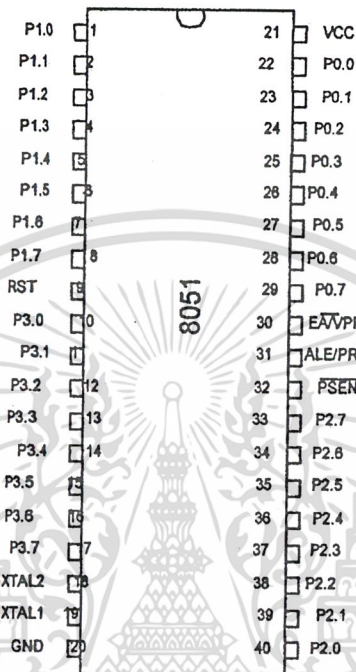
โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051 แสดงได้ดังรูปที่ 2.5



รูปที่ 2.5 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051

รายละเอียดของขาสัญญาณ 8051

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกัน ดังรูปที่ 2.6



รูปที่ 2.6 แสดง โครงสร้างขาของ 8051

อธิบายได้ดังนี้

Vcc (20) ขาแหล่งจ่ายไฟฟ้า (+5 V)

Vss (40) ขากราวนด

P0 (32-39) เป็นขาพอร์ต 0 ของ 8051 ที่มีขนาด 8 บิต ชนิดสองทิศทาง โดยแต่ละสัญญาณสามารถต่อพ่วงอุปกรณ์ TTL แบบ LS ได้ 8 ตัว และเป็นขาให้ สัญญาณมัลติเพล็กซ์ระหว่างสัญญาณข้อมูลกับ 8 บิตล่างของสัญญาณตำแหน่งในกรณีที่ใช้นหน่วยความจำภายนอก

P1 (1-8) เป็นขาพอร์ต 1 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasi-bidirectional โดยถ้าต้องการให้พอร์ตเส้นใดเป็นอินพุต จะให้ค่า "1" ที่บิตนั้น และสามารถต่อพ่วงกับอุปกรณ์ LS TTL ได้ 4 ตัว

P2 (21-28) เป็นขาพอร์ต 2 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasi-bidirectional เช่นเดียวกับพอร์ต 1 นอกจากนี้พอร์ต 2 ยังทำหน้าที่ให้สัญญาณตำแหน่ง 8 บิตบนในกรณีที่ใช้นหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำภายนอก ในกรณีอ้างตำแหน่งหน่วยความจำ 16 บิต ดังนั้นขณะที่ใช้หน่วยความจำภายนอกจะต้องไม่มีการเขียนข้อมูลใด ๆ ไปที่พอร์ต 2 จะทำให้เกิดความผิดพลาดในการทำงานได้

P3 (10-17) เป็นพอร์ต 3 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasi-bidirectional เช่นเดียวกับพอร์ต 1 และพอร์ต 2 แต่พอร์ต 3 นี้มีหน้าที่พิเศษดังตารางข้างล่าง ดังนั้นเมื่อมีการใช้สัญญาณดังกล่าว จึงไม่ควรเขียนข้อมูลไปที่พอร์ต 3 เพราะจะทำให้การทำงานของ 8051 ผิดพลาดได้

พอร์ต	หน้าที่พิเศษ
P3.0	RxD (สำหรับรับข้อมูลแบบอนุกรม)
P3.1	TxD (สำหรับส่งข้อมูลแบบอนุกรม)
P3.2	INT0 (ขาอินเตอร์รัพท์ภายนอก 0)
P3.3	INT1 (ขาอินเตอร์รัพท์ภายนอก 1)
P3.4	T0 (ขาพุทของ Timer0)
P3.5	T1 (ขาพุทของ Timer1)
P3.6	WR (สำหรับสัญญาณเขียนหน่วยความจำข้อมูลภายนอก)
P3.7	RD (สำหรับสัญญาณอ่านหน่วยความจำข้อมูลภายนอก)

ตารางที่ 1 แสดงหน้าที่พิเศษของพอร์ต 3

RST (9) ขาสัญญาณรีเซ็ตการทำงานของ 8051 โดยการให้ลอจิก “1” อย่างน้อย 2 ช่วงเมกซ์ซีน ไชเกิล

ALE / PROG (30) ขาสัญญาณออกของ Address Latch Enable สำหรับแลทช์ (Latch) ค่าตำแหน่ง 8 บิตที่ได้จากพอร์ต 0 โดยจะมีความถี่ออกมาที่ 1/6 ของความถี่อ้างอิงของ 8051

PSEN (29) ขาสัญญาณ Program Store Enable ใช้สำหรับหน่วยความจำภายนอก

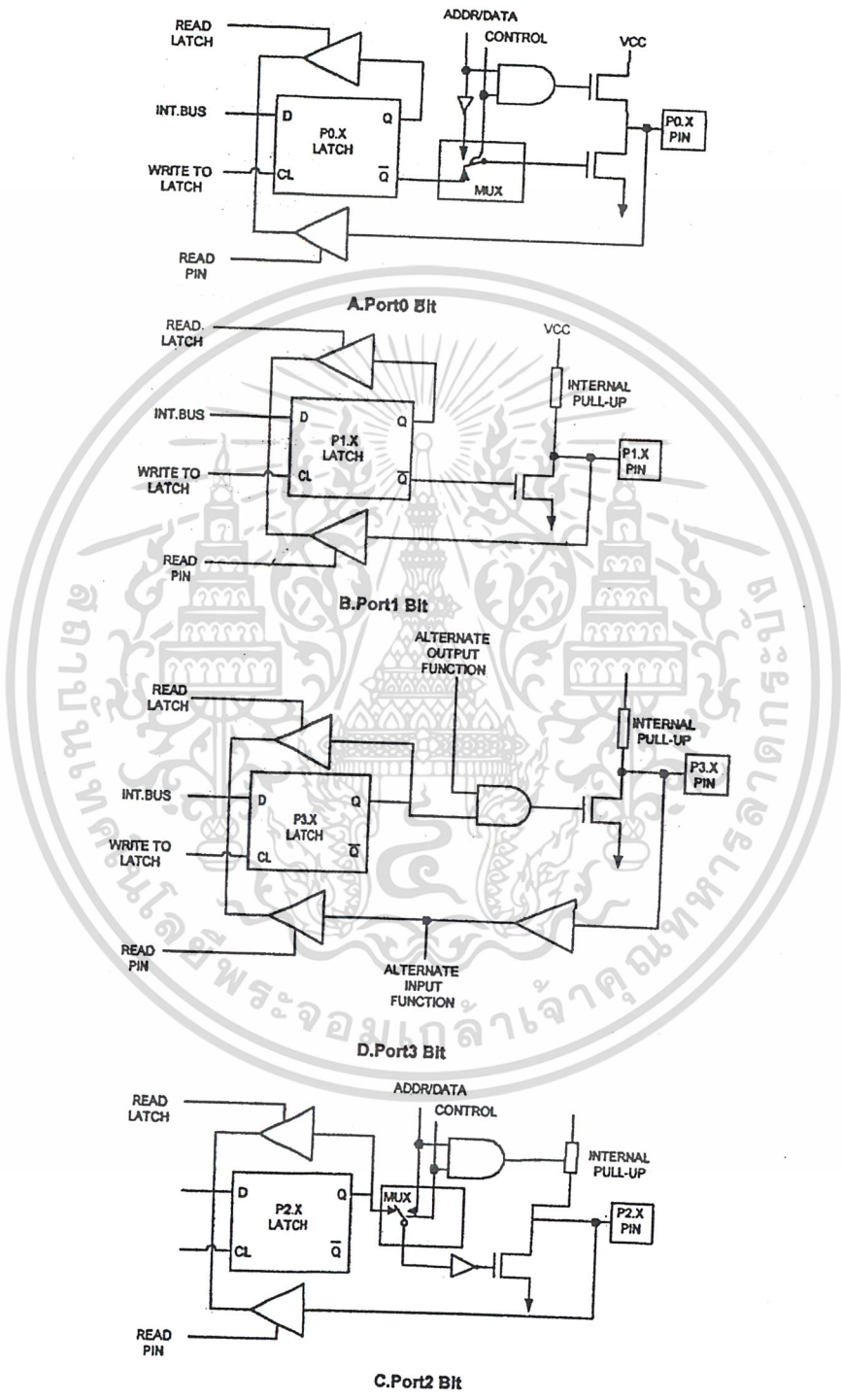
EA/Vpp (31) ขาสัญญาณ External Access Enable สำหรับกำหนดให้ 8051 อ่านหน่วยความจำโปรแกรมภายในหรือภายนอก โดยให้ลอจิก “1” จะเป็นอ่านหน่วยความจำภายใน ลอจิก “0” จะเป็นอ่านหน่วยความจำภายนอก 8051

XTAL1 (19) ขาเข้าของวงจรกำเนิดความถี่อ้างอิงภายใน 8051

XTAL2 (18) ขาออกของวงจรกำเนิดความถี่อ้างอิงภายใน 8051

โครงสร้างภายในของพอร์ตของ 8051

โครงสร้างภายในของพอร์ตของ 8051 แต่ละพอร์ตแสดงได้ดังรูปที่ 2.7



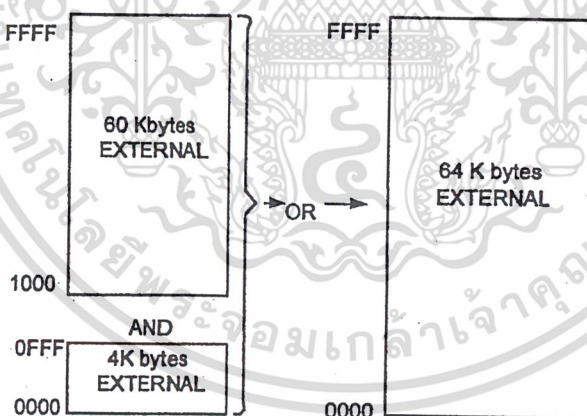
รูปที่ 2.7 แสดงโครงสร้างภายในพอร์ตอินพุต/เอาต์พุตของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะเห็นว่า พอร์ต 0 จะเป็นพอร์ตแบบ 2 ทิศทางอย่างแท้จริง โดยข้อมูลที่ส่งออกและอ่านเข้าจะไม่รบกวนกัน แต่โครงสร้างของพอร์ต 1, พอร์ต 2 และพอร์ต 3 จะเป็นเอาต์พุตให้เฟตเพียงตัวเดียว ดังนั้นข้อมูลที่ส่งออกโดยเฉพะลอจิก “0” จะรบกวนข้อมูลที่อ่านได้ ดังนั้นจึงไม่สามารถทำงานแบบ 2 ทิศทางได้อย่างแท้จริง โดยถ้าต้องการจะใช้บิตหนึ่งของพอร์ตดังกล่าวเป็นอินพุต จะต้องเขียนลอจิก “1” ไปที่บิตดังกล่าว

โครงสร้างหน่วยความจำของ 8051

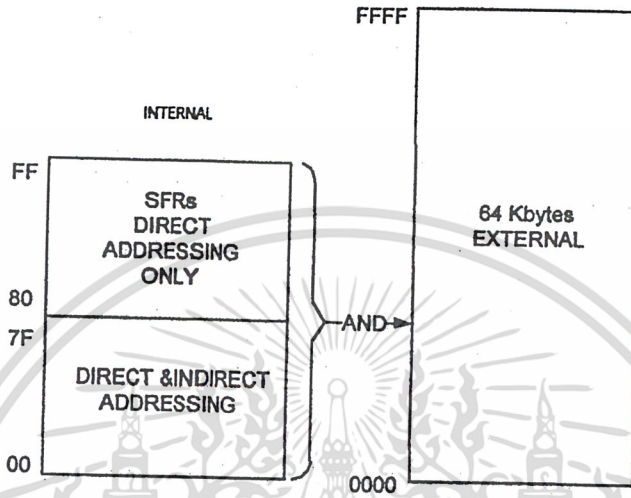
ไมโครคอนโทรลเลอร์ 8051 จะมีการแบ่งหน่วยความจำออกเป็น 2 ส่วนได้แก่ หน่วยความจำสำหรับโปรแกรมและหน่วยความจำสำหรับเก็บข้อมูล โดยที่ขนาดของแต่ละส่วนเท่ากับ 64 กิโลไบต์ ในส่วนของหน่วยความจำโปรแกรมและเป็นหน่วยความจำสำหรับอ่านเพียงอย่างเดียว โดยที่ 8051 จะใช้สัญญาณ PSEN ในการอ่านเท่านั้น แต่หน่วยความจำข้อมูลของ 8051 จะสามารถอ่านและเขียนโดยใช้สัญญาณ RD และ WR ตามลำดับ แต่อย่างไรก็ตามผู้ใช้สามารถรวมหน่วยความจำโปรแกรมและหน่วยความจำของข้อมูลเข้าด้วยกันได้ โดยนำสัญญาณ RD และ PSEN มา AND กัน สำหรับสร้างสัญญาณอ่านหน่วยความจำ นอกจากนี้หน่วยความจำดังกล่าวยังแบ่งออกเป็นหน่วยความจำภายนอกและภายในของ 8051 ดังแสดงในรูปที่ 2.8 และรูปที่ 2.9



รูปที่ 2.8 แสดงหน่วยความจำโปรแกรมของ 8051

จากรูปที่ 2.8 แสดงหน่วยความจำโปรแกรมในกรณีที่ใช้หน่วยความจำภายในหรือภายนอก ด้านซ้ายมือเป็นส่วนของหน่วยความจำโปรแกรมภายในที่มีขนาด 4 กิโลไบต์ ของ 8051 ส่วนที่เหลือจะเป็นหน่วยความจำภายนอก ส่วนด้านขวาแสดงหน่วยความจำโปรแกรมเมื่อเลือกให้ติดต่อกับหน่วยความจำภายนอกทั้งหมด

สำหรับหน่วยความจำข้อมูลของ 8051 สามารถแบ่งออกเป็นภายนอกและภายใน โดยหน่วยความจำภายนอกแสดงไว้ด้านขวามือของรูปที่ 2.9 ซึ่งมีขนาด 64 กิโลไบต์ ส่วนหน่วยความจำข้อมูลภายในแสดงไว้ด้านซ้ายมือของรูปที่ 2.9



รูปที่ 2.9 แสดงหน่วยความจำข้อมูลของ 8051

โดยหน่วยความจำของ 8051 แบ่งออกได้เป็น 2 ส่วน ได้แก่ส่วนหน่วยความจำข้อมูลที่สามารถอ้างอิงแบบโดยตรง (Direct) และโดยอ้อม (Indirect) ซึ่งมีขนาด 128 กิโลไบต์ กับหน่วยความจำที่อ้างอิงได้เฉพาะแบบโดยตรงเท่านั้น หรือในส่วนที่เรียกอีกแบบหนึ่งว่า รีจิสเตอร์หน้าที่พิเศษ (Special Function Register)

ในส่วนของหน่วยความจำข้อมูลภายในที่อ้างอิงแบบโดยตรงและโดยอ้อมนั้น จะสามารถแบ่งออกได้ 3 ส่วน ได้ดังรูปที่ 2.10 โดยมีรายละเอียดดังนี้

- Register Bank 0-3 เป็นส่วนที่อยู่ในตำแหน่งหน่วยความจำข้อมูลภายในตั้งแต่ 00H-1FH จำนวน 32 ไบต์ โดยจะแบ่งออกเป็นชุด ชุดละ 8 ไบต์ จำนวน 4 ชุด ซึ่งแต่ละชุดจะมีชื่อเรียกเป็น R0 ถึง R7 เป็นรีจิสเตอร์ที่ใช้งานร่วมในแต่ละคำสั่ง โดยเมื่อ 8051 ถูกรีเซ็ต Register Bank 0 จะถูกเลือก

- Bit Address Area เป็นส่วนที่มีขนาด 16 ไบต์ ที่ตำแหน่งหน่วยความจำข้อมูล 20H ถึง 2FH ในส่วนนี้สามารถที่จะอ้างอิงถึงข้อมูลได้เป็นระดับบิตถึง 128 บิต โดยการอ้างอิงตำแหน่งโดยตรงในลักษณะบิต ตั้งแต่ตำแหน่ง 00H ถึง 7FH

- Scratch Pad Area เป็นส่วนที่อยู่ตำแหน่งตั้งแต่ 30H ถึง 7FH ซึ่งเป็นบริเวณหน่วยความจำข้อมูลภายในเอนกประสงค์ที่ผู้ใช้สามารถใช้ได้โดยตรง นอกจากนี้ยังสามารถใช้หน่วยความจำข้อมูลบริเวณนี้สำหรับเก็บข้อมูลแบบสแต็ก (Stack) ได้ด้วย

4 กิโลไบต์ ที่สามารถใช้เครื่องคอมพิวเตอร์ทำการลบหรือเขียนโปรแกรมได้โดยตรงโดยไม่ต้องถอด MCU ออกจากการ์ดหรือแผงวงจรในลักษณะที่ว่า in system programming หรือจะใช้เครื่องโปรแกรม (Universal Programmer) โดยตรงก็ได้ ซึ่งสามารถโปรแกรมได้เป็นพันครั้ง ไมโครคอนโทรลเลอร์ตระกูล AT89C51 สามารถใช้งานร่วมและแทนกันได้กับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ได้แก่ เบอร์ 80C51 , 80C51/52 , 87C51/52 เป็นต้น ซึ่ง AT89C51 เหมือนกับตระกูลเหล่านี้ของอินเทลทำให้ด้านชุดคำสั่งและการจัดเรียงขา นั้นคือเราสามารถนำ AT89C51 มาใช้แทน MCS-51 ได้เลย

การขยายพอร์ตอินพุตหรือเอาต์ของไมโครคอนโทรลเลอร์ MCS-51 เพิ่มเติมโดยใช้ไอซี 8255

การขยายพอร์ตอินพุตหรือเอาต์ของไมโครคอนโทรลเลอร์ 8051 เพิ่มเติมนั้น นอกจากจะใช้ไอซีประเภทแลตซ์และบัฟเฟอร์ประกอบกันเข้ากับระบบบัสแล้ว ยังสามารถใช้ไอซีวงจรรวมความจุสูงเบอร์ 8255 ได้ ซึ่งสามารถทำหน้าที่ได้ทั้งพอร์ตอินพุตหรือเอาต์พุตด้วยโปรแกรม ทำให้มีความอ่อนตัวในการนำไปใช้งานตามความต้องการมากขึ้น 8255 ได้รับการออกแบบมาเพื่อทำหน้าที่เป็นพอร์ตสำหรับการรับ-ส่งข้อมูลแบบขนานระหว่างอุปกรณ์ภายนอกกับไมโครคอนโทรลเลอร์ สามารถเปลี่ยนแปลงลักษณะการทำงานของพอร์ตให้เป็นอินพุตหรือเอาต์พุตได้สะดวก

ลักษณะพื้นฐานของไอซี 8255

ไอซี 8255 ประกอบด้วยส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกโดยตรงผ่านสายสัญญาณที่ระบุชื่อ ว่า PA0-PA7, PB0-PB7 และ PC0-PC7 กลุ่มสัญญาณเหล่านี้จำแนกออกได้เป็น 3 กลุ่มคือ พอร์ต A (PA) พอร์ต B (PB) และพอร์ต C (PC) นอกจากนี้ Group A Control และ Group B Control ยังทำหน้าที่กำหนดการทำงานของพอร์ตทั้งสาม สองส่วนนี้เชื่อมต่อกับส่วนอื่น ๆ ผ่านทางบัสข้อมูลภายใน (internal bus) ของ 8255 สำหรับส่วนการทำงานที่มีชื่อว่า Data bus buffer และ Read/Write Control Logic ทำหน้าที่เชื่อมต่อระหว่างระบบบัสของไมโครคอนโทรลเลอร์กับ 8255 เพื่อรับส่งข้อมูลระหว่างกันตามระดับลอจิกของขาสัญญาณ RD และ WR ตามลำดับ

การจำแนกกลุ่มของพอร์ต 8255

จากรูปที่ 2.12 จะเห็นว่า 8255 มีพอร์ตจำนวน 3 พอร์ต คือ พอร์ต A พอร์ต B และพอร์ต C โดยพื้นฐานนั้นล้วนเป็นพอร์ตแบบขนานที่ประกอบด้วยสัญญาณ 8 เส้น ซึ่งแต่ละเส้นจะแทนบิตของข้อมูลพอร์ตซึ่งอาจกล่าวได้อีกลักษณะว่าเป็นพอร์ตแบบ 8 บิต นอกจากนี้ยังสามารถอ้างอิงถึงแต่ละบิตของเส้นสัญญาณพอร์ตนี้ได้โดยอิสระ อย่างไรก็ตาม 8255 นี้ก็ยังจัดกลุ่มของพอร์ตได้เป็นสองกลุ่มคือ กลุ่ม A และกลุ่ม B เพื่อประโยชน์ในการกำหนดรูปแบบการทำงานของพอร์ตดังตารางที่ 2

ชื่อกลุ่ม	ลักษณะ
GROUP A	พอร์ต A จำนวน 8 บิต (ทุกบิตของพอร์ต) พอร์ต C จำนวน 4 บิต (เฉพาะ 4 บิตบนของพอร์ต)
GROUP B	พอร์ต B จำนวน 8 บิต (ทุกบิตของพอร์ต) พอร์ต C จำนวน 4 บิต (เฉพาะ 4 บิตล่างของพอร์ต)

ตารางที่ 2 การกำหนดรูปแบบการทำงานของพอร์ต 8255

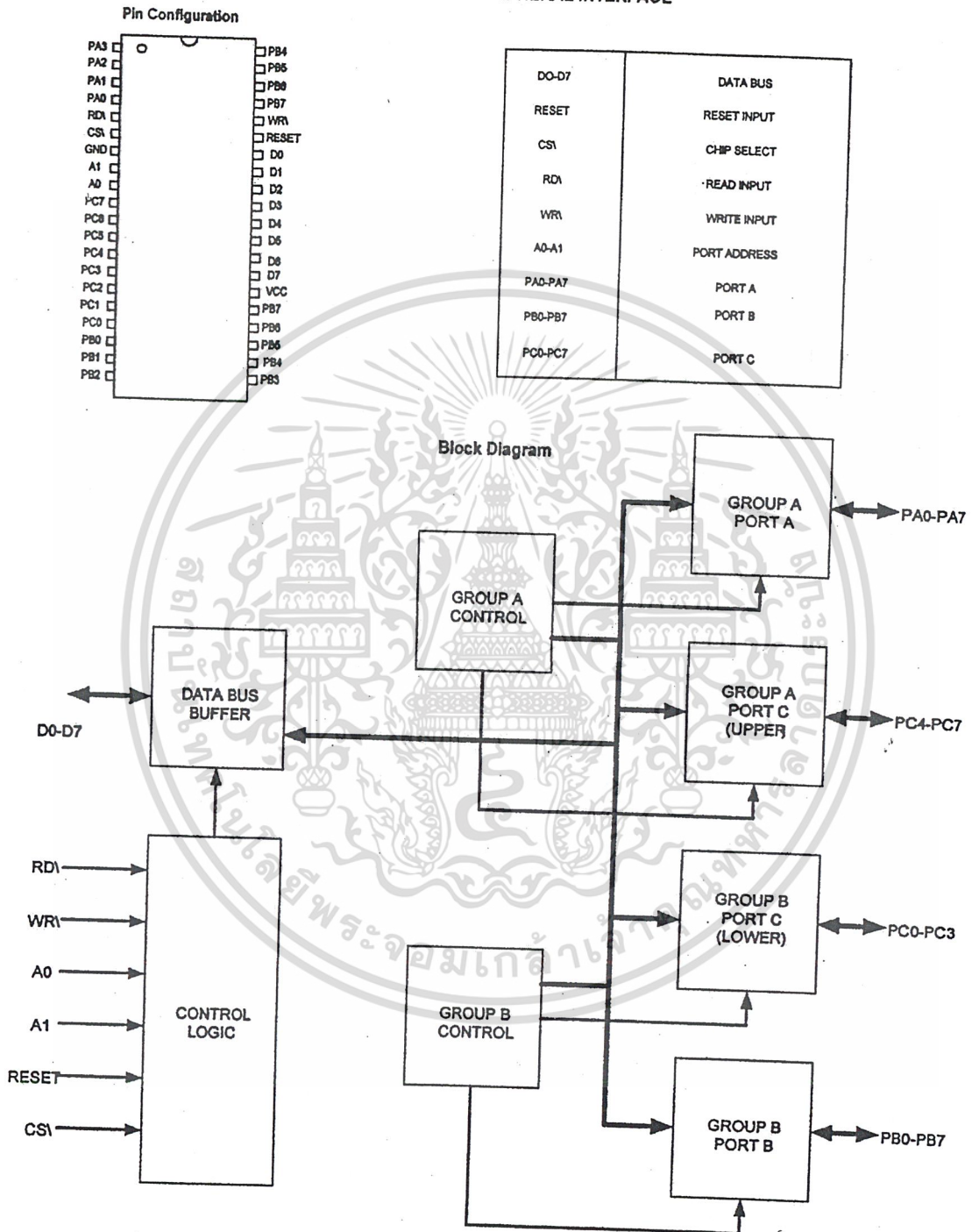
จากตารางที่ 2 จะเห็นว่าจำนวนเส้นสัญญาณทั้งหมดของพอร์ต C (PC0-PC7) ได้ถูกแยกออกเป็นกลุ่มคือ กลุ่มของ 4 บิตล่าง (lower nibble) จาก PC0-PC3 และกลุ่มของ 4 บิตบน (upper nibble) จาก PC4-PC7 ดังนั้นกลุ่ม A และกลุ่ม B ของ 8255 จึงมีจำนวนบิตในแต่ละกลุ่มเป็น 12 บิต

ส่วนหน้าที่การทำงานของขาต่างๆ แสดงได้ดังตารางที่ 3

ขาสัญญาณ	ความหมาย
D0-D7	กลุ่มของเส้นสัญญาณข้อมูลของ 8255 เมื่อมีการเขียนหรืออ่าน
CS	กลุ่มของสัญญาณเลือกอุปกรณ์ เมื่อขาของสัญญาณนี้เป็นระดับลอจิกต่ำ แสดงว่าสามารถเขียนหรืออ่านข้อมูลจาก 8255 ได้
RD	สัญญาณบอกสถานะว่าต้องการอ่านข้อมูลจากรีจิสเตอร์ของ 8255
WR	สัญญาณบอกสถานะว่าต้องการเขียนข้อมูลจากรีจิสเตอร์ของ 8255
A0-A1	สัญญาณระบุตำแหน่งรีจิสเตอร์ภายใน 8255 ที่ต้องการ
RESET	สัญญาณรีเซ็ตวงจรทำงานภายใน 8255 เพื่อเริ่มต้นใหม่
PA0-PA7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต A ของ 8255
PB0-PB7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต B ของ 8255
PC0-PC7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต C ของ 8255

ตารางที่ 3 แสดงหน้าที่การทำงานของขาสัญญาณไอซี 8255

8255
PROGRAMMABLE PERIPHERAL INTERFACE



รูปที่ 2.12 แสดงขาสัญญาณต่างๆ และโครงสร้างภายในของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อ 8255 กับ 8051

เมื่อพิจารณาจากรูปที่ 2.12 จะเห็นว่ามิชาสัญญาณระบุตำแหน่งจำนวน 2 เส้น คือ A0 และ A1 ทำให้มีจำนวนตำแหน่งที่จะอ้างถึงได้มีค่าเป็น 2^2 ตำแหน่ง หรือเท่ากับ 4 ตำแหน่งนั่นเอง ซึ่งแต่ละตำแหน่งมีความหมายถึงระบบรีจิสเตอร์หรือพอร์ตภายใน 8255 ดังตารางที่ 4

A1	A0	ชื่อรีจิสเตอร์
0	0	พอร์ตA
0	1	พอร์ตB
1	0	พอร์ตC
0	1	รีจิสเตอร์ควบคุม

ตารางที่ 4 ระบบรีจิสเตอร์ของ 8255

เมื่อพิจารณาค่าตำแหน่งเหล่านี้ร่วมกับระดับลอจิกของขาสัญญาณ RD / WR จะเป็นการอ่านหรือเขียนข้อมูลทางขาสัญญาณ D0-D7 ให้กับรีจิสเตอร์นั้นตามลำดับ ดังตารางที่ 5

RD	WR	A1	A2	ความหมาย
0	1	0	0	ส่ง (เขียน) ข้อมูลให้กับพอร์ตA
1	0	0	0	ส่ง (อ่าน) ข้อมูลให้กับพอร์ตA
0	1	0	1	ส่ง (เขียน) ข้อมูลให้กับพอร์ตB
1	0	0	1	ส่ง (อ่าน) ข้อมูลให้กับพอร์ตB
0	1	1	0	ส่ง (เขียน) ข้อมูลให้กับพอร์ตC
1	0	1	0	ส่ง (อ่าน) ข้อมูลให้กับพอร์ตC
0	1	1	1	ส่ง (เขียน) ข้อมูลให้กับพอร์ตควบคุม
1	0	1	1	เป็นสถานะที่ไม่ถูกต้อง

ตารางที่ 5 แสดงข้อมูลทางขาสัญญาณ

ดังนั้นโดยทั่วไปจึงมักกำหนดให้ตำแหน่งของ 8255 ทั้ง 4 ตำแหน่งนี้ อยู่ในตำแหน่งช่วงใดช่วงหนึ่งของระบบ เช่น 10h, 11h, 12h และ 13h โดยขาสัญญาณระบุตำแหน่งที่นอกเหนือไปจาก A0 และ A1 นำมาเข้ายังตัวถอดรหัสตำแหน่ง เพื่อสร้างสัญญาณเลือกอุปกรณ์ CS นี้ จะเป็นสภาวะลอจิก

“low” ก็ต่อเมื่อค่าบิตตำแหน่ง A0-A7 มีค่าเท่ากับ 0000100XX (ตัวอักษร XX ใช้ระบุถึงรีจิสเตอร์ภายใน 8255 เพื่อทำการอ่านหรือเขียนข้อมูล)

ดังนั้นตำแหน่งของรีจิสเตอร์ภายใน 8255 จะมีค่าตามตารางที่ 6

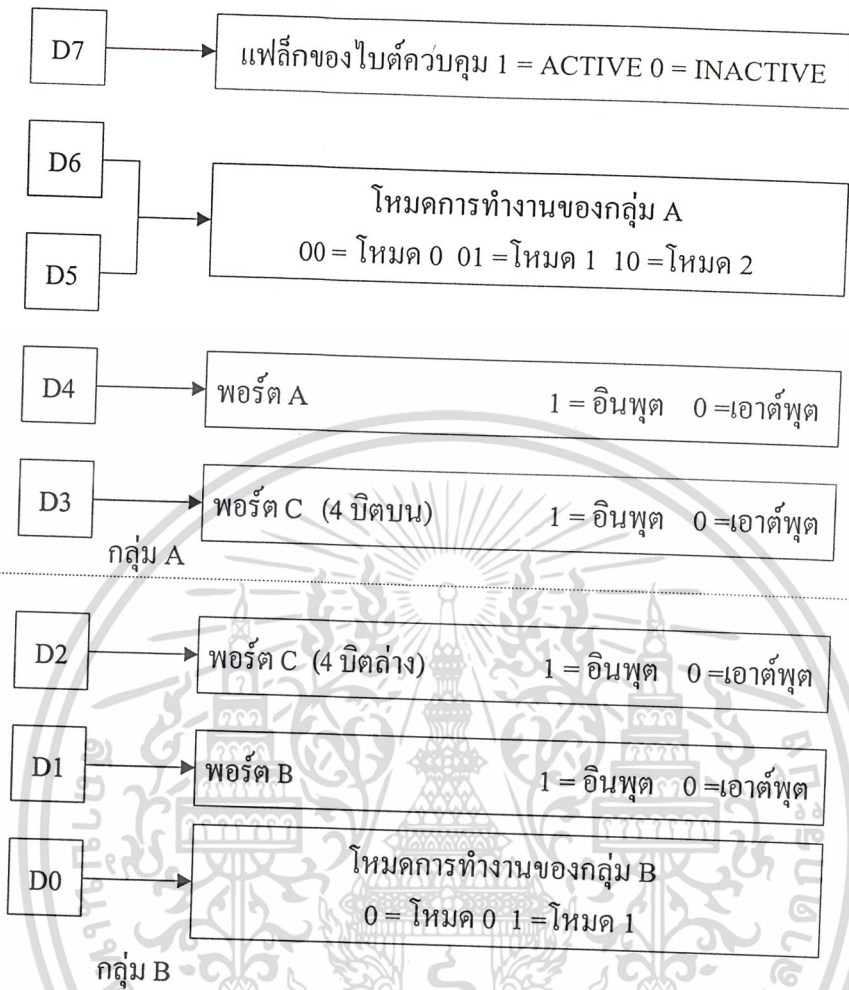
ตำแหน่ง	ความหมาย
10h	พอร์ตA
11h	พอร์ตB
12h	พอร์ตC
13h	รีจิสเตอร์ควบคุม

ตารางที่ 6 ตำแหน่งของรีจิสเตอร์

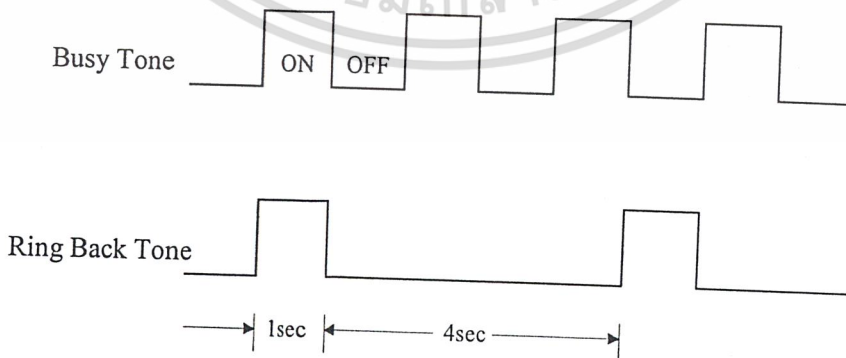
ขาสัญญาณอื่น ๆ เช่น RD และ WR มักจะเชื่อมต่อเข้ากับขาสัญญาณชื่อเดียวกับของ 8051 โดยตรง ทำให้ตำแหน่งของ 8255 อยู่ในพื้นที่หน่วยความจำข้อมูลของ 8051 สำหรับขาสัญญาณรีเซ็ตของ 8255 ซึ่งมีผลทำให้เกิดสภาวะการทำงานใหม่เมื่อระดับขาของสัญญาณเป็นลอจิก “high” ดังนั้นหากว่าขาสัญญาณการรีเซ็ตเดียวกับของ 8051 หากต้องการที่จะรีเซ็ต 8255 ด้วยก็สามารถต่อได้ตรงส่วนขาสัญญาณ D0-D7 ก็สามารถนำไปเชื่อมต่อโดยตรงเข้ากับสายสัญญาณของ 8051 ได้เช่นกัน ความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ 8255 แสดงได้ดังรูปที่ 2.13

2.3 ภาคตรวจจับสัญญาณเสียง

ภาคตรวจจับสัญญาณเสียงนี้ มีหน้าที่ตรวจจับสัญญาณเสียงต่าง ๆ ที่รับมาจากสายโทรศัพท์ สัญญาณที่รับเข้ามาจากสายโทรศัพท์จะประกอบไปด้วย สัญญาณเสียงพูด , สัญญาณเสียงเรียก(Ring Back Tone) และสัญญาณเสียงสายไม่ว่าง (Busy Tone) สัญญาณเสียงที่ถูกกำหนดโดย CCITT จะอยู่ระหว่างความถี่ 300 - 3,400 Hz และเสียง Ring Back Tone กับ Busy Tone จะมีความถี่ 400 - 450 Hz โดยที่ใช้เวลาในการส่งต่างกัน ในประเทศไทยโดยองค์การโทรศัพท์แห่งประเทศไทยเป็นผู้ให้บริการ ได้กำหนดให้สัญญาณ Ring Back Tone มีช่วงดัง 1 วินาที ช่วงหยุด 4 วินาที และสัญญาณ Busy Tone มีช่วงดัง 1 วินาที ช่วงหยุด 1 วินาที ดังรูปที่ 2.14



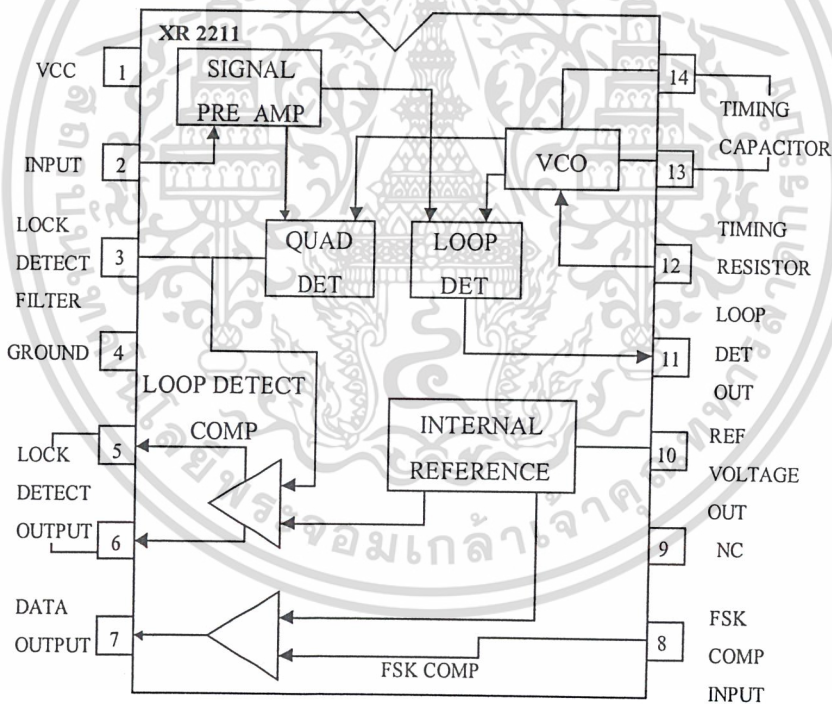
รูปที่ 2.13 แสดงความหมายของบิตภายในไบต์ข้อมูลคว่ำคุมสำหรับ 8255



รูปที่ 2.14 ลักษณะของสัญญาณโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในภาคตรวจจับสัญญาณเสียงนี้ใช้ IC เบอร์ XR-2211 เป็นตัวตรวจจับ เราสามารถกำหนดความถี่ในการตรวจจับได้โดยการกำหนดค่าตัวต้านทานและตัวเก็บประจุ โดยการคำนวณจากสูตรใน DATA SHEET โครงสร้างภายในของไอซี เบอร์ XR-2211 ดังแสดงตามรูปที่ 2.15 ไอซีเบอร์นี้ถูกออกแบบมาให้ใช้ทำหน้าที่เป็น FSK (Frequency Shift Keying) demodulator และ tone decoder โครงสร้างของไอซีประกอบไปด้วย input preamplifier ทำหน้าที่ขยายสัญญาณความถี่เสียงที่รับเข้ามาให้มีขนาดสัญญาณแรงขึ้น วงจรเฟสดีเทคเตอร์ (Phase detector) ทำหน้าที่เปรียบเทียบเฟสของสัญญาณ และวงจรโวลเตจคอนโทรลเลอร์ออสซิลเลเตอร์ (Voltage Controlled Oscillator:VCO) ทำหน้าที่ควบคุมการผลิตความถี่ด้วยแรงเคลื่อน โดยการกำหนดค่าความถี่ที่ตรวจจับแล้วนำไปคำนวณหาค่า R0, R1, C0, C1 ไอซีเบอร์ XR2211 จะให้เอาต์พุตเมื่อเฟสล็อกรูปทำการล็อกสัญญาณที่รับเข้ามาเมื่อเปรียบเทียบเฟสกับสัญญาณความถี่ที่ออกแบบไว้ตรงกันเอาต์พุตที่ได้จากขา 5 จะเป็นลอจิก “0” และเอาต์พุตที่ได้จากขา 6 จะเป็นลอจิก “1”



รูปที่ 2.15 โครงสร้างภายใน IC XR-2211

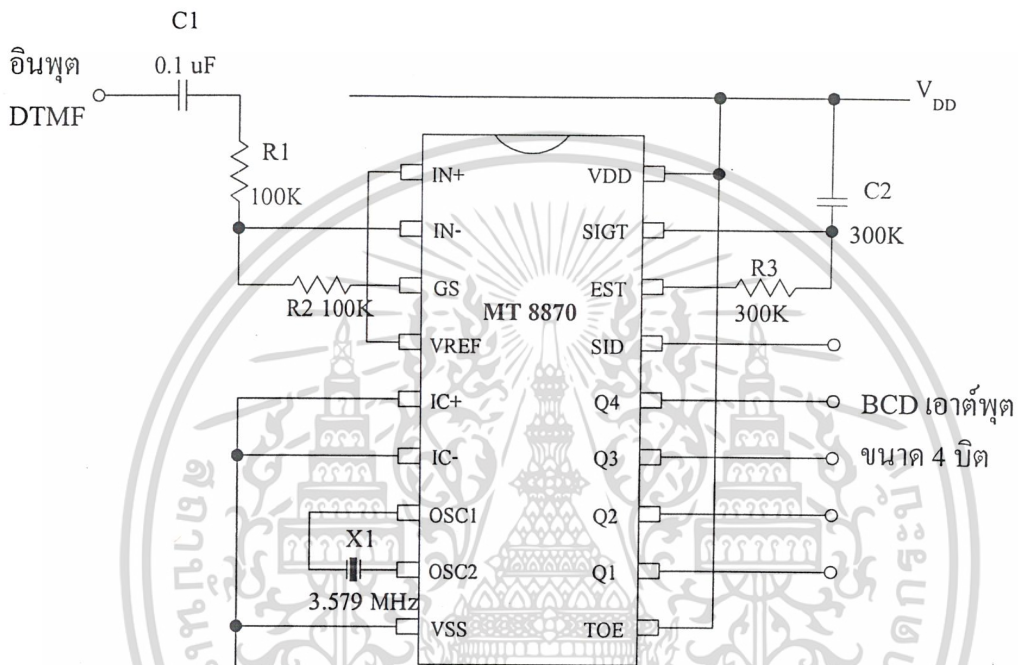
2.4 วงจรถอดรหัสเลขหมาย (DTMF Decode) โดยใช้ไอซี MT8870

ในยุคก่อนการออกแบบรหัสความถี่ของสัญญาณโทรศัพท์ มักใช้ไอซีเฟสล็อกลูป (Phase Lock Loop) ซึ่งสร้างปัญหามากมายเช่น ความถี่ที่เปลี่ยนแปลงไป การปรับแต่งวงจร ขนาดของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะต้องการใช้ไอซีจำนวนมาก จากปัญหาสารพัดเหล่านี้ ทำให้ไอซีเฟสลือกถูกไม่เป็นที่นิยมใช้กัน ในปัจจุบันนิยมใช้ไอซีเบอร์ MT 8870 มีคุณสมบัติที่ดีอยู่หลายประการ โดยการถอดรหัสจะกระทำเมื่อมีสัญญาณความถี่แบบ DTMF เข้าสู่วงจร DTMF Decode MT8870 จะทำการถอดรหัสเลขหมาย ซึ่งผลที่ออกมาจะเป็นสัญญาณดิจิทัล 4 บิต แล้วส่งให้ส่วนควบคุมสัญญาณต่อไป ซึ่ง

รายละเอียดของตัวอย่างวงจรถอดรหัสเลขหมายแสดงไว้ดังรูปที่ 2.16



รูปที่ 2.16 วงจรถอดรหัสเลขหมาย

ข้อกำหนดของวงจรถอดรหัสเลขหมายแบบดิจิทัลเอ็มเอฟ

1. วงจรจะยังคงถอดรหัสได้อย่างถูกต้อง ถึงแม้สัญญาณที่รับเข้ามาจะมีความเบี่ยงเบนไปจากค่าที่กำหนดไว้เป็นค่ามาตรฐาน แต่ต้องไม่เกินบวกลบ 2 เปอร์เซ็นต์
2. วงจรถอดรหัสจะทำการถอดรหัสได้ ก็ต่อเมื่อได้รับสัญญาณเข้ามามีระยะเวลาอย่างน้อย 40 มิลลิวินาที
3. วงจรถอดรหัสจะทำการถอดรหัสได้ถูกต้อง ก็ต่อเมื่อสัญญาณดิจิทัลเอ็มเอฟที่รับเข้ามาในวงจรจะต้องมีช่วงเวลาเท่ากับสัญญาณดิจิทัลเอ็มเอฟที่รับเข้ามาก่อนหน้านี้เป็นเวลาอย่างน้อย 35 มิลลิวินาที
4. วงจรถอดรหัสจะทำการถอดรหัสที่มีโคเนนามิกเรนจ์สูงกว่า 27.5 เดซิเบล ได้โดยไม่เกิดความผิดพลาด และยังสามารถทำงานได้ในกรณีที่มีสัญญาณ 2 ความถี่ ที่มีแอมพลิจูดแตกต่างกันมากกว่า 6 เดซิเบล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. วงจรถอดรหัสยังคงทำงานได้ตลอดเวลา ไม่ว่าจะขณะนั้นจะปรากฏเสียงพูดหรือมีสัญญาณรบกวนภายนอกเข้ามายังวงจรถอดรหัสก็จะไม่ทำให้การถอดรหัสผิดพลาด

คุณสมบัติของไอซีเบอร์ MT8870

- เป็นตัวรับและถอดรหัสความถี่ (DTMF Receiver)
- ใช้กำลังไฟฟ้าน้อย และใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถปรับการ์ดไทม์ (Guard time) ได้
- สามารถปรับอัตราขยายในตัวไอซีได้
- เป็นไอซีคุณภาพสูง

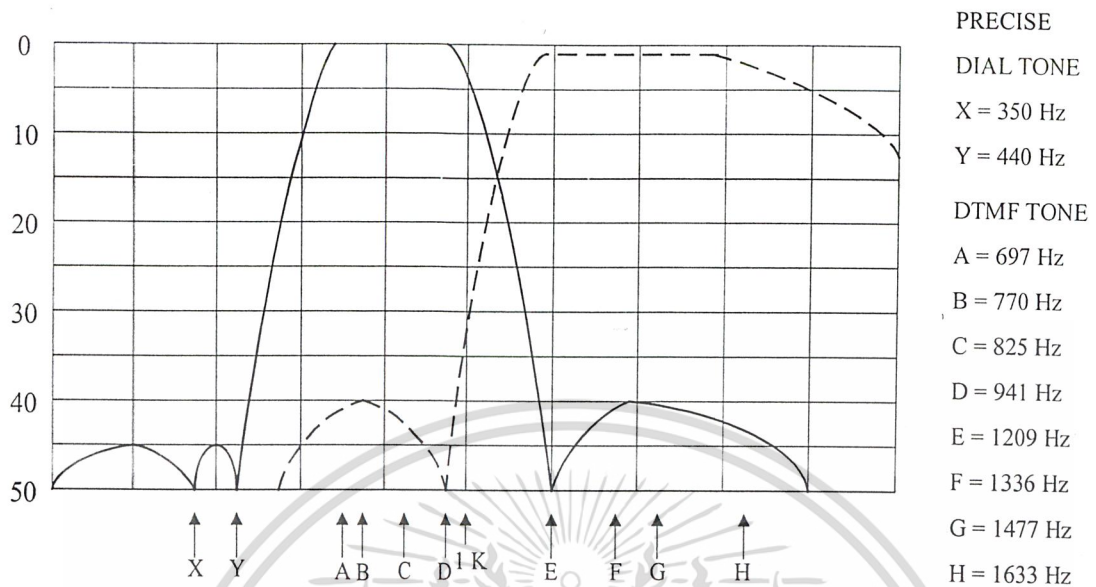
โครงสร้างของ ไอซีเบอร์ MT8870 (MTEL , 1997)

โครงสร้างภายในของ MT8870 ประกอบด้วยวงจรกรองความถี่และวงจรถอดรหัส เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO2-CMOS ในส่วนวงจรกรองความถี่ใช้เทคนิคสวิตช์คาปาซิเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิทัล เพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ ออกเป็นเลขฐานสองขนาด 4 บิต และเช็คช่วงเวลาที่ยืนยันเข้า ส่วนภาคอินพุตเป็นออปแอมป์ ซึ่งสามารถปรับอัตราขยายได้ โดยต่ออุปกรณ์ภายนอกเอาต์พุตเป็นวงจรแลตซ์ 3 สถานะ ภายในโครงสร้างของ MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน

- ภาคกรองความถี่ (Filter Section)
- ภาคถอดรหัส (Decoder Section)
- ภาคควบคุมตรวจสอบสัญญาณ (Steering Circuit)
- ภาคขยายสัญญาณความแตกต่าง (Differential Amplifier)
- ภาคกำเนิดความถี่ (Oscillator)

ภาคกรองความถี่ (Filter Section)

ในส่วนนี้ จะแยกสัญญาณดีทีเอ็มเอฟที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่ผ่านอันดับ 6 ชนิดใช้สวิตช์ตัวเก็บประจุ (Sixth order Capacitor Band Pass Filter) ซึ่งความถี่ที่แยกได้มี 2 ช่วง คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ แสดงได้ดังรูปที่ 2.17



รูปที่ 2.17 แสดงความถี่ที่ได้จากภาคกรองความถี่

ภาคถอดรหัส (Decoder Section)

ความถี่ที่เอ็มเอฟที่ถูกกรองเรียบร้อยแล้ว จะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลขโดยใช้เทคนิคของวงจรรีบแบบดิจิทัล และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐานดีทีเอ็มเอฟหรือไม่ เพื่อป้องกันความถี่อื่นที่จะเข้าผสม เมื่อตรวจสอบความถี่นั้นถูกต้อง สัญญาณที่ขา EST (Early Steering) หรือขา 16 ก็จะทำงาน

สำหรับค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ นั้น แสดงไว้ดังตารางที่ 7

F_{low}	F_{high}	On	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

ตารางที่ 7 แสดงค่ารหัสที่ได้จากความถี่ต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

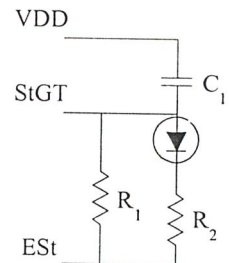
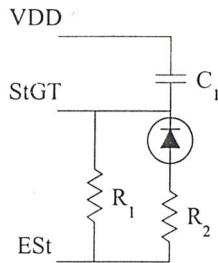
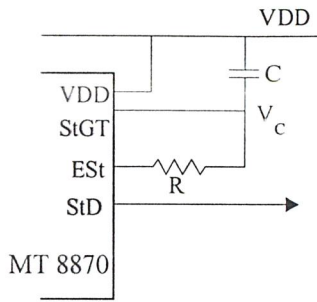
852	1447	9	H	1	0	0	1
941	1633	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1447	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	-	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

ตารางที่ 7 (ต่อ) แสดงการรหัสที่ได้จากความถี่ต่าง ๆ

ภาคควบคุมตรวจสอบสัญญาณ (Steering Circuit)

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ที่มีมีความถี่ออกมาเป็นช่วงเวลานานพอสมควร มิฉะนั้นวงจรส่วนจะไม่รับและถือว่าสัญญาณนั้น ไม่ถูกต้องส่วนช่วงเวลายาวเท่าใดนั้นสามารถตั้งได้โดยใช้ ตัวต้านทานและตัวเก็บประจุ (R, C) ต่อจากภายนอกตามภาพที่ 2.18 สัญญาณที่ขา EST จะเป็นระดับสูงนานใกล้เคียงกับระยะเวลาดีที่เอ็มเอฟเข้ามาทำให้ V_C สูงขึ้นตัวเก็บประจุ V_C จะคายประจุ ทำให้แรงดัน V_C สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสจึงจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิต รายละเอียดการทำงาน ดูได้จากแผนภูมิเวลา (Timing Diagram) จะเข้าใจได้ง่าย

สำหรับการ์ดใหม่ (Guardtime) หมายถึง ช่วงเวลาของความถี่ที่เข้ามาซึ่งจะต้องนานกว่าหรือเวลาที่เรที่ตั้งไว้ จึงจะได้รับการยอมรับสัญญาณความถี่นั้นถูกต้อง หรือพูดได้ว่าเวลาที่เรที่ตั้งไว้โดยตัวต้านทานและตัวเก็บประจุคือการ์ดใหม่นั้นเอง ถ้าสัญญาณความถี่เข้ามานานเท่ากับหรือมากกว่าเวลาที่ตั้งไว้จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่ที่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสสัญญาณตัวเลขออกมา



$$t_{GTA} = (RC)\ln(VDD/V_{tst})$$

$$t_{GTP} = (RC)\ln[VDD/(VDD-V_{tst})]$$

$$t_{GTP} = (RpC1)\ln[VDD/(VDD-V_{tst})]$$

$$t_{GTA} = (R1C1)\ln(VDD/V_{tst})$$

$$Rp = R1/R2$$

$$t_{GTP} = (R1C1)\ln[VDD/(VDD-V_{tst})]$$

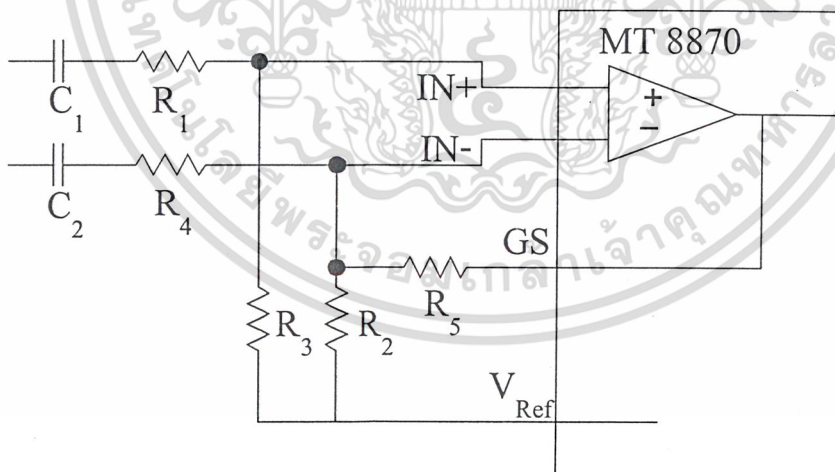
$$t_{GTA} = (RpC1)\ln(VDD/V_{tst})$$

$$Rp = R1/R2$$

รูปที่ 2.18 แสดงการต่อวงจรควบคุมตรวจสอบสัญญาณเบื้องต้นของไอซี MT8870

ภาคขยายสัญญาณความแตกต่าง (Differential Amplifier)

วงจรส่วนอินพุตของ MT8870 เป็นภาคขยายออปแอมป์ ที่สามารถปรับอัตราขยายได้ โดยต่อวงจรภายนอกเพิ่มเข้าดังภาพที่ ซึ่งสามารถคำนวณอัตราขยายความแตกต่างอินพุตและ อิมพีแดนซ์ ได้ดังนี้



รูปที่ 2.19 การต่อวงจรภาคอินพุตของไอซี MT8870

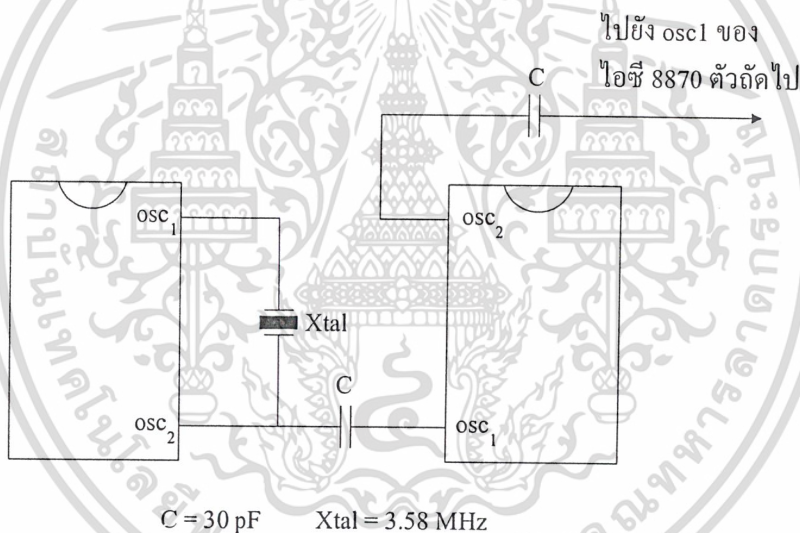
จากรูปที่ 2.19 สามารถคำนวณอัตราขยายความแตกต่างของอินพุตและอิมพีแดนซ์ได้ดังนี้

$$\text{อัตราขยาย } (A_{V \text{ diff}}) = \frac{R_5}{R_1}$$

$$\text{อินพุตอิมพีแดนซ์ } (Z_{IN \text{ diff}}) = 2(R_1^2 + \left(\frac{1}{\omega_c}\right)^2)^{1/2}$$

ภาคกำเนิดความถี่ (Oscillator)

ในภาคนี้อยู่ในไอซี MT8870 จะมีวงจรเวลาอยู่ภายใน เพียงแค่ต่อแค่คริสตอลขนาด 3.579 เมกะเฮิร์ตซ์ ก็จะสามารถใช้งานได้ทันที การต่อวงจรกำเนิดความถี่แสดงในรูปที่ 2.20



รูปที่ 2.20 การต่อวงจรผลิตความถี่

เพื่อให้ทราบถึงรายละเอียดต่างๆ ในการทำงานและหน้าที่ของขาต่างๆ ของ MT 8870 ในการต่อเข้าเป็นระบบเชื่อมโยงกับส่วนต่างๆ สามารถที่อธิบายด้วยไดอะแกรมสัญญาณ ดังรูปที่ 2.21

อธิบายศัพท์

Vin สัญญาณความถี่สี่เหลี่ยม

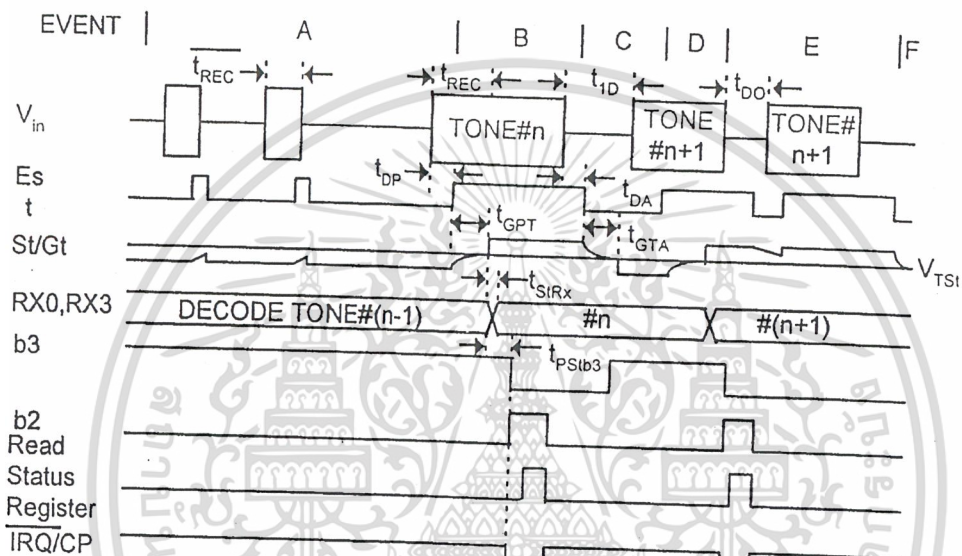
Est Early Steering Output ใช้แสดงความถี่ที่ถูกต้อง

St/GT SteeringInput / GuardTimeOutputสำหรับต่อกับ RC ภายนอก

Q1-Q4 เอาต์พุต BCD ขนาด 4 บิต

STD Delay Steering Output ใช้แสดงค่าความถี่ที่ได้รับ หรือหายไป
มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

TOE Tone Output Enable (input) ใช้ควบคุม Q1-Q4 ให้เป็น
อิมพีแดนซ์สูง ๆ



รูปที่ 2.21 ไคอะแกรมสัญญาณต่างๆ ของไอซี MT8870

2.5 วงจรถอดรหัสคีย์แพด (16 keypad decoder) โดยใช้ไอซี 74C922

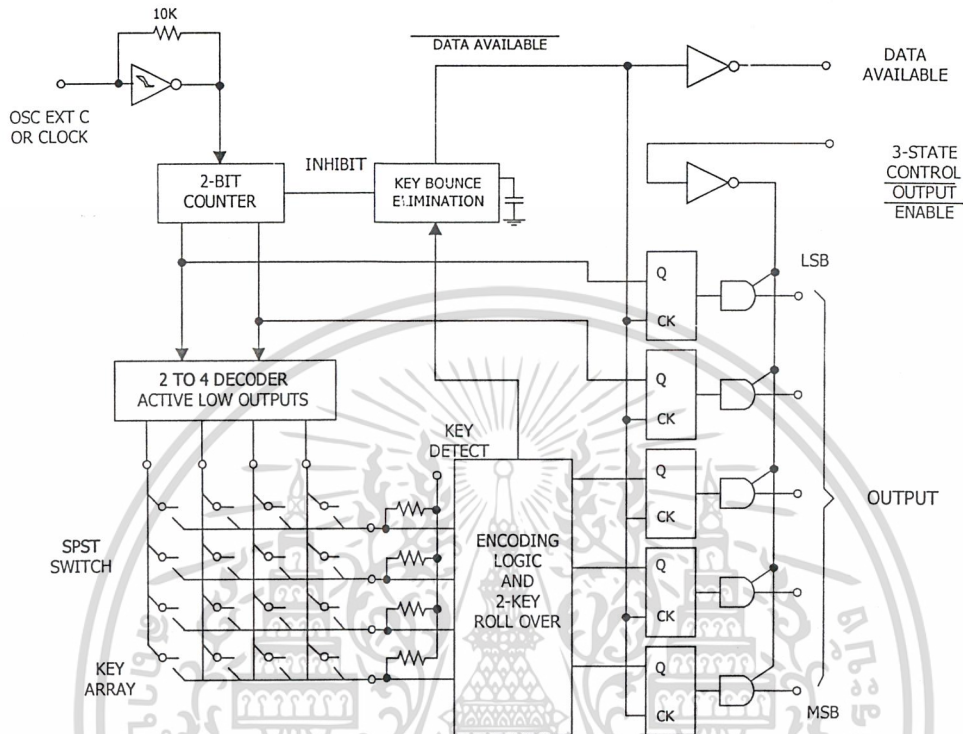
ในการ scankey เราสามารถทำได้ ในทางด้านซอฟต์แวร์และฮาร์ดแวร์ ทางด้านซอฟต์แวร์คือการตรวจโดยการเขียน โปรแกรม ทางด้านฮาร์ดแวร์นี้เราสามารถใช้อิซีตระกูล TTL เบอร์ 74C922 ซึ่งมีคุณสมบัติในการถอดรหัส key matrix 4x4 หรือ 16 keypad ซึ่งเอาท์พุทที่ได้จากไอซีจะเป็นสัญญาณดิจิทัล 4 bit

คุณสมบัติของไอซี 74C922

- เป็นตัวถอดรหัส keypad 4x4
- ใช้กำลังไฟน้อย ใช้ไฟเลี้ยงได้ตั้งแต่ 3 – 15 โวลต์
- มี pull-up ที่แถว
- สามารถแก้ไข key bounce ได้โดยตัวเก็บประจุตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างภายในไอซี 74C922



รูปที่ 2.22 โครงสร้างภายในของไอซี 74C922

ในไอซี 74C922 แบ่งเป็น 3 ส่วนใหญ่ ๆ คือ

- วงจรถอดรหัสคีย์
- วงจรภาคเอาต์พุต
- วงจรภาคkey bounce

วงจรถอดรหัสคีย์

วงจรถอดรหัสคีย์ จะประกอบด้วย ภาค row scan ซึ่งจะมีค่า internal pull up Encoding Logic และ 2 key roll over เพื่อจะได้เอาต์พุตทางด้าน row ไปให้แก่วงจรภาคเอาต์พุต ส่วนในภาค scan coloum จะถูกควบคุมโดย osc ทางอินพุตจะเป็นแบบ schmitt trigger osc ประกอบด้วย C_{ext} ที่ต่ออยู่กับขา 5 ของไอซี และ วงจร 2-bit counter และ 2-4 bit decode

วงจรรากเอาต์พุต

วงจรรากเอาต์พุตจะประกอบด้วย D-flipflop ทำหน้าที่ latch data output ให้มีสถานะค้าง สถานะเอาต์พุตเก่าอยู่จนกว่าจะมีการกดคีย์ เอาต์พุตจึงเปลี่ยนไปตามคีย์ที่กด

วงจรราก keybounce

วงจรราก keybounce จะมีการต่อ C_{ext} ที่ขา 6 เพื่อใช้ตั้งค่าคาบเวลาในการแก้ไขปัญหา bounce

2.6 การขับโมดูลแสดงผลแบบผลึกเหลว (LCD MODULE)

ในปัจจุบันแอลซีดี (LCD:Liquid Crystal Display) เป็นที่นิยมใช้งานเพื่อเป็นส่วนแสดงผลใน อุปกรณ์และเครื่องใช้งานต่าง ๆ เป็นอย่างมาก ทั้งนี้เนื่องจากมีความเหมาะสมหลาย ๆ ด้าน เช่น ใช้ กระแสไฟต่ำ สามารถแสดงผลเป็นตัวอักษร แสดงผลเป็นตัวเลข หรือแสดงผลเป็นกราฟฟิก (graphic) ได้ ส่วนที่ยังมีปัญหาอยู่บ้างก็คือเรื่องวงจร เพราะแอลซีดีมีระบบการทำงานที่ซับซ้อนและหาอุปกรณ์ ในวงจรได้ค่อนข้างยาก แต่ในขณะนี้ผู้ผลิตแอลซีดีจะผลิตแอลซีดีรุ่นที่เป็นโมดูลออกมา คือเป็น โมดูล ที่มีตัวแอลซีดีและวงจรควบคุมมาให้พร้อม ซึ่งสามารถทำให้ผู้ใช้สามารถต่อแอลซีดีเข้ากับ ไมโคร คอนโทรลเลอร์ได้ง่ายและสะดวกสำหรับการโปรแกรมน ทำให้ผู้ใช้งานด้าน ไมโครคอนโทรลเลอร์หัน มาใช้แผงแสดงผลด้วย แอลซีดีโมดูลกันมากขึ้น

แอลซีดีโมดูลมีอยู่หลายรุ่นและมีคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลักคือ

1. แบบดอตเมทริกซ์ (Dot Matrix) จะแสดงผลเป็นตัวอักษรขนาด 5×8 จุด มีจำนวนอักษร และจำนวนบรรทัดแตกต่างกันในแต่ละรุ่น

2. แบบกราฟฟิก (Graphic) จะสามารถแสดงผลในแบบบิตแมพ (Bit map) คือจะสร้างเป็น ภาพใด ๆ ก็ได้ตามต้องการ

แนวทางในการใช้งานของทั้งสองแบบจะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้ แบบดอตเมทริกซ์มากกว่า เนื่องจากราคาถูกและเพียงพอกับการใช้งานส่วนใหญ่

คุณสมบัติของดอตเมทริกซ์

คุณสมบัติของดอตเมทริกซ์แอลซีดีโมดูล สามารถสรุปเป็นข้อ ๆ ได้ดังนี้

1. มีให้เลือกหลายรุ่นตามความต้องการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วยดอตเมทริกซ์ขนาด 5×8 จุด
3. สามารถต่อ ไมโครคอนโทรลเลอร์เข้ากับระบบได้ 2 ลักษณะคือ ต่อแบบ memory map และ แบบผ่านพอร์ตของไอซี 8255 ซึ่งจะใช้ขาสัญญาณทั้งหมด 14 ขา
4. ใช้งานง่ายและสะดวก เพียงให้ระบบไมโครคอนโทรลเลอร์ส่งข้อมูลให้กับแอลซีดีโมดูล เท่านั้น ข้อความก็จะปรากฏบนแผงแสดงผล และจะค้างไว้ตลอด

5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากเช่น CLEAR, DISPLAY, HOME, CHARACTER เป็นต้น
6. สามารถแสดงผลเป็นเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดตัวอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสไฟน้อย มีน้ำหนักรวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์เท่านั้น

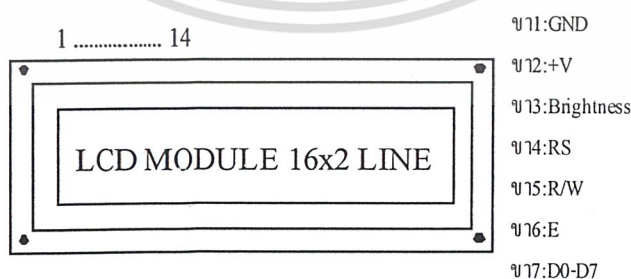
ในแอลซีดีโมดูลจะประกอบด้วยส่วนหลัก ๆ 3 ส่วน ดังนี้

1. ตัวแสดงผล (display) ภายในจะเป็นผลึกเหลวที่สามารถแสดงผลให้เห็น โดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอแอลซีดี
2. ตัวควบคุม (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของแอลซีดีโมดูล เช่น ควบคุมภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะ ชิพที่นิยมใช้คือเบอร์ HD44780 สำหรับควบคุมแอลซีดีแบบอักขระ และ HD61830 ใช้ควบคุมแอลซีดีแบบกราฟฟิก
3. ตัวขับ (driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลตามที่กำหนด ชิพที่ใช้ทำหน้าที่เป็นตัวขับได้แก่เบอร์ HD44100H และ MSM5259 เป็นต้น

ขาสัญญาณของแอลซีดีโมดูล

แอลซีดีโมดูลที่เลือกใช้ในโครงการเป็นแบบ 16 ตัวอักษร 2 บรรทัด เพราะเป็นแอลซีดีโมดูลที่มีโครงสร้างเป็นมาตรฐาน มีผู้ผลิตหลายราย และมีการระบุเบอร์แตกต่างกันออกไปตามผู้ผลิต เช่น LM020L ของฮิตาชิ, DMC-16117A ของคอปเท็กซ์ เป็นต้น แต่อย่างไรก็ตามคอนโทรลเลอร์ที่ใช้จะเป็นเบอร์เดียวกัน คือ HD44780 ของฮิตาชิ

รูปร่างและการจัดขาสัญญาณของแอลซีดีโมดูลแสดงได้ดังรูปที่ 2.23



รูปที่ 2.23 รูปร่างและการจัดขาสัญญาณของแอลซีดีโมดูล

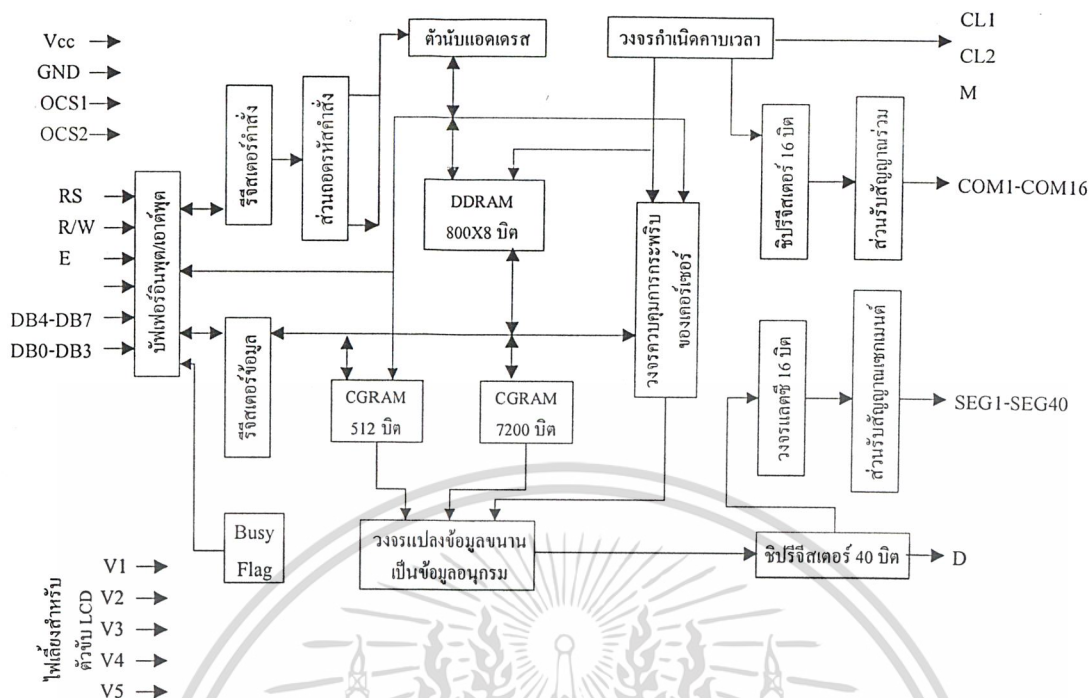
แอลซีดีโมดูลขนาด 16×2 มีขาสัญญาณที่ต่อใช้งานทั้งสิ้น 14 ขา รายละเอียดของขาสัญญาณของแอลซีดีโมดูล แสดงได้ดังตารางที่ 8

ขา	สัญลักษณ์	ระดับ	หน้าที่
1	V _{SS}	-	0 V GND
2	V _{CC}	-	+5 V POWER SUPPLY
3	V _{EE}	-	+V FOR LIQUID CRYSTAL DRIVE
4	RS	H/L	REGISTER SELECT H: data input L: introduction input H: data read L: data write
5	R/W	H/L	Enable signal
6	E	H	Data Bit bit 1
7	DB 0	H/L	Data Bit bit 2
8	DB 1	H/L	Data Bit bit 3
9	DB 2	H/L	Data Bit bit 4
10	DB 3	H/L	Data Bit bit 5
11	DB 4	H/L	Data Bit bit 6
12	DB 5	H/L	Data Bit bit 7
13	DB 6	H/L	Data Bit bit 8
14	DB 7	H/L	Data Bit bit 9

ตารางที่ 8 แสดงขาสัญญาณของแอลซีดีโมดูล

โครงสร้างภายในตัวควบคุมแอลซีดีโมดูล

ในการใช้งานแอลซีดีโมดูล จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมก่อน โดยแอลซีดีโมดูลแบบอักษระสามารถเข้าใจได้ง่าย โครงสร้างภายในแอลซีดีโมดูลแบบอักษระแสดงได้ดังรูปที่ 2.24 เป็นบล็อกไดอะแกรมภายในชิปควบคุมเบอร์ HD44780 ซึ่งใช้ในแอลซีดีโมดูลแบบอักษระ



รูปที่ 2.24 ไคอะแกรมการทำงานของแอลซีดีโมดูลแบบอักษร

โครงสร้างภายในแอลซีดีโมดูลแบบอักษรประกอบไปด้วย

1. บัฟเฟอร์อินพุต/เอาต์พุต (I/O Buffer) เป็นที่ใช้ในการติดต่อรับส่งกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม
2. รีจิสเตอร์คำสั่ง (Instruction Register:IR) เป็นรีจิสเตอร์ที่ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมการแสดงผล
3. รีจิสเตอร์ข้อมูล (Data Register:DR) เป็นรีจิสเตอร์ที่ใช้รับข้อมูลจากอุปกรณ์ภายนอก เพื่อถ่ายทอดไปยังหน่วยความจำ ที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร
4. แรมเก็บข้อมูลแสดงผล (Display Data RAM:DDRAM) เป็นหน่วยความจำที่ทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ข้อมูล ตัวควบคุมจะนำข้อมูลในแรมเก็บข้อมูลแสดงผลนี้ไปเปิดตารางของตัวอักษรที่เก็บไว้ในหน่วยความจำรวมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล
5. รมเก็บตัวอักษร (Character Generator ROM:CGROM) เป็นหน่วยความจำรวมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลในแรมเก็บข้อมูลแสดงผล
6. แรมเก็บตัวอักษร (Character Generator RAM:CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บตัวอักษรที่มีการสร้างเพิ่มเติมเข้าไปใหม่ในกรณีที่ตัวอักษรในรมเก็บตัวอักษรไม่เพียงพอ มีขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

512 บิต การเขียนและการอ่านค่าไปใช้นั้นทำเช่นเดียวกับรอมเก็บตัวอักษร คือเขียนข้อมูลลงในแรมเก็บข้อมูลแสดงผล แล้วตัวควบคุมจะมาอ่านข้อมูลจากแรมเก็บตัวอักษรเอง

7. แฟล็ก Busy เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งได้หรือไม่ ดังนั้นก่อนส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟล็ก busy นี้เสียก่อน

การต่อแอลซีดีโมดูลเข้ากับไมโครคอนโทรลเลอร์

แอลซีดีโมดูลสามารถต่อไมโครคอนโทรลเลอร์เข้ากับระบบได้ 2 ลักษณะคือ ต่อแบบเมมโมรีแมพ (memory map) และแบบผ่านพอร์ตของไอซี 8255 ในโครงงานนี้จะใช้การต่อแอลซีดีโมดูลแบบอินพุต/เอาต์พุตพอร์ต ซึ่งมีลักษณะดังนี้

1. สามารถต่อเข้ากับอินพุต/เอาต์พุตพอร์ตใด ๆ ก็ได้โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมาให้ตรงกับแอลซีดีโมดูล
2. ผู้ใช้จะเขียนข้อมูลให้แอลซีดีโมดูลได้อย่างเดียว ซึ่งผู้ใช้ควรที่จะกำหนดหน่วยความจำส่วนหนึ่งให้เสมือนว่าเป็นบัฟเฟอร์ให้กับแอลซีดีโมดูล
3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบเองเพื่อรอให้แอลซีดีโมดูลกระทำกระบวนการต่าง ๆ
4. ใช้ได้กับแผงวงจรทั่ว ๆ ไปที่มีพอร์ต
5. ไม่เปลืองหน่วยความจำในการใช้งาน
6. การจัดหาสัญญาณกระทำได้อย่างอิสระ

ชุดคำสั่งควบคุมและการแสดงข้อความ

ขาสัญญาณ V_{EE} มีไว้สำหรับกำหนดความเข้มของตัวอักษร โดยถ้าต่อกับ GND จะมีความเข้มของตัวอักษรสูงสุด แต่ถ้าต่อกับ V_{CC} จะมีความเข้มต่ำที่สุด ปกติแล้วแอลซีดีรุ่นธรรมดาอาจต่อกับ GND ไว้เลยก็ได้ไม่จำเป็นต้องใส่ตัวต้านทานปรับค่าได้ แต่ถ้าเป็นรุ่น STN (มุมมองกว้าง) ให้ใช้ตัวต้านทาน 2 กิโลโอห์ม ต่อดึงกราวด์อีกที เพื่อให้ความเข้มมีความเหมาะสม การเขียนหรืออ่านข้อมูลกับแอลซีดีโมดูลจะเป็นการกำหนดคุณสมบัติต่าง ๆ ในการใช้งานของแอลซีดีตามชุดคำสั่งควบคุมและรวมไปถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏลงบนแผงแสดงด้วย โดยมีรายละเอียดดังตารางแสดงชุดคำสั่งต่อไปนี้ต่อไป

INSTRUCTION	RS	R/W	DATA BIT								EXECUTE 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								40	
CGRAM, DDRAM WR	1	0	WRITE DATA								40	
CGRAM, DDRAM RD	1	1	READ DATA								40	

ตารางที่ 9 แสดงชุดคำสั่ง

ความเข้าใจพื้นฐานเกี่ยวกับแอลซีดีโมดูล

1. การเขียนข้อมูลให้กับแอลซีดีโมดูลจะแบ่งเป็น 2 ลักษณะคือ ส่วนเริ่มต้น และส่วนข้อมูล

โดยจะกำหนดด้วยขาสัญญาณ RS คือถ้า $RS = 0$ จะหมายถึงส่งสัญญาณควบคุม ส่วนเริ่มต้นของแอลซีดีโมดูล หรือค่าแฟล็กของสภาพการทำงานของแอลซีดีโมดูล และถ้า $RS = 1$ จะหมายถึงการเขียนและอ่านข้อมูลกับแอลซีดีโมดูล

2. หลักในการเขียนข้อมูลให้กับแอลซีดีโมดูลนี้คือ เมื่อมีการเขียนข้อมูลลงไปแล้ว ตัวแอลซีดีโมดูลจะต้องใช้เวลาในการส่งขณะหนึ่ง (ตามค่า execute time ในตารางที่ 9) ซึ่งระบบไมโครคอนโทรลเลอร์จะสามารถตรวจสอบได้จากแฟล็ก BF (Busy Flag) และถ้าเรียบร้อยแล้วจึงจะสามารถเขียนข้อมูลอันต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบอินพุต / เอาต์พุตพอร์ตจะไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครคอนโทรลเลอร์ต้องใช้เวลาแทน

3. การเขียนข้อมูลให้กับแอลซีดีโมดูลนี้สามารถทำได้ทั้งแบบ 4 บิตและ แบบ 8 บิต โดยในกรณีแบบ 4 บิต จะใช้สายสัญญาณข้อมูล 4 เส้นคือ DB4-DB7 (ใช้สำหรับระบบไมโครคอนโทรลเลอร์แบบ 4 บิต เพื่อประหยัดสาย) การเขียนข้อมูลแบบ 8 บิต สามารถกระทำเหมือนการเขียนข้อมูลแบบ 4 บิต เพียงแต่ให้เขียน 2 ครั้งคือ DB4-DB7 แล้วตามด้วย DB0-DB3 และต้องกำหนดคุณสมบัติตามค่า DL ตามคำสั่งในชุดหน้าที่ของคำสั่งด้วย

4. DDRAM (Display Data Ram) คือหน่วยความจำภายในตัวแอลซีดีโมดูลที่เป็นบัฟเฟอร์ (buffer) ของข้อมูลโดยถ้าเขียนรหัสแอสกีใด ๆ ลงไปในหน่วยความจำนี้ จะปรากฏเป็นตัวอักษรที่แสดงผลทันที

5. CGRAM (Character Generator Ram) คือหน่วยความจำภายในตัวแอลซีดีโมดูล สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างเองได้ (8 ตัวอักษร) โดยจะอ้างตำแหน่งได้ทั้งหมด 64 ไบต์ (ได้มาจากจำนวน 8 ตัวอักษรคูณกับจำนวนแถว 8 แถว)

รายละเอียดของแต่ละคำสั่ง

1. CLEAR DISPLAY

สำหรับการ clear display จะทำได้โดยการเขียนตัวอักษร space ลงไปใน DDRAM ทั้งหมด และทำการกำหนดค่าตำแหน่ง DDRAM ให้เป็น 0 พร้อมทั้ง cursor จะกลับไปตำแหน่งซ้ายบนสุดของจอภาพ

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

ตารางที่ 10 แสดงการกำหนดค่า clear display

2. CURSOR AT HOME

สำหรับกำหนดค่าตำแหน่ง DDRAM ให้เป็น 0 พร้อมทั้ง cursor จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

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

ตารางที่ 11 แสดงการกำหนดค่าตำแหน่ง DDRAM

3. ENTRY MODE SET

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

ตารางที่ 12 แสดงการกำหนดค่า mode set

I/D = 0 กำหนดทิศทางของ cursor และ DDRAM ให้เป็นแบบลดค่า (decrement)

I/D = 1 กำหนดทิศทางของ cursor และ DDRAM ให้เป็นแบบเพิ่มค่า (increment)

S = 0 เมื่อเขียนข้อมูลแล้ว cursor จะถูกเลื่อนไปตามทิศทางของค่า I/D

S = 1 เมื่อเขียนข้อมูล cursor จะอยู่กับที่ และตัวอักษรจะถูกดันไปตามทิศทางของค่า I/D

การกำหนด I/D และ S นี้ ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้วจะต้องไม่ใช่คำสั่ง 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	1	0	0

ตารางที่ 13 แสดงการกำหนดค่า ON/OFF

D = 0 กำหนดให้ปิดการแสดงผล

D = 1 กำหนดให้เปิดการแสดงผล

C = 0 กำหนดให้ปิด cursor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- C = 1 กำหนดให้ cursor เป็นเส้นขีดใต้ตัวอักษร
 B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่ง cursor
 B = 1 กำหนดให้มีการกระพริบที่ตำแหน่ง cursor (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	*	*

ตารางที่ 14 แสดงการกำหนดค่าการเลื่อน

- S/C = 0 กำหนดให้ cursor R/L
 S/C = 1 กำหนดให้ R/L
 R/L = 0 กำหนดให้มีทิศทางเลื่อนไปทางซ้าย
 R/L = 1 กำหนดให้มีทิศทางเลื่อนไปทางขวา

6. FUNCTION SET

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

ตารางที่ 15 แสดงการกำหนดค่าการเซตฟังก์ชัน

- DL = 0 กำหนดให้การติดต่อกับแอลซีดีโมดูลเป็นแบบ 4 บิต
 DL = 1 กำหนดให้การติดต่อกับแอลซีดีโมดูลเป็นแบบ 8 บิต จะสังเกตว่าการกำหนดค่า D/L นี้สามารถกระทำได้ที่ DB4-DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 บิตตั้งแต่แรกหลังจากจ่ายไฟเลี้ยงก็จะทำให้แอลซีดีโมดูลมีการรับข้อมูลเป็นแบบ 4 บิตอีกที
 N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 duty และ 1/6 duty
 N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 duty
 F = 0 กำหนดจำนวนตัวอักษรเป็นแบบ 5×7 จุด
 F = 1 กำหนดจำนวนตัวอักษรเป็นแบบ 5×10 จุด
 (ในกรณีที่แอลซีดีโมดูลเป็นแบบ 5×7 อยู่แล้วจะไม่มีผลอะไร

7. SET CGRAM ADDRESS

สำหรับการกำหนดตำแหน่งของ CGRAM เมื่อได้ทำการกำหนดไว้แล้วการอ่านและเขียนข้อมูลที่ต้องการนี้จะ เป็นไปตามตำแหน่งที่กำหนดทันที

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

ตารางที่ 16 แสดงการกำหนดค่าตำแหน่งของ CGRAM

8. SET DDRAM ADDRESS

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

ตารางที่ 17 แสดงการกำหนดค่าตำแหน่งของ DDRAM

สำหรับการกำหนดค่าตำแหน่งของ DDRAM เมื่อได้ทำการกำหนดไว้แล้วการอ่านและการเขียนข้อมูลที่ต้องการนี้จะ เป็นไปตามตำแหน่งที่กำหนดทันที ตำแหน่งในแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนตัวอักษรต่อบรรทัดไม่เท่ากัน ซึ่งแสดงตามตารางที่ 15

(ตารางที่ 17 นี้ จะกำหนดให้บิตที่ 7 เป็นลอจิก “1” เสมอ เพื่อความสะดวกในการเลือกใช้)

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

ตารางที่ 18 แสดงการกำหนดค่าตำแหน่งของ DDRAM ที่ใช้งาน

9. BUSY FLAG AND ADDRESS READ

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

ตารางที่ 19 แสดงการกำหนดค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการอ่านค่าแฟล็ก BF ซึ่งบอกถึงความพร้อมของแอลซีดีโมดูลในการรับข้อมูล

BF = 0 หมายความว่าพร้อมที่จะรับข้อมูลต่อไปได้

BF = 1 หมายความว่ายังไม่พร้อม

นอกจากนี้ยังเป็นการอ่านค่าตำแหน่งของ CGRAM หรือ DDRAM ด้วย

การอ่านและเขียนข้อมูลกับ DDRAM/CGRAM

1. WRITE DATA TO DDRAM OR CGRAM

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

ตารางที่ 20 แสดงการเขียนข้อมูลลง DDRAM หรือ CGRAM

สำหรับการเขียนข้อมูลลงในหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ตำแหน่งจะถูกเพิ่มขึ้นหรือลดลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง entry mode set และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้านี้นี้คำสั่งนี้มีการกำหนดตำแหน่งที่ตำแหน่งใด

2. READ DATA TO DDRAM OR CGRAM

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

ตารางที่ 21 แสดงการอ่านข้อมูลลง DDRAM หรือ CGRAM

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการอ่านแล้ว ตำแหน่งจะถูกเพิ่มขึ้นหรือลดลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง entry mode set และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้านี้นี้คำสั่งนี้มีการกำหนดตำแหน่งที่ตำแหน่งใด

แนวทางการเขียนโปรแกรมควบคุม

1. เมื่อจ่ายไฟเลี้ยงให้กับแอลซีดีโมดูลครั้งแรก ภายในจะมีการ reset ระบบโดยอัตโนมัติ ซึ่ง

จะใช้เวลาประมาณ 10 มิลลิวินาที หลังจากทีระดับแรงดันไฟขึ้นถึง 4.5 โวลต์แล้ว ทั้งนี้ระบบรีเซ็ตดังกล่าวจะกระทำสิ่งต่าง ๆ ต่อไปนี้

- ทำการ clear จอภาพทั้งหมด
 - กำหนดคุณสมบัติด้วยคำสั่ง function set คือ $DL = 1$ (ติดต่อกับระบบไมโครคอนโทรลเลอร์ในแบบ 8 บิต), $N = 0$ (แสดงข้อมูล 1 บรรทัด), $F = 0$ (กำหนดตัวอักษรแบบ 5×7 จุด)
 - 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)
- การใช้งานแอลซีดี โมดูลต้องรอให้ขบวนการรีเซ็ตภายในทำงานให้เรียบร้อยก่อน ซึ่งจะตรวจสอบได้ด้วยแฟล็ก BF หรืออาจจะใช้การหน่วงเวลาก็ได้

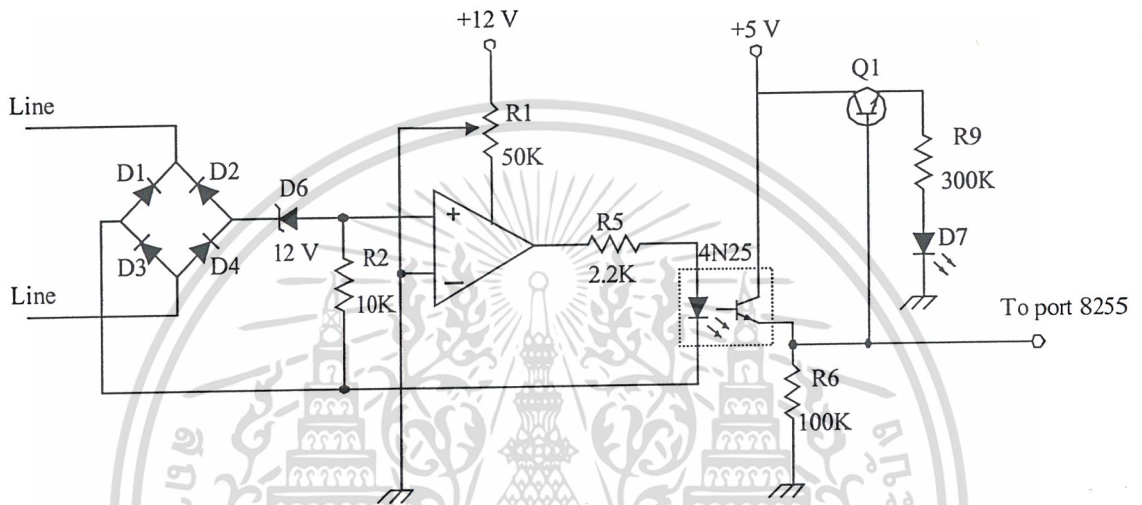
2. การใช้งานแอลซีดี โมดูลจะต้องเกี่ยวข้องกับทางด้าน โปรแกรมเป็นส่วนใหญ่ ชุดคำสั่งต่าง ๆ รวมทั้งการอ่านและเขียนข้อมูลนั้นจะต้องถูกกำหนดด้วยขาสัญญาณทั้งหมดที่มีอยู่ โดยปกติ โปรแกรมจะต้องกำหนดคุณสมบัติต่าง ๆ ที่ต้องการไว้ที่ส่วนเริ่มต้น จากนั้นจะเป็นการเขียนข้อมูลลงในส่วนข้อมูล ซึ่งก็คือข้อความที่จะให้แสดงผลนั่นเอง

บทที่ 3

การออกแบบและการสร้าง

การออกแบบส่วนที่เป็น HARD WARE

3.1 วงจรตรวจสอบสัญญาณโทรศัพท์



รูปที่ 3.1 วงจรตรวจสอบการยกหูวางหู

วงจรส่วนนี้หลักการทำงานคือ ใช้การเปรียบเทียบแรงดันของสัญญาณการยกหูและวางหู การทำงานของวงจรในสภาวะปกติ (วางหู) จะมีแรงดันคร่อมคู่สายโทรศัพท์ประมาณ 48 โวลต์ เมื่อยกหู จะมีแรงดันลดลงเหลือประมาณ 6-10 โวลต์แล้วแต่นชนิดของ โทรศัพท์แต่ละเครื่อง วงจรทำงานจะใช้ ออปแอมป์ ตั้งระดับแรงดันอ้างอิงที่ขา 2 ไว้ที่ระดับ 11 โวลต์ และใช้ออปโต้ไอโซเลเตอร์เป็นตัว เปลี่ยนระดับแรงดันจาก 12 โวลต์ เป็น 5 โวลต์

ขณะที่วางหู แรงดันไฟตรง 48 โวลต์ จะผ่านวงจรบริดจ์ซึ่งจะทำให้ขั้วของแรงดันแน่นอน หลังจากนั้นจะคร่อมตัวต้านทานและซีเนอร์ไดโอด 12 โวลต์ ดังนั้นแรงดันที่ตกคร่อมที่ขา 3 มีค่าสูงสุดได้แค่ 12 โวลต์เพราะซีเนอร์ไดโอดจำกัดแรงดันไว้ เมื่อแรงดันที่เข้ามามีค่ามากกว่าแรงดันอ้างอิง ทำให้เอาต์พุตของออปแอมป์มีค่าประมาณ 12 โวลต์ เป็นผลทำให้เกิดการนำกระแสของ LED ในออปโต้ไอโซเลเตอร์ และไปกระตุ้นให้ทรานซิสเตอร์นำกระแสตามไปด้วย เกิดแรงดันตกคร่อมตัวต้านทาน 100 กิโลโอห์มที่เอาต์พุตเป็น 5 โวลต์ จะแทนด้วยระดับ ลอจิก "1"

ขณะที่ยกหู แรงดันไฟฟ้าลดลงเหลือประมาณ 6-9 โวลต์ ตกคร่อมที่ขา 3 ของออปแอมป์ ทำให้มีแรงดันเข้ามาน้อยกว่าแรงดันอ้างอิง ทำให้เอาต์พุตของออปแอมป์มีค่าประมาณ 0 โวลต์ เป็น

ผลทำให้ LED และทรานซิสเตอร์ในออปโตไดโอดโซเลเตอร์ไม่นำกระแส แรงดันเอาต์พุตจึงตกลงเป็น 0 โวลต์ ซึ่งจะแทนด้วยระดับ ลอจิก “0”

3.2 การออกแบบวงจรตรวจจับสัญญาณเสียง

การออกแบบวงจรตรวจจับสัญญาณเสียงนั้น ก่อนอื่นเราจะต้องกำหนดความถี่ที่จะตรวจจับเสียงก่อน แล้วจึงนำความถี่นั้นไปคำนวณหาค่าตัวต้านทานและตัวเก็บประจุ จากวงจรตามรูปที่ 3.2 จะต้องคำนวณหาค่า R_1 , R_0 , R_x และ C_1

การคำนวณ

ความถี่เสียงพูดที่ใช้ในโทรศัพท์อยู่ระหว่าง 300-3,400 เฮิร์ต และความถี่ของสัญญาณโทนต่างๆ จะอยู่ระหว่าง 400-500 เฮิร์ต ดังนั้นเราจะกำหนดให้ความถี่เสียงพูดมีความถี่ตั้งแต่ 500-4,000 เฮิร์ต เพื่อมิให้รบกวนกับความถี่สัญญาณโทน โดยกำหนดให้สัญญาณโทนอยู่ระหว่าง 400-500 เฮิร์ต

คำนวณวงจรตรวจสอบสัญญาณ

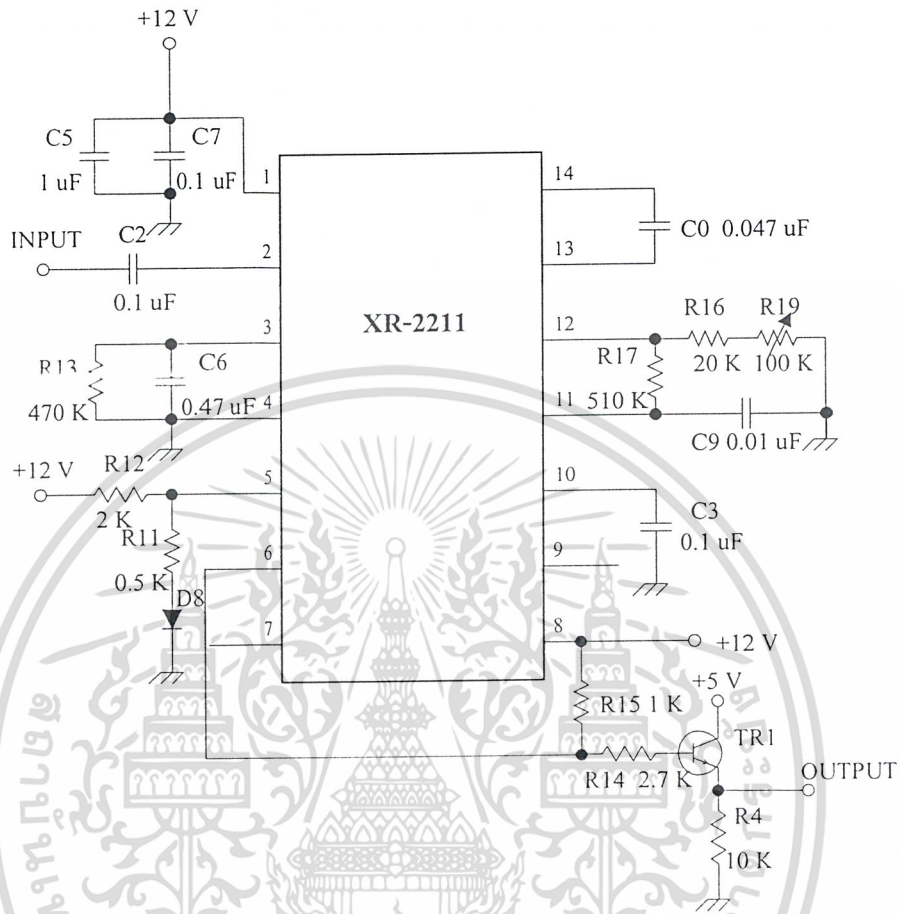
1. เลือก $C_0 = 0.047 \mu\text{F}$

2. เลือก $R_x = 5 \text{ K}\Omega$

3. หา $R_0 = \frac{1}{C_0 f_0} = \frac{1}{0.047 \times 10^{-6} \times 400} = 53 \text{ K}\Omega$

4. หา $C_1 = 0.25C_0 = 0.25 \times 0.047 = 0.011 \mu\text{F}$

5. หา $R_1 = R_0 \times \frac{f_0}{\Delta f} = 53 \times 10^3 \times \frac{400}{40} = 530 \text{ K}\Omega$



รูปที่ 3.2 วงจรตรวจจับสัญญาณเสียง

การทำงานและปรับแต่ง

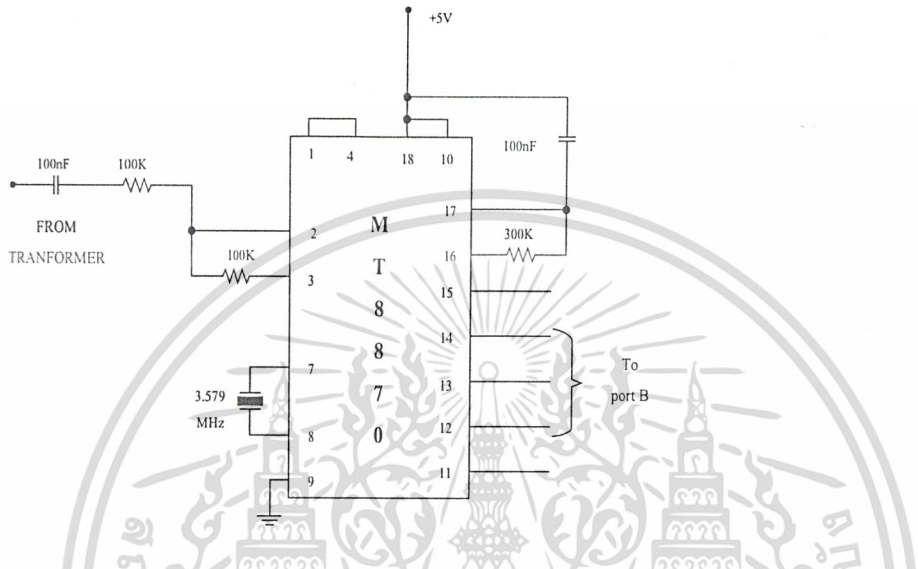
ก่อนที่จะนำวงจรไปใช้งานจะต้องมีการปรับแต่งวงจรเสียก่อน โดยการป้อนความถี่ 400 เฮิร์ต เข้าไปที่อินพุตของวงจรตรวจสอบสัญญาณ สังเกตว่าก่อนป้อนอินพุต LED จะติด และเมื่อป้อนอินพุต เข้าไปแล้ว LED จะดับ ถ้า LED ยังติดอยู่ให้ปรับ R_x จนกระทั่ง LED ดับ จากนั้นให้เพิ่มความถี่อินพุต ไปเรื่อยๆ จนถึงค่าประมาณ 480 เฮิร์ต LED จะติด ถ้า LED ยังดับอยู่ให้ปรับ R_x จน LED ติด เอาต์พุตที่ขา 6 ของไอซี XR-2211 จะเป็นลอจิก “0” เมื่อ LED ติด และจะเป็นลอจิก “1” เมื่อ LED ดับ เมื่อปรับแต่งวงจรเสร็จแล้วจึงจะนำวงจรไปใช้งานได้โดยที่ความถี่ 400-450 เฮิร์ต วงจร detect tone จะให้เอาต์พุตเป็นลอจิก “1”

นำสัญญาณเอาต์พุตที่ขา 6 ไปจับทรานซิสเตอร์เพื่อให้ส่งลอจิก “1” เป็นบวก 5 โวลต์ ไปที่ ไมโครคอนโทรลเลอร์ สัญญาณเอาต์พุตที่รับเข้ามาจากสายโทรศัพท์จะถูกขยายสัญญาณให้แรงขึ้น โดยไอซีเบอร์ LM386 และปรับเกนการขยายได้ที่ VR 10 กิโลโอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบวงจรถอดรหัสสัญญาณ DTMF

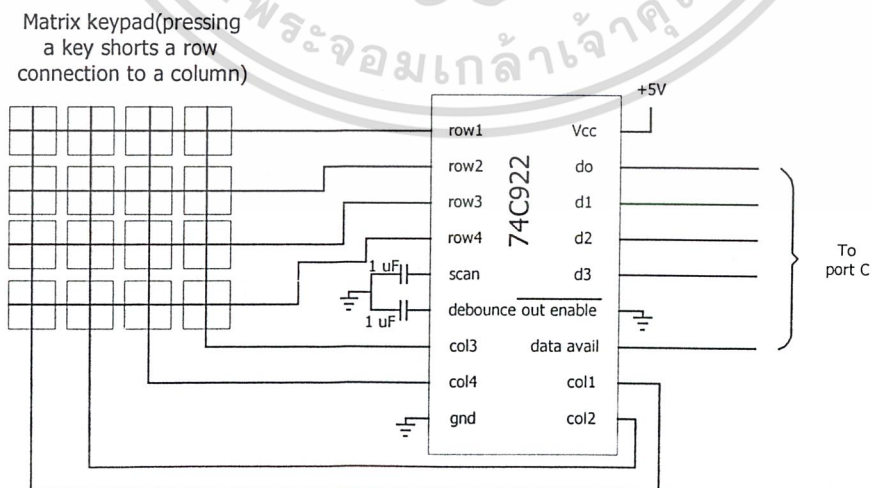
ในการออกแบบวงจร decode สัญญาณ DTMF นั้น เราได้กำหนดเกณฑ์การขยาย (Gain) ได้ตาม Application ของข้อมูลไอซี โดยกำหนดค่า R_1, R_2 ไว้ที่ 100 กิโลโอห์ม และกำหนดค่าเวลาการรีดไทม์ได้ด้วยค่า RC ที่ขา 16, 17 ที่ $R=300$ กิโลโอห์ม $C=100$ นาโนฟารัด



รูปที่ 3.3 แสดงวงจร Decode DTMF

3.4 การออกแบบวงจรถอดรหัส keypad

ในการออกแบบ decode สัญญาณ DTMF นั้น จาก Application ที่ได้ศึกษาสามารถใช้ค่า C_{ext} ในการออกแบบเพื่อแก้ปัญหา Bounce โดยใช้ค่า 1 นาโนฟารัด และความถี่ ออสซิลเลเตอร์ที่ใช้ในการสแกนก็ใช้ ตัวเก็บประจุค่า 1 นาโนฟารัด เช่นกัน แสดงดังรูปที่ 3.4



รูปที่ 3.4 แสดงวงจร scan keypads

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

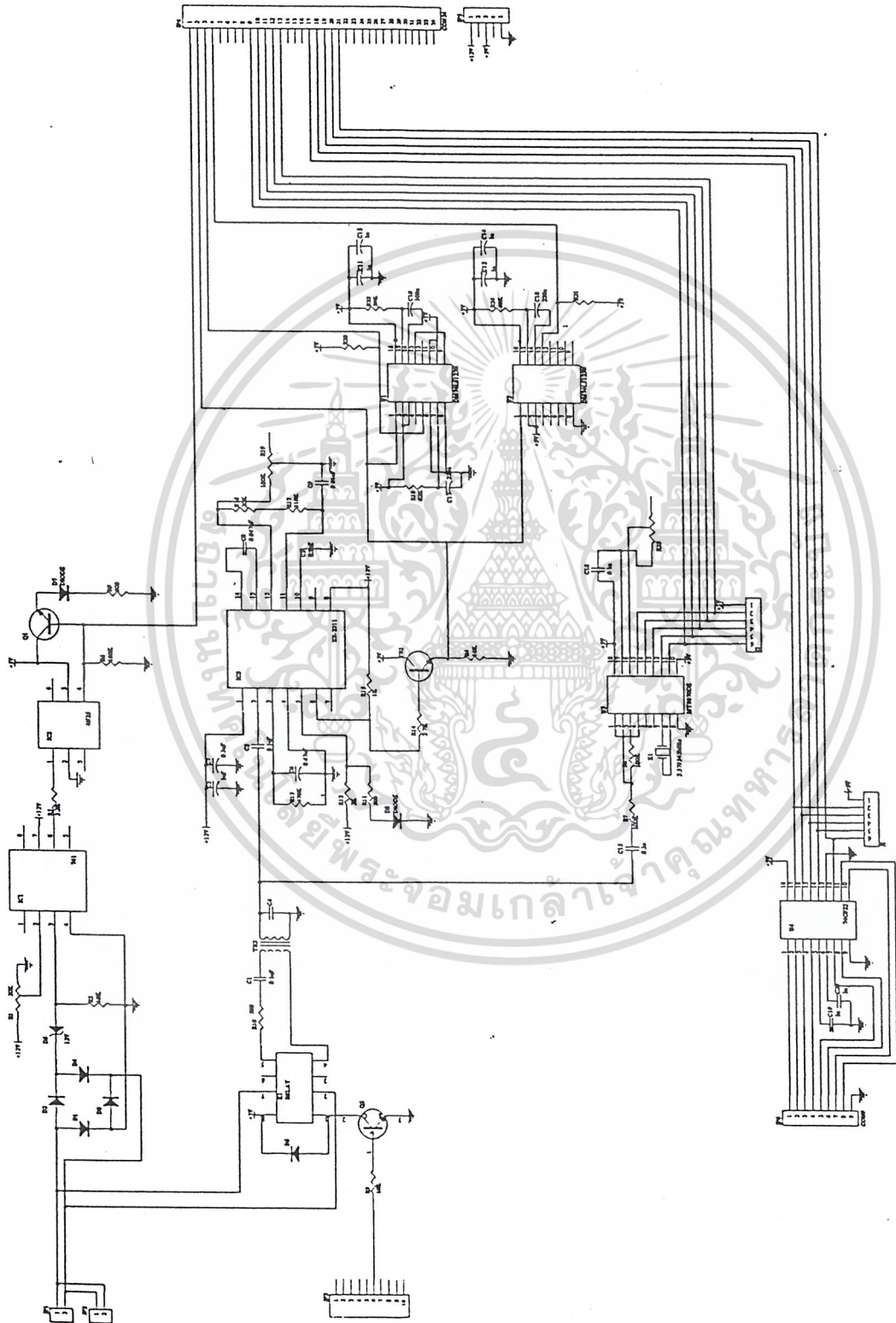
3.5 การทำงานรวมของวงจร

ที่ภาคตรวจสอบสถานะขงหวงหวง ววงหวง เมื่อมีการขงหวงหวงจะมีระดับแรงดันลดลงเหลือ 9-10 โวลต์จากเดิม 49-50 โวลต์ ทำให้ค่าแรงดันมีค่าน้อยกว่าแรงดันอ้างอิงที่ตั้งไว้จากวงจรตรวจสอบสถานะการขงหวงหวงหวงหวงที่มีค่าประมาณ 11 โวลต์ ทำให้เอาต์พุตเปลี่ยนจากลอจิก “1” เมื่อวงหวงหวงเป็นลอจิก “0” บอกให้ไมโครคอนโทรลเลอร์ทราบว่ามีการขงหวงหวง ไมโครคอนโทรลเลอร์จะทำการ on relay ทำให้มีสัญญาณ dial tone, ring back tone, busy tone ไหลผ่านหม้อแปลงอิมพีแดนซ์ดับปลิง 600 โอห์ม ไปสู่วงจรภาคตรวจสอบสัญญาณต่าง ๆ เพื่อตรวจจับสัญญาณที่เข้ามาตามคาบเวลาของสัญญาณที่เข้ามานั้น ๆ แล้วส่งให้ไมโครคอนโทรลเลอร์ทราบ หลังจากนั้นก็จะทำการประมวลผลต่อไปยังส่วนวงจรตรวจสอบสัญญาณ DTMF สัญญาณ DTMF จะผ่านหม้อแปลงเข้ามาเพื่อป้อนให้กับวงจรภาคตรวจสอบสัญญาณ DTMF เอาต์พุตที่ได้จะเป็นสัญญาณดิจิทัล 4 บิต ส่งต่อไปยังพอร์ต B ของไอซี 8255 เพื่อส่งไปให้ไมโครคอนโทรลเลอร์ประมวลผลต่อไป ส่วนในภาควงจรถอดรหัสการกดคีย์ เมื่อมีการกดคีย์ใด ๆ ไอซี 74C922 จะทำการถอดรหัสออกมาทำให้ได้เอาต์พุตเป็นดิจิทัล 4 บิต เช่นเดียวกับไอซี MT8870 หลังจากนั้นจะส่งต่อไปยังพอร์ต C ของ 8255 เพื่อส่งต่อไปให้ไมโครคอนโทรลเลอร์ทำการประมวลผลต่อไป รูปวงจรรวมของส่วนฮาร์ดแวร์แสดงได้ดังรูปที่ 3.5

การออกแบบส่วน SOFTWARE

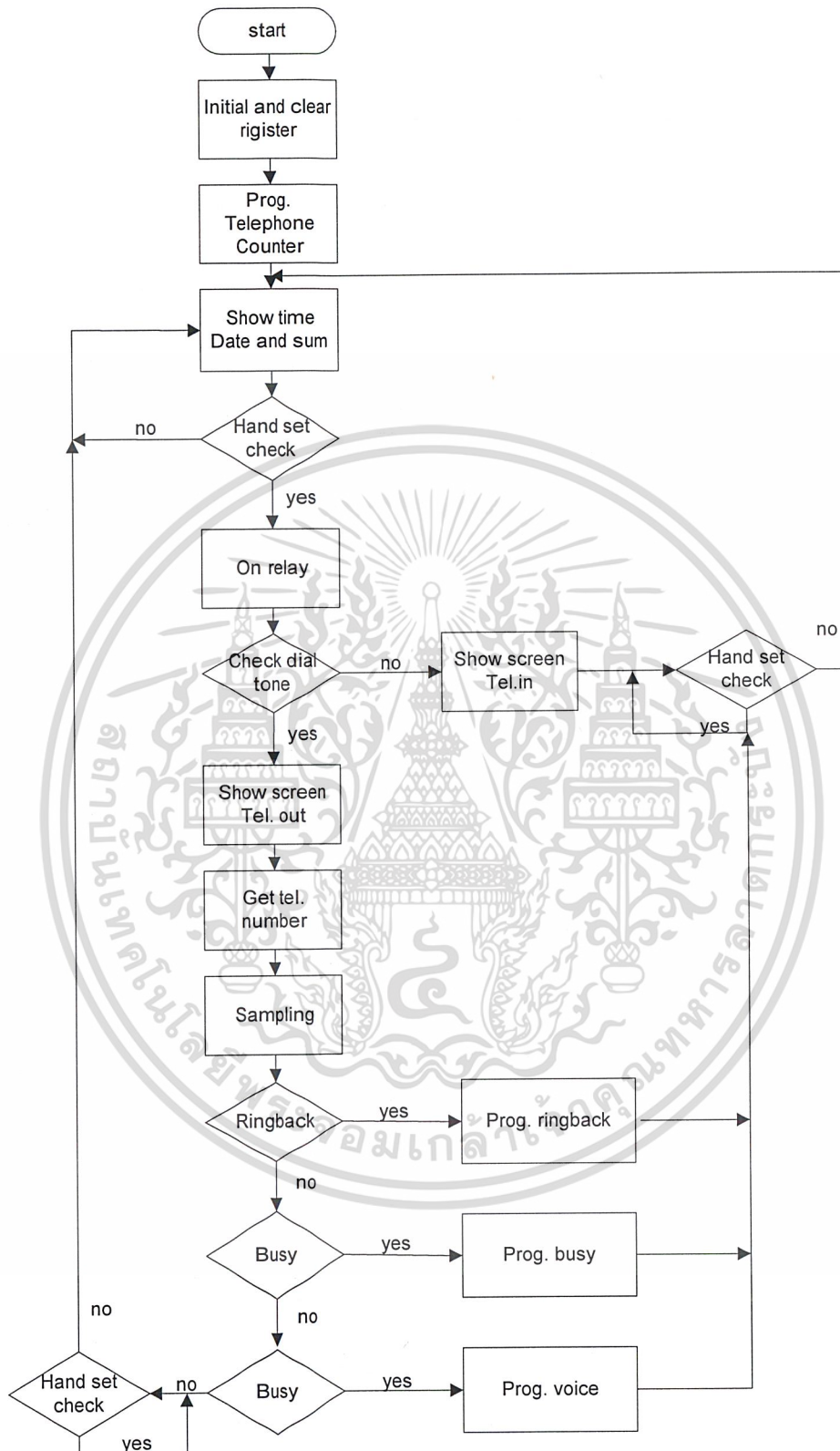
โปรแกรมที่อยู่ในส่วนควบคุมโดยไมโครคอนโทรลเลอร์ เขียนเป็นโฟลว์ชาร์ท (Flow chart) ดังรูปที่ 3.6

การออกแบบส่วนของโปรแกรม จะใช้การตรวจจับสัญญาณเพื่อใช้ในการนับจำนวนการใช้โทรศัพท์ เมื่อเริ่มต้นโปรแกรมจะทำการเคลียร์รีจิสเตอร์ต่าง ๆ และเข้าสู่โหมดการตั้งวันที่และเวลา เมื่อขงหวงหวงโทรศัพท์โปรแกรมจะเช็คสัญญาณให้หมุน และหน้าจอจะแสดงคำว่า “Tel Out” และจะรอรับการกดเลขหมาย เมื่อกดเลขหมายแล้วโปรแกรมจะเช็คสัญญาณที่ส่งมาจากชุมสายโทรศัพท์ ถ้าสถานะของเครื่องโทรศัพท์ทางผู้รับมีสถานะ “ไม่ว่าง” ทางชุมสายโทรศัพท์จะส่งสัญญาณ “สายไม่ว่าง” กลับมา โปรแกรมก็จะทำการเช็คและจะแสดงผลที่หน้าจอว่า “Line Busy” แต่ถ้าสถานะของเครื่องโทรศัพท์ทางผู้รับมีสถานะ “ว่าง” ทางชุมสายโทรศัพท์ก็จะส่ง “สัญญาณเรียก” กลับมาโปรแกรมก็จะทำการเช็คสัญญาณ เมื่อผู้รับทำการรับสายโปรแกรมก็จะทำการนับการใช้โทรศัพท์และจะนำรายละเอียดของเบอร์โทรศัพท์ วัน เดือน ปี ที่เรียกออก, เวลาเริ่มต้น และเวลาสิ้นสุดในการเรียกออกไปเก็บไว้ในหน่วยความจำภายในเครื่อง นอกจากนี้เรายังสามารถนำข้อมูลต่างๆเหล่านี้ส่งออกทางพอร์ตอนุกรมเพื่อส่งข้อมูลให้กับเครื่องคอมพิวเตอร์ได้ โดยโปรแกรมจะทำการติดต่อกับเครื่องคอมพิวเตอร์ให้ทำการแสดงข้อมูลและทำการพิมพ์ออกทางเครื่องพิมพ์ ส่วนซอร์ซโค้ด (Source code) ของโปรแกรมได้แสดงไว้ในส่วนของภาคผนวก ก



รูปที่ 3.5 แสดงวงจรรวมของตัวฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดง Flow chart รวมของโปรแกรมในส่วนควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

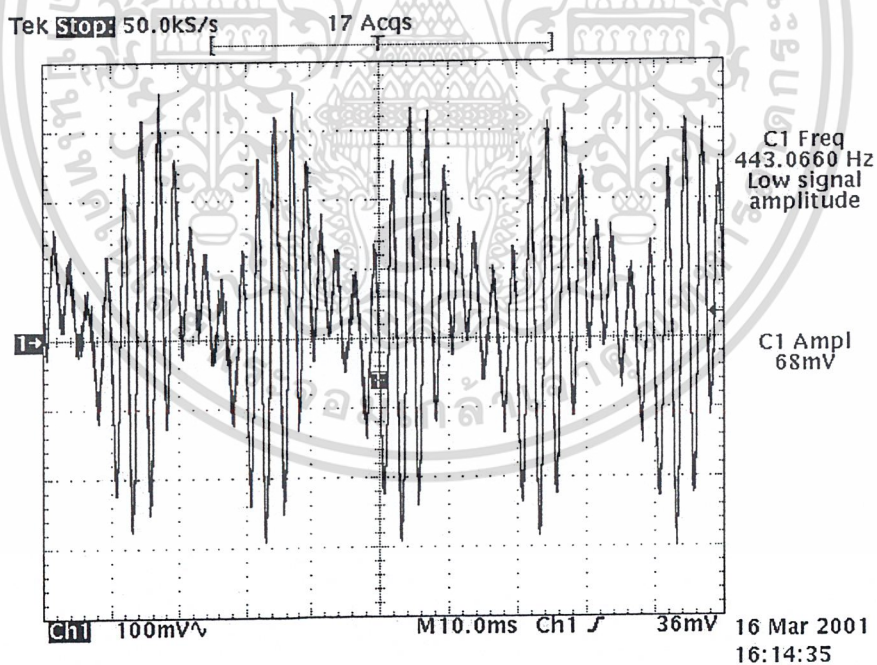
4.1 วงจรตรวจสอบสถานะการยกหู วางหู

ผลการทดลองในส่วนวงจรตรวจสอบสถานะการยกหู วางหู ได้ผลดังนี้คือ เมื่อไม่มีการยกหู เาต์พุตที่ได้จากวงจรจะเป็นสถานะลอจิก “1” และเมื่อมีการยกหูเอาต์พุตที่ได้จากวงจรจะให้สถานะลอจิก “0”

4.2 วงจรตรวจสอบสัญญาณเสียง

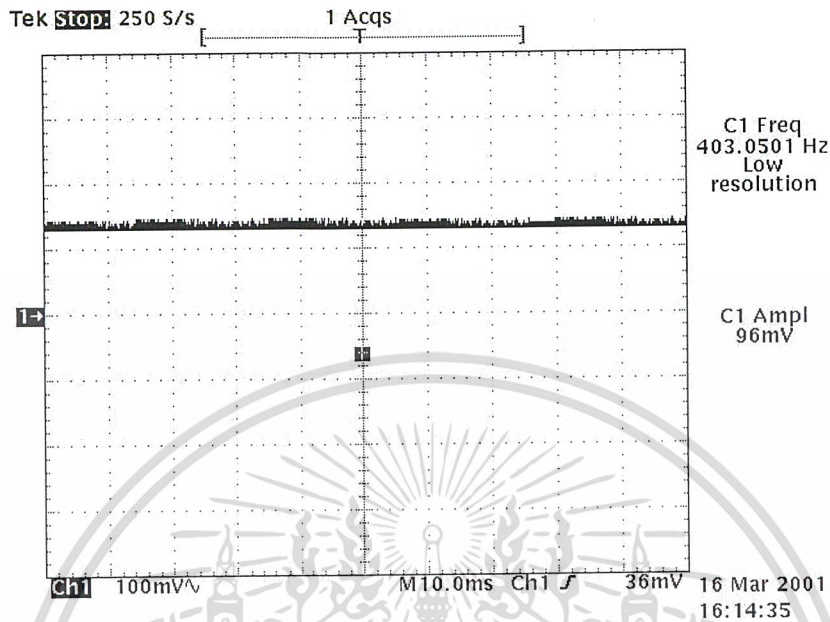
จากการทดลองเราได้นำเอาสัญญาณทางโทรศัพท์ที่ป้อนเข้าสู่วงจร แล้วได้ทำการวัดสัญญาณทางเอาต์พุต ได้รูปสัญญาณเอาต์พุตตามคาบเวลาเหมือนกับสัญญาณทางอินพุตเพียงแต่ว่ารูปสัญญาณที่ออกมา นั้นจะเป็นสัญญาณพัลส์ทางดิจิทัลเพื่อนำไปให้ ไมโครคอนโทรลเลอร์ประมวลผลต่อไป

สัญญาณ Dial Tone



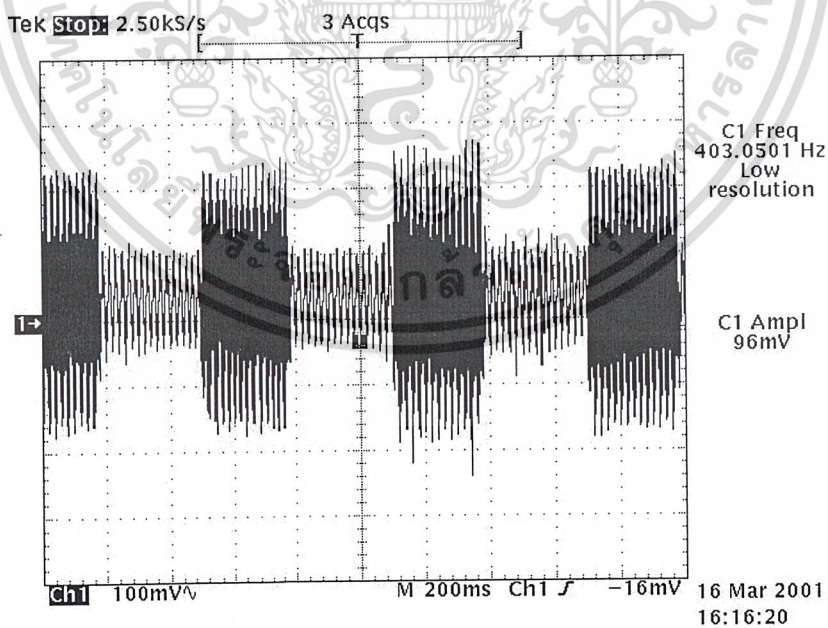
รูปที่ 4.1 สัญญาณ Dial Tone ที่วัดได้จากการทดลอง

สัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Dial Tone)



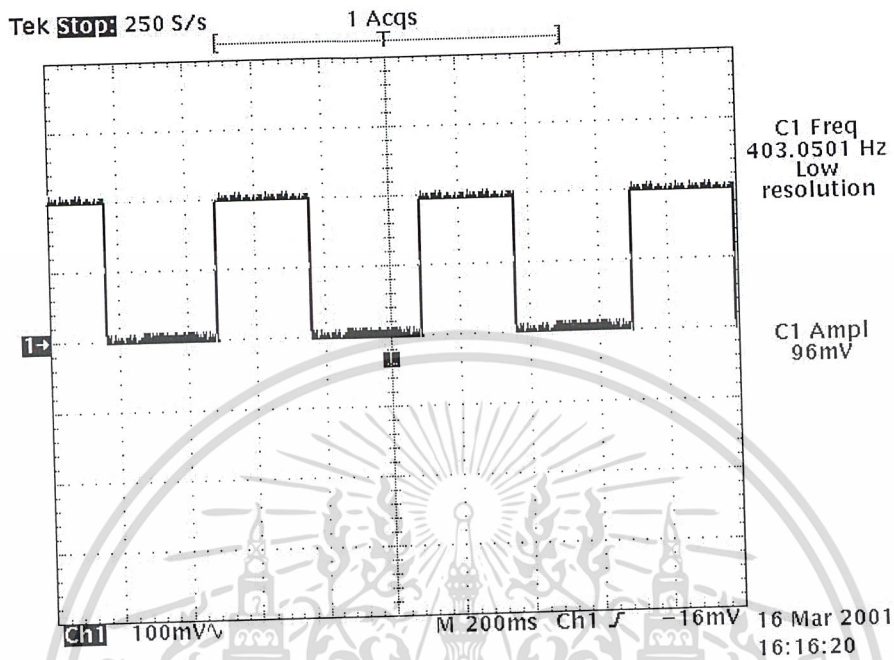
รูปที่ 4.2 แสดงสัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Dial tone) ที่วัดได้จากการทดลอง

สัญญาณ Busy Tone



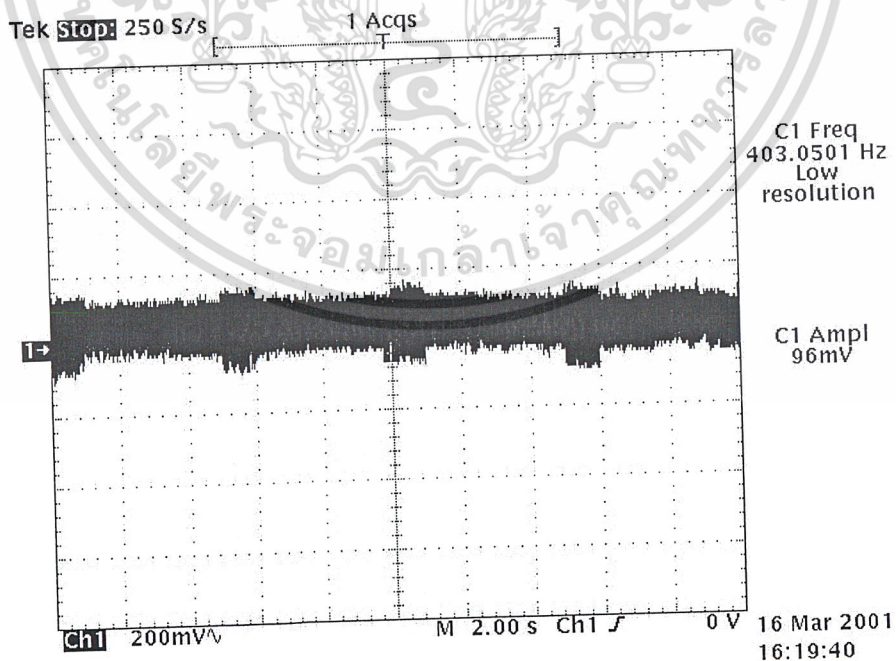
รูปที่ 4.3 แสดงสัญญาณ Busy Tone ที่วัดได้จากการทดลอง

สัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Busy Tone)



รูปที่ 4.4 แสดงสัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Busy Tone)

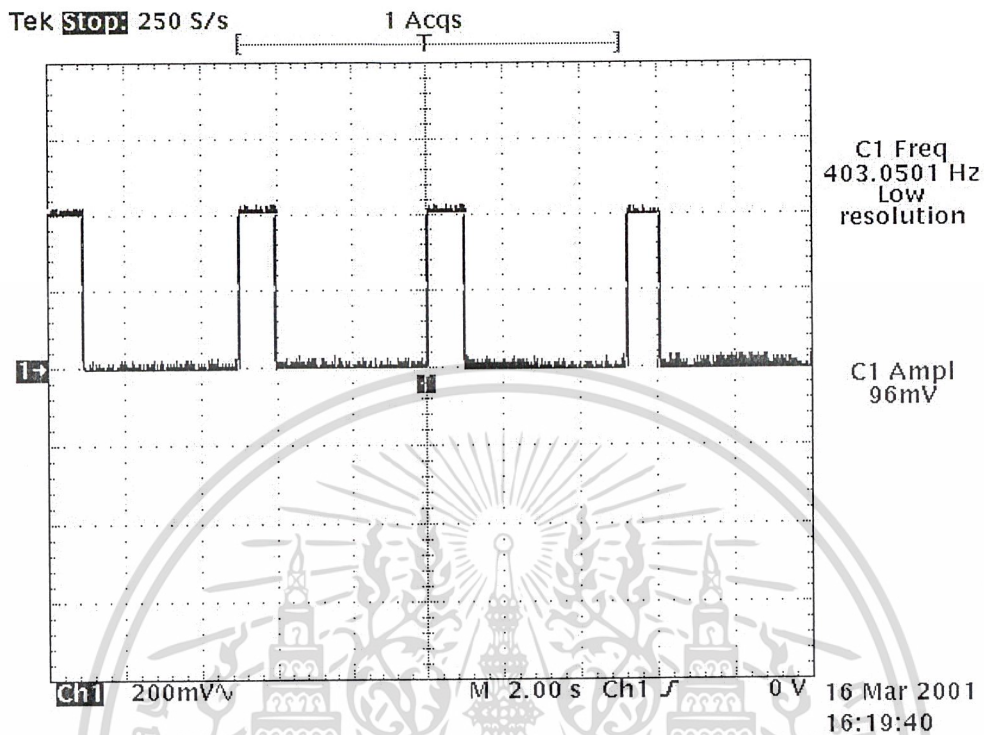
สัญญาณ Ringback Tone



รูปที่ 4.5 แสดงสัญญาณ Ringback Tone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Ringback Tone)



รูปที่ 4.6 สัญญาณเอาต์พุต (เมื่อได้รับอินพุตเป็นสัญญาณ Ringback Tone)

4.3 วงจร Decode สัญญาณ DTMF

ผลการทดลองของวงจรเรานำเอาตารางเอาต์พุตของวงจรมาแสดง เพราะว่าจากการทดลองเขียนโปรแกรมรับข้อมูลเมื่อมีการถอดรหัสสัญญาณแล้ว ได้ข้อมูลเอาต์พุต 4 บิต ดังตารางข้างล่างนี้

F_{low}	F_{high}	DIGIT	D3	D2	D1	D0
697	1219	1	0	0	0	1
697	1336	2	0	0	1	0
697	1477	3	0	0	1	1
770	1209	4	0	1	0	0
770	1336	5	0	1	0	1
770	1477	6	0	1	1	0
852	1209	7	0	1	1	1
852	1336	8	1	0	0	0
852	1477	9	1	0	0	1
941	1209	0	1	0	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

941	1336	*	1	0	1	1
941	1477	#	1	1	0	0
697	1633	A	1	1	0	1
770	1633	B	1	1	1	0
852	1633	C	1	1	1	1
941	1633	D	0	0	0	0

ตารางที่ 22 แสดงการทดสอบวงจรส่วนการถอดรหัสเลขหมายโดยใช้ไอซี MT8870

4.4 วงจรถอดรหัสการกดคีย์

ผลการทดลองจากวงจรนี้ จะได้เอาต์พุตเป็นสัญญาณดิจิทัล 4 บิต ในการทดลองเขียนโปรแกรมรับข้อมูลจะได้เอาต์พุตดังตารางข้างล่างนี้

ROW	COLUMN	D	C	B	A
Y1	X1	0	0	0	0
Y1	X2	0	0	0	1
Y1	X3	0	0	1	0
Y1	X4	0	0	1	1
Y2	X1	0	1	0	0
Y2	X2	0	1	0	1
Y2	X3	0	1	1	0
Y2	X4	0	1	1	1
Y3	X1	1	0	0	0
Y3	X2	1	0	0	1
Y3	X3	1	0	1	0
Y3	X4	1	0	1	1
Y4	X1	1	1	0	0
Y4	X2	1	1	0	1
Y4	X3	1	1	1	0
Y4	X4	1	1	1	1

ตารางที่ 23 แสดงเอาต์พุตที่ได้จากการทดสอบวงจรถอดรหัสการกดคีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป ปัญหาและข้อเสนอแนะ

สรุปการจัดทำโครงการ

จากวัตถุประสงค์ที่ได้ตั้งไว้ว่าจะสร้างเครื่องนับจำนวนการใช้งานโทรศัพท์ ที่สามารถตรวจสอบจำนวนการใช้งานโทรศัพท์ภายในพื้นที่เดียวกันได้นั้น หลังจากที่ได้ศึกษาข้อมูลแล้วนำไปออกแบบสร้างวงจรและทดสอบใช้งานจริงแล้ว ทำให้สามารถสร้างเครื่องนับการใช้งานโทรศัพท์ที่สามารถตรวจสอบจำนวนการใช้งานโทรศัพท์ที่มีความถูกต้อง น่าเชื่อถือ ตามที่ได้ตั้งวัตถุประสงค์ไว้

ปัญหาที่เกิดขึ้นจากการทำโครงการ

จากการทดสอบวงจรต่าง ๆ ที่สร้างขึ้นจากการนำหลักการที่ได้ศึกษามาออกแบบสามารถสรุปปัญหาที่เกิดขึ้นได้ดังนี้

1. ในวงจรตรวจสอบการขहु้นั้น เมื่อนำไปใช้งานกับชุมสายโทรศัพท์อัตโนมัติ (PABX) จะไม่สามารถใช้งานได้ เพราะว่าค่าแรงดันที่ทำการเปรียบเทียบระหว่างการขहुและวงหของคู่สายโทรศัพท์ของทางองค์การโทรศัพท์และในชุมสายโทรศัพท์อัตโนมัติจะมีค่าไม่เท่ากัน ในขณะที่วงหค่าแรงดันในชุมสายโทรศัพท์อัตโนมัติจะมีค่าน้อยกว่าค่าเปรียบเทียบในวงจร จึงทำให้วงจรตรวจสอบว่าเป็น “การขहु” ตลอดเวลา และเมื่อมีสัญญาณเสียงกริ่งเรียกเข้ามาวงจรก็จะทำการตรวจสอบว่ามีกรขहुด้วยตามคาบเวลาของสัญญาณ
2. ในวงจรภาคตรวจสอบสัญญาณเสียง (dial tone detector, ringback tone detector, busy tone detector) เมื่อมีการพูดเข้าไปในโทรศัพท์ ถ้ามีการเปล่งเสียงที่มีความถี่ตรงกับที่ได้ทำการลือกไว้คือประมาณ 400-425 เฮิร์ต วงจรก็จะตรวจสอบพบสัญญาณในคาบที่ตรงกับช่วงที่เปล่งความถี่นั้น
3. ปัญหาในเรื่องการใช้งานกับการบริการชนิดที่คิดอัตราพิเศษ
4. ปัญหาที่เกิดจากสัญญาณรบกวนที่เกิดจากคู่สายโทรศัพท์ในบางพื้นที่

ข้อเสนอแนะ

ในการที่จะสร้างเครื่องนับการใช้งานโทรศัพท์ต้องปรับปรุงและแก้ไขดังนี้

1. ในภาควงจรตรวจสอบสถานะการขहु ต้องทำการปรับค่าแรงดันอ้างอิงให้เหมาะสมกับแรงดันอินพุตที่จะทำการตรวจเช็คสถานะการขहु ส่วนในเรื่องของการมีสัญญาณเสียงกริ่งเรียกเข้ามาสามารถทำการแก้ไขได้โดยเขียนโปรแกรมตรวจสอบการขहुที่แท้จริง เปรียบเทียบกับการขहुจากการที่มีเสียงกริ่ง

2. ในภาคตรวจจับสัญญาณเสียง ควรทำการออกแบบวงจรกรองความถี่ โดยจะทำการกรองเอาความถี่ที่เป็นเสียงพูดของคนออกไปไม่ให้นำเข้ามาในวงจร ให้นำเข้ามาเฉพาะความถี่เสียงที่เป็นสัญญาณทางโทรศัพท์เท่านั้น

3. ในกรณีที่เป็นการคิดค่าบริการในอัตราพิเศษ ควรมีการเขียนโปรแกรมบันทึกการใช้งานพิเศษนั้นทุกครั้ง แล้วนำข้อมูลที่บันทึกไว้ส่งไปประมวลผลทางคอมพิวเตอร์ เพื่อตรวจสอบอีกครั้งหนึ่งว่ามีการใช้งานจริง

4. ในการแก้ไขปัญหาที่เกิดจากสัญญาณรบกวนจากคู่สายโทรศัพท์ในบางพื้นที่นั้น ข้อควรปฏิบัติที่ดีที่สุดนั้นควรจะตรวจสอบสภาพของคู่สายโทรศัพท์ให้อยู่ในสภาพใช้งานได้ดีอยู่เสมอ ถ้ามีความบกพร่องหรือผิดปกติที่ไม่สามารถแก้ไขปัญหาที่เกิดขึ้นได้เอง ควรติดต่อกับชุมสายโทรศัพท์เพื่อแจ้งปัญหาที่เกิดเพื่อทำการแก้ไขต่อไป.

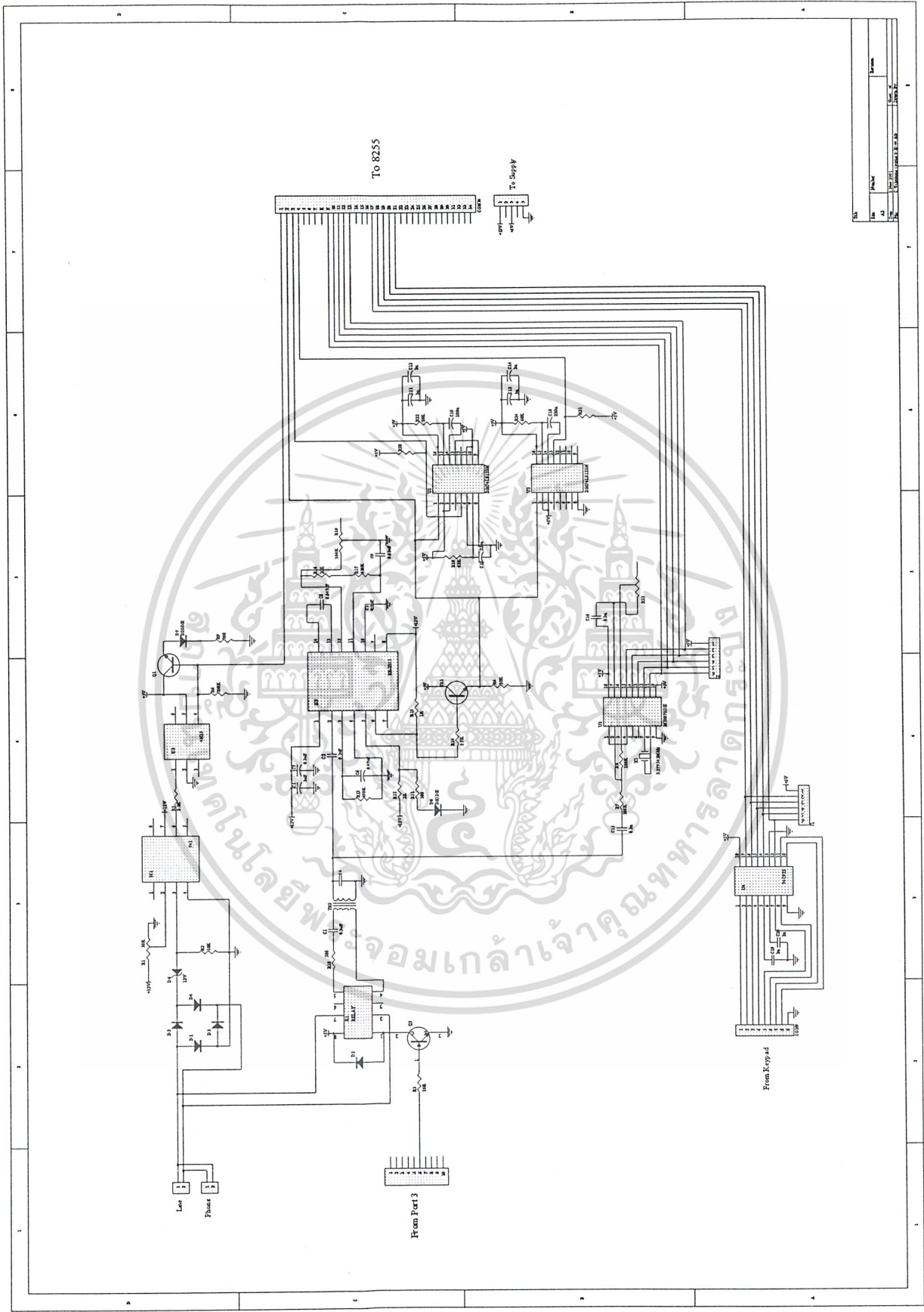


บรรณานุกรม

1. ธวัชชัย เตื่อนฉวี, เทคโนโลยีโทรศัพท์, โรงพิมพ์สุภาลัย, พิมพ์ครั้งที่ 3, กรุงเทพมหานคร, พ.ศ. 2533
2. ถวิล พึ่งมา, ระบบชุมสายโทรศัพท์แบบดิจิทัล, โรงพิมพ์คุรุสภาลาดพร้าว, กรุงเทพมหานคร, พ.ศ. 2535
3. วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิมพรจิตรวิไล, **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบเฟลช ฉบับ AT89C5x ของ Atmel**, บริษัทอิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด, กรุงเทพมหานคร
4. วิสันต์ อาชาติโชพล, ระบบโทรศัพท์ดิจิทัล, สำนักพิมพ์ฟิสิกส์เซ็นเตอร์, กรุงเทพมหานคร, พ.ศ. 2536
5. สุชิน จำจด, วิศวกรรมโทรศัพท์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พิมพ์ครั้งที่ 3, กรุงเทพมหานคร, พ.ศ. 2526
6. <http://www.atmel.com>
7. <http://www.exar.com>
8. <http://www.fairchildsemi.com>
9. <http://mitelsemi.com>
10. <http://www.motorola.com>

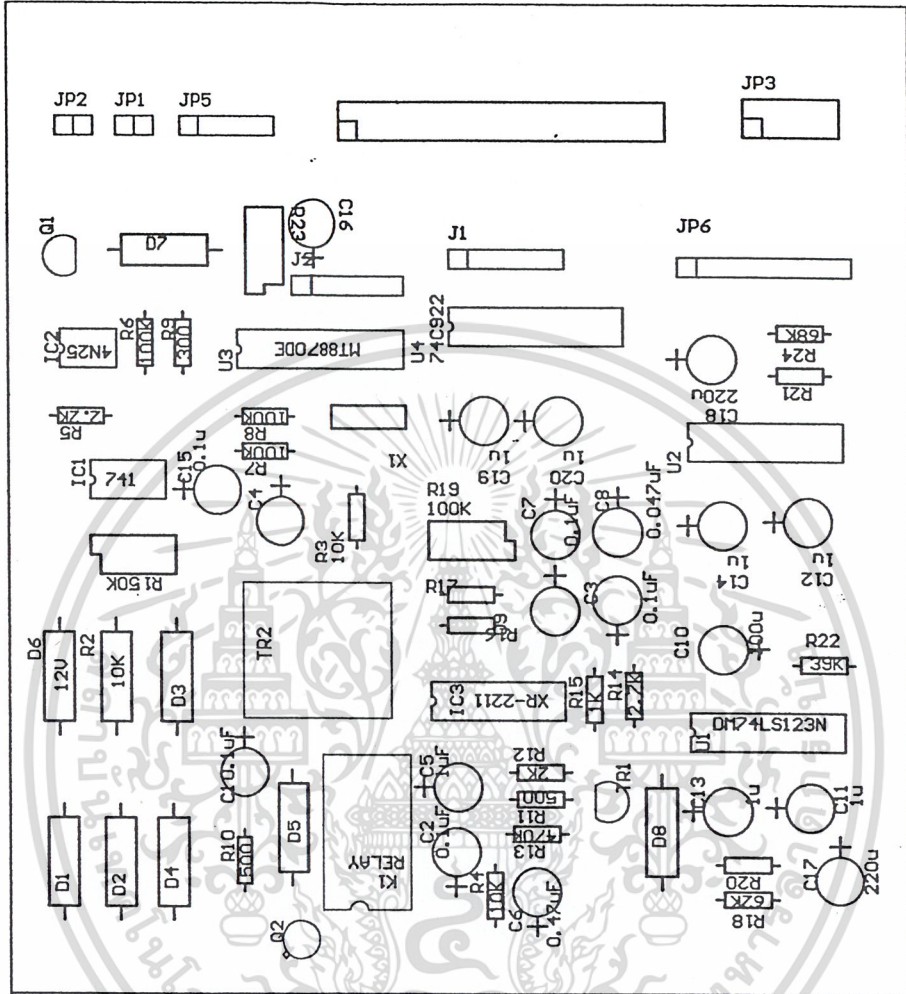


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



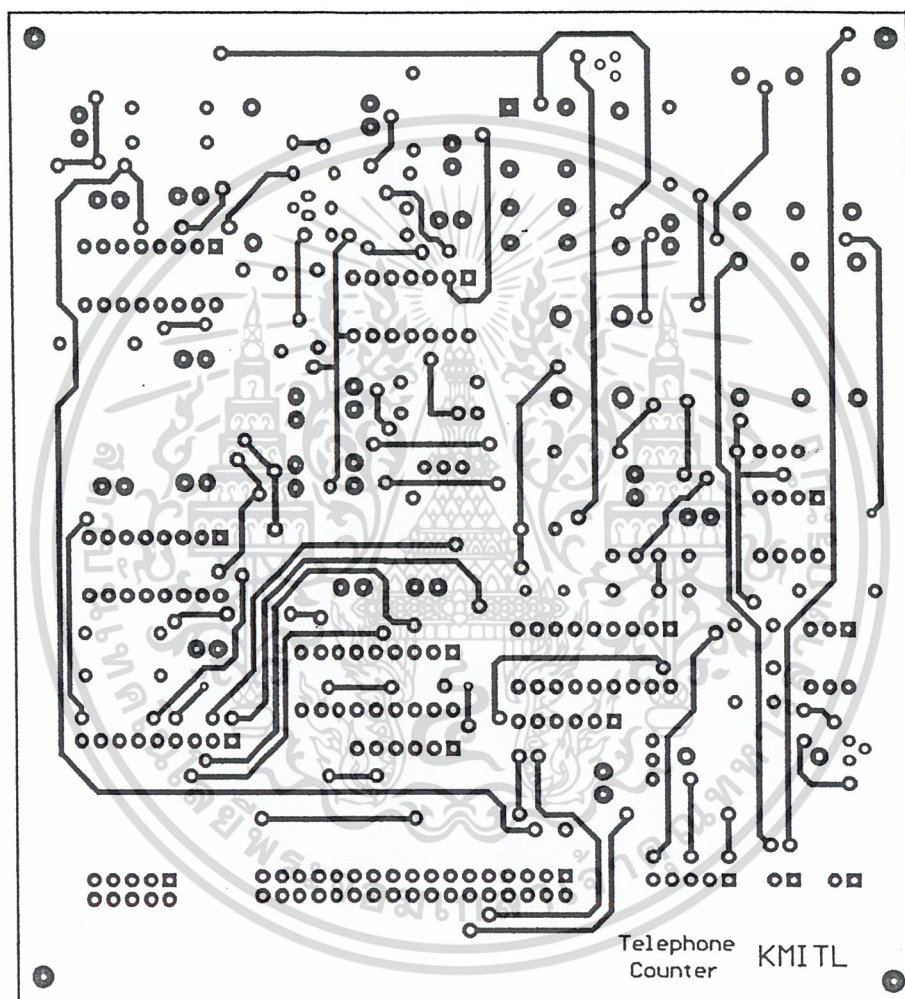
รูปแสดงวงจรรวมของเครื่องนับการใช้งานโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



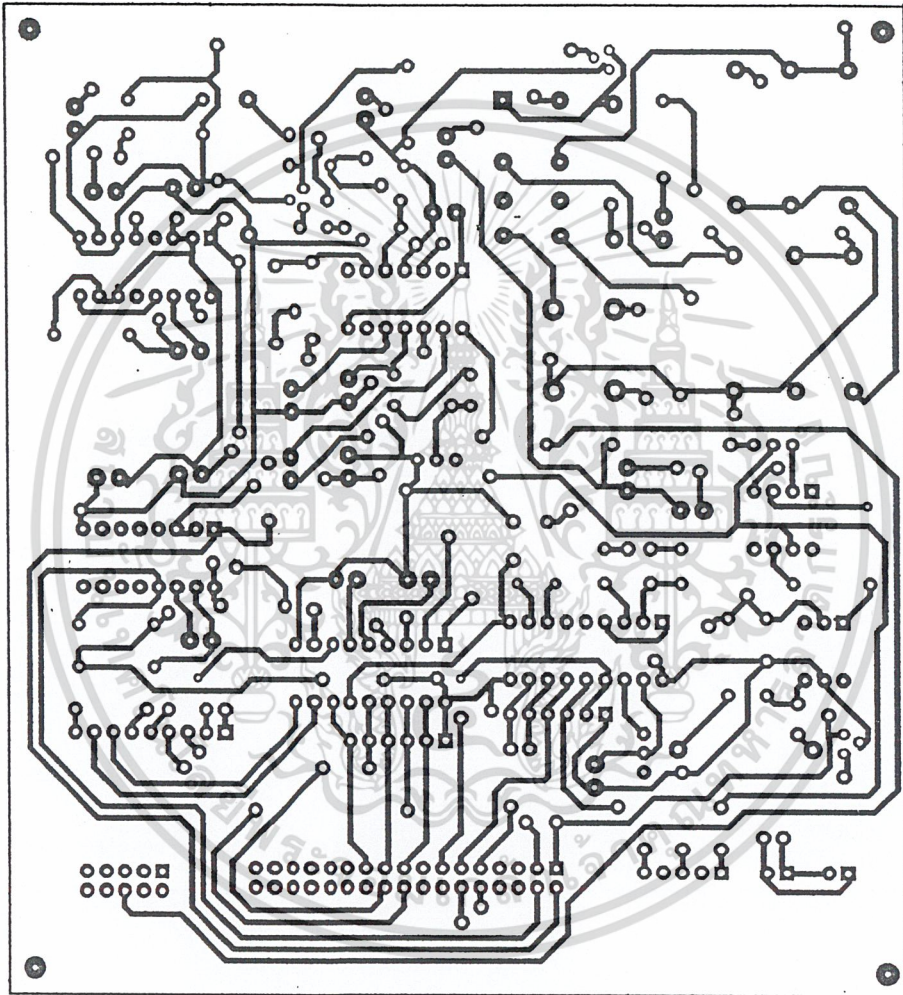
รูปแสดงการวางอุปกรณ์ของเครื่องนับการใช้งานโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



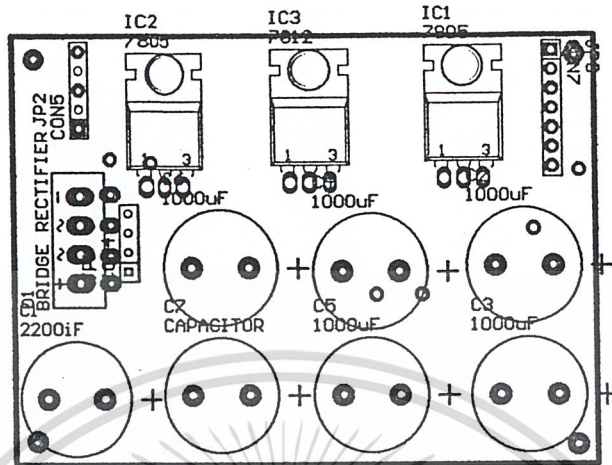
รูปแสดงลายวงจรของเครื่องนับการใช้งานโทรศัพท์ (ด้านบน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

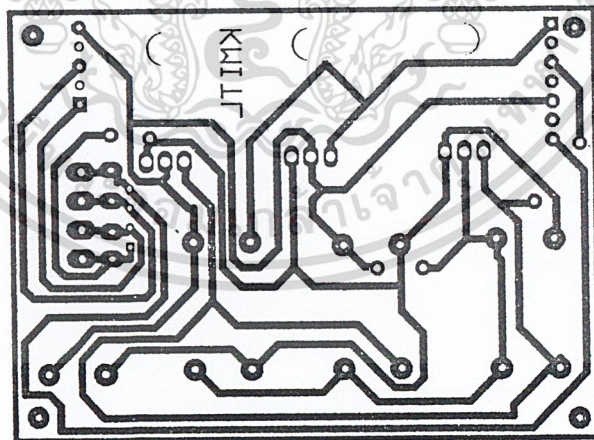


รูปแสดงลายวงจรของเครื่องนับการใช้งาน โทรศัพท์ (ด้านล่าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการวางอุปกรณ์วงจรจ่ายไฟ



รูปแสดงการลายวงจรของวงจรจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

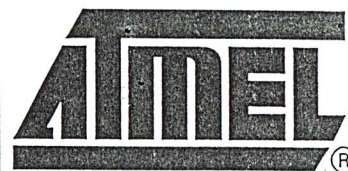
Features

- Compatible with MCS-51® Products
- 8K Bytes of In-System Programmable (ISP) Flash Memory
 - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.



8-bit Microcontroller with 8K Bytes In-System Programmable Flash

AT89S52

Preliminary

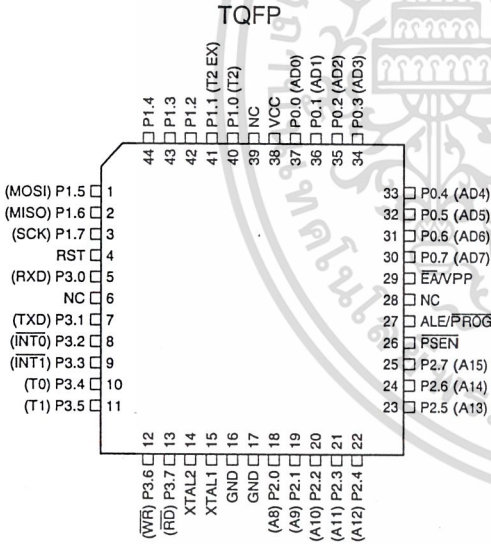
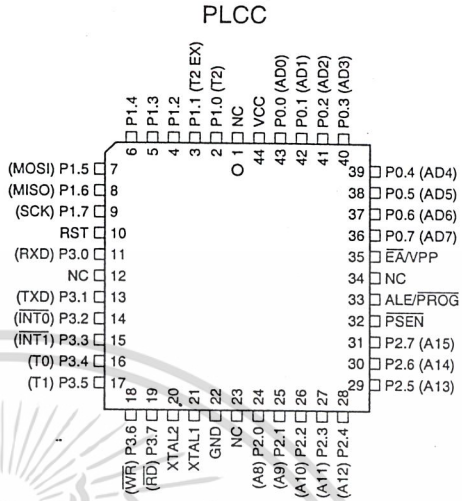
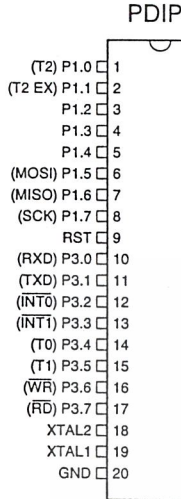
Rev. 1919A-07/01



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Pin Configurations

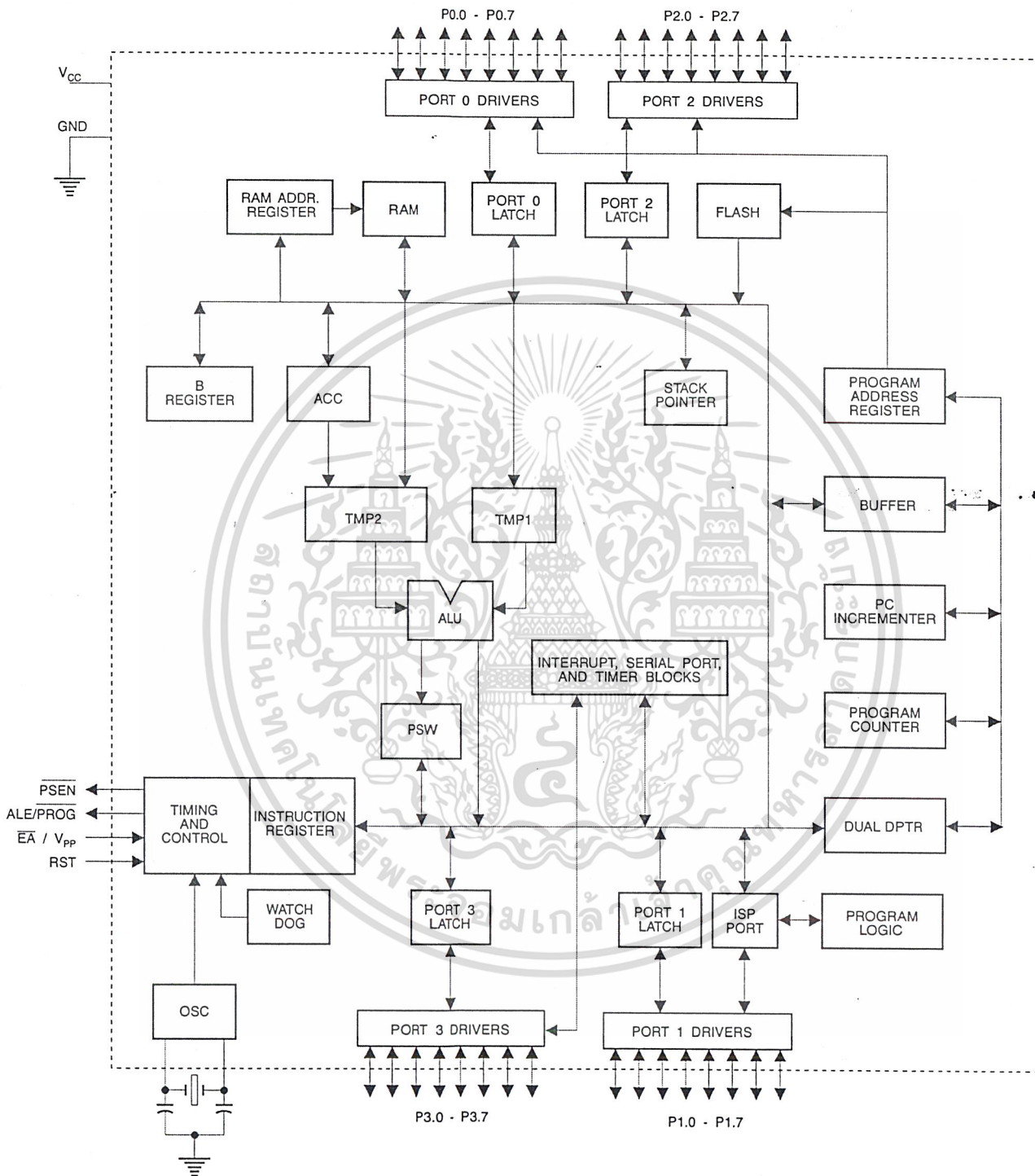


AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S52

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



XR-2211

FSK Demodulator/ Tone Decoder

June 1997-3

FEATURES

- Wide Frequency Range, 0.01Hz to 300kHz
- Wide Supply Voltage Range, 4.5V to 20V
- HCMOS/TTL/Logic Compatibility
- FSK Demodulation, with Carrier Detection
- Wide Dynamic Range, 10mV to 3V rms
- Adjustable Tracking Range, $\pm 1\%$ to 80%
- Excellent Temp. Stability, ± 50 ppm/ $^{\circ}\text{C}$, max.

APPLICATIONS

- Caller Identification Delivery
- FSK Demodulation
- Data Synchronization
- Tone Decoding
- FM Detection
- Carrier Detection

GENERAL DESCRIPTION

The XR-2211 is a monolithic phase-locked loop (PLL) system especially designed for data communications applications. It is particularly suited for FSK modem applications. It operates over a wide supply voltage range of 4.5 to 20V and a wide frequency range of 0.01Hz to 300kHz. It can accommodate analog signals between 10mV and 3V, and can interface with conventional DTL, TTL, and ECL logic families. The circuit consists of a basic PLL for tracking an input signal within the pass band, a

quadrature phase detector which provides carrier detection, and an FSK voltage comparator which provides FSK demodulation. External components are used to independently set center frequency, bandwidth, and output delay. An internal voltage reference proportional to the power supply is provided at an output pin.

The XR-2211 is available in 14 pin packages specified for military and industrial temperature ranges.

ORDERING INFORMATION

Part No.	Package	Operating Temperature Range
XR-2211M	14 Pin CDIP (0.300")	-55 $^{\circ}\text{C}$ to +125 $^{\circ}\text{C}$
XR-2211N	14 Pin CDIP (0.300")	-40 $^{\circ}\text{C}$ to +85 $^{\circ}\text{C}$
XR-2211P	14 Pin PDIP (0.300")	-40 $^{\circ}\text{C}$ to +85 $^{\circ}\text{C}$
XR-2211ID	14 Lead SOIC (Jedec, 0.150")	-40 $^{\circ}\text{C}$ to +85 $^{\circ}\text{C}$

Rev. 3.01

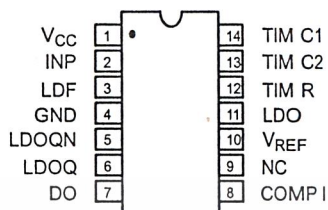
©1992

EXAR Corporation, 48720 Kato Road, Fremont, CA 94538 ♦ (510) 668-7000 ♦ FAX (510) 668-7017

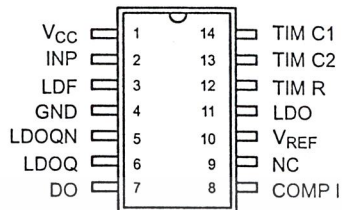
TOM™

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN CONFIGURATION



14 Lead CDIP, PDIP (0.300")



14 Lead SOIC (Jedec, 0.150")

PIN DESCRIPTION

Pin #	.Symbol	Type	Description
1	V _{CC}		Positive Power Supply.
2	INP	I	Receive Analog Input.
3	LDF	O	Lock Detect Filter.
4	GND		Ground Pin.
5	LDOQN	O	Lock Detect Output Not. This output will be low if the VCO is in the capture range.
6	LDOQ	O	Lock Detect Output. This output will be high if the VCO is in the capture range.
7	DO	O	Data Output. Decoded FSK output.
8	COMP I	I	FSK Comparator Input.
9	NC		Not Connected.
10	V _{REF}	O	Internal Voltage Reference. The value of V _{REF} is V _{CC} /2 - 650mV.
11	LDO	O	Loop Detect Output. This output provides the result of the quadrature phase detection.
12	TIM R	I	Timing Resistor Input. This pin connects to the timing resistor of the VCO.
13	TIM C2	I	Timing Capacitor Input. The timing capacitor connects between this pin and pin 14.
14	TIM C1	I	Timing Capacitor Input. The timing capacitor connects between this pin and pin 13.

XR-2211

EXAR

BLOCK DIAGRAM

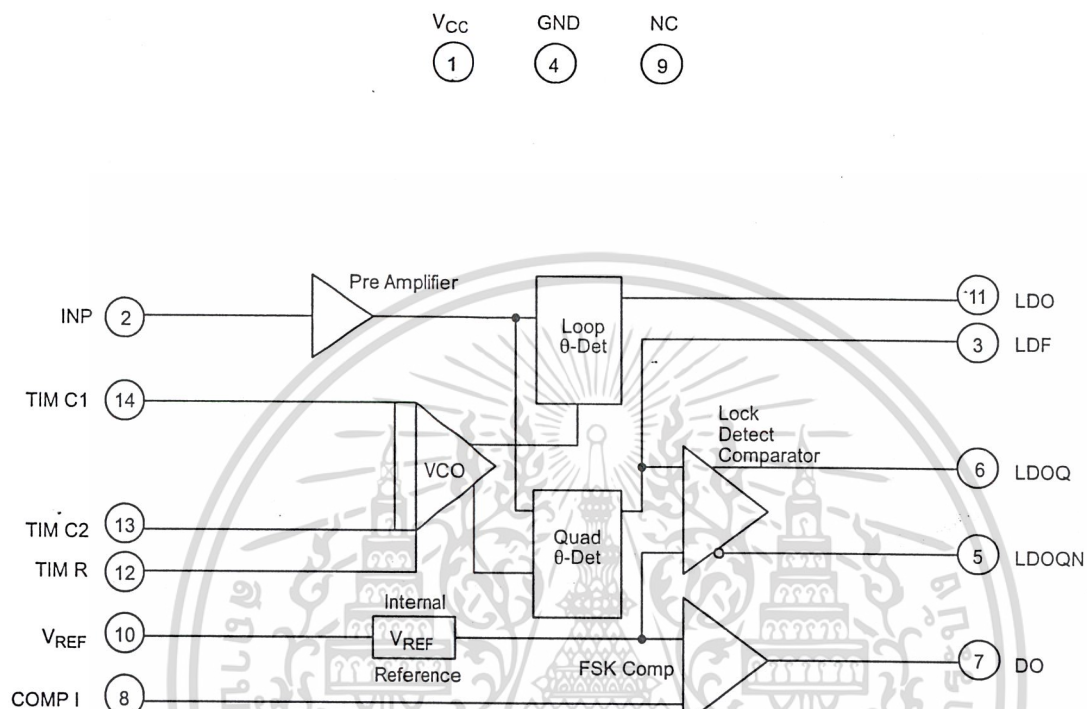


Figure 1. XR-2211 Block Diagram

XR-2211



PRINCIPLES OF OPERATION

Signal Input (Pin 2): Signal is AC coupled to this terminal. The internal impedance at pin 2 is 20K Ω . Recommended input signal level is in the range of 10mV rms to 3V rms.

Quadrature Phase Detector Output (Pin 3): This is the high impedance output of quadrature phase detector and is internally connected to the input of lock detect voltage comparator. In tone detection applications, pin 3 is connected to ground through a parallel combination of R_D and C_D (see *Figure 3*) to eliminate the chatter at lock detect outputs. If the tone detect section is not used, pin 3 can be left open.

Lock Detect Output, Q (Pin 6): The output at pin 6 is at "low" state when the PLL is out of lock and goes to "high" state when the PLL is locked. It is an open collector type output and requires a pull-up resistor, R_L , to V_{CC} for proper operation. At "low" state, it can sink up to 5mA of load current.

Lock Detect Complement, (Pin 5): The output at pin 5 is the logic complement of the lock detect output at pin 6. This output is also an open collector type stage which can sink 5mA of load current at low or "on" state.

FSK Data Output (Pin 7): This output is an open collector logic stage which requires a pull-up resistor, R_L , to V_{CC} for proper operation. It can sink 5mA of load current. When decoding FSK signals, FSK data output is at "high" or "off" state for low input frequency, and at "low" or "on" state for high input frequency. If no input signal is present, the logic state at pin 7 is indeterminate.

FSK Comparator Input (Pin 8): This is the high impedance input to the FSK voltage comparator. Normally, an FSK post-detection or data filter is connected between this terminal and the PLL phase detector output (pin 11). This data filter is formed by R_F and C_F (see *Figure 3*.) The threshold voltage of the comparator is set by the internal reference voltage, V_{REF} , available at pin 10.

Reference Voltage, V_{REF} (Pin 10): This pin is internally biased at the reference voltage level, $V_{REF} = V_{CC}/2 - 650\text{mV}$. The DC voltage level at this pin forms an internal reference for the voltage levels at pins 5, 8, 11 and 12. Pin

10 must be bypassed to ground with a 0.1 μF capacitor for proper operation of the circuit.

Loop Phase Detector Output (Pin 11): This terminal provides a high impedance output for the loop phase detector. The PLL loop filter is formed by R_1 and C_1 connected to pin 11 (see *Figure 3*.) With no input signal, or with no phase error within the PLL, the DC level at pin 11 is very nearly equal to V_{REF} . The peak to peak voltage swing available at the phase detector output is equal to $2 \times V_{REF}$.

VCO Control Input (Pin 12): VCO free-running frequency is determined by external timing resistor, R_0 , connected from this terminal to ground. The VCO free-running frequency, f_0 , is:

$$f_0 = \frac{1}{R_0 \cdot C_0} \text{ Hz}$$

where C_0 is the timing capacitor across pins 13 and 14. For optimum temperature stability, R_0 must be in the range of 10K Ω to 100K Ω (see *Figure 9*.)

This terminal is a low impedance point, and is internally biased at a DC level equal to V_{REF} . The maximum timing current drawn from pin 12 must be limited to $\leq 3\text{mA}$ for proper operation of the circuit.

VCO Timing Capacitor (Pins 13 and 14): VCO frequency is inversely proportional to the external timing capacitor, C_0 , connected across these terminals (see *Figure 6*.) C_0 must be non-polar, and in the range of 200pF to 10 μF .

VCO Frequency Adjustment: VCO can be fine-tuned by connecting a potentiometer, R_X , in series with R_0 at pin 12 (see *Figure 10*.)

VCO Free-Running Frequency, f_0 : XR-2211 does not have a separate VCO output terminal. Instead, the VCO outputs are internally connected to the phase detector sections of the circuit. For set-up or adjustment purposes, the VCO free-running frequency can be tuned by using the generalized circuit in *Figure 3*, and applying an alternating bit pattern of 0's and 1's at the known mark and space frequencies. By adjusting R_0 , the VCO can then be tuned to obtain a 50% duty cycle on the FSK output (pin 7). This will ensure that the VCO f_0 value is accurately referenced to the mark and space frequencies.



October 1987
Revised April 2001

MM74C922 • MM74C923 16-Key Encoder • 20-Key Encoder

General Description

The MM74C922 and MM74C923 CMOS key encoders provide all the necessary logic to fully encode an array of SPST switches. The keyboard scan can be implemented by either an external clock or external capacitor. These encoders also have on-chip pull-up devices which permit switches with up to 50 kΩ on resistance to be used. No diodes in the switch array are needed to eliminate ghost switches. The internal debounce circuit needs only a single external capacitor and can be defeated by omitting the capacitor. A Data Available output goes to a high level when a valid keyboard entry has been made. The Data Available output returns to a low level when the entered key is released, even if another key is depressed. The Data Available will return high to indicate acceptance of the new key after a normal debounce period; this two-key roll-over is provided between any two switches.

An internal register remembers the last key pressed even after the key is released. The 3-STATE outputs provide for easy expansion and bus operation and are LPTTL compatible.

Features

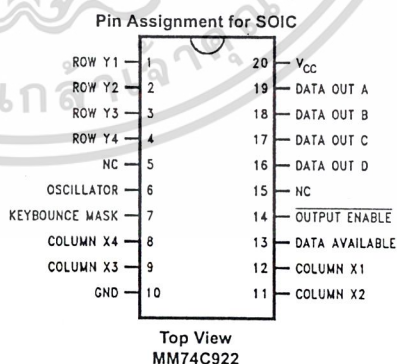
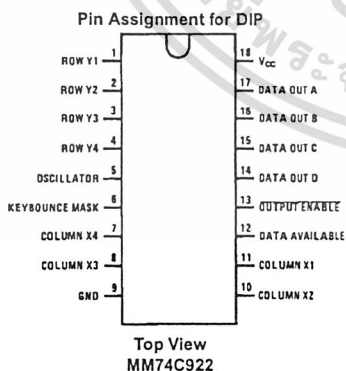
- 50 kΩ maximum switch on resistance
- On or off chip clock
- On-chip row pull-up devices
- 2 key roll-over
- Keybounce elimination with single capacitor
- Last key register at outputs
- 3-STATE output LPTTL compatible
- Wide supply range: 3V to 15V
- Low power consumption

Ordering Code:

Order Number	Package Number	Package Description
MM74C922WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C922N	N18B	18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
MM74C923WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C923N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

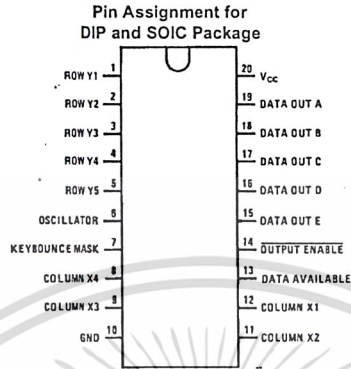
Device also available in Tape and Reel. Specify by appending suffix letter "X" to the ordering code.

Connection Diagrams



MM74C922 • MM74C923

Connection Diagrams (Continued)



Top View
MM74C923

Truth Tables

(Pins 0 through 11)

Switch Position	0	1	2	3	4	5	6	7	8	9	10	11
	Y1, X1	Y1, X2	Y1, X3	Y1, X4	Y2, X1	Y2, X2	Y2, X3	Y2, X4	Y3, X1	Y3, X2	Y3, X3	Y3, X4
D												
A A	0	1	0	1	0	1	0	1	0	1	0	1
T B	0	0	1	1	0	0	1	1	0	0	1	1
A C	0	0	0	0	1	1	1	1	0	0	0	0
O D	0	0	0	0	0	0	0	0	1	1	1	1
U E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0
T												

(Pins 12 through 19)

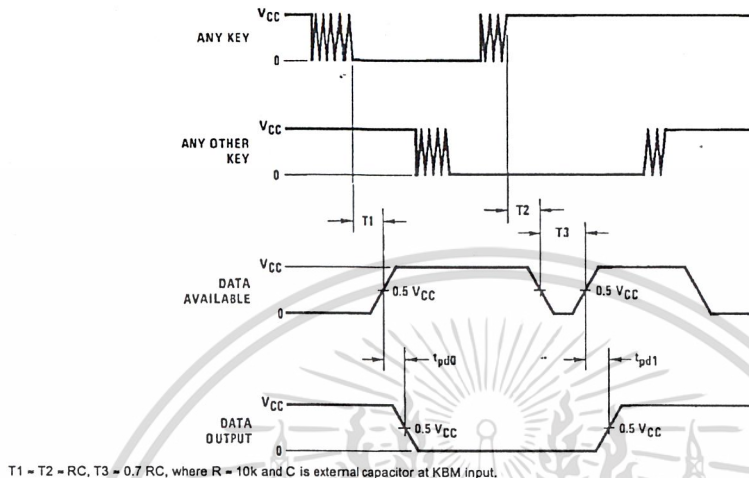
Switch Position	12	13	14	15	16	17	18	19
	Y4, X1	Y4, X2	Y4, X3	Y4, X4	Y5 (Note 1), X1	Y5 (Note 1), X2	Y5 (Note 1), X3	Y5 (Note 1), X4
D								
A A	0	1	0	1	0	1	0	1
T B	0	0	1	1	0	0	1	1
A C	1	1	1	1	0	0	0	0
O D	1	1	1	1	0	0	0	0
U E (Note 1)	0	0	0	0	1	1	1	1
T								

Note 1: Omit for MM74C922

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MM74C922 • MM74C923

Switching Time Waveforms



$T1 = T2 = RC$, $T3 = 0.7 RC$, where $R = 10k$ and C is external capacitor at KBM input.

FIGURE 1.

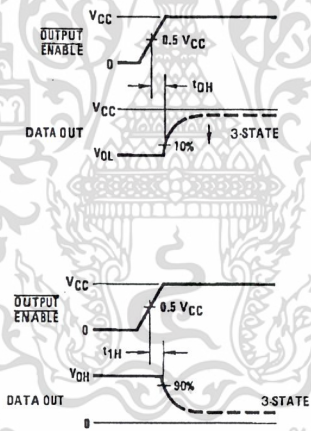


FIGURE 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ISO²-CMOS MT8870D/MT8870D-1 Integrated DTMF Receiver

Features

- Complete DTMF Receiver
- Low power consumption
- Internal gain setting amplifier
- Adjustable guard time
- Central office quality
- Power-down mode
- Inhibit mode
- Backward compatible with MT8870C/MT8870C-1

ISSUE 5

March 1997

Ordering Information

MT8870DE/DE-1	18 Pin Plastic DIP
MT8870DS/DS-1	18 Pin SOIC
MT8870DN/DN-1	20 Pin SSOP
-40 °C to +85 °C	

Description

The MT8870D/MT8870D-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions. The filter section uses switched capacitor techniques for high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone-pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.

Applications

- Receiver system for British Telecom (BT) or CEPT Spec (MT8870D-1)
- Paging systems
- Repeater systems/mobile radio
- Credit card systems
- Remote control
- Personal computers
- Telephone answering machine

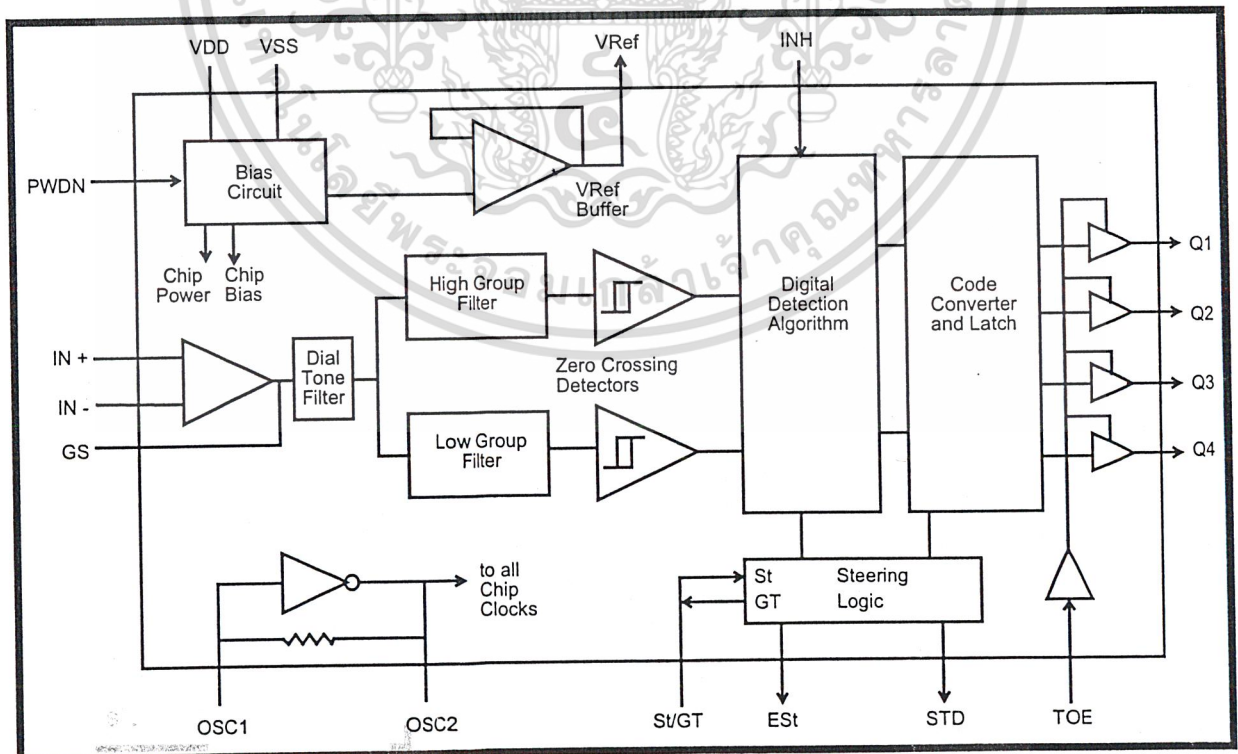


Figure 1 - Functional Block Diagram

MT8870D/MT8870D-1 ISO²-CMOS

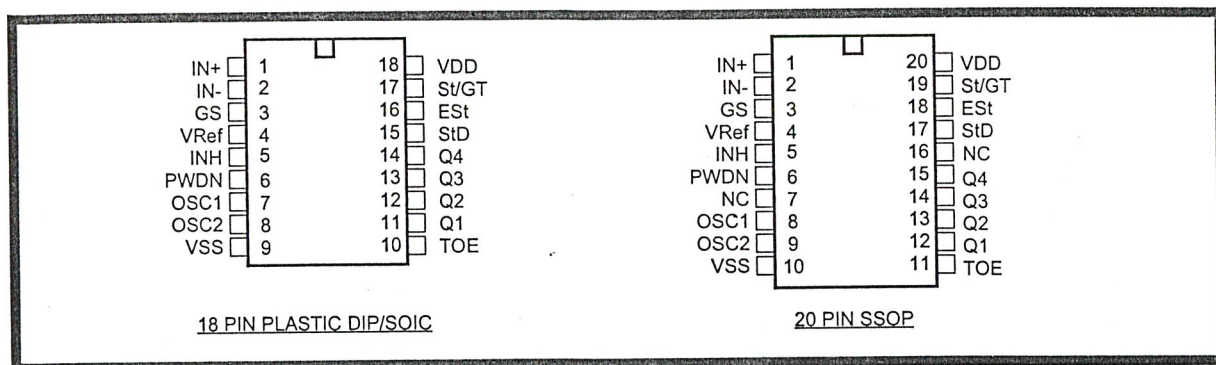


Figure 2 - Pin Connections

Pin Description

Pin #		Name	Description
18	20		
1	1	IN+	Non-Inverting Op-Amp (Input).
2	2	IN-	Inverting Op-Amp (Input).
3	3	GS	Gain Select. Gives access to output of front end differential amplifier for connection of feedback resistor.
4	4	V _{Ref}	Reference Voltage (Output). Nominally V _{DD} /2 is used to bias inputs at mid-rail (see Fig. 6 and Fig. 10).
5	5	INH	Inhibit (Input). Logic high inhibits the detection of tones representing characters A, B, C and D. This pin input is internally pulled down.
6	6	PWDN	Power Down (Input). Active high. Powers down the device and inhibits the oscillator. This pin input is internally pulled down.
7	8	OSC1	Clock (Input).
8	9	OSC2	Clock (Output). A 3.579545 MHz crystal connected between pins OSC1 and OSC2 completes the internal oscillator circuit.
9	10	V _{SS}	Ground (Input). 0V typical.
10	11	TOE	Three State Output Enable (Input). Logic high enables the outputs Q1-Q4. This pin is pulled up internally.
11-14	12-15	Q1-Q4	Three State Data (Output). When enabled by TOE, provide the code corresponding to the last valid tone-pair received (see Table 1). When TOE is logic low, the data outputs are high impedance.
15	17	StD	Delayed Steering (Output). Presents a logic high when a received tone-pair has been registered and the output latch updated; returns to logic low when the voltage on St/GT falls below V _{TSt} .
16	18	Est	Early Steering (Output). Presents a logic high once the digital algorithm has detected a valid tone pair (signal condition). Any momentary loss of signal condition will cause Est to return to a logic low.
17	19	St/GT	Steering Input/Guard time (Output) Bidirectional. A voltage greater than V _{TSt} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V _{TSt} frees the device to accept a new tone pair. The GT output acts to reset the external steering time-constant; its state is a function of Est and the voltage on St.
18	20	V _{DD}	Positive power supply (Input). +5V typical.
	7, 16	NC	No Connection.

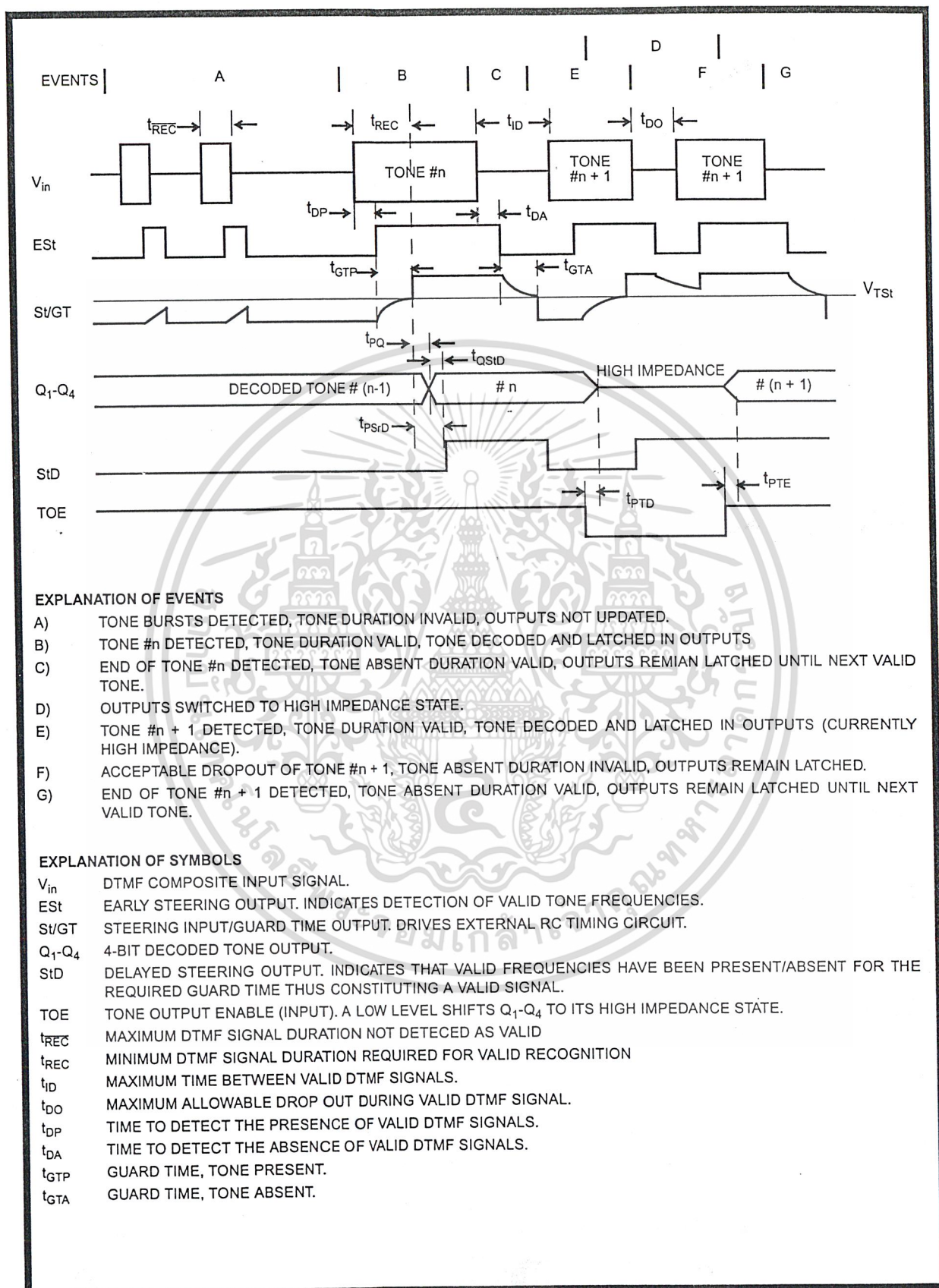


Figure 11 - Timing Diagram



GlobalOptoisolator™



6-Pin DIP Optoisolators Transistor Output

The 4N25/A, 4N26, 4N27 and 4N28 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Most Economical Optoisolator Choice for Medium Speed, Switching Applications
- Meets or Exceeds All JEDEC Registered Specifications
- *To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.*

Applications

- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid State Relays

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
--------	--------	-------	------

INPUT LED

Reverse Voltage	V _R	3	Volts
Forward Current — Continuous	I _F	60	mA
LED Power Dissipation @ T _A = 25°C with Negligible Power in Output Detector Derate above 25°C	P _D	120	mW
		1.41	mW/°C

OUTPUT TRANSISTOR

Collector–Emitter Voltage	V _{CEO}	30	Volts
Emitter–Collector Voltage	V _{ECO}	7	Volts
Collector–Base Voltage	V _{CBO}	70	Volts
Collector Current — Continuous	I _C	150	mA
Detector Power Dissipation @ T _A = 25°C with Negligible Power in Input LED Derate above 25°C	P _D	150	mW
		1.76	mW/°C

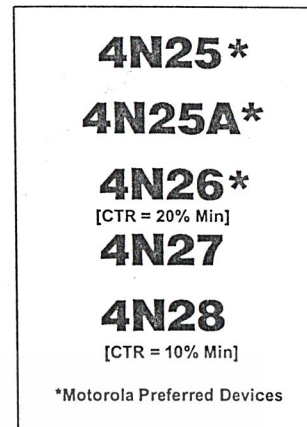
TOTAL DEVICE

Isolation Surge Voltage ⁽¹⁾ (Peak ac Voltage, 60 Hz, 1 sec Duration)	V _{ISO}	7500	Vac(pk)
Total Device Power Dissipation @ T _A = 25°C Derate above 25°C	P _D	250	mW
		2.94	mW/°C
Ambient Operating Temperature Range ⁽²⁾	T _A	–55 to +100	°C
Storage Temperature Range ⁽²⁾	T _{stg}	–55 to +150	°C
Soldering Temperature (10 sec, 1/16" from case)	T _L	260	°C

1. Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.
GlobalOptoisolator is a trademark of Motorola, Inc.

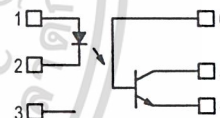
REV 5

© Motorola, Inc. 1995
Motorola Optoelectronics Device Data

STYLE 1 PLASTIC


 STANDARD THRU HOLE
 CASE 730A–04

SCHEMATIC



- PIN 1. LED ANODE
- LED CATHODE
- N.C.
- EMITTER
- COLLECTOR
- BASE


MOTOROLA



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*****
;*          Program TELEPHONE COUNTER          *
;*          Compiler : SXA51                    *
*****

        ORG 0000H

        LJMP INIT

        ORG 0023H

        LJMP SER_INT

E_LCD   BIT P1.3           ; E lcd (pin int1)
RS_LCD  BIT P1.2           ; RS lcd (pin t0)
SCL     BIT P1.0           ; Clock ds1307
SDA     BIT P1.1           ; Data ds1307
RELAY   BIT P3.5           ; Trig relay
RX_OK   BIT 0F0H           ; Flag Rx
CONT_BYTE_W EQU 0D0H       ; Control byte write
CONT_BYTE_R EQU 0D1H       ; Control byte read
CONT    EQU 8003H          ; Control port 8255
PORT_A  EQU 8000H          ; Port a
PORT_B  EQU 8001H          ; Port b
PORT_C  EQU 8002H          ; Port c
SUM_1   EQU 0005H          ; 1
SUM_10  EQU 0004H          ; 10
SUM_100 EQU 0003H          ; 100
COUNT EQU 0002H          ; Keep Flame
LAST_ADDR_L EQU 0001H      ; Low byte last address ram
LAST_ADDR_H EQU 0000H      ; High byte last address ram
SAMP    EQU 20H
SAM7    BIT SAMP.7
SAM6    BIT SAMP.6

```

SAM5	BIT	SAMP.5
SAM4	BIT	SAMP.4
SAM3	BIT	SAMP.3
SAM2	BIT	SAMP.2
SAM1	BIT	SAMP.1
SAM0	BIT	SAMP.0
FLAG	EQU	21H
RINGING_TONE	BIT	FLAG.0
OFF_HOOK	BIT	FLAG.1
RINGBACK_TONE	BIT	FLAG.2
BUSY_TONE	BIT	FLAG.3
VOICE	BIT	FLAG.4
MOBILE	BIT	FLAG.5
PAY	BIT	FLAG.6
KEEP	BIT	FLAG.7
ORG	30H	
ADD_LOW:	DS	1
DATA:	DS	1
SEC:	DS	1
MIN:	DS	1
HOUR:	DS	1
DAY:	DS	1
DATE:	DS	1
MONTH:	DS	1
YEAR:	DS	1
PTR_MONTH:	DS	1
PTR_DAY:	DS	1
PTR_ASC_D:	DS	1
PTR_DATA_D:	DS	1
PTR_ASC_K:	DS	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PTR_DATA_K:    DS    1
PTR_2ND:       DS    1
```

```
ORG    40H
```

```
KEY:          DS    1
SUM_R_1:      DS    1
SUM_R_10:     DS    1
SUM_R_100:    DS    1
NUM1:         DS    1
NUM2:         DS    1
NUM3:         DS    1
NUM4:         DS    1
NUM5:         DS    1
NUM6:         DS    1
NUM7:         DS    1
NUM8:         DS    1
NUM9:         DS    1
HEX1:         DS    1
HEX2:         DS    1
TX_DATA:      DS    1
```

```
ORG    50H
```

```
HEX3:        DS    1
```

```
INIT:        MOV    SP,#256-32    ; Initial Stack 32 Byte
              CLR    RELAY
              SETB   SCL          ; Initial clock ds1307
              SETB   SDA          ; Initial data ds1307
              MOV    ADD_LOW,#0
              MOV    DATA,#0
              MOV    SEC,#0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV  MIN,#0
MOV  HOUR,#0
MOV  DAY,#0
MOV  DATE,#0
MOV  MONTH,#0
MOV  YEAR,#0
MOV  PTR_MONTH,#0
MOV  PTR_DAY,#0
MOV  PTR_ASC_D,#0
MOV  PTR_DATA_D,#0
MOV  PTR_ASC_K,#0
MOV  PTR_DATA_K,#0
MOV  PTR_2ND,#0
MOV  KEY,#0
MOV  SUM_R_100,#0
MOV  SUM_R_10,#0
MOV  SUM_R_1,#0
MOV  HEX1,#0
MOV  HEX2,#0
;*****
;
;          *
;*****
LCALL INIT_LCD_MO          ; Initial LCD for Monitor
MOV  R0,#0
MOV  DPTR,#TITLE
MOV  A,#0FH
LCALL ADDR_LCD
MO1:  CLR  A
      MOVC A,@A+DPTR
      LCALL WR_LCD
      LCALL DELAY_500mS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC DPTR
INC R0
CJNE R0,#33,MO1

;*****
;
;           MAIN_1           *
;*****
LCALL INIT_LCD           ; Initial LCD for Normal
LCALL PW_DELAY
MOV A,#9BH           ;Set 8255 Port A,B,C i/p
MOV DPTR,#CONT
MOVX @DPTR,A
MAIN1:
LCALL MAIN_SHOW
MOV DPTR,#PORT_A
MOVX A,@DPTR
JB ACC.4,MAIN1           ;1 Check Hand set On/Off
LCALL DELAY_1S
MOV DPTR,#PORT_A
MOVX A,@DPTR
JNB ACC.4,ON_R
LJMP MAIN1

ON_R:
SETB RELAY
LCALL DELAY_500mS

CHK_DIAL1:
MOV DPTR,#PORT_A
MOVX A,@DPTR
JB ACC.1,TEL_OUT           ; Check Dial
LCALL CLR_LCD
MOV R0,#0
MOV DPTR,#SCR_TEL_IN

TEL1:
CLR A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV C  A,@A+DPTR
LCALL WR_LCD
INC  DPTR
INC  R0
CJNE R0,#16,TEL1
CLR  RELAY
MOV  DPTR,#PORT_A
H1:  MOVX A,@DPTR
      JNB ACC.4,H1
      LCALL DELAY_500mS
H2:  MOVX A,@DPTR
      JNB ACC.4,H2
      LCALL CLR_LCD
      LJMP MAIN1
TEL_OUT: LCALL CLR_LCD
        MOV  NUM1,#'X'
        MOV  NUM2,#'X'
        MOV  NUM3,#'X'
        MOV  NUM4,#'X'
        MOV  NUM5,#'X'
        MOV  NUM6,#'X'
        MOV  NUM7,#'X'
        MOV  NUM8,#'X'
        MOV  NUM9,#'X'
        MOV  R0,#0
        MOV  FLAG,#0
        MOV  SAMP,#0
        MOV  DPTR,#SCR_TEL_OUT
OUT:  CLR  A
      MOV C  A,@A+DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,OUT

```

```

MOV A,#40H
LCALL ADDR_LCD

```

DTMF1:

```

MOV KEY,#0
MOV DPTR,#PORT_B
MOVX A,@DPTR
LCALL SCAN_DTMF
MOV DPTR,#PORT_A
MOVX A,@DPTR
JNB ACC.4,N1

```

N1:

```

LJMP HC1
MOV R2,KEY
CJNE R2,#1,DTMF1
MOV A,PTR_ASC_D
LCALL WR_LCD
MOV A,PTR_ASC_D
MOV NUM1,A ;Get Num1
MOV A,PTR_DATA_D
CJNE A,#0,SP1

```

```

LCALL GET_NUM0
JB OFF_HOOK,GOTO_H
JNB MOBILE,H4
CLR MOBILE
MOV DPTR,#PORT_A

```

H3:

```

MOVX A,@DPTR
JNB ACC.4,H3

```

GOTO_H: LJMP HC1

```

SP1:          LCALL GET_NUM1
              JB  OFF_HOOK,HC1
              LCALL CHK_NUM1
              MOV  A,#4AH
              LCALL ADDR_LCD
              MOV  A,#'*'
              LCALL WR_LCD
              LCALL DELAY_1S
              LCALL DELAY_1S
              LCALL DELAY_1S
              LCALL DELAY_1S
              LCALL DELAY_1S
              MOV  DPTR,#PORT_A
              MOVX A,@DPTR
              JNB  ACC.4,GO
              LJMP HC1

GO:           JB  PAY,P4
              MOV  DPTR,#PORT_A

H10:         MOVX A,@DPTR
              JNB  ACC.4,H10
              LJMP HC1

P4:          CLR  PAY
              LCALL KEEP1
              MOV  A,#'$'
              LCALL WR_LCD
              MOV  DPTR,#PORT_A

H11:         MOVX A,@DPTR
              JNB  ACC.4,H11
              LCALL DELAY_500mS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR,#PORT_A
H12: MOVX A,@DPTR
      JNB ACC.4,H12
      LCALL KEEP2
      LJMP HC2

H4:   LCALL SAMPING
      JB OFF_HOOK,HC1
      LCALL CHK_COUNT
      JB OFF_HOOK,HC1

      MOV A,#4DH
      LCALL ADDR_LCD
      MOV A,SAMP
      LCALL SHOWASC

      MOV SAMP,#0
      MOV R0,#1
      JB RINGBACK_TONE,RINGBACK
      JB BUSY_TONE,GOTO_B
      JB VOICE,GOTO_E

HC1:  CLR OFF_HOOK
      CLR MOBILE
      LCALL DELAY_500mS
      MOV DPTR,#PORT_A
      MOVX A,@DPTR
      JNB ACC.4,CHK_DIAL2

HC2:  CLR RELAY
      LCALL CLR_LCD
      LJMP MAIN1

```

```

CHK_DIAL2:      LJMP  CHK_DIAL1
GOTO_B:         LJMP  BUSY
GOTO_E:         LJMP  EXTRA

RINGBACK:      MOV   A,#4AH
                LCALL ADDR_LCD
                MOV   A,#'*'
                LCALL WR_LCD

RINGBACK1:     CLR   RINGBACK_TONE
                LCALL SAMPING
                JB    OFF_HOOK,HC1
                LCALL CHK_COUNT
                JB    OFF_HOOK,HC1

                MOV   A,#4DH
                LCALL ADDR_LCD
                MOV   A,SAMP
                LCALL SHOWASC

                MOV   SAMP,#0
                MOV   DPTR,#PORT_A
                MOVX  A,@DPTR
                JNB   ACC.4,CHK_N
                LJMP  HC1

CHK_N:         JB    RINGBACK_TONE,RINGBACK1
                JB    VOICE,AIS
                LCALL SAMPING
                JB    OFF_HOOK,HC1
                LCALL CHK_COUNT
                JB    OFF_HOOK,HC1

```

```

MOV A,#4DH
LCALL ADDR_LCD
MOV A,SAMP
LCALL SHOWASC

MOV SAMP,#0
JB BUSY_TONE,BUSY

```

```

AIS: CLR VOICE
      LCALL KEEP1
      MOV A,#4BH
      LCALL ADDR_LCD
      MOV A,#$'
      LCALL WR_LCD
      MOV DPTR,#PORT_A
H5:  MOVX A,@DPTR
      JNB ACC.4,H5
      LCALL DELAY_500mS
      MOV DPTR,#PORT_A
H6:  MOVX A,@DPTR
      JNB ACC.4,H6
      LCALL KEEP2
      LJMP HC2

```

```

BUSY: CLR BUSY_TONE
      MOV A,#00H
      LCALL ADDR_LCD
      MOV R0,#0
      MOV DPTR,#SCR_BUSY

```

```

BUSY_1: CLR A
        MOVC A,@A+DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL WR_LCD
        INC DPTR
        INC R0
        CJNE R0,#16,BUSY_1
        MOV DPTR,#PORT_A
H7:     MOVX A,@DPTR
        JNB ACC.4,H7
GOTO_HC1: LJMP HC1

EXTRA:  CLR VOICE
        MOV A,#4AH
        LCALL ADDR_LCD
        MOV A,#+'
        LCALL WR_LCD
WAIT:   MOV KEY,#0
        MOV DPTR,#PORT_B
        MOVX A,@DPTR
        LCALL SCANKEY
        MOV DPTR,#PORT_A
        MOVX A,@DPTR
        JNB ACC.4,NEXT2
        LJMP HC1
NEXT2:  MOV A,KEY
        CJNE A,#1,WAIT
        LCALL KEEP1
        MOV A,#'$'
        LCALL WR_LCD
        MOV DPTR,#PORT_A
H8:     MOVX A,@DPTR
        JNB ACC.4,H8
        LCALL DELAY_500mS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR,#PORT_A
H9: MOVX A,@DPTR
JNB ACC.4,H9
LCALL KEEP2
LJMP HC2

;*****
; Check Telephone Counter *
;*****
CHK_COUNT: MOV A,SAMP
CJNE A,#0C0H,RI1
SETB RINGBACK_TONE
LJMP END_COUNT
RI1: CJNE A,#80H,BU1
SETB RINGBACK_TONE
LJMP END_COUNT
BU1: CJNE A,#0AAH,BU2
SETB BUSY_TONE
LJMP END_COUNT
BU2: CJNE A,#0BFH,BU3
SETB BUSY_TONE
LJMP END_COUNT
BU3: CJNE A,#0F7H,BU4
SETB BUSY_TONE
LJMP END_COUNT
BU4: CJNE A,#8EH,MO_PCT
SETB BUSY_TONE
LJMP END_COUNT
MO_PCT: SETB VOICE
END_COUNT: MOV DPTR,#PORT_A
MOVX A,@DPTR
JNB ACC.4,END_COUNT1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SETB OFF_HOOK

END_COUNT1:    RET

;*****
;          Keep Telephone Num & Time start          *
;          & Increment Sum                          *
;*****

KEEP1:        PUSH 01H
              PUSH 02H
              PUSH 03H
              PUSH 04H
              PUSH DPH
              PUSH DPL
              MOV  R2,#0
              MOV  DPTR,#LAST_ADDR_H
              MOVX A,@DPTR
              MOV  R3,A          ; Get1 last addr_high
              MOV  DPTR,#LAST_ADDR_L
              MOVX A,@DPTR
              MOV  R4,A          ; Get1 last addr_low
              MOV  DPH,R3
              MOV  DPL,R4          ; DPTR=0006H
              ;MOV  R1,#NUM1      ; RAM_ADDR_44H
              MOV  R1,#044H

LOOP_KEEP:    MOV  A,@R1
              MOVX @DPTR,A      ; 9 Number
              INC  R1
              INC  R2
              INC  DPTR
              CJNE R2,#9,LOOP_KEEP
              MOV  ADD_LOW,#04H

              LCALL READ_BYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL CHANGE
MOV A,HEX1
MOVX @DPTR,A ; Date
INC DPTR
MOV A,HEX2
MOVX @DPTR,A ; Date
INC DPTR
MOV ADD_LOW,#05H
LCALL READ_BYTE
LCALL CHANGE
MOV A,HEX1
MOVX @DPTR,A ; Month
INC DPTR
MOV A,HEX2
MOVX @DPTR,A ; Month
INC DPTR
MOV ADD_LOW,#06H
LCALL READ_BYTE
LCALL CHANGE
MOV A,HEX1
MOVX @DPTR,A ; Year
INC DPTR
MOV A,HEX2
MOVX @DPTR,A ; Year
INC DPTR
MOV ADD_LOW,#02H ; Start time
LCALL READ_BYTE
LCALL CHANGE
MOV A,HEX1
MOVX @DPTR,A ; Hour
INC DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV  A,HEX2
MOVX @DPTR,A          ; Hour
INC  DPTR
MOV  ADD_LOW,#01H
LCALL READ_BYTE
LCALL CHANGE
MOV  A,HEX1
MOVX @DPTR,A          ; Min
INC  DPTR
MOV  A,HEX2
MOVX @DPTR,A          ; Min
INC  DPTR
MOV  ADD_LOW,#00H
LCALL READ_BYTE
LCALL CHANGE
MOV  A,HEX1
MOVX @DPTR,A          ; Sec
INC  DPTR
MOV  A,HEX2
MOVX @DPTR,A          ; Sec
INC  DPTR
MOV  R3,DPH
MOV  R4,DPL
MOV  DPTR,#LAST_ADDR_H
MOV  A,R3
MOVX @DPTR,A          ;Keep last addr. of ram
MOV  DPTR,#LAST_ADDR_L
MOV  A,R4
MOVX @DPTR,A
LCALL INC_SUM
POP  DPL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP   DPH
POP   04H
POP   03H
POP   02H
POP   01H
RET

```

```

;*****
;
;           Keep Time stop           *
;*****

```

```

KEEP2:    PUSH   03H
           PUSH   04H
           PUSH   DPH
           PUSH   DPL
           MOV    DPTR,#LAST_ADDR_H
           MOVX   A,@DPTR
           MOV    R3,A           ; Get2 last addr_high
           MOV    DPTR,#LAST_ADDR_L
           MOVX   A,@DPTR
           MOV    R4,A           ; Get2 last addr_low
           MOV    DPH,R3
           MOV    DPL,R4
           MOV    ADD_LOW,#02H   ; Stop time
           LCALL  READ_BYTE
           LCALL  CHANGE
           MOV    A,HEX1
           MOVX   @DPTR,A       ; Hour
           INC    DPTR
           MOV    A,HEX2
           MOVX   @DPTR,A       ; Hour
           INC    DPTR
           MOV    ADD_LOW,#01H

```

```

LCALL READ_BYTE
LCALL CHANGE
MOV A,HEX1
MOVX @DPTR,A ; Min
INC DPTR
MOV A,HEX2
MOVX @DPTR,A ; Min
INC DPTR
MOV ADD_LOW,#00H
LCALL READ_BYTE
LCALL CHANGE
MOV A,HEX1
MOVX @DPTR,A ; Sec
INC DPTR
MOV A,HEX2
MOVX @DPTR,A ; Sec
INC DPTR
MOV R3,DPH
MOV R4,DPL
MOV DPTR,#LAST_ADDR_H
MOV A,R3
MOVX @DPTR,A ;Keep2 last addr. of ram
MOV DPTR,#LAST_ADDR_L
MOV A,R4
MOVX @DPTR,A
MOV DPTR,#COUNT
MOVX A,@DPTR
INC A ;Counter Flame of data
MOVX @DPTR,A
POP DPL
POP DPH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP 04H
POP 03H
RET

```

```

;*****
;          Get Telephone Number ;If Start NUM=>(0)      *
;*****

```

```

GET_NUM0:      PUSH 02H
DTMF20:       MOV  KEY,#1
              MOV  DPTR,#PORT_B
              MOVX A,@DPTR
              LCALL SCAN_DTMF
              MOV  A,PTR_DATA_D
              CJNE A,#1,AREA
              SETB MOBILE
AREA:         MOV  DPTR,#PORT_A
              MOVX A,@DPTR
              JNB  ACC.4,N2
              SETB OFF_HOOK
              LJMP HAND_D0
N2:          MOV  R2,KEY
              CJNE R2,#2,DTMF20
              MOV  A,PTR_ASC_D
              LCALL WR_LCD
              MOV  A,PTR_ASC_D
              MOV  NUM2,A          ;Get Phone Num2
DTMF30:      MOV  KEY,#2
              MOV  DPTR,#PORT_B
              MOVX A,@DPTR
              LCALL SCAN_DTMF
              MOV  DPTR,#PORT_A
              MOVX A,@DPTR

```

```

JNB ACC.4,N3
SETB OFF_HOOK
LJMP HAND_D0
N3:
MOV R2,KEY
CJNE R2,#3,DTMF30
MOV A,PTR_ASC_D
LCALL WR_LCD
MOV A,PTR_ASC_D
MOV NUM3,A ;Get Phone Num3
DTMF40:
MOV KEY,#3
MOV DPTR,#PORT_B
MOVX A,@DPTR
LCALL SCAN_DTMF
MOV DPTR,#PORT_A
MOVX A,@DPTR
JNB ACC.4,N4
SETB OFF_HOOK
LJMP HAND_D0
N4:
MOV R2,KEY
CJNE R2,#4,DTMF40
MOV A,PTR_ASC_D
LCALL WR_LCD
MOV A,PTR_ASC_D
MOV NUM4,A ;Get Phone Num4
DTMF50:
MOV KEY,#4
MOV DPTR,#PORT_B
MOVX A,@DPTR
LCALL SCAN_DTMF
MOV DPTR,#PORT_A
MOVX A,@DPTR
JNB ACC.4,N5

```

```

SETB OFF_HOOK
LJMP HAND_D0
N5: MOV R2,KEY
CJNE R2,#5,DTMF50
MOV A,PTR_ASC_D
LCALL WR_LCD
MOV A,PTR_ASC_D
MOV NUM5,A ;Get Phone Num5
DTMF60: MOV KEY,#5
MOV DPTR,#PORT_B
MOVX A,@DPTR
LCALL SCAN_DTMF
MOV DPTR,#PORT_A
MOVX A,@DPTR
JNB ACC.4,N6
SETB OFF_HOOK
LJMP HAND_D0
N6: MOV R2,KEY
CJNE R2,#6,DTMF60
MOV A,PTR_ASC_D
LCALL WR_LCD
MOV A,PTR_ASC_D
MOV NUM6,A ;Get Phone Num6
DTMF70: MOV KEY,#6
MOV DPTR,#PORT_B
MOVX A,@DPTR
LCALL SCAN_DTMF
MOV DPTR,#PORT_A
MOVX A,@DPTR
JNB ACC.4,N7
SETB OFF_HOOK

```

```

N7:          LJMP  HAND_D0
            MOV   R2,KEY
            CJNE  R2,#7,DTMF70
            MOV   A,PTR_ASC_D
            LCALL WR_LCD
            MOV   A,PTR_ASC_D
            MOV   NUM7,A           ;Get Phone Num7

DTMF80:     MOV   KEY,#7
            MOV   DPTR,#PORT_B
            MOVX  A,@DPTR
            LCALL SCAN_DTMF
            MOV   DPTR,#PORT_A
            MOVX  A,@DPTR
            JNB   ACC.4,N8
            SETB  OFF_HOOK
            LJMP  HAND_D0

N8:          MOV   R2,KEY
            CJNE  R2,#8,DTMF80
            MOV   A,PTR_ASC_D
            LCALL WR_LCD
            MOV   A,PTR_ASC_D
            MOV   NUM8,A           ;Get Phone Num8

DTMF90:     MOV   KEY,#8
            MOV   DPTR,#PORT_B
            MOVX  A,@DPTR
            LCALL SCAN_DTMF
            MOV   DPTR,#PORT_A
            MOVX  A,@DPTR
            JNB   ACC.4,N9
            SETB  OFF_HOOK
            LJMP  HAND_D0

```

```

N9:          MOV  R2,KEY
            CJNE R2,#9,DTMF90
            MOV  A,PTR_ASC_D
            LCALL WR_LCD
            MOV  A,PTR_ASC_D
            MOV  NUM9,A          ;Get Phone Num9

HAND_D0:    POP  02H
            RET

;*****
;          Get Telephone Number ;If Start NUM=>(1) *
;*****
GET_NUM1:   PUSH  01H
            PUSH  02H
            MOV  HEX3,#0
            MOV  R1,#NUM2      ;RAM_ADDR_45H

DTMF_1:    MOV  KEY,#0
            MOV  DPTR,#PORT_B
            MOVX A,@DPTR
            LCALL SCAN_DTMF
            MOV  DPTR,#PORT_A
            MOVX A,@DPTR
            JNB  ACC.4,WI1
            SETB OFF_HOOK
            LJMP HAND_D1

WI1:       MOV  A,KEY
            CJNE A,#1,DTMF_1
            MOV  A,PTR_ASC_D
            LCALL WR_LCD
            MOV  A,PTR_ASC_D
            MOV  @R1,A          ;Get Phone Num2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC HEX3
INC R1
MOV A,HEX3
CJNE A,#4,WI2
LJMP HAND_D1
WI2: CJNE A,#3,DTMF_1
HAND_D1: POP 02H
POP 01H
RET

```

```

;*****
; Check Telephone Number ;If Start NUM=>(1) *
;*****

```

```

CHK_NUM1: MOV A,NUM2
CJNE A,#1,NUMB4
MOV A,NUM3
CJNE A,#2,NUMB_1141_2_4
MOV A,NUM4
CJNE A,#5,NO1
SETB PAY
NO1: LJMP NO_PAY

```

```

NUMB_1141_2_4: CJNE A,#4,NUMB_1187_8
MOV A,NUM4
CJNE A,#1,NUMB1142_4
SETB PAY
LJMP NO_PAY

```

```

NUMB1142_4: CJNE A,#2,NUMB1144
SETB PAY
LJMP NO_PAY

```

```

NUMB1144: CJNE A,#4,NO2
SETB PAY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NO2:                LJMP NO_PAY

NUMB_1187_8:        CJNE A,#8,NO3
                    MOV  A,NUM4
                    CJNE A,#7,NUMB1188
                    SETB PAY
                    LJMP NO_PAY

NUMB1188:           CJNE A,#8,NO3
                    SETB PAY

NO3:                LJMP NO_PAY

NUMB4:              CJNE A,#4,NUMB5
                    MOV  A,NUM3
                    CJNE A,#1,NUMB_142
                    SETB PAY
                    LJMP NO_PAY

NUMB_142:           CJNE A,#2,NO4
                    SETB PAY

NO4:                LJMP NO_PAY

NUMB5:              CJNE A,#5,NUMB6
                    MOV  A,NUM3
                    CJNE A,#0,NUMB_151_152_153_1567
                    MOV  A,NUM4
                    CJNE A,#0,NUMB1506_7
                    SETB PAY
                    LJMP NO_PAY

NUMB1506_7:         CJNE A,#6,NUMB1507
                    SETB PAY
                    LJMP NO_PAY

NUMB1507:           CJNE A,#7,NO5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                SETB PAY
NO5:            LJMP NO_PAY

NUMB_151_152_153_1567:  CJNE A,#1,NUMB_152_153_1567
                SETB PAY
                LJMP NO_PAY

NUMB_152_153_1567: CJNE A,#2,NUMB_153_1567
                SETB PAY
                LJMP NO_PAY

NUMB_153_1567:  CJNE A,#3,NUMB_1567
                SETB PAY
                LJMP NO_PAY

NUMB_1567:      CJNE A,#6,NO6
                MOV  A,NUM4
                CJNE A,#7,NO6
                SETB PAY
NO6:            LJMP NO_PAY

NUMB6:          CJNE A,#6,NUMB8
                MOV  A,NUM3
                CJNE A,#1,NUMB_162_163_1665
                SETB PAY
                LJMP NO_PAY

NUMB_162_163_1665:  CJNE A,#2,NUMB_163_1665
                SETB PAY
                LJMP NO_PAY

```

```

NUMB_163_1665:   CJNE  A,#3,NUMB_1665
                  SETB  PAY
                  LJMP  NO_PAY

```

```

NUMB_1665:       CJNE  A,#6,NO7
                  MOV   A,NUM4
                  CJNE  A,#5,NO7
                  SETB  PAY

```

```

NO7:             LJMP  NO_PAY

```

```

NUMB8:           CJNE  A,#8,NO8
                  MOV   A,NUM3
                  CJNE  A,#1,NUMB_182_4
                  SETB  PAY

```

```

NO8:             LJMP  NO_PAY

```

```

NUMB_182_4:     CJNE  A,#2,NUMB_184
                  SETB  PAY
                  LJMP  NO_PAY

```

```

NUMB_184:       CJNE  A,#4,NO_PAY
                  SETB  PAY

```

```

NO_PAY:         RET

```

```

;*****
;

```

```

;           Sampling Signal           *

```

```

;*****
;

```

```

SAMPING:        PUSH  01H

```

```

SA1:            MOVX  A,@DPTR

```

```

                JNB  ACC.1,SA1

```

```

                LCALL DELAY_50mS

```

```

                MOV  R1,#0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SA2:      MOV  DPTR,#PORT_A
          MOVX A,@DPTR
          INC  R1
          CJNE R1,#1,BIT_6
          JNB  ACC.1,Z1
          SETB SAM7
          LJMP DEL_S

Z1:       CLR  SAM7
          LJMP DEL_S

BIT_6:    CJNE R1,#2,BIT_5
          JNB  ACC.1,Z2
          SETB SAM6
          LJMP DEL_S

Z2:       CLR  SAM6
          LJMP DEL_S

BIT_5:    CJNE R1,#3,BIT_4
          JNB  ACC.1,Z3
          SETB SAM5
          LJMP DEL_S

Z3:       CLR  SAM5
          LJMP DEL_S

BIT_4:    CJNE R1,#4,BIT_3
          JNB  ACC.1,Z4
          SETB SAM4
          LJMP DEL_S

Z4:       CLR  SAM4
          LJMP DEL_S

```

```

BIT_3:      CJNE  R1,#5,BIT_2
             JNB   ACC.1,Z5
             SETB  SAM3
             LJMP  DEL_S
Z5:         CLR   SAM3
             LJMP  DEL_S

BIT_2:      CJNE  R1,#6,BIT_1
             JNB   ACC.1,Z6
             SETB  SAM2
             LJMP  DEL_S
Z6:         CLR   SAM2
             LJMP  DEL_S

BIT_1:      CJNE  R1,#7,BIT_0
             JNB   ACC.1,Z7
             SETB  SAM1
             LJMP  DEL_S
Z7:         CLR   SAM1
             LJMP  DEL_S

BIT_0:      CJNE  R1,#8,END_BIT
             JNB   ACC.1,Z8
             SETB  SAM0
             LJMP  END_BIT
Z8:         CLR   SAM0
             LJMP  END_BIT

DEL_S:      MOV   DPTR,#PORT_A
             MOVX  A,@DPTR
             JNB   ACC.4,AK
             SETB  OFF_HOOK

```

```

AK:          LCALL DELAY_500mS
            LJMP SA2

END_BIT:     POP  01H

            RET

```

```

;*****
;
;          MAIN SHOW          *
;*****
MAIN_SHOW:  MOV  A,#00H          ;FIRST LINE
            LCALL ADDR_LCD
            MOV  ADD_LOW,#02H
            LCALL READ_BYTE
            MOV  HOUR,DATA
            LCALL SHOWASC
            MOV  A,#':'
            LCALL WR_LCD
            MOV  ADD_LOW,#01H
            LCALL READ_BYTE
            MOV  MIN,DATA
            LCALL SHOWASC
            MOV  A,#':'
            LCALL WR_LCD
            MOV  ADD_LOW,#00H
            LCALL READ_BYTE
            MOV  SEC,DATA
            LCALL SHOWASC
            MOV  A,#' '
            LCALL WR_LCD
            MOV  A,#'S'
            LCALL WR_LCD
            MOV  A,#'U'
            LCALL WR_LCD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,#M'
LCALL WR_LCD
MOV A,#='
LCALL WR_LCD
LCALL SHOW_SUM

MOV A,#40H ;SECOND LINE
LCALL ADDR_LCD
MOV ADD_LOW,#03H
LCALL READ_BYTE
MOV DAY,DATA
LCALL WRITE_DAY
MOV A,#'
LCALL WR_LCD
MOV ADD_LOW,#04H
LCALL READ_BYTE
MOV DATE,DATA
LCALL SHOWASC
MOV A,#/
LCALL WR_LCD
MOV ADD_LOW,#05H
LCALL READ_BYTE
MOV MONTH,DATA
LCALL WRITE_MON
MOV A,#/
LCALL WR_LCD
MOV ADD_LOW,#06H
LCALL READ_BYTE
MOV YEAR,DATA
LCALL SHOWASC
MOV A,#'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL WR_LCD
MOV A,#' '
LCALL WR_LCD
MOV A,#' '
LCALL WR_LCD

MOV DPTR,#PORT_C ; Check Set
MOVX A,@DPTR
MOV KEY,#0
LCALL SCANKEY
MOV R7,PTR_DATA_K
CJNE R7,#1FH,CHK_END
MOV PTR_DATA_K,#0
LCALL CLR_LCD
MOV R0,#0
MOV DPTR,#SCR_SELECT
CLR A
MOVC A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,MS1
LCALL DELAY_1S
LCALL SELECT1

CHK_END: RET

;*****
;*      Select      *
;*      i/p var     *
;*****

SELECT1: LCALL SHOW1

S1:      MOV KEY,#0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#1,S1
MOV R2,PTR_DATA_K
CJNE R2,#13H,CHK_D
LCALL SHOW1
LJMP S1
CHK_D: CJNE R2,#17H,CHK_SET1
LCALL SHOW2
LJMP S1
CHK_SET1: CJNE R2,#1,CHK_MEM1
LCALL SET_MODE
LJMP SEL_END1
CHK_MEM1: CJNE R2,#2,CHK_TX1
LCALL MEM_MODE
LJMP SEL_END1
CHK_TX1: CJNE R2,#3,SEL_CLR1
LCALL TX_MODE
LJMP SEL_END1
SEL_CLR1: CJNE R2,#1CH,S1
SEL_END1: MOV PTR_DATA_K,#0
POP 02H
POP 00H
RET

SHOW1: PUSH 00H
MOV R0,#0
LCALL CLR_LCD
MOV R0,#0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR,#SCR_SELECT3
SH1: CLR A
      MOVC A,@A+DPTR
      LCALL WR_LCD
      INC DPTR
      INC R0
      CJNE R0,#16,SH1
      MOV A,#40H
      LCALL ADDR_LCD
      MOV R0,#0
      MOV DPTR,#SCR_SELECT4
SH2: CLR A
      MOVC A,@A+DPTR
      LCALL WR_LCD
      INC DPTR
      INC R0
      CJNE R0,#16,SH2
      POP 00H
      RET
SHOW2: PUSH 00H
        MOV R0,#0
        LCALL CLR_LCD
        MOV R0,#0
        MOV DPTR,#SCR_SELECT4
SH3: CLR A
      MOVC A,@A+DPTR
      LCALL WR_LCD
      INC DPTR
      INC R0
      CJNE R0,#16,SH3

```

```

MOV A,#40H
LCALL ADDR_LCD
MOV R0,#0
MOV DPTR,#SCR_SELECT5
SH4: CLR A
      MOVC A,@A+DPTR
      LCALL WR_LCD
      INC DPTR
      INC R0
      CJNE R0,#16,SH4
      POP 00H
      RET
;*****
;*      Clear Data Memory      *
;*****
MEM_MODE: PUSH 00H
          PUSH 02H
          LCALL BLINK_LCD
          LCALL CLR_LCD
          MOV R0,#0
          MOV DPTR,#SCR_MEM_MODE
ME1:     CLR A
        MOVC A,@A+DPTR
        LCALL WR_LCD
        INC DPTR
        INC R0
        CJNE R0,#14,ME1
        MOV A,#0
        MOV R2,#0
        MOV DPTR,#COUNT
CL1:     MOVX @DPTR,A

```

```

INC DPTR
INC R2
CJNE R2,#4,CL1
MOV R2,#0
MOV A,#0FFH
CL2: MOVX @DPTR,A
INC DPTR
MOV R2,DPH
CJNE R2,#80H,CL2
MOV A,#00H
MOV DPTR,#LAST_ADDR_H
MOVX @DPTR,A
MOV A,#06H
MOV DPTR,#LAST_ADDR_L
MOVX @DPTR,A
LCALL DELAY_1S
MOV A,#40H
LCALL ADDR_LCD
MOV R0,#0
MOV DPTR,#SCR_COMPLETE
ME2: CLR A
MOVC A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,ME2
LCALL DELAY_1S
LCALL ON_LCD
POP 02H
POP 00H

```

RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;*      Tranfer Data to Computer      *
;*****
TX_MODE:      PUSH  00H
               PUSH  01H
               LCALL BLINK_LCD
               LCALL CLR_LCD
               MOV   R0,#0
               MOV   DPTR,#SCR_TX_MODE
TX1:          CLR   A
               MOVC  A,@A+DPTR
               LCALL WR_LCD
               INC   DPTR
               INC   R0
               CJNE  R0,#11,TX1
               MOV   TMOD,#21H
               MOV   TH1,#0FBH
               MOV   TL1,#0FBH
               MOV   IE,#10010000B
               SETB  TR1
               MOV   SCON,#50H
               MOV   DPTR,#COUNT
               MOVX  A,@DPTR
               MOV   R1,A
               MOV   DPTR,#0006H
LOOP:         CLR   RX_OK
               JNB  RX_OK,$
               LCALL TX_TEXT
               DJNZ  R1,LOOP
               MOV   A,#40H
               LCALL ADDR_LCD

```

```

MOV R0,#0
MOV DPTR,#SCR_COMPLETE
TX2: CLR A
MOV A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,TX2
LCALL DELAY_1S
LCALL ON_LCD
POP 01H
POP 00H
RET
;*****
;* Set time & date *
;* i/p var *
;*****
SET_MODE: PUSH 00H
PUSH 02H
LCALL BLINK_LCD
SET_DAY: LCALL CLR_LCD
MOV R0,#0 ;SET DAY
MOV DPTR,#SCR_SET
SE1: CLR A
MOV A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,SE1

```

```
MOV A,#40H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL ADDR_LCD
MOV R0,#0
MOV DPTR,#SCR_SET_DAY
SE2: CLR A
MOV A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,SE2

```

```

MOV A,#47H
LCALL ADDR_LCD
MOV R0,#0
MOV R2,#0
SE3: MOV KEY,#0
MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#1,SE3
MOV A,PTR_DATA_K
CJNE A,#00H,SE3

```

```

MOV A,PTR_ASC_K
LCALL WR_LCD
LCALL DELAY_500mS
MOV A,PTR_DATA_K
SWAP A
MOV R2,A
MOV KEY,#1

```

SE4:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#2,SE4

MOV A,PTR_DATA_K
CJNE A,#01H,MON
LJMP OK1
MON: CJNE A,#02H,TUE
LJMP OK1
TUE: CJNE A,#03H,WED
LJMP OK1
WED: CJNE A,#04H,THU
LJMP OK1
THU: CJNE A,#05H,FRI
LJMP OK1
FRI: CJNE A,#06H,SAT
LJMP OK1
SAT: CJNE A,#07H,SE4
OK1: MOV A,PTR_ASC_K
LCALL WR_LCD
LCALL DELAY_500mS
MOV A,PTR_DATA_K
ORL A,R2
MOV DAY,A

LCALL CLR_LCD
MOV R0,#0
MOV DPTR,#SCR_CONFIRM1
CON1: CLR A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVC A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,CON1

```

```

MOV A,#40H
LCALL ADDR_LCD
MOV R0,#0
MOV DPTR,#SCR_CONFIRM2

```

CON2:

```

CLR A
MOVC A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,CON2

```

CON3:

```

MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV A,PTR_DATA_K
CJNE A,#1FH,CON4
LJMP SET_DATE

```

CON4:

```

CJNE A,#1CH,CON3
LJMP SET_DAY

```

```

SET_DATE:    LCALL CLR_LCD           ;SET DATE

```

```

MOV R0,#0
MOV DPTR,#SCR_SET

```

SE21:

```

CLR A
MOVC A,@A+DPTR
LCALL WR_LCD

```

```

INC DPTR
INC R0
CJNE R0,#16,SE21

```

```

MOV A,#40H
LCALL ADDR_LCD
MOV R0,#0
MOV DPTR,#SCR_SET_DATE

```

SE5:

```

CLR A
MOVC A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,SE5
MOV A,#47H
LCALL ADDR_LCD

```

SE6:

```

MOV R0,#0
MOV R2,#0
MOV KEY,#0
MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#1,SE6

```

DATE10:

```

MOV A,PTR_DATA_K
CJNE A,#00H,DATE10
LJMP OK2
CJNE A,#01H,DATE20
LJMP OK2

```

```

DATE20:      CJNE  A,#02H,DATE30
              LJMP  OK2

DATE30:      CJNE  A,#03H,SE6

OK2:         MOV   A,PTR_ASC_K
              LCALL WR_LCD
              LCALL DELAY_500mS
              MOV   A,PTR_DATA_K
              SWAP  A
              MOV   R2,A
SE7:         MOV   KEY,#1
              MOV   DPTR,#PORT_C
              MOVX  A,@DPTR
              LCALL SCANKEY
              MOV   R0,KEY
              CJNE  R0,#2,SE7

              MOV   A,PTR_DATA_K
              CJNE  R2,#00H,CHK_30
              LCALL CHK_COLOR
              MOV   R0,KEY
              CJNE  R0,#2,SE7
              CJNE  A,#1DH,OK3
              LJMP  SE7

CHK_30:      CJNE  R2,#30H,CHK_10_20
              CJNE  A,#00H,ERR10
              LJMP  OK3

ERR10:       CJNE  A,#01H,SE7
              LJMP  OK3

```

```

CHK_10_20:      LCALL  CHK_COLOR
                MOV   R0,KEY
                CJNE  R0,#2,SE7

OK3:           MOV   A,PTR_ASC_K
                LCALL WR_LCD
                LCALL DELAY_500mS
                MOV   A,PTR_DATA_K
                ORL  A,R2
                MOV   DATE,A

SET_MONTH:     MOV   A,#4AH          ;SET MONTH
                LCALL ADDR_LCD
                MOV   R0,#0
                MOV   R2,#0
                MOV   KEY,#0
SE8:          MOV   DPTR,#PORT_C
                MOVX  A,@DPTR
                LCALL SCANKEY
                MOV   R0,KEY
                CJNE  R0,#1,SE8

                MOV   A,PTR_DATA_K
                CJNE  A,#00H,OCT_DEC
                LJMP  OK4

OCT_DEC:      CJNE  A,#01H,SE8

OK4:          MOV   A,PTR_ASC_K
                LCALL WR_LCD
                LCALL DELAY_500mS
                MOV   A,PTR_DATA_K

```

```

SWAP A
MOV R2,A
SE9: MOV KEY,#1
MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#2,SE9

MOV A,PTR_DATA_K
CJNE R2,#00H,OCT_DEC_
LCALL CHK_COLOR
MOV R0,KEY
CJNE R0,#2,SE9
CJNE A,#1DH,OK5
OCT_DEC_: CJNE A,#00H,NOV
LJMP OK5
NOV: CJNE A,#01H,DEC
LJMP OK5
DEC: CJNE A,#02H,SE9

OK5: MOV A,PTR_ASC_K
LCALL WR_LCD
LCALL DELAY_500mS
MOV A,PTR_DATA_K
ORL A,R2
MOV MONTH,A

SET_YEAR: MOV A,#4DH ;SET YEAR
LCALL ADDR_LCD

```

```

MOV R0,#0
MOV R2,#0
SE10: MOV KEY,#0
MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#1,SE10

```

```

MOV A,PTR_DATA_K
LCALL CHK_COLOR
MOV R0,KEY
CJNE R0,#1,SE10

```

```

OK7: MOV A,PTR_ASC_K
LCALL WR_LCD
LCALL DELAY_500mS
MOV A,PTR_DATA_K
SWAP A
MOV R2,A

```

```

SE11: MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#2,SE11

```

```

MOV A,PTR_DATA_K
LCALL CHK_COLOR
MOV R0,KEY
CJNE R0,#2,SE11

```

```

MOV  A,PTR_ASC_K
LCALL WR_LCD
LCALL DELAY_500mS
MOV  A,PTR_DATA_K
ORL  A,R2
MOV  YEAR,A

```

```

LCALL CLR_LCD
MOV  R0,#0
MOV  DPTR,#SCR_CONFIRM1
CON5: CLR  A
      MOVC A,@A+DPTR
      LCALL WR_LCD
      INC  DPTR
      INC  R0
      CJNE R0,#16,CON5
      MOV  A,#40H
      LCALL ADDR_LCD
      MOV  R0,#0
      MOV  DPTR,#SCR_CONFIRM2

```

```

CON6: CLR  A
      MOVC A,@A+DPTR
      LCALL WR_LCD
      INC  DPTR
      INC  R0
      CJNE R0,#16,CON6

```

```

CON7: MOV  DPTR,#PORT_C
      MOVX A,@DPTR
      LCALL SCANKEY
      MOV  A,PTR_DATA_K

```

```

                CJNE A,#1FH,CON8
                LJMP SET_TIME
CON8:          CJNE A,#1CH,CON7
                LJMP SET_DATE

SET_TIME:     LCALL CLR_LCD           ;SET_TIME
                MOV R0,#0
                MOV DPTR,#SCR_SET

SE22:         CLR A
                MOVC A,@A+DPTR
                LCALL WR_LCD
                INC DPTR
                INC R0
                CJNE R0,#16,SE22

                MOV A,#40H
                LCALL ADDR_LCD
                MOV R0,#0
                MOV DPTR,#SCR_SET_TIME

SE12:         CLR A
                MOVC A,@A+DPTR
                LCALL WR_LCD
                INC DPTR
                INC R0
                CJNE R0,#16,SE12

SET_HOUR:     MOV A,#47H           ;SET_HOUR
                LCALL ADDR_LCD
                MOV R0,#0
                MOV R2,#0

SE13:         MOV KEY,#0

```

```

MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#1,SE13

```

```

MOV A,PTR_DATA_K
CJNE A,#00H,HR_10_19
LJMP OK8

```

```

HR_10_19: CJNE A,#01H,HR_20_23
LJMP OK8

```

```

HR_20_23: CJNE A,#02H,SE13

```

```

OK8: MOV A,PTR_ASC_K
LCALL WR_LCD
LCALL DELAY_500mS
MOV A,PTR_DATA_K
SWAP A
MOV R2,A

```

```

SE14: MOV KEY,#1
MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#2,SE14

```

```

MOV A,PTR_DATA_K
CJNE R2,#00H,HR_10_19_
LCALL CHK_COLOR
MOV R0,KEY
CJNE R0,#2,SE14

```

```

                                LJMP  OK9
HR_10_19_:                      CJNE  R2,#10H,HR_20_23_
                                LCALL  CHK_COLOR
                                MOV   R0,KEY
                                CJNE  R0,#2,SE14
                                LJMP  OK9
HR_20_23_:                      CJNE  A,#00H,HR_21
                                LJMP  OK9
HR_21:                          CJNE  A,#01H,HR_22
                                LJMP  OK9
HR_22:                          CJNE  A,#02H,HR_23
                                LJMP  OK9
HR_23:                          CJNE  A,#03H,SE14
OK9:                             MOV   A,PTR_ASC_K
                                LCALL  WR_LCD
                                LCALL  DELAY_500mS
                                MOV   A,PTR_DATA_K
                                ORL   A,R2
                                MOV   HOUR,A
SET_MIN:                         MOV   A,#4AH ;SET MIN
                                LCALL  ADDR_LCD

                                MOV   R0,#0
                                MOV   R2,#0
SE15:                            MOV   KEY,#0
                                MOV   DPTR,#PORT_C
                                MOVX  A,@DPTR
                                LCALL  SCANKEY
                                MOV   R0,KEY

```

```

CJNE R0,#1,SE15

MOV A,PTR_DATA_K
CJNE A,#00H,MIN_10
LJMP OK10
MIN_10: CJNE A,#01H,MIN_20
LJMP OK10
MIN_20: CJNE A,#02H,MIN_30
LJMP OK10
MIN_30: CJNE A,#03H,MIN_40
LJMP OK10
MIN_40: CJNE A,#04H,MIN_50
LJMP OK10
MIN_50: CJNE A,#05H,SE15
OK10: MOV A,PTR_ASC_K
LCALL WR_LCD
LCALL DELAY_500mS
MOV A,PTR_DATA_K
SWAP A
MOV R2,A
SE16: MOV KEY,#1
MOV DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV R0,KEY
CJNE R0,#2,SE16

MOV A,PTR_DATA_K
LCALL CHK_COLOR

MOV R0,KEY

```

```
CJNE R0,#2,SE16
```

```
MOV A,PTR_ASC_K
```

```
LCALL WR_LCD
```

```
LCALL DELAY_500mS
```

```
MOV A,PTR_DATA_K
```

```
ORL A,R2
```

```
MOV MIN,A
```

```
SET_SEC: MOV A,#4DH ;SET SEC
```

```
LCALL ADDR_LCD
```

```
MOV R0,#0
```

```
MOV R2,#0
```

```
SE17: MOV KEY,#0
```

```
MOV DPTR,#PORT_C
```

```
MOVX A,@DPTR
```

```
LCALL SCANKEY
```

```
MOV R0,KEY
```

```
CJNE R0,#1,SE17
```

```
MOV A,PTR_DATA_K
```

```
CJNE A,#00H,SEC_10
```

```
LJMP OK11
```

```
SEC_10: CJNE A,#01H,SEC_20
```

```
LJMP OK11
```

```
SEC_20: CJNE A,#02H,SEC_30
```

```
LJMP OK11
```

```
SEC_30: CJNE A,#03H,SEC_40
```

```
LJMP OK11
```

```
SEC_40: CJNE A,#04H,SEC_50
```

```
LJMP OK11
```

```

SEC_50:      CJNE  A,#05H,SE17

OK11:       MOV   A,PTR_ASC_K
            LCALL WR_LCD
            LCALL DELAY_500mS
            MOV   A,PTR_DATA_K
            SWAP  A
            MOV   R2,A
SE18:       MOV   KEY,#1
            MOV   DPTR,#PORT_C
            MOVX  A,@DPTR
            LCALL SCANKEY
            MOV   R0,KEY
            CJNE  R0,#2,SE18
            MOV   A,PTR_DATA_K
            LCALL CHK_COLOR
            MOV   R0,KEY
            CJNE  R0,#2,SE18

            MOV   A,PTR_ASC_K
            LCALL WR_LCD
            LCALL DELAY_500mS
            MOV   A,PTR_DATA_K
            ORL   A,R2
            MOV   SEC,A

            LCALL CLR_LCD
            MOV   R0,#0
            MOV   DPTR,#SCR_CONFIRM1
CON9:       CLR   A

```

```

MOV  A,@A+DPTR
LCALL WR_LCD
INC  DPTR
INC  R0
CJNE R0,#16,CON9

```

```

MOV  A,#40H
LCALL ADDR_LCD
MOV  R0,#0
MOV  DPTR,#SCR_CONFIRM2

```

CON10:

```

CLR  A
MOV  A,@A+DPTR
LCALL WR_LCD
INC  DPTR
INC  R0
CJNE R0,#16,CON10

```

CON11:

```

MOV  DPTR,#PORT_C
MOVX A,@DPTR
LCALL SCANKEY
MOV  A,PTR_DATA_K
CJNE A,#1FH,CON12
LJMP OK

```

CON12:

```

CJNE A,#1CH,CON11
LJMP SET_TIME

```

OK:

```

LCALL CLR_LCD
MOV  R0,#0
MOV  DPTR,#SCR_SET_MODE

```

SE20:

```

CLR  A
MOV  A,@A+DPTR
LCALL WR_LCD

```

```

INC DPTR
INC R0
CJNE R0,#16,SE20
MOV A,#40H
LCALL ADDR_LCD
MOV R0,#0
MOV DPTR,#SCR_COMPLETE

SE23:
CLR A
MOVC A,@A+DPTR
LCALL WR_LCD
INC DPTR
INC R0
CJNE R0,#16,SE23
LCALL DELAY_1S
LCALL ON_LCD

MOV ADD_LOW,#00H ;SEC
MOV DATA,SEC
LCALL WRITE_BYTE
MOV ADD_LOW,#01H ;MIN
MOV DATA,MIN
LCALL WRITE_BYTE

MOV ADD_LOW,#02H ;HOUR
MOV DATA,HOUR
LCALL WRITE_BYTE

MOV ADD_LOW,#03H ;DAY
MOV DATA,DAY
LCALL WRITE_BYTE

MOV ADD_LOW,#04H ;DATE
MOV DATA,DATE
LCALL WRITE_BYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV  ADD_LOW,#05H    ;MONTH
MOV  DATA,MONTH
LCALL WRITE_BYTE
MOV  ADD_LOW,#06H    ;YEAR
MOV  DATA,YEAR
LCALL WRITE_BYTE
POP  02H
POP  00H
RET

```

```

;*****
;

```

```

;*      Scankeydtmf      *

```

```

;*      i/p acc o/p ptr_asc      *
;*****

```

```

SCAN_DTMF:  PUSH  00H
AGAIN_D:    ANL   A,#1FH
            MOV   R0,A
            CJNE R0,#11H,CHK_2
            MOV   PTR_ASC_D,#1'
            MOV   PTR_DATA_D,#1
            LJMP  CHK_PRESS_D

```

```

CHK_2:      CJNE R0,#12H,CHK_3
            MOV   PTR_ASC_D,#2'
            MOV   PTR_DATA_D,#2
            LJMP  CHK_PRESS_D

```

```

CHK_3:      CJNE R0,#13H,CHK_4
            MOV   PTR_ASC_D,#3'
            MOV   PTR_DATA_D,#3
            LJMP  CHK_PRESS_D

```

```

CHK_4:      CJNE  R0,#14H,CHK_5
             MOV   PTR_ASC_D,#4'
             MOV   PTR_DATA_D,#4
             LJMP  CHK_PRESS_D

```

```

CHK_5:      CJNE  R0,#15H,CHK_6
             MOV   PTR_ASC_D,#5'
             MOV   PTR_DATA_D,#5
             LJMP  CHK_PRESS_D

```

```

CHK_6:      CJNE  R0,#16H,CHK_7D
             MOV   PTR_ASC_D,#6'
             MOV   PTR_DATA_D,#6
             LJMP  CHK_PRESS_D

```

```

CHK_7D:     CJNE  R0,#17H,CHK_8D
             MOV   PTR_ASC_D,#7'
             MOV   PTR_DATA_D,#7
             LJMP  CHK_PRESS_D

```

```

CHK_8D:     CJNE  R0,#18H,CHK_9D
             MOV   PTR_ASC_D,#8'
             MOV   PTR_DATA_D,#8
             LJMP  CHK_PRESS_D

```

```

CHK_9D:     CJNE  R0,#19H,CHK_0D
             MOV   PTR_ASC_D,#9'
             MOV   PTR_DATA_D,#9
             LJMP  CHK_PRESS_D

```

เอกสารนี้ D: สารที่สงวนไว้สำหรับบริการเชิงในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV PTR_ASC_D,#0'
MOV PTR_DATA_D,#0
LJMP CHK_PRESS_D

CHK_SH:    CJNE R0,#1CH,CHK_NO_PRESS_D
           MOV  PTR_DATA_D,#1CH

```

```

CHK_PRESS_D:  MOV  DPTR,#PORT_B
              MOVX A,@DPTR
              JNB  ACC.4,PRESS_D
              LJMP AGAIN_D

```

```

PRESS_D:     INC  KEY

```

```

CHK_NO_PRESS_D: POP  00H

```

```

RET

```

```

;*****
;
;*      Scankeypad      *
;*      i/p acc o/p ptr_asc & ptr_data *
;*****
;

```

```

SCANKEY:    PUSH  00H

```

```

            PUSH  02H

```

```

AGAIN_K:    ANL  A,#1FH

```

```

            MOV  R0,A

```

```

            CJNE R0,#10H,CHK_2_B

```

```

            MOV  PTR_ASC_K,#'1'

```

```

            MOV  R2,PTR_DATA_K

```

```

            CJNE R2,#1BH,SC1

```

```

            MOV  PTR_2ND,#'A'

```

```

SC1:        MOV  PTR_DATA_K,#1

```

```

            LJMP  CHK_PRESS_K

```

```

CHK_2_B:    CJNE R0,#11H,CHK_3_C

```

```

MOV PTR_ASC_K,#2'
MOV R2,PTR_DATA_K
CJNE R2,#1BH,SC2
MOV PTR_2ND,#'B'
SC2: MOV PTR_DATA_K,#2
LJMP CHK_PRESS_K

```

```

CHK_3_C: CJNE R0,#12H,CHK_U_L
MOV PTR_ASC_K,#3'
MOV R2,PTR_DATA_K
CJNE R2,#1BH,SC3
MOV PTR_2ND,#'C'
SC3: MOV PTR_DATA_K,#3
LJMP CHK_PRESS_K

```

```

CHK_U_L: CJNE R0,#13H,CHK_4_D
MOV R2,PTR_DATA_K
CJNE R2,#1BH,SC4
MOV PTR_2ND,#23H
SC4: MOV PTR_DATA_K,#13H
LJMP CHK_PRESS_K

```

```

CHK_4_D: CJNE R0,#14H,CHK_5_E
MOV PTR_ASC_K,#4'
MOV R2,PTR_DATA_K
CJNE R2,#1BH,SC5
MOV PTR_2ND,#'D'
SC5: MOV PTR_DATA_K,#4
LJMP CHK_PRESS_K

```

```

CHK_5_E: CJNE R0,#15H,CHK_6_F

```

```

CHK_9K:      CJNE  R0,#1AH,CHK_2ND
              MOV   PTR_ASC_K,#9'
              MOV   PTR_DATA_K,#9
              LJMP  CHK_PRESS_K

CHK_2ND:     CJNE  R0,#1BH,CHK_CLR
              MOV   PTR_DATA_K,#1BH
              LJMP  CHK_PRESS_K

CHK_CLR:     CJNE  R0,#1CH,CHK_OK
              MOV   PTR_DATA_K,#1CH
              LJMP  CHK_PRESS_K

CHK_OK:      CJNE  R0,#1DH,CHK_H
              MOV   PTR_ASC_K,#0'
              MOV   PTR_DATA_K,#0
              LJMP  CHK_PRESS_K

CHK_H:       CJNE  R0,#1EH,CHK_E
              MOV   PTR_DATA_K,#1EH
              LJMP  CHK_PRESS_K

CHK_E:       CJNE  R0,#1FH,CHK_NO_PRESS_K
              MOV   PTR_DATA_K,#1FH

CHK_PRESS_K: MOV   DPTR,#PORT_C
              MOVX  A,@DPTR
              JNB  ACC.4,PRESS_K
              LJMP  AGAIN_K

PRESS_K:     INC   KEY

```

```

CHK_NO_PRESS_K: POP 02H
                 POP 00H
                 RET

```

```

;*****
;

```

```

;*          Check Press Color Key          *
;

```

```

;*****
;

```

```

CHK_COLOR:      CJNE A,#13H,ER66

```

```

                 DEC KEY

```

```

                 LJMP STOP

```

```

ER66:           CJNE A,#17H,ER67

```

```

                 DEC KEY

```

```

                 LJMP STOP

```

```

ER67:           CJNE A,#1BH,ER68

```

```

                 DEC KEY

```

```

                 LJMP STOP

```

```

ER68:           CJNE A,#1CH,ER69

```

```

                 DEC KEY

```

```

                 LJMP STOP

```

```

ER69:           CJNE A,#1EH,ER70

```

```

                 DEC KEY

```

```

                 LJMP STOP

```

```

ER70:           CJNE A,#1FH,STOP

```

```

                 DEC KEY

```

```

STOP:           RET

```

```

;*****
;

```

```

;*          Show Sum on lcd          *
;

```

```

;*****
;

```

```

SHOW_SUM:      PUSH DPH

```

```

                 PUSH DPL

```

```

                 MOV DPTR,#SUM_100

```

```

                 MOVX A,@DPTR

```

```

ANL  A,#0FH
LCALL HEXASC
LCALL WR_LCD
MOV  DPTR,#SUM_10
MOVX A,@DPTR
ANL  A,#0FH
LCALL HEXASC
LCALL WR_LCD
MOV  DPTR,#SUM_1
MOVX A,@DPTR
ANL  A,#0FH
LCALL HEXASC
LCALL WR_LCD
POP  DPL
POP  DPH
RET
;*****
;*      Increment Sum      *
;*****
INC_SUM:  PUSH  DPH
          PUSH  DPL
          MOV   DPTR,#SUM_1
          MOVX  A,@DPTR
          ANL  A,#0FH
          MOV  SUM_R_1,A
          MOV  DPTR,#SUM_10
          MOVX A,@DPTR
          ANL  A,#0FH
          MOV  SUM_R_10,A
          MOV  DPTR,#SUM_100
          MOVX A,@DPTR

```

```

ANL  A,#0FH
MOV  SUM_R_100,A
INC  SUM_R_1
MOV  A,SUM_R_1
CJNE A,#10,SU
MOV  SUM_R_1,#0
INC  SUM_R_10
MOV  A,SUM_R_10
CJNE A,#10,SU
MOV  SUM_R_10,#0
INC  SUM_R_100
MOV  A,SUM_R_100
CJNE A,#10,SU
MOV  SUM_R_100,#0
MOV  A,SUM_R_1
MOV  DPTR,#SUM_1
MOVX @DPTR,A
MOV  A,SUM_R_10
MOV  DPTR,#SUM_10
MOVX @DPTR,A
MOV  A,SUM_R_100
MOV  DPTR,#SUM_100
MOVX @DPTR,A
POP  DPL
POP  DPH
RET
    
```

SU:

```

;*****
;*          Change Data          *
;*          Bcd to Ascii & Show to LCD      *
;*****
    
```

SHOWASC: PUSH ACC

```

SWAP A
ANL A,#0FH
LCALL HEXASC
LCALL WR_LCD
POP ACC
ANL A,#0FH
LCALL HEXASC
LCALL WR_LCD
RET

```

CHANGE:

```

PUSH ACC
SWAP A
ANL A,#0FH
LCALL HEXASC
MOV HEX1,A
POP ACC
ANL A,#0FH
LCALL HEXASC
MOV HEX2,A
RET

```

HEXASC:

```

PUSH DPH
PUSH DPL
MOV DPTR,#HEXASC_TAB
MOVC A,@A+DPTR
POP DPL
POP DPH
RET

```

```

;*****
;

```

```

;* Write Month to LCD *
```

```

;*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WRITE_MON:    PUSH  00H
              MOV   A,MONTH
              CJNE  A,#10H,CHK_MONTH_1
              MOV   A,#0AH
              LJMP  WRITE_MONTH

CHK_MONTH_1:  CJNE  A,#11H,CHK_MONTH_2
              MOV   A,#0BH
              LJMP  WRITE_MONTH

CHK_MONTH_2:  CJNE  A,#12H,WRITE_MONTH
              MOV   A,#0CH

WRITE_MONTH:  MOV   PTR_MONTH,A
              MOV   DPTR,#MONTH_JAN
              MOV   R0,#0
              MOV   A,PTR_MONTH
              DEC   A
              MOV   B,#3
              MUL   AB
WRITE_CHAR1:  MOVC  A,@A+DPTR
              LCALL WR_LCD
              INC   DPTR
              INC   R0
              MOV   A,PTR_MONTH
              DEC   A
              MOV   B,#3
              MUL   AB
              CJNE R0,#3,WRITE_CHAR1
              POP   00H
              RET

```

```

;*****
;

```

```

;*          Write Day to LCD          *
;

```

```

;*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WRITE_DAY:    PUSH  00H
              MOV   A,DAY
              MOV   PTR_DAY,A
              MOV   DPTR,#SUNDAY
              MOV   R0,#0
              MOV   A,PTR_DAY
              DEC   A
              MOV   B,#3
              MUL   AB
WRITE_CHAR2:  MOVC  A,@A+DPTR
              LCALL WR_LCD
              INC   DPTR
              INC   R0
              MOV   A,PTR_DAY
              DEC   A
              MOV   B,#3
              MUL   AB
              CJNE  R0,#3,WRITE_CHAR2
              POP   00H
              RET

```

```

;*****
;

```

```

;*      TX serial Text from ram Pointer      *

```

```

;*****
;

```

```

TX_TEXT:     PUSH  01H
              MOV   R1,#28
              CLR   TI
TX_TEXT_LOOP: MOVX  A,@DPTR
              ;CJNE A,#0FFH,TX_TEXT_CHAR
              DJNZ  R1,TX_TEXT_CHAR
              POP   01H

```

```

                                RET
TX_TEXT_CHAR:  MOV  TX_DATA,A
                                SETB  TI
                                INC   DPTR
                                LJMP  TX_TEXT_LOOP

;*****
;*      Serial interrupt subroutine      *
;*****

SER_INT:      PUSH  ACC
              JBC  RI,SER_RX

SER_TX:      CLR  TI
              MOV  SBUF,TX_DATA

SER_TX_WAIT: JBC  TI,SER_EXIT
              LJMP SER_TX_WAIT

SER_RX:      CLR  RX_OK
              MOV  A,SBUF
              CJNE A,#0DH,SER_EXIT
              SETB RX_OK

SER_EXIT:    POP  ACC
              RETI

;*****
;*      Initial LCD                      *
;*      4-Bit Interface for Monitor      *
;*****

INIT_LCD_MO: LCALL DELAY_INIT
              CLR  RS_LCD
              MOV  A,#33H           ; Set DL = 1 3-time
              LCALL WR_INS
              MOV  A,#32H           ; Clear DL = 0 1-time
              LCALL WR_INS
              MOV  A,#28H           ; Function set

```

```

LCALL WR_INS           ; DL=0 4Bit,N=1 2Line,F=0 5X7
LCALL ON_LCD          ; Entry display,cursor off,cursor not blink
MOV  A,#05H
LCALL WR_INS
LCALL CLR_LCD
RET

```

```

;*****
;

```

```

;*      Initial LCD      *
;

```

```

;*      4-Bit Interface  *
;

```

```

;*****
;

```

```

INIT_LCD:      LCALL  DELAY_INIT
               CLR   RS_LCD
               MOV   A,#33H           ; Set DL = 1 3-time
               LCALL WR_INS
               MOV   A,#32H           ; Clear DL = 0 1-time
               LCALL WR_INS
               MOV   A,#28H           ; Function set
               LCALL WR_INS           ; DL=0 4Bit,N=1 2Line,F=0 5X7
               LCALL ON_LCD          ; Entry display,cursor off,cursor not blink
               MOV   A,#06H
               LCALL WR_INS
               LCALL CLR_LCD
               RET

```

```

;*****
;

```

```

;*      Input : ACC (ASCII)      *
;

```

```

;*      Output : Data bus LCD    *
;

```

```

;*****
;

```

```

WR_LCD:      PUSH  02H
               MOV   B,A
               ANL   A,#0F0H
               ORL   A,#0001100B     ;RS/E HI

```

```

MOV R2,A
MOV A,P1
ANL A,#00000011B
ORL A,R2
MOV P1,A
LCALL EN_LCD
MOV A,B ;Low byte
SWAP A
ANL A,#0F0H
ORL A,#00001100B
MOV R2,A
MOV A,P1
ANL A,#00000011B
ORL A,R2
MOV P1,A
LCALL EN_LCD
POP 02H
RET
;*****
;* Write Instruction LCD *
;* Input : ACC (Command) *
;* Output : Data bus LCD *
;*****
WR_INS: PUSH 02H
MOV B,A
ANL A,#0F0H
SETB ACC.3 ;E HI
MOV R2,A
MOV A,P1
ANL A,#00000011B ;else bit
ORL A,R2

```

```

MOV P1,A ; High byte
LCALL EN_LCD
MOV A,B ; Low byte
SWAP A
ANL A,#0F0H
SETB ACC.3
MOV R2,A
MOV A,P1
ANL A,#00000011B
ORL A,R2
MOV P1,A
LCALL EN_LCD
POP 02H
RET
;*****
;* Set Address position of LCD *
;* Input : ACC (addr.) *
;*****
ADDR_LCD: SETB ACC.7
LCALL WR_INS
RET
;*****
;* Enable Pin E LCD *
;* Active Chip select *
;*****
EN_LCD: SETB E_LCD ; Disable LCD
LCALL DELAY_EN ; Busy delay time
CLR E_LCD ; Enable LCD
LCALL DELAY_EN ; Busy delay time
RET
;*****

```

```

;*          CLEAR LCD DISPLAY          *
;
;*          Set DD RAM address=0      *
;
;*****
CLR_LCD:    MOV    A,#01H
            LCALL  WR_INS
            RET

;*****
;*          RETURN HOME LCD DISPLAY   *
;
;*****
HOME_LCD:   MOV    A,#02H
            LCALL  WR_INS
            RET

;*****
;*          LCD DISPLAY OFF           *
;
;*****
OFF_LCD:    MOV    A,#08H
            LCALL  WR_INS
            RET

;*****
;*          LCD DISPLAY ON            *
;
;*****
ON_LCD:     MOV    A,#0CH
            LCALL  WR_INS
            RET

;*****
;*          LCD CURSOR ON & BLINK     *
;
;*****
BLINK_LCD:  MOV    A,#0FH
            LCALL  WR_INS
            RET

```

```

;*          LCD CURSOR SHIFT LEFT          *
;
;*****
LEFT_CUR:      MOV  A,#10H
               LCALL WR_INS
               RET

;*****

;*          LCD CURSOR SHIFT RIGHT         *
;
;*****
RIGHT_CUR:     MOV  A,#14H
               LCALL WR_INS
               RET

;*****

;*          LCD CHARECTER SHIFT LEFT      *
;
;*****
LEFT_CHAR:     MOV  A,#18H
               LCALL WR_INS
               RET

;*****

;*          LCD CHARECTER SHIFT RIGHT     *
;
;*****
RIGHT_CHAR:    MOV  A,#1CH
               LCALL WR_INS
               RET

;*****1307*****

;*          WRITE DATA TO EEPROM 1 BYTE   *
;
;*INPUT : ADD_HIGH          *
;* : ADD_LOW                *
;* : DATA                  *
;
;*****
WRITE_BYTE:    CLR  SDA          ;start bit
               CLR  SCL

```

```

MOV  A,#CONT_BYTE_W           ;send control byte
LCALL LOOP_SEND
SETB  SDA
SETB  SCL
JB   SDA,WRITE_BYTE           ;loop until busy
CLR   SCL
MOV  A,ADD_LOW                ;send address low
LCALL LOOP_SEND
SETB  SDA
SETB  SCL
JB   SDA,WRITE_BYTE           ;loop until busy
CLR   SCL
MOV  A,DATA                   ;send data
LCALL LOOP_SEND
SETB  SDA
SETB  SCL
JB   SDA,WRITE_BYTE           ;loop until busy
CLR   SDA
CLR   SCL
SETB  SCL                      ;stop bit
SETB  SDA
RET

```

```

,*****
,

```

```

;*      READ DATA FROM EEPROM 1 BYTE      *

```

```

;* INPUT : ADD_HIGH      *

```

```

;*      : ADD_LOW      *

```

```

;* OUTPUT : DATA      *

```

```

,*****
,

```

```

READ_BYTE:      CLR  SDA           ;start bit

```

```

CLR  SCL

```

```

MOV  A,#CONT_BYTE_W           ;send control byte

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL LOOP_SEND
SETB SDA
SETB SCL
JB SDA,READ_BYTE ;loop until busy
CLR SCL
MOV A,ADD_LOW ;send address low
LCALL LOOP_SEND
SETB SDA
SETB SCL
JB SDA,READ_BYTE ;loop until busy
CLR SCL
SETB SCL
SETB SDA
CLR SDA ;start bit
CLR SCL
MOV A,#CONT_BYTE_R ;send control byte
LCALL LOOP_SEND
SETB SDA
SETB SCL
JB SDA,READ_BYTE ;loop until busy
CLR SCL
LCALL LOOP_READ
SETB SDA
SETB SCL
CLR SCL
SETB SCL ;stop bit
SETB SDA
RET

```

```

;*****
;

```

```

;* WRITE *

```

```

;* INPUT: ACC *

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****
```

```
LOOP_SEND:    PUSH  02H
              MOV   R2,#08H
LOOP_SEND1:   RLC   A
              MOV   SDA,C
              SETB  SCL
              CLR   SCL
              DJNZ  R2,LOOP_SEND1
              POP   02H
              RET
```

```
*****
```

```
;*          READ          *
;* OUTPUT: ACC          *
*****
LOOP_READ:    PUSH  02H
              MOV   R2,#08H
LOOP_READ1:   SETB  SCL
              MOV   C,SDA
              CLR   SCL
              RLC   A
              DJNZ  R2,LOOP_READ1
              MOV   DATA,A
              POP   02H
              RET
```

```
*****
```

```
;*          Delay Time    *
*****
```

```
DELAY_1S:    MOV   R5,#100
DEL1:        MOV   R6,#010
DEL2:        MOV   R7,#0FFH
DEL3:        NOP
```

```

NOP
NOP
NOP
DJNZ R7,DEL3
DJNZ R6,DEL2
DJNZ R5,DEL1
RET

```

```

DELAY_500mS: MOV R5,#50
DEL4:        MOV R6,#010
DEL5:        MOV R7,#0FFH
DEL6:

```

```

NOP
NOP
NOP
DJNZ R7,DEL6
DJNZ R6,DEL5
DJNZ R5,DEL4
RET

```

```

DELAY_200mS: MOV R5,#20
DEL7:        MOV R6,#010
DEL8:        MOV R7,#0FFH
DEL9:

```

```

NOP
NOP
NOP
NOP
DJNZ R7,DEL9
DJNZ R6,DEL8
DJNZ R5,DEL7

```

```
RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DELAY_100mS:   MOV    R6,#100
DEL10:         MOV    R7,#0FFH
DEL11:         NOP
               NOP
               NOP
               NOP
               DJNZ   R7,DEL11
               DJNZ   R6,DEL10
               RET
DELAY_50mS:    MOV    R6,#050
DEL12:         MOV    R7,#0FFH
DEL13:         NOP
               NOP
               NOP
               NOP
               DJNZ   R7,DEL13
               DJNZ   R6,DEL12
               RET
DELAY_10mS:    MOV    R6,#010
DEL14:         MOV    R7,#0FFH
DEL15:         NOP
               NOP
               NOP
               NOP
               DJNZ   R7,DEL15
               DJNZ   R6,DEL14
               RET

```

```

DELAY_2mS:      MOV    R6,#002
DEL16:          MOV    R7,#0FFH
DEL17:          NOP
                NOP
                NOP
                NOP
                DJNZ   R7,DEL17
                DJNZ   R6,DEL16
                RET

PW_DELAY:       PUSH   DPH
                PUSH   DPL
                PUSH   ACC
                MOV    DPTR,#8000H
PW_DEL:         INC    DPTR
                MOV    A,DPL
                ORL    A,DPH
                JNZ   PW_DEL
                POP    ACC
                POP    DPL
                POP    DPH
                RET

DELAY_INIT:     PUSH   07H
                PUSH   06H
                MOV    R6,#0FFH
DEL_INIT1:     MOV    R7,#0FFH
                DJNZ   R7,$
                DJNZ   R6,DEL_INIT1
                POP    06H
                POP    07H

```

```

RET

DELAY_EN:    PUSH  07H
             PUSH  06H
             MOV   R6,#05H
EN1:         MOV   R7,#0FFH
             DJNZ  R7,$
             DJNZ  R6,EN1
             POP   06H
             POP   07H
             RET

```

```

TITLE:       DB   'Telephone Counter '
SCR_SELECT:  DB   'Select Mode '
SCR_SELECT1: DB   '1=Set Time 3=TX '
SCR_SELECT2: DB   '2=Clr Mem '
SCR_SELECT3: DB   '1=Set Time&Date '
SCR_SELECT4: DB   '2=Clear Memory '
SCR_SELECT5: DB   '3=Tranfer Data '
SCR_SET:     DB   '#Set '
SCR_TEL_IN:  DB   ' Tel.In '
SCR_TEL_OUT: DB   ' Tel. Out '
SCR_NO_DIAL: DB   ' No. Dial '
SCR_BUSY:    DB   ' Line Busy '
SCR_SET_DAY: DB   'Day = dd '
SCR_SET_DATE: DB   'Date = dd/mm/yy '
SCR_SET_TIME: DB   'Time = hh/mm/ss '
SCR_SET_MODE: DB   'Set Date & Time '
SCR_MEM_MODE: DB   'Clear Memory..'
SCR_TX_MODE: DB   'Send Data..'
SCR_COMPLETE: DB   ' Complete '

```

SCR_CONFIRM1: DB 'Enter = Yes '
 SCR_CONFIRM2: DB 'Clear = No '
 MONTH_JAN: DB 'Jan'
 MONTH_FEB: DB 'Feb'
 MONTH_MAR: DB 'Mar'
 MONTH_APR: DB 'Apr'
 MONTH_MAY: DB 'May'
 MONTH_JUN: DB 'Jun'
 MONTH_JUL: DB 'Jul'
 MONTH_AUG: DB 'Aug'
 MONTH_SEP: DB 'Sep'
 MONTH_OCT: DB 'Oct'
 MONTH_NOV: DB 'Nov'
 MONTH_DEC: DB 'Dec'
 SUNDAY: DB 'Sun'
 MONDAY: DB 'Mon'
 TUESDAY: DB 'Tue'
 WEDNESDAY: DB 'Wed'
 THURSDAY: DB 'Thu'
 FRIDAY: DB 'Fri'
 SATURDAY: DB 'Sat'
 HEXASC_TAB: DB 30H,31H,32H,33H,34H,35H,36H,37H,38H,39H
 DB 41H,42H,43H,44H,45H,46H

END