ปริญญานิพนธ์

เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ

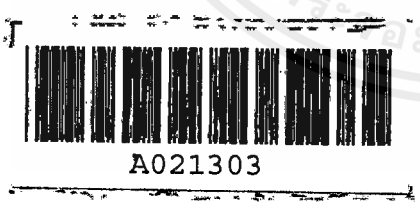
AUTOMATIC TELEPHONE SERVICE CHARGE CALCULATOR

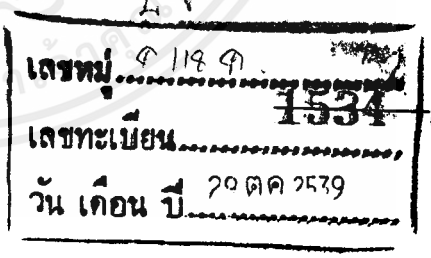
นายคเชนท์ ประสาน

นายชูชาติ เหลืองอร่ามวิฑูร

นายธเนศ ตั้งจิตเจริญเลิศ

นายนฤพนธ์ น้อยภู่





ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ

AUTOMATIC TELEPHONE SERVICE

CHARGE CALCULATOR

ชื่อนักศึกษา

1. นายคเชนทร์	ประสาน	**รหัสประจำตัว**	37031202
2. นายชูชาติ	เหลืองอร่ามวิฑูร	**รหัสประจำตัว**	37037203
3. นายธเนศ	ตั้งจิตเจริญเลิศ	**รหัสประจำตัว**	37031205
4. นายนฤพนธ์	น้อยภู่	**รหัสประจำตัว**	37031208

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** วิศวกรรมโทรคมนาคม

อาจารย์ผู้ควบคุมปริญญานิพนธ์

1. อาจารย์พีระวุฒิ สุวรรณจันทร์
2. อาจารย์กิติพงศ์ มะโน
3. อาจารย์วรวิทย์ สมหา

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. อาจารย์กิติพงศ์	มะโน	
2. อาจารย์วิสุทธิ์	อธิพรธรรม	
3. อาจารย์โกศล	ตราชู	

วัน/เดือน/ปี ที่สอบ 7 พ.ค. 2539 **เวลา** 17.00 น. ถึงเวลา 18.00 น.

สถานที่สอบ ห้อง ค.301 คณะครุศาสตร์อุตสาหกรรม

ภาควิชารับรองแล้ว

ลงชื่อ.....

(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 21 เดือน พ.ค. พ.ศ. 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ

AUTOMATIC TELEPHONE SERVICE CHARGE CALCULATOR

ผู้จัดทำ

1. นายคเชนท์ ประสาน
2. นายชูชาติ เหลืองอร่ามวิฑูร
3. นายธเนศ ตั้งจิตเจริญเลิศ
4. นายนฤพนธ์ น้อยภู่

อาจารย์ที่ปรึกษา

ลงนาม.....

(อาจารย์พีระวุฒิ สุวรรณจันทร์)

ลงนาม.....

(อาจารย์กิติพงศ์ มะโน)

ลงนาม.....

(อาจารย์วรวิทย์ สมหา)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม.....

(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ

AUTOMATIC TELEPHONE SERVICE CHARGE CALCULATOR

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของระบบเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ
2. เพื่อวิเคราะห์ ออกแบบวงจรเครื่องคำนวณค่าบริการการใช้โทรศัพท์
3. เพื่อสร้างวงจรต่างๆ ประกอบเป็นเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ
4. เพื่อใช้แสดงผลการทำงานของเครื่องคำนวณค่าบริการการใช้โทรศัพท์โดยผ่าน LCD และเครื่องพิมพ์
5. เพื่อนำเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติไปติดตั้งใช้งานตามสถานที่ต่างๆได้

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเข้าใจหลักการทำงานของระบบโทรศัพท์
2. สามารถเข้าใจการทำงานของเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ
3. ออกแบบวงจรการทำงานของเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ
4. สามารถเข้าใจการทำงานของไมโครคอนโทรลเลอร์
5. สามารถเขียนโปรแกรมควบคุมการทำงานของครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ
6. สามารถสร้างเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ
7. สามารถนำเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติไปติดตั้งตามสถานที่ ที่ให้บริการโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ

นายคเชนท์ ประสาน
นายชูชาติ เหลืองอร่ามวิฑูร
นายธเนศ ตั้งจิตเจริญเลิศ
นายนฤพนธ์ น้อยภู่

อาจารย์ที่ปรึกษา
อาจารย์พีระวุฒิ สุวรรณจันทร์
อาจารย์กิติพงค์ มะโน
อาจารย์วรวิทย์ สมหา
ปีการศึกษา 2538

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เสนอเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ สามารถทำหน้าที่เป็นเครื่องจับเวลาสำหรับคิดค่าบริการโทรศัพท์ทางไกล โดยจะทำการคิดค่าโทรศัพท์เมื่อปลายสายรับโทรศัพท์ และจะคิดค่าบริการในการใช้โทรศัพท์ในอัตราที่สอดคล้องกับค่าบริการที่องค์โทรศัพท์แห่งประเทศไทยเป็นผู้คิดจากผู้เช่า การแสดงผลจะปรากฏทางจอ LCD และสามารถพิมพ์ทางเครื่องพิมพ์ เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัตินี้ใช้ไมโครโปรเซสเซอร์ 8 บิท ในตระกูล MCS-51 เบอร์ 8032 เป็นตัวควบคุมการทำงานของระบบ ดังนั้นจึงมีความคล่องตัวและสะดวกในการใช้งาน และสามารถพัฒนาเพื่อเพิ่มประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC TELEPHONE SERVICE CHARGE CALCULATOR

MR.KHACHEN PRASAN
MR.CHUCHAT LEANGARAMWITOON
MR.THANET TUNGJITJAROENLERT
MR.NARUEPON NOYPU

ADVISORS

MR.PEERAWUT SUWANCHAN
MR.KITIPHONG MANO
MR.WORAWIT SOMHA

1995

ABSTRACT

This thesis present counling the automatic telephone service charge calculator.This machainne is use to count the time for calculating the long distant telephone service charge. It start calculting when the receiver answers the call . The service will be charged in corresponde with the rate charged from receiver by aurithory telephone Thailand .The figures of the charge is show on the LCD screen , and printed in the slip by dot matrix printer The automatic telephone service charge calculater is controlled by microprocessor 8 bit model MCS 51 NO.8032 in order to provice facility and convenience when using the machine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สามารถสำเร็จลุล่วงได้ด้วยดีนั้น กลุ่มผู้จัดทำขอขอบคุณอาจารย์ที่ภาควิชา ทุกท่านที่ได้ให้คำแนะนำและให้คำปรึกษาเป็นอย่างดี โดยอาจารย์โกศล ตราชู และอาจารย์พีระวุฒิ สุวรรณจันทร์ ที่กรุณาอนุเคราะห์วัสดุ อุปกรณ์เครื่องมือเครื่องใช้ และสถานที่ในการทำโครงงาน และเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือทางด้านคำปรึกษาและเครื่องมืออุปกรณ์เป็นอย่างดี และที่สำคัญที่สุดคือ บิดา มารดา ที่ให้การสนับสนุนด้านการศึกษาตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1บทนำ	3
2.2 หน้าที่หลักของโทรศัพท์	3
2.3 สัญญาณต่างๆ ในระบบชุมสายโทรศัพท์	5
2.3.1 สัญญาณสมาชิก	5
2.4 ระบบการกำหนดหมายเลขโทรศัพท์	7
2.5 ลักษณะของสัญญาณโทรศัพท์	7
2.5.1 เมื่อเป็นผู้โทรศัพท์เรียก	8
2.5.2 เมื่อเป็นผู้รับ	10
2.6 ภาคถอดรหัส	11
2.7 ภาคตรวจสอบสัญญาณ	12
2.8 ANT-32 บอร์ดไมโครคอนโทรลเลอร์	19
2.8.1 คุณสมบัติ ANT-32	19
2.8.2 OPTIONAL MICROPROCESSOR SUPERVISORY CIRCUIT	21
2.8.3 Power Fail Detector (จัมพ์เปอร์)	22
2.8.4 CMOS RAM Write Protection (จัมพ์เปอร์ BACK UP)	22
2.9 Watchdog Timer (จัมพ์เปอร์ WD)	23
2.10 OPTIONAL REAL TIME CLOCK & CALANDER	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
2.11 แนวทางในการพัฒนาโปรแกรม	26
บทที่ 3 การออกแบบการสร้างและการทำงาน	27
3.1 บทนำ	27
3.2 ขั้นตอนการตรวจสอบการใช้โทรศัพท์	27
3.3.ผังการทำงานของเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ	28
3.4 การและออกแบบวงจร	28
3.4.1 ส่วนของวงจรตรวจสอบ	29
3.4.2 วงจรตรวจสอบการยกหูโทรศัพท์	29
3.4.3 วงจรตรวจสัญญาณกริ่งเรียก	30
3.4.4 วงจรตรวจแยกสัญญาณโทน	31
3.4.5 วงจรถอดรหัสความถี่โทรศัพท์ชนิดกดปุ่ม (DTMF)	32
3.4.6 วงจรแหล่งจ่ายไฟ(POWER SUPPLY)	32
3.5 ขั้นตอนการออกแบบวงจรกรองความถี่สัญญาณชนิดแถบความถี่ผ่าน	34
3.5.1 การทำงานของวงจร	35
3.6 โปรแกรมควบคุมการทำงานของระบบ	36
บทที่ 4 การทดลองและผลการทดลอง	38
4.1 บทนำ	38
4.2 ผลการทดลอง	38
4.2.1 วงจรตรวจสอบการยกหูโทรศัพท์	38
4.2.2 วงจรตรวจสอบสัญญาณกริ่งเรียก	40
4.2.3 วงจรตรวจแยกสัญญาณโทรศัพท์	41
วิธีการทดลองการใช้เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ	41
บทที่ 5 บทสรุปปัญหา แนวทางแก้ไขและพัฒนา	45
5.1 บทสรุป	45
5.2 แนวความคิดและข้อเสนอแนะในการพัฒนาโครงงาน	45
ภาคผนวก ก โปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์	47
ภาคผนวก ข รายการอุปกรณ์เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
ภาคผนวก ค รูปต้นแบบของเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ	77
ภาคผนวก ง รายการข้อมูลและคุณสมบัติของอุปกรณ์	92
บรรณานุกรม	99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปภาพ	หน้า
รูปที่ 2.1 สัญญาณให้หมุน	5
รูปที่ 2.2 สัญญาณไม่ว่าง	6
รูปที่ 2.3 สัญญาณเรียกกลับ	6
รูปที่ 2.4 สัญญาณกริ่งเรียก	7
รูปที่ 2.5 ลักษณะสัญญาณโทรศัพท์	8
รูปที่ 2.6 การเข้ารหัส	10
รูปที่ 2.7 รายละเอียดขา MT8870	14
รูปที่ 2.8 โครงสร้างภายในของ MT8870	14
รูปที่ 2.9 แผนภูมิเวลาของ MT8870	15
รูปที่ 2.10 ขาของไอซีเบอร์ LM567	16
รูปที่ 2.11 ผังการทำงานของไอซี LM567	17
รูปที่ 2.12 ตัวอย่างการต่อใช้งานไอซี LM567	18
รูปที่ 2.13 จัมพ์เปอร์รีเซท	21
รูปที่ 2.14 จัมพ์เปอร์ของวงจรตรวจจับแรงดันไฟตก	22
รูปที่ 2.15 จัมพ์เปอร์ของการสำรองข้อมูล	23
รูปที่ 2.16 จัมพ์เปอร์ Watchdog timer	24
รูปที่ 3.1 ผังการทำงานของเครื่องคำนวรค่าบริการการใช้โทรศัพท์	28
รูปที่ 3.2 วงจรการตรวจสอบการยกหูโทรศัพท์	29
รูปที่ 3.3 วงจรตรวจสัญญาณกริ่งเรียก	30
รูปที่ 3.4 วงจรตรวจแยกสัญญาณโทน	31
รูปที่ 3.5 วงจรจ่ายไฟ 5 V ชนิดมีแบตเตอรี่สำรอง	34
รูปที่ 3.6 แบนด์พาสฟิลเตอร์แบบเปลี่ยนแปลงสถานะโดยมีการป้อนกลับแบบลบ	36
รูปที่ 4.1 ตำแหน่งของจุดวัดสัญญาณโทรศัพท์	38
รูปที่ 4.2 ผังการทำงานวงจรการตรวจสอบการยกหู	39
รูปที่ 4.3 สัญญาณเอาท์พุตวงจรตรวจสอบการยกหูโทรศัพท์	39
รูปที่ 4.4 การวัดสัญญาณเอาท์พุตตามจุดต่างๆ	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ **หน้า**

รูปที่ 4.5 สัญญาณเอาท์พุตวงจรตรวจสอบสัญญาณกริ่งโทรศัพท์ 40

รูปที่ 4.6 ผังการทำงานของวงจรแยกสัญญาณโทรศัพท์ 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การมอดูเลตของสัญญาณโทรศัพท์	4
ตารางที่ 2.2 ความถี่และการจัดปุ่ม	8
ตารางที่ 2.3 การถอดรหัสความถี่	11
ตารางที่ 2.4 การเลือกค่า Reset Pulse Width และ Watching Timeout	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โทรศัพท์ได้ถูกนำมาใช้งานในประเทศไทยครั้งแรกเมื่อปี พ.ศ. 2424 โดยระบบของโทรศัพท์ที่นำมาใช้เป็นแบบระบบแม็กนีโต (Magneto) ต่อมาได้มีการพัฒนาระบบขึ้นเรื่อยๆ จนถึงปัจจุบัน

ปัจจุบันโทรศัพท์เป็นสิ่งที่จำเป็นอย่างยิ่งที่ใช้ในการสื่อสารที่มีความจำเป็นในระบบธุระกิจในยุคของการการสื่อสารเสรี ด้วยเหตุผลที่ว่าระบบโทรศัพท์เป็นเครือข่ายสื่อสารที่กว้างขวางและควบคุมพื้นที่ได้กว้างขวาง ซึ่งสามารถอำนวยความสะดวกและรวดเร็วเสมือนผู้พูดอยู่ตรงหน้ากัน ช่วยในการประหยัดเวลาในการสื่อสาร และลดปัญหาของการเดินทาง ส่งผลให้สามารถประหยัดทรัพยากรอื่นๆ

การที่โทรศัพท์มีข้อดีอยู่มากมาย ทำให้ความต้องการใช้โทรศัพท์มีปริมาณมากขึ้นตามลำดับแต่ในขณะที่จำนวนโทรศัพท์มีจำนวนจำกัด ซึ่งปัจจัยดังกล่าวเป็นผลกระทบต่อระบบเศรษฐกิจของประเทศในขณะนี้ ส่งผลให้การประกอบธุรกิจการค้าและการลงทุนหยุดชงักลง ส่งผลให้ธุรกิจไม่สามารถขยายกิจการให้เติบโตขึ้นได้

ข้อจำกัดของการมีโทรศัพท์จึงมีการให้บริการการใช้โทรศัพท์เพื่อใช้สื่อสารในปัจจุบันมีจำนวนมาการคิดค่าบริการในแต่ละที่ก็มีค่าบริการต่างกันทำให้เสียเวลาในการคำนวณหาค่าการใช้บริการ ในบางครั้งก็เกิดการผิดพลาด ซึ่งจะส่งผลกระทบต่อผู้บริโภค ดังนั้นการพัฒนาเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัตินี้ขึ้น เพื่อสามารถนำไปใช้คิดค่าบริการโทรศัพท์ตามศูนย์บริการด้านโทรศัพท์ต่างๆ และเกิดความถูกต้องมากยิ่งขึ้น เพราะมีการแสดงผลระยะเวลาที่ใช้บริการ ตลอดจนแสดงจำนวนค่าบริการให้ผู้บริโภคเห็นได้ชัดเจน

วัตถุประสงค์ของปริญญานิพนธ์นี้ จัดทำขึ้นเพื่อแก้ไขปัญหาดังกล่าวข้างต้น ซึ่งเป็นการนำเอาเทคโนโลยีของไมโครโปรเซสเซอร์แบบ SINGLE BOARD มาใช้ใน การควบคุมการทำงานของระบบในเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติเพราะในปัจจุบันนี้มีการนำเอาไมโครโปรเซสเซอร์ มาประยุกต์ใช้งานอย่างแพร่หลาย เนื่องจากเป็นการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้มีประสิทธิภาพและขีดความสามารถการใช้งานสูง นอกจากนี้แล้วปริญญานิพนธ์นี้ยังทำออกแบบเพื่อการใช้งานหรือการพัฒนาในอนาคต ซึ่งในรายระเอียดของระบบในเครื่องจะกล่าวในบทต่างๆ

บทที่2 กล่าวถึงทฤษฎีและหลักการต่างๆของเครื่องและระบบโทรศัพท์ สัญญาณต่างๆ ในคู่สายโทรศัพท์ ตลอดจนการทำงานของภาคต่างในโทรศัพท์ของเครื่อง คำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ และในส่วนท้ายจะกล่าวถึง ลักษณะการใช้งานของ บอร์ดไมโครคอนโทรลเลอร์ ANT-32 สำหรับในการพัฒนาโปรแกรมของระบบการทำงานให้มีประสิทธิภาพมากยิ่งขึ้น

บทที่ 3 กล่าวถึงขั้นตอนการทำงานของระบบการคำนวณและการออกแบบวงจรในส่วนต่างๆ เพื่อศึกษาการทำงานแต่ละส่วน เพื่อที่จะใช้ MICROCONTROLLER BOARD-ANT-32 ควบคุมการทำงานในแต่ละส่วนได้อย่างถูกต้องและสัมพันธ์กัน

บทที่ 4 กล่าวถึงผลการทดลองของส่วนต่างๆ ที่ได้กล่าวไว้ในบทที่ 3

บทที่ 5 เป็นส่วนของการสรุปการทำงานของเครื่องและปัญหา ข้อแนะนำในการที่จะนำปริญญานิพนธ์นี้ไปปรับปรุงให้เหมาะสมกับยุคสมัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 บทนำ

ปัจจุบันนี้การสื่อสารได้เข้ามามีบทบาทเป็นอย่างมากในชีวิตประจำวันเรียกได้ว่าจะมีการติดต่อสื่อสารกันตลอดเวลาก็ว่าได้และระบบโทรศัพท์ก็จัดว่าเป็นระบบสื่อสารที่ใกล้ตัวเรามากทีเดียวเชื่อว่าทุกคนต้องเคยใช้โทรศัพท์ในการติดต่อสื่อสารกันมาแล้ว คราวนี้จะมาดูกันว่าโทรศัพท์ที่ใช้กันอยู่ทุกวันนี้ ต้องมีขั้นตอนอย่างไรกันบ้าง ถึงทำให้เราสามารถพูดคุยกันได้

โทรศัพท์ที่เคยเห็นทั่วๆกันไปก็มีอยู่ 2 แบบคือ แบบกดปุ่ม และแบบหมุน แต่หน้าที่ของทั้ง 2 ระบบก็จะเหมือนๆกันจะต่างกันก็ตรงที่แบบกดปุ่มจะส่งสัญญาณออกไปเป็นความถี่ที่แตกต่างส่วนแบบหมุนจะส่งสัญญาณเป็นจำนวนพัลส์

2.2 หน้าที่หลักของโทรศัพท์

- เครื่องโทรศัพท์จะรับรู้ว่ามีผู้ต้องการใช้โทรศัพท์ เมื่อมีการยกหูโทรศัพท์ขึ้น

- เครื่องโทรศัพท์ส่งสัญญาณที่ เรียกว่า สัญญาณหมุน (DIAL TONE)จะบอกว่าพร้อมที่จะให้ทำการกดหรือหมุนหมายเลขที่จะติดต่อได้ซึ่งก็คือ เสียงที่ได้ยินเมื่อเวลายกหูเป็นสัญญาณเสียงที่มีความถี่ 350 เฮิรตซ์ กับ 440 เฮิรตซ์ มอดูเลต รวมกัน

- เครื่องโทรศัพท์จะทำหน้าที่ส่งรหัสหมายเลขที่ผู้เรียกต้องการจะติดต่อด้วยไปยังชุมสาย

- เครื่องโทรศัพท์จะส่งสัญญาณบอกผู้เรียกว่าหมายเลขที่ต้องการติดต่อด้วยว่างหรือไม่ถ้าว่างก็จะส่ง สัญญาณกลับ (RING BACK) ซึ่งมีความถี่ 440 เฮิรตซ์ กับ 480 เฮิรตซ์ มอดูเลตกันมา โดยจะดัง 2 วินาที แล้วเงียบ 4 วินาที สลับกันไป แต่ถ้าหมายเลขที่ต้องการจะเรียกไม่ว่างก็จะส่งสัญญาณความถี่ 480 เฮิรตซ์ กับ 620 เฮิรตซ์ มอดูเลตกันมา

- สามารถเปลี่ยนรูปพลังงานเสียงเป็นสัญญาณไฟฟ้าและสัญญาณไฟฟ้ากลับมาเป็นพลังงานเสียง

- เครื่องโทรศัพท์จะปรับระดับแรงดันอย่างอัตโนมัติ ในกรณีที่เกิดการเปลี่ยนแปลงของแรงดันขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องโทรศัพท์จะส่งสัญญาณไปยังชุมสายเพื่อแจ้งให้ทราบว่าสิ้นสุดการใช้งานแล้ว และให้ชุมสายเลิกทำการติดต่อกับอีกฝ่ายหนึ่งได้

สำหรับการหมุนหมายเลขโทรศัพท์ก็คือ การส่งสัญญาณพัลส์ (PULSE TRAIN) ตั้งแต่ 1 ถึง 10 พัลส์ เช่น ถ้ามีการส่งพัลส์ 1 พัลส์ก็หมายถึงการหมุนหมายเลขศูนย์ ส่ง 2 พัลส์ ก็หมายถึงเลขหนึ่ง ดังนั้นถ้าหมุนเลข 9 ก็จะมีการพัลส์จำนวน 10 พัลส์นั่นเอง และความเร็วในการส่งก็คือ 10 พัลส์ต่อวินาที

สำหรับโทรศัพท์ที่ใช้การกดปุ่มนั้นก็จะเป็นการส่งสัญญาณที่มีค่าความถี่ที่แตกต่างกันออกไป สำหรับแต่ละหมายเลขที่มีอยู่ 10 ตัว ความถี่ที่ส่งออกไปเป็นความถี่ที่อยู่ในย่านความถี่เสียง เพียงแต่ว่าในการกดครั้งหนึ่งจะมีสัญญาณเสียงที่มอดูเลตแล้วส่งออกไป 2 ความถี่ ตามตารางที่ 1

ความถี่ (Hz)	รหัสหรือหมายเลข			
697	1	2	3	
770	4	5	6	
852	7	8	9	
941	*	0	#	
	1209	1336	1477	ความถี่

ตารางที่ 2.1 การมอดูเลตของสัญญาณโทรศัพท์

ทางชุมสายเมื่อได้รับข้อมูลจากผู้เรียกแล้ว ก็จะแปลงสัญญาณที่ได้รับมาสั่งให้อุปกรณ์สวิตซ์ชิ่งทำงาน เพื่อทำการต่อสายให้กับผู้เรียก ถ้าปลายสายที่ต้องการติดต่อด้วยไม่ว่าง ชุมสายก็จะส่งสัญญาณไม่ว่า (busy tone) ไปยังผู้เรียกเพื่อแจ้งให้ทราบว่าไม่สามารถต่อวงจรให้ได้ แต่ถ้าปลายสายว่างชุมสายก็จะส่ง สัญญาณเรียก (ringing signal) ไปยังปลายสาย และส่งสัญญาณเรียกกลับ (ringback tone) ไปยังผู้เรียกเพื่อแจ้งให้ทราบว่าสามารถต่อวงจรให้ได้ตามต้องการแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

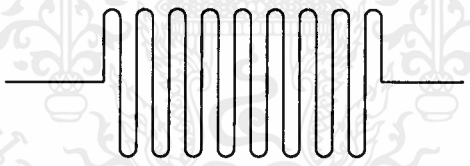
2.3 สัญญาณต่างๆ ในระบบชุมสายโทรศัพท์

สัญญาณต่างๆในระบบชุมสายโทรศัพท์ที่ส่งมายังสมาชิกผู้ใช้ (SUBSCRIBER) เป็นการบอกสถานะการติดต่อของอุปกรณ์ส่วนต่างๆในระบบโทรศัพท์และแจ้งให้ผู้ใช้ทราบว่าควรทำอย่างไรเมื่อได้รับสัญญาณแต่ละชนิด ในบทนี้จะกล่าวถึงรายละเอียดของสัญญาณต่าง ๆ ที่จำเป็นต้องในการควบคุมการทำงานของเครื่องคิดเงินการใช้โทรศัพท์

2.3.1 สัญญาณสมาชิก

สัญญาณสมาชิกคือสัญญาณที่เครื่องชุมสายโทรศัพท์แจ้งสภาวะต่างๆ ในการติดต่อให้ผู้เรียกทราบว่าควรทำอย่างไรเมื่อได้รับสัญญาณสัญญาณสมาชิกประกอบด้วย

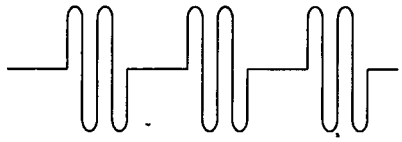
สัญญาณให้หมุน (DT : dial tone) เป็นสัญญาณเพื่อให้สมาชิกทราบว่าสามารถเริ่มทำการติดต่อส่งเลขหมายของผู้รับได้ สัญญาณให้หมุนได้นี้เป็นสัญญาณแบบ sine wave มีความถี่ 400 Hz ส่งมาอย่างต่อเนื่องและมีระดับขนาด 400 mVp-p ดังรูปที่ 2.1



รูปที่ 2.1 สัญญาณให้หมุน

สัญญาณไม่ว่าง(BT:busy tone)เป็นสัญญาณเพื่อให้สมาชิกทราบว่าผู้รับสายไม่ว่างหรือการต่อระหว่างชุมสาย (TRUNK) ไม่ว่างผู้เรียกจึงควรวางหูโทรศัพท์สักระยะเวลาหนึ่ง แล้วจึงเริ่มทำการเรียกใหม่ สัญญาณไม่ว่างเป็นสัญญาณ sine wave มีความถี่ 400 Hz ส่งมาในคู่สายเป็นช่วงๆ โดยเป็นจังหวะดัง 0.5 วินาทีสลับกันและมีขนาด 250-300 mVp-p ดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



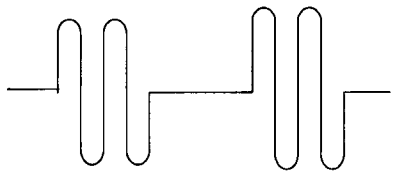
รูปที่ 2.2 สัญญาณไม่ว่าง

สัญญาณเรียกลับ (RBT :Ringback tone) เป็นสัญญาณเพื่อว่าการต่อทุกขั้นตอนตามความต้องการของผู้เรียก เครื่องชุมสายโทรศัพท์สามารถดำเนินการติดต่อสำเร็จและแจ้งให้ผู้เรียกทราบสัญญาณเรียกกลับเป็นสัญญาณ sine wave มีความถี่ 400 Hz ส่งออกมาเป็นช่วงๆ เป็นจังหวะดัง 1 วินาที และเงียบ 4 วินาที สลับกันไปและมีระดับขนาด 400 mVp-p ดังรูปที่ 2.3

รูปที่ 2.3 สัญญาณเรียกกลับ

สัญญาณกริ่งเรียก (RGT : Ringing tone) เป็นสัญญาณเพื่อว่าการต่อทุกขั้นตอนตามความต้องการของผู้เรียกไปยังผู้รับ เครื่องชุมสายโทรศัพท์สามารถดำเนินการติดต่อสำเร็จและส่งสัญญาณกริ่งให้ผู้รับมาตอบรับการเรียก สัญญาณกริ่งเรียกเป็นสัญญาณ sine wave ที่มีความถี่ 20 Hz ส่งมาเป็นช่วงๆโดยมีจังหวะดัง 1 วินาทีและเงียบ 4 วินาที มีระดับขนาด 100 Vp-p ดังรูป 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 สัญญาณกริ่งเรียก

2.4 ระบบการกำหนดหมายเลขโทรศัพท์

การต่อโทรศัพท์ของสมาชิกผู้เช่าแต่ละรายผ่านชุมสายโทรศัพท์จำเป็นต้องมีเลขหมายแทนชื่อผู้เช่าและเลขหมายที่ใช้จะต้องเกิดความสะดวกรวดเร็วในการเรียกติดต่อในประเทศไทย การเรียกโทรศัพท์ภายในท้องถิ่นเดียวกัน กำหนดว่าจะต้องหมุนหมายเลขของชุมสายท้องถิ่น (LOCAL OFFICE) และจึงหมุนหมายเลขของผู้เช่าในกรณีการต่อในเขตอื่นจะต้องมีเลขการผ่านศูนย์ทางไกลซึ่งเป็นเลข 0 ก่อน แล้วจึงหมุนหมายเลขของศูนย์โทรศัพท์ทางไกล (TOLLCENTER) ต่อจากนั้นจะเป็นเลขหมายของชุมสายท้องถิ่นและเลขหมายของผู้เช่า

กรณีการเรียกท้องถิ่นเดียวกัน

K K K - X X X X

หมายเลขชุมสายท้องถิ่น หมายเลขของผู้เช่า

กรณีเรียกทางไกล

0 F F K K K - X X X X

หมายเลขศูนย์ทางไกล ชุมสายท้องถิ่น หมายเลขผู้เช่า

2.5 ลักษณะของสัญญาณโทรศัพท์

สัญญาณโทรศัพท์มีทั้งสัญญาณดีซี (DC) และเอซี (AC) ซึ่งจะพิจารณาสัญญาณโทรศัพท์ได้เป็น 2 ลักษณะตามวิธีการใช้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 เมื่อเป็นผู้เรียก

- เมื่อโทรศัพท์ยังไม่ใช้หรือยังไม่ได้ยกหู สัญญาณระหว่างคู่สายเป็น 48 โวลท์ ดีซี
- เมื่อยกหูขึ้นสัญญาณระหว่างสายโทรศัพท์ตกลงเป็น 10 โวลท์ดีซี ในขณะเดียวกันมีสัญญาณ 600 มิลลิโวลท์ผสมมา

ความถี่ (Hz)		กลุ่มความถี่สูง (K1)			
		H1 1,209 (11.15)	H2 1,336 (11.14)	H3 1,477 (11.13)	H4 1,633
กลุ่มความถี่ต่ำ (K1)	L1 697 (1-5)	1	2	3	A
	L2 770 (1-4)	4	5	6	B
	L3 852 (1-3)	7	8	9	C
	L4 941 (1-2)	*	0	#	D

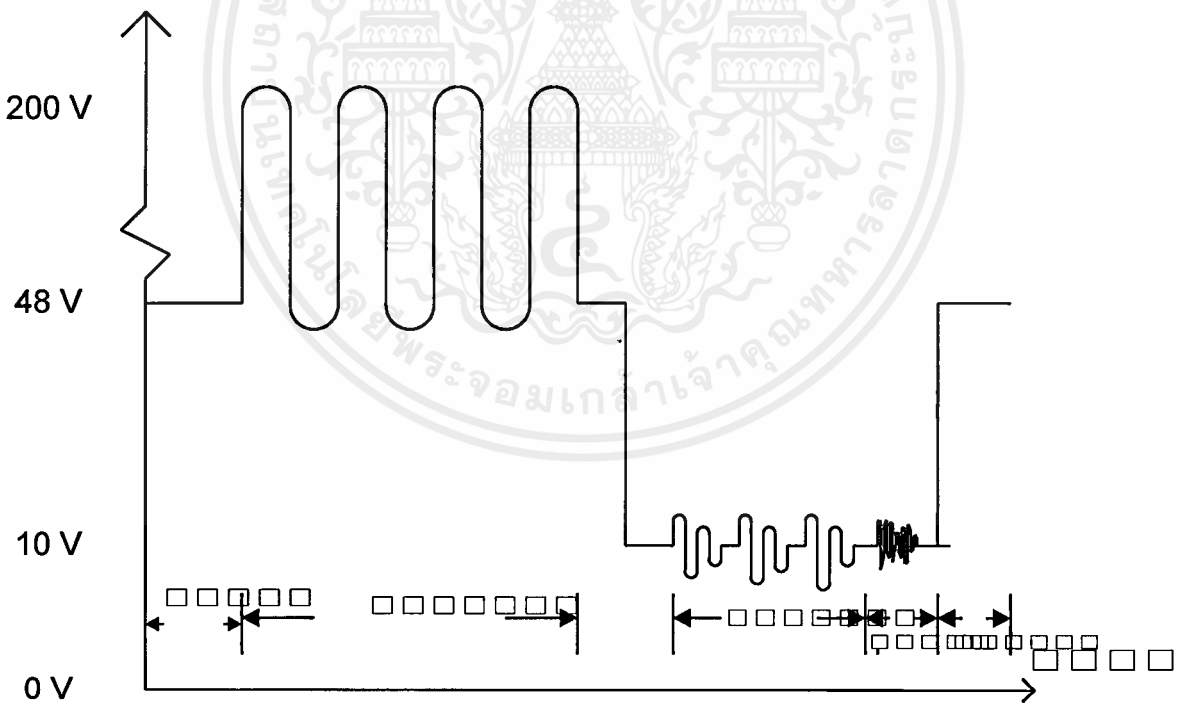
ตารางที่ 2.2 ความถี่และการจัดปุ่ม

- เมื่อหมุนหรือกดหน้าปัทม์ เพื่อเรียกไปยังหมายเลขที่ต้องการติดต่อด้วย ในแบบหมุนหน้าปัทม์ (DIAL) จะส่งพัลซ์ (PULSE) จำนวนลูกเท่ากับจำนวนพัลซ์ที่หมุนโดยจะส่งพัลซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแบบ 10 PPS หรือ 20 PPS สำหรับในแบบกดปุ่มหน้าปัทม์ จะส่งสัญญาณคู่ความถี่ (DUAL TONE MULTI-FREQUENCY) ซึ่งเป็นสัญญาณของกลุ่มความถี่ต่ำ และกลุ่มความถี่สูงเป็นความถี่มาตรฐานที่กำหนดไว้ในตารางที่ 2.2

- รอสัญญาณหลังการหมุนหรือกดปุ่มหน้าปัทม์ ถ้าได้รับสัญญาณเรียกกลับ ว่ากำลังมีการเรียกไปยังหมายเลขที่ต้องการติดต่ออยู่จะเป็นสัญญาณเป็นจังหวะดัง 1 วินาที หยุด 4 วินาที สลับกัน ความถี่ 440 เฮิร์ต (Hz) ระดับสัญญาณ 200 มิลลิโวลท์ แต่ถ้าหากว่าได้รับสัญญาณสายไม่ว่าง จะเป็นสัญญาณดังและหยุดสลับกันเป็นจังหวะทุก ๆ 0.5 วินาที ความถี่ 560 เฮิร์ต (Hz) ระดับสัญญาณ 400 มิลลิโวลท์

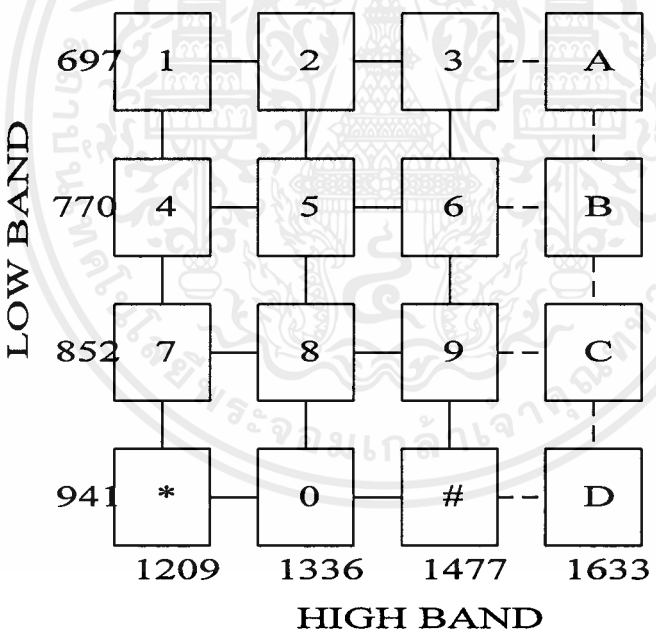


รูปที่ 2.5 ลักษณะสัญญาณโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 DTMF

DTMF (Dual Tone Multi Frequency) เป็นการเข้ารหัสความถี่คู่ของหมายเลขโทรศัพท์และเรียกว่าโทรศัพท์กดปุ่ม (Touch Tone) ระบบ DTMF ดีกว่าระบบพัลส์ คือลดเวลาของการหมุนเลขหมายของผู้ใช้โทรศัพท์ให้มีช่วงเวลาในการหมุนเลขหมายน้อยลง ซึ่งระบบของ DTMF จะประกอบด้วยคู่ความถี่ และจะส่งไปยังชุมสายโทรศัพท์โดยใช้คู่ความถี่ดังกล่าว ในคู่ความถี่นั้นจะมีความถี่สูง (High Band) และความถี่ต่ำ (Low Band) รวมกันไปโดยเมื่อส่งไปยังชุมสาย ชุมสายจะถูกถอดรหัสเป็นเลขหมายอีกครั้งหนึ่งระบบ DTMF จะลดเวลาการหมุนได้ดีกว่าระบบพัลส์หรือแบบหมุนถึง 10 เท่า และมีการผิดพลาดน้อยมาก



รูปที่ 2.6 การเข้ารหัส

2.5.3 เมื่อเป็นผู้รับ

ขณะยังวางหูอยู่ สัญญาณระหว่างสายโทรศัพท์เป็น 48 โวลท์ดีซี

เมื่อมีสัญญาณกระดิ่ง (RINGING) จะมีสัญญาณเอซีความถี่ 16 เฮิรตซ ระดับสัญญาณ 270 โวลท์ ดัง 1 วินาทีและหยุด 4 วินาทีสลับกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อยกหูโทรศัพท์ สัญญาณระหว่างสายโทรศัพท์จะตกลงไปเป็น 10 โวลท์ดีซี

2.6 ภาคถอดรหัส

FW	F	NO	TOE	Q4	Q3	Q2	Q1
697	1290	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	*	H	1	0	1	0
941	1209	0	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

ตารางที่ 2.3 การถอดรหัสความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลขโดยใช้เทคนิคการนับแบบดิจิตอล และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสมเมื่อตรวจสอบว่าความถี่นั้นมีความถูกต้องสัญญาณที่ ขา Est (EARLY STEERING) ก็จะแอคทีฟสำหรับค่าที่ถอดรหัสได้จากความถี่ต่างๆ นั้น ในตารางที่ 2.3

ข้อดีของการใช้โทรศัพท์แบบโทน

1. ลดระยะเวลาในการส่งเลขหมายโทรศัพท์ไปยังชุมสายโทรศัพท์
2. สามารถใช้ไอซีสำเร็จรูปทำให้มีขนาดเล็กลง
3. ลดจำนวนอุปกรณ์ในการกำหนดเลขหมาย ทำให้ชุมสายมีขนาดเล็กลง
4. สามารถนำไปใช้เชื่อมต่อกับอุปกรณ์ภายในชุมสายโทรศัพท์ได้อย่างมีประสิทธิรูป
5. สามารถนำไปใช้กับระบบอัตโนมัติต่างๆ ได้กว้างขวางขึ้น เช่น ระบบต่อเลขหมายภายในอัตโนมัติ และชุดตอบรับและโอนสายโทรศัพท์อัตโนมัติ เป็นต้น

2.7 ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาท์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้าว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตุจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความออกมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา Est จะเป็น “HIGH” นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา จากรูปที่ 9 เมื่อขา Est เป็น “HIGH” ทำให้ Vc สูงขึ้น ตัวเก็บประจุ C จะคายประจุทำให้แรงดัน Vc สูงขึ้นจนถึงค่าเทรชโฮลด์วงจรถอดรหัสจึงจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิท รายละเอียดการทำงานขอให้ดูจากแผนภูมิเวลาหรือไทมิ่งไดอะแกรม (TIMING DIAGRAM) ในรูปที่ 8 จะเข้าใจได้ง่ายกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MT8870

การถอดรหัสความถี่โทรศัพท์ หมายถึง การแปลงสัญญาณความถี่คู่ ซึ่งเกิดจากการกดปุ่มตัวเลขของโทรศัพท์ชนิดกดปุ่ม ให้เป็นระบบตัวเลขทางดิจิตอล ซึ่งไอซี MT8870 ทำหน้าที่แปลงความถี่คู่ให้เป็นเลขฐานสองขนาด 4 บิต

คุณสมบัติของ MT8870

- เป็นตัวรับและถอดรหัสความถี่คู่ (DTMF Receiver)
- ใช้กระแสน้อยและระดับไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการ์ดไทม์ (Guard Time) ได้
- เป็นไอซีคุณรูปสูง

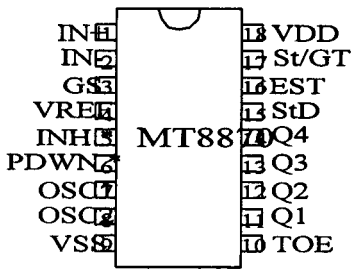
การนำ MT8870 ไปใช้งาน

- นำไปใช้งานด้านการควบคุมระยะไกล
- ใช้ในงานเกี่ยวกับเครดิตการ์ด
- ใช้งานร่วมกับคอมพิวเตอร์
- ใช้ในเครื่องชุมสายขนาดย่อย หรือ PABX
- ใช้ในโทรศัพท์ทั่วไป
- เครื่องกันขโมย
- การควบคุมอุปกรณ์ทางโทรศัพท์
- ใช้ทำเครื่องสอบถามทางโทรศัพท์

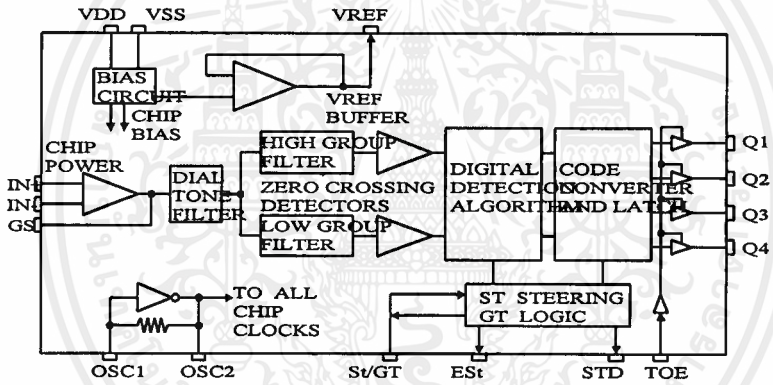
โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่และวงจรถอดรหัสฟังก์ชั่นทางดิจิตอล เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO^2-CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิตช์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอลเพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐานสองขนาด 4 บิต และตรวจสอบช่วงเวลาที่สัญญาณเข้ามา ส่วนภาคอินพุตเป็นออปแอมป์ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอกเอาต์พุตเป็นวงจรแลตช์ 3 สถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 รายละเอียดขา MT8870

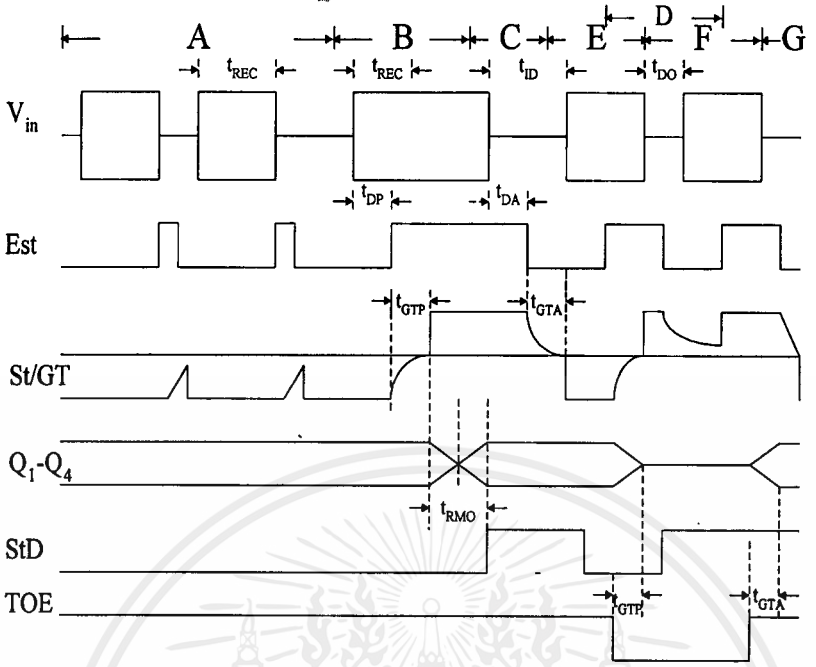


รูปที่ 2.8 โครงสร้างภายในของ MT8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศ 118 ศ 2538

ห้องสมุด
คณะครุศาสตร์อุตสาหกรรม สจล.



รูปที่ 2.9 แผนภูมิเวลาของ MT8870

อธิบายขั้นตอนการทำงาน

A - ตรวจพบความถี่เข้ามาแต่คาบเวลาที่ถูกต้อง เอาท์พุตไม่เปลี่ยน

B - ความถี่ #n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัสและแลตซ์ไว้ที่เอาท์พุต

C - จบความถี่ #n ช่วงห่างถูกต้อง เอาท์พุตยังคงแลตซ์อยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่

D - เอาท์พุตเปลี่ยนเป็นไฮอิมพีแดนซ์

E - ความถี่ #n+1 ถูกตรวจพบคาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแลตซ์ไว้

F - ความถี่ #n+1 หายไป ช่วงห่างไม่ถูกต้อง เอาท์พุตยังคงแลตซ์อยู่

G - จบความถี่ #n+1 ช่วงห่างถูกต้อง เอาท์พุตยังคงแลตซ์อยู่จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายคำศัพท์

V_{in} - สัญญาณความถี่ DTMF ที่เข้ามา

Est - EARLY STEERING OUTPUT ใช้ความถี่ที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1534 021303

St/GT - STEERING INPUT / GUARD TIME OUTPUT สำหรับต่อกับ RC ภายนอก

Q_1-Q_4 - เอาท์พุต BCD ขนาด 4 บิท

StD - DELAYED STEERING OUTPUTใช้ว่าความถี่ที่ได้รับหรือหายไปมีคาบเวลาตามที่กำหนด เพื่อความถูกต้องของสัญญาณ

t_{ID} - TONE OUTPUT ENABLE (INPUT) ใช้ควบคุม Q1-Q4 ให้เป็นไฮอิมพีแดนซ์

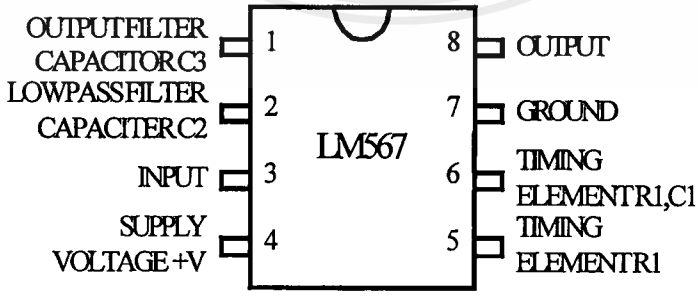
t_{REC} - คาบเวลาที่นานที่สุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง t_{REC} - คาบเวลาที่สั้นที่สุดที่ต้องการเพื่อว่าสัญญาณถูกต้อง

t_{ID} - เวลาสั้นที่สุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

t_{DO} - เวลานานที่ได้ที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

LM 567

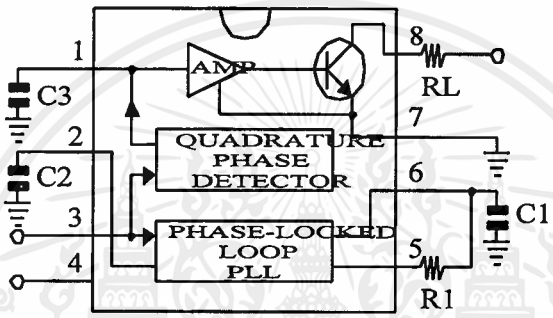
LM567 เป็นไอซีโทนดีโค้ดเดอร์ที่ใช้ถอดรหัสสัญญาณความถี่เสียง อินพุตที่รับเข้ามาภายในตัวไอซีนี้จะประกอบด้วย วงจรเฟสล็อกลูป (Phase lock loop) และทรานซิสเตอร์ที่ทำหน้าที่เป็นสวิตช์ เอาต์พุตของทรานซิสเตอร์ตัวนี้ จะต่อออกไปจากขาคอลเล็คเตอร์ ให้สัญญาณรูปสี่เหลี่ยมออกมา เมื่อสัญญาณเสียงที่เป็นสัญญาณอินพุตถูกส่งเข้ามาทางขาอินพุตของไอซีโทนดีโค้ดเตอร์ จะทำการถอดรหัสสัญญาณเสียงได้หลายความถี่ ซึ่งในที่นี้ได้นำมาตรวจจับสัญญาณโทรศัพท์



รูปที่ 2.10 ขาของไอซีเบอร์ LM567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นผังจากรูปที่การทำงานหลักภายในไอซี คือ วงจรเฟสล็อกลูป (PLL), วงจรแยกเฟส 90 องศา (QUADRATURE PHASE DETECTOR), วงจรขยาย (AMPLIFIER) และทรานซิสเตอร์ที่ต่ออยู่ทางด้านเอาต์พุตของตัวไอซี ภายในเฟสล็อกลูป (PLL) จะประกอบด้วยวงจรออสซิลเลเตอร์ที่ควบคุมด้วยกระแส (Current Controlled Oscillator :CCO), Pase detector และวงจรกรองสัญญาณป้อนกลับ



รูปที่ 2.11 ผังการทำงานของไอซีเบอร์ LM567

หลักการทำงาน

เมื่อ LM567 ทำงานในรูปของสวิตช์ทางเสียง จะตรวจจับความถี่กลางซึ่งเป็นความถี่ที่ผู้ใช้กำหนดให้เป็นค่าใดก็ได้ภายในช่วงความถี่ 0.1 KHz ถึง 500 KHz และความกว้างของแถบความถี่หรือแบนด์วิดท์ที่มันจะจับได้ (ซึ่งสามารถกำหนดได้สูงสุด 14% ของความถี่กลาง)

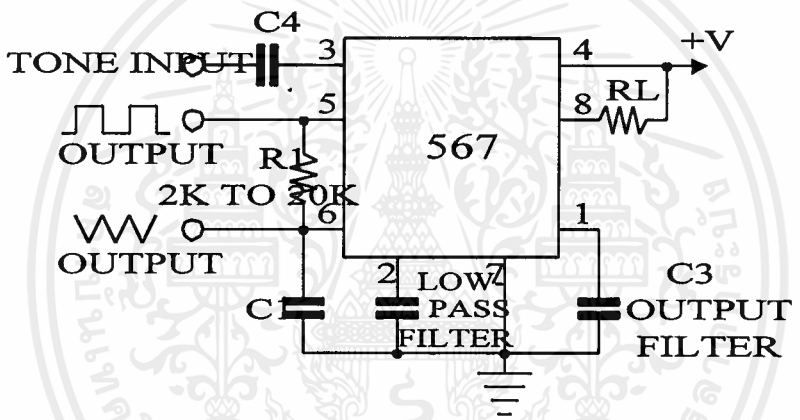
การหน่วงเวลาของไอซี คือ ระยะเวลาที่เริ่มตั้งแต่ที่ไอซีได้รับสัญญาณเข้ามา จนกระทั่งให้สัญญาณเอาต์พุตออกมาทางขาเอาต์พุต เราสามารถเพิ่มการหน่วงเวลาได้โดยการกำหนดค่าตัวต้านทานและตัวเก็บประจุที่อยู่ภายนอกได้

วงจร CCO ของไอซีนี้ สามารถเพิ่มความกว้างของช่วงความถี่ทำงาน จากการกำหนดค่าความต้านทานของ R1 และค่าของ C1 ที่ต่ออยู่ภายนอก (รูปที่ 2.14) แต่ความถี่ที่เราควบคุมนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ในช่วงแคบๆ เท่านั้น โดยเมื่อรับสัญญาณอินพุตเข้ามาที่ขา 2 จะทำให้วงจร PPL ล็อกเอาช่วงความถี่แคบๆ ของสัญญาณนี้ไว้

เฟสดีเท็คเตอร์แบบวงจรแยกเฟสของไอซีนี้ จะเปรียบเทียบความถี่ และเฟสของสัญญาณอินพุต/เอาต์พุตของวงจรเฟสดีเท็คเตอร์จะถูกส่งไปขับทรานซิสเตอร์ Q1 เพื่อให้ทำงาน เมื่อสัญญาณที่ทำการเปรียบเทียบทั้งสองสอดคล้องกัน ความถี่กลางของไอซีจะมีค่าเท่ากับความถี่ฟรีรันนิ่งของออสซิลเลเตอร์ และค่าแบนด์วิดท์จะมีค่าเท่ากับช่วงความถี่ที่มีการล็อกเอาไว้ในวงจร PLL



รูปที่ 2.12 ตัวอย่างการต่อใช้งานไอซี LM567

จากรูปที่ 2.13 ไอซี LM567 จะทำงานเป็นสวิตช์ทางเสียง สัญญาณอินพุตจะผ่านตัวเก็บประจุ C4 ไปยังขา 3 ซึ่งมีอินพุตอิมพีแดนซ์ประมาณ20 กิโลโอห์มส่วนตัวต้านทานทางด้านเอาต์พุต ต่อระหว่างขา 8 และจ่ายแรงดันค่าบวก ซึ่งค่าของแรงดันมีค่ามากที่สุดได้ 15 โวลต์

สำหรับตัวต้านทานที่ขา 8 จะดึงกระแสสูงถึง 100 มิลลิแอมป์ ส่วนขา 7 ต่อลงกราวด์ ขา 4ต่อกับแหล่งจ่ายแรงดันแบบบวก (ค่าน้อยที่สุดที่วงจรจะทำงานได้คือ 4.75 โวลต์ ค่ามากที่สุดคือ 9 โวลต์)

ความถี่กลางของวงจรสามารถคำนวณหาได้จากสูตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$fo = \frac{1.1}{(R1 \times C1)}$$

เมื่อ R1 มีหน่วยเป็นกิโลโอห์ม และ C1 มีหน่วยเป็นไมโครฟารัด

จากสูตร $fo = \frac{1.1}{(R1 \times C1)}$ ค่า R1 สามารถหาได้จาก

$$R1 = \frac{1.1}{(fo \times C1)}$$

ค่า R1 ที่ใช้งานควรอยู่ในช่วง 2 ถึง 20 กิโลโอห์ม ส่วนค่า C1 สามารถกำหนดได้ตามความเหมาะสม ค่า R1 ก็คำนวณหาค่าจากสูตรข้างบน

วงจรออสซิลเลเตอร์ในไอซี LM567 จะให้เอาต์พุตออกมาเป็นรูปคลื่นฟันเลื่อยที่ขา 6และรูปคลื่นสี่เหลี่ยมที่ขา 5, C2 จะเป็นตัวกำหนดแบนด์วิดท์ของไอซี เพื่อทำงานเป็นสวิตช์ทางเสียง

2.8 ANT-32 บอร์ดไมโครคอนโทรลเลอร์

ชิพไมโครคอนโทรลเลอร์ที่ถูกนำมาประยุกต์ใช้กับระบบควบคุมอัตโนมัติในระดับ 8 บิท ที่โดดเด่นมากตัวหนึ่งก็คือ ชิพไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของ INTEL ซึ่งประกอบไปด้วย CPU เบอร์ต่าง ๆ ได้แก่ 8032, 805, 8052, 8751, 8752 และ 8052 AHBASIC

ANT-32 เป็นบอร์ดไมโครคอนโทรลเลอร์นำไปใช้ในลักษณะ EMBEDDED CONTROLLER กล่าวคือเป็นบอร์ดที่ถูกออกแบบมาเพื่อควบคุมงานโดยเฉพาะโดยถูกติดตั้งอยู่ในเครื่องจักรกลเครื่องใช้ไฟฟ้า รวมทั้งระบบอัตโนมัติต่างๆบอร์ดนี้สามารถกับเบอร์ CPU ดังกล่าวมาแล้วทั้งสิ้น ANT-32 ได้ถูกออกแบบมาประกอบไปด้วยวงจรในส่วนของ Watchdog Timer, Battery Backup และ POWER Fail Detector ใช้ชิพ MAX691 วงจร Real Time Clock ใช้ชิพ DS1202

2.8.1 คุณสมบัติ ANT-32

เป็นบอร์ดคอนโทรลใช้กับไมโครคอลโทรลเลอร์ในตระกูล MCS-51 (8031/8032) ใช้กับ CPU เบอร์ 8032 ทำงานที่ความถี่ 11.0592 Mhz

ใช้งานหน่วยความจำบนบอร์ด 3 ตำแหน่งด้วยกันคือ U2 เป็นหน่วยความจำโปรแกรม (PROGRAM MEMORY) ใช้กับ EPROM ขนาด 8-32 Kby เบอร์ 2764, 27128 หรือ 27256

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

U3 เป็นหน่วยความจำข้อมูล (DATA MEMORY) ใช้กับ RAM ขนาด 8 Kbyte เบอร์ 6264 หรือ 32 Kbyte เบอร์ 62256 สามารถแบคอัพข้อมูลได้โดยใช้แบตเตอรี่ลิเธี่ยม

U4 เป็นหน่วยความจำโปรแกรมและข้อมูล (PROGRAM AND DATA MEMORY) ใช้กับ EPROM, RAM หรือ EPROM ขนาด 8-32 Kbyte โดยใช้ EPROM เบอร์ 2764(A), 28256(A)

มีพอร์ต I/O เบอร์ 8255 จำนวน 2 ตัว (48) สำหรับต่อสายนอก

มีวงจร SERIAL INTER DRIVER RS232 ด้วยชิพเบอร์ MX232 สำหรับต่อเข้ากับเครื่องไมโครคอมพิวเตอร์

มีวงจร Watching Timer, Powerup/down Reset ด้วยชิพเบอร์ MAX691

มีวงจร RTC (Real Time Clock) ใช้ชิพเบอร์ DS 1202

มีคอนเน็ตเตอร์สำหรับหา Port 1 ของไมโครคอนโทรเลอร์โดยเฉพาะ

มีคอนเน็ตเตอร์สำหรับ SYSTEM BUS ทำให้ขยายระบบได้ง่ายและสามารถใช้กับบอร์ดขยายต่าง ๆ ที่จะมีขึ้นในอนาคต

สามารถเลือกเบอร์และชนิดหน่วยความจำ หรือกำหนดคุณสมบัติต่าง ๆ ของบอร์ดได้ด้วยจัมพ์เปอร์

สามารถพัฒนาโปรแกรมได้ทั้ง ภาษาเบสิค และ แอสเซมบลี้โดยใช้บอร์ดฟรอม BASIC32 และ REM31

ก่อนใช้งานบอร์ด ANT-32 ผู้ใช้จำเป็นต้องเลือกขนาด, เบอร์ของหน่วยความจำและกำหนดคุณสมบัติต่าง ๆ ของบอร์ดให้ถูกต้องด้วยจัมพ์เปอร์ซึ่งมีทั้งหมด 9 ชุดดังนี้

จัมพ์เปอร์ EA สำหรับเลือกใช้หน่วยความจำโปรแกรม (PROGRAM MEMORY) ตำแหน่งแอดเดรสเริ่มต้น OOOOH เป็น INT (INTERNAL) หรือ EXT. (EXTERNAL)

จัมพ์เปอร์ REST สำหรับเลือกสัญญาณรีเซท CPU จากวงจร RC หรือ MAX691

จัมพ์เปอร์ U2 SIZE สำหรับเลือกขนาดหน่วยความจำโปรแกรม U2 (EPROM) เป็น 8, 16 Kbyt (2764, 27128) หรือ Kbyte (27256)

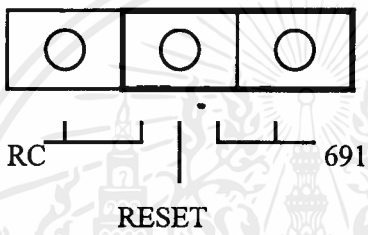
จัมพ์เปอร์ U3 SIZE สำหรับเลือกขนาดหน่วยความจำข้อมูล U3 (RAM) เป็น 82 Kbyte (6264) หรือ 32 Kbyte (62256)

จัมพ์เปอร์ U4 TYPE สำหรับเลือกชนิดหน่วยความจำ U4 เป็น EPROM (หน่วยความจำโปรแกรม) หรือ RAM (หน่วยความจำข้อมูล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 OPTIONAL MICROPROCESSOR SUPERVISORY CIRCUIT

MAX691 Microprocessor Supervisory Circuit เป็นชิพของ MAXIM ใช้สำหรับจัดการเกี่ยวกับการมอนิเตอร์เพาเวอร์ซัพพลาย (Power Supply Monitoring) และการควบคุมแบตเตอรี่ (Battery Control) ซึ่งเป็นวงจรในส่วนที่มีความสำคัญอย่างยิ่งกับระบบไมโครโปรเซสเซอร์อันได้แก่ Microprocesser reset, Power Fail Detector, CMOS RAM Write Protection และ Watchdog Time ในที่นี้จะขอกล่าวถึงรายละเอียดส่วนต่าง ๆ ของ MAX691 (U) เท่าที่จำเป็นต่อการใช้งานของบอร์ด ANT-32 เท่านั้น



รูปที่ 2.13 จัมพ์เปอร์รีเซท

Microprocesser reset (จัมพ์เปอร์ RESET)

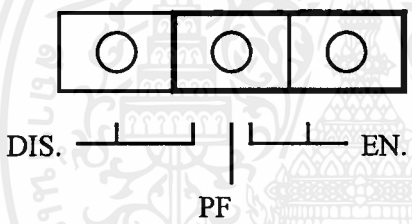
สัญญาณรีเซท CPU บนบอร์ด ANT-32 สามารถเลือกใช้ได้จาก 2 แหล่งคือ RC (จากวงจร R/C รีเซท) หรือ 691 (จากวงจรรีเซท MAX691) โดยจัมพ์เปอร์ RESET
ถ้าไม่ได้ใช้งาน MAX691 ให้ปรับจัมพ์ไปที่ตำแหน่ง RC วงจร R/C รีเซทจะทำการรีเซท CPU ในช่วง POWER UP เท่านั้น ส่วนสัญญาณรีเซท CPU ของ MAX691 จะทำการรีเซท CPU ในช่วง Power up และ Power down โดยที่ขา RESET จะเป็นลอจิก "1" เมื่อแรงดัน Vcc 50 mS ซึ่งก็หมายความว่า CPU จะถูกรีเซทเมื่อเริ่มจ่ายไฟด้วยพัลส์ที่มีความกว้าง 50 mS และจะรีเซทอีกครั้งเมื่อไฟตกนอกจากนี้แล้วขา RESET จะถูกใช้เมื่อเลือกใช้งาน Watching Timer

2.8.3 Power Fail Detector (จัมพ์เปอร์)

วงจรตรวจจับแรงดันไฟตก(Power Fail Detector)สำหรับระบบไมโครคอนโทรลเลอร์นับว่ามีความสำคัญมากกับระบบที่ต้องการเก็บค่าพารามิเตอร์หรือข้อมูลบางอย่างลง RAM ก่อนที่ระบบจะหยุดการทำงานพื่อครั้งจะได้นำข้อมูลที่เก็บไว้ก่อนไฟดับมาประมวลเพื่อใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานต่อไป โครงสร้างภายในของวงจรนี้เป็นวงจร (Voltage Comparator) โดยรับแรงดันอินพุตที่ต้องการตรวจจากภายนอกเข้าที่ขา PFI (Power Fail Input) นำมาเปรียบเทียบกับแรงดันอ้างอิง (reference voltage) 1.3 V ซึ่งขาเอาท์พุตคือ ขา PFO (Power Fail Output) จะเป็นลอจิก "0" เมื่อแรงดันที่ขา PFI ต่ำกว่า 1.3 V ขา PFI รับแรงดันมาจากวงจร Voltage divider ภายนอกซึ่งก็คือ Rx และ Ry โดยทำการตรวจแรงดัน Vcc 5 V ค่าอัตราของวงจร voltage divider สามารถกำหนดได้จากหลักการที่ว่าแรงดันที่ขา PFI จะตกลงถึงค่า 1.3 V ก่อนที่แรง +5 V จากแหล่งจ่ายจะตกลงถึง 4.75 V โดยปกติแล้วขา PFO จะต่อเข้ากับอินเตอร์รัพท์ของ CPU เพื่อที่ว่าที่เกิดสรูปวะไฟตก CPU จะถูกอินเตอร์รัพท์เพื่อให้ CPU ไปทำขบวนการเอาข้อมูลที่จำเป็นเก็บลง RAM ก่อนที่แรงดัน Vcc จะตกลงต่ำกว่า 4.75 V และแรงดันนี้เองที่ CPU จะถูกรีเซทอีกครั้ง



รูปที่ 2.14 จัมพ์เปอร์ของวงจรตรวจจับแรงดันไฟตก

ขา PFO ของ MAX691 จะต่อเข้าขา INT 1 ของ CPU สามารถเลือกใช้งานวงจร Power Fail detector นี้ด้วยจัมพ์เปอร์ PF โดยปรับไปที่ EN. (enable) หรือไม่ใช้งานโดยปรับไปที่ DIS. (disable)

เลือกใช้ค่า Rx = 27 K, Ry = 10 K

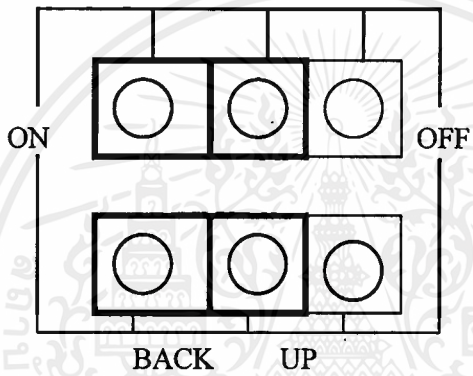
* ค่าแรงดัน 4.8 V สามารถเปลี่ยนแปลงได้ตามลักษณะงานที่ใช้ *

2.8.4 CMOS RAM Write Protection (จัมพ์เปอร์ BACKUP)

การสำรองข้อมูลของ CMOS RAM (U3) และ RTC (DS 1202) ในช่วงไฟดับสำหรับบอร์ด ANT-32 ใช้ MAX691 จัดการในส่วนของแบตเตอรี่ และสัญญาณ Chip ของ RAM โดย MAX691 รับแรงดังจากแบตเตอรี่ลิเธี่ยม 3 V เข้าที่ขา Vbatt และแรงดัน Vout จาก MAX691

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อเข้าที่ขา Vcc ของ CMOS RAM และ RTC ในการใช้งานปกติใช้ไฟ 5V ที่ขา Vout จะเสมือนต่อขากับขา Vcc ภายใน MAX691 และจะต่อเข้ากับขา Vabtt เมื่อแรงดัน Vcc ต่ำกว่าแรงดันแบตเตอรี่ ดังนั้นแบตเตอรี่จะถูกใช้งานในช่วงไฟดับเท่านั้น สำหรับ CMOS RAM เพื่อป้องกันข้อมูลสูญหายในขณะไฟดับ MAX691 ได้ออกแบบวงจรควบคุมสัญญาณ Chip Enable ของ RAM โดยที่ขา CE IN ของ MAX691 รับสัญญาณ Chip Enable มาจากวงจร Address Decoder และสัญญาณที่ขา CE OUT จาก MAX691 จะต่อเข้าขา Chip Enable ของ RAM



รูปที่ 2.15 จัมพ์เปอร์ของการสำรองข้อมูล

สัญญาณที่ขา CE OUT จะเป็นไปตามสัญญาณที่ขา CE IN ก็ต่อเมื่อแรงดัน Vcc ต่ำกว่า 4.65 v และจะเป็นลอจิก "1" เมื่อแรงดัน Vcc ต่ำกว่า 4.65 V เพื่อป้องกัน CPU เขียนค่าข้อมูลที่ไม่ถูกต้องลง RAM ในระหว่าง Poe\wer up และ Power down การสำรองข้อมูลของ RAM (U3) สามารถเลือก ON/OFF ได้ด้วยจัมพ์เปอร์ BACKUP ซึ่งมีอยู่ 2 ตัว ส่วน RTC นั้นจะทำการสำรองข้อมูลตลอดเวลาไม่สามารถ OFF ได้

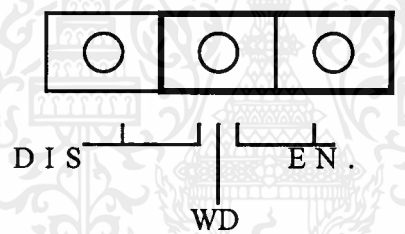
2.9 Watchdog Timer (จัมพ์เปอร์ WD)

Watchdog Timer เป็นวงจรที่ทำการตรวจสอบการทำงานของระบบไมโครคอนโทรลเลอร์ ว่าทำงานในสภาวะปกติหรือไม่ ถ้าระบบการทำงานผิดปกติ วงจรส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้จะทำการรีเซท CPU ให้เริ่มทำงานใหม่อีกครั้ง Watching Timer จึงมีความสำคัญและจำเป็นมากสำหรับระบบไมโครฯ ซึ่งจะเป็นการเพิ่มเสถียรรูปการทำงานของระบบให้ดีขึ้น

หลักการของ Watchdog Timer ก็คือ CPU ต้องส่งสัญญาณไปกระตุ้นที่ขา WDI (WADCHDOG INPUT) ของ MAX691 โดยพอร์ต P1.7 ที่ขา OSC IN และ OSC SEL ของ MAX691 ไม่ได้ต่อใช้งาน CPU ต้องทำการเปลี่ยนสภาวะที่ขา WDI ทุก ๆ 1.6 วินาที โดยใช้คำสั่ง CPL P1.7 เพื่อให้แน่ใจว่าซอฟท์แวร์ทำงานอย่างถูกต้อง ถ้าฮาร์ดหรือซอฟท์แวร์ทำงานผิดพลาด ซึ่งจะมีผลทำให้สภาวะที่ขา WDI ไม่เปลี่ยนแปลงตามเวลาที่กำหนดไว้ MAX691 จะส่งสัญญาณรีเซทเป็นพลัส์บวกที่ขา RESET กว้าง 50 mS เพื่อรี เซทให้ CPU กลับไปทำงานใหม่อีกครั้ง และที่ขา RESET จะส่งพลัส์รีเซทออกมาทุก ๆ 1.6 วินาทีจนกว่าจะมีการเปลี่ยนแปลงสถานะที่ขา WDI อีกครั้ง



รูปที่ 2.16 จัมพ์เปอร์ Watchdog Timer

การใช้งาน Watchdog Timer สามารถเลือกใช้งานด้วยจัมพ์เปอร์ WD โดยปรับไปที่ EN. (ENABLE) เมื่อต้องการใช้งานหรือปรับไปที่ DIS. (disable) เมื่อไม่ต้องการใช้และในกรณีต้องการเปลี่ยนค่า Watchdog Timeout Period เป็นค่าอื่น สามารถกระทำได้โดยต่อสายจัมพ์เปอร์ที่ขา OSC SEL (ขา 8) ต่อลง ground และเพิ่มค่าคาปาซิลสเตอร์ cx ที่ขา 7 ตามตำแหน่งที่พิมพ์ไว้บนบอร์ด ANT-32 ซึ่งจะมีผลทำให้ความกว้างของพลัส์รีเซท (Reset Timout Period) เปลี่ยนไปตามค่า cx ดังไว้ในตารางที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OSC SEL ขา 8	OSC IN ขา	Watchdog Timeout Period		Reset Timeout Period
		Normal	After Reset	
Floating	Floating	1.6 Sec	1.6 Sec	50 ms
Low	Cx	(400/47pF)Cx	(1.6Sec/47pF)Cx	(200ms/47pF)Cx

ตารางที่ 2.4 การเลือกค่า Reset Pulse Width และ Watching Timeout

2.10 OPTINAL REALTIMECLOCK & CALANDER

สำหรับการใช้งานระบบไมโครฯ ที่มีเวลาเกี่ยวข้องด้วย จำเป็นต้องมีวงจรในส่วนที่ทำหน้าที่เป็น RTC (Real Time Clock) คือ นาฬิกาเวลาจริง ซึ่งบอร์ด ANT-32 ใช้ชิพ RTC เบอร์ DS1202 Serial Timekeeper Chip ของ DALLAS SEMICONTROL โดยต่อร่วมกับอุปกรณ์ภายนอกเพียงเล็กน้อยและที่สำคัญคือ ไม่ต้องทำการปรับแต่ง ซึ่งเมื่อจะใช้ RTC ต้องมีชิพ DS1202 และ MAX691 รวมทั้งคริสตอล 32.768 Khz และแบตเตอรี่ลิเธี่ยมด้วย

DS1202 (U7) ประกอบไปด้วย Real Time Clock/Calender และ Static RAM ขนาด 24 ไบท์ ทำการอินเตอร์เฟสกับ CPU ในแบบอนุกรม โดยใช้สายเพียง 3 เส้น คือ (1) ขา RST\(Reset), (2) ขา I/O(Data line) และ (3) ขา SCLK (Serial Clook) ขาสัญญาณทั้ง 3 นี้จะต่อเข้ากับขา P1.6, P1.4 และ P1.5 ของ CPU ตามลำดับเมื่อต้องการทราบค่าเวลา CPU ต้องทำการอ่านเวลาจาก RTC เพราะว่า DS1202 ไม่มีขาสำหรับไปอินเตอร์รัพท์ CPU CPU สามารถเขียนหรืออ่านข้อมูลของ CLOCK หรือ RAM ได้ 2 วิธี คือ Singie-byte และ Multiple-byte โดยทั้ง 2 วิธี CPU ต้องส่ง Command byte (8 บิท) ให้ DS1202 เพื่อบอกให้ DS1202 ทราบว่าจะทำการเขียนหรืออ่าน Clock หรือ ram พร้อมตำแหน่งและตามด้วยข้อมูลในขณะที่กำลังติดต่อกับ DS1202 สัญญาณที่ขา RST\ต้องเป็นลอจิก “1” ขา SCLK จะเป็นสัญญาณ Serial Clock เพื่อทำการเขียนหรืออ่านข้อมูล โดยจะใช้สัญญาณ clock 1 ลูก สำหรับข้อมูล 1 บิท ส่วนขา I/O เป็นข้อมูลอนุกรม โดยจะเป็นอินพุตเมื่อทำการเขียน และเป็นเอาท์พุตเมื่อทำการอ่าน โดยข้อมูลที่จะทำการเขียนหรืออ่านนี้จะเริ่มจากบิท 0 และจบด้วยบิท 7 ค่าของ command byte ในการเขียนและอ่าน clock และ ram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 แนวทางในการพัฒนาโปรแกรม

การใช้งานบอร์ด ANT-32 ผู้ใช้จำเป็นต้องเขียนโปรแกรมควบคุมที่เรียกกันว่า มอนิเตอร์โปรแกรมขึ้นมาโดยเฉพาะเพื่อทำให้งานที่ต้องการพัฒนาสำเร็จได้ในขั้นตอนการพัฒนานี้เอง ที่เป็นจุดเด่นของ ANT-32 โดยมีโปรแกรมให้เลือก 2 ลักษณะด้วยกันคือ REM31 และเบสิก 32 หลักการของทั้งสองโปรแกรมก็คือให้ผู้ใช้นำ อีพรอม ที่บรรจุโปรแกรมนี้ลงบนบอร์ด ANT-32 ที่ตำแหน่งหน่วยความจำ U2 (อีพรอม) แล้วทำการต่อสาย Serial port ระหว่างบอร์ด ANT-32 กับเครื่องไมโครคอมพิวเตอร์จากนั้นที่เครื่องพีซี ให้ใช้โปรแกรมสำหรับการสื่อสารข้อมูลอนุกรม ผู้ใช้จะสามารถติดต่อกับบอร์ด ANT-32 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบการสร้างและการทำงาน

3.1 บทนำ

จากรายละเอียดที่ได้ศึกษาเกี่ยวกับสัญญาณโทนต่าง ๆ ของระบบชุมสายโทรศัพท์ที่ส่งมายังสมาชิกผู้ใช้บริการสามารถสรุปแนวความคิดในการตรวจสอบการใช้งานโทรศัพท์

แนวความคิดการตรวจสอบการตอบรับสายของผู้รับโดยใช้สัญญาณเรียกกลับเป็นสัญญาณตรวจการรับสาย เพราะระบบการทำงานของเครื่องชุมสายระบบอิเล็คทรดนิกส์ หรือ SPC (STORED PROGRAM CONTROL) ในปัจจุบันเมื่อผู้เรียกและผู้รับสามารถเรียกติดต่อได้สำเร็จขั้วของคู่สายโทรศัพท์จะไม่มีการสลับขั้ว กรณีที่มีการสลับขั้วของคู่สายก็มีเฉพาะชุมสายในระบบครอสบาร์เท่านั้น แนวความคิดของการใช้สัญญาณเรียกกลับนี้สามารถใช้ตรวจสอบกับระบบชุมสายโทรศัพท์ทั่วไป และแก้ปัญหาเกี่ยวกับความไม่แน่นอนของการสลับขั้วจากชุมสายโทรศัพท์ รายละเอียดของบทนี้กล่าวถึงขั้นตอนการตรวจสอบการใช้งานโทรศัพท์และแนวความคิดที่ใช้ ในการออกแบบ และสร้างวงจรตรวจสอบ

3.2. ขั้นตอนการตรวจสอบการใช้โทรศัพท์

1 เริ่มต้นเมื่อมีการวางหูโทรศัพท์ ส่วนของซอฟแวร์จะสั่งให้ซีพียูทำการเคลียร์ข้อมูลของ Port I/Oต่าง ๆ

2 วงจรตรวจสอบการยกหูโทรศัพท์จะตรวจสอบว่ามีผู้เรียกทำการยกหูโทรศัพท์เพื่อเรียกติดต่อหรือไม่

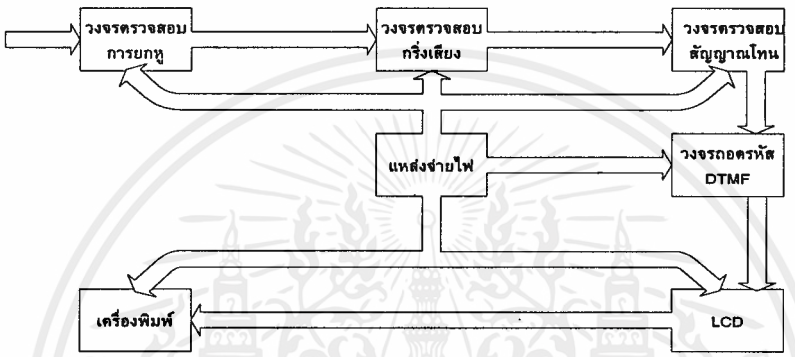
3. ถ้ามีการยกหูโทรศัพท์วงจรสอบสัญญาณกริ่งเรียกจะตรวจสอบว่ามีสัญญาณกริ่งเรียกเข้ามาหรือไม่

4. ถ้ามีสัญญาณกริ่งเรียกเข้ามาแสดงว่าเป็นการยกหูโทรศัพท์เพื่อตอบรับการเรียก กรณีนี้จะไม่มีผลต่อการทำงานในส่วนของเครื่องคิดเงินค่าโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ถ้าไม่มีสัญญาณกริ่งเรียกเข้ามา แสดงว่าเป็นการยกหูโทรศัพท์ทำการติดต่อออกวงจรตรวจสอบสัญญาณให้หมุนจะตรวจสอบว่า ผู้เรียกได้รับสัญญาณให้หมุนเพื่อเริ่มติดต่อได้หรือไม่

3.3 ผังการทำงานของเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ



รูปที่ 3.1 ผังการทำงานของเครื่องคำนวณค่าบริการการใช้โทรศัพท์

3.4 การออกแบบการสร้างวงจร

จากขั้นตอนการทำงานของระบบพบว่า มีการตรวจสอบเป็นขั้นตอนตามลำดับและต้องตรวจสอบแยกให้ถูกต้อง เป็นการใช้งานโทรศัพท์ในฐานะผู้เรียกหรือผู้รับ และนำผลที่ตรวจสอบได้มาสรุปรวมทั้งมีส่วนของการเก็บข้อมูลเพื่อสามารถให้ทำงานได้อย่างถูกต้อง

เครื่องคิดเงินค่าใช้โทรศัพท์ ประกอบด้วยส่วนใหญ่ ๆ 2 ส่วน คือ

1 .ส่วนของวงจรตรวจสอบ (Hardware)

2 .ส่วนของโปรแกรมควบคุมการทำงาน (Software)

ส่วนของวงจรตรวตสอบการใช้งานโทรศัพท์ จะตรวจสอบสัญญาณต่าง ๆ จากคู่สายโทรศัพท์และอาศัยส่วนของโปรแกรมควบคุมการทำงาน ทำให้สามารถทำการตัดสินใจและคำนวณจัดการข้อมูล เป็นการเพิ่มประสิทธิภาพและให้ความถูกต้องในการทำงาน

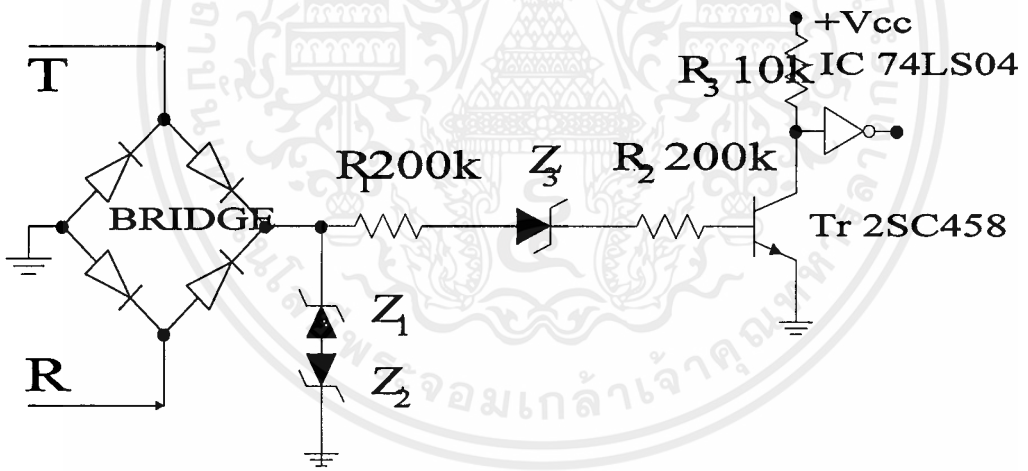
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 ส่วนของวงจรตรวจสอบ

แบ่งออกเป็นส่วน ๆ ตามหน้าที่การทำงาน

- วงจรการตรวจสอบการยกหูโทรศัพท์ (CHECK HANDSET)
- วงจรตรวจสอบสัญญาณกริ่งเรียก (CHECK RINGING TONE)
- วงจรตรวจสอบสัญญาโทน (TONE DECODER)
- วงจรถอดรหัสความถี่โทรศัพท์ชนิดกดปุ่ม (DTMF)
- วงจรนาฬิกา (TIMER)
- วงจรอินเทอร์เฟส (INTERFACE)
- ภาคจ่ายพลังงาน (POWER SUPPLY)

3.4.2 วงจรตรวจสอบการยกหูโทรศัพท์



รูปที่ 3.2 วงจรตรวจสอบการยกหูโทรศัพท์

หลักการทำงานของวงจร คือ เมื่อมีการยกหูโทรศัพท์จะทำให้ระดับสัญญาณไฟตรงของคู่สายโทรศัพท์เปลี่ยนจาก 48 โวลท์ เป็น 10 โวลท์ อาศัยความแตกต่างของสัญญาณในการตรวจสอบ จากรูป 3.4 คู่สายโทรศัพท์จะผ่านบริดจ์ไดโอด เพื่อไม่ต้องคำนึงถึงขั้วบวก และขั้วลบของคู่สายโทร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศัพท์ โดยใช้ความต้านทานและซีเนอร์ไดโอด เป็นวงจรแบ่งระดับแรงดัน ไปไบแอสทรานซิสเตอร์ ซึ่งทำงานในลักษณะสวิทซ์ปิดเปิด

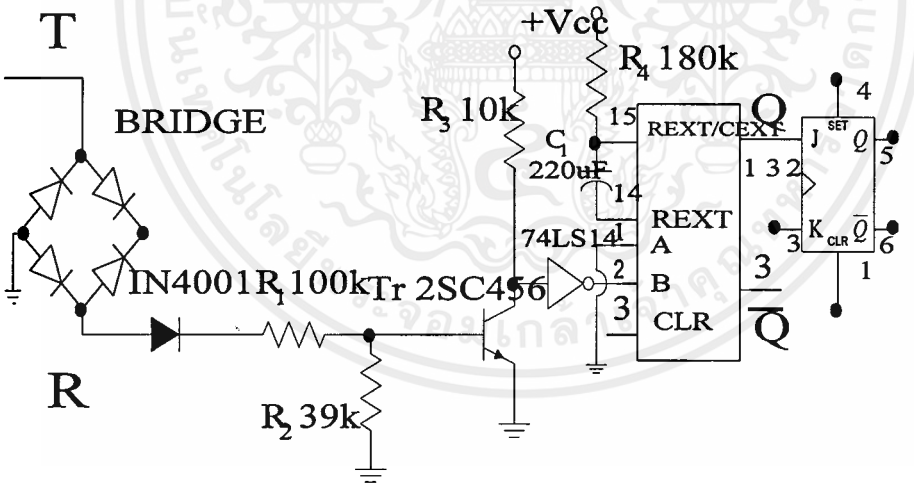
กรณีที่เป็นการวางหูระดับแรงดัน 48 โวลท์ ถูกแบ่งไปตกคร่อมความต้านทาน ทำให้ทรานซิสเตอร์ถูก ไบแอสโดยตรง ให้นำกระแส ทำให้สัญญาณเอาท์พุทเป็น ลอจิก 1

กรณีเป็นการยกหูระดับแรงดัน 10 โวลท์ จะตกคร่อมที่ซีเนอร์ไดโอดทั้งหมด ทำให้ไม่มีแรงดันไปไบแอสทรานซีสเตอร์ให้นำกระแส ทำให้ได้สัญญาณเอาท์พุทเป็น ลอจิก 0

ส่วนของซีเนอร์ไดโอด 66 โวลต์ มีเพื่อป้องกันความเสียหายแก่วงจรที่เกิดจากแรงดันไฟฟ้าเนื่องจากสัญญาณกริ่งเรียก

สัญญาณลอจิก 1 และ 0 ใช้เป็นสัญญาณให้ CPU มาอ่านในกรณีที่เป็นลอจิก 1 หรือเป็น การวางหู CPU จะทำการรีเซ็ตวงจรนับพัลส์หมายเลข ในกรณีที่เป็นลอจิก 0 หรือ เป็นการยกหูจะใช้เป็นสัญญาณคล็อคให้ไอซี 7404 (D FLIP FLOP) เพื่อเนข้อมูลว่ามีสัญญาณกริ่งเรียกเข้ามาหรือไม่

3.4.3 วงจรตรวจสัญญาณกริ่งเรียก



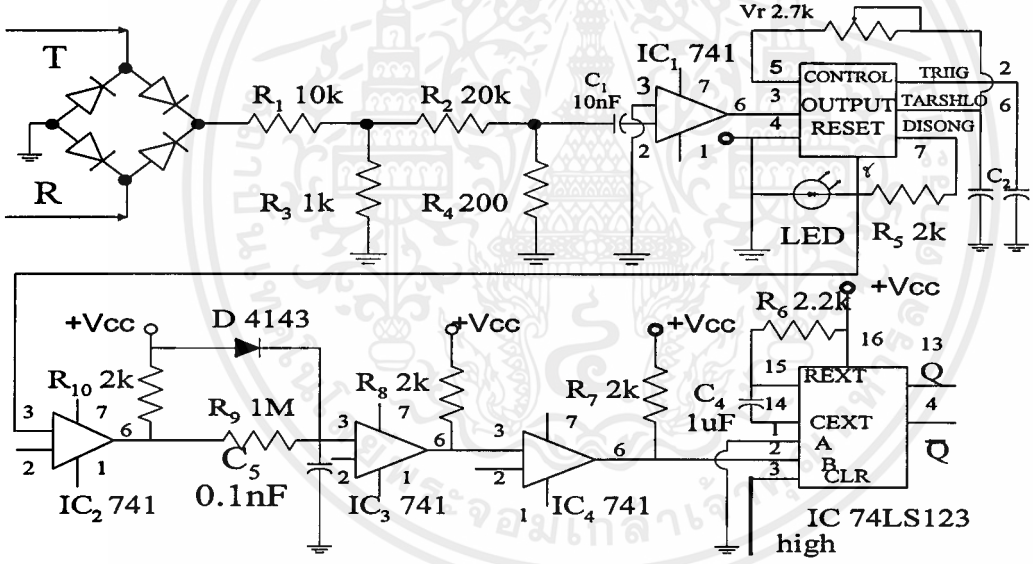
รูปที่ 3.3 วงจรตรวจสัญญาณกริ่งเรียก

จากรูปที่ 3.3 อินพุทของวงจรต่อเข้ากับขั้วลบของคู่สายโทรศัพท์ เมื่อไม่มีสัญญาณกริ่งเรียกเข้ามา ไดโอดจะถูกไบแอสกลับทิศทางทำให้ไม่นำกระแสเมื่อมีสัญญาณกริ่งเรียก เข้ามาซึ่งเป็นสัญญาณ A.C. 100 Vp-p ทำให้ขั้วของคู่สาย มีการสลับขั้วไปตามจังหวะของสัญญาณทำให้ไดโอดนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระแสและไม่นำกระแสสลับกันสัญญาณที่ขาคอลเลคเตอร์จะเป็นลอจิก 1 และ 0 สลับกัน ซึ่งใช้เป็นสัญญาณคล็อคให้ ไอซี 74123 (monostable retrigalbe) เนื่องจากสัญญาณกริ่งเรียกมีช่วงดัง 1 วินาที และเงียบช่วง 4 วินาที จึงกำหนดให้ความกว้างของพลัส์เท่ากับ 6 วินาที จากสูตร T(W) = 0.29 RC โดยกำหนดค่า C = 100 ไมโครฟารัด R = 200 กิโลโอหม์ เมื่อมีสัญญาณกริ่งเรียกเข้ามาทำให้เอาท์พุทของ ไอซี 74123 เป็นลอจิก 1 เพื่อใช้เป็นสัญญาณอินพุทของไอซี 7474 เพื่อค้างสถานะว่ามีสัญญาณกริ่งเรียกเข้ามาให้ CPU อ่านเพื่อให้ทราบว่าเป็นการตอบรับการเรียก เมื่อมีการวางหู CPU จะทำการเคลียร์ข้อมูลสัญญาณกริ่งเรียกเดิม

3.4.4 วงจรตรวจแยกสัญญาณโทน



รูปที่ 3.4 วงจรตรวจแยกสัญญาณโทน

เนื่องจากสัญญาณในคู่สายโทรศัพท์ มีสัญญาณความถี่อื่น ๆ เกิดเป็นสัญญาณรบกวนและมีระดับขนาดสัญญาณยังไม่เหมาะสมกับการทำงานของวงจร เพื่อแก้ปัญหาดังกล่าวเราจึงเพิ่มวงจรกรองความถี่ เพื่อกำจัดสัญญาณรบกวนที่ไม่ต้องการ ให้ได้สัญญาณโทนที่สะอาดและมีระดับขนาดที่เหมาะสม ส่วนของวงจรกรองความถี่ที่ใช้เป็นวงจรกรองความถี่แบบแถบผ่าน (BANDPASS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILTER)โดยใช้ OPAMP 741 ประกอบร่วมกับ R,C กำหนดคุณสมบัติที่ต้องการและให้ขยายสัญญาณโทนให้ได้สัญญาณเอาท์พุทระดับขนาด 200 mVp-p และมีแบนด์วิดท์ของวงจรกว้างพอที่สามารถจะตรวจสอบได้ถูกต้อง ซึ่งจากข้อกำหนดเราเลือกใช้ วงจรกรองความถี่แบบ SECOND ORER MULTIPLE FEEDBACK

3.4.5 วงจรถอดรหัสความถี่โทรศัพท์ชนิดกดปุ่ม (DTMF)

MT8870 ไอซีถอดรหัสความถี่โทรศัพท์ (INTERGRATE DTMF RECIEVER) เมื่อเอ่ยถึงไอซีทางโทรศัพท์ ช่างหรือวิศวกรส่วนใหญ่มักจะส่ายหน้าสาเหตุเนื่องมาจากบ้านเราไม่ค่อยมีผู้นำเข้ามาจำหน่าย หายาก คู่มือไม่มี ขาดการเผยแพร่ ทั้ง ๆ ที่ในปัจจุบันโทรศัพท์มีส่วนสำคัญกับชีวิตประจำวันของเราอย่างมากมาย และนับวันก็ยิ่งทวีความสำคัญเพิ่มมากขึ้นทุกทีและอุตสาหกรรมทางด้านโทรศัพท์ และการสื่อสารก็ขยายตัวขึ้นอย่างรวดเร็ว ดังนั้น ช่างหรือวิศวกรอย่างพวกเราน่าจะศึกษาเอาไว้ เพื่อจะได้นำมาใช้งานได้ในโอกาสต่อไป ความถี่ซึ่งเกิดจากการกดปุ่มตัวเลขของโทรศัพท์ชนิดกดปุ่ม (ชนิด TONE หรือ DTMF) ให้เป็นระบบตัวเลขทางดิจิตอล ซึ่ง ไอซี MT8870 ใช้แปลงความถี่โทรศัพท์ให้เป็นเลขฐานสองขนาด 4 บิท ในยุคก่อนการออกแบบวงจรถอดรหัสความถี่โทรศัพท์ มักใช้ไอซีจำพวกเฟสล็อกลูบซึ่งสร้างปัญหาสารพัดไม่ว่าจะเป็นเรื่องของความถี่ที่เปลี่ยนแปลงไป การปรับแต่งวงจร ขนาดของวงจรที่ใหญ่ เพราะใช้ไอซีจำนวนมาก

3.4.6 วงจรแหล่งจ่ายไฟ (POWER SUPPLY)

แหล่งจ่ายไฟลบจากแหล่งจ่ายไฟบวก

ในบางครั้งมีความจำเป็นต้องใช้แรงดังไฟที่เป็นลบ ขณะที่แหล่งจ่ายไฟที่มีทั้งหมดจ่ายแต่ไฟบวก เช่นต้องการไฟลบจ่ายให้แก่ออปแอมป์ในขณะที่วงจรอื่นเป็นวงจรลอจิกซึ่งต้องการแรงดันบวก โดยเฉพาะอย่างยิ่งในกรณีที่วงจรทั้งหมดใช้แหล่งจ่ายไฟจากแบตเตอรี่ซึ่งทำให้เกิดปัญหาในการออกแบบขึ้นได้มาก ตามวงจรรูปที่ 2 จะนำกระแสและหยุดนำกระแสด้วยสัญญาณอินพุตที่เป็นคลื่นรูปเหลี่ยมซึ่งมีดิวตี้ไซเคิลเป็น 50 และมีความถี่ประมาณ 10 Khz สัญญาณลักษณะนี้มักหาได้ง่ายในวงจรลอจิกอยู่แล้ว เช่น สัญญาณนาฬิกา แต่ถ้าไม่มีก็อาจสร้างขึ้นจากวงจรกำเนิดคลื่นที่ใช้เกต NAND ได้

เมื่อ T1 เลิกนำกระแส ขั้วบวกของ C1 จะมีแรงดันลดเหลือ 0.8V โดยประมาณ โดยคายประจุผ่าน D1 ขั้วลบของ C1 จะคายประจุผ่าน D3 ไปยัง C2 และ C2 จะถูกประจุ ถ้าไม่มีการดึงกระแส แรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดันที่ C2 จะมีค่าประมาณ -10 V และจะลดลงตามรูปที่ 1 เมื่อมีการดึงกระแสทางเอาท์พุต เมื่อถึงจุดหนึ่งสัญญาณ 10 Khz จากอินพุตจะมาปรากฏด้วยวงจรเครื่องจ่ายไฟตรง 5 Volt ที่จ่ายกระแสได้ประมาณ 5A

เมื่อเริ่มเปิดสวิตซ์ ภาคเรกติฟายเออร์ชนิดเต็มคลื่น (D1 ถึง D4) จะจ่ายไฟตรงที่ยังไม่เรียบดีนักให้แก่ไอซีเรกูเลเตอร์ (IC1) ซึ่งทำหน้าที่ควบคุมให้ไฟที่เอาท์พุตของเครื่องออกมาคงที่ 5 V D5 ใส่เข้าไปเพื่อช่วยยกแรงดันที่เอาท์พุทของตัว IC ขึ้น เพื่อชดเชยกับแรงดันที่ไปตกคร่อมในตัว D7

เมื่อเริ่มต้นแรงดันที่ขาคอลเลคเตอร์ของ Q1 จะถูกรักษาให้มีแรงดันต่ำโดย C2 และ R6 ขณะที่ C1 และ C4 กำลังประจุให้แรงดันคร่อมตัวมันมีค่าสูงขึ้น การที่แรงดัน คอลเลคเตอร์มีค่าต่ำในตอนเริ่มต้นนี้ทำให้ SCR1 ไม่ทำงานเพราะไม่มีกระแสมากระตุ้นที่ขาเกท แบตเตอรี่แบบนิเกิลแคดเมี่ยมจะถูกประจุน้อย ๆ (trickle-charged) โดยผ่านทาง R1 และ LED 1 LED 1 จะสว่างแสดงว่าแบตเตอรี่กำลังถูกประจุไฟอยู่และ LED1 จะทำหน้าที่คอยกันไม่ให้แบตเตอรี่ถูกใช้งานไปทาง IC1 ขณะที่เครื่องถูกปิดแล้ว

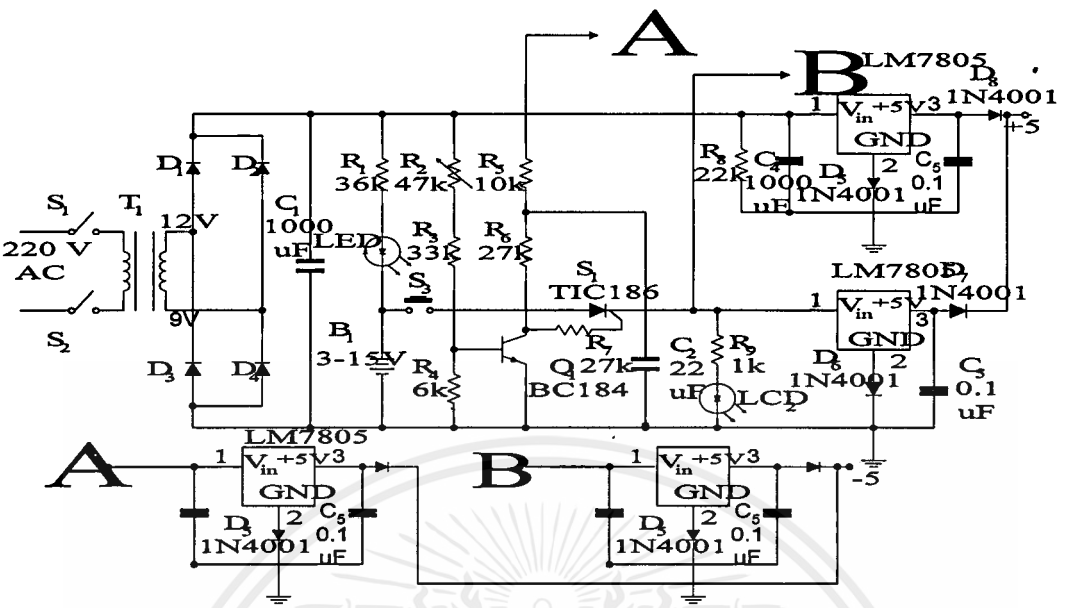
เมื่อไฟบ้านเกิดขาดตอนขึ้นมาไม่ว่าจะด้วยเหตุผลใดก็ตามแรงดันที่ป้อนให้แก่ IC1 ยังคงพอเพื่อที่จะทำให้แรงดันเอาท์พุตเป็น 5V ได้อีกชั่วระยะเวลาหนึ่งแต่เมื่อแรงดันนี้ตกต่ำถึงระดับหนึ่งที่ทำให้แรงดันที่ขาเบสไม่เพียงพอที่จะทำให้ Q1 นำกระแสอยู่ได้ C2 จะคายประจุมากระตุ้นขาเกตของ SCR1 โดยผ่านทาง R6 และ R7 เมื่อ SCR1 นำกระแสก็จะเป็นทางผ่านให้แบตเตอรี่จ่ายไฟมาป้อนแก่ไอซีเรกูเลเตอร์ตัวที่สอง (IC2) ซึ่ง IC ตัวนี้จะเข้ามาทำหน้าที่จ่ายไฟเลี้ยงแทน IC1 ก่อนที่แรงดันเอาท์พุตจะตกต่ำกว่า 5V ในภาวะนี้ LED1จะดับ และ LED2 จะสว่างเป็นการแสดงว่าแบตเตอรี่กำลังถูกใช้งานอยู่

ที่นี้ถ้าแรงดันไฟบ้านกลับคืนมาอีก LED1 ก็สว่างขึ้นอีกครั้งหนึ่งเพื่อแสดงว่ามีไฟบ้านมาแล้ว แต่ LED2 ยังคงสว่างอยู่จนกว่าสวิตซ์ S2 จะถูกกดเพื่อรีเซตให้ SCR1 หยุดนำกระแส

เมื่อต้องการปิดเครื่องจ่ายไฟให้โยกสวิตซ์ S1 เพื่อตัดไฟบ้านออกก่อน แล้วจึงค่อยกดสวิตซ์ S2 เพื่อรีเซทให้ SCR1 หยุดทำงานในสภาวะนี้จะไม่มีการดึงกระแสออกจาแบตเตอรี่เลย

ถ้าต้องการไฟเลี้ยงจ่ายกระแสได้น้อยกว่า 1 A ก็อาจเปลี่ยนเบอร์ IC จาก 78H05 มาเป็น 7805 และใช้ D7, D8 เป็นขนาด 1A ส่วนหม้อแปลง D1 ถึง D4 ก็ต้องเลือกใช้ให้มีขนาดสอดคล้องกับระดับกระแสที่ต้องการ ในตอนเริ่มแรกให้ตั้ง R2 ไว้ที่ ตำแหน่งกึ่งกลางแล้วค่อยปรับแต่งให้ได้สภาวะการทำงานที่ดีที่สุด (หมายถึงให้แบตเตอรี่จ่ายไฟมาแทนไฟบ้านได้ ทันเวลาโดยที่เอาท์พุทไม่ตกต่ำกว่า 5V) ในกรณีที่แบตเตอรี่มีแรงดันมากกว่า 9.6 V จะต้องเพิ่มค่า R8 และลดค่า R1 ลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 วงจรจ่ายไฟ 5 V. ชนิดมีแบตเตอรี่สำรอง

3.5 ขั้นตอนการออกแบบวงจรกรองสัญญาณชนิดแถบความถี่ผ่าน

ในวงจรลักษณะนี้จะต้องออกแบบตามความต้องการ 2 ประการ คือ

1. ช่วงความถี่กลางของแถบความถี่ที่จะนำไปใช้งานได้
2. ค่า Q ของวงจร

ขั้นตอนที่ 1 หาค่าของ R_3 และ R_4

$$R_3 = R_4 = \frac{10^6}{f_0} =$$

ค่าที่คำนวณได้นี้อาจจะเปลี่ยนแปลงได้ โดยอาจจะใช้เพียงครึ่งเดียวหรือเพิ่มเป็น 2 เท่าก็ได้ ให้ R3 = R4 เสมอ ถ้า A1 เป็นออปแอมป์ชิดโมโนลิธิคไบโพลาร์ ค่า R3 = R4 จะต้องไม่เกิน 1 M ถ้า A1 มีค่ากระแสไบแอส (Ib) ต่ำ R3 และ R4 จะมีค่าเกินกว่า 10 M

ขั้นตอนที่ 2 เลือกค่า C_1 และ C_2

$$C_1 = C_2 = \frac{10^{-7}}{f_o}$$

สำหรับค่าที่ใช้จริงอาจจะมีเพียงครึ่งเดียว หรือ 2 เท่าของค่าที่คำนวณได้ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 3 คำนวณค่า R_5 และ R_6

$$R_5 = R_6 = \frac{1}{2 \sim f_0 C_1}$$

ถ้าค่านี้สูงเกินไป (มากกว่า 1 M) จะมีผลทำให้เกิดปัญหาเกี่ยวกับ offset ขึ้นได้ (หรืออาจจะสูงถึง 10 ถ้า A2 และ A3 มีกระแสไบแอสต่ำ) ดังนั้นถ้าพบปัญหาเช่นนี้จะต้องหาค่า C1 และ C2 ที่เหมาะสมใหม่ แล้วคำนวณค่า R5 และ R6 อีกครั้งเช่นกันแต่ถ้าในกรณีที่ค่า C1 และ C2 สูงมาก เราสามารถแทน R5 และ R6 ได้ด้วยเน็ทเวิร์ด T ดังรูปข้างต้น และหาค่า R_x ได้จาก

$$R_X = \frac{10^{10}}{R_5 \times 2*10^5}$$

ส่วนค่าออฟเซ็ทที่เอาท์พุทของ A2 สามารถทำให้ลดลงได้โดยต่อขาอินพุทบวกของ A2 ลงกราวด์ โดยผ่านค่าความต้านทานที่มีค่าเท่ากับ R5 ในลักษณะเดียวกัน ออฟเซ็ทที่เกิดขึ้นที่เอาท์พุทของ R3 ก็สามารถลดผลลงได้โดยต่อขาอินพุทบวกของ A3 ลงกราวด์โดยผ่าน R6

ขั้นตอนที่ 4 ให้ $R_1 = R_3$ และหาค่า R_2

$$R_2 = R_1 \times (2Q - 1)$$

ขั้นตอนที่ 5 คำนวณค่า H (ที่ความถี่รีโซแนนท์) จากสมการที่ 3

ขั้นตอนที่ 6 คำนวณค่าเซนซิติวิตี้ของ f_0 ต่อ การเปลี่ยนแปลงของพารามิเตอร์จากสมการที่ 5

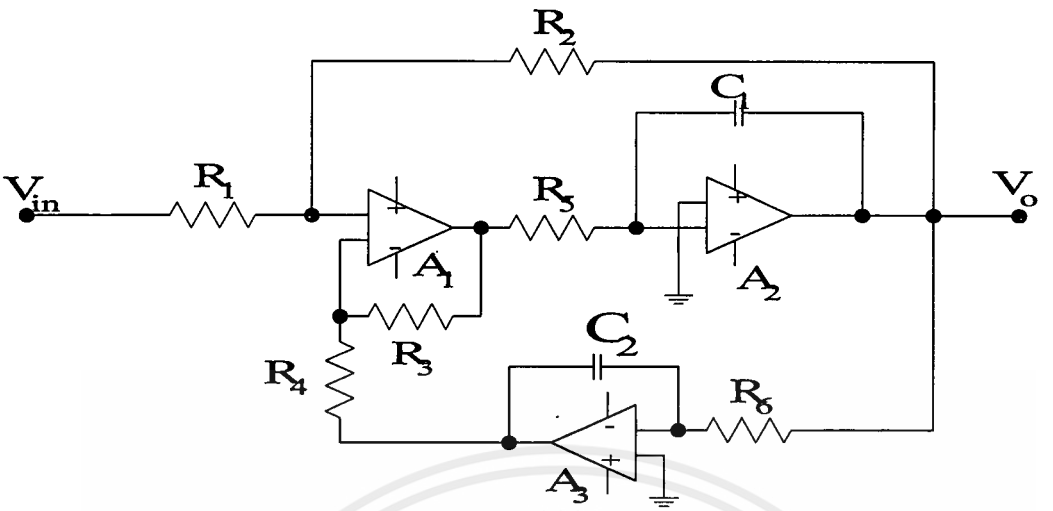
ขั้นตอนที่ 7 คำนวณค่าเซนซิติวิตี้ของ Q ต่อ การเปลี่ยนแปลงของพารามิเตอร์ โดยใช้สมการที่ 6

3.5.1 การทำงานของวงจร

วงจรดังรูปแสดงถึงการออกแบบวงจรฟิลเตอร์ที่ใช้ออปแอมป์ถึง 3 ตัว ซึ่งมีข้อดีเหนือวงจรฟิลเตอร์ที่ใช้ออปแอมป์เพียงตัวเดียวหลายประการ คือ

1. ถ้าใช้อุปกรณ์ที่มีค่าเหมาะสม ค่าความถี่กลาง f_0 จะไม่ขึ้นกับค่า Q ของวงจร
2. ค่าเซนซิติวิตี้ของ f_0 และ Q ต่อการเปลี่ยนแปลงของพารามิเตอร์อื่น ๆ มีค่าต่ำมาก
3. สามารถให้ค่า Q ของวงจรที่มีค่าสูง ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แบนด์พาสฟิลเตอร์แบบเปลี่ยนแปลงสถานะโดยมีการป้อนกลับแบบลบ

3.6 โปรแกรมควบคุมการทำงานของระบบ

การควบคุมการทำงานของเครื่องคิดค่าโทรศัพท์ด้วยเครื่องไมโครคอนโทรลเลอร์จะมีความสามารถและประสิทธิภาพได้ดีนั้นขึ้นอยู่กับส่วนของโปรแกรมควบคุมการทำงาน เราใช้ภาษาระดับกลางที่เขียนด้วย "ภาษาซี" เนื่องจากมีความสามารถในการเข้าถึงระบบฮาร์ดแวร์และสามารถที่จะใช้ในการประมวลผลข้อมูลได้ดีอีกด้วย ส่วนของโปรแกรมควบคุมทำงานแบ่งได้เป็น 2 ส่วน คือ

1.โปรแกรมตรวจสอบสถานะการใช้งานโทรศัพท์ เป็นส่วนตรวจสอบ โดยการอ่านข้อมูลจากพอรท์อินพุทเอาท์พุทต่างๆ เพื่อให้ทราบสถานะการใช้โทรศัพท์และจะแสดงผลสถานะต่าง ๆ ของการใช้โทรศัพท์ออกจอภาพ และจัดการเคลียร์ข้อมูลของระบบเพื่อให้สามารถเริ่มต้นการทำงานได้ถูกต้อง

2.โปรแกรมประมวลผลและคำนวณค่าบริการโทรศัพท์ โปรแกรมส่วนนี้จะประกอบด้วยข้อมูลเกี่ยวกับรหัสเลขหมายทางไกลของจังหวัดต่างๆ และอัตราค่าบริการต่อนาทีเพื่อนำมาใช้ในการคำนวณเงิน

ในส่วนของการคิดเงินจะมีตัวแปรเข้ามาเกี่ยวข้องในการที่คิดค่าบริการ ก็คือช่วงเวลาที่ใช้ในการโทรศัพท์จะแบ่งเป็นช่วงเวลาในการคิดตามช่วงเวลาที่ องค์การโทรศัพท์เป็นผู้กำหนด

ถ้าเริ่มใช้บริการตั้งแต่เวลา 7.00 น. - 18.00 น. จะใช้ค่าตัวแปรเป็น 1

ถ้าเริ่มใช้บริการตั้งแต่เวลา 18.01 น. - 21.59 น. จะใช้ค่าตัวแปรเป็น 0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเริ่มใช้บริการตั้งแต่เวลา 22.00 น. - 06.59 น. จะใช้ค่าตัวแปรเป็น 0.33

ซึ่งอัตราค่าบริการจะคิดจากผลคูณของตัวแปรทั้งสามนี้ คือ

ค่าบริการ = CODE RATE * TIME USE * TIME RATE

เมื่อจบการสนทนาแต่ละครั้งโปรแกรมจะแสดง หมายเลขปลายทางที่ติดต่อ เวลาที่ใช้ในการสนทนา และค่าบริการในแต่ละครั้ง เมื่อจบการทำงานนี้แล้วโปรแกรมจะกลับไปทำงานที่จุดเริ่มต้นใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

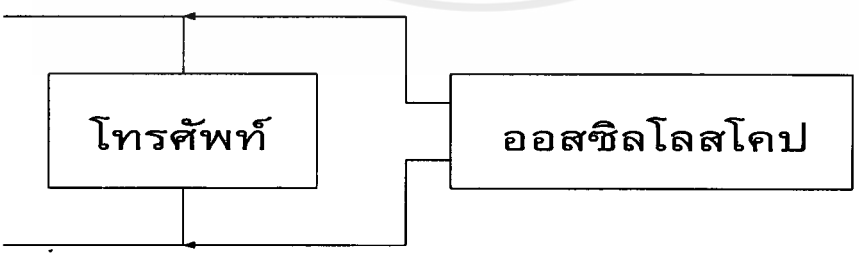
4.1 บทนำ

วงจรในส่วนต่างๆ ก่อนที่จะนำมาประกอบเข้าด้วยกันต้องมีการตรวจสอบการทำงานเสียก่อนเมื่อไม่แน่ใจว่าสามารถทำงานร่วมกันได้ จึงค่อยนำมาเชื่อมต่อเข้าด้วยกัน แล้วจึงเขียนโปรแกรมควบคุมทั้งหมด

ปัญหาที่สำคัญก็คือส่วนที่ต่อเข้ากับคู่สายโทรศัพท์ โดยเฉพาะกรณีสัญญาณภายในสายมีการรบกวนสูงและสัญญาณอ่อน จะมีผลต่อวงจรตรวจจับ RING BACK TONE วิธีทางแก้ไขสามารถทำได้ โดยการเพิ่มวงจรกรองความถี่ และพยายามลดอิมพีแดนซ์ของวงจรลง

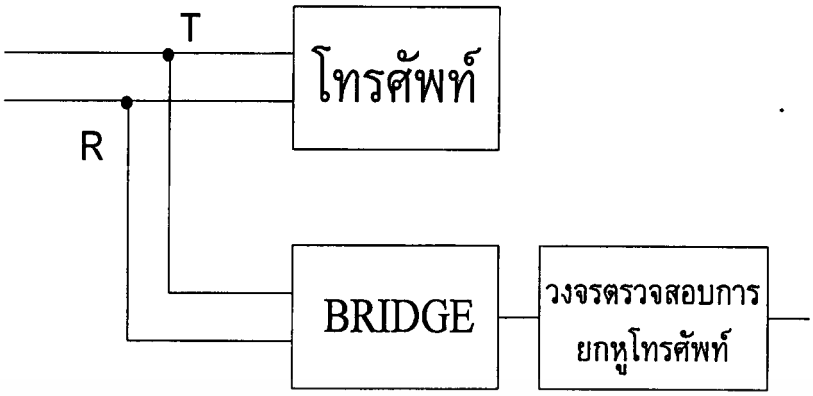
4.2 ผลการทดลอง

4.2.1 วงจรตรวจสอบการยกหูโทรศัพท์

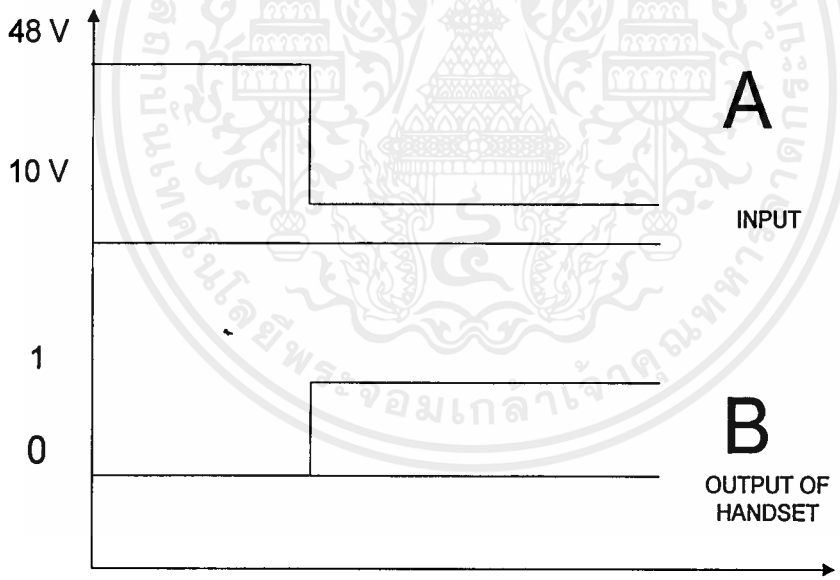


รูปที่ 4.1 ตำแหน่งของจุดวัดสัญญาณโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ผังการทำงานวงจรการตรวจสอบการยกหู



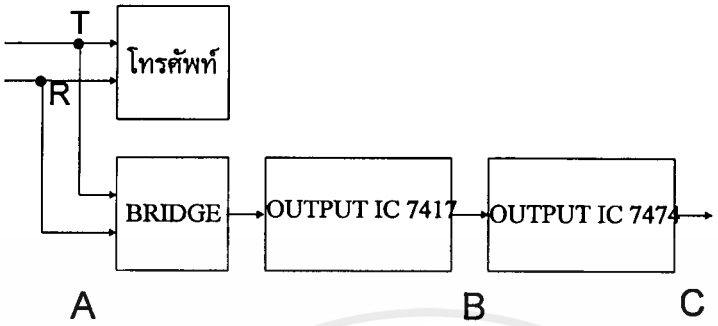
รูปที่ 4.3 สัญญาณเอาท์พุตวงจรตรวจสอบการยกหูโทรศัพท์

รูป A - ขณะวางหูวัดได้ 48 โวลท์

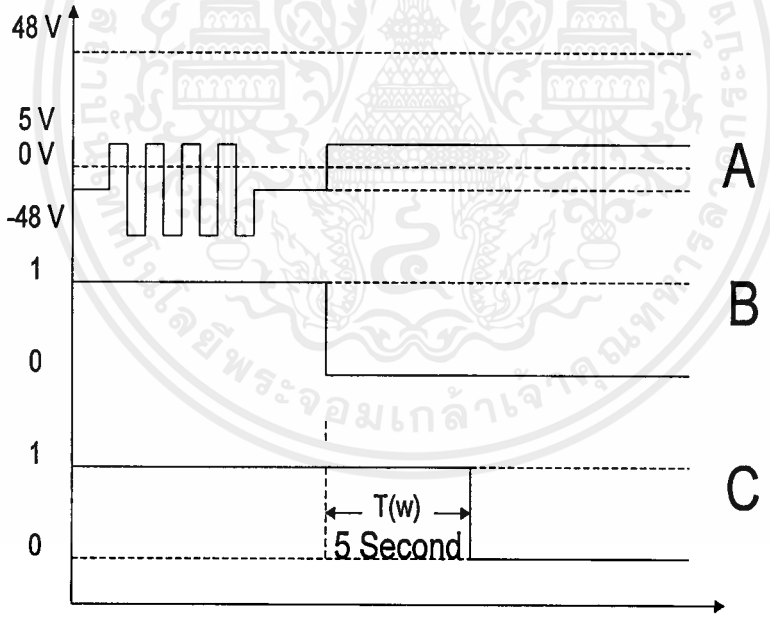
รูป B - ขณะยกหูวัดได้ 10 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 วงจรตรวจสอบสัญญาณกริ่งเรียก



รูปที่ 4.4 การวัดสัญญาณเอาท์พุทตามจุดต่าง ๆ



รูปที่ 4.5 สัญญาณเอาท์พุตวงจรตรวจสอบสัญญาณกริ่งโทรศัพท์

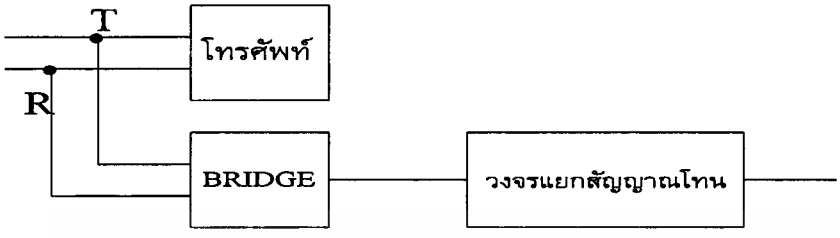
จุด A - สัญญาณในคู่สายโทรศัพท์เมื่อมีกริ่งเรียก

จุด B - สัญญาณเวลา ไอซี 7414

จุด C - สัญญาณข้อมูลว่ามีสัญญาณกริ่งเรียกของ ไอ ซี 7414

• เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 วงจรตรวจแยกสัญญาณโทรศัพท์



รูปที่ 4.6 ผังการทำงานของวงจรแยกสัญญาณโทรศัพท์

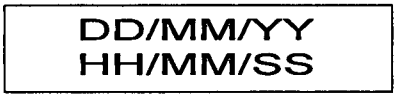
A ขณะรับสัญญาณ Dial tone จะได้สัญญาณเอาท์พุตดังรูป A

B ขณะรับสัญญาณ Busy tone จะได้สัญญาณเอาท์พุตดังรูป B

C ขณะรับสัญญาณ Ring-back toneจะได้สัญญาณเอาท์พุตดังรูป C

วิธีการทดลองการใช้เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ

1. เปิดสวิทซ์ POWER DISPLAY จะแสดงอักษรดังรูป



2. กดปุ่ม FUNCTION จะเป็นการตั้ง วัน เดือน ปี ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กดปุ่ม SET จะแสดง CURSOR ในหลักแรก จากนั้นกดปุ่ม UP / DOWN เพื่อตั้งวันที่ แล้วกดปุ่ม SET อีกครั้งเพื่อการตั้งเดือนและปีเช่นเดียวกัน

เมื่อตั้งวัน เดือน ปี เสร็จแล้ว CURSOR จะหายไปแสดงว่า การตั้งวัน เดือน ปี ได้เสร็จสิ้นแล้ว

3. กดปุ่ม FUNCTION จะเป็นการตั้งเวลา

กดปุ่ม SET จะแสดง CURSOR ในหลักแรก จากนั้นกดปุ่ม UP /DOWN เพื่อตั้งชั่วโมง แล้วกดปุ่ม SET อีกครั้ง เพื่อตั้งนาทีและวินาที

เมื่อตั้ง ชั่วโมง นาที วินาที เสร็จเรียบร้อย

4. กดปุ่ม SET ตำแหน่งของเครื่องตามรหัสจังหวัด

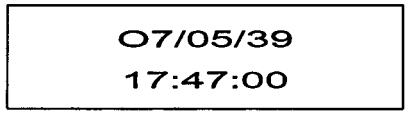
กดปุ่ม SET จะเกิด CURSOR ที่ตำแหน่งสุดท้ายของบรรทัดที่สอง และหากต้องการที่จะเปลี่ยนให้กดปุ่ม UP /DOWN ตำแหน่งรหัสจังหวัดจะเปลี่ยนไป

5. กดปุ่ม FUNCTION เพื่อตั้งอัตราค่าบริการ

กดปุ่ม SET จะเกิด CURSOR ที่ตำแหน่งสุดท้ายของบรรทัดที่สอง หากต้องการที่จะเปลี่ยนอัตราให้กดปุ่ม UP /DOWN ตามต้องการ

6. กดปุ่ม FUNCTION จะแสดงวัน เดือน ปี และเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



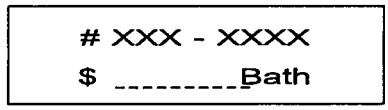
7. ต่อคู่สายโทรศัพท์เข้าที่ line telephone ที่ด้านหลังของเครื่องคำนวณค่าโทรศัพท์แบบอัตโนมัติและเมื่อต้องการยกหูโทรศัพท์ จอจะแสดงเครื่องหมาย " # :" เพื่อให้ทำการกดเลขหมายปลายทางที่ต้องการ

ถ้าหากมีสัญญาณ Ring back tone เครื่องจะรอจนกว่าปลายทางยกหูตอบรับ เครื่องจะเริ่มนับเวลา

แต่ถ้าหากเป็นสัญญาณ Busy tone DISPLAY จะแสดงอักษร BUSY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เมื่อเสร็จสิ้นการสนทนา DISPLAY จะแสดงเลขหมายปลายทาง และจำนวนเงิน



ถ้าต้องการตรวจสอบหมายเลข และค่าบริการครั้งล่าสุดให้กดปุ่ม Print DISPLAY จะแสดงดังรูป

9. ถ้ามีสัญญาณ Ringing ที่ DISPLAY จะแสดงตัวอักษรดังรูป

10. ในกรณีของหมายเลขที่ขึ้นต้นด้วยเลข 1 โทรศัพท์เคลื่อนที่ DISPLAY จะแสดงอักษรดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

5.1 บทสรุป

จากโครงงานนี้ได้เสนอเครื่องคิดค่าบริการโทรศัพท์อัตโนมัติรายงานผลทางจอภาพLCD และทางเครื่องพิมพ์ โดยใช้อุปกรณ์ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ของ INTELซึ่งประกอบไปด้วย CPU เบอร์ต่างๆได้แก่ 8031,8051,8052,8751,8752 และ8052ควบคุมการทำงานทั้งหมด จึงทำให้มีประสิทธิภาพในการใช้งานดีขึ้นและลดอุปกรณ์ทางด้านฮาร์ดแวร์ลงได้มากเป็นผลทำให้มีความสะดวกสบายในการสร้างอีกมากด้วยประโยชน์ที่จะได้รับคือ สามารถที่จะตรวจสอบการใช้โทรศัพท์สามารถจับและบันทึกหมายเลขโทรศัพท์ของผู้รับ เวลาสนทนาและจำนวนเงินที่ใช้บริการทั้งหมดทั้งแบบโทรภายในท้องถิ่นและโทรทางไกล โดยการพิมพ์ออกทางเครื่องพิมพ์เมื่อเลิกการใช้โทรศัพท์ทำให้สามารถนำข้อมูลไปตรวจสอบการใช้โทรศัพท์กับบิลล์เก็บเงินจากองค์การโทรศัพท์ได้

จากการทดลองผลการการทำงานของเครื่องปรากฏว่าเครื่องสามารถทำงานได้ตามที่กำหนดและมีผลที่น่าเชื่อถือได้ในระดับหนึ่งซึ่งจากการทดลองนี้ทำให้ทราบถึงปัญหาที่เกิดขึ้นและข้อเสียต่างๆ สามารถสรุปได้ดังนี้

1. สัญญาณภายในชุมสายโทรศัพท์ในห้องทดลองมีเสียงรบกวนสูงมากโดยเฉพาะในขณะที่เครื่องกำลังตรวจสอบสัญญาณตอบรับอยู่ ถ้ามีสัญญาณรบกวนสูงมากจะทำให้เกิดการผิดพลาดการทำงานจึงไม่ถูกต้องจึงควรที่จะหาวิธีการควบคุมภาคตอบรับให้ทำงานได้ถูกต้อง
2. ระดับสัญญาณต่างๆ ของเครื่องชุมสายโทรศัพท์ต่างๆ ภายในกรุงเทพที่จะส่งไปยังคู่สายโทรศัพท์ของผู้ใช้จะมีขนาดเพียงพอสำหรับเครื่องควบคุมการใช้โทรศัพท์ แต่ถ้าเป็นการติดต่อระหว่างจังหวัดระดับสัญญาณจะน้อยมากและมีสัญญาณรบกวนสูง ทั้งนี้เป็นเพราะการทำงานเครื่องควบคุมสายโทรศัพท์ขององค์การโทรศัพท์ควรที่จะเพิ่มอัตราการขยายขึ้นไปอีกพร้อมทั้งกำจัดสัญญาณรบกวน
3. เนื่องจากการทดลองที่ห้องทดลอง ซึ่งคู่สายโทรศัพท์เป็นคู่สายที่ผ่านชุมสายย่อยของสถาบัน ซึ่งลักษณะของสัญญาณโทรศัพท์ต่างๆ ที่ห้องทดลองกับชุมสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โทรศัพท์มีขนาดของสัญญาณที่แตกต่างกันจึงทำให้ผลการทดลอง ที่ได้ผิดไปจากมาตรฐานไปบ้าง

5.2 แนวความคิดในการพัฒนา

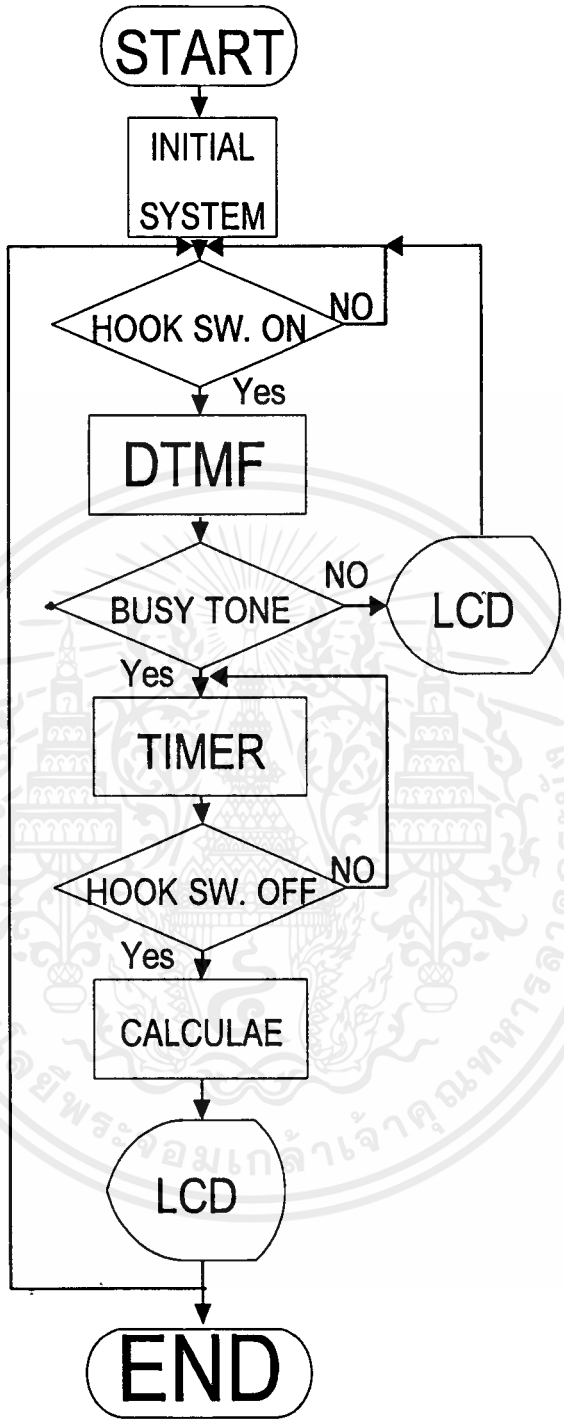
1. เนื่องจากเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติจำเป็นต้องใช้สัญญาณภายในชุมสายโทรศัพท์ ถ้าเสียงรบกวนสูงมากควรใช้วงจรกรองความถี่ที่มีความเรียบและละเอียดมากยิ่งขึ้น
2. ในการพัฒนาโครงงานชิ้นนี้ควรที่จะมีการแก้ไขในเรื่องของตัวเครื่องและโปรแกรมที่ใช้ควบคุมการทำงานของระบบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมควบคุมการทำงานไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************
        CPU     "8051.TBL"
        HOF     "INT8"
;*****************************************************
;MCS-51 INTERNAL REGISTERS
B:          EQU   0F0H          ;B REGISTER
A:          EQU   0E0H          ;ACCUMULATOR
PSW:        EQU   0D0H          ;PROGRAM STATUS WORD
IP:         EQU   0B8H          ;INTERRUPT PRIORITY
IE:         EQU   0A8H          ;INTERRUPT ENABLE
SCON:       EQU   98H           ;SERIAL CONTROL
P1:         EQU   90H           ;PORT 1
TH1:        EQU   8DH           ;TIMER 1 HIGH
TH0:        EQU   8CH           ;TIMER 0 HIGH
TL1:        EQU   8BH           ;TIMER 1 LOW
TL0:        EQU   8AH           ;TIMER 0 LOW
TMOD:       EQU   89H           ;TIMER MODE
TCON:       EQU   88H           ;TIMER CONTROL
DPH:        EQU   83H           ;DATA POINTER HIGH
DPL:        EQU   82H           ;DATA POINTER LOW
SP:         EQU   81H           ;STACK POINTER
R0:         EQU   00H           ;GENERAL REGISTER 0 ( BANK 0 )
R1:         EQU   01H           ;GENERAL REGISTER 1 ( BANK 0 )
R2:         EQU   02H           ;GENERAL REGISTER 2 ( BANK 0 )
R3:         EQU   03H           ;GENERAL REGISTER 3 ( BANK 0 )
R4:         EQU   04H           ;GENERAL REGISTER 4 ( BANK 0 )
R5:         EQU   05H           ;GENERAL REGISTER 5 ( BANK 0 )
R6:         EQU   06H           ;GENERAL REGISTER 6 ( BANK 0 )
R7:         EQU   07H           ;GENERAL REGISTER 7 ( BANK 0 )
;
;MCS-51 INTERNAL BIT ADDRESSES
;
C:          EQU   0D7H          ;CARRY FLAG
F0:         EQU   0D5H          ;USER FLAG 0
TR1:        EQU   08EH          ;TIMER 1 RUN CONTROL BIT
TR0:        EQU   08CH          ;TIMER 0 RUN CONTROL BIT
;
;USER DEFINE ADDRESS
;*****************************************************
NUM_BUFF      EQU       77h       ;BUFFER 9 BYTE 7F - 77
TIME_BUFF     EQU       71h       ;76h - 71h
TIME_USE      EQU       6Fh       ;USE TIME 70h - 6Fh
DIVCNT        EQU       6EH       ;COUNTER DIVISION
PROD          EQU       69H       ;5 BYTES ( 4+ EXTENSION) FOR DIVISION
FAC1          EQU       65H       ;OVERLAY FACTORS OF 16*16 BIT
FAC2          EQU       67H       ;OVERLAY
DIVI          EQU       FAC2      ;DIVISOR FOR 32/16 BIT DIVISION
QUOT          EQU       FAC1      ;QUOTIENT FOR 32/16 BIT DIVISION
OPERATOR1     EQU       63H
OPERATOR2     EQU       61H
OPERATOR3     EQU       5FH
DIGIT         EQU       5EH
TT_COST       EQU       5CH       ;TOTAL COST 5Ch -5Dh
TIME_B        EQU       5BH
TEMP          EQU       5AH       ;TEMP BYTE
U_FLAG        EQU       7FH       ;USER FLAG
U_PORT1A      EQU       0F800H    ;USER PORT 1A ADDRESS
U_PORT1B      EQU       0F801H    ;USER PORT 1B ADDRESS
U_PORT1C      EQU       0F802H    ;USER PORT 1C ADDRESS
U_PORT1_CTRL  EQU       0F803H    ;USER PORT 1 CONTRAL
U_PORT2A      EQU       0FC00H    ;USER PORT 2A ADDRESS
U_PORT2B      EQU       0FC01H    ;USER PORT 2B ADDRESS
U_PORT2C      EQU       0FC02H    ;USER PORT 2C ADDRESS
U_PORT2_CTRL  EQU       0FC03H    ;USER PORT 2 CONTRAL
LCDWRC        EQU       0FA00H     ;WRITE COMMAND TO LCD
LCDRDC        EQU       0FA01H     ;READ STATUS FROM LCD
LCDWRD        EQU       0FA02H     ;WRITE DATA TO LCD
LCDRDD        EQU       0FA03H     ;READ DATA FROM LCD
ST_L1         EQU       04d
ST_L2         EQU       20d
LOCATE_MAX    EQU       72d

;USER DEFINE RAM ADDRESS
;*****************************************************
LOCATE        EQU       00h
CHARGE_COST   EQU       01H         ;1 byte
LAST_NUM      EQU       10H         ;9 byte
N_DIGIT       EQU       20H         ;1 byte
LAST_COST     EQU       30H         ;2 byte


              org   0000h
              sjmp  start
;*********************************************************************
              org   0003h           ;interupt INT0
              reti
;*********************************************************************
              org   000bh           ;interupt TF0
              reti
;*********************************************************************
              org   0013h           ;interupt INT1
              reti                  ;not use
;*********************************************************************
              org   001bh           ;interupt TF1
              reti                  ;not use
;*********************************************************************
              org   0023h           ;interupt SERIAL
              reti
;*********************************************************************
start:
              mov   SP,#08h         ;set stack pointer to 08h
              lcall initial         ;initial set
              lcall demo
main:
              lcall hook_on
              lcall calendar
              lcall date            ;date function
              lcall time            ;time function
              lcall location        ;location function
              lcall charge          ;charge function
              sjmp  main

;*********************************************************************
; initial : set initial hardware & software
;*********************************************************************
initial:      mov   IE,#00h         ;not use  interupt
              mov   IP,#00h
              mov   TMOD,#00h       ;set not ues counter 0,1
              clr   PSW.3           ;set genaral registers
              clr   PSW.4           ;bank 0
              lcall init_disp       ;set inintial display LCD
              mov   DPTR,#U_PORT1_CTRL
              mov   A,#99h          ;port B is output other port input
              movx  @DPTR,A
              mov   DPTR,#U_PORT1B  ;out user port 1 B
              mov   A,#0ffh
              movx  @DPTR,A         ;
              lcall rtc_delay       ;
              mov   A,#00h          ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            movx  @DPTR,A          ;clear ringing D filp flop
            lcall rtc_delay        ;
            mov   A,#0ffh          ;
            movx  @DPTR,A          ;
            lcall rtc_delay
            mov   DPTR,#U_PORT2_CTRL
            mov   A,#9Bh           ;all port input
            movx  @DPTR,A

            mov   DPTR,#N_DIGIT
            mov   A,#9d            ; initial DIGIT = 9
            movx  @DPTR,A          ;
            mov   B,#09d
            mov   DPTR,#LAST_NUM
            mov   A,#0d            ; initial LAST_NUM = 000 00 0000
ini_LS_NUM: movx  @DPTR,A          ;
            inc   DPTR
            djnz  B,ini_LS_NUM
            mov   DPTR,#LAST_COST
            mov   A,#0d            ; initial LAST_COST = 000000
            movx  @DPTR,A          ;
            inc   DPTR
            movx  @DPTR,A
            ret
;*********************************************************************
; init_disp : set LCD
;*********************************************************************
init_disp:
            mov   a,#38h             ;function set  00111000b
            lcall lcdwi              ;dl=1 8 bit, n=0 1/16 duty,f=0 5x7
            mov   a,#0ch             ;function set 00001110b
            lcall lcdwi              ;disaplay on/off control
            mov   a,#06h             ;function set 00000110b
            lcall lcdwi              ;i/d=1 increment,s=0 right
            mov   a,#01h             ;function set 00000001b
            lcall lcdwi              ;clear all display
            ret
;*********************************************************************
; demo : display demo
;*********************************************************************
demo:       mov   DPTR,#st_start1
            lcall lcdldp             ;out string demo
            lcall delay1s
            lcall delay1s
            lcall delay1s
            mov   DPTR,#st_start2
            lcall lcdldp             ;out string demo
            lcall delay1s
            lcall delay1s
            lcall delay1s
            mov   a,#01h             ;function set 00000001b
            lcall lcdwi              ;clear all display
            ret
;*********************************************************************
; INTERUPT SERVICE ROUTINE INT0
;*********************************************************************
isr_int0:   mov   IE,#00h            ;not use INT0
            mov   SP,#08h
            mov   DPTR,#main
            push  DPL
            push  DPH
            reti
;*********************************************************************
;                RTC routine
;     P1.4 = I/O DATA
;     P1.5 = SCLK
;     P1.6 = RST
;*********************************************************************
;bytewr: write single byte to RTC
;in :  R6 = command
;      R7 = data
;reg:  A,B,R6,R7
;*********************************************************************
bytewr:     push  A
            push  B
            push  R6
            push  R7
            clr   P1.4               ;command byte 'write'
            lcall rtc_delay
            setb  P1.6               ;RST = '1'
            lcall rtc_delay
            mov   B,#08h
            clr   C
bytewr1:    mov   A,R6               ;
            rrc   A                  ;
            mov   R6,A               ;
            mov   P1.4,C             ;send command to RTC
            lcall sclkrw             ;
            djnz  B,bytewr1          ;
            mov   B,#08h
            clr   C
bytewr2:    mov   A,R7               ;
            rrc   A                  ;
            mov   R7,A               ;
            mov   P1.4,C             ;send data byte to RTC
            lcall sclkrw             ;
            djnz  B,bytewr2          ;
            clr   P1.6               ;RST = '0'
            lcall rtc_delay
            pop   R7
            pop   R6
            pop   B
            pop   A
            ret


;*********************************************************************
;byterd: read single byte from RTC
;in :  R6 = command
;out:  R7 = data
;reg:  A,B,R6,R7
;*********************************************************************
byterd:     push  A
            push  B
            push  R6
            setb  P1.4               ;command byte 'read'
            lcall rtc_delay
            setb  P1.6               ;RST = '1'
            lcall rtc_delay
            mov   B,#08h
            clr   C
byterd1:    mov   A,R6               ;
            rrc   A                  ;
            mov   R6,A               ;
            mov   P1.4,C             ;send command to RTC
            lcall sclkcom            ;
            djnz  B,byterd1          ;
            mov   B,#08h             ;receive data byte
            mov   R7,#00h            ;clear data buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
byterd2:    lcall sclkrw
            mov   A,R7
            mov   C,P1.4            ;read serial data
            rrc   A
            mov   R7,A
            djnz  B,byterd2         ;
            clr   P1.6              ;RST = '0'
            lcall rtc_delay
            pop   R6
            pop   B
            pop   A
            ret

;*********************************************************************
sclkcom:    clr   P1.5              ;serial clock for write command
            lcall rtc_delay
            setb  P1.5
            lcall rtc_delay
            ret
;*********************************************************************
sclkrw:     setb  P1.5              ;serial clock for
            lcall rtc_delay         ;read/write data
            clr   P1.5
            lcall rtc_delay
            ret
;*********************************************************************
rtc_delay:  push  R1
            mov   R1,#9
            djnz  R1,$
            pop   R1
            ret
;*********************************************************************
;                   LCD routine
;*********************************************************************
;lcdwi : LCD write instruction (RS = 0)
;in  : A
;reg : A,DPTR
;*********************************************************************
lcdwi:      push  DPH
            push  DPL
            mov   DPTR,#LCDWRC
            movx  @DPTR,A
            mov   DPTR,#LCDRDC
lcdwi1:     movx  A,@DPTR           ;wait for BF = 0
            jb    A.7,lcdwi1
            pop   DPL
            pop   DPH
            ret

;*********************************************************************
;lcdwd : LCD write data (RS = 1)
;in  : A
;reg : A,DPTR
;*********************************************************************
lcdwd:      push  DPH
            push  DPL
            mov   DPTR,#LCDWRD
            movx  @DPTR,A
            mov   DPTR,#LCDRDC
lcdwd1:     movx  A,@DPTR           ;wait for BF = 0
            jb    A.7,lcdwd1
            pop   DPL
            pop   DPH
            ret

;*********************************************************************
;lcdldp : load program memory to LCD(DMC202)
;in  : DPTR start block
;reg : A,R2,DPTR
;*********************************************************************
lcdldp:     push  A
            push  R2
            push  DPH
            push  DPL
            mov   A,#80h            ;set address line 1
            lcall lcdldps
            mov   A,#0c0h           ;set address line 2
            lcall lcdldps
            pop   DPL
            pop   DPH
            pop   R2
            pop   A
            ret

lcdldps:    lcall lcdwi             ;load one line
            mov   R2,#16d
lcdldps1:   clr   A
            movc  A,@A+DPTR         ;move program memory
            lcall lcdwd
            inc   DPTR
            djnz  R2,lcdldps1
            ret

;*********************************************************************
; goto : move cursor to position
; in  : A = address of LCD (0 - 31)
;*********************************************************************
goto:       push  A                 ;save acc
            push  B
            push  DPH               ;save data pointer
            push  DPL
            mov   B,A
            clr   C
            subb  A,#16d
            jnc   more_15
            mov   A,B
            add   A,#80h
            ljmp  goto1
more_15:    add   A,#0C0h
goto1:      lcall lcdwi             ;goto position
            pop   DPL               ;return data pointer
            pop   DPH
            pop   B
            pop   A                 ;return acc
            ret                     ;return to caller

;*********************************************************************
; writenum :write bcd number
; in : a = number (bcd 0-9)
;*********************************************************************
writenum:   push   A
            add    A,#30h
            lcall  lcdwd
            pop    A
            ret

;*********************************************************************
; write2num :write 2 character bcd number
; in : a = number (bcd)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************************************
write2num:  push   A
            push   A
            swap   A
            anl    A,#0fh
            lcall  writenum
            pop    A
            anl    A,#0fh
            lcall  writenum
            pop    A
            ret
;*********************************************************************
; cursoron :
;*********************************************************************
cursoron:   push  A
            mov   A,#0eh          ;function display on/off
            lcall lcdwi
            pop   A
            ret

;*********************************************************************
; cursoroff :
;*********************************************************************
cursoroff:  push  A
            mov   A,#0ch           ;function display on/off
            lcall lcdwi
            pop   A
            ret

;*********************************************************************
;procedure = delayls
;*********************************************************************
delayls:    push R0
            push R1
            push R2
            mov  R0,#8             ;n second r0 = n*10
delayls1:   mov  R1,#199           ;delay 100 ms
delayls2:   mov  R2,#250           ;delay 500 us
delayls3:   djnz R2,delayls3
            djnz R1,delayls2
            djnz R0,delayls1
            pop  R2
            pop  R1
            pop  R0
            ret

;*********************************************************************
;procedure = delay
;*********************************************************************
delay:      push R0
            push R1
            push R2
            mov  R0,#1             ;n second r0 = n*10
delay1:     mov  R1,#199           ;delay 100 ms
delay2:     mov  R2,#250           ;delay 500 us
delay3:     djnz R2,delay3
            djnz R1,delay2
            djnz R0,delay1
            pop  R2
            pop  R1
            pop  R0
            ret

;*********************************************************************
;calendar: sub to display date & time
;input  : -
;output : LCD
;usr    : A,R6,R7
;*********************************************************************
calendar:   push  A
            push  R6
            push  R7
            mov   A,#01h
            lcall lcdwi
calendar1:  lcall inkey
            push  A
            push  PSW
            lcall cursoroff
            mov   R6,#81h
            lcall byterd               ;read sec
            mov   A,R7
            clr   C                    ;
            subb  A,TIME_B             ;return if sec not change
            jnz   cale_next            ;
            ljmp  cale_ret             ;
cale_next:  mov   R6,#87h
            lcall byterd               ;read date
            mov   A,R7
            anl   A,#0f0h
            cjne  A,#00h,is_10date
            mov   A,#ST_L1
            lcall goto                 ;
            mov   A,#20h
            lcall lcdwd
            mov   A,R7                 ;no 10 digit of date
            lcall writenum             ;
            ajmp  wri1
is_10date:  mov   A,#ST_L1             ;
            lcall goto                 ;have 10 digit of date
            mov   A,R7                 ;
            lcall write2num
wri1:       mov   A,#"/"               ;write '/'
            lcall lcdwd
            mov   R6,#89h
            lcall byterd               ;read month
            mov   A,R7
            lcall write2num            ;write momth
            mov   A,#"/"               ;write '/'
            lcall lcdwd
            mov   R6,#8Dh
            lcall byterd               ;read year
            mov   A,R7
            lcall write2num            ;write year
;;;time
            mov   R6,#85h
            lcall byterd               ;read hour
            mov   A,R7
            anl   A,#030h
            cjne  A,#00h,is_10hour
            mov   A,#ST_L2
            lcall goto                 ;
            mov   A,#20h
            lcall lcdwd
            mov   A,R7                 ;no 10 digit of hour
            anl   A,#0fh               ;
            lcall writenum             ;
            ajmp  wri2
caleT:      ljmp  calendar1
is_10hour:  mov   A,#ST_L2             ;
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            lcall goto              ;have 10 digit of date
            mov   A,R7              ;
            anl   A,#3fh            ;
            lcall write2num
wri2:       mov   A,#":"            ;write ':'
            lcall lcdwd
            mov   R6,#83h
            lcall byterd            ;read min
            mov   A,R7
            lcall write2num         ;write min
            mov   A,#":"            ;write ':'
            lcall lcdwd
            mov   R6,#81h
            lcall byterd            ;read sec
            mov   A,R7
            mov   TIME_B,A
            anl   A,#7fh
            lcall write2num         ;write sec
cale_ret:   pop   PSW
            pop   A
            jc    caleT
            jz    caleT
            cjne  A,#05,cale_Fn     ;is Print key press ,no jump
            lcall last_use
            ljmp  calendar1
cale_Fn:    cjne  A,#01,caleT       ;is Fn key press ,no jump
            pop   R7
            pop   R6
            pop   A
            ret
;*********************************************************************
;last_use : Display last phone number & cost
;*********************************************************************
last_use:   push  DPH
            push  DPL
            push  A
            push  B
            push  R1
            push  R2
            push  R3
            mov   DPTR,#ST_LAST     ;
            lcall lcdldp            ;display ST_LAST
            mov   A,#03d
            lcall goto
            mov   DPTR,#N_DIGIT
            movx  A,@DPTR           ;
            mov   B,A               ;
            mov   R1,#09d
            mov   DPTR,#LAST_NUM    ;
last1:      movx  A,@DPTR           ;display last phone number
            lcall writenum          ;
            inc   DPTR              ;
            djnz  R1,last2
            sjmp  last3
last2:      djnz  B,last1
last3:      mov   A,#19d
            lcall goto
            mov   DPTR,#LAST_COST
            movx  A,@DPTR
            mov   B,A
            inc   DPTR
            movx  A,@DPTR
            mov   DPH,A
            mov   DPL,B             ;
            lcall hex2dec           ;
            mov   A,R1              ;display last cost
            lcall write2num         ;
            mov   A,R2
            lcall write2num
            mov   A,R3
            lcall write2num
            lcall delay1s           ;
            lcall delay1s           ;
            lcall delay1s           ;delay 5 sec
            lcall delay1s           ;
            lcall delay1s           ;
            mov   A,#01h
            lcall lcdwi             ;clear LCD
            pop   R3
            pop   R2
            pop   R1
            pop   B
            pop   A
            pop   DPL
            pop   DPH
            ret


;*********************************************************************
;inkey  : sub to check key is pressed
;input  : key
;output : A is number of key pressed
;       : C is 0 = normal , 1 = error
;use    : A
;*********************************************************************
inkey:      mov   DPTR,#U_PORT2C
            movx  A,@DPTR           ;read key from user Port2 C
            lcall delay            ;delay for switch pluase
            anl   A,#0f8h           ;select A.3 to A.7
            cjne  A,#0f8h,inkey1
            lcall chk_hook_on
            clr   A                 ;no key pressed
            clr   C
            ret
inkey1:     cjne  A,#78h,inkey2
wait1:      movx  A,@DPTR           ;read key from user Port2 C
            jnb   A.7,wait1
            mov   A,#01h            ;key 1 pressed
            clr   C
            ret
inkey2:     cjne  A,#0b8h,inkey3
wait2:      movx  A,@DPTR           ;read key from user Port2 C
            jnb   A.6,wait2
            mov   A,#02h            ;key 2 pressed
            clr   C
            ret
inkey3:     cjne  A,#0d8h,inkey4
            mov   A,#03h            ;key 3 pressed
            clr   C
            lcall delay
            ret
inkey4:     cjne  A,#0e8h,inkey5
            mov   A,#04h            ;key 4 pressed
            clr   C
            lcall delay
            ret
inkey5:     cjne  A,#0f0h,in_error
wait5:      movx  A,@DPTR           ;read key from user Port2 C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            jnb   A.6,wait5
            mov   A,#05h          ;key 5 pressed
            clr   C
            ret
in_error:   setb  C               ;error many key pressed
            ret

;*********************************************************************
;time   : sub to display & call set system time (format HH:MM:SS)
;input  : key
;output : LCD
;usr    : DPTR,A,R6,R7
;*********************************************************************
time:       push  DPH
            push  DPL
            push  A
            push  R6
            push  R7
            mov   DPTR,#TIME_TITLE
            lcall lcdldp          ;write date title to LCD

time_pro_1: lcall inkey           ;get key
            push  A               ;save key to stack
            push  PSW
            mov   R6,#85h
            lcall byterd              ;read hour
            mov   A,R7
            anl   A,#030h
            cjne  A,#00h,time_10
            mov   A,#ST_L2
            lcall goto                ;
            mov   A,#20h
            lcall lcdwd
            mov   A,R7                ;no 10 digit of hour
            anl   A,#0fh              ;
            lcall writenum            ;
            ajmp  time1
time_10:    mov   A,#ST_L2            ;
            lcall goto                ;have 10 digit of date
            mov   A,R7                ;
            anl   A,#3fh              ;
            lcall write2num
time1:      mov   A,#":"              ;write ':'
            lcall lcdwd
            mov   R6,#83h
            lcall byterd              ;read min
            mov   A,R7
            lcall write2num           ;write min
            mov   A,#":"              ;write ':'
            lcall lcdwd
            mov   R6,#81h
            lcall byterd              ;read sec
            mov   A,R7
            anl   A,#7fh
            lcall write2num           ;write sec
            pop   PSW             ;restore key
            pop   A               ;
            jz    time_pro_1      ;no key pressed
            jc    time_pro_1      ;many key pressed
            cjne  A,#02h,time_Fn  ;Is set time key pressed? no,jump
            lcall set_time        ;call set time subroutine
            ljmp  time_pro_1
time_Fn:    cjne  A,#01h,time_pro_1 ;Is time funct. key pressed?
            pop   R7              ;yes ,now return
            pop   R6
            pop   A
            pop   DPL
            pop   DPH
            ret

;*********************************************************************
;set_time : sub to set time RTC
;input  : key
;output : real time system
;use    : DPTR,A,R0,R1,R6,R7
;*********************************************************************

set_time:   push  A
            push  B
            push  R0
            push  R1
            push  R6
            push  R7
            lcall cursoron

            mov   R1,#time_buff   ;set time buffer
            mov   R6,#85h
            lcall byterd          ;    ;read hour
            mov   A,R7            ;
            anl   A,#03fh         ;
            mov   B,A             ;
            swap  A               ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 10Hr
            inc   R1              ;
            mov   A,B             ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 1Hr
            inc   R1              ;
            mov   R6,#83h          ;get time from RTC to time bffer
            lcall byterd          ;    ;read min
            mov   A,R7            ;
            mov   B,A             ;
            swap  A               ;          *
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 10min
            inc   R1              ;
            mov   A,B             ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 1min
            inc   R1              ;
            mov   R6,#81h          ;
            lcall byterd          ;    ;read sec
            mov   A,R7            ;
            mov   B,A             ;
            swap  A               ;                                    *
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 10sec
            inc   R1              ;
            mov   A,B             ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 1sec

            mov   A,#ST_L2
            lcall goto
            mov   R1,#TIME_BUFF
            mov   A,@R1
            lcall writenum        ;write hour
            inc   R1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov   A,@R1
            lcall writenum
            mov   A,#":"            ;write ':'
            lcall lcdwd
            inc   R1
            mov   A,@R1
            lcall writenum          ;write min
            inc   R1
            mov   A,@R1
            lcall writenum
            mov   A,#":"            ;write ':'
            lcall lcdwd
            inc   R1
            mov   A,@R1
            lcall writenum          ;write sec
            inc   R1
            mov   A,@R1
            lcall writenum
            mov   R1,#TIME_BUFF   ;R1 porint to TIME BUFFER
            mov   R0,#ST_L2       ;R0 point LCD cursor position
            mov   A,R0
            lcall goto

Set_t_key:  lcall inkey           ;get key
            jz    set_t_key
            jc    set_t_key
            cjne  A,#03h,time_down1 ;Is up key press? no,jump
            cjne  R0,#ST_L2+0,hr1_up ;yes,Is Hr10 digit? no,jump
            inc   TIME_BUFF       ;yes,inc Hr10
            mov   A,TIME_BUFF     ;
            cjne  A,#03h,show10hru  ;check Hr10 is over range( > 2)
            mov   TIME_BUFF,#00h  ;
show10hru:  mov   A,TIME_BUFF
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
            sjmp  Set_t_key

hr1_up:     cjne  R0,#ST_L2+1,min10_up ;Is Hr1 digit? no,jump
            inc   TIME_BUFF+1     ;yes,inc Hr1
            mov   A,TIME_BUFF     ;
            cjne  A,#02h,hr1_10   ;check Hr1 is over range
            mov   A,TIME_BUFF+1   ;    (Hr10 = 2 => Hr1 < 4)
            cjne  A,#04h,hr1_10   ;    (Hr10 <> 2 => Hr1 = 0-9)
            mov   TIME_BUFF+1,#00h  ;
            sjmp  show1hru
hr1_10:     mov   A,TIME_BUFF+1
            cjne  A,#0ah,show1hru
            mov   TIME_BUFF+1,#00h
show1hru:   mov   A,TIME_BUFF+1
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
            ljmp  Set_t_key

min10_up:   cjne  R0,#ST_L2+3,min1_up ;Is Min10 digit? no,jump
            inc   TIME_BUFF+2     ;yes,inc Min10
            mov   A,TIME_BUFF+2   ;
            cjne  A,#06h,show10minu ;check Min10 is out range( > 5)
            mov   TIME_BUFF+2,#00h ;
show10minu: mov   A,TIME_BUFF+2
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
            ljmp  Set_t_key

min1_up:    cjne  R0,#ST_L2+4,sec10_up ;Is Min1 digit? no,jump
            inc   TIME_BUFF+3     ;yes,inc Min1
            mov   A,TIME_BUFF+3   ;
            cjne  A,#0ah,show1minu ;check Min1 is out range( > 9)
            mov   TIME_BUFF+3,#00h ;
show1minu:  mov   A,TIME_BUFF+3
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
            ljmp  Set_t_key

time_down1: ljmp  time_down

sec10_up:   cjne  R0,#ST_L2+6,sec1_up ;Is Sec10 digit? no,jump
            inc   TIME_BUFF+4     ;yes,inc Sec10
            mov   A,TIME_BUFF+4   ;
            cjne  A,#06h,show10secu ;check sec10 is out range( > 5)
            mov   TIME_BUFF+4,#00h ;
show10secu: mov   A,TIME_BUFF+4
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
            ljmp  Set_t_key

sec1_up:    inc   TIME_BUFF+5     ;Sec1 digit,inc Sec1
            mov   A,TIME_BUFF+5   ;
            cjne  A,#0ah,show1secu ;check Sec1 is out range( > 9)
            mov   TIME_BUFF+5,#00h ;
show1secu:  mov   A,TIME_BUFF+5
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
            ljmp  Set_t_key

time_down:  cjne  A,#04h,time_set1 ;Is down key press? no,jump
            cjne  R0,#ST_L2+0,hr1_down ;yes,Is Hr10 digit? no,jump
            dec   TIME_BUFF+0       ;yes,dec Hr10
            mov   A,TIME_BUFF+0     ;
            cjne  A,#0ffh,show10hrd ;check Hr10 is out range( < 0)
            mov   TIME_BUFF+0,#02h  ;
show10hrd:  mov   A,TIME_BUFF+0
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
            ljmp  Set_t_key

hr1_down:   cjne  R0,#ST_L2+1,min10_down ;Is Hr1 digit? no,jump
            dec   TIME_BUFF+1     ;yes,dec Hr1
            mov   A,TIME_BUFF+1   ;
            cjne  A,#02h,hr1_0    ;check Hr1 is out range
            mov   A,TIME_BUFF+0   ;    (Hr10 = 2 => Hr1 < 3)
            cjne  A,#0ffh,hr1_0   ;    (Hr10 <> 2 => Hr1 = 0-9)
            mov   TIME_BUFF+1,#03h   ;
            sjmp  show1hrd
hr1_0:      mov   A,TIME_BUFF+1
            cjne  A,#0ffh,show1hrd
            mov   TIME_BUFF+1,#09h
show1hrd:   mov   A,TIME_BUFF+1
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ljmp  Set_t_key

time_set1:  ljmp  time_set

min10_down: cjne  R0,#ST_L2+3,min1_down ;Is Min10 digit? no,jump
            dec   TIME_BUFF+2         ;yes,dec Min10
            mov   A,TIME_BUFF+2       ;
            cjne  A,#0ffh,show10mind ;check Min10 is out range( < 0)
            mov   TIME_BUFF+2,#05h    ;
show10mind: mov   A,TIME_BUFF+2
            lcall writenum
            mov   A,R0
            lcall goto              ;don't change cursor
            ljmp  Set_t_key

min1_down:  cjne  R0,#ST_L2+4,sec10_down ;Is Min1 digit? no,jump
            dec   TIME_BUFF+3         ;yes,dec Min1
            mov   A,TIME_BUFF+3       ;
            cjne  A,#0ffh,show1mind ;check Min1 is out range( < 0)
            mov   TIME_BUFF+3,#09h    ;
show1mind:  mov   A,TIME_BUFF+3
            lcall writenum
            mov   A,R0
            lcall goto              ;don't change cursor
            ljmp  Set_t_key

sec10_down: cjne  R0,#ST_L2+6,sec1_down ;Is Sec10 digit? no,jump
            dec   TIME_BUFF+4         ;yes,dec Sec10
            mov   A,TIME_BUFF+4       ;
            cjne  A,#0ffh,show10secd ;check sec10 is out range( > 0)
            mov   TIME_BUFF+4,#05h    ;
show10secd: mov   A,TIME_BUFF+4
            lcall writenum
            mov   A,R0
            lcall goto              ;don't change cursor
            ljmp  Set_t_key

sec1_down:  dec   TIME_BUFF+5         ;Sec1 digit,dec Sec1
            mov   A,TIME_BUFF+5       ;
            cjne  A,#0ffh,show1secd ;check Sec1 is out range( > 0)
            mov   TIME_BUFF+5,#09h    ;
show1secd:  mov   A,TIME_BUFF+5
            lcall writenum
            mov   A,R0
            lcall goto              ;don't change cursor
Set_t_key2: ljmp  Set_t_key

time_set:   cjne  A,#02h,Set_t_Fn   ;Is set key pressed? no,jump

            cjne  R0,#ST_L2+7,no_over   ;yes

            mov   A,TIME_BUFF+0       ; check Hr1 is out range
            cjne  A,#02h,time_save    ;    (Hr10 = 2 => Hr1 < 3)
            mov   A,#04h              ;
            clr   C                   ;
            subb  A,TIME_BUFF+1       ;
            jnc   time_save           ;
            mov   TIME_BUFF+1,#03h    ;
            sjmp  time_save

no_over:    inc   R0                  ;increase  digit pointer
            cjne  R0,#ST_L2+2,no_inc1
            inc   R0
no_inc1:    cjne  R0,#ST_L2+5,no_inc2
            inc   R0
no_inc2:    mov   A,R0
            lcall goto
            ljmp  Set_t_key

time_save:  mov   R6,#8Eh             ;write protection
            mov   R7,#00h
            lcall bytewr
            mov   R0,#TIME_BUFF       ;R0 point to TIME_BUFF
            mov   R6,#84h             ;first command
w_time_RTC: mov   A,@R0               ;write time set to RTC
            swap  A
            mov   B,A
            inc   R0
            mov   A,@R0
            orl   A,B
          ;  orl   A,#80h             ;*** 12/24 hour
            mov   R7,A                ;write hour
            lcall bytewr
            inc   R0
            dec   R6                  ;chang command
            dec   R6                  ;
            cjne  R0,#TIME_BUFF+6,w_time_RTC ;
            mov   R6,#8Eh             ;write protection "active"
            mov   R7,#80h             ;
            lcall bytewr
            sjmp  Set_t_ret

Set_t_Fn:   cjne  A,#01h,Set_t_key2 ;Is set time funct. key press?

Set_t_ret   lcall cursoroff
            pop   R7                  ;yes,return
            pop   R6
            pop   R1
            pop   R0
            pop   B
            pop   A
            ret

;*********************************************************************
;date   : sub to display & call set system date
;input  : key
;output : LCD
;usr    : DPTR,A,R6,R7
;*********************************************************************
date:       push  DPH
            push  DPL
            push  A
            push  R6
            push  R7
            mov   DPTR,#DATE_TITLE
            lcall lcdldp            ;write date title to LCD
date_pro_1: lcall inkey             ;get key
            push  A                 ;save key to stack
            push  PSW
            mov   R6,#87h
            lcall byterd              ;read date
            mov   A,R7
            anl   A,#0f0h
            cjne  A,#00h,date_10
            mov   A,#ST_L2
            lcall goto                ;
            mov   A,#20h
            lcall lcdwd
            mov   A,R7                ;no 10 digit of date
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            lcall writenum          ;
            ajmp  date1
date_10:    mov   A,#ST_L2          ;
            lcall goto              ;have 10 digit of date
            mov   A,R7              ;
            lcall write2num
date1:       mov   A,#"/"           ;write '/'
            lcall lcdwd
            mov   R6,#89h
            lcall byterd            ;read month
            mov   A,R7
            lcall write2num         ;write momth
            mov   A,#"/"            ;write '/'
            lcall lcdwd
            mov   R6,#8Dh
            lcall byterd            ;read year
            mov   A,R7
            lcall write2num         ;write year
            pop   PSW           ;restore key
            pop   A             ;
            jz    date_pro_1    ;no key pressed
            jc    date_pro_1    ;many key pressed
            cjne  A,#02h,date_Fn  ;Is set date key pressed? no,jump
            lcall set_date        ;call set date subroutine
            ljmp  date_pro_1
date_Fn:    cjne  A,#01h,date_pro_1 ;Is date funct. key pressed?
            pop   R7              ;yes ,now return
            pop   R6
            pop   A
            pop   DPL
            pop   DPH
            ret

;*********************************************************************
;set_date : sub to set date RTC
;input  : key
;output : date system
;use    : A,R0,R1,R6,R7
;*********************************************************************
set_date:   push  A
            push  B
            push  R0
            push  R1
            push  R6
            push  R7

            lcall cursoron
            mov   R1,#time_buff   ;set time buffer
            mov   R6,#87h
            lcall byterd          ;    ;read date
            mov   A,R7            ;
            mov   B,A             ;
            swap  A               ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 10date
            inc   R1              ;
            mov   A,B             ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 1date
            inc   R1              ;
            mov   R6,#89h         ;get date from RTC to time bffer
            lcall byterd          ;    ;read month
            mov   A,R7            ;
            mov   B,A             ;
            swap  A               ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 10month
            inc   R1              ;
            mov   A,B             ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 1month
            inc   R1              ;
            mov   R6,#8Dh         ;
            lcall byterd          ;    ;read year
            mov   A,R7            ;
            mov   B,A             ;
            swap  A               ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 10year
            inc   R1              ;
            mov   A,B             ;
            anl   A,#0fh          ;
            mov   @R1,A           ;    save 1year

            mov   A,#ST_L2
            lcall goto
            mov   R1,#TIME_BUFF
            mov   A,@R1
            lcall writenum          ;write date
            inc   R1
            mov   A,@R1
            lcall writenum
            mov   A,#"/"            ;write '/'
            lcall lcdwd
            inc   R1
            mov   A,@R1
            lcall writenum          ;write month
            inc   R1
            mov   A,@R1
            lcall writenum
            mov   A,#"/"            ;write '/'
            lcall lcdwd
            inc   R1
            mov   A,@R1
            lcall writenum          ;write year
            inc   R1
            mov   A,@R1
            lcall writenum
            mov   R1,#TIME_BUFF   ;R1 porint to TIME BUFFER
            mov   R0,#ST_L2       ;R0 point LCD cursor position
            mov   A,R0
            lcall goto


Set_d_key:  lcall inkey           ;get key
            jz    set_d_key
            jc    set_d_key
            cjne  A,#03h,date_down1 ;Is up key press? no,jump
            cjne  R0,#ST_L2+0,dy1_up  ;yes,Is Dy10 digit? no,jump
            inc   TIME_BUFF+0       ;yes,inc Dy10
            mov   A,TIME_BUFF+0     ;
            cjne  A,#04h,show10dyu ;check DY10 is out range( > 3)
            mov   TIME_BUFF+0,#00h  ;
show10dyu:  mov   A,TIME_BUFF+0
            lcall writenum
            mov   A,R0
            lcall goto            ;don't change cursor
            ljmp  Set_d_key
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dy1_up:     cjne  R0,#ST_L2+1,mon10_up ;Is Dy1 digit? no,jump
            inc   TIME_BUFF+1        ;yes,inc Dy1
            mov   A,TIME_BUFF+0      ;
            cjne  A,#03h,dy1_30      ;
            mov   A,TIME_BUFF+1      ;check Dy1 is out range
            cjne  A,#02h,dy1_30      ; (Dy10 = 3 => Dy1 > 1)
            mov   TIME_BUFF+1,#00h   ; (Dy10 <> 3 => Dy1 > 9)
            sjmp  showldyu           ;
dy1_30:     mov   A,TIME_BUFF+1      ;
            cjne  A,#0ah,showldyu
            mov   TIME_BUFF+1,#00h
showldyu:   mov   A,TIME_BUFF+1
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key

mon10_up:   cjne  R0,#ST_L2+3,mon1_up ;Is Mon10 digit? no,jump
            inc   TIME_BUFF+2        ;yes,inc Mon10
            mov   A,TIME_BUFF+2      ;
            cjne  A,#02h,show10monu ;check Mon10 is out range( > 1)
            mov   TIME_BUFF+2,#00h   ;
show10monu: mov   A,TIME_BUFF+2
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key

date_down1: ljmp   date_down

mon1_up:    cjne  R0,#ST_L2+4h,yr10_up ;Is Mon1 digit? no,jump
            inc   TIME_BUFF+3        ;yes,inc Min1
            mov   A,TIME_BUFF+2      ;
            cjne  A,#01h,mon1_10     ;
            mov   A,TIME_BUFF+3      ;check Mon1 is out range
            cjne  A,#03h,mon1_10     ;     (Mon10 = 1 => Mon1 > 2)
            mov   TIME_BUFF+3,#00h   ;     (Mon10 <> 1 => Mon1 > 9)
            sjmp  showlmonu          ;
mon1_10:    mov   A,TIME_BUFF+3      ;
            cjne  A,#0ah,showlmonu
            mov   TIME_BUFF+3,#00h   ;
showlmonu:  mov   A,TIME_BUFF+3
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key

yr10_up:    cjne  R0,#ST_L2+6h,yr1_up ;Is Yr10 digit? no,jump
            inc   TIME_BUFF+4        ;yes,inc Yr10
            mov   A,TIME_BUFF+4      ;
            cjne  A,#0ah,show10yru ;check Yr10 is out range( > 9)
            mov   TIME_BUFF+4,#00h ;
show10yru:  mov   A,TIME_BUFF+4
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key

yr1_up:     inc   TIME_BUFF+5         ;Yr1 digit,inc Yr1
            mov   A,TIME_BUFF+5      ;
            cjne  A,#0ah,show1yru  ;check Yr1 is out range( > 9)
            mov   TIME_BUFF+5,#00h ;
show1yru:   mov   A,TIME_BUFF+5
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key

date_down:  cjne  A,#04h,date_set1 ;Is down key press? no,jump
            cjne  R0,#ST_L2+0,dy1_down ;yes,Is Dy10 digit? no,jump
            dec   TIME_BUFF+0        ;yes,dec Dy10
            mov   A,TIME_BUFF+0      ;
            cjne  A,#0ffh,show10dyd ;check Hr10 is out range( < 0)
            mov   TIME_BUFF+0,#03h   ;
show10dyd:  mov   A,TIME_BUFF+0
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key

dy1_down:   cjne  R0,#ST_L2+1,mon10_down ;Is Dy1 digit? no,jump
            dec   TIME_BUFF+1        ;yes,dec Dy1
            mov   A,TIME_BUFF+0      ;
            cjne  A,#03h,dy1_0       ;check Dy1 is out range
            mov   A,TIME_BUFF+1      ;     (Dy10 = 3 => Dy1 > 1)
            cjne  A,#0ffh,dy1_0      ;     (Dy10 <> 3 => Dy1 < 0)
            mov   TIME_BUFF+1,#01h   ;
            sjmp  showldyd
dy1_0:      mov   A,TIME_BUFF+1
            cjne  A,#0ffh,showldyd
            mov   TIME_BUFF+1,#09h
showldyd:   mov   A,TIME_BUFF+1
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key

mon10_down: cjne  R0,#ST_L2+3,mon1_down ;Is Mon10 digit? no,jump
            dec   TIME_BUFF+2        ;yes,dec Mon10
            mov   A,TIME_BUFF+2      ;
            cjne  A,#0ffh,show10mond ;check Min10 is out range( < 0)
            mov   TIME_BUFF+2,#01h   ;
show10mond: mov   A,TIME_BUFF+2
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key
date_set1:  ljmp  date_set
mon1_down:  cjne  R0,#ST_L2+4h,yr10_down ;Is Mon1 digit? no,jump
            dec   TIME_BUFF+3        ;yes,dec Mon1
            mov   A,TIME_BUFF+2      ;
            cjne  A,#01h,mon1_0      ;check Mon1 is out range
            mov   A,TIME_BUFF+3      ;     (Mon10 = 1 => Mon1 > 2)
            cjne  A,#0ffh,mon1_0     ;     (Mon10 <> 1 => Mon1 < 0)
            mov   TIME_BUFF+3,#02h
            sjmp  showlmond
mon1_0:     mov   A,TIME_BUFF+3      ;
            cjne  A,#0ffh,showlmond ;check Mon1 is out range( < 0)
            mov   TIME_BUFF+3,#09h   ;
showlmond:  mov   A,TIME_BUFF+3
            lcall writenum
            mov   A,R0
            lcall goto               ;don't change cursor
            ljmp  Set_d_key

yr10_down:  cjne  R0,#ST_L2+6,yr1_down ;Is Yr10 digit? no,jump
            dec   TIME_BUFF+4        ;yes,dec Yr10
            mov   A,TIME_BUFF+4      ;
            cjne  A,#0ffh,show10yrd ;check Yr10 is out range( < 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov   TIME_BUFF+4,#09h  ;
show10yrd:  mov   A,TIME_BUFF+4
            lcall writenum
            mov   A,R0
            lcall goto              ;don't change cursor
            ljmp  Set_d_key

yr1_down:   dec   TIME_BUFF+5       ;Yr1 digit,dec Yr1
            mov   A,TIME_BUFF+5     ;
            cjne  A,#0ffh,showlyrd  ;check Yr1 is out range( < 0)
            mov   TIME_BUFF+5,#09h  ;
showlyrd:   mov   A,TIME_BUFF+5
            lcall writenum
            mov   A,R0
            lcall goto              ;don't change cursor
Set_d_key2: ljmp  Set_d_key

date_set:   cjne  A,#02h,Set_d_Fn   ;Is set key pressed? no,jump

            inc   R0                ;increase  digit pointer
            cjne  R0,#ST_L2+2,no_inc1d
            inc   R0
no_inc1d:   cjne  R0,#ST_L2+5,no_inc2d
            inc   R0
no_inc2d:   mov   A,R0
            lcall goto
            cjne  R0,#ST_L2+8,Set_d_key2

            mov   R0,#TIME_BUFF
            cjne  R0,#03h,date_save
            mov   R0,#TIME_BUFF+1
            mov   A,R0
            clr   C
            subb  A,#01h
            jnc   date_save
            mov   TIME_BUFF+1,#01h

date_save:  mov   R6,#8Eh           ;write protection
            mov   R7,#00h
            lcall bytewr
            mov   R0,#TIME_BUFF     ;R0 point to TIME_BUFF
            mov   R6,#86h           ;first command
w_date_RTC: mov   A,@R0             ;write date set to RTC
            swap  A
            mov   B,A
            inc   R0
            mov   A,@R0
            orl   A,B
            mov   R7,A              ;write
            lcall bytewr
            inc   R0
            inc   R6                ;chang command
            inc   R6
            cjne  R6,#8Ah,no_DAY    ;if it DAY address, no jump
            inc   R6                ;yes,increase to YEAR address
            inc   R6                ;
no_DAY:     cjne  R0,#TIME_BUFF+6,w_date_RTC ;
            mov   R6,#8Eh           ;write protection "active"
            mov   R7,#80h           ;
            lcall bytewr
            sjmp  Set_d_ret

Set_d_Fn:   cjne  A,#01h,Set_d_key2 ;Is set time funct. key press?

Set_d_ret   lcall cursoroff
            pop   R7                ;yes,return
            pop   R6
            pop   R1
            pop   R0
            pop   B
            pop   A
            ret

;*********************************************************************
;hex2dec : convert heximal to dacimal
;in  :    DPTR
;out :    R1,R2,R3
;reg :    A,R0,R1,R2,R3,R4,R5,DPTR
;*********************************************************************
hex2dec:    push  DPH
            push  DPL
            push  A
            push  R0
            push  R4
            push  R5
            clr   A
            mov   R1,A
            mov   R2,A
            mov   R3,A
            mov   R4,#16d
h2d1:       mov   A,DPL
            rlc   A
            mov   DPL,A
            mov   A,DPH
            rlc   A
            mov   DPH,A
            mov   R5,#03h           ;add decimal
            mov   R0,#03h           ;index to R3
h2d2:       mov   A,@R0
            addc  A,A
            da    A
            mov   @R0,A
            dec   R0
            djnz  R5,h2d2
            djnz  R4,h2d1
            pop   R5
            pop   R4
            pop   R0
            pop   A
            pop   DPL
            pop   DPH
            ret


;*********************************************************************
;location: display & call set location of telephone
;out : LCD
;reg : DPTR,A,B,R1,R2,R3
;*********************************************************************
location:   push  DPH
            push  DPL
            push  A
            push  B
            push  R0
            push  R1
            push  R2
            push  R3
            lcall cursoroff
            mov   DPTR,#LOCATE_TITLE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            inc   R6
            cjne  R6,#8Ah,no_DAY    ;if it DAY address, no jump
            inc   R6                ;yes,increase to YEAR address
            inc   R6                ;
no_DAY:     cjne  R0,#TIME_BUFF+6,w_date_RTC ;
            mov   R6,#8Eh           ;write protection "active"
            mov   R7,#80h           ;
            lcall bytewr
            sjmp  Set_d_ret

Set_d_Fn:   cjne  A,#01h,Set_d_key2 ;Is set time funct. key press?

Set_d_ret   lcall cursoroff
            pop   R7                ;yes,return
            pop   R6
            pop   R1
            pop   R0
            pop   B
            pop   A
            ret

;*********************************************************************
;hex2dec : convert heximal to dacimal
;in  :    DPTR
;out :    R1,R2,R3
;reg :    A,R0,R1,R2,R3,R4,R5,DPTR
;*********************************************************************
hex2dec:    push  DPH
            push  DPL
            push  A
            push  R0
            push  R4
            push  R5
            clr   A
            mov   R1,A
            mov   R2,A
            mov   R3,A
            mov   R4,#16d
h2d1:       mov   A,DPL
            rlc   A
            mov   DPL,A
            mov   A,DPH
            rlc   A
            mov   DPH,A
            mov   R5,#03h           ;add decimal
            mov   R0,#03h           ;index to R3
h2d2:       mov   A,@R0
            addc  A,A
            da    A
            mov   @R0,A
            dec   R0
            djnz  R5,h2d2
            djnz  R4,h2d1
            pop   R5
            pop   R4
            pop   R0
            pop   A
            pop   DPL
            pop   DPH
            ret


;*********************************************************************
;location: display & call set location of telephone
;out : LCD
;reg : DPTR,A,B,R1,R2,R3
;*********************************************************************
location:   push  DPH
            push  DPL
            push  A
            push  B
            push  R0
            push  R1
            push  R2
            push  R3
            lcall cursoroff
            mov   DPTR,#LOCATE_TITLE
            lcall lcdldp            ;write locate title to LCD
locate_1:   mov   DPTR,#LOCATE
            movx  A,@DPTR
            push  A
            mov   B,#03h
            mul   AB
            push  A
            push  A
            push  A
            mov   A,#ST_L2
            lcall goto
            pop   A
            mov   DPTR,#LOCATE_DATA
            movc  A,@A+DPTR         ;
            lcall lcdwd             ;
            inc   DPTR              ;
            pop   A
            movc  A,@A+DPTR         ;write country char. code
            lcall lcdwd             ;
            inc   DPTR              ;
            pop   A
            movc  A,@A+DPTR         ;
            lcall lcdwd             ;
            mov   A,#ST_L2+5
            lcall goto
            pop   A
            mov   DPL,A             ;
            mov   DPH,#00h          ;
            inc   DPTR
            lcall hex2dec           ;write country num. code
            mov   A,R3              ;
            lcall write2num         ;


locate_2:   lcall inkey
            jz    locate_2
            jc    locate_2
            cjne  A,#02h,locate_3   ;is set press ,no jump
            lcall locate_set
            ljmp  locate_1
locate_3:   cjne  A,#01h,locate_2   ;is Function press ,no jump
            pop   R3
            pop   R2
            pop   R1
            pop   R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pop   B
            pop   A
            pop   DPL
            pop   DPH
            ret

;*********************************************************************
;locate_set : set location of telephone
;*********************************************************************

locate_set: push  DPH
            push  DPL
            push  A
            push  B
            push  R0
            push  R1
            push  R2
            push  R3
            lcall cursoron
            mov   DPTR,#LOCATE
            movx  A,@DPTR                ;read locate old data
            mov   R0,A
locate_s1:  lcall inkey
            push  A
            push  PSW
            mov   A,R0
            mov   B,#03h
            mul   AB
            push  A
            push  A
            push  A
            mov   A,#ST_L2
            lcall goto
            pop   A
            mov   DPTR,#LOCATE_DATA
            movc  A,@A+DPTR          ;
            lcall lcdwd              ;
            inc   DPTR               ;
            pop   A
            movc  A,@A+DPTR          ;write country char. code
            lcall lcdwd              ;
            inc   DPTR               ;
            pop   A
            movc  A,@A+DPTR          ;
            lcall lcdwd              ;
            mov   A,#ST_L2+5
            lcall goto
            mov   DPL,R0             ;
            mov   DPH,#00h           ;
            inc   DPTR               ;
            lcall hex2dec            ;write country num. code
            mov   A,R3               ;
            lcall write2num          ;
            mov   A,#ST_L2+6
            lcall goto
            pop   PSW
            pop   A
            jz    locate_s1
            jc    locate_s1
            cjne  A,#03,locate_down ;is locate code up,no jump
            inc   R0
            cjne  R0,#LOCATE_MAX+1,no_max ;is locate max,no jump
            mov   R0,#00h
no_max:     ljmp  locate_s1

locate_down: cjne  A,#04,locate_st ;is locate code down,no jump
            dec   R0
            cjne  R0,#0ffh,no_min ;is locate min,no jump
            mov   R0,#LOCATE_MAX
no_min:     ljmp  locate_s1
locate_st:  cjne  A,#02h,locate_Fn ;is set press,no jump
            mov   DPTR,#LOCATE
            mov   A,R0
            movx  @DPTR,A
            sjmp  locate_ret
locate_Fn:  cjne  A,#01,locate_s1
locate_ret: lcall cursoroff
            pop   R3
            pop   R2
            pop   R1
            pop   R0
            pop   B
            pop   A
            pop   DPL
            pop   DPH
            ret


;*********************************************************************
;charge: display & call set charge of serviec
;out : LCD
;reg : DPTR,A,B,R1,R2,R3
;*********************************************************************


charge:     push  DPH
            push  DPL
            push  A
            push  B
            push  R0
            push  R1
            push  R2
            push  R3
            mov   DPTR,#CHARGE_TITLE
            lcall lcdldp             ;write charge title to LCD
charge_1:   mov   DPTR,#CHARGE_COST
            movx  A,@DPTR
            push  A
            mov   A,#ST_L2+5
            lcall goto
            pop   A
            mov   DPL,A              ;
            mov   DPH,#00h           ;
            lcall hex2dec            ;write charge cost
            mov   A,R3               ;
            lcall write2num          ;

charge_2:   lcall inkey
            jz    charge_2
            jc    charge_2
            cjne  A,#02h,charge_3   ;is set press ,no jump
            lcall cursoron
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            lcall charge_set
            lcall cursoroff
            ljmp  charge_1
charge_3:   cjne  A,#01h,charge_1   ;is Function press ,no jump
            mov   A,#01h            ;clear LCD
            lcall lcdwi
            pop   R3
            pop   R2
            pop   R1
            pop   R0
            pop   B
            pop   A
            pop   DPL
            pop   DPH
            ret

;*********************************************************************
;charge_set : set charge of service
;*********************************************************************
charge_set: push  DPH
            push  DPL
            push  A
            push  B
            push  R0
            push  R1
            push  R2
            push  R3
            lcall cursoron
            mov   DPTR,#CHARGE_COST
            movx  A,@DPTR
            mov   R0,A
charge_s1:  lcall inkey
            push  A
            push  PSW
            mov   A,#ST_L2+5
            lcall goto
            lcall goto
            mov   DPL,R0            ;
            mov   DPH,#00h          ;
            lcall hex2dec           ;write charge cost
            mov   A,R3              ;
            lcall write2num         ;
            mov   A,#ST_L2+6
            lcall goto
            pop   PSW
            pop   A
            jz    charge_s1
            jc    charge_s1
            cjne  A,#03,charge_down ;is charge up,no jump
            inc   R0
            cjne  R0,#064h,c_no_max   ;is charge max(99),no jump
            mov   R0,#00h
c_no_max:   ljmp  charge_s1

charge_down: cjne  A,#04,charge_st ;is charge down,no jump
             dec   R0
             cjne  R0,#0ffh,c_no_min ;is charge min,no jump
             mov   R0,#63h
c_no_min:   ljmp  charge_s1
charge_st:  cjne  A,#02h,charge_Fn ;is set press,no jump
            mov   DPTR,#CHARGE_COST
            mov   A,R0
            movx  @DPTR,A
            sjmp  charge_ret
charge_Fn:  cjne  A,#01,charge_s1
charge_ret: lcall cursoroff
            pop   R3
            pop   R2
            pop   R1
            pop   R0
            pop   B
            pop   A
            pop   DPL
            pop   DPH
            ret

;*********************************************************************
;mul16  : sub to multiple 16 bit * 16 bit
;input  : R0,R1 point to data
;output : R0 point to result(32 bit)
;use    : A,B,R0,R1
;*********************************************************************
mul16       push  A
            push  B                 ;@r0 * @r1 goes to prod
            mov   fac1+0,@R0        ;R0 point mul1
            mov   fac2+0,@R1        ;R1 point mul2
            inc   r0
            inc   r1
            mov   fac1+1,@R0
            mov   fac2+1,@R1
            mov   A,fac1+0          ;16*16 multply -> 32 bit
            mov   B,fac2+0
            mul   AB                ;low * low byte
            mov   prod+0,A
            mov   prod+1,B
            mov   A,fac1+1
            mov   B,fac2+1
            mul   AB                ;high * high byte
            mov   prod+2,A
            mov   prod+3,B
            mov   A,fac1+1          ;mixed prod 1
            mov   B,fac2+0            *
            mul   AB
            add   A,prod+1          ;add 3 byte wide
            mov   prod+1,A
            mov   A,prod+2
            addc  A,B
            mov   prod+2,A
            mov   A,prod+3
            addc  A,#0
            mov   prod+3,A
            mov   A,fac1+0          ;mixed prod 2
            mov   B,fac2+1
            mul   AB
            add   A,prod+1          ;add 3 byte wide
            mov   prod+1,A
            mov   A,prod+2
            addc  A,B
            mov   prod+2,A
            mov   A,prod+3
            addc  A,#0
            mov   prod+3,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov   R0,#prod        ;result pointer in R0
            pop   B
            pop   A
            ret
;*********************************************************************
;div16  : sub to division for 32 bit / 16 bit
;input  : R0 => dividend (32 bit)
;         R1 => divisor (16 bit)
;output : R0 => quotient
;         R1 => remain
;ues    : A,R0,R1
;*********************************************************************

div16:      push  A
            mov   prod+0,@R0    ;R0 point dividend (32 bit)
            inc   R0
            mov   prod+1,@R0
            inc   R0
            mov   prod+2,@R0
            inc   R0
            mov   prod+3,@R0
            mov   divi,@R1        ;R1 point divisor (16 bit)
            inc   R1
            mov   divi+1,@R1



            mov   prod+4,#00h     ; clear extension
            mov   divcnt,#16      ; 15 normal steps, last step: remain
divlp:      acall shr5
            acall trysub          ; try and perform sub, results in cy
            acall shlq            ; shift cy into quotient
            djnz  divcnt,divlp
            mov   R0,#quot        ; quotient pointer
            mov   R1,#prod+2      ; remainder pointer
            pop   A



            ret

trysub:     clr   C               ; no borrow
            mov   A,prod+2
            subb  A,divi+0
            mov   A,prod+3
            subb  A,divi+1
            mov   A,prod+4        ;is accu extension
            subb  A,#0
            jnc   subit
            ret

shlq:       cpl   C
            mov   A,quot+0        ;use carry from before &
            rlc   A               ;shift into quotient
            mov   quot+0,A
            mov   A,quot+1
            rlc   A
            mov   quot+1,A
            ret
;
shr5:       clr   C               ;shift 5 bytes =
            mov   A,prod+0        ;product+acc 17 bit extension
            rlc   A
            mov   prod+0,A
            mov   A,prod+1
            rlc   A
            mov   prod+1,A
            mov   A,prod+2
            rlc   A
            mov   prod+2,A
            mov   A,prod+3
            rlc   A
            mov   prod+3,A
            mov   A,prod+4
            rlc   A
            mov   prod+4,A
            ret


subit:      mov   A,prod+2        ;now really subtract,
            subb  A,divi+0        ;we have no borrow from before
            mov   prod+2,A
            mov   A,prod+3
            subb  A,divi+1
            mov   prod+3,A
            mov   A,prod+4        ;is accu extension
            subb  A,#0
            mov   prod+4,A
            ret

;*********************************************************************
;chk_hook_on : check hook sw. on
;*********************************************************************
chk_hook_on:
            push DPL
            push DPH
            push A
            mov  DPTR,#U_PORT1A
            movx A,@DPTR
            jb   A.6,hook_on_ret
            mov  SP,#08h
            ljmp main
hook_on_ret:
            pop  A
            pop  DPH
            pop  DPL
            ret
;*********************************************************************
;chk_hook_off : check hook sw. on
;*********************************************************************
chk_hook_off:
            push DPL
            push DPH
            push A
            mov  DPTR,#U_PORT1A
            movx A,@DPTR
            jnb  A.6,hook_off_ret
            mov  SP,#08h
            ljmp main
hook_off_ret:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pop  A
            pop  DPH
            pop  DPL
            ret


;*********************************************************************
;hook_on :  if hook switck on it operate
;user port 1 A
;   A.3 - A.0 = data from DTMF
;   A.4  =  '1' press DTMF key
;   A.5  =  '0' ringback
;   A.6  =  '0' hook sw. on (on hand)
;           '1' hook sw. off
;   A.7  =  '1' have ringing
;*********************************************************************
hook_on:
            mov   DPTR,#U_PORT1A
            movx  A,@DPTR
            jb    A.6,to_ret          ;ckeck hook sw on ,no jump
            push  A
            pop   A
            jnb   A.7,read_dtmf       ;check ringing on,jump
            mov   DPTR,#ST_READY
            lcall lcdldp
            mov   DPTR,#U_PORT1A
wait_off:   movx  A,@DPTR             ;receive telephone only
            jnb   A.6,wait_off
to_ret:     ret

read_dtmf:  mov   A,#01               ;clear LCD
            lcall lcdwi
            mov   A,#01h              ;set cursor position
            lcall goto
            mov   A,#23h              ;write '#'
            lcall lcdwd
            mov   A,#03h              ;set cursor position
            lcall goto
            mov   R0,#NUM_BUFF
            mov   DIGIT,#00h            ;for count digit of phone num.
rep_read:   lcall DTMF_read
            jc    read_end            ;it had ring back
            mov   @R0,A
            lcall writenum
            inc   DIGIT
            inc   R0
            mov   R1,DIGIT
            cjne  R1,#09h,rep_read    ;check 9 digit max
            mov   DPTR,#U_PORT1A
wait_r:     movx  A,@DPTR
            jb    A.5,wait_r
            jb    A.4,wait_r

read_end:   lcall is_r_b
            jc    a_ringback
            mov   DPTR,#ST_BUSY
            lcall lcdldp
            mov   DPTR,#U_PORT1A
repeat:     movx  A,@DPTR             ;wait for hook sw. off
            jnb   A.6,repeat          ;ckeck hook sw off ,no jump
            ret

a_ringback: lcall time_start
            lcall time_count
            lcall find_d_locate
            jnc   f_cost
            mov   DPTR,#ST_OF_SV    ;  display out off service
            lcall lcdldp
            mov   R0,#TIME_USE
            mov   DPL,@R0
            inc   R0
            mov   DPH,@R0
            mov   R0,#ST_L2
            lcall show_ct           ; display time use
            mov   R0,#2d
wait_dis:   lcall delay1s
            djnz  R0,wait_dis
            ret
f_cost:     lcall total_cost
            lcall disp_t_cost
            ret

;*********************************************************************
;time_count : count use time & display
; out : TIME_USE keep use time(sec)
;*********************************************************************
time_count:
            mov   DPH,#00h
            mov   DPL,#00h
            mov   R0,#ST_L2
            lcall show_ct
sec_rep:    mov   R6,#81h
            lcall byterd              ;read sec
            mov   A,TEMP
            cjne  A,R7,inc_time
            sjmp  sec_rep

inc_time:   mov   TEMP,R7
            inc   DPTR
            mov   TIME_USE,DPL
            mov   TIME_USE+1,DPH
            mov   R0,#ST_L2
            lcall show_ct
            push  DPL
            push  DPH
            mov   DPTR,#U_PORT1A
            movx  A,@DPTR             ;check hook sw.off
            pop   DPH
            pop   DPL
            jnb   A.6,sec_rep
            ret

;*********************************************************************
;show_ct : display use time (HH:MM:SS)
;in : DPTR (in SEC)
;   : R0 = start position to display
;*********************************************************************
show_ct:    push  DPH
            push  DPL
            push  R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        push  R1
        mov   A,R0
        lcall goto
        mov   OPERATOR1+0,#00h
        mov   OPERATOR1+1,#00h
        mov   OPERATOR2+0,DPL
        mov   OPERATOR2+1,DPH
        mov   OPERATOR3+0,#10H     ;divided by 3600
        mov   OPERATOR3+1,#0EH     ;
        mov   R0,#OPERATOR2
        mov   R1,#OPERATOR3
        lcall div16                ;find hour
        push  R1
        mov   DPL,@R0
        inc   R0
        mov   DPH,@R0
        lcall hex2dec
        mov   A,R3
        lcall write2num          ;write hour
        mov   A,#":"
        lcall lcdwd
        pop   R1
        mov   OPERATOR2+0,@R1
        inc   R1
        mov   OPERATOR2+1,@R1
        inc   R1
        mov   OPERATOR1+0,@R1
        inc   R1
        mov   OPERATOR1+1,@R1

        mov   OPERATOR3+0,#60d     ;divided by 60
        mov   OPERATOR3+1,#00H     ;
        mov   R0,#OPERATOR2
        mov   R1,#OPERATOR3
        lcall div16                ;find min
        push  R1
        mov   DPL,@R0
        inc   R0
        mov   DPH,@R0
        lcall hex2dec
        mov   A,R3
        lcall write2num          ;write min
        mov   A,#":"
        lcall lcdwd
        pop   R1
        mov   DPL,@R1
        inc   R1
        mov   DPH,@R1
        lcall hex2dec
        mov   A,R3
        lcall write2num          ;write sec
        pop   R1
        pop   R0
        pop   DPL
        pop   DPH
        ret
;*********************************************************************
;time_start : save start time to time_buffer
;*********************************************************************
time_start:
        mov   R1,#TIME_BUFF   ;set time buffer
        mov   R6,#85h
        lcall byterd          ;    ;read hour
        mov   A,R7            ;
        anl   A,#03fh         ;
        mov   B,A             ;
        swap  A               ;
        anl   A,#0fh          ;
        mov   @R1,A           ;    save 10Hr
        inc   R1              ;
        mov   A,B             ;
        anl   A,#0fh          ;
        mov   @R1,A           ;    save 1Hr
        inc   R1              ;
        mov   R6,#83h         ;get time from RTC to time bffer
        lcall byterd          ;    ;read min
        mov   A,R7            ;
        mov   B,A             ;
        swap  A               ;
        anl   A,#0fh          ;
        mov   @R1,A           ;    save 10min
        inc   R1              ;
        mov   A,B             ;
        anl   A,#0fh          ;
        mov   @R1,A           ;    save 1min
        inc   R1              ;
        mov   R6,#81h         ;
        lcall byterd          ;    ;read sec
        mov   A,R7            ;
        mov   B,A             ;
        swap  A               ;
        anl   A,#0fh          ;
        mov   @R1,A           ;    save 10sec
        inc   R1              ;
        mov   A,B             ;
        anl   A,#0fh          ;
        mov   @R1,A           ;    save 1sec
        ret

;*********************************************************************
;DTMF_read  :  wait for DTMF key press & read value
;out : A  = DTMF value
;      CY = '1' have ringback
;*********************************************************************
DTMF_read:  push  DPL
            push  DPH
            push  R0
            push  R1
            push  R2.
            mov   R0,#2             ;n second r0 = n*10
dela1:      mov   R1,#199           ;delay 100 ms
dela2:      mov   R2,#250           ;delay 500 us
dela3:      mov   DPTR,#U_PORT1A
            movx  A,@DPTR
            lcall chk_hook_off
            jb    A.4,wait_stp1     ;check strope
            djnz  R2,dela3
            djnz  R1,dela2
            djnz  R0,dela1
stp0:       movx  A,@DPTR
            lcall chk_hook_off
            jb    A.5,stp0          ;check ringback
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            setb  C
            sjmp  read_ret
wait_stp1:  movx  A,@DPTR
            jb    A.4,wait_stp1
            lcall rtc_delay
            movx  A,@DPTR
            anl   A,#0fh
            cjne  A,#0Ah,wait_stp3
            clr   A
wait_stp3:  clr   C
read_ret:   pop   R2
            pop   R1
            pop   R0
            pop   DPH
            pop   DPL
            ret

;*********************************************************************
;is_r_b:    Detect ringback or busy
;out :  CY  = '1' is ringback
;           = '0' is busy
;*********************************************************************
is_r_b:     mov   DPTR,#U_PORT1A
wait_ring0: movx  A,@DPTR           ;
            jnb   A.5,wait_ring0

            lcall delay_rg
            mov   B,#5d
bs_rd1:     movx  A,@DPTR
            jnb   A.5,it_busy
            djnz  B,bs_rd1
            sjmp  chk_rigb_off
it_busy:    clr   C
            ret
chk_rigb_off:
            clr   U_FLAG
            mov   R0,#05d
rigb0:      mov   R1,#169
rigb1:      mov   R2,#250
rigb2:      movx  A,@DPTR
            lcall chk_hook_off
            jb    A.5,rigb3
            setb  U_FLAG
rigb3:      djnz  R2,rigb2
            djnz  R1,rigb1
            djnz  R0,rigb0
            jb    F0,chk_rigb_off
            setb  C
            ret

;*********************************************************************
;delay_rg : delay for check busy or ringback
;*********************************************************************
delay_rg:   push R0
            push R1
            push R2
            mov  R0,#6              ;n second r0 = n*10
delay_rg1:  mov  R1,#199            ;delay 100 ms
delay_rg2:  mov  R2,#250            ;delay 500 us
delay_rg3:  djnz R2,delay_rg3
            djnz R1,delay_rg2
            djnz R0,delay_rg1
            pop  R2
            pop  R1
            pop  R0
            ret
;*********************************************************************
;find_d_locate :
;in : NUM_BUFF
;out : A  = destination locate
;      CY = '0' normal
;         = '1' error not found
;*********************************************************************
find_d_locate:
            mov   R0,#NUM_BUFF
            mov   A,@R0
            cjne  A,#00h,no_0
            inc   R0
            mov   A,@R0
            cjne  A,#02h,no_02          ;check exchange 02 (BKK)
            mov   A,#00h                ;save locate in A
            clr   C
            ret

no_0:       cjne  A,#01h,no_1           ;it spacial 1xxx number
            mov   A,#0ffh
            setb  C
            ret

no_1:       mov   DPTR,#LOCATE          ;it local exchange
            movx  A,@DPTR
            clr   C
            ret

no_02:      swap  A                     ;
            mov   B,A                   ;
            inc   R0                    ;read exchange code
            mov   A,@R0                 ;
            orl   A,B                   ;
            mov   R0,A                  ;
            mov   TEMP,A                ;save exchange code in TEMP
            mov   DPTR,#local_exchange_data
rate1:      inc   DPTR
            clr   A
            movc  A,@A+DPTR
            cjne  A,#0ffh,rate2         ;is not found A = ff ,no jump
            mov   A,#0feh
            setb  C                     ;not found exchange code
            ret
rate2:      cjne  A,R0,rate1            ;is found ,no jump find next
            dec   DPL
            mov   A,DPL                 ;
            cjne  A,#0ffh,rate3         ;decrease pointer
            dec   DPH                   ;
rate3:      clr   A                     ;
            movc  A,@A+DPTR
            cjne  A,#00h,rate4           ;before it is 00h ,no jump
            lcall find_local_exchange   ;yes to find local exchange
            ret
rate4:      inc   DPTR
            ljmp  rate1                 ;jump to find next
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************************************
;find_local_exchang:
;in : DPTR = point first byte of exchange code
;   : TEMP = exchange code
;out : A  = locate code number
;    : CY = '0' normal
;         = '1' local exchange code not found
;*********************************************************************
find_local_exchange:
            mov   R0,#NUM_BUFF+3
            mov   A,@R0
            swap  A           ;
            mov   B,A         ;
            inc   R0          ;read user local exchange code
            mov   A,@R0       ;
            orl   A,B
            mov   R0,A        ;save local exchange code in R0
            inc   DPTR
lc_ex_next1:
            inc   DPTR
            push  DPH
            push  DPL
lc_ex_next2:
            inc   DPTR
            clr   A
            movc  A,@A+DPTR
            cjne  A,#00h,not_00
            inc   DPTR
            clr   A
            movc  A,@A+DPTR
            cjne  A,TEMP,not_same  ;check same exchange code,no jump
            pop   A           ;pop return stack (not use)
            pop   A           ;
            sjmp  lc_ex_next1
not_same:   pop   A           ;pop return stack (not use)
            pop   A           ;
            setb  C
            ret
not_00:     cjne  A,R0,lc_ex_next2
            pop   DPL
            pop   DPH
            clr   A
            movc  A,@A+DPTR
            clr   C
            ret
;*********************************************************************
;find_rate :  find rate ( Bath/min)
; in  : A = destination locate
;       LOCATE = machine local
; out : A  = rate
;       CY = '0' local use 3 Bath/Times
;          = '1' differance local Bath/min
;*********************************************************************
find_rate:
            mov   B,A
            mov   DPTR,#LOCATE
            movx  A,@DPTR
            cjne  A,B,no_equl
            mov   A,#00d
            clr   C           ;it local use 5 bath
            ret

no_equl:    push  A           ;
            clr   C           ;
            subb  A,B         ;
            jnc   no_carry    ; if A > B swap A,B
            pop   A           ;
            sjmp  to_find     ;
no_carry:   mov   A,B         ;
            pop   B

to_find:    inc   A
            inc   B
            mov   R0,A        ;here A < B ( A = row,B = col)
            mov   R1,B
            mov   A,#147d
            clr   C
            subb  A,R0        ; 147d - row     ;
            mov   B,R0        ;                ;
            mul   AB          ; mul by row     ;
            mov   R0,A                         ;  find address in rate_data
            mov   A,B         ;                ;  row(147-row)/2 -73 + col
            clr   C           ;                ;
            rrc   A           ; divided by 2   ;
            mov   B,A         ;                ;
            mov   A,R0        ;                ;
            rrc   A           ;                ;
            add   A,R1                         ;
            jnc   not_carry1  ; add by col     ;
            inc   B           ;                ;
not_carry1: clr   C                            ;
            subb  A,#73d      ;                ;
            jnc   not_carry2  ; sub by 73d     ;
            dec   B           ;                ;
not_carry2:
            mov   DPTR,#RATE_DATA
            add   A,DPL
            mov   DPL,A
            mov   A,B
            addc  A,DPH
            mov   DPH,A
            clr   A
            movc  A,@A+DPTR
            setb  C
            ret

;*********************************************************************
;total_cost :  find total cost
; in  : A = destination locate
;       LOCATE = machine local
;       time in TIME_BUFF
;       TIME_USE use time
; out : TT_COST  = total cost (2 byte)
;*********************************************************************
total_cost
            lcall find_rate
            jc    diff_local
            mov   TT_COST,#05h
            mov   TT_COST+1,#00h
            ret
diff_local: push  A
            mov   R0,#TIME_USE      ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                mov   DPL,@R0             ;save use time to DPTR
                inc   R0                  ;
                mov   DPH,@R0             ;
                inc   DPTR                  ;
                inc   DPTR                  ;increase 4 sec for
                inc   DPTR                  ;check ringback
                inc   DPTR                  ;
                mov   OPERATOR1+0,#00h
                mov   OPERATOR1+1,#00h
                mov   OPERATOR2+0,DPL
                mov   OPERATOR2+1,DPH
                mov   OPERATOR3+0,#60d    ;divided by 3600
                mov   OPERATOR3+1,#00h    ;
                mov   R0,#OPERATOR2
                mov   R1,#OPERATOR3
                lcall div16               ;calculate to min.
                mov   DPL,@R0
                inc   R0
                mov   DPH,@R0
                mov   A,@R1
                jz    no_incr1
                inc   DPTR
no_incr1:       pop   A
                mov   OPERATOR2+0,DPL     ;
                mov   OPERATOR2+1,DPH     ;
                mov   OPERATOR3+0,A       ;mul by rate
                mov   OPERATOR3+1,#00h    ;
                mov   R0,#OPERATOR2       ;
                mov   R1,#OPERATOR3       ;
                lcall mul16
                mov   OPERATOR2+0,@R0
                inc   R0
                mov   OPERATOR2+1,@R0
                inc   R0
                mov   OPERATOR1+0,@R0
                inc   R0
                mov   OPERATOR1+1,@R0
                mov   R0,#OPERATOR2

                mov   A,TIME_BUFF         ;check time rate
                swap  A
                mov   B,A
                mov   A,TIME_BUFF+1
                orl   A,B
                cjne  A,#00h,next1
                ljmp  rate03
next1:          cjne  A,#01h,next2
                ljmp  rate03
next2:          cjne  A,#02h,next3
                ljmp  rate03
next3:          cjne  A,#03h,next4
                ljmp  rate03
next4:          cjne  A,#04h,next5
                ljmp  rate03
next5:          cjne  A,#05h,next6
                ljmp  rate03
next6:          cjne  A,#06h,next7
                ljmp  rate03
next7:          cjne  A,#07h,next8
                ljmp  rate01
next8:          cjne  A,#08h,next9
                ljmp  rate01
next9:          cjne  A,#09h,next10
                ljmp  rate01
next10:         cjne  A,#10h,next11
                ljmp  rate01
next11:         cjne  A,#11h,next12
                ljmp  rate01
next12:         cjne  A,#12h,next13
                ljmp  rate01
next13:         cjne  A,#13h,next14
                ljmp  rate01
next14:         cjne  A,#14h,next15
                ljmp  rate01
next15:         cjne  A,#15h,next16
                ljmp  rate01
next16:         cjne  A,#16h,next17
                ljmp  rate01
next17:         cjne  A,#17h,next18
                ljmp  rate01
next18:         cjne  A,#18h,next19
                ljmp  rate02
next19:         cjne  A,#19h,next20
                ljmp  rate02
next20:         cjne  A,#20h,next21
                ljmp  rate02
next21:         cjne  A,#21h,rate03
                ljmp  rate02
rate03:         mov   OPERATOR3+0,#03d
                ljmp  to_divi

rate01:         mov   DPL,OPERATOR2+0
                mov   DPH,OPERATOR2+1
                ljmp  no_incr2

rate02:         mov   OPERATOR3+0,#02d
to_divi:        mov   OPERATOR3+1,#00h
                mov   R1,#OPERATOR3

                lcall div16
                mov   DPL,@R0
                inc   R0
                mov   DPH,@R0
                mov   A,@R1
                jz    no_incr2
                inc   DPTR
no_incr2:       push  DPH
                push  DPL
                mov   DPTR,#CHARGE_COST
                movx  A,@DPTR
                pop   DPL
                pop   DPH
                jz    chg_is0

charge_rep:     inc   DPTR
                djnz  A,charge_rep
chg_is0:        mov   TT_COST+0,DPL
                mov   TT_COST+1,DPH
                ret
;*****************************************************************
;disp_t_cost :  display total cost & save last number & cost
;in  : TT_COST = total cost
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************************************
disp_t_cost:
            mov   A,#16d            ;
            lcall goto              ;
            mov   DPTR,#ST_TOTAL    ;
            mov   B,#15d            ;
write_loop: clr   A                 ;write ST_TOTAL string
            movc  A,@A+DPTR         ;
            lcall lcdwd             ;
            inc   DPTR              ;
            djnz  B,write_loop      ;
            mov   A,#20d
            lcall goto
            mov   DPL,TT_COST
            mov   DPH,TT_COST+1
            lcall hex2dec
            mov   A,R1              ;
            lcall write2num         ;
            mov   A,R2              ;display total cost
            lcall write2num         ;
            mov   A,R3              ;
            lcall write2num         ;
            mov   A,#05d
d5sec:      lcall delay1s
            djnz  A,d5sec

            mov   R0,#NUM_BUFF
            mov   DPTR,#LAST_NUM
sav_rep:    mov   A,@R0                     ;
            movx  @DPTR,A                   ;save last phone number
            inc   R0                        ;
            inc   DPTR                      ;
            cjne  R0,#NUM_BUFF+09h,sav_rep  ;
            mov   DPTR,#N_DIGIT
            mov   A,DIGIT
            movx  @DPTR,A
            mov   DPTR,#LAST_COST           ;
            mov   A,TT_COST+0               ;
            movx  @DPTR,A                   ;save last total cost
            inc   DPTR                      ;
            mov   A,TT_COST+1               ;
            movx  @DPTR,A
            mov   A,#01h                    ;
            lcall lcdwi                     ;clear LCD
            ret


;USER DEFINE BYTE
;*****************************************************
ST_READY:     DFB "     READY      "
              DFB "                "
ST_OF_SV:     DFB " OUT OF SERVICE "
              DFB "                "
ST_BUSY       DFB "      BUSY      "
              DFB "                "
CHARGE_TITLE: DFB "CHARGE SET      "
              DFB "                "
LOCATE_TITLE  DFB "LOCATION SET    "
              DFB "                "
DATE_TITLE:   DFB "DATE SET        "
              DFB "                "
TIME_TITLE:   DFB "TIME SET        "
              DFB "                "
ST_LAST       DFB " #              "
              DFB " $         Bath "
ST_TOTAL      DFB " $         Bath "
st_start1:    DFB "     KMITL      "
              DFB " ED.ENGINEER 16 "
st_start2:    DFB "    TELECOM     "
              DFB "    TWILIGHT    "

LOCATE_DATA:
            DFB  "BKK"
            DFB  "PBIRBRPKN"
            DFB  "NPTSKNSKMKRI"
            DFB  "AYASPBATG"
            DFB  "SRILRISBR"
            DFB  "PRISKWNYK"
            DFB  "CBIRYGCCO"
            DFB  "CTITRT"
            DFB  "UDNNGPNKINPMMDHSNKLEI"
            DFB  "KKNMKMRETKSN"
            DFB  "NMACPMBRMSRN"
            DFB  "UBNACRYSTSSK"
            DFB  "CMILPNMSNCRI"
            DFB  "LPGPRENANPYO"
            DFB  "PLKUTTSTIKPTTAK"
            DFB  "NSNUGICNTPBNPCT"
            DFB  "YLAPTNNWT"
            DFB  "SKASTNPLG"
            DFB  "NRTTRGKBI"
            DFB  "PKTPNA"
            DFB  "SNIRNGCPN"


local_exchange_data:
  ;032
        DFB  0h,32h,01h,41h,42h,43h,46h,47h,48h,49h
        DFB  0h,32h,02h,20h,21h,22h,23h,24h,25h,26h,27h,28h,29h,31h,32h
             DFB 33h,34h,38h,39h
        DFB  0h,32h,03h,51h,52h,53h,54h,60h,61h,62h,63h,66h,67h,68h,69h
  ;034
        DFB  0h,34h,04h,21h,22h,23h,24h,25h,26h,31h,32h,33h,35h,38h,39h
        DFB  0h,34h,05h,41h,42h,47h,48h
        DFB  0h,34h,06h,71h,72h,75h,76h,81h,82h,83h,84h
        DFB  0h,34h,07h,51h,54h,56h,57h,58h,59h,61h,62h
  ;035
        DFB  0h,35h,08h,20h,21h,22h,24h,25h,26h,27h,31h,33h,34h,35h,36h,37h
             DFB 38h,39h
        DFB  0h,35h,09h,50h,51h,52h,53h,54h,55h,56h,57h,58h,59h
        DFB  0h,35h,0Ah,61h,62h,63h,66h,69h
  ;036
        DFB  0h,36h,0Bh,20h,21h,22h,24h,25h,26h,28h,31h,34h,35h,37h,39h
        DFB  0h,36h,0Ch,41h,42h,43h,44h,46h,47h,48h,49h,61h
        DFB  0h,36h,0Dh,51h,52h,58h,59h
  ;037
        DFB  0h,37h,0eh,20h,21h,23h
        DFB  0h,37h,0fh,25h
        DFB  0h,37h,10h,32h,38h,39h
  ;038
        DFB  0h,38h,11h,26h,27h,28h,30h,31h,32h,33h,34h,35h,36h,37h,38h,39h,40h
```

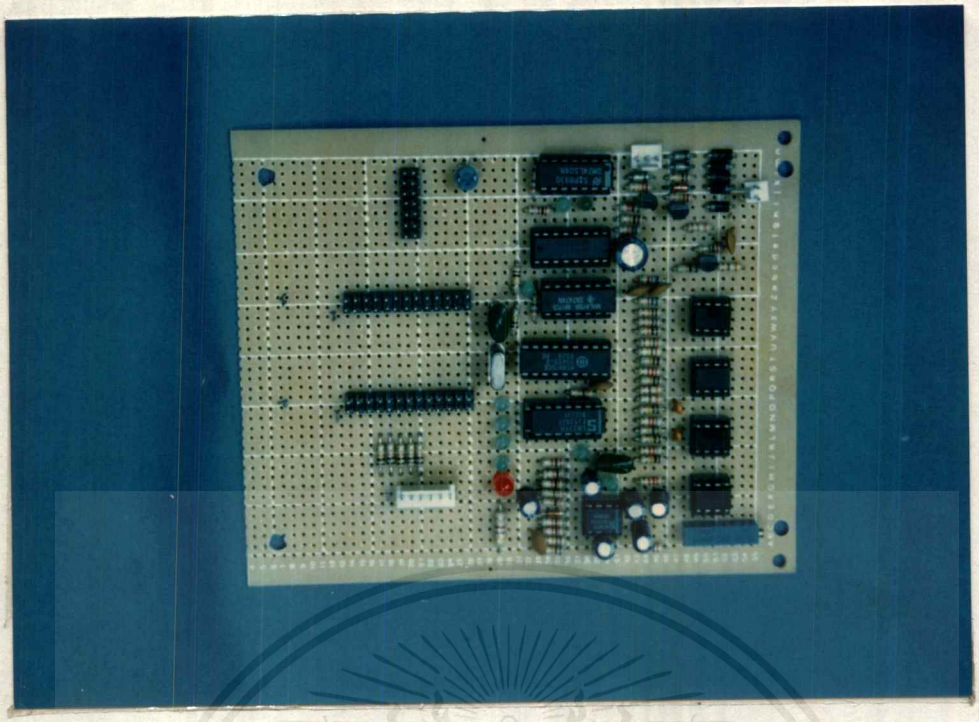
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DFB 41h,42h,43h,44h,45h,46h,47h,48h,49h,71h,72h,73h,76h,77h,78h,79h
        DFB  0h,38h,12h,60h,61h,62h,63h,64h,65h,66h,67h,68h,69h,80h,86h,87h,88h
        DFB  0h,38h,13h,51h,53h,54h,55h,57h,58h,59h,81h,82h,83h
;039
        DFB  0h,39h,14h,31h,32h,33h,34h,35h,37h,38h,39h,43h,44h
        DFB  0h,39h,15h,51h,52h,53h,58h,59h
;042
        DFB  0h,42h,16h,22h,24h,25h,26h,27h,28h,32h,33h,34h,39h
        DFB  0h,42h,17h,31h,35h,38h
        DFB  0h,42h,18h,41h,42h,43h,44h,45h,46h,47h,48h,49h
        DFB  0h,42h,19h,51h,52h,54h,57h,58h,59h
        DFB  0h,42h,1ah,61h,63h,68h,69h
        DFB  0h,42h,1bh,71h,72h,73h,75h,76h,77h,78h,79h
        DFB  0h,42h,1ch,81h,82h,83h,84h,87h,88h,89h
;043
        DFB  0h,43h,1dh,22h,23h,24h,25h,26h,27h,28h,29h,31h,32h,33h,34h
             DFB 39h,41h,43h,44h,49h
        DFB  0h,43h,1eh,71h,72h,76h,77h,78h,79h
        DFB  0h,43h,1fh,51h,52h,55h,56h,57h,58h,59h
        DFB  0h,43h,20h,81h,82h,85h,86h,87h,88h,89h
;044
        DFB  0h,44h,21h,21h,23h,24h,25h,26h,27h,31h,36h,37h,38h,39h,41h
             DFB 44h,45h,46h,47h,49h
        DFB  0h,44h,22h,81h,82h,85h,86h,87h,88h,89h
        DFB  0h,44h,23h,61h,62h,63h,64h,66h,67h,68h,69h
        DFB  0h,44h,24h,51h,52h,54h,55h,56h,57h,58h,59h
;045
        DFB  0h,45h,25h,24h,25h,26h,31h,32h,35h,36h,37h,39h,44h,48h,49h
        DFB  0h,45h,26h,45h,46h
        DFB  0h,45h,27h,71h,72h,78h,79h
        DFB  0h,45h,28h,61h,62h,65h,66h,67h,68h,69h
;053
        DFB  0h,53h,29h,20h,21h,22h,23h,24h,25h,26h,27h,28h,29h,30h,31h
             DFB 32h,33h,34h,38h,39h,40h,41h,42h,43h,44h,45h,46h,47h,48h
             DFB 49h,80h,81h,82h,83h,85h,86h,87h,89h
        DFB  0h,53h,2ah,51h,52h,53h,54h,55h,56h,57h,58h,59h
        DFB  0h,53h,2bh,61h,62h,68h,69h
        DFB  0h,53h,2ch,64h,65h,66h,70h,71h,72h,73h,74h,75h,76h,77h,78h,79h
;054
        DFB  0h,54h,2dh,21h,25h,26h,27h,28h,29h,31h,32h,33h
        DFB  0h,54h,2eh,51h,52h,54h,58h,59h,61h,62h,63h
        DFB  0h,54h,2fh,71h,75h,76h,77h,78h,79h
        DFB  0h,54h,30h,41h,42h,43h,45h,48h,49h
;055
        DFB  0h,55h,31h,21h,22h,24h,25h,26h,31h,36h,37h,38h,39h
        DFB  0h,55h,32h,41h,42h,43h,44h,48h,49h
        DFB  0h,55h,33h,61h,62h,64h,65h,67h,68h,69h
        DFB  0h,55h,34h,71h,72h,76h,77h,78h,79h
        DFB  0h,55h,35h,51h,53h,54h,56h,58h,59h
;056
        DFB  0h,56h,36h,21h,22h,23h,24h,25h,26h,27h,28h,29h,31h,34h
        DFB  0h,56h,37h,51h,52h,53h,59h
        DFB  0h,56h,38h,41h,42h,45h,46h,48h,49h
        DFB  0h,56h,39h,70h,71h,72h,73h,74h,76h,77h,78h,79h
        DFB  0h,56h,3ah,61h,62h,63h,64h,65h,68h,69h
;073
        DFB  0h,73h,3bh,21h,22h,23h,24h,29h
        DFB  0h,73h,3ch,31h,33h,34h,41h,43h,48h,49h
        DFB  0h,73h,3dh,51h,52h,56h,57h,58h,59h,61h,65h,67h
;074
        DFB  0h,74h,3eh,21h,22h,23h,24h,25h,29h,31h,32h,33h,34h,35h,36h
             DFB 37h,38h,39h,41h,42h,43h,44h,52h,54h
        DFB  0h,74h,3fh,71h,72h,73h,78h,79h
        DFB  0h,74h,40h,61h,62h,68h,69h
;075
        DFB  0h,75h,41h,31h,32h,33h,34h,35h,38h,39h,41h,42h,44h,48h,49h
             DFB 51h,52h,53h
        DFB  0h,75h,42h,21h,22h,23h,24h,25h,27h,28h,29h
        DFB  0h,75h,43h,61h,62h,63h,68h,69h
;076
        DFB  0h,76h,44h,20h,21h,22h,23h,24h,25h,27h,28h,29h,31h,32h,33h
             DFB 34h,38h,39h
        DFB  0h,76h,45h,41h,42h,43h,49h,57h,58h,59h
;077
        DFB  0h,77h,46h,20h,21h,22h,23h,24h,25h,27h,28h,31h,34h,35h,36h
             DFB 37h,38h,39h,42h
        DFB  0h,77h,47h,81h,82h,83h,89h,91h
        DFB  0h,77h,48h,41h,42h,43h,50h,51h,52h,53h,54h,55h,57h,58h,59h

        DFB  0h,0FFh

RATE_DATA: DFB 0
  DFB 0,6,6,12,3,3,6,9,6,6,6,6,9,9,6,9,6,6,9,6,12,12,15,15,18,18,18,18,15
      DFB 15,15,15,15,12,12,12,12,15,15,15,15,18,18,18,18,18,15,18,18,12
      DFB 15,15,12,15,12,9,9,12,12,18,18,18,18,18,18,18,18,18,18,18,15,15
  DFB 0,3,9,6,6,3,9,9,9,9,9,12,12,9,12,9,9,9,9,12,12,18,18,18,18,18,18,18
      DFB 15,15,18,18,12,15,15,15,18,18,18,18,18,18,18,18,18,18,18,18,15
      DFB 18,15,15,15,12,12,12,15,15,18,18,18,18,18,18,18,18,18,18,15
      DFB 15,12
  DFB 0,9,3,3,3,6,9,9,9,9,9,9,9,12,9,9,9,9,12,12,18,15,18,18,18,18,15,15
      DFB 15,15,18,12,12,15,15,18,18,18,18,18,18,18,18,18,18,18,15,15
      DFB 15,12,15,12,12,9,15,12,18,18,18,18,18,18,18,18,18,18,18,15,15,12
  DFB 0,12,9,9,12,12,12,12,12,12,12,12,12,12,12,9,12,12,12,18,18,18,18,18
      DFB 18,18,18,18,18,18,15,18,18,18,18,18,18,18,18,18,18,18,18,18,18
      DFB 18,18,18,18,18,18,15,15,15,18,18,18,18,18,18,18,15,15,15,15,15
      DFB 15,12,12,9
  DFB 0,3,3,6,6,6,6,9,9,9,9,12,9,9,9,9,12,12,15,15,18,18,18,18,15,15,15,15
      DFB 15,12,12,15,15,18,18,15,15,18,18,18,18,15,15,18,18,12,15,15,12
      DFB 15,12,9,9,12,12,18,18,18,18,18,18,18,18,18,18,18,18,15,15
  DFB 0,3,6,6,9,9,9,9,9,9,9,9,6,9,6,12,12,18,15,18,18,18,18,15,15,15,15,15,12
      DFB 12,12,15,18,18,15,15,18,18,18,18,18,18,18,18,15,15,15,12,15,12
      DFB 12,9,12,12,18,18,18,18,18,18,18,18,18,18,18,18,15,15,15
  DFB 0,6,9,9,9,9,9,9,9,12,9,9,9,9,12,12,18,15,18,18,18,18,18,15,15,15,15,18
      DFB 12,12,15,15,18,18,15,18,18,18,18,18,18,18,18,18,15,15,15,12,15
      DFB 12,12,9,15,12,18,18,18,18,18,18,18,18,18,18,18,18,15,15,12
  DFB 0,9,6,9,9,9,9,9,12,9,9,12,9,12,15,18,15,18,18,18,18,18,15,15,15,15,15,12
      DFB 12,15,15,18,18,18,18,18,18,18,18,18,15,15,18,18,12,15,12,12,12,9,9,9
      DFB 12,12,18,18,18,18,18,18,18,18,18,18,18,18,18,15,15
  DFB 0,3,3,3,3,6,6,9,6,9,9,6,12,12,15,15,15,18,18,15,15,12,15,15,15,9,12,12
      DFB 12,15,15,15,15,18,15,18,18,15,15,15,18,12,15,12,12,12,9,9,9,12,12
      DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,15
  DFB 0,3,6,6,6,9,12,9,9,12,9,12,15,15,15,15,18,18,18,15,15,15,15,15,12,12
      DFB 12,15,18,18,15,15,15,15,18,18,15,15,15,18,12,12,12,12,12,9,6,6,12
      DFB 12,18,18,18,18,18,18,18,18,18,18,18,18,18,18,15
  DFB 0,3,3,3,9,9,9,6,9,12,9,12,12,15,15,15,18,18,15,12,12,15,15,15,9,12,12,12
      DFB 15,15,15,15,15,15,15,18,18,15,15,15,18,12,12,12,12,12,12,9,6,6,12,12,18
      DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,15
  DFB 0,3,6,6,9,3,3,9,12,6,12,12,15,12,15,18,15,15,12,12,12,12,12,9,9,12,12,15
      DFB 15,15,15,18,15,18,18,15,15,15,15,18,12,15,12,12,12,9,9,9,12,12,18,18
      DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,15
  DFB 0,3,9,9,6,9,12,9,12,12,15,12,15,18,15,15,12,12,12,12,12,12,9,6,6,9,9,18,18,18
      DFB 15,15,15,15,15,15,18,18,15,15,15,15,12,12,12,12,12,12,9,6,6,9,9,18,18,18
```

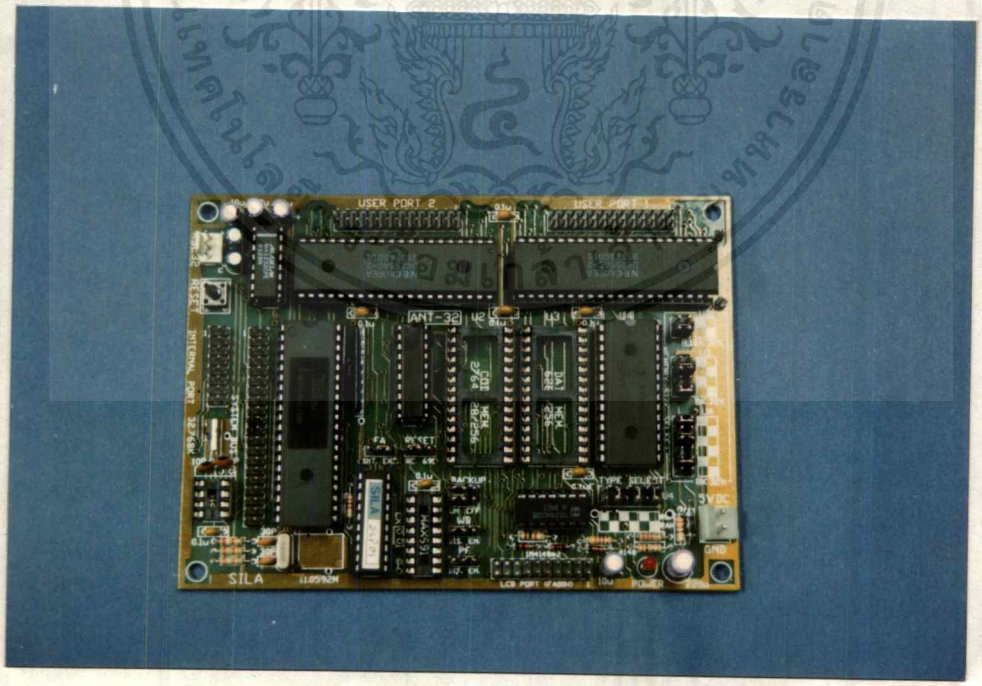
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DFB 18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,9,12,9,9,12,9,12,15,15,12,15,18,15,15,12,12,12,15,15,9,12,12,12,15
        DFB 15,15,15,15,15,18,18,15,15,15,15,12,12,12,9,12,6,6,3,9,9,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,3,6,9,6,9,12,15,15,15,18,18,15,15,15,12,12,12,12,12,9,12,12,12,15,15,15
        DFB 12,18,18,18,18,18,18,15,18,18,18,12,15,15,12,15,12,12,9,12,12,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,15
    DFB 0,9,9,9,9,9,9,9,15,15,15,15,15,15,15,15,12,12,12,12,12,9,12,9,9,9,12,15,12,12,18
        DFB 18,18,18,18,18,18,18,18,15,15,15,15,15,12,12,12,12,12,12,18,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,15
    DFB 0,6,9,9,6,12,12,15,15,15,18,15,15,15,15,12,12,12,12,12,9,9,12,12,15,15,15,12
        DFB 18,18,18,18,18,15,15,18,18,18,12,15,12,12,15,12,9,9,9,12,12,18,18,18,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,15
    DFB 0,6,3,9,12,15,15,18,18,18,18,18,15,15,15,15,15,12,12,12,12,15,15,15,15,15,18
        DFB 18,18,18,18,18,18,18,18,15,15,15,15,15,15,12,12,12,12,12,12,18,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,15,15,15
    DFB 0,9,6,9,18,18,18,18,18,18,18,18,15,15,15,15,15,12,15,12,12,15,18,18,15,18,18
        DFB 18,18,18,18,18,18,18,18,15,18,18,15,18,15,12,12,15,15,18,18,18,18,18,18,18
        DFB 15,18,18,18,18,15,15,15,12
    DFB 0,9,12,15,15,15,15,18,18,18,18,15,15,15,15,15,9,12,12,12,15,15,18,15,18,18
        DFB 18,18,18,18,18,18,18,18,15,15,15,12,15,12,12,9,9,12,12,18,18,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,15,15
    DFB 0,6,18,18,18,18,18,18,18,18,15,15,15,15,15,12,15,12,12,15,15,15,15,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,15,15,15,15,15,15,18,18,18,18,18,18,18
        DFB 18,18,18,18,18,15,15,15
    DFB 0,18,18,18,18,18,18,18,18,15,15,15,15,15,12,15,15,12,12,12,15,15,15,15,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,18,15,15,15,15,15,15,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18,15,15,15
    DFB 0,3,6,9,9,9,9,9,9,9,9,12,9,12,12,12,12,12,12,15,15,18,15,15,12,12,15
        DFB 12,12,12,15,15,12,15,15,12,12,18,18,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18,18
    DFB 0,6,9,9,9,9,6,6,9,9,9,12,9,12,12,12,12,12,12,15,15,18,15,12,12,12,12,12
        DFB 12,12,12,15,12,12,12,9,12,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,9,9,9,9,9,9,12,9,12,12,12,12,15,12,12,12,15,15,18,15,12,12,12,12,12
        DFB 12,12,15,15,15,15,15,12,12,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18
    DFB 0,6,6,9,12,12,9,9,15,12,12,12,12,9,9,12,18,18,18,18,18,15,15,18,15,15
        DFB 18,18,18,18,18,18,18,15,15,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,9,9,9,9,9,9,12,12,12,12,9,6,9,9,18,18,18,18,18,18,15,18,18,18
        DFB 18,15,18,18,15,15,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,9,9,9,9,9,9,12,12,12,12,12,9,9,12,18,18,18,18,18,15,15,15,15,15,15,15,15
        DFB 18,15,15,15,12,15,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,9,12,12,12,12,9,12,12,15,15,12,15,12,15,12,12,15,12,12,9,9,12,9,9,12,12,12
        DFB 12,12,12,9,9,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,6,6,9,9,9,9,9,12,12,9,12,15,15,18,15,15,12,12,15,12,12,12,15,15,12
        DFB 12,12,9,12,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,3,3,9,9,9,9,9,9,9,6,9,18,18,18,18,18,15,15,15,15,15,12,15,15,15,15,15,15,15
        DFB 12,12,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,3,12,9,9,9,9,9,9,6,9,18,18,18,18,18,18,15,15,18,15,15,15,15,15,15,15,15
        DFB 12,15,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,12,9,9,9,9,9,9,6,9,18,18,18,18,18,15,15,15,15,12,15,15,15,15,15,15,15,12
        DFB 12,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,9,9,12,12,12,12,18,18,18,18,18,15,15,15,15,18,12,15,12,12,15,12,12,12,9
        DFB 12,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,9,9,12,12,12,12,15,15,18,18,15,12,15,15,12,12,12,12,12,12,12,12,9
        DFB 9,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,3,9,9,9,9,9,18,18,18,18,18,18,15,15,18,15,15,15,15,15,12,12,12,12,12,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,9,9,9,6,18,18,18,18,18,18,18,18,18,15,15,15,15,15,15,15,12,15,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,6,6,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,15,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,6,18,18,18,18,18,18,18,18,18,15,18,18,18,18,15,15,15,15,15,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,18,18,18,18,18,18,18,18,15,15,15,15,15,15,18,15,15,15,12,15,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,3,9,9,9,6,9,9,9,12,9,12,12,12,15,15,15,15,12,18,18,18,18,18,18,18,18,18
        DFB 18,18,18,18,18,18,18,18
    DFB 0,9,9,6,9,9,9,9,12,9,9,12,9,12,15,15,12,12,18,18,18,18,18,18,18,18,18,18
        DFB 18,18,18,18
    DFB 0,9,9,12,12,12,15,12,12,15,12,15,15,18,15,15,18,18,18,18,18,18,18,18,18,18
        DFB 18,18,18,18,18,18,18
    DFB 0,9,9,9,6,12,12,12,15,12,15,18,18,15,15,18,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18,18,18,18
    DFB 0,6,9,9,9,6,9,9,9,12,12,12,12,12,18,18,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18,18
    DFB 0,6,9,9,6,9,9,9,12,12,12,12,9,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18
    DFB 0,9,12,9,12,12,12,12,15,15,12,12,18,18,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18,18
    DFB 0,12,9,12,12,12,15,15,15,12,12,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
        DFB 18
    DFB 0,6,6,6,9,9,9,9,9,3,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,9,9,12,12,12,9,9,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,6,9,9,12,9,6,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,9,9,9,9,6,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,9,9,12,12,9,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,3,6,9,6,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,3,9,9,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,9,9,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,6,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,18,18,18,18,18,18,18,18,18,18,18,18,18,18,18
    DFB 0,3,6,9,9,9,12,12,12,15,15,15,15,15
    DFB 0,6,6,9,9,12,9,12,12,12,12,15,15
    DFB 0,9,9,12,12,12,15,15,15,15,18,18
    DFB 0,6,6,9,9,12,12,12,12,15,15
    DFB 0,9,9,9,12,12,12,12,15,15
    DFB 0,6,6,9,9,9,9,12,12
    DFB 0,6,9,9,9,9,12,12
    DFB 0,6,9,9,9,12,12
    DFB 0,6,6,9,12,12
    DFB 0,6,9,12,12
    DFB 0,9,9,12
    DFB 0,9,9
    DFB 0,6
    DFB 0

show:   push A
        lcall write2num
        lcall cursoron
sdd:    lcall inkey
        jc    sdd
        jz    sdd
        cjne  A,#01h,sdd
        lcall cursoroff
        pop  a
        ret

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

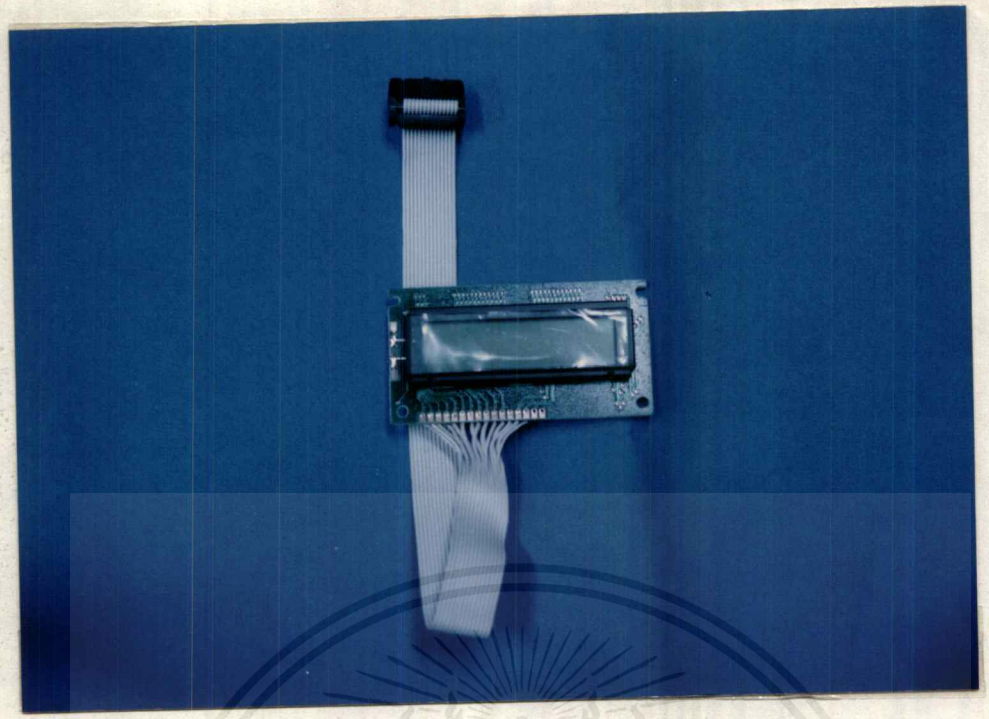


รูปที่ 1 แผ่นวงจรรวม



รูปที่ 2 บอร์ดไมโครคอนโทรลเลอร์ ANT-32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

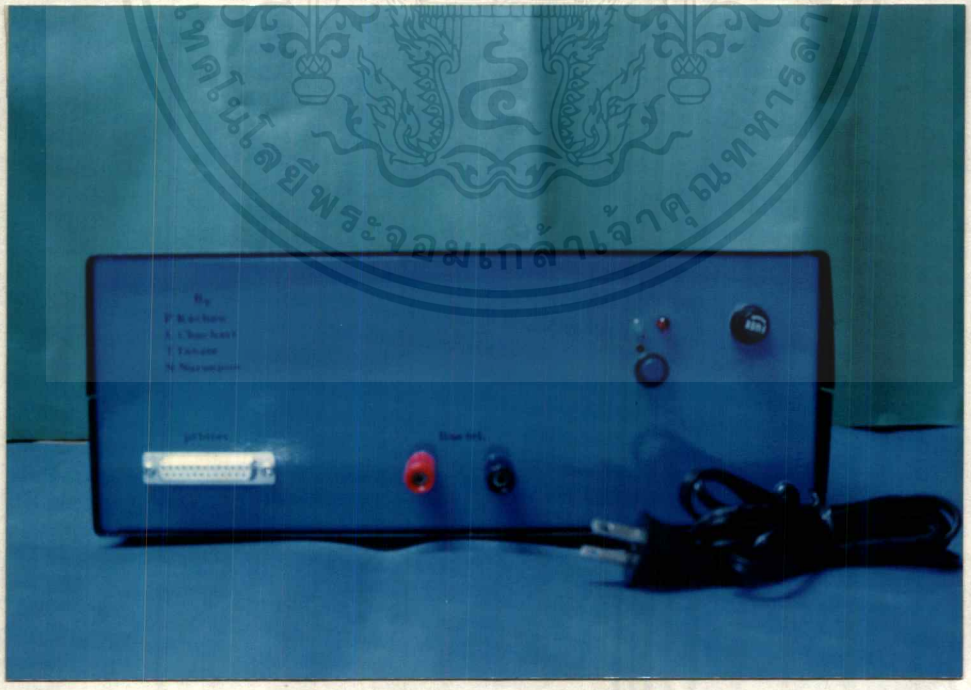


รูปที่ 3 จอ LCD

รูปที่ 4 DMC-162

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5 ด้านหน้าเครื่อง



รูปที่ 6 ด้านหลังเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รายการอุปกรณ์เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์ของวงจรตรวจสอบการยกหูโทรศัพท์

อุปกรณ์	ค่า/เบอร์	จำนวน
R_1,R_2	200 kΩ	2 ตัว
R_3	10 kΩ	1 ตัว
Z_1	1N4748A	1 ตัว
Z_2	1N4759A	1 ตัว
Z_3	1N4742A	1 ตัว
D1-D4	1N4001	4 ตัว
TR	2SC458	1 ตัว
IC	74LS14	1 ตัว

รายการอุปกรณ์ของวงจรตรวจสัญญาณกริ่งเรียก

อุปกรณ์	ค่า/เบอร์	จำนวน
R_1	100 kΩ	1 ตัว
R2	39 kΩ	1 ตัว
R_3	10 kΩ	1 ตัว
R4	180 kΩ	1 ตัว
C1	220 uF	1 ตัว
TR	2SC458	1 ตัว
IC_1	74LS14	1 ตัว
IC_2	74LS123	1 ตัว
IC_3	74LS74	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ของวงจรจ่ายไฟชนิดมีแบตเตอร์สำรอง

อุปกรณ์	ค่า/เบอร์	จำนวน
R1	36 kΩ	1 ตัว
R2	47 kΩ	1 ตัว
R3,R5	33 kΩ	2 ตัว
R4	5.6 kΩ	1 ตัว
R6	27 kΩ	1 ตัว
R7	22 kΩ	1 ตัว
R8,R9	1 kΩ	2 ตัว
C1,C3,C6.C8	1000 uF	4 ตัว
C2	22 uF	1 ตัว
C4,C7,C9	0.1 uF	1 ตัว
C5	2.2 uF	1 ตัว
D1-D4,D7	1N4001	4 ตัว
D5,D6,D8,D9	1N4005	4 ตัว
TR	2SC485	1 ตัว
IC_1-IC_4	LM7805	4 ตัว
SCR	TIC186	1 ตัว
LED	สีเขียว	2 ตัว
BATTERY	12 V, 6.5 A	1 ตัว
หม้อแปลง	9,12 V	1 ตัว
S1,S2	กดติดปล่อยดับ	2 ตัว
IC555	LM555	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์ของวงจรตรวจแยกสัญญาณโทน

อุปกรณ์	ค่า/เบอร์	จำนวน
R_1	10 kΩ	1 ตัว
R_2	20 kΩ	1 ตัว
R3	1 kΩ	1 ตัว
R4	200 Ω	1 ตัว
R5,R7,R8	2 kΩ	3 ตัว
R6	2.2 kΩ	1 ตัว
R9	1 MΩ	1 ตัว
VR	2.7 kΩ	1 ตัว
C_1	0.01 uF	1 ตัว
C_2	0.1 uF	1 ตัว
C3	1.7 uF	1 ตัว
C4	1 uF	1 ตัว
C5	0.1 nF	1 ตัว
IC_1-IC4	LM741	4 ตัว
IC5	LM567	1 ตัว
IC6	74LS123	1 ตัว
D	1N4148	1 ตัว
LED	สีเขียว	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์อื่นๆ

อุปกรณ์	ค่า/เบอร์	จำนวน
SWITCH	SPST	6 ตัว
LINE	AC. 220 Vac	1 เส้น
ตะกั่ว	60/40 %	1 ม้วน
กล่องเหล็ก	สีดำ	1 ใบ
CENTROPORT	25 PIN	1 ตัว
JACK		1 ตัว
กระบอก FUSE	3 A	1 ตัว
FUSE	3 A	1 ตัว
ที่รองปริ้นท์		
SOCKET IC 18 ขา		1 ตัว
SOCKET IC 16 ขา		1 ตัว
SOCKET IC 14 ขา		3 ตัว
SOCKET IC 8 ขา		5 ตัว
ANT-32	V.3	1 บอร์ด
LCD		1 ชุด
HEAT SINK		5 แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวัด	รหัสทางไกล	หมายเลขตั้งต้น
39. เชียงราย	(053)	
	แม่สาย	73-XXXX
	พาน	72-XXXX
	แม่จัน	77-XXXX
40. ลำปาง	(054)	22-XXXX
		21-XXXX
41. พเยา	(054)	48-XXXX
		43-XXXX
42. แพร่	(054)	52-XXXX
		51-XXXX
		54-XXXX
	เด่นชัย	61-XXXX
43. น่าน	(054)	77-XXXX
		71-XXXX

เขตโทรศัพท์ภูมิภาคที่ 6

จังหวัด	รหัสทางไกล	หมายเลขตั้งต้น
44. เพชรบุรี	(032)	42-XXXX
		33-XXXX
	ท่ายาง	46-XXXX
45. ประจวบคีรีขันธ์	(032)	61-XXXX
		60-XXXX
	หัวหิน	51-XXXX
	ปราณบุรี	62-XXXX
	ทับสะแก	67-XXXX
46. ราชบุรี	(032)	32-XXXX
		33-XXXX
	บ้านโป่ง	21-XXXX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวัด	รหัสทางไกล	หมายเลขตั้งต้น
46. ราชบุรี	(032)	
	โพธาราม	23-XXXX
	ดำเนินสะดวก	25-XXXX
		24-XXXX
48. นครปฐม	(034)	25-XXXX
		24-XXXX
	สามพราน	31-XXXX
		32-XXXX
	นครชัยศรี	33-XXXX
		25-XXXX
49. สมุทรสาคร	(034)	42-XXXX
		41-XXXX
	กระทุ่มแบน	47-XXXX
50. สมุทรสงคราม	(034)	71-XXXX
51. กาญจนบุรี	(034)	51-XXXX
	ท่าม่วง	61-XXXX
	ท่ามะกา	54-XXXX

เขตโทรศัพท์ภูมิภาคที่ 7

จังหวัด	รหัสทางไกล	หมายเลขตั้งต้น
52. กระบี่	(075)	61-XXXX
53. ตรัง	(075)	21-XXXX
	กันตัง	25-XXXX
	ห้วยยอด	27-XXXX
	ย่านตาขาว	28-XXXX
54. นครศรีธรรมราช	(075)	34-XXXX
55. พังงา	(076)	41-XXXX
	ตะกั่วป่า	42-XXXX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวัด	รหัสทางไกล	หมายเลขตั้งต้น
56. ภูเก็ต	(076)	21-XXXX
	ถลาง	31-XXXX
57. ชุมพร	(077)	50-XXXX
		52-XXXX
	หลังสวน	54-XXXX
	สวี	53-XXXX
58. ระนอง	(077)	82-XXXX
		81-XXXX
59. สุราษฎร์ธานี	(077)	28-XXXX
		27-XXXX
	พุนพิน	31-XXXX
	บ้านนาสาร	34-XXXX
	ไชยา	43-XXXX
	เกาะสมุย	42-XXXX
	เวียงสระ	36-XXXX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักษรย่อแสดงตำแหน่งจังหวัด

ลำดับที่	อักษรย่อ	จังหวัด
1	BKK	กรุงเทพมหานคร
2	PBI	เพชรบุรี
3	RBR	ราชบุรี
4	PKN	ประจวบคีรีขันธ์
5	NPT	นครปฐม
6	SKN	สมุทรสาคร
7	SKM	สมุทรสงคราม
8	KRI	กาญจนบุรี
9	AYA	อยุธยา
10	SPB	สุพรรณบุรี
11	ATG	อ่างทอง
12	SRI	สระบุรี
13	LRI	ลพบุรี
14	SBR	สิงห์บุรี
15	PRI	ปราจีนบุรี
16	SKW	สระแก้ว
17	NYK	นครนายก
18	CBI	ชลบุรี
19	RYG	ระยอง
20	CCO	ฉะเชิงเทรา
21	CTI	จันทบุรี
22	TRT	ตราด
23	UDN	อุดรธานี
24	NGP	หนองบัวลำภู
25	NKI	หนองคาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

26	NPM	นครพนม
ลำดับที่	อักษรย่อ	จังหวัด
27	MDH	มุกดาหาร
28	SNK	สกลนคร
29	LEI	เลย
30	KKN	ขอนแก่น
31	MKM	มหาสารคาม
32	RET	ร้อยเอ็ด
33	KSN	กาฬสินธุ์
34	NMA	นครราชสีมา
35	CPM	ชัยภูมิ
36	BRM	บุรีรัมย์
37	SRN	สุรินทร์
38	UBN	อุบลราชธานี
39	ACR	อำนาจเจริญ
40	YST	ยโสธร
41	SSK	ศรีษะเกษ
42	CMI	เชียงใหม่
43	LPN	ลำพูน
44	MSN	แม่ฮ่องสอน
45	CRI	เชียงราย
46	LPG	ลำปาง
47	PRE	แพร่
48	NAN	น่าน
49	PYO	พะเยา
50	PLK	พิษณุโลก
51	UTT	อุตรดิตถ์
52	STI	สุโขทัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

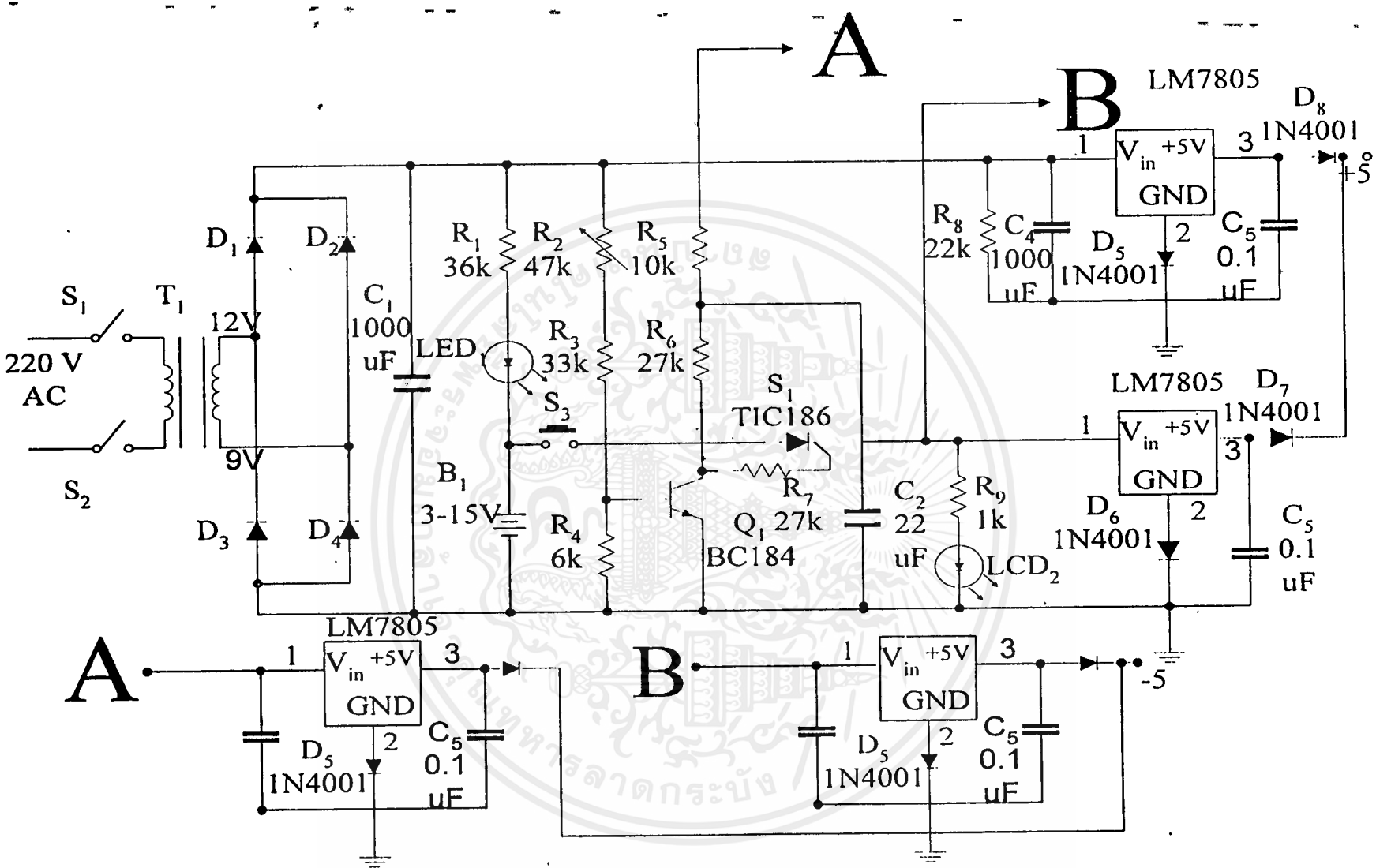
ลำดับที่	อักษรย่อ	จังหวัด
53	KPT	กำแพงเพชร
54	TAK	ตาก
55	NSN	นครสวรรค์
56	UGI	อุทัยธานี
57	CNT	ชัยนาท
58	PBN	เพชรบูรณ์
59	PCT	พิจิตร
60	YLA	ยะลา
61	PTN	ปัตานี
62	NWT	นราธิวาส
63	SKA	สงขลา
64	STN	สตูล
65	PLG	พัทลุง
66	NRT	นครศรีธรรมราช
67	TRG	ตรัง
68	KBI	กระบี่
69	PKT	ภูเก็ต
70	PNA	พังงา
71	SNI	สุราษฎร์ธานี
72	RNG	ระนอง
73	CPN	ชุมพร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

รูปต้นแบบของเครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรแหล่งจ่ายไฟ

ภาคผนวก ง

รายการข้อมูลและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

Industrial Blocks

LM567/LM567C Tone Decoder

General Description

The LM567 and LM567C are general purpose tone decoders designed to provide a saturated transistor switch to ground when an input signal is present within the passband. The circuit consists of an I and Q detector driven by a voltage controlled oscillator which determines the center frequency of the decoder. External components are used to independently set center frequency, bandwidth and output delay.

Features

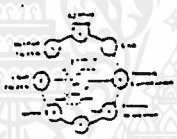
- 20 to 1 frequency range with an external resistor
- Logic compatible output with 100 mA current sinking capability
- Bandwidth adjustable from 0 to 14%
- High rejection of out of band signals and noise
- Immunity to false signals
- Highly stable center frequency
- Center frequency adjustable from 0.01 Hz to 500 kHz

Applications

- Touch tone decoding
- Precision oscillator
- Frequency monitoring and control
- Wide band FSK demodulation
- Ultrasonic controls
- Carrier current remote controls
- Communications paging decoders

Schematic and Connection Diagrams

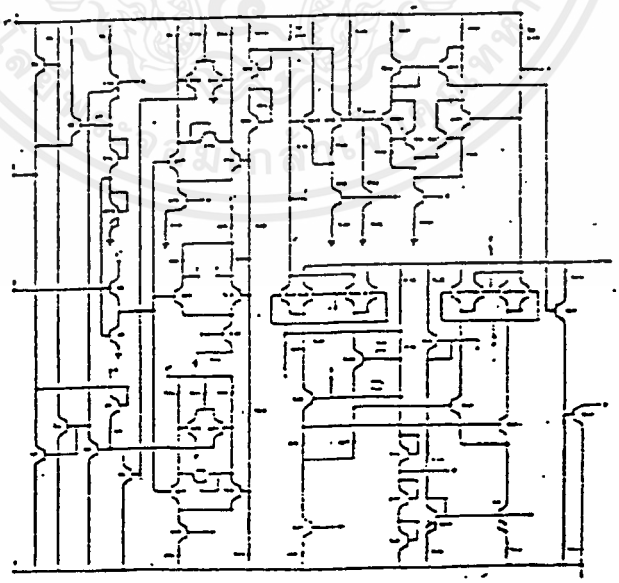
Metal Can Package



Order Number LM567H or LM567CH
See NS Package H08C

Dual-In-Line Package

Order Number LM567CN
See NS Package N08E



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Applications Information

MONOSTABLE OPERATION

In this mode of operation, the timer functions as a one-shot (*Figure 1*). The external capacitor is initially held discharged by a transistor inside the timer. Upon application of a negative trigger pulse of less than 1/3 V_{CC} to pin 2, the flip-flop is set which both releases the short circuit across the capacitor and drives the output high.

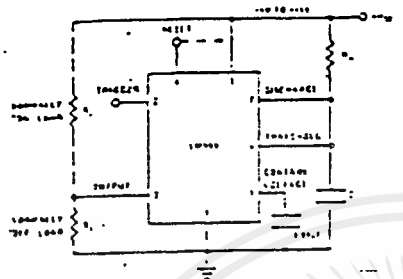


FIGURE 1. Monostable

The voltage across the capacitor then increases exponentially for a period of $t = 1.1\ R_A C$, at the end of which time the voltage equals 2/3 V_{CC}. The comparator then resets the flip-flop which in turn discharges the capacitor and drives the output to its low state. *Figure 2* shows the waveforms generated in this mode of operation. Since the charge and the threshold level of the comparator are both directly proportional to supply voltage, the timing interval is independent of supply.

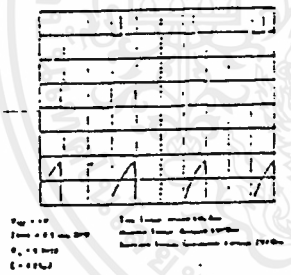


FIGURE 2. Monostable Waveforms

During the timing cycle when the output is high, the further application of a trigger pulse will not affect the circuit. However the circuit can be reset during this time by the application of a negative pulse to the reset terminal (pin 4). The output will then remain in the low state until a trigger pulse is again applied.

When the reset function is not in use, it is recommended that it be connected to V_{CC} to avoid any possibility of false triggering.

Figure 3 is a nomograph for easy determination of R, C values for various time delays.

NOTE: In monostable operation, the trigger should be driven high before the end of timing cycle.

ASTABLE OPERATION

If the circuit is connected as shown in *Figure 4* (pins 2 and 6 connected) it will trigger itself and free run as a

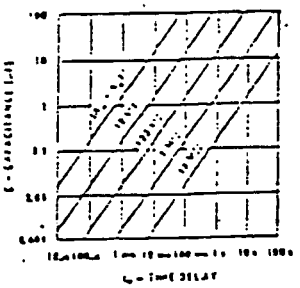


FIGURE 3. Time Delay

multivibrator. The external capacitor charges through $R_A + R_B$ and discharges through R_B. Thus the duty cycle may be precisely set by the ratio of these two resistors.

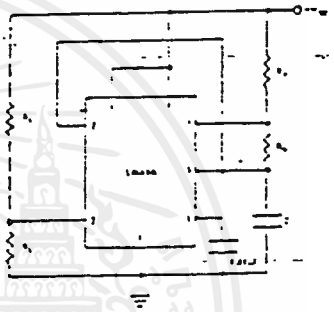


FIGURE 4. Astable

In this mode of operation, the capacitor charges and discharges between 1/3 V_{CC} and 2/3 V_{CC}. As in the triggered mode, the charge and discharge times, and therefore the frequency are independent of the supply voltage.

Figure 5 shows the waveforms generated in this mode of operation.

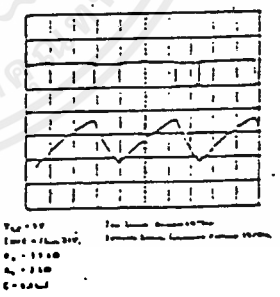


FIGURE 5. Astable Waveforms

The charge time (output high) is given by:

$$t_1 = 0.693\ (R_A + R_B)\ C$$

And the discharge time (output low) by:

$$t_2 = 0.693\ (R_B)\ C$$

Thus the total period is:

$$T = t_1 + t_2 = 0.693\ (R_A + 2R_B)\ C$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

Industrial Blocks

LM555/LM555C Timer

General Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200 mA or drive TTL circuits.

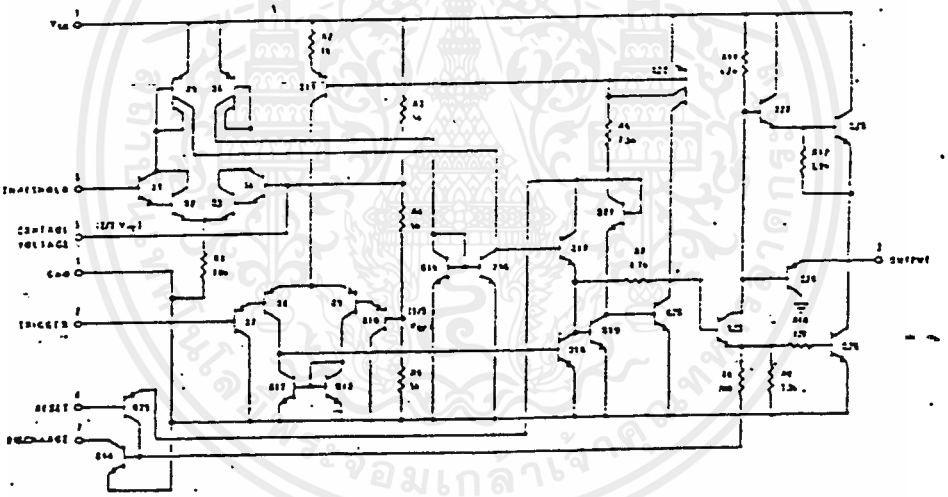
Features

- Direct replacement for SE555/NE555
- Timing from microseconds through hours
- Operates in both astable and monostable modes
- Adjustable duty cycle
- Output can source or sink 200 mA
- Output and supply TTL compatible
- Temperature stability better than 0.005% per °C
- Normally on and normally off output

Applications

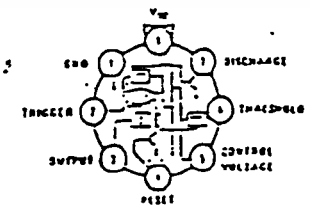
- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Linear ramp generator

Schematic Diagram



Connection Diagrams

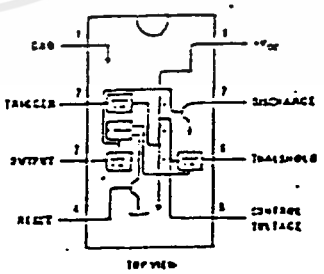
Metal Can Package



Top View

Order Number LM555H, LM555CH
See NS Package H08C

Dual-In-Line Package

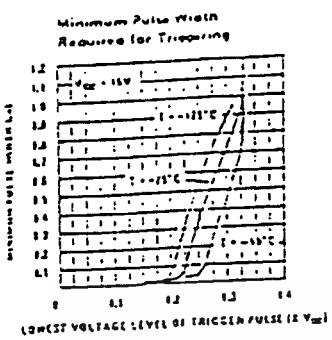


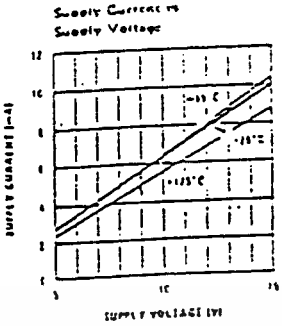
Top View

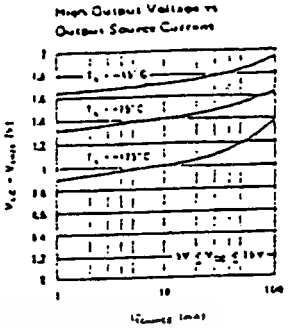
Order Number LM555CN
See NS Package N08B
Order Number LM555J or LM555CJ
See NS Package J08A

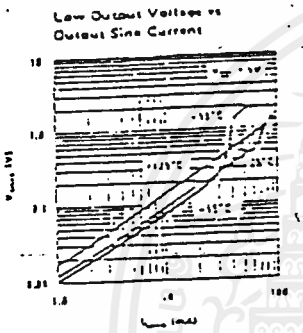
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

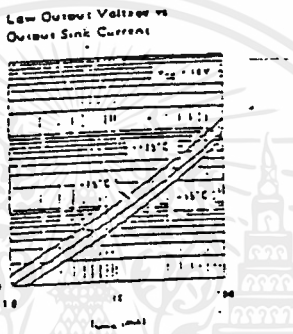
Typical Performance Characteristics

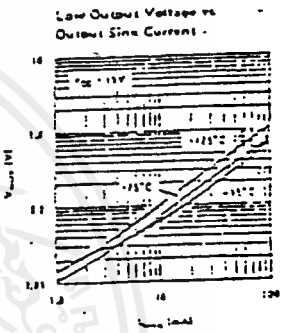


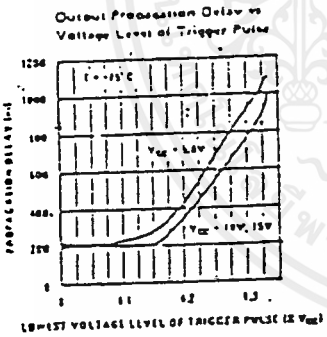


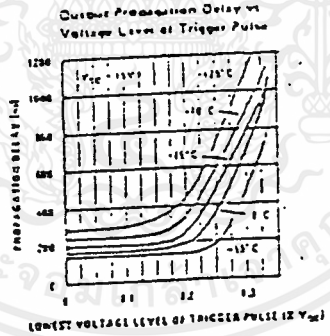


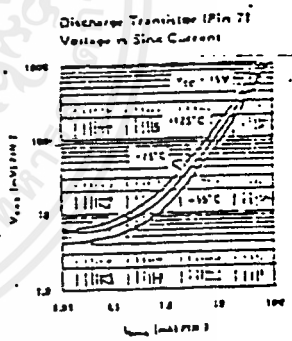


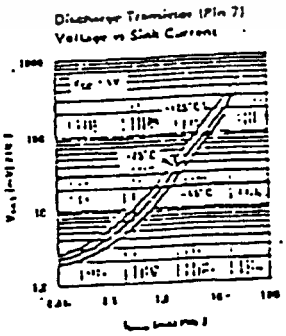












เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings†

	Parameter	Symbol	Min	Max	Units
1	DC Power Supply Voltage	V_{DD}		7	V
2	Voltage on any pin	V_I	V_{SS}-0.3	V_{DD}+0.3	V
3	Current at any pin (other than supply)	I_I		10	mA
4	Storage temperature	T_{STG}	-65	+150	°C
5	Package power dissipation	P_D		500	mW

† Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied. Derate above 75 °C at 16 mW / °C. All leads soldered to board.

Recommended Operating Conditions - Voltages are with respect to ground (V_{SS}) unless otherwise stated.

	Parameter	Sym	Min	Typ‡	Max	Units	Test Conditions
1	DC Power Supply Voltage	V_{DD}	4.75	5.0	5.25	V	
2	Operating Temperature	T_O	-40		+85	°C	
3	Crystal/Clock Frequency	fc		3.579545		MHz	
4	Crystal/Clock Freq.Tolerance	Δfc		±0.1		%	

‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

DC Electrical Characteristics - V_{DD}=5.0V± 5%, V_{SS}=0V, -40°C≤ T_O ≤ +85°C, unless otherwise stated.

		Characteristics	Sym	Min	Typ‡	Max	Units	Test Conditions
1	SUPPLY	Standby supply current	I_{DDQ}		10	25	μA	PWDN=V_{DD}
2		Operating supply current	I_{DD}		3.0	9.0	mA	
3		Power consumption	P_O		15		mW	fc=3.579545 MHz
4	INPUTS	High level input	V_{IH}	3.5			V	V_{DD}=5.0V
5		Low level input voltage	V_{IL}			1.5	V	V_{DD}=5.0V
6		Input leakage current	I_{IH}/I_{IL}		0.1		μA	$V_{IN}=V_{SS}$ or V_{DD}
7		Pull up (source) current	I_{SO}		7.5	20	μA	TOE (pin 10)=0, V_{DD}=5.0V
8		Pull down (sink) current	I_{SI}		15	45	μA	INH=5.0V, PWDN=5.0V, V_{DD}=5.0V
9		Input impedance (IN+, IN-)	R_{IN}		10		MΩ	@ 1 kHz
10		Steering threshold voltage	V_{TSt}	2.2	2.4	2.5	V	V_{DD} = 5.0V
11	OUTPUTS	Low level output voltage	V_{OL}			V_{SS}+0.03	V	No load
12		High level output voltage	V_{OH}	V_{DD}-0.03			V	No load
13		Output low (sink) current	I_{OL}	1.0	2.5		mA	V_{OUT}=0.4 V
14		Output high (source) current	I_{OH}	0.4	0.8		mA	V_{OUT}=4.6 V
15		V_{Ref} output voltage	V_{Ref}	2.3	2.5	2.7	V	No load, V_{DD} = 5.0V
16		V_{Ref} output resistance	R_{OR}		1		kΩ	

‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operating Characteristics - V_{DD}=5.0V±5%, V_{SS}=0V, -40°C≤ T_O ≤ +85°C ,unless otherwise stated.

Gain Setting Amplifier

	Characteristics	Sym	Min	Typ‡	Max	Units	Test Conditions
1	Input leakage current	I_{IN}			100	nA	$V_{SS} \le V_{IN} \le V_{DD}$
2	Input resistance	R_{IN}	10			MΩ	
3	Input offset voltage	V_{OS}			25	mV	
4	Power supply rejection	PSRR	50			dB	1 kHz
5	Common mode rejection	CMRR	40			dB	0.75 V ≤ V_{IN} ≤ 4.25 V biased at V_{Ref} =2.5 V
6	DC open loop voltage gain	A_{VOL}	32			dB	
7	Unity gain bandwidth	f_C	0.30			MHz	
8	Output voltage swing	V_O	4.0			V_{pp}	Load ≥ 100 kΩ to V_{SS} @ GS
9	Maximum capacitive load (GS)	C_L			100	pF	
10	Resistive load (GS)	R_L			50	kΩ	
11	Common mode range	V_{CM}	2.5			V_{pp}	No Load

MT8870D AC Electrical Characteristics - V_{DD}=5.0V ±5%, V_{SS}=0V, -40°C≤ T_O ≤ +85°C , using Test Circuit shown in Figure 10.

	Characteristics	Sym	Min	Typ‡	Max	Units	Notes*
1	Valid input signal levels (each tone of composite signal)		-29		+1	dBm	1,2,3,5,6,9
			27.5		869	mV_{RMS}	1,2,3,5,6,9
2	Negative twist accept				8	dB	2,3,6,9,12
3	Positive twist accept				8	dB	2,3,6,9,12
4	Frequency deviation accept		±1.5% ± 2 Hz				2,3,5,9
5	Frequency deviation reject		±3.5%				2,3,5,9
6	Third tone tolerance			-16		dB	2,3,4,5,9,10
7	Noise tolerance			-12		dB	2,3,4,5,7,9,10
8	Dial tone tolerance			+22		dB	2,3,4,5,8,9,11

‡ Typical figures are at 25 °C and are for design aid only: not guaranteed and not subject to production testing.

*NOTES
1. dBm= decibels above or below a reference power of 1 mW into a 600 ohm load.
2. Digit sequence consists of all DTMF tones.
3. Tone duration= 40 ms, tone pause= 40 ms.
4. Signal condition consists of nominal DTMF frequencies.
5. Both tones in composite signal have an equal amplitude.
6. Tone pair is deviated by ±1.5 %± 2 Hz.
7. Bandwidth limited (3 kHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2 %.
9. For an error rate of better than 1 in 10,000.
10. Referenced to lowest level frequency component in DTMF signal.
11. Referenced to the minimum valid accept level.
12. Guaranteed by design and characterization.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] องค์การโทรศัพท์แห่งประเทศไทย " สมุดรายนามผู้ใช้โทรศัพท์ของหน่วยงานองค์การโทรศัพท์แห่งประเทศไทย " , กองบริการศูนย์บริการทั่วไป ฝ่ายธุระการ , ทศ.ท ฉบับปี 2538 , หน้า 226-400

[2] ทีมงานศิลารีเสิร์ช " คู่มือการใช้งาน ANT-32 " , บริษัทศิลารีเสิร์ช จำกัด

[3] สุธินท์พรศิริกุล " ลึกอีกนิดกับโทรศัพท์ " วารสารเซมิคอนดัคเตอร์อิเล็กทรอนิคส์ " , ฉบับที่ 120-121 , 2535

[3] ยืน ภู่สุวรรณ , วัฒนา เชียงกุล , "สถาปัตยกรรม 8051" , บริษัทซีเอ็ดยูเคชั่นจำกัด , 2532

[4] " ไอซีน่าสนใจ MT8870" วารสารเซมิคอนดัคเตอร์ , ฉบับที่81, 2530

[5] วีระพล ตุ้มสุด , อนุพงศ์ อิศรภัคดี , นิติพัฒน์ เบญจพลาภรณี , " ปริญญานิพนธ์ เครื่องคำนวณค่าบริการการใช้โทรศัพท์แบบอัตโนมัติ " ภาควิชาครุศาสตร์วิศวกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2536

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
หอสมุดคณะครุศาสตร์อุตสาหกรรม สจล.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้