

ระบบการควบคุมอัตโนมัติในบ้าน
HOME AUTOMATIC



โดย

นายสุนทรเทพ
นายเสกสรรค์
นายอิทธิพล

รัตนพลแสนย์
ชินเกียรติยศ
จิตรประสาทิพย์

รหัส 37012122
รหัส 37012124
รหัส 37012129

อาจารย์ที่ปรึกษา

ศศ.พิพัฒน์ เสาทองธรรม

วัน เดือน ปี.....	๒๙ ก.ค. ๒๕๕๑
เลขทะเบียน.....	038031
เลขเรียกหนังสือ.....	T 39091 ๙๖๘๗

ปริญญาบัตรฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรีวิศวกรรมศาสตร
สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2539

ปริญญาบัตร ปีการศึกษา 2539

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบการควบคุมอัตโนมัติในบ้าน

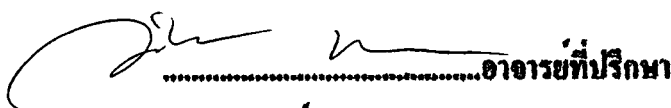
HOME AUTOMATIC

ผู้จัดทำ

นาย อุนทรเทพ รัตนพลแสนย์ รหัส 87012122

นาย เสกสรร ชื่นเทียนคต รหัส 87012124

นาย อธิรพีล จิตรประสาทชัย รหัส 87012129


.....อาจารย์ที่ปรึกษา
(รท. พิพัฒน์ เถาหงคราม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อปริญญาโท
ชื่อผู้จัดทำ

ระบบควบคุมอัตโนมัติภายในบ้าน

นาย ชุมนทรเทพ รัตน์พลแสนย์

นาย เสกสรร ชื่นเทียนคค

นาย อธิรพีท จิตรประสาทชัย

อาจารย์ที่ปรึกษา

อาจารย์ พิพัฒน์ เถาหงคราม

บทคัดย่อ

ในปัจจุบันเทคโนโลยีที่ใช้กันอย่างแพร่หลาย และได้มีการพัฒนาเป็นอย่างมาก คือ โทรศัพท์ ปริมาณวิทยุวิทยุฉบับนี้ กล่าวถึงการพัฒนาทางโทรศัพท์ การควบคุมอัตโนมัติภายในบ้านซึ่งเป็นการประยุกต์นำระบบโทรศัพท์มาตั้งงานหรือ บริหารงานอุปกรณ์ไฟฟ้าโดยผ่านระบบการเข้าและถอดรหัส DTMF (Dual Tone multifrequency type) ซึ่งมีไมโครคอนโทรลเลอร์ MCS-51 เป็นตัวควบคุมส่วนกลาง เราสามารถควบคุม อุปกรณ์ไฟฟ้าได้ทั้งหมด 16 ช่องสัญญาณ โดยที่ 8 ช่องสัญญาณแรกสามารถตั้งเวลา ในการ เปิด-ปิด ได้ตลอด 24 ชม. เราสามารถทำการตรวจสอบสถานะของอุปกรณ์ไฟฟ้าต่าง ๆ ได้ อีกทั้งยังมีการรองรับโทรศัพท์การวัดอุณหภูมิอีกด้วย

ข้อมูลจากการทำงานทั้งหมด จะถูกเก็บบันทึกค่าไว้ในหน่วยความจำที่มีการ BACKUP ด้วย NVSRAM (Non Volatile SRAM) ทำให้ไม่มีการผิดพลาดของค่าข้อมูล อันเนื่องมาจาก กระแสไฟฟ้าขัดข้อง

นอกจากนี้ ยังสามารถทำการทดสอบแต่ละฟังก์ชันการทำงานได้ เช่น ทดสอบการทำงาน ของ MV8870 ,ISD1420 ,DS1620 เป็นต้น

PROJECT REPORT TITLE HOME AUTOMATIC
NAME MR. SUNTORNTEP RATPHOLSAEN
MR. SEKSAN CHUENKIANKOT
MR. ITTIPHOL JITPRASARTCHAI
PROJECT REPORT ADVISOR ASST. PROF. PIPAT LAOHASONGKHRAM

ABSTRACT

In present , the technology is use to be widespread and developed so much such as telephone. This thesis is explained telephone development.

Home automatic is explained to application of system and management electrical device. By pass decoder and encoder DTMF (Dual Tone Multifrequency type) system and micro processer MCS-51 to be central control.

This project can control 16 channels electrical device. So frist 8 channels can set time to open and close all day. Although this project can be check status device , measure temperature and automatic answer telephone.

All programs will backup by memmory units in NVSRAM (Non Volatile SRAM). So it does not have error to be due to electrical fail.

Besides , it can test each function such as test worked of MV8870 , ISD1420 , DS 1620

สารบัญ

	หน้าที่
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	3
-ความรู้เกี่ยวกับโทรศัพท์	3
-พื้นฐานทางโทรศัพท์	9
-โครงสร้างของไมโครคอนโทรลเลอร์ MCS-51	19
-การใช้งาน 8255 กับ 8051	23
-การถอดรหัสความถี่โทรศัพท์	32
-การส่งรหัสหมายเลข	38
-6242 REAL TIME CLOCK	40
-DOT MATRIX LCD MODULE	43
-ISD 12xx / 14xx SERIES (ไอซีบันทึกเสียง)	51
-DS1620 เทอร์โมมิเตอร์	55
บทที่ 3 การคำนวณและการออกแบบ	60
1. ส่วนตรวจจับสัญญาณกระดิ่ง	60
2. ส่วนการถอดสัญญาณการกดหมายเลข	60
3. ส่วนประมวลผลกลางและหน่วยความจำ	61
4. พอร์ตใช้งาน 8255	63
5. ส่วนควบคุมฐานเวลา	66
6. ส่วนการแสดงผล	68
7. OUTPUT DRIVER	69
8. การควบคุมการอัดเสียงโดย ISD 1420	71
9. สรุปล INPUT/ OUTPUT	72
บทที่ 4 การทดลอง ผลการทดลอง การใช้งานและการควบคุม	73
-การทดลองและผลการทดลอง	73
-การใช้งานและการควบคุม	74
บทสรุปและวิจารณ์	81
เอกสารอ้างอิง	
กิตติกรรมประกาศ	

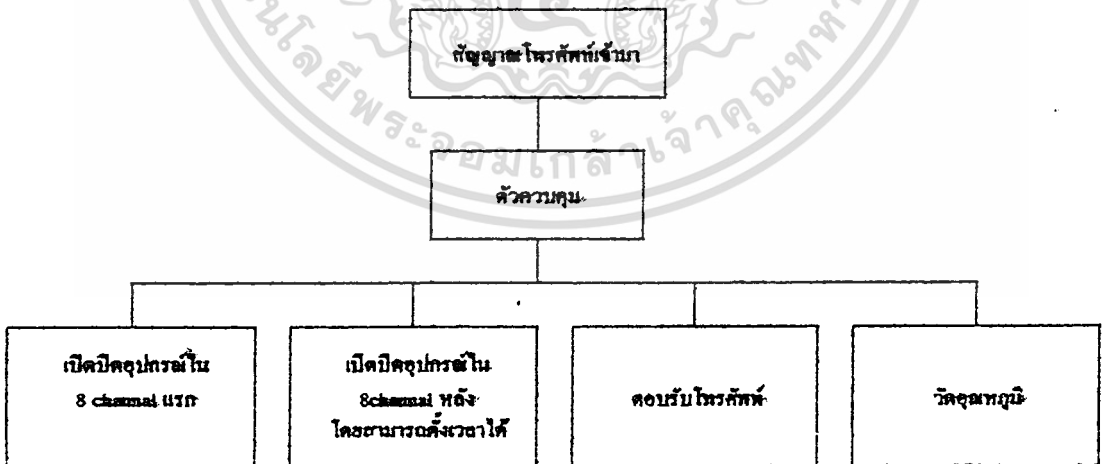
บทที่ 1

บทนำ

ปัจจุบันมีเทคโนโลยีที่ตอบสนองความต้องการเพื่อความสะดวกสบายในการดำรงชีวิตความปลอดภัยของมนุษย์ ซึ่งโครงการนี้ก็เป็นอย่างหนึ่งที่ตอบสนองความต้องการดังกล่าว

โดยใช้วิธีการตรวจสอบสถานะของอุปกรณ์ไฟฟ้าใดๆและทำการเปิดปิดได้ตามความต้องการการเปิดปิดอุปกรณ์ไฟฟ้าต่างๆ เราสามารถจะกระทำได้ โดยผ่านทางสายโทรศัพท์ซึ่งปัจจุบันได้รับการพัฒนาไปเป็นอันมาก โดยโครงการนี้นอกจากจะใช้เปิดปิดอุปกรณ์ไฟฟ้าใดๆแล้ว ยังสามารถนำมาประยุกต์ใช้งานอย่างอื่นได้อีก ได้แก่

1. เครื่องใช้ไฟฟ้าและอุปกรณ์ไฟฟ้าต่างๆ เช่น แอร์ โทรทัศน์ วิทยุ ฯลฯ สามารถตั้งโปรแกรมให้เปิด/ปิดอัตโนมัติได้ ซึ่งในขณะที่ไม่อยู่บ้าน
2. ประตูบ้าน สามารถสั่งให้เปิด/ปิดด้วยโทรศัพท์ได้ ตั้งเวลาเปิด/ปิด ไฟหน้าบ้านตามเวลาพระอาทิตย์ขึ้นหรือตก
3. มันทีกและตอบรับ โทรศัพท์ได้ และยังสามารรถโทรศัพท์เข้ามาเปลี่ยนค่าชุดในการตอบรับใหม่ได้ ในขณะที่ไม่อยู่บ้าน
4. วัดค่าอุณหภูมิและฟังก์ชันการเปรียบเทียบเพื่อเป็นตัว sensor ได้ การทำงานทั้งหมดจะแสดงผลด้วย LCD ให้เป็นโหมดการทำงานต่างๆและฐานเวลาที่สามารถตั้งเวลาได้อย่างแม่นยำ โดยกำหนดเป็น วินาที, นาที, ชั่วโมง, วัน, เดือน, ปี, สัปดาห์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณประโยชน์ที่ได้รับอาจสรุปได้ดังนี้

1. สะดวกสบาย
2. ควบคุมแสงสว่าง
3. อนุรักษ์พลังงาน
4. รายงานสภาวะในบ้าน
5. ควบคุมอุปกรณ์ไฟฟ้า
6. บันทึกโทรศัพท์ตอบรับ

ในโครงการนี้มีจุดเด่นอย่างหนึ่งคือการอ่านและเขียนค่าข้อมูลอนุกรมระหว่าง MCS-51 กับ DS1620 โดยพอร์ตถูกใช้ควบคุมคอกันจำนวน 3 พอร์ต ซึ่งเราใช้เป็นพอร์ตส่งค่าอุณหภูมิ และยังมีการใช้ IC ISD1420 ซึ่งเป็นบันทึกเสียงได้ในตัว

สุดท้ายนี้คณะผู้จัดทำหวังว่า โครงการนี้จะเป็นประโยชน์แก่ผู้ที่สนใจบ้างไม่มากนักน้อย

คณะผู้จัดทำ

นาย สุนทรเทพ

รัตนพลแสนย์

นาย เสกสรร

ชินเคียนคค

นาย อธิธิพล

จิตรประสาทรชัย

บทที่ 2

ทฤษฎี

ความรู้เกี่ยวกับโทรศัพท์

หน้าที่ของเครื่องโทรศัพท์

โทรศัพท์ที่ใช้กันอยู่ทุกวันนี้ ที่เคยเห็นกันทั่วๆ ไปก็มีอยู่ 2 แบบคือ แบบกดปุ่มและแบบหมุน แต่หน้าที่ของทั้ง 2 ระบบเหมือนกัน แต่จะแตกต่างกันที่จะส่งสัญญาณออกไปเป็นความถี่ที่แตกต่างกัน ส่วนแบบหมุนจะส่งสัญญาณเป็นจำนวนพัลส์ หน้าที่หลักๆ ของทั้ง 2 แบบ สามารถสรุปได้ดังนี้

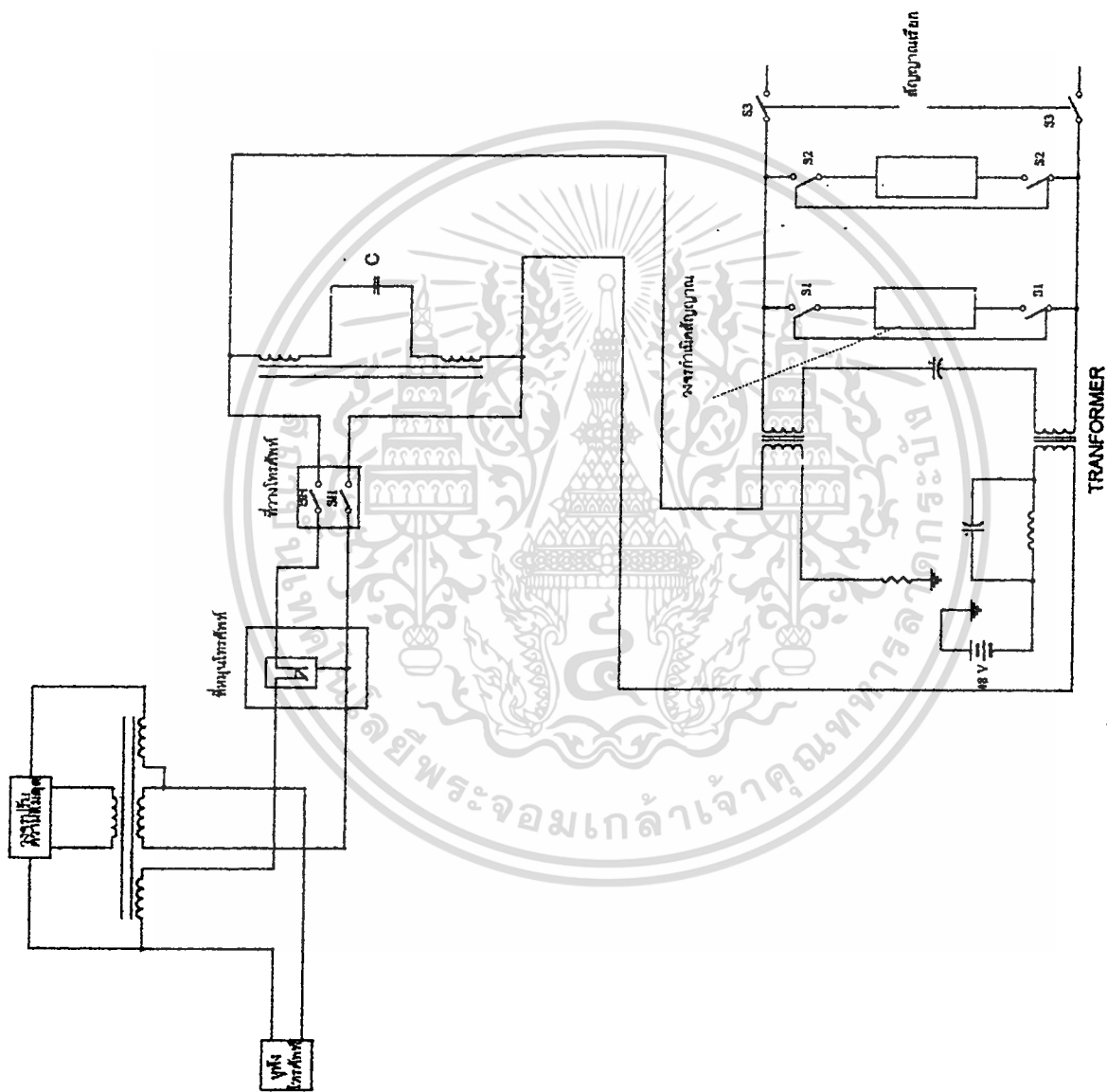
1. เครื่องโทรศัพท์จะรับรู้ว่ามีผู้ต้องการใช้โทรศัพท์ เมื่อมีการยกหูโทรศัพท์ขึ้น
2. เครื่องโทรศัพท์จะส่งสัญญาณที่เรียกว่า สัญญาณหมุน (dial tone) นอกว่าพร้อมที่จะทำการกดปุ่ม หรือหมุนหมายเลขที่จะติดต่อได้ซึ่งก็คือ เสียงที่ได้ยินเมื่อเวลายกหู เป็นสัญญาณเสียงที่มีความถี่ 350 เฮิรตซ์ มอดูเลตรวมกัน
3. เครื่องโทรศัพท์จะทำหน้าที่ส่งรหัสตามหมายเลขที่ผู้เรียกต้องการจะติดต่อด้วย ไปยังชุมสายที่ควบคุม
4. เครื่องโทรศัพท์จะส่งสัญญาณบอกผู้เรียกว่า หมายเลขที่ต้องการติดต่อด้วยไม่ว่างก็จะส่งสัญญาณกลับ (ring back) ซึ่งมีความถี่ 440 เฮิรตซ์ กับ 480 เฮิรตซ์ มอดูเลตกันมา โดยจะดัง 2 วินาที แล้วเงียบ 4 วินาที สลับกันไปแต่ถ้าหมายเลขที่ต้องการจะเรียกไม่ว่างก็จะส่งสัญญาณความถี่ 480 เฮิรตซ์ กับ 620 เฮิรตซ์ มอดูเลตกันมา
5. สามารถเปลี่ยนรูปพลังงานเสียงเป็นสัญญาณไฟฟ้า และสัญญาณไฟฟ้ากลับมาเป็นพลังงานเสียง
6. เครื่องโทรศัพท์จะส่งสัญญาณ ไปชุมสายเพื่อแจ้งให้ทราบว่สิ้นสุดการใช้งานแล้ว และให้ชุมสายเล็กทำการติดต่อกับอีกฝ่ายหนึ่งได้

กลไกการเชื่อมต่อวงจร

วงจรพื้นฐานข้างในรวมทั้งการเชื่อมต่อกับชุมสายเบื้องต้นกันเลข ตามรูปที่ 1 จะเห็นว่า จะเชื่อมต่อกับชุมสายด้วยสาย 2 เส้นคือ T(TIP) และ R(RING) เมื่อผู้ใช้ยกหูโทรศัพท์ขึ้นแหล่งจ่ายไฟตรงของชุมสาย (48 โวลต์) ก็จะถูกต่อเข้ากับวงจรของโทรศัพท์โดย สุกสวิทช์ (hook-switch) ในส่วนที่เชื่อมต่อระหว่างหูฟัง (ซึ่งรวมทั้งไมโครโฟนด้วย) กับสายโทรศัพท์ก็จะต้องมี หม้อแปลงอัตโนมัติ (auto transformer) ทำหน้าที่ปรับอิมพีแดนซ์ของหูโทรศัพท์ให้สมดุลกัน เพื่อให้การรับส่งสัญญาณมีประสิทธิภาพที่สุด รวมไปถึงการทำให้ผู้พูดได้ยินเสียงที่ตัวเองพูดไป (side tone) ในระดับที่เหมาะสมด้วย

เมื่อมีการติดต่อกันระหว่างเครื่องโทรศัพท์กับชุมสายแล้ว ก็จะมีการส่งสัญญาณไปยังอุปกรณ์ สวิตซ์เพื่อบอกให้รู้ขณะนี้คู่สายนี้ไม่ว่างแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1

แสดงวงจรภายในเครื่องโทรศัพท์และการเชื่อมต่ออเนก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการหมุนหมายเลขโทรศัพท์ก็คือ การส่งสัญญาณพัลส์ (pulse train) ตั้งแต่ 1 ถึง 10 พัลส์ เช่น ถ้ามีการส่งพัลส์ 1 พัลส์ ก็หมายถึงการหมุนหมายเลขศูนย์ส่ง 2 พัลส์ ก็หมายถึงเลข 1 ดังนั้น ถ้าหมุนเลข 9 ก็จะมีการส่งพัลส์จำนวน 10 พัลส์นั่นเอง และความเร็วในการส่งก็คือ 10 พัลส์ต่อวินาที

สำหรับโทรศัพท์ที่ใช้การกดปุ่มนั้นก็จะเป็นการส่งสัญญาณที่มีค่าความถี่ที่แตกต่างกันออกไป เป็นความถี่ที่อยู่ในย่านความถี่เสียง เพียงแค่ว่าในการกดครั้งหนึ่งจะมีสัญญาณเสียงที่มอดูเลตแล้วถูกส่งออกไป 2 ความถี่ ตามตารางที่ 1

ตารางที่ 1 แสดงความถี่ที่มอดูเลตกันเมื่อกดหมายเลข

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

ทางชุมสายเมื่อได้รับข้อมูลจากผู้เรียกไว้แล้วก็จะแปลงสัญญาณที่ได้รับมาส่งให้อุปกรณ์ทำงาน เพื่อทำการต่อสายให้กับผู้เรียก ถ้าปลายทางที่ต้องการติดต่อด้วยไม่ว่างชุมสายก็จะส่ง สัญญาณไม่ว่าง (busy tone) ไปยังผู้เรียกเพื่อแจ้งให้ทราบว่าไม่สามารถต่อวงจรให้ได้ แต่ถ้าปลายทางว่างชุมสายก็จะส่ง สัญญาณเรียก (ringing signal) ไปยังปลายทางและ ส่งสัญญาณเรียกกลับ (ringing tone) ไปยังผู้เรียกเพื่อแจ้งให้ทราบว่าสามารถต่อวงจรให้ได้ตามต้องการแล้ว

การสนทนา

เมื่อปลายทางหรือผู้เรียกหรือผู้ถูกเรียกกดหูโทรศัพท์ขึ้น การทำงานในส่วนควบคุมของชุมสาย โทรศัพท์ก็จะหยุดเพื่อที่จะรอทำงานให้กับผู้อื่นที่เรียกเข้ามาต่อไปแต่หน้าที่ของชุมสายสำหรับตอนนี้ก็คือ การทำงานของมิเตอร์สำหรับเรียกเก็บค่าบริการ ในภายหลัง

ในระหว่างที่ทำการสนทนาอยู่เครื่องโทรศัพท์ก็จะทำงาน 2 โหมดไปพร้อมๆกันคือ แปลงจาก สัญญาณไฟฟ้าให้เป็น สัญญาณเสียง (acoustic energy) ซึ่งจะเรียกว่า โหมดรับสัญญาณ (receiver mode) และในทางกลับกันโหมดที่ทำหน้าที่แปลงจากสัญญาณเสียงให้เป็นสัญญาณไฟฟ้า (transmitter mode) ในโหมดนี้เองที่มีเรื่องของ การป้อนกลับของสัญญาณเข้ามานี้เกี่ยวข้องกับสัญญาณที่ผู้พูดสามารถได้ยินเสียงของตนเองจากหูฟังด้วยเรียกเสียงนี้ว่า side tone ซึ่งจำเป็นอย่างมากที่จะต้องป้อน

กลับมา เพราะไม่เช่นนั้นจะไม่สามารถรู้ได้เลยว่าควรพูดให้มีเสียงดังระดับใดจึงจะพอเหมาะที่คู่สนทนา
ได้ยินเสียงพูดของผู้เรียกอย่างชัดเจน

เมื่อสิ้นสุดการสนทนาทั้ง 2 ฝ่าย วางหูโทรศัพท์ลง สัญญาณจากสวิตซ์ก็จะบอกให้ทวนสายทำ
การเปิดวงจรที่ทำการติดต่ออยู่ออก อุปกรณ์ต่าง ๆ ก็จะว่างและพร้อมสำหรับการติดต่อครั้งต่อไป

ระบบการส่งสัญญาณในสายส่ง

ตอนนี้มาคุยกันว่าในสายส่งโทรศัพท์นั้นมีสัญญาณอะไรบ้างที่เข้ามาเกี่ยวข้องกับจนทำให้เราสามารถ
พูดคุยกันในระยะทางไกล ๆ ได้ สัญญาณที่จะปรากฏในสายส่งจะสามารถแยกได้เป็น 2 ชนิดใหญ่ ๆ
ได้ก็คือ สัญญาณเสียงที่พูดคุยกันและอีกสัญญาณก็คือ สัญญาณเสียงที่ใช้ในการควบคุมระบบสวิตซ์
ซึ่งใช้ในการเชื่อมต่อวงจรระหว่างผู้เรียกนั่นเองรวมทั้งสัญญาณเรียกกลับ สัญญาณบอกไม่ว่าง

เจ้าสัญญาณควบคุมที่ว่านี้เองอาจจะเป็นได้ทั้งสัญญาณอะนาลอกหรือสัญญาณดิจิทัลก็ได้ อย่า
ลืมว่าโทรศัพท์แบบหมุนกับแบบกดปุ่มมีการส่งรหัสหมายเลขโทรศัพท์คนละแบบกัน

ดังนั้นในการส่งสัญญาณออกไปในสายส่งบางครั้งอาจจะมีการส่งทั้งสัญญาณดิจิทัล และ
สัญญาณอะนาลอกไปพร้อม ๆ กันก็ได้ มาลองดูรายละเอียดของแต่ละสัญญาณกัน

สัญญาณเสียงพูด

สัญญาณเสียงพูดจัดเป็นสัญญาณอะนาลอก เสียงพูดมีแบนด์วิดท์ตั้งแต่ 100 เฮิรตซ์ไปจนถึง 6
กิโลเฮิรตซ์ แต่จริง ๆ แล้วเสียงพูดที่ทำให้คนทั้งสามารถฟังแล้วจับใจความได้สบาย ๆ จะอยู่ในช่วง
200-4,000 เฮิรตซ์เท่านั้น วงจรกรองความถี่จึงได้นำมาใช้เพื่อป้องกันสัญญาณที่ไม่ต้องการเข้ามาภายใน
ระบบ โดยจะยอมให้สัญญาณที่มีความถี่ตั้งแต่ 0-4,000 เฮิรตซ์ สามารถผ่านเข้าไปในระบบได้และ
ความถี่ย่านนี้ก็มีชื่อเรียกว่า ช่องสัญญาณเสียงพูด (voice channel) หรือ VF channel แต่อย่างไรก็
ตาม แบนด์วิดท์ของเสียงพูดในการส่งจริงจะอยู่ในช่วง 300-3,000 เฮิรตซ์เท่านั้น ไม่ได้มีการใช้ช่อง
สัญญาณในการส่งเต็มย่านความถี่

ในช่วงความถี่ 300-3,000 Hz ประกอบไปด้วยสัญญาณต่าง ๆ หลายสัญญาณ ไม่ว่าจะเป็น
สัญญาณหมุน (dial tone) หรือสัญญาณควบคุมต่าง ๆ ก็จะถูกส่งไปในช่วงความถี่นี้ทั้งสิ้น

เมื่อกล่าวถึงระดับความดังของเสียงที่ได้ยินนั้นก็คือขนาดแอมพลิจูดของสัญญาณ ซึ่งสามารถ
อธิบายให้เห็นภาพพจน์ได้คือขึ้น โดยอาศัยสมการทางคณิตศาสตร์ โดยอธิบายในรูปของพลังงานที่
ปรากฏที่โหลด เช่น สายโทรศัพท์คู่หนึ่งมีอิมพีแดนซ์ 600 โอห์ม พลังงานที่ปรากฏที่โหลดก็คือ

$$P_{load} = e_s^2 / 600$$

โดย P_{load} คือ กำลังที่ตกคร่อมโหลด (วัตต์)

e_s คือ ระดับแรงดันของสัญญาณที่ส่งไป (โวลต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในระบบโทรศัพท์และวงจรรีเลย์โทรอนิกส์ที่เกี่ยวข้องกับเสียงแล้ว มักจะใช้การเปรียบเทียบกับกำลังขนาดมิลลิวัตต์ต่ออู่อเสมอโดยอยู่ในรูปของเดซิเบล ซึ่งมีสมการดังนี้

$$dB = 10 \log (P_1/P_2)$$

สัญญาณรบกวน

ในระบบใด ๆ ของวงจรรีเลย์โทรอนิกส์ย่อมจะต้องมีสัญญาณรบกวนเข้ามาเกี่ยวข้องด้วยเสมอ อาจจะทำให้เกิด ความผิดเพี้ยน (distort) ของสัญญาณเสียงพูดได้ และสิ่งที่ทำให้เกิดสัญญาณรบกวนขึ้นก็เป็นสิ่งแวดล้อมรอบ ๆ ข้างนี้เอง เช่น ความร้อน การสวิตช์ของอุปกรณ์โทรศเคอร์ สายไฟฟ้ากำลังสูงที่อยู่ใกล้ ๆ กับสายส่งสัญญาณ หรือแม้แต่ข้อต่อของสายที่บกพร่อง สิ่งเหล่านี้ล้วนแต่มีผลให้ประสิทธิภาพของสัญญาณรบกวนอีกชนิดหนึ่งคือ เสียงสะท้อน (echo) ในสายโทรศัพท์ สาเหตุของการเกิดเสียงสะท้อนก็คือ เกิดการไม่สมดุลกัน (mismatching) ระหว่างอิมพีแดนซ์ของสายส่งกับอุปกรณ์ทางด้านเอาต์พุต โดยมากแล้วมักจะพบในการเชื่อมต่อกันระหว่างระบบโทรศัพท์ที่มี 2 สาย กับระบบที่มี 4 สาย และมีปัญหาเรื่องเสียงสะท้อนจะมากยิ่งขึ้น ถ้าหากระยะทางระหว่างจุดที่ทำให้เกิดเสียงสะท้อนไกลกันมาก ๆ แต่โดยปกติแล้วมักจะไม่มีรู้สึกถึงการสะท้อนกลับของเสียงจนถึงขั้นที่ทำให้รู้สึกรำคาญ

การมัลติเพล็กซ์สัญญาณ

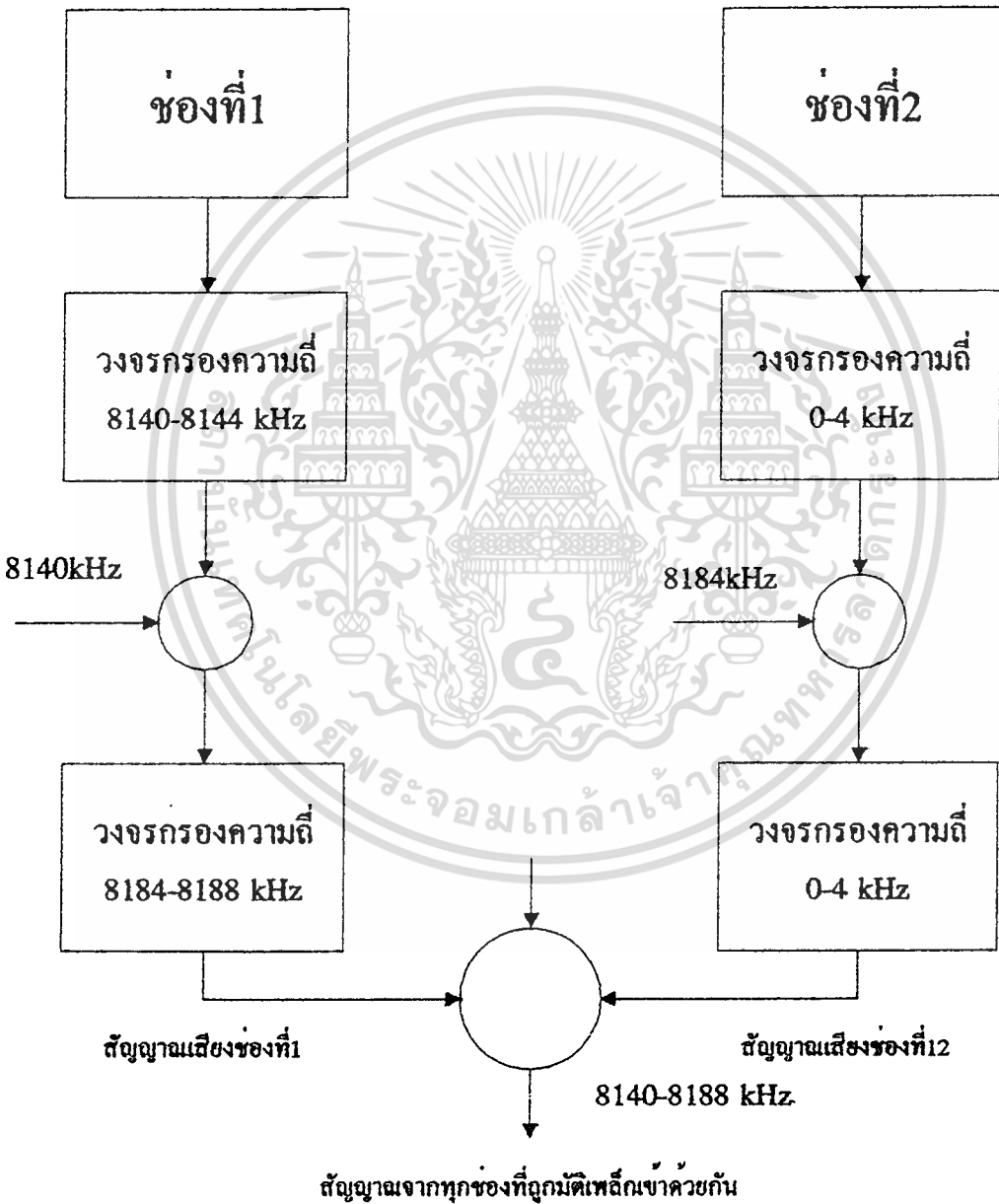
ถ้าหากทำให้สายส่งสัญญาณโทรศัพท์เส้นหนึ่ง ๆ สามารถใช้ส่งสัญญาณจากหลาย ๆ เครื่องได้ในเส้นเดียวกันก็จะทำให้สะดวกและประหยัดในการเดินสาย ดังนั้นในระบบส่งสัญญาณที่เชื่อมต่อกันระหว่างพื้นที่บริการ (service area) กับชุมสายท้องถิ่นนั้น จึงได้มีการนำวิธีการมัลติเพล็กซ์สัญญาณมาใช้ ซึ่งเป็นระบบที่เรียกว่า frequency division multiplexing (FDM) ซึ่งระบบนี้จะทำให้สามารถส่งสัญญาณที่มีความถี่ต่างกันไปในสายส่งเดียวกันได้

ดังที่ได้กล่าวมาแล้วข้างต้นว่าความถี่ของสัญญาณเสียงที่เหมาะสม ที่ใช้ในการส่งไปนั้นจะอยู่ในช่วงความถี่ตั้งแต่ 0-4 กิโลเฮิร์ตซ์ ถ้าหากต้องการจะส่งสัญญาณเสียงจากหลาย ๆ แหล่งไปพร้อมกัน จะต้องทำการมอดูเลตสัญญาณเหล่านั้นด้วยคลื่นพาหะที่มีความถี่สูงกว่าสัญญาณเสียงมาก ๆ สำหรับวิธีการมอดูเลตที่ใช้จะเป็นแบบ AM เป็นการมอดูเลตความถี่พาหะขนาด 8140 กิโลเฮิร์ตซ์ เข้าไปกับสัญญาณเสียง

สำหรับสัญญาณเสียงที่มาจากแหล่งอื่นก็ต้องนำมอดูเลตกับคลื่นพาหะที่มีความถี่แตกต่างกันออกไปจาก 8140 กิโลเฮิร์ตซ์ สมมติว่าในสายส่งเส้นหนึ่งต้องการจะส่งสัญญาณเสียงพูดจากโทรศัพท์ 12 เครื่อง ภายในสายส่งเส้นเดียวกันก็สามารถทำได้ ก่อนอื่นต้องพิจารณาก่อนว่าแบนด์วิดท์ของสัญญาณเสียงมีค่าเท่ากับ 4 กิโลเฮิร์ตซ์ แต่มีสัญญาณที่ที่ต้องการทั้งหมด 12 สัญญาณ เพราะฉะนั้นแบนด์วิดท์ใหม่จึงเท่ากับ 84 กิโลเฮิร์ตซ์ (12 X 4 kHz)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตัวอย่างที่ขมความถี่ของคลื่นพาห้ที่ต่ำสุดคือ 8140 กิโลเฮิร์ตซ์ เพราะฉะนั้นความถี่ของคลื่นพาห้สำหรับที่จะมอดูเลตกับสัญญาณเสียงจากโทรศัพท์ เครื่องที่ 12 คือ 8,188 กิโลเฮิร์ตซ์ (8,140 kHz + 48 kHz) เห็นได้ว่าวิธีการนี้จะทำให้ส่งสัญญาณได้มากช่องขึ้นโดยสัญญาณไม่เกี่ยวข้องกับกันเลย สำหรับทางฝ่ายรับก็จะต้องมี การดีมอดูเลต (demodulate) เพื่อจะเอาสัญญาณเสียงพูดจริง ๆ ผ่านเข้าไปยังหูของผู้ฟังของผู้รับ โดยการใช้วงจรกรองความถี่เข้ามาช่วยในการดีมอดูเลต เท่านั้นก็สามารถประหยัดค่าใช้จ่ายและความยุ่งยากในเรื่องของสายส่งสัญญาณลงไปได้



รูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นฐานโทรศัพท์

ในตอนนี้เป็นรายละเอียดของวงจรหมุนหมายเลขทั้งแบบปุ่มกดและแบบแป้นหมุน โดยจะกล่าวถึงทั้งวงจรพื้นฐานและวงจรที่ได้รับการพัฒนาแล้วรวมทั้งการเปรียบเทียบข้อดีของระบบการส่งหมายเลขทั้ง 2 ระบบด้วย

ในรูปที่ 3 เป็นบล็อกไดอะแกรมของส่วนต่าง ๆ ที่จำเป็นในเครื่องโทรศัพท์โดยจะเชื่อมต่อกับขุมสาย T (tip) และขุมสาย R (ring) วงจรแรกที่เชื่อมต่อระหว่างวงจรถ่ายในเครื่องโทรศัพท์กับอุปกรณ์ของขุมสายก็คือ วงจรกำเนิดเสียงเรียก (ringing signal) เมื่อมีการติดต่อมาจากผู้อื่น เหตุผลประการสำคัญที่ต้องนำวงจรส่วนนี้มาเชื่อมต่อกับขุมสายโดยตรงก็คือ เมื่อวางหูโทรศัพท์ไว้กับที่วางตามปกติ สุกสวิช (switch hook) จะถูกเปิดวงจรออกทำให้ไม่มีแรงดันจากขุมสายผ่านไปยังวงจรส่วนที่อยู่หลังสุกสวิชได้ ดังนั้นถ้าวงจรกำเนิดสัญญาณเรียกอยู่หลังจากสุกสวิชหรือเรียกได้ในเวลาที่มีผู้คิด ต่อเข้ามา

เมื่อมีการวางขุมโทรศัพท์ขึ้นสุกสวิช S1 และ S2 ก็จะปิดวงจรทำให้มีกระแสจากขุมสายไหลครบวงจรผ่านเครื่องโทรศัพท์ได้ ในขณะที่เดียวกันกระแสค่าเดียวกันนี้จะไหลผ่านขดลวดของรีเลย์ที่ขุมสายด้วย ก็จะทำหน้าที่สัมผัสของรีเลย์ขุมสายถูกปิดลง เพื่อที่จะให้อุปกรณ์ต่าง ๆ ในขุมสายพร้อมที่จะทำการติดต่อกับเครื่องโทรศัพท์ได้ จากนั้นขุมสายก็จะส่งสัญญาณหมุน (dial tone) ไปยังผู้ที่อกหูโทรศัพท์ เพื่อให้ผู้นั้นส่งหมายเลขโทรศัพท์ของผู้ที่ต้องการจะติดต่อด้วยมายังขุมสาย หลังจากที่ขุมสายได้รับหมายเลขแรกที่ถูกส่งมาแล้ว ขุมสายก็จะเลิกส่งสัญญาณหมุน ซึ่งกระบวนการตอนนี้จะเกิดขึ้นอย่างรวดเร็ว

ถ้าได้ติดตามบทความในตอนที่แล้วคงจำได้ว่าการส่งหมายเลขโทรศัพท์ไปยังขุมสายนั้นสามารถกระทำได้ 2 วิธี วิธีแรกเป็นการส่งสัญญาณพัลส์ที่แสดงถึงค่าของหมายเลขค่า ๆ และอีกวิธีหนึ่งก็คือการส่งสัญญาณเป็นความถี่ต่าง ๆ กัน โดยค่าของตัวเลขจะถูกแทนด้วยค่าความถี่ 2 ความถี่ที่มอดูเลตกัน คราวนี้จะลองมาเปรียบเทียบถึงลักษณะการใช้งานของแต่ละแบบกัน

ระบบโทรศัพท์แบบหมุนหมายเลข (Rotating - type)

จะเป็นวงจรที่ใช้การส่งหมายเลขโทรศัพท์ในแบบหมุนจะเห็นว่าสวิช S3 จะถูกเปิดวงจรออกเมื่อมีการหมุนหมายเลขโทรศัพท์ เมื่อสวิช S3 ถูกเปิดวงจรออกก็จะไม่มีกระแสไหลผ่านเข้าไปในวงจรส่วนที่อยู่ถัดไปได้ จึงเสมือนว่าเป็นการขัดจังหวะ (interruption) การไหลของกระแส สำหรับจำนวนครั้งที่สวิช S3 ถูกเปิดออกจะขึ้นกับระยะห่างของแป้นหมุน (dialer) ที่ถูกหมุนไป กับตำแหน่งปกติในขณะที่ไม่มีการหมุนหมายเลขใด ๆ เป็นต้นว่า ถ้าหมุนหมายเลข 4 สวิช S3 ก็จะถูกทำให้เปิดออก 4 ครั้ง หรือว่าหมุนหมายเลข 7 สวิช S3 ก็จะถูกเปิดออก 7 ครั้ง ซึ่งสวิช S3 จะถูกเปิดวงจรในช่วงที่ปล่อยให้แป้นหมุนกลับสู่ตำแหน่งเดิมเท่านั้น ไม่ได้เกิดขึ้นในระหว่างที่ทำการหมุนหมายเลขอยู่

ลักษณะของรูปสัญญาณเมื่อมีการหมุนหมายเลขโทรศัพท์ ในคอนแรกโทรศัพท์ที่อยู่ในสถานะ
ออนฮุก (on - hook) คือ หูโทรศัพท์จะถูกวางอยู่บนที่วางหูโทรศัพท์ตามปกติ ไม่มีกระแสจาก
ชุมสายเข้าสู่โทรศัพท์ เพราะขณะนี้วงจรถูกเปิดออกโดยสวิตช์ แต่เมื่อมีการยกหูโทรศัพท์ขึ้น
โทรศัพท์จะอยู่ในสถานะออฟฮุก (off - hook) สวิตช์ จะถูกปิดวงจรลงทำให้มีกระแสไหลครบวงจร
ได้ และเมื่อมีการหมุนหมายเลขโดยในรูปจะเป็นการหมุนหมายเลข “4” ก็จะทำให้วงจรถูกเปิดออก
ด้วยสวิตช์ S3 เป็น จำนวน 4 ครั้ง ก็จะได้รูปสัญญาณ ออกมาอย่างที่เห็น

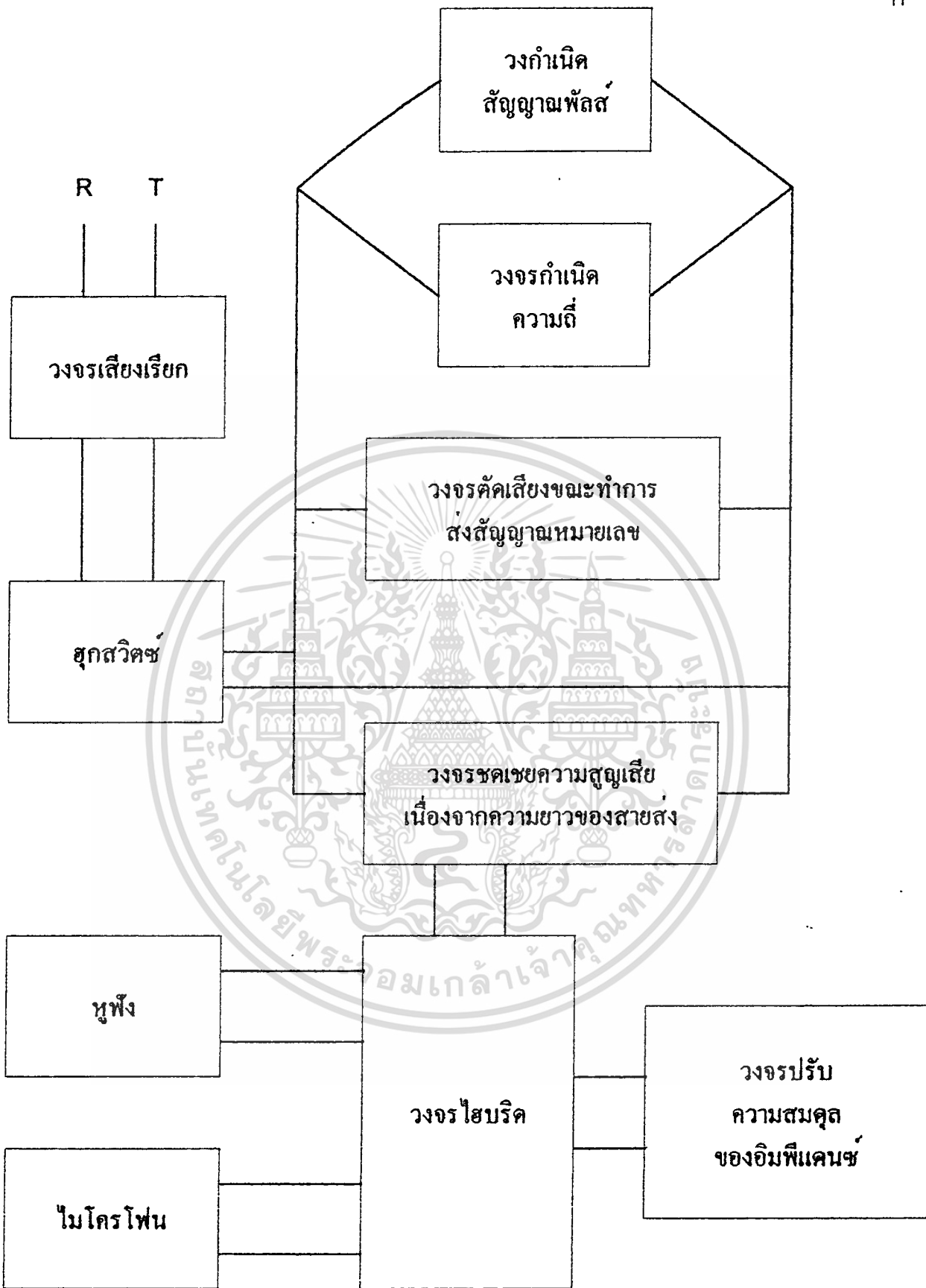
ในระบบโทรศัพท์แบบที่ส่งสัญญาณด้วยจำนวนพัลส์นี้จะถูกกำหนดให้สามารถส่งสัญญาณใน
อัตรา 10 พัลส์ต่อวินาที หรือ 10 pps (pulses per second) และเพื่อความเข้าใจที่ตรงกันในการพิจารณา
สัญญาณที่เกิดขึ้นจึงควรที่จะทราบความหมายของคำต่อไปนี้

1. คาบของสัญญาณพัลส์ (pulse period) = ช่วงเวลาที่วงจรถูกเปิด (break duration) +
ช่วงเวลาที่วงจรถูกปิด (make duration) ซึ่งคาบของสัญญาณพัลส์จะถูกออกแบบให้มีค่า
อย่างต่ำ 100 มิลลิวินาที
2. อัตราการส่งสัญญาณพัลส์ (pulse rate) = จำนวนพัลส์ที่ถูกส่งออกไปใน 1 วินาที =
1000/คาบเวลาของสัญญาณพัลส์ (เป็นมิลลิวินาที)
3. เปอร์เซ็นต์ของการเปิดวงจร (percent break) = 100 X อัตราส่วนการเปิดวงจร (break
ratio) = 100 X ช่วงเวลาที่วงจรถูกเปิด / คาบของสัญญาณพัลส์
4. ช่วงเวลาระหว่างกลุ่มของสัญญาณ (interdigit interval) ถูกกำหนดให้ค่าอย่างต่ำ 700
มิลลิวินาที

สำหรับในสำหรับอเมริกาจะกำหนดค่ามาตรฐานของสัญญาณไว้แน่นอน เช่น ช่วงเวลาที่วงจร
ถูกเปิดจะต้องไม่ต่ำกว่า 60 มิลลิวินาที หรืออัตราการเปิดวงจรเท่ากับ 60% สำหรับประเทศอื่น ๆ
มักจะใช้ที่อัตรา 67% เป็นส่วนใหญ่

ความเพี้ยนของสัญญาณเนื่องจากอุปกรณ์แฝง

ความปกติในสายส่งสัญญาณที่เชื่อมต่อกันระหว่างที่ชุมสายกับเครื่องโทรศัพท์จะมีความต้านทาน
ตัวเก็บประจุ และขดลวดเหนี่ยวนำแฝงอยู่ โดยเฉลี่ยแล้วทุก ๆ ระยะทาง 1 ไมล์ ที่เพิ่มขึ้นของสาย
ส่งจะเสมือนว่ามีตัวเก็บประจุค้ำคร่อมอยู่ระหว่างสายส่ง มีค่าประมาณ 0.07 ไมโครฟารัด และมีตัว
ต้านทานกับขดลวดเหนี่ยวนำค้ำคร่อมกันอยู่ โดยจะมีค่าประมาณ 42 โอห์ม และ 1 มิลลิเฮนรี
ตามลำดับ ซึ่งอุปกรณ์แฝงพวกนี้จะมีผลทำให้สัญญาณ pulse ที่ถูกส่งไปตามสายส่งเกิดความผิดเพี้ยน
ทั้งขนาด (amplitude) และคาบเวลา (period) ดังนั้นชุมสายจึงจำเป็นต้องมีวงจรที่สามารถจะรับรู้
สัญญาณที่ผิดเพี้ยนเหล่านี้ไว้ และไม่ทำให้เกิดความผิดพลาดในการติดต่อ



รูปที่ 8

แสดงบล็อกไออะแกรมของโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรศัพท์แบบส่งสัญญาณความถี่คู่ (Dual tone multifrequency type)

เป็นระบบการส่งสัญญาณอีกแบบหนึ่ง ซึ่งจะพบได้มากกว่าในระบบการส่งเป็นสัญญาณพัลส์ ระบบนี้หรือเรียกชื่อย่อว่า DTMF มีวิธีการส่งหมายเลขของผู้ที่ต้องการจะติดต่อด้วย โดยการส่ง สัญญาณความถี่ 2 ความถี่มอดูเลตกันไป ซึ่งจะเป็นตัวแทนของหมายเลขที่กด ซึ่งความถี่ที่ถูกส่งออกไปจะอยู่ในย่านความถี่ของเสียงพูด (0-4 กิโลเฮิรตซ์) ซึ่งค่าความถี่ที่ต่ำกว่าจะเป็นความถี่ที่แสดงในแนวนอนและอีกค่าหนึ่งก็จะเป็นความถี่ในแนวตั้งซึ่งค่าต่าง ๆ จะแสดงไว้ในรูปที่ 3 ตัวอย่างเช่น เมื่อมีการกดหมายเลข 5 ก็จะมีความถี่ 770 เฮิรตซ์ และ 1336 เฮิรตซ์ มอดูเลตกันออกมา

สำหรับวงจรออสซิลเลเตอร์ที่สร้างความถี่เหล่านี้ขึ้นมาก็คือ วงจรในรูปที่ 4 ซึ่งเป็นวงจรที่ยังคงใช้อุปกรณ์ต่าง ๆ มาต่อรวมกันเป็นวงจรอยู่ ซึ่งปัจจุบันจะมีการใช้อุปกรณ์ที่ถูกลผลิตในรูปไอซีสำเร็จรูปมาใช้งานมากกว่า แต่บทความนี้ต้องการให้ผู้อ่านได้เข้าใจถึงหลักการของระบบจึงหาอวงจรพื้นฐานมาประกอบคำอธิบาย

การทำงานของวงจรนี้จะเริ่มจากสวิตช์ S1 (สวิตช์ในแนวนอน) S2 (สวิตช์ในแนวตั้ง) และ S3 จะถูกเปิดวงจรอยู่เมื่อมีการยกหูโทรศัพท์ขึ้น กระแสจากขุมสายจะผ่าน RV1, L1A และ L2A ทรานซิสเตอร์ Q1 จะไม่นำกระแส เมื่อมีการกดหมายเลข สวิตช์ S1, S2 จะถูกปิดลงตามตำแหน่งของหมายเลขที่ถูกกด C1, C2 จะถูกต่อเข้ากับ L1A และ L2A ตามลำดับ เกิดเป็นวงจรออสซิลเลเตอร์ขึ้น โดย L1A และ C1 จะเป็นออสซิลเลเตอร์ที่ผลิตความถี่ต่ำกว่าความถี่ที่เกิดจาก L2A และ C2 และขณะที่ S3 จะถูกปิดลงเช่นกัน ทำให้ทรานซิสเตอร์ Q1 ทำหน้าที่มอดูเลตสัญญาณจากออสซิลเลเตอร์ทั้งสองเข้าด้วยกันและส่งไปยังขุมสาย ในขณะที่ทำการกดหมายเลขอยู่นั้นส่วนของหูฟังและไมโครโฟนก็จะต่อขนานกัน จึงทำให้ได้ยินสัญญาณที่เกิดขึ้นจากวงจรออสซิลเลเตอร์ด้วย สำหรับทางขุมสายก็จะมียังวงจรความถี่ป้องกันไม่ให้มีความถี่แปลกปลอมอื่น ๆ เข้าไปในขุมสายด้วย

ข้อเปรียบเทียบระหว่างระบบทั้งสอง

คราวนี้ลองเปรียบเทียบระหว่างระบบโทรศัพท์ทั้ง 2 ระบบว่าแบบไหนจะมีประสิทธิภาพมากกว่า ในคอนต้นทราบแล้วว่าการส่งสัญญาณแบบพัลส์ 1 ลูก ต้องใช้เวลายาวอย่างน้อย 100 มิลลิวินาที (60 มิลลิวินาที สำหรับช่วงการเปิดวงจรและ 40 มิลลิวินาที สำหรับช่วงการปิดวงจร) และยังคงมีช่วงเวลาที่แยกสัญญาณมากยิ่งขึ้น ตัวอย่างเช่น หมายเลข 555 - 5555 จะใช้เวลาในการส่งสัญญาณพัลส์ = 5 (พัลส์ / มิลลิวินาที) x 1000 (มิลลิวินาที / พัลส์) x 7 (หมายเลข) = 3.5 วินาที

และระยะเวลาของช่องว่างระหว่างกลุ่มสัญญาณ = 700 (มิลลิวินาที) X 6 = 4.2 วินาที

จะใช้เวลาในการส่งทั้งหมด = 3.5 + 4.2 = 7.7 วินาที

แต่ถ้าเป็นโทรศัพท์ที่ใช้การส่งระบบ DTMF จะใช้เวลาเท่ากับ 7 x 100 มิลลิวินาที = 0.7

วินาทีเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะเห็นได้ชัดเจนประการหนึ่งแล้วว่าระบบ DTMF จะสามารถประหยัดเวลาในการส่งหมายเลขไปยังชุมสายได้มากกว่าระบบที่ใช้การส่งสัญญาณพัลส์ ซึ่งเป็นผลให้ชุมสายสามารถใช้อุปกรณ์ประเภทหน่วยความจำได้เป็นอย่างดีมีประสิทธิภาพมากขึ้นตามไปด้วย

ข้อควรคำนึงในการเชื่อมต่อระบบ DTMF กับสายส่งสัญญาณ

1. ระดับแรงดันและกระแสลูปจะต้องรักษาให้คงที่ตลอดระยะเวลาของสายส่งสัญญาณ
2. ความถี่ที่ถูกผลิตขึ้นจะต้องไม่มีความผิดเพี้ยนทั้งคาบและขนาดของสัญญาณ
3. วงจรออสซิลเลเตอร์จะต้องมีอิมพีแดนซ์ที่สมมูล (matching) กับสายส่ง สัญญาณ ซึ่งจากข้อสรุปข้างบนถ้าจะแยกรายละเอียดของแต่ละหัวข้อดังต่อไปนี้

ความผิดเพี้ยนของสัญญาณ

สำหรับข้อกำหนดของความผิดเพี้ยนของสัญญาณมีดังต่อไปนี้

1. สัญญาณอื่นที่แทรกสอดเข้ามาในสายส่งสัญญาณรวมกันแล้วจะต้องน้อยกว่าระดับของสัญญาณที่ถูกส่งออกไปจริงอย่างน้อย 20 dB
2. สำหรับสัญญาณที่สอดแทรกสอดเข้ามาจะต้องมีข้อกำหนดดังต่อไปนี้
 - 2.1. ต้องมีค่าไม่มากกว่า -33 dBm ในช่วง 300 ถึง 3,400 เฮิรตซ์
 - 2.2. ที่ความถี่ที่สูงกว่า 3,400 เฮิรตซ์ สัญญาณแทรกสอดจะต้องลดลง 12 dB ต่อออกเตฟไปความถี่ 50 กิโลเฮิรตซ์
 - 2.3. ต้องมีระดับสัญญาณไม่มากกว่า -80 dB ที่ความถี่ที่สูงกว่า 50 กิโลเฮิรตซ์ โดยที่ความผิดเพี้ยนถ้าถูกกำหนดในรูปของเดซิเบลคือ

$$\text{ความผิดเพี้ยน} = 20 \log \left[\frac{(v_1^2 + v_2^2 + \dots + v_n^2)^{1/2}}{(v_L^2 + v_H^2)^{1/2}} \right]$$

โดยที่ v_1 ถึง v_n คือ ระดับแรงดันของสัญญาณที่แทรกสอดเข้ามา

v_L คือ ระดับแรงดันของสัญญาณความถี่ต่ำกว่า

v_H คือ ระดับแรงดันของสัญญาณความถี่สูงกว่า

ไดนามิกอิมพีแดนซ์

วงจรถ้าเกิดความถี่สำหรับระบบ DTMF จะต้องอิมพีแดนซ์อย่างน้อย 900 โอห์ม ขณะที่ทำการผลิตความถี่ออกมา และจะต้องมีอิมพีแดนซ์ให้ต่ำที่สุดขณะที่ไม่ทำการผลิตสัญญาณ

ความสูญเสียที่เกิดจากการสะท้อนกลับของสัญญาณ (Return loss)

ก็เป็นพารามิเตอร์อีกตัวที่จะต้องควบคุม โดยกำหนดค่าความสูญเสียในการสะท้อนกลับของสัญญาณหรือ RL ด้วยสมการ

$$RL = 20 \log [(ZL+Zg)/(ZL-Zg)]$$

โดยที่ ZL คือ อิมพีแดนซ์ของสายส่งสัญญาณ

Zg คือ อิมพีแดนซ์ของเครื่องโทรศัพท์

ค่ามาตรฐานสำหรับ RL จะต้องมากกว่า 14 dB ในช่วงความถี่ระหว่าง 300 ถึง 3,400 Hz และมากกว่า 10 dB ในช่วงความถี่ 50 ถึง 300 เฮิรตซ์ และ 3,400 ถึง 20,000 เฮิรตซ์

ข้อดีสำหรับระบบการส่งสัญญาณแบบ DTMF

จากข้อมูลทั้งหมดสามารถสรุปข้อดีของระบบ DTMF ได้คือ

1. ลดระยะเวลาในการส่งหมายเลขโทรศัพท์ไปยังชุมสาย
 2. สามารถใช้วงจรที่ใช้อุปกรณ์โซลิดสเตตได้ ซึ่งจะทำให้เกิดความประหยัด และ
- สะดวก
3. ลดอุปกรณ์จำพวกหน่วยความจำที่ใช้ภายในชุมสาย
 4. สามารถนำไปเชื่อมต่อกับอุปกรณ์ภายในชุมสายอย่างมีประสิทธิภาพ

วงจรที่ได้รับการพัฒนาขึ้น

พื้นฐานในโทรศัพท์แบบหมุนอยู่ จะประกอบไปด้วยไอซี เป็นส่วนใหญ่ ดังนั้นจึงทำให้มีฟังก์ชันที่อำนวยความสะดวกแก่ผู้ใช้มากขึ้นเป็นต้นว่าการเรียกซ้ำหรือมีหน่วยความจำสำหรับเก็บหมายเลขโทรศัพท์ที่ต้องการได้ เป็นต้น

วงจรของระบบที่ใช้การสื่อสารสัญญาณแบบพัลส์

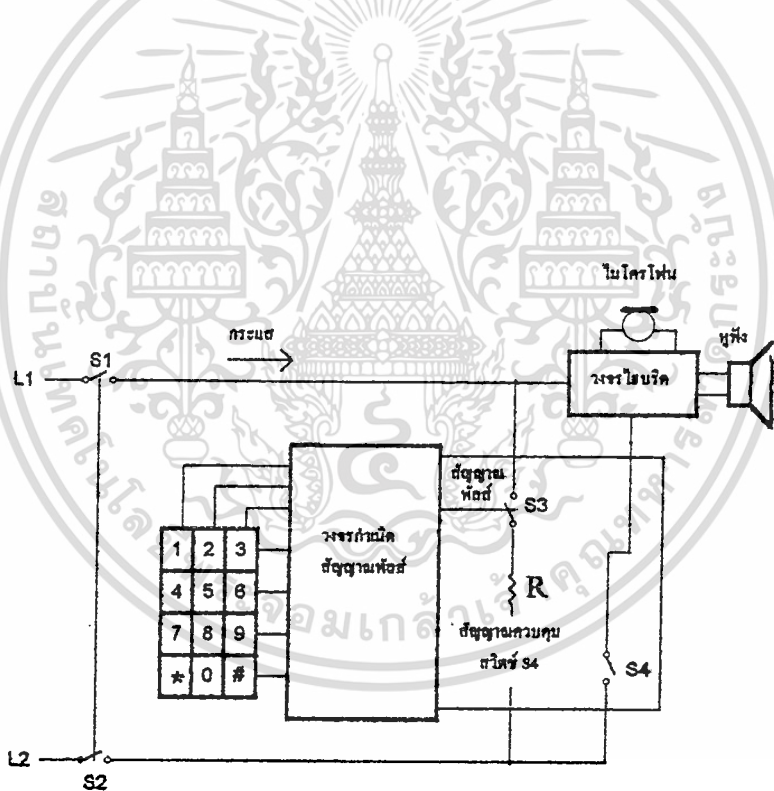
การส่งรหัสแบบพัลส์ก็คือ การชักจูงหวัะการไหลของกระแสตนเอง ดังนั้นวงจรที่ได้รับการพัฒนาแล้วก็ยังคงใช้หลักการเดียวกันอยู่ เพียงแต่ใช้อุปกรณ์ที่เป็นไอซี แทนอุปกรณ์ทางกล (mechanical) โคธ วงจรที่จะกล่าวถึงนี้ขอเรียกว่า วงจรกำเนิดสัญญาณพัลส์ (pulse dialer) ซึ่งยังคงสามารถสร้างสัญญาณด้วยความเร็ว 10 พัลส์ต่อวินาที และจะต้องมีวงจรที่ทำหน้าที่ป้องกันไม่ให้เกิดเสียงเล็ดลอดเข้าไปสู่หูของผู้ฟัง เนื่องจากการส่งสัญญาณพัลส์ จากวงจรที่ได้ถูกออกแบบใหม่นี้จะมีอยู่ 2 แบบด้วยกัน

แบบแรกจะเป็นวงจรที่ออกแบบให้วงจรหมายเลข (dial circuit) ต่อขนานกับวงจรไฮบริด (hybrid) หรือวงจรเสียงพูด (speech network)

สำหรับวงจรไฮบริดหรือวงจรเสียงพูดก็คือ วงจรที่ใช้สำหรับเชื่อมต่อ (interface) ระหว่างระบบ 4 สายเข้ากับระบบ 2 สาย โดยเมื่อเชื่อมต่อเข้าด้วยกันแล้วทำให้โทรศัพท์สามารถสื่อสารแบบดูเพล็กซ์ (duplex) ได้ หรือติดต่อสื่อสารแบบสวนทางกันได้ในเวลาเดียวกันนั่นเอง

วงจรในเครื่องโทรศัพท์จะใช้ระบบ 4 สาย คือ ใช้ 2 สายสำหรับส่งสัญญาณเสียงพูดและอีก 2 สายสำหรับนำเข้าสู่หูฟัง แต่ในการเชื่อมต่อระหว่างเครื่องโทรศัพท์กับชุมสายระดับท้องถิ่นจะเป็นการติดต่อโดยผ่านสาย T และ R เพียง 2 สายและ 4 สายให้สามารถติดต่อกันได้นั่นเอง

นอกเหนือจากวงจรภายในเครื่องโทรศัพท์จะมีขดลวดเหนี่ยวนำต่อเป็นวงจรไฮบริดแล้ว การติดต่อระหว่างชุมสายท้องถิ่นด้วยกันก็จำเป็นต้องมีวงจรที่ทำหน้าที่ปรับความสมดุลอิมพีแดนซ์ของระบบที่จะถูกเชื่อมต่อเข้าด้วยกัน เพื่อให้สัญญาณที่ใช้ในการติดต่อสื่อสารกันมีประสิทธิภาพที่สุด



รูปที่ 4

วงจรหมายเลขต่อแบบขนานกับวงจรไฮบริด

เมื่อทราบถึงหน้าที่ของวงจรไฮบริดแล้วก็ย้อนกลับมาดูวงจรหมุนหมายเลขกันต่อ ในรูปที่ 4 สวิตช์ S3 จะทำหน้าที่ขัดจังหวะการไหลของกระแส ทำให้เกิดเป็นสัญญาณพัลส์ขึ้น ในรูปจะเห็นว่า สวิตช์ S3 จะขนาบอยู่กับวงจรไฮบริด สวิตช์ S4 จะเป็นตัวควบคุมไม่ให้เราได้ยินเสียงของสัญญาณพัลส์ที่เกิดขึ้น ขั้นตอนการทำงานเริ่มจากตอนที่กดโทรศัพท์ขึ้น สวิตช์ S4 จะปิดวงจร ส่วนสวิตช์ S3 จะเปิดวงจรออก กระแสจะไหลผ่านวงจรไฮบริด (ซึ่งเป็นขดลวดเหนี่ยวนำ) ทำให้ผู้ใช้โทรศัพท์สามารถได้ยินสัญญาณหมุน เมื่อผู้ใช้โทรศัพท์เริ่มกดหมายเลข วงจรกำเนิดสัญญาณพัลส์จะส่งสัญญาณออกไปควบคุมให้สวิตช์ S3 ปิดวงจร และ S4 เปิดวงจร ดังนั้นจึงทำให้กระแสไหลผ่านวงจรกำเนิดสัญญาณพัลส์แต่จะไม่ไหลผ่านวงจรไฮบริด จึงเสมือนว่าส่วนของหูฟังถูกตัดออกจากวงจร ผู้ใช้โทรศัพท์จึงไม่ได้ยินเสียงที่เกิดจากการส่งสัญญาณพัลส์ในระหว่างที่ทำการส่งสัญญาณอยู่ สวิตช์ S3 จะถูกปิดลงเป็นจำนวนครั้งตามหมายเลขที่กด เมื่อสิ้นสุดการกดหมายเลขแล้ว สวิตช์ S3 จะอยู่ในตำแหน่งเปิดวงจรและสวิตช์ S4 จะปิดวงจรตามเดิม

สำหรับวงจรอีกแบบหนึ่งจะเป็นวงจรที่ออกแบบให้วงจรหมุนหมายเลขโดยต่ออนุกรมกับวงจรไฮบริด

สวิตช์ S3 จะถูกต่ออนุกรมกับวงจรไฮบริดอยู่ โทรศัพท์ระบบนี้จะใช้สัญญาณจากวงจรกำเนิดสัญญาณพัลส์มาควบคุมสวิตช์ S3 ให้เปิดปิดวงจรเพื่อทำให้เกิดเป็นสัญญาณพัลส์ขึ้น ซึ่งเหมือนกับระบบแรกที่ได้กล่าวไปแล้ว ขณะเดียวกันก็จะมีสัญญาณควบคุมให้สวิตช์ S4 เปิดวงจรเพื่อไม่ให้เกิดสัญญาณออกทางหูฟังขณะที่ทำการกดหมายเลขอยู่ จะเห็นว่าหลักการการทำงานของทั้ง 2 ระบบจะเหมือน ๆ กัน จะแตกต่างกันก็ตรงลักษณะการเชื่อมต่อวงจรเท่านั้นเอง

หน้าที่ของไอซีกำเนิดสัญญาณพัลส์และสัญญาณควบคุมต่าง ๆ กันดูบ้าง เอาต์พุตของไอซีจะเป็นสัญญาณพัลส์และสัญญาณควบคุมที่จำเป็นต่อวงจร ส่วนสัญญาณอินพุตจะเป็นรหัสไบนารี ซึ่งใช้แทนหมายเลขโทรศัพท์, สัญญาณให้ถือหูรอก, สัญญาณหมุนซ้ำ เป็นต้น วงจรออสซิลเลเตอร์จะถูกใช้เพื่อสร้างสัญญาณนาฬิกา (clock) ให้กับวงจรภายในไอซี สำหรับภาคขับสัญญาณเอาต์พุต จำเป็นต้องใช้ทรานซิสเตอร์ที่ทนแรงดันสูง ๆ ได้ และต้องสามารถทำงานที่ความถี่สูง ๆ ได้

สัญญาณลوجิกที่ควบคุมช่วงเวลาในการส่งสัญญาณที่มีความจำเป็นก็คือ สัญญาณอนสกุ / ออฟสกุ สัญญาณบอกช่วงเวลาระหว่างกลุ่มของสัญญาณพัลส์ และสัญญาณกำหนดอัตราส่วนการเปิด / ปิดวงจร สำหรับสัญญาณจากวงจรออสซิลเลเตอร์นั้นจะไม่สามารถผลิตความถี่ออกมาได้ถ้าวงหูโทรศัพท์ไว้ตำแหน่งปกติ

สัญญาณที่กำหนดช่วงห่างระหว่างกลุ่มของสัญญาณพัลส์ หรือสัญญาณ IDT จะกำหนดช่วงห่างของเวลาได้ระหว่าง 200 ถึง 1,000 มิลลิวินาที เมื่ออัตราในการส่งสัญญาณพัลส์อยู่ที่ 10 พัลส์ต่อวินาที และอยู่ระหว่าง 100 ถึง 500 มิลลิวินาที ถ้าส่งสัญญาณพัลส์ในอัตรา 20 พัลส์ต่อวินาที ส่วนค่าของอัตราส่วนการเปิด / ปิดวงจรจะมีค่าประมาณ 67% ก็ได้ เช่นกัน

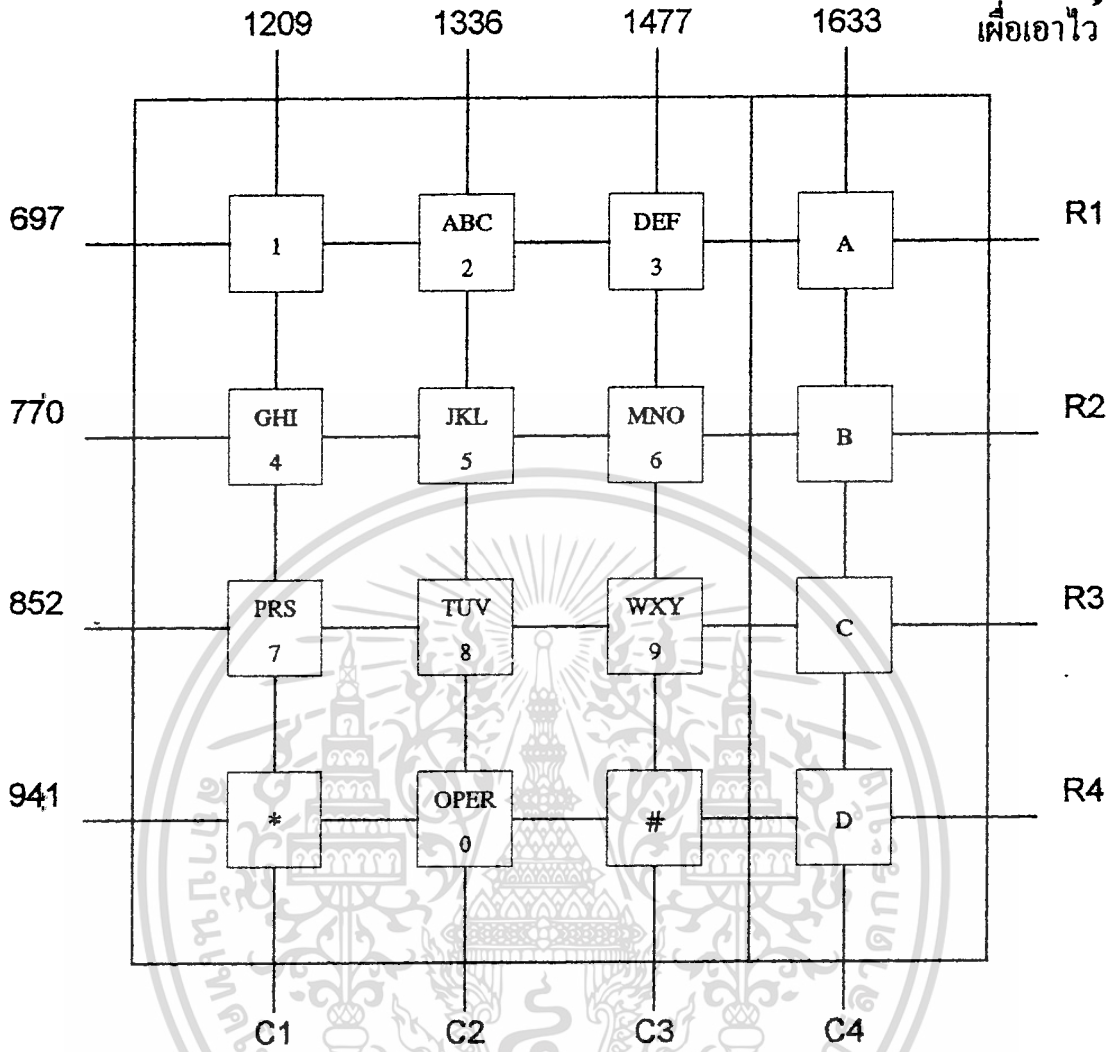
ในปัจจุบัน ไอซีที่ทำหน้าที่กำเนิดสัญญาณพัลส์ก็มีเบอร์ MC14408 ซึ่งเป็นของ Motorola โครงสร้างภายในเป็นแบบ CMOS และเนื่องจากการใช้งานต้องการสัญญาณอินพุตเป็นรหัสไบนารี จึงจำเป็นต้องมีตัวเข้ารหัส (encoder) เบอร์ MC 14419 มาอินเตอร์เฟสเข้ากับแป้นกดหมายเลขด้วย ซึ่งเป็นวงจรที่ได้รับการพัฒนาให้สามารถนำมาใช้งานได้ สามารถอธิบายได้ด้วยไดอะแกรมของเวลา (timing diagram) ซึ่งจะช่วยให้เข้าใจได้ดียิ่งขึ้น

เมื่อมีการกดปุ่มโทรศัพท์ขึ้น สัญญาณแสดงความต้องการจะใช้โทรศัพท์ (call request : CRQ) จะเป็นลอจิก “0” การที่ สัญญาณ CRQ เป็นลอจิก “0” ก็เท่ากับเป็นการรีเซ็ตให้ MC 14408 เริ่มต้นที่จะรับหมายเลขใหม่เข้ามาหรืออาจจะให้ส่งสัญญาณพัลส์ของหมายเลขชุดเก่าเมื่อมีการใช้ฟังก์ชันหมุนซ้ำ หลังจากนั้นสัญญาณแสดงการหมุน (dial rotating output : DRO) ซึ่งเป็นลอจิก “1” จะถูกส่งออกไป สัญญาณ DRO นี้เองก็จะนำออกไปใช้ในการคิดเลขที่เกิดขึ้นในระหว่างการส่งสัญญาณพัลส์ออกไปโดยการทำให้ทรานซิสเตอร์ MPS - L01 ไม่นำกระแสตลอดเวลาที่สัญญาณ CRO ยังคงเป็นลอจิก “1” อยู่สำหรับสัญญาณ (output pulse : OPL) จะถูกนำไปอินพุตให้กับไดโอดและทรานซิสเตอร์เพื่อส่งสัญญาณพัลส์ออกไปอีกทีหนึ่ง หลังจากนั้นก็จะทำหน้าที่ของทุบสายที่จะรับข้อมูลไปและติดต่อกับหมายเลขที่ถูกเรียกให้กับผู้ใช้ให้ถูกต้อง

กลุ่มความถี่สูง(แนวตั้ง)

ปุ่มที่ได้รับการ
เพื่อเอาไว้

กลุ่มความ
ถี่ต่ำ
(แนวอน)



รูปที่ 5

แสดงเป็นคณมาเลขและค่าความถี่ในแนวอนและแนวตั้งของมาเลขนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีสมาชิกในตระกูลหลายเบอร์ด้วยกัน แต่ละเบอร์จะมีคุณสมบัติพิเศษแตกต่างกัน เช่น มีหน่วยความจำภายในสำหรับเก็บโปรแกรมและข้อมูลภายในชิปเพิ่มขึ้น มีวงจรเปลี่ยนค่าสัญญาณอนาล็อกเป็นดิจิทัลในตัว สามารถรับสัญญาณอินเทอร์รัปต์ได้หลายชนิด ทำกระบวนการ DMA (Direct Memory Access) ได้ในตัว มีรีจิสเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์เพิ่มขึ้น

ไมโครคอนโทรลเลอร์เบอร์ที่นับว่าเป็นเบอร์พื้นฐานสำหรับตระกูล MCS-51 นี้ได้แก่เบอร์ 8051, 8031, 8751 โดยเบอร์ 8051 จัดเป็นสมาชิกตัวแรกของตระกูลมีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเป็น ROM ขนาด 4 กิโลไบต์ และหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายใน MCS-51 (RAM) เองจำนวน 128 กิโลไบต์ มีพอร์ตขนาน 8 บิต 4 พอร์ต มีรีจิสเตอร์สำหรับไทม์เมอร์หรือเคาน์เตอร์ขนาด 16 บิตรวม 2 ตัวรับสัญญาณอินเทอร์รัปต์จากภายนอกได้ 2 ชนิด สามารถรับและส่งข้อมูลแบบอนุกรมผ่านทางพอร์ตสื่อสารข้อมูลแบบอนุกรม มีวงจรออสซิลเลเตอร์สำหรับสร้างสัญญาณนาฬิกาสำหรับควบคุมการทำงานในตัวเอง

ส่วนเบอร์ 8751 จะมีคุณสมบัติเหมือนกับเบอร์ 8051 ทุกอย่าง ต่างกันที่ชนิดหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปของเบอร์ 8751 จะเป็น EPROM แทนที่จะเป็น ROM ส่วนเบอร์ 8031 จะเหมือนกับเบอร์ 8051 ต่างกันเพียง ในเบอร์ 8031 จะไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป

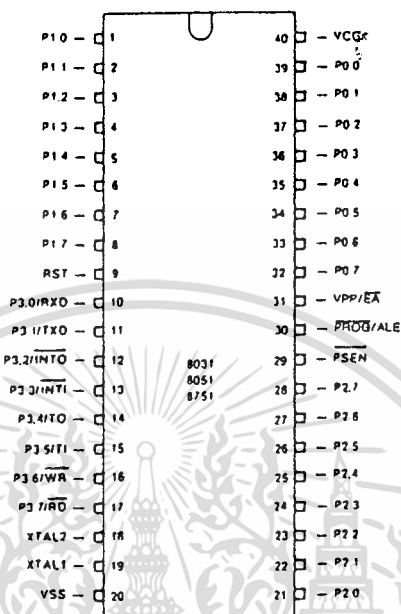
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์ใช้แรงดันไฟฟ้าเพียง 5 โวลต์ในการทำงาน ส่วนกระแสไฟฟ้าที่ใช้จะแตกต่างกันไปตามชนิดของเทคโนโลยีที่ใช้ในการผลิต ไมโครคอนโทรลเลอร์ตระกูลนี้ที่มีอักษร C อยู่ตรงกลางเบอร์เช่น 80C31, 80C51 จะเป็นเบอร์ของชิปที่ผลิตโดยอาศัยเทคโนโลยี CMOS ซึ่งใช้พลังงานในการทำงานน้อยกว่าและสามารถควบคุมการใช้พลังงานของชิปได้จากโปรแกรมเพื่อการประหยัดพลังงานในระบบ

MCS-51 เป็นตระกูลไมโครคอนโทรลเลอร์ที่ถูกพัฒนามาจากตระกูล MCS-48 ดังนั้นจึงมีความสามารถเหนือกว่าหลายอย่าง

ตำแหน่งขาของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกันดัง

รูปที่ 6



รูปที่ 6

แสดงการกำหนดขาของ 8051

หน้าที่การใช้งานแต่ละขาของชิปไมโครคอนโทรลเลอร์ในตระกูล MCS-51 มีดังนี้

- ขา Vss (ขา 20) สำหรับต่อลงกราวด์
- ขา Vcc (ขา 40) สำหรับต่อแหล่งจ่ายแรงดันกระแสตรงขนาด 5 โวลต์ (DC 5 Volt)
- ขาพอร์ต 0 (ขา 32 - 39) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 0 ขนาด 8 บิต (P0.0 - P0.7) แบบ Open Drain Bidirectional พอร์ตนี้สามารถใช้งานเป็นอินพุต-เอาต์พุต พอร์ตทั่วไปได้ โดยหากใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้เพื่อบังคับให้ขาอยู่ในลักษณะถูกปล่อย (มีสถานะ High impedance) นอกจากนี้ ใช้งานเป็นอินพุต-เอาต์พุตแล้วพอร์ตเบอร์ 0 นี้ยังใช้ในการติดต่อกับหน่วยความจำ สำหรับเก็บโปรแกรมและข้อมูลภายนอกชิปอีกด้วย โดยส่งค่าแอดเดรสไบท์ค่า (A0-A7) และผลิตเฟลตซ์กับการส่งข้อมูล (D0-D7) จากหน่วยความจำภายนอก ในระหว่างการเขียนหรือการอ่านข้อมูลโดยมีวงจรถูกฝังภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขาพอร์ต 1 (ขา 1 - 8) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 1 ขนาด 8 บิต (P1.0 - P1.7) สามารถใช้งานเป็นอินพุตหรือเอาต์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็นอินพุต พอร์ตต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้เพื่อให้มีสถานะ High impedance โดยมีวงจรถูล์อ์ภายในขา P1.0, P1.1 ในเบอร์ 8052 จะใช้งานหน้าที่อย่างอื่นนอกเหนือจากอินพุต-เอาต์พุตพอร์ตทั่วไปด้วย
- ขาพอร์ต 2 (ขา 21 - 28) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 2 ขนาด 8 บิต (P2.0 - P2.7) สามารถใช้งานเป็นอินพุต-เอาต์พุต พอร์ตทั่วไปได้ โดยหากใช้งานเป็นอินพุต พอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้เพื่อบังคับให้ขาอยู่ในลักษณะถูกปล่อย (มีสถานะ High impedance) นอกจากใช้งานเป็นอินพุต-เอาต์พุตแล้วพอร์ตเบอร์ 2 นี้ยังใช้ในการติดต่อกับหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลภายนอกชิปอีกด้วย โดยส่งค่าแอดเดรสไบท์สูง (A8-A15) และมีวงจรถูล์อ์ภายใน
- ขาพอร์ต 3 (ขา 10-17) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 3 (P3.0-P3.1) สามารถใช้งานเป็นอินพุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็นอินพุตพอร์ตต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้เพื่อให้มีสถานะเป็น High impedance โดยใช้วงจรถูล์อ์ภายใน นอกจากนี้ยังสามารถใช้งานในหน้าที่พิเศษต่างๆอีกหลายอย่างดังนี้
 - ขา P3.0 ใช้รับข้อมูลภายนอกแบบอนุกรม
 - ขา P3.1 ใช้ส่งข้อมูลออกไปภายนอกแบบอนุกรม
 - ขา P3.2 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเทอร์รัปต์ชนิดที่ 0
 - ขา P3.3 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเทอร์รัปต์ชนิดที่ 0
 - ขา P3.4 สัญญาณอินพุตให้เคาน์เตอร์ของ ไทม์เมอร์ 0
 - ขา P3.5 สัญญาณอินพุตให้เคาน์เตอร์ของ ไทม์เมอร์ 1
 - ขา P3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป
 - ขา P3.7 ใช้เป็นสัญญาณควบคุมการอ่านข้อมูลไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป

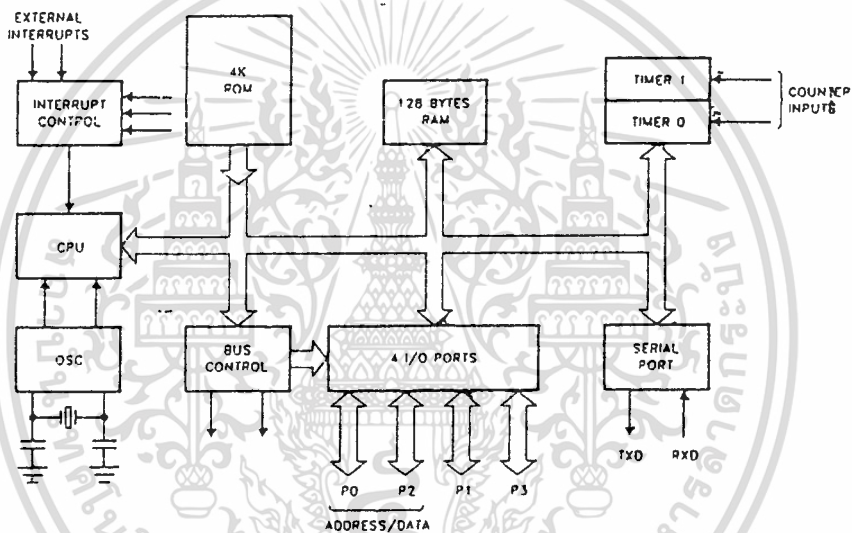
การใช้งานพอร์ตในหน้าที่พิเศษดังกล่าวนี้จะต้องโหลดค่า 1 ไปยังแต่ละบิตที่ต้องการใช้ก่อนทุกครั้ง

- ขา RST (ขา 9) ใช้สำหรับรีเซ็ตวงจรทุกอย่างภายในชิปเพื่อเริ่มต้นการทำงานใหม่ การรีเซ็ตใช้เมื่อเริ่มจ่ายพลังงานหรือเมื่อโปรแกรมทำงานผิดพลาด เมื่อต้องการรีเซ็ตชิป MCS-51 ขานี้ต้องมีสถานะ 1 เป็นเวลาอย่างน้อย 2 คาบเวลา (Machine cycle) ของ CPU ระหว่างที่แหล่งกำเนิดความถี่(Oscillator) ยัง

ทำงานอยู่ โดยต้องต่อตัวต้านทานค่า 8.2 กิโลโอห์มเพื่อทำหน้าที่พูลดาวน์ (รักษาค่าแรงดันไฟฟ้าให้มีสถานะเป็นกราวด์) และกรณีที่ให้ตัวชิปรีเซตเอง เมื่อเริ่มจ่ายพลังงานต้องต่อตัวเก็บประจุขนาด 10 ไมโครฟารัดคร่อมระหว่างขา RST กับ Vcc

- ขา ALE/PROG (ขา 30) เป็นขาสำหรับใช้ส่งสัญญาณออกไปภายนอก เพื่อควบคุมการ Latch ค่าแอดเดรสไบต์ต่ำจากพอร์ท 0 ในระหว่างการติดค้อหน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอก ปกติเมื่อไม่มีการติดค้อหน่วยความจำภายนอก ขานี้จะส่งสัญญาณพัลส์ออกมาด้วยความถี่ 1/8 ของความถี่ออสซิลเลเตอร์ที่ใช้ตลอดเวลา ดังนั้นเราจึงสามารถใช้ความถี่ที่ได้จากขานี้นำไปใช้งานเพื่อควบคุมอย่างอื่นได้ แต่เมื่อมีการติดค้อกับหน่วยความจำสำหรับเก็บข้อมูลที่อยู่นอกชิปความถี่ที่ขานี้จะลดลงครึ่งหนึ่ง นอกจากนี้ขา ALE ยังใช้สำหรับควบคุมการเขียนโปรแกรมลงใน EPROM สำหรับ MCS-51 เบอร์ที่มีหน่วยความจำสำหรับเก็บโปรแกรมลงในชิปเป็น EPROM
- ขา PSEN (ขา 29) ใช้ส่งสัญญาณสไตรบเพื่ออ่านคำสั่งจากโปรแกรมที่เก็บไว้ในหน่วยความจำภายนอกชิป เมื่อชิปทำงานด้วยโปรแกรมจากภายนอกขานี้จะส่งสัญญาณสไตรบสองครั้งในแต่ละสัญญาณนาฬิกาของ CPU แต่ในช่วงการเขียนหรือการอ่านหรืออ่านข้อมูลกับหน่วยความจำภายนอกหรือเมื่อใช้โปรแกรมจากหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปจะไม่มีสัญญาณออกมาจากขานี้
- ขา EA/Vpp (ขา 31) เป็นขาสำหรับใช้เลือกให้ MCS-51 ทำงานจากโปรแกรมที่อยู่ในหรือภายนอกชิป โดยหากขานี้มีสถานะเป็น 0 หมายถึงให้ใช้โปรแกรมจากหน่วยความจำที่เก็บโปรแกรมภายนอก และหากขานี้มีสถานะเป็น 1 หมายถึงบังคับให้ MCS-51 ใช้โปรแกรมจากหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป และสำหรับ MCS-51 เบอร์ที่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในหรือจากโปรแกรมที่เก็บไว้ในหน่วยความจำภายนอกชิปด้วยการต่อขา EA กับไฟเลี้ยงหรือกราวด์ตามลำดับ ส่วนใน MCS-51 เบอร์ที่ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปให้ค้อขานี้ลงกราวด์เสมอ
- ขา XTAL1 (ขา 19) ใช้การต่อคริสตอลภายนอก โดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์

- ขา XTAL2 (ขา 18) ใช้การต่อคริสตัลภายนอก โดยเป็นอินพุตออกจากวงจรรอสจิลเลเตอร์



รูปที่ 7

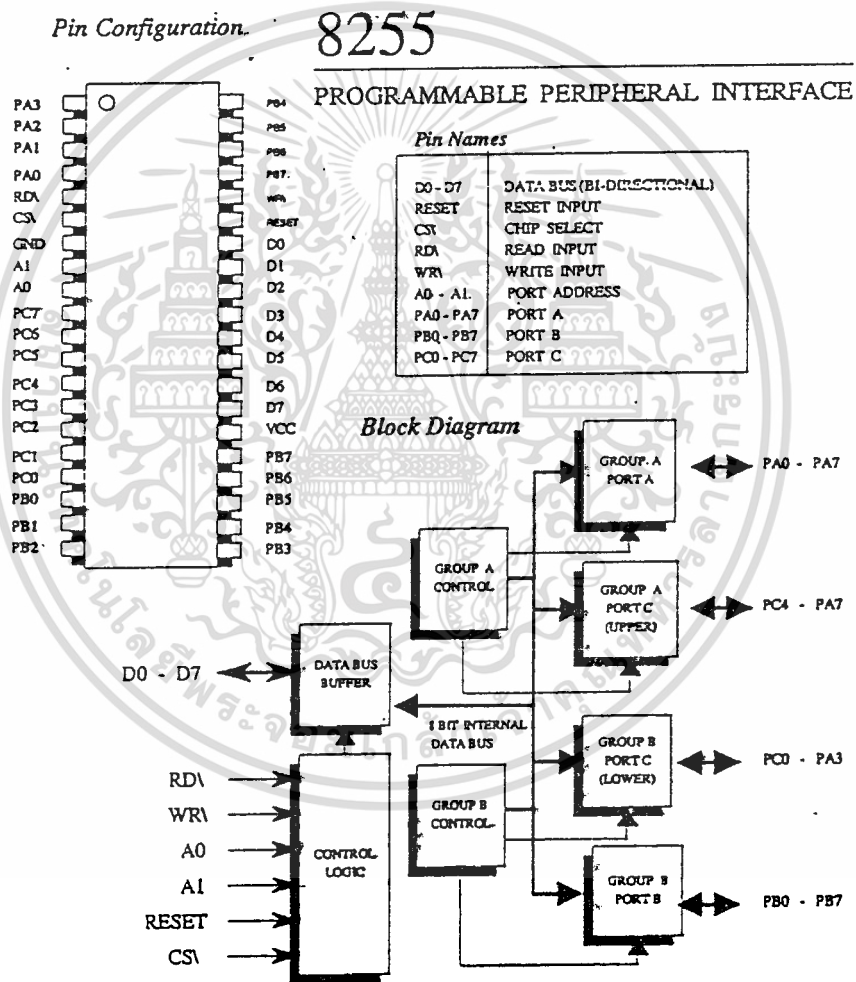
ลักษณะโครงสร้างของ MCS-51

การใช้งาน 8255 กับ 8051

การขยายเพิ่มพอร์ตอินพุต/เอาต์พุตของ 8051 นอกจากจะสามารถใช้ IC ประเภท Latch และบัฟเฟอร์ ประกอบกันเข้ากับบัสของระบบแล้วยังสามารถใช้ IC ประเภทวงจรรวมความสูงหรือ LSI เมอร์ 8255 ซึ่งสามารถทำหน้าที่ได้ทั้งพอร์ตอินพุตหรือเอาต์พุตตามคำสั่งในโปรแกรม ทำให้มีความอ่อนตัวอย่างมากในการนำไปใช้งานได้ตามความประสงค์ เนื้อหาต่อไปนี้จะกล่าวแนะนำถึงลักษณะพื้นฐานของไอซีชนิดนี้ รวมถึงการโปรแกรมเพื่อใช้งานและการเชื่อมต่อกับบัสของ 8051 ซึ่งจะทำได้สามารถนำไปประยุกต์ใช้ได้อย่างเหมาะสมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะพื้นฐานของ 8255



รูปที่ 8

แผนภาพแบบบล็อกภายในและขาต่อของไอซีเบอร์ 8255

ไอซีเบอร์ 8255 ถูกออกแบบมาเพื่อที่จะเป็นพอร์ตสำหรับการรับส่งข้อมูลแบบขนานระหว่างอุปกรณ์ภายนอกกับไมโครคอนโทรลเลอร์ ความอ่อนตัวในการนำไปใช้ของ 8255 เกิดขึ้นเนื่องจาก การที่เราสามารถเปลี่ยนแปลงลักษณะการทำงานของพอร์ต ให้เป็นอินพุทหรือเอาต์พุตได้อย่างสะดวกเพียงการส่งข้อมูลควบคุมจากไมโครคอนโทรลเลอร์ให้กับ 8255 ก่อนที่จะเริ่มต้นใช้งานพอร์ตเท่านั้น โดยที่ความสามารถเหล่านี้จะถูกเรียกว่า Programmable คือสามารถโปรแกรมการทำงานได้ทำให้ได้รับความนิยมกันอย่างแพร่หลาย

จากแผนภาพในรูปที่ 8 จะเห็นได้ว่า 8255 ประกอบด้วยบล็อกของหน่วยการทำงานหลายส่วน ภายในบล็อกทางขวามือทั้ง 4 บล็อกเป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกโดยตรง โดยผ่านทางเส้นสัญญาณที่ระบุชื่อว่า PA0-PA7, PB0-PB7 และ PC0-PC7 กลุ่มของสัญญาณเหล่านี้จำแนกเป็น 3 กลุ่มคือพอร์ต A, B และ C สำหรับบล็อกถัดเข้ามาบริเวณส่วนกลางมีชื่อว่า Group A Control และ Group B Control ทำหน้าที่กำหนดการทำงานของพอร์ตทั้งสาม(ซึ่งจะอธิบายในช่วงถัดไป)บล็อกทั้งสองนี้เชื่อมต่อกับบล็อกอื่นๆผ่านทางบัสข้อมูลภายใน 8255 เอง สำหรับบล็อกการทำงานทางด้านซ้ายที่มีชื่อว่า Data bus buffer และ read/write control logic ทำหน้าที่เชื่อมต่อระหว่างระบบบัสของไมโครคอนโทรลเลอร์กับ 8255 เพื่อรับส่งข้อมูลระหว่างกันตามระดับของลอจิกของขาสัญญาณ RDA และ WR\ ตามลำดับ

การจำแนกกลุ่มของพอร์ต 8255

ในบรรดาพอร์ตทั้งสามของ 8255 มีชื่อเรียกว่าพอร์ต A, B และ C โดยพื้นฐานนั้นล้วนเป็นพอร์ตแบบขนานที่ประกอบด้วยขาสัญญาณ 8 เส้น(8 bits) ซึ่งแต่ละเส้นจะแทนบิตข้อมูลของพอร์ต(พอร์ตข้อมูลขนาน 8 บิต) ซึ่งเราสามารถอ้างถึงแต่ละบิตของเส้นสัญญาณพอร์ตนี้ได้โดยอิสระ อย่างไรก็ตาม 8255 ได้ทำการแบ่งกลุ่มของพอร์ตเหล่านี้เป็น 2 กลุ่มคือ กลุ่ม A และกลุ่ม B เพื่อประโยชน์ในการกำหนดรูปแบบการทำงานของพอร์ตดังตารางต่อไปนี้

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

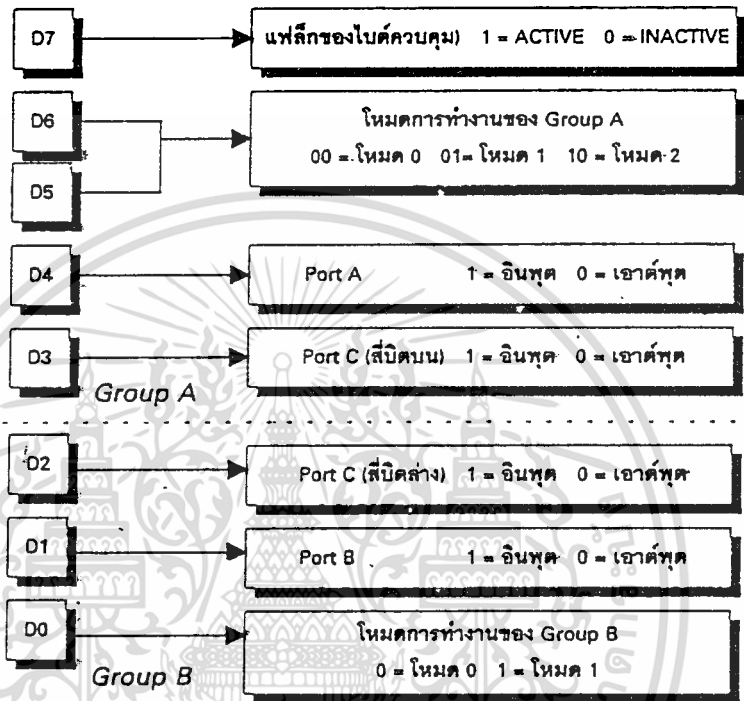
จากตารางข้างต้นจะเห็นได้ว่าจำนวนสัญญาณทั้งหมดของพอร์ต C (PC0-PC7) ได้ถูกแยกออกเป็นสองกลุ่ม คือกลุ่มของ 4 บิตล่าง(Lower nibble)จาก PC0-PC3 และกลุ่มของ 4 บิตบน(Upper nibble)จาก PC4-PC7 ดังนั้นพอร์ตกลุ่ม A และกลุ่ม B ของ 8255 จึงมีจำนวนบิตในแต่ละกลุ่มเป็นจำนวนถึง 12 บิต

สัญญาณ	ความหมาย
D0-D7	กลุ่มของเส้นสัญญาณข้อมูลของ 8255 เมื่อมีการเขียนหรืออ่าน
CS\	สัญญาณเลือกอุปกรณ์ เมื่อขาของสัญญาณนี้เป็นระดับลอจิกต่ำ CPU ก็จะสามารถเขียนหรืออ่านข้อมูลจาก 8255 ได้
RDA	สัญญาณบอกสถานะต้องการอ่านข้อมูลจากรีจิสเตอร์ของ 8255
WR\	สัญญาณบอกสถานะต้องการเขียนข้อมูลให้กับ รีจิสเตอร์ของ 8255
A0-A1	สัญญาณระบุตำแหน่งรีจิสเตอร์ภายใน 8255 ที่ต้องการ
RESET	สัญญาณการรีเซ็ตวงจรทำงานภายใน 8255 เพื่อเริ่มต้นใหม่
PA0-PA7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต A
PB0-PB7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต B
PC0-PC7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต C

รูปแบบคำสั่งเพื่อกำหนดการทำงานของ 8255

การกำหนดให้พอร์ตทั้งสามของ 8255 ทำงานในลักษณะต่างๆกันหรือที่เรียกว่าโหมด (MODE)การทำงานจะเริ่มจากเราจะต้องส่งค่าข้อมูลขนาดหนึ่ง ไบท์ให้กับรีจิสเตอร์ควบคุมการทำงานภายใน 8255 ข้อมูลนี้จะเรียกว่า ไบท์ข้อมูลควบคุม(Control word) โดยแต่ละบิตของข้อมูลนี้จะมีความหมายที่จะระบุถึงความต้องการต่างกันไป ดังที่แสดงรูปที่ 9 การส่งข้อมูลไบท์นี้จะต้องเริ่มต้นเป็นลำดับแรกก่อนที่จะมีการดำเนินการใดๆกับ 8255 ทั้งสิ้น

ตามความหมายของบิตภายในตารางของรูปที่ 9 จะเห็นได้ว่าการเลือกให้พอร์ตใดทำหน้าที่เป็นพอร์ตอินพุตก็เพียงแต่กำหนดค่าข้อมูลเท่ากับ 1 ให้กับบิตที่เกี่ยวข้องกับกับพอร์ตนั้น หรือในกรณีตรงกันข้ามสำหรับการกำหนดให้เป็นพอร์ตเอาต์พุตก็ต้องกำหนดค่าให้เป็น 0 ใดๆก็ตามการกำหนดให้ไบท์ข้อมูลควบคุมนี้มีผลอย่างถูกต้องก็ต้องกำหนดให้บิต D7 มีค่าเป็น 1 เสมอ ส่วนรายละเอียดของบิต D5-D6 และ D2 ที่บอกถึงโหมดการทำงานนั้นจะได้อธิบายในรายละเอียดต่อไป



รูปที่ ๑

ความหมายของบิตภายในไบทข้อมูลควบคุมสำหรับ 8255

การเชื่อม 8255 กับ 8051

เมื่อพิจารณาแผนภาพของ 8255 จะเห็นได้ว่ามีขาสัญญาณแอดเดรสจำนวนสองเส้นคือ A0 และ A1 ทำให้ตำแหน่งของแอดเดรสที่จะอ้างถึงได้มีค่าเป็น 2² หรือเท่ากับ 4 ตำแหน่งซึ่งแต่ละตำแหน่งจะมีความหมายใช้ในการระบุรีจิสเตอร์หรือพอร์ตภายใน 8255 ดังแสดงได้ดังนี้

A1	A0	ชื่อรีจิสเตอร์
0	0	พอร์ต A
0	1	พอร์ต B
1	0	พอร์ต C
1	1	รีจิสเตอร์ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาค่าของแอดเดรสเหล่านี้ร่วมกับระดับลอจิกของขาสัญญาณ RD\ และ WR\ จะเป็นการอ่านหรือเขียนข้อมูลทางขาสัญญาณ D0-D7 ให้กับรีจิสเตอร์นั้นตามลำดับดังตารางต่อไปนี้

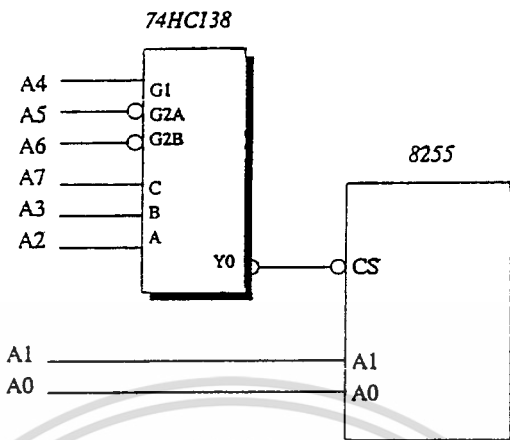
RD\	WR\	A1	A0	ความหมาย
0	1	0	0	ส่ง(หรือเขียน) ข้อมูลให้กับพอร์ต A
1	0	0	0	รับ(หรืออ่าน) ข้อมูลจากพอร์ต A
0	1	0	0	ส่ง(หรือเขียน) ข้อมูลให้กับพอร์ต B
1	0	0	1	รับ(หรืออ่าน) ข้อมูลจากพอร์ต B
0	1	1	0	ส่ง(หรือเขียน) ข้อมูลให้กับพอร์ต C
1	0	1	0	รับ(หรืออ่าน) ข้อมูลจากพอร์ต C
0	1	1	1	ส่ง(หรือเขียน) ข้อมูลให้กับรีจิสเตอร์ควบคุม
1	0	1	1	เป็นสถานะที่ไม่ถูกต้อง

ดังนั้นโดยทั่วไปมักจะกำหนดให้แอดเดรสของ 8255 ทั้งสี่ตำแหน่งอยู่ในแอดเดรสช่วงใดช่วงหนึ่งของระบบเช่น 10h, 11h, 12h และ 13h โดยที่ขาสัญญาณแอดเดรสที่นอกเหนือไปจาก A0 และ A1 นำเข้ามายังตัวถอดรหัสแอดเดรสเพื่อสร้างสัญญาณเลือกอุปกรณ์(CS)ในช่วงแอดเดรสที่ต้องการ ให้คู่ตัวอย่างในรูปที่ 4 สัญญาณ CS นี้จะเป็นสถานะลอจิกต่ำก็ต่อเมื่อค่าในแอดเดรส A2-A7 มีค่าเท่ากับ 0000100xx (xx ใช้เพื่อระบุถึงรีจิสเตอร์ภายใน 8255 เพื่อทำการอ่านหรือเขียนข้อมูล) ดังนั้นจากวงจรนี้แอดเดรสของรีจิสเตอร์ภายใน 8255 จะมีค่าตามตารางต่อไปนี้

ตำแหน่งแอดเดรส	ความหมาย
10h	พอร์ต A
11h	พอร์ต B
12h	พอร์ต C
13h	รีจิสเตอร์ควบคุม

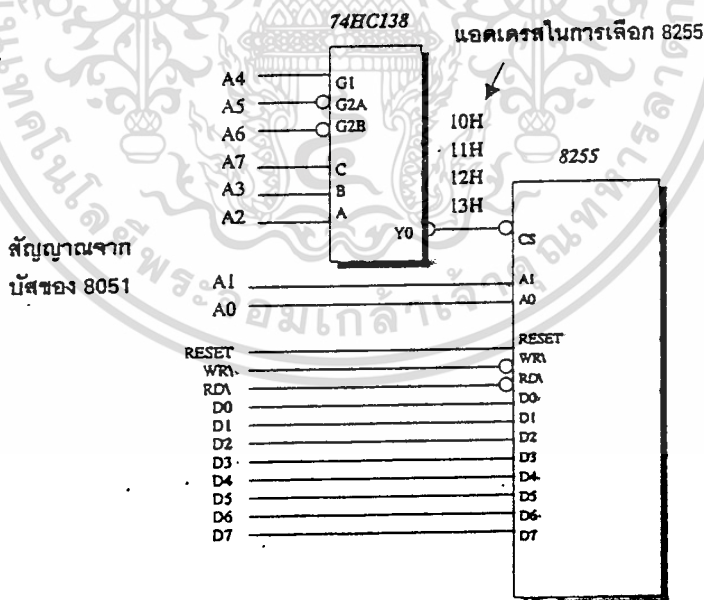
ขาสัญญาณควบคุมอื่นๆคือ RD\ และ WR\ มักจะเชื่อมต่อกับขาสัญญาณชื่อเดียวกันของ 8051 ได้โดยตรงทำให้แอดเดรสพอร์ตของ 8255 อยู่ในพื้นที่ของหน่วยความจำข้อมูลของ 8051 สำหรับขาสัญญาณรีเซตของ 8255 ซึ่งจะมีผลให้เกิดการรีเซต หรือเริ่มสถานะการทำงานใหม่เมื่อระดับของขาสัญญาณเป็นลอจิกสูง ดังนั้นหากว่าจะใช้สัญญาณการรีเซตเดียวกันกับ 8051 เพื่อที่จะรีเซต 8255 ด้วยก็สามารถทำได้โดยตรง

ค่าจากบัสแอดเดรสที่นำมาถอดรหัส A7 A6 A5 A4 A3 A2 A1 A0
 0 0 0 1 0 0 X X



รูปที่ 10

แผนภาพการสร้างสัญญาณเลือกอุปกรณ์(CS) ให้กับ 8255 โดยการถอดรหัส
 จากบัสแอดเดรส A2-A7



รูปที่ 11

แสดงภาพวงจรการเชื่อมระหว่าง 8255 กับ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนขาสัญญาณ D0-D7 ก็สามารถนำไปเชื่อมต่อได้โดยตรงเข้ากับบัสของ 8051 ได้เช่นกัน ขอให้ดูวงจรการทำงานแสดงการเชื่อมต่อระหว่าง 8255 และ 8051 ในรูปที่ 11 ซึ่งในที่นี้ สมมุติว่าไม่จำเป็นต้องมีการใช้วงจรหรือ ไอซีบัฟเฟอร์ขับสัญญาณบัสข้อมูล

การทำงานโหมด 0 ของ 8255

เมื่อ 8255 ได้รับการกำหนดให้ทำงานในโหมดนี้จะทำให้พอร์ตต่างๆมีหน้าที่เป็นพอร์ตอินพุตหรือเอาต์พุตได้เพียงลักษณะเดียวเท่านั้น การเริ่มต้นจะทำได้โดยการส่งไบท์ข้อมูลควบคุมให้กับรีจิสเตอร์ควบคุม ต่อไปจะแสดงให้เห็นถึงรูปแบบการกำหนดคิพเมื่อต้องการให้พอร์ต A, B และ C ทำหน้าที่เป็นพอร์ตเอาต์พุตทั้งหมดดังตารางต่อไปนี้

ตำแหน่งบิต	ค่าข้อมูล	ความหมาย
D7	1	ระบุให้ทราบว่าเป็นไบท์ข้อมูลควบคุม
D6 และ D5	00	กำหนดโหมดการทำงานให้กับพอร์ต A ให้เป็นโหมด 0
D4	0	ระบุว่าพอร์ต A เป็นเอาต์พุตของข้อมูล
D3	0	กำหนดให้เส้นสัญญาณสีบิทบนของพอร์ต C เป็นเอาต์พุตข้อมูล
D0	0	กำหนดโหมดการทำงานให้กับพอร์ต B เป็นโหมด 0
D1	0	ระบุว่าพอร์ต B เป็นเอาต์พุตข้อมูล
D2	0	กำหนดให้เส้นสัญญาณสีบิทล่างของพอร์ต C เป็นเอาต์พุต

ค่าของไบท์ข้อมูลควบคุมนี้จะต้องส่ง(หรือเขียน) ให้กับรีจิสเตอร์ควบคุมซึ่งถ้าหากใช้วงจรตามรูปที่ 5 จะเป็นแอดเดรส 13h และสามารถแสดงคำสั่งได้ดังต่อไปนี้

MOV DPTR, #4003H ; Set the Control Register Address

MOVX @DPTR, #80H ; Output the Control Word to 8255

ตามตัวอย่างข้างต้นเนื่องจากแอดเดรสมีค่าไม่เกิน 256(OFFH) จึงสามารถอ้างถึงตำแหน่งแอดเดรสของรีจิสเตอร์ควบคุมของ 8255 ผ่านทางรีจิสเตอร์ R0 ได้ แต่ในบางกรณีค่าของแอดเดรสนี้อยู่ที่ตำแหน่งในช่วงที่มีค่าเกิน ไปจากนี้ เช่นแอดเดรส 4000h ถึง 4003h ก็จะต้องอ้างอิงในการอ้างผ่านทางรีจิสเตอร์ DPTR แทนดังนี้

MOV DPTR, #4003H ; Set the Control Register Address

MOVX @DPTR, #80H ; Output the Control Word to 8255

ภายหลังจาก 8255 ได้รับการโปรแกรมจากค่าของไบท์ข้อมูลควบคุมนี้แล้วก็จะสามารถใช้งานพอร์ตทั้งหมดในฐานะพอร์ตเอาต์พุตเพื่อส่งออกข้อมูลได้ตามต้องการ ตัวอย่างเช่นหากต้องการส่งข้อมูลค่า 23h ออกทางพอร์ต A ค่าข้อมูล 41h ออกทางพอร์ต B และค่าข้อมูล 73h ออกทางพอร์ต C จะมีรูปแบบคำสั่งดังนี้

```

MOV R0, #10H ; Setup Port A Address
MOV @R0, 23H ; Output Data to 8255
MOV R0, #11H ; Setup Port B Address
MOV @R0, 41H ; Output Data to 8255
MOV R0, #12H ; Setup Port C Address
MOV @R0, 73H ; Output Data to 8255

```

กรณีต่อไป หากต้องการให้พอร์ตของ 8255 มีทั้งพอร์ตอินพุตและเอาต์พุต ตัวอย่างเช่น ต้องการให้พอร์ต A และ C เป็นพอร์ตเอาต์พุตสำหรับพอร์ต B เป็นพอร์ตอินพุต ค่าของไมท์ควมคุมจะเป็นดังนี้

ตำแหน่งบิต	D7	D6	D5	D4	D3	D2	D1	D0
ค่าของบิต	1	0	0	0	0	0	1	0

ตัวอย่างคำสั่งเพื่อการอ่านข้อมูลเข้ามาทางพอร์ต B

```

MOV R0, #11H ; Setup Port B Address
MOV A, @R0 ; Read Data From Port B

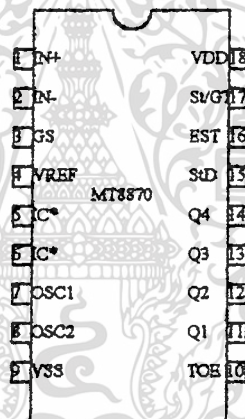
```

การถอดรหัสความถี่โทรศัพท์

MT8870 ไอซีถอดรหัสความถี่โทรศัพท์

MT8870 ใช้ถอดรหัสสัญญาณโทรศัพท์ชนิดกดปุ่ม (DTMF) ให้เป็นตัวเลข BCD ขนาด 4 บิตโดยเพียงใช้งานร่วมกับคริสตอล 3.579 Mhz เท่านั้น

ถอดรหัสความถี่โทรศัพท์ หมายถึง การแปลงสัญญาณความถี่ซึ่งเกิดจากการกดปุ่ม(ชนิด Tone หรือ DTMF) ให้เป็นระบบตัวเลขทางดิจิทัลซึ่งไอซี MT8870 ไข์แปลงความถี่โทรศัพท์ให้เป็นเลขฐานสองขนาด 4 บิต



MT8870BE 18 PIN PLASTIC
MT8870BC 18 PIN CERDIP

รูปที่ 12

แสดงรายละเอียดของ MT8870

คุณสมบัติของ MT8870

1. เป็นตัวรับและถอดรหัสความถี่(DTMF reciever)
2. กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
3. สามารถตั้งอัตราขยายภายในตัวไอซีได้
4. สามารถตั้งการ์ด ไทม์(Guard Time) ได้
5. เป็นไอซีคุณภาพสูง

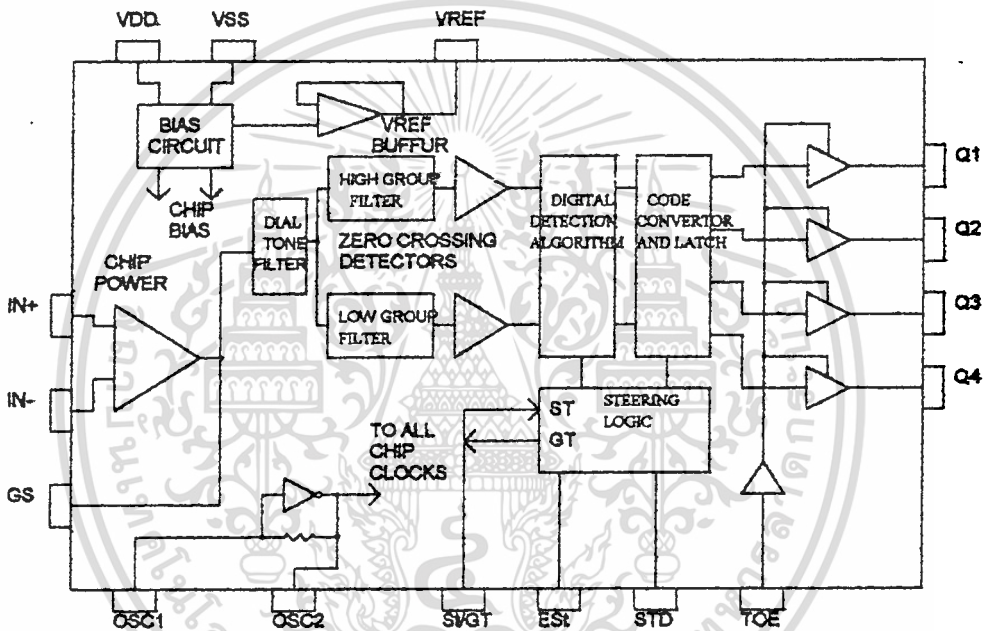
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่และวงจรถอดรหัสฟังก์ชันทางดิจิทัล ในส่วนของวงจรกรองความถี่ใช้เทคนิคของของสวิทช์ค่าปาดิเตอร์ฟิลเตอร์สำหรับกรองความถี่สูงและต่ำ

ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิทัลเพื่อตรวจนับและถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐานสองขนาด 4 บิต

ส่วนภาคอินพุทเป็นออปแอมป์ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอก เอาท์พุทเป็นวงจรแลตช์ 3 สถานะ



รูปที่ 18

แสดงโครงสร้างภายในของ MT8870

การทำงานภายใน MT8870 ประกอบด้วย 5 ส่วน

1. ภาคกรองความถี่ (Filter section)
2. ภาคถอดรหัส (Decoder section)
3. ภาคตรวจสอบขอมูล (Steering circuit)
4. ภาคขยายความแตกต่าง (Differential input)
5. ภาคกำเนิดความถี่ (Oscillator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคกรองสัญญาณความถี่

ส่วนนี้จะแยกสัญญาณเป็น DTMF ที่เข้ามาเป็น 2 กลุ่มความถี่คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ โดยวงจรกรองแถบความถี่อันดับ 6 ชนิดสวิทช์คาปาซิเตอร์

ภาคถอดรหัส

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบคิวิตอด และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่เพื่อป้องกันความถี่อื่นๆเข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้องสัญญาณที่เข้า EST ก็จะแยกทีฟสำหรับค่าที่ถอดรหัสได้จากความถี่ต่างๆนั้น แสดงในรูปข้างล่าง

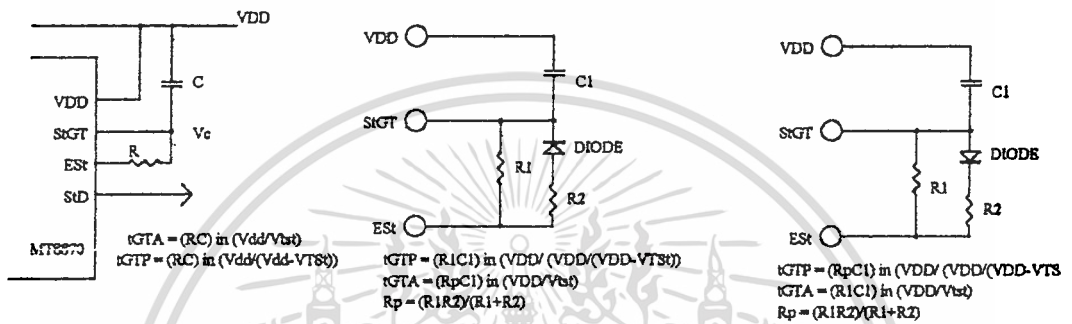
F _{LOW}	F _{HIGH}	NO	TOE	Q ₁	Q ₂	Q ₃	Q ₄
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	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

รูปที่ 14

แสดงค่าที่ถอดรหัสได้จากความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแสดงวงจรตรวจสอบสัญญาณอย่างง่ายและแสดงการกำหนดเวลาคาร์ด์ใหม่พร้อมวิธีคำนวณ



รูปที่ 15
แสดงการตรวจสอบจรภาคอินพุต

ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุตจะมีการตรวจสอบช่วงความถี่ที่เข้ามา มีระยะเวลาตามที่กำหนดหรือไม่โดยสังเกตจากระยะเวลาการกดปุ่มใหม่มีความถี่ออกมาเป็นช่วงเวลาพอสมควรมีจะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณส่วนนี้ไม่ถูกต้องส่วนความยาวของช่วงเวลาสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา EST จะเป็น "High" นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา จากรูปที่ 5 เมื่อขา EST เป็น "High" จะทำให้ VC สูงขึ้นตัวเก็บประจุ C จะคายประจุทำให้แรงดัน VC สูงขึ้นจนถึงค่าที่ทรานซิสเตอร์จะถอดรหัสออกเป็นตัวเลขฐานสองขนาด 4 บิต

สำหรับคำว่าคาร์ด์ใหม่นั้นหมายถึงช่วงคาบเวลาของความถี่ของสัญญาณที่เข้ามาซึ่งจะต้องนานมากกว่าหรือเท่ากับช่วงเวลาที่ตั้งไว้จึงจะได้รับการยอมรับว่าสัญญาณความถี่นั้นถูกต้องหรือพูดได้ว่าเวลาที่เรากำหนดไว้โดยวงจร RC คือคาร์ด์ใหม่นั้นเอง เมื่อสัญญาณความถี่เข้ามานานเท่าที่คิดการหรือมากกว่าเวลาที่ตั้งไว้จึงจะสามารถแปลงเป็นตัวเลขได้สัญญาณความถี่เข้ามาสั้นกว่าก็จะมีผลการถอดรหัสออกมาเป็นตัวเลข

ภาคขยายสัญญาณความแตกต่าง

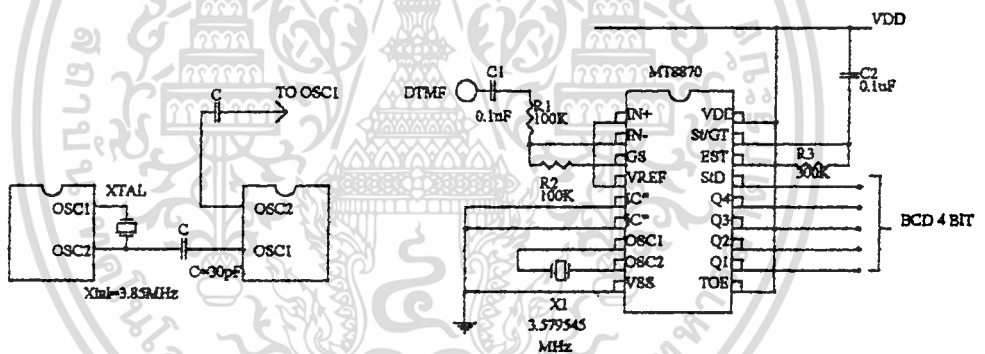
วงจรถ่วงอินพุตของ MT8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยาย โดยต่อวงจรภายนอกเพิ่มเข้าไป รูปที่ 6 แสดงถึงการต่อวงจรภายนอกเพิ่มเข้าไปกับอินพุตซึ่งสามารถคำนวณอัตราขยายความแตกต่างของอินพุตและอิมพีแดนซ์ได้จากสูตรนี้

$$\text{อัตราขยาย}(A_v\text{-diff}) = \frac{R5}{R2}$$

$$\text{อินพุตอิมพีแดนซ์}(Z_{in}\text{-diff}) = 2 * \sqrt{(R1 * R1 + (\frac{1}{WC}) * (\frac{1}{WC}))}$$

ภาคกำเนิดความถี่

ในภาคนี้ภายในไอซีจะมีวงจรถูกภายในเพียงแค่อะคริสตอลขนาดความถี่ 3.58Mhz ก็สามารถใช้งานได้ทันที โดยในการต่อวงจรถ่วงกำเนิดความถี่แสดงในรูปข้างล่าง



รูปที่ 16

แสดงการต่อวงจรผลิตความถี่

วงจรกำเนิดสัญญาณ

ไอซีประเภท CMOS เป็นไอซีที่ใช้พลังงานไฟฟ้าขณะทำงานน้อยกว่าไอซีประเภท TTL เพราะว่า TTL ใช้พลังงานในขณะที่ทำงานมีหน่วยเป็นมิลลิวัตต์ และไอซีชนิด CMOS ยังทำงานได้ในช่วงไปเลี้ยงที่กว้างกว่าอีกด้วยซึ่งอาจจะสูงมากกว่า 5 โวลท์ก็ได้

วงจรกำเนิดสัญญาณรูปที่ 17 นี้ใช้ไอซี 4047 ทำงานเป็นวงจรอะอสซิลเลเตอร์แบบเรเตอร์ สังเกตจากวงจรในรูปจะเห็นได้ว่ามีขาเอาต์พุตใช้งานอยู่ 3 ขา โดยที่ขา OSC (ขาเอาต์พุตของสจิมิลเลเตอร์) จะเป็นขาสัญญาณเอาต์พุตของวงจรออสซิลเลเตอร์ที่ต่อตรงออกมาจากวงจรรภายในไอซี ส่วนขาเอาต์พุตอีกสองขาคือ Q และ Q̄ จะให้เอาต์พุตที่มีความถี่เป็นครึ่งหนึ่งของความถี่ที่ขา OSC และที่สำคัญเอาต์พุตทั้งสามขานี้สามารถนำไปใช้งานได้ทันที

ความถี่ของวงจรกำเนิดสัญญาณที่เอาต์พุต OSC สามารถกำหนดได้จากค่า R1 และ C1

จากสูตร

$$f = \frac{1}{2.2 * R1 * C1}$$

ค่าของ R1 มีค่าอยู่ระหว่าง 10 กิโลโอห์มถึง 1 เมกะโอห์ม ในขณะที่ค่าของ C1 ควรจะใช้ค่าเริ่มต้นตั้งแต่ 1 พิโคฟารัดเป็นขึ้นไป

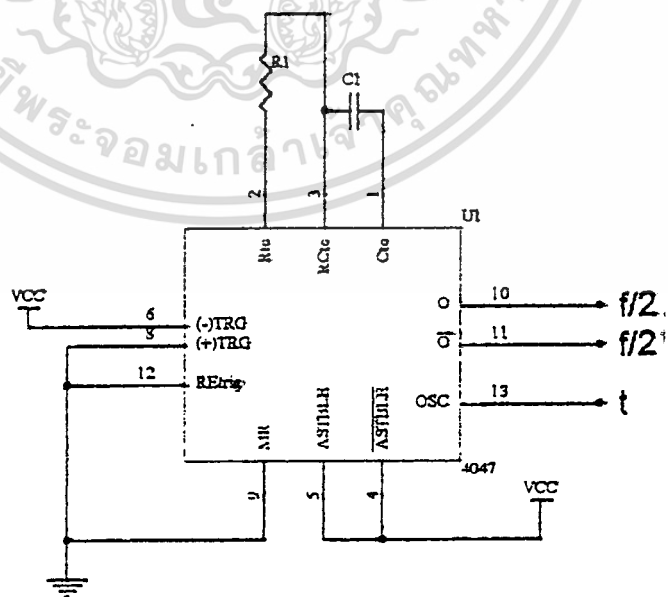
จากวงจร R1 มีค่า 1 เมกะโอห์ม และ C1 มีค่า 4.7 ไมโครฟารัด

$$f = \frac{1}{2.2 * 1 \times 10^6 * 4.7 \times 10^{-6}}$$

$$f = \frac{1}{2.2 * 1 * 4.7} = 0.096666K \approx 0.0967 \text{ Hz}$$

$$f = \frac{1}{T}$$

$$T = 10.34 \text{ Sec}$$



รูปที่ 17 วงจรกำเนิดสัญญาณที่ใช้ไอซีประเภท CMOS เบอร์ 4047

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

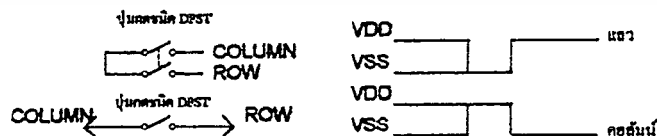
การส่งรหัสหมายเลข

โดยการใช้ไอซีสำเร็จรูปการส่งสัญญาณแบบ DTMF ซึ่งในระบบนี้ยังคงต้องใช้อุปกรณ์จำพวกพาสซีฟ(Passive element) ในการนำมาสร้างวงจรออสซิลเลเตอร์ซึ่งเป็นที่แน่นอนว่าวงจรจะต้องมีปัญหาเกี่ยวกับความคลาดเคลื่อนในการทำงานที่จะเกิดขึ้นเนื่องจากสภาพแวดล้อมที่เปลี่ยนแปลงไปและอายุการใช้งาน จึงจะมีผลที่จะกระทบคือ ความถี่ที่จะผลิตออกมาจะมีค่าเปลี่ยนแปลงไปด้วย ผลสุดท้ายก็อาจจะทำให้ชุมชนเกิดการดำเนินงานผิดพลาดขึ้นในการติดต่อกับผู้ถูกเรียก

ไอซีที่จะนำมาใช้สร้างสัญญาณในระบบ DTMF ซึ่งวงจรภายในจะประกอบไปด้วย วงจรนับและถอดรหัส(Counter and Decoder) ซึ่งวงจรถอดรหัสก็จะแยกแยะว่าการกดหมายเลขแต่ละครั้งจะตรงกับตำแหน่งใดบ้างในแนวแถวและแนวคอลัมน์ เมื่อทำการถอดรหัสสัญญาณจากการกดได้แล้วก็นำค่าในแนวแถวและแนวคอลัมน์ไปหารจากค่าความถี่หลักสัญญาณที่ออกมาจากวงจรนับและถอดรหัสก็จะได้สัญญาณดิจิทัล 2 สัญญาณที่มีความถี่แตกต่างกัน จากนั้นก็จะนำทั้ง 2 สัญญาณ ไปผ่านวงจรแปลงสัญญาณจากดิจิทัลเป็นอะนาล็อก (D/A converter) และนำมารวมกันโดยผ่านวงจรรวมและขยายสัญญาณ (Summing amp) แล้วจึงถูกส่งไปยังวงจรควบคุมเสียงพูด (Speech network) และผ่านต่อไปยังชุมสายโทรศัพท์

การรวมระบบที่ลดและ DTMF ภายในไอซีตัวเดียวกัน

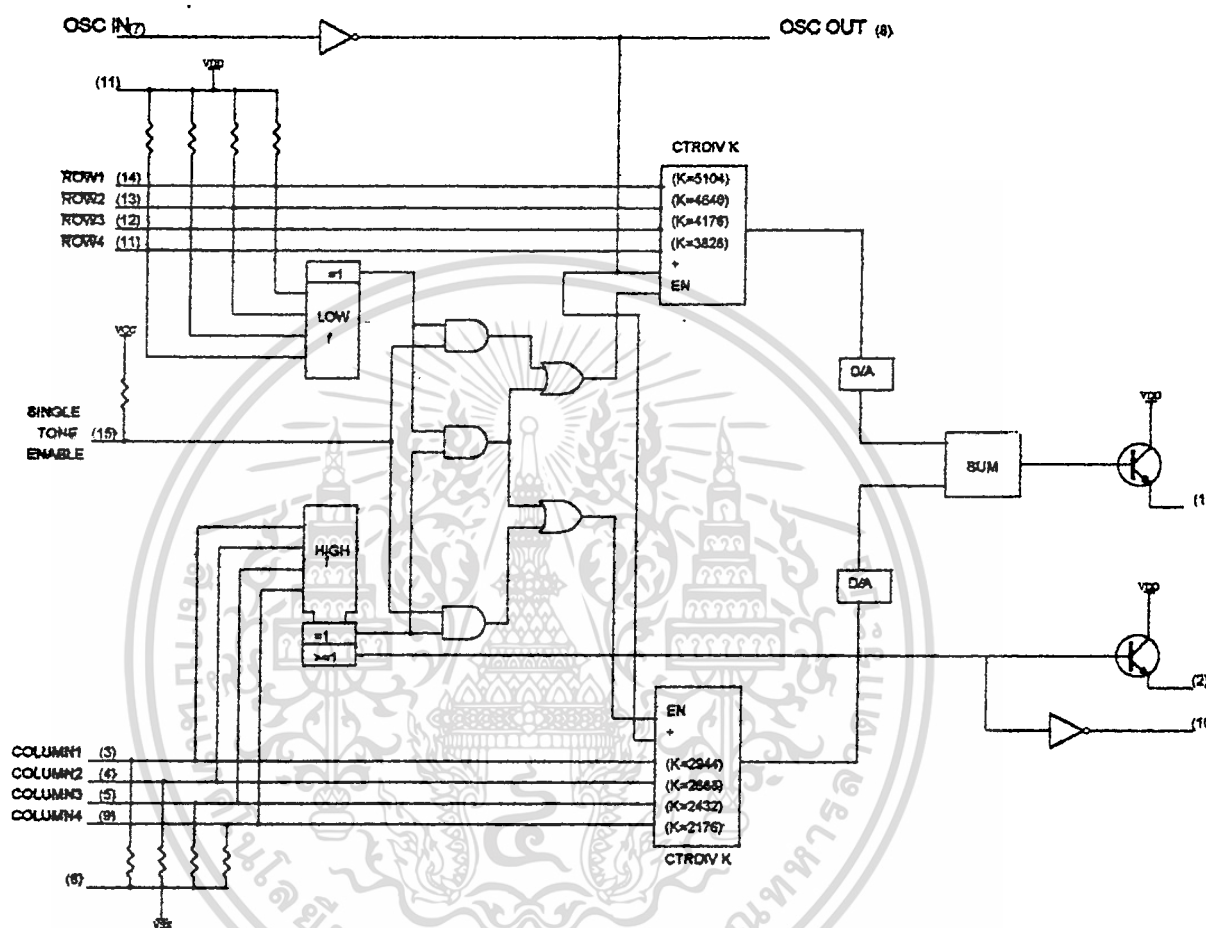
ไอซีเบอร์ที่ทำงานได้ทั้ง 2 โหมดที่ใช้งานกันอยู่ตั้งแต่เบอร์ MC145412, MC145413 หรือ MC1445512



รูปที่ 18

แสดงชนิดของปุ่มกดและรูปสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 19

บล็อกไดอะแกรมของวงจรภายในไอซีเบอร์ TCM5087

เป็นการนำ MC145412 มาทำหน้าที่ในการส่งรหัสหมายเลขโดยการอินเตอร์เฟสกับปุ่มกดชนิด 3 x 4 ตามรูปไอซีเบอร์นี้ยังคงใช้ความถี่ 3.579 Mhz เป็นความถี่หลักในการทำงานอยู่ การเลือกให้ไอซีทำงานโหมดการส่งสัญญาณแบบพัลส์สามารถเลือกอัตราการส่งได้ 10 pps หรือ 20 pps (pps: pulse per second) ถ้าหาก ไอซีกำลังทำงานในพัลส์โหมด DTMF ขา OPL (pulse output) ก็จะเป็น high impedance แทน ขาสัญญาณ MO (mute signal) ก็จะนำไปใช้ควบคุมโมโม่มีสัญญาณเสียงที่สามารถผ่านเข้าสู่ไมโครโฟนได้ สำหรับสัดส่วนการเปิด/ปิดวงจร (Make/Break)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6242 REAL TIME CLOCK

6242 REAL TIME CLOCK ชิปนาฬิกาในระบบไมโครซึ่งเป็นที่นิยมกันอย่างแพร่หลาย โดยมีการใช้งานที่ง่าย การอ่านและเขียนข้อมูลจะกระทำในลักษณะ Memory รวมทั้งมีความเที่ยงตรงในการทำงาน และสามารถตั้งให้ทำการ Interrupt กับตัวไมโครได้ นอกจากนี้ยังมีข้อมูลเกี่ยวกับ วัน เดือน ปี วันในสัปดาห์ และจำนวนปีอธิกสุรทินโดยอัตโนมัติ

การอ้างแอดเดรส(ADDRESS)ของ 6242 จะมองเห็นเป็นช่วง 16 byte โดยแต่ละไบต์จะมีลักษณะแตกต่างกันไปช่วงตั้งแต่ 0-C จะเป็นส่วนของการอ่านและเขียนค่า เวลา วัน เดือน ปี และช่วงตั้งแต่ D-F จะเป็นส่วนสำหรับการกำหนดคุณสมบัติต่างๆในการทำงานของ 6242 โดยจะอ่านและเขียนข้อมูลนี้จะกระทำเพียง 4 บิต (D0-D3) เท่านั้น เช่นถ้าอ่านค่าวินาทีจาก 6242 ก็ทำได้ โดยการอ่าน ADDRESS ที่ 1 ก่อนซึ่งจะได้วินาทีหลักแรก จากนั้นจึงค่อยอ่าน ADDRESS ที่ 0 ตามมาซึ่งจะได้หลักวินาทีที่สอง และจึงนำมารวมกันเป็นวินาทีครบสองหลัก รายละเอียดมีดังนี้

ADDRESS	REGISTER	D8	D2	D1	D0	COUNT	DESCRIPTION
XXX0	S1	S8	S4	S2	S1	0-9	1 SECOND DIGIT
XXX1	S10	--	S40	S20	S10	0-5	10 SECOND DIGIT
XXX2	M1	M8	M4	M2	M1	0-9	1 MINUTE DIGIT
XXX3	M10	--	M40	M20	M10	0-5	10 MINUTE DIGIT
XXX4	H1	H8	H4	H2	H1	0-9	1 HOUR DIGIT
XXX5	H10	--	PM/A M	H20	H10	0-1-2	PM/AM, 10 HOUR DIGIT
XXX6	D1	D8	D4	D2	D1	0-9	1 DAY DIGIT
XXX7	D10	--	--	D20	D10	0-3	10 DAY HOUR DIGIT
XXX8	M01	M08	M04	M02	M01	0-9	1 MOUNTH DIGIT
XXX9	M010	--	--	--	M010	0-1	10 MOUNTH DIGIT
XXXA	Y1	Y8	Y4	Y2	Y1	0-9	1 YEAR COUNT
XXXB	Y10	Y80	Y40	Y20	Y10	0-9	10 YEAR DIGIT
XXXC	W	--	W4	W2	W1	0-6	WEEK DIGIT
XXXD	CD	30 ADJ	IRQ	BUSY	HOLD	--	CONTROL D
XXXE	CE	T1	T0	IT/ST	MASK	--	CONTROL E
XXXF	CF	TEST	24/12	STOP	RESET	--	CONTROL F

CONTROL D REGISTER

1. HOLD เมื่อให้ HOLD = 1 จะเป็นการกำหนดให้คงสถานะของเวลาเอาไว้เพื่อการอ่านหรือการเขียนโดยจะตรวจสอบกับบิต Busy อีกที และเมื่อทำการอ่านหรือเขียนเรียบร้อยแล้วก็ให้ HOLD = 0 ตามปกติ การให้คงสถานะของเวลาเอาไว้จะไม่ทำให้ระบบเกิดเวลาทำงานคิดเขียนแต่อย่างใดเพราะภายในยังทำงานเป็นปกติ และการอ่านหรือการเขียนนี้ควรกระทำให้เสร็จเรียบร้อยภายใน 1 วินาทีหลังจากที่กำหนดให้ HOLD = 1
2. BUSY สำหรับแสดงความพร้อมของข้อมูลโดยถ้า Busy = 1 ก็แสดงว่าข้อมูลยังไม่สามารถอ่านหรือเขียนได้ แต่ถ้า Busy = 0 จะแสดงว่าข้อมูลพร้อมที่จะทำการอ่านและเขียนได้

IRQ สำหรับบอกสถานะที่เกิดการ Interrupt โดยถ้าขา STDP มีค่าเป็น 0, IRQ = 1 แต่ถ้าขา STDP มีค่าเป็น 1, IRQ = 0 เป็นการตั้งค่า Interrupt จากบิต Mask และตั้งค่านวลาจากบิต τ_0 และ τ_1 แต่ถ้าบิต IT/ST = 1 (Interrupt Mode) เมื่อ STDP มีค่าเป็น 0 บิต IRQ จะกลับเป็น 1 ก็ต่อเมื่อมีการเขียนค่า IRQ ให้เท่ากับ 0 เท่านั้น แต่ถ้าในกรณีที่ตั้งค่าของบิต IT/ST = 0 ค่า IRQ จะกลับเป็น 0 หลังจากผ่านไป 7.8125 ms 30ADJ สำหรับการกำหนดให้ 6242 ทำการปรับเวลาโดยดูจากค่าวินาทีที่กำลังดำเนินอยู่ ถ้ามีค่าน้อยกว่า 30 จะปรับค่าวินาทีเป็น 0 พร้อมกับปรับค่านาทีเพิ่มขึ้นอีก 1 ค่าการกำหนด 30ADJ จะกระทำได้โดยการกำหนดให้บิตนี้มีค่าเป็น 1 และบิตนี้ถูกเซตค่ากลับเป็น 0 โดยอัตโนมัติเมื่อเวลาผ่านไป 0.125 ms

CONTROL E REGISTER

1. MASK เมื่อให้ Mask = 1 จะทำให้ขาสัญญาณ STDP = 1 ด้วย (สถานะ Open) แต่ถ้าให้ Mask = 0 ก็จะทำให้ STDP มีค่าเท่ากับเป็น Output Mode ซึ่งก็คือการ Enable ขาสัญญาณ STDP นั่นเอง
2. IT/ST สำหรับการกำหนดลักษณะของขาสัญญาณ STDP โดยถ้าให้เป็น 0 จะหมายถึง Standard Timing Waveforms โดยจะให้ Output คงที่เป็นพัลส์ลบซึ่งมีความกว้าง 7.8125 ms และถ้าให้ STDP = 1 จะหมายถึง Interrupt โดยจะทำให้ Output เป็นพัลส์ลบเช่นกัน ซึ่งจะ เป็นพัลส์บวกได้ก็ต่อเมื่อมีการกำหนดให้บิต IRQ = 0
3. τ_0 , τ_1 สำหรับกำหนดช่วงขาของสัญญาณ STDP และรวมไปถึงความกว้างของพัลส์ในกรณีที่ ตั้งเป็น Standard Timing Waveforms ด้วย

τ_0	τ_1	Period	ความกว้างพัลส์ลบ(IT/ST=0)
0	0	1/64 second	4
0	1	1 second	1/128
1	0	1 minute	1/7680
1	1	1 hour	1/460800

CONTROL F REGISTER

1. **REST** เมื่อกำหนด Rest = 1 ตัว RTC จะทำการ Reset การนับ Clock ภายในซึ่งจะไม่ทำให้ค่าเวลาเปลี่ยนแปลง แต่จะหยุดการนับจนกว่าจะกำหนดให้ Rest = 0 อนึ่งถ้าขาสัญญาณ CSI = 0 ก็จะทำให้ Rest = 0 โดยอัตโนมัติ
2. **STOP** เมื่อกำหนดให้ Stop = 1 จะเป็นการหยุดเวลาเอาไว้ และถ้าให้ Stop = 0 ก็จะทำให้เดินตามเวลาต่อไป ทั้งนี้จะใช้สำหรับหยุดค่าเวลาเดิมเพื่อกำหนดค่าเวลาใหม่ลงไป
3. **24/12** สำหรับกำหนดให้ค่าเวลาทำงานในแบบ 24 ชม.(24/12 = 1) หรือแบบ 12 ชม.(24/12 = 0) การกำหนดให้ทำงานแบบ 12 ชม.นี้เราสามารถดูค่า AM(0) หรือ PM(1) ได้จากบิต AM/PM ด้วย และการกำหนดบิต 24/12 นี้จะทำได้ในขณะที่บิต Rest = 1 เท่านั้น
4. **TEST** สำหรับการ Test การทำงานของ RTC โดยถ้าให้ Test = 1 ค่าเวลาจะเดินด้วยความเร็วที่มากขึ้นกว่าเดิม 15 เท่า ทั้งนี้เพื่อการทดสอบต่างๆ ขาสัญญาณ STDP จะกำหนดการต่อเข้ากับขา INTO ของตัวไมโครฯ

สำหรับการใช้งานในลักษณะที่เป็น Interrupt ทั้งนี้จะสามารถ Enable or Disable ได้ด้วยตัว Jumper RTC รายละเอียดของการอ่านและการเขียนค่าเวลาพอจะสรุปเป็นแนวทางได้ดังนี้

การเริ่ม SETUP ครั้งแรก

1. กำหนดให้ Stop และ Rest มีค่าเป็น 1
2. กำหนดการเดินเวลาแบบ 24 ชม. หรือ 12 ชม. ที่บิต 24/12
3. กำหนดเรื่องการ Interrupt ตามการใช้งานที่ Control E
4. กำหนดให้ Hold = 0
5. กำหนดให้ Stop & Rest = 0 และพร้อมที่จะใช้งานได้ต่อไป

การอ่านหรือการเขียนค่าเวลา

1. กำหนดให้ Hold = 1 เพื่อค้างค่าเวลา
2. ตรวจสอบบิต Busy โดยถ้าเท่ากับ 1 หมายถึงยังไม่พร้อมที่จะอ่านหรือเขียนข้อมูลให้กำหนด Hold = 0 และกลับไปทำงานที่ข้อ 1 ใหม่ แต่ Busy = 0 ให้ทำข้อต่อไปได้
3. ทำการอ่านหรือเขียนค่าเวลาได้ตามต้องการโดยขั้นตอนนี้ควรจะทำให้เสร็จเรียบร้อยภายใน 1 วินาที
4. กำหนดให้ Hold = 0 เป็นการสิ้นสุดการตั้งเวลา

SOURCE CODE โปรแกรม TIMER.ASM

DOT MATRIX LCD MODULE

ปัจจุบัน LCD เป็นที่นิยมกันอย่างกว้างขวาง สำหรับการแสดงผลในเครื่องมือเครื่องใช้ต่าง ๆ ไม่ว่าจะเป็นเครื่องเล่น VIDEO, เครื่องถ่ายภาพ, เครื่องมือวัดคุมต่างๆ, เครื่องคอมพิวเตอร์

LCD สามารถแบ่งประเภทตามลักษณะการแสดงผลได้ 3 แบบ

1. LCD แบบอักขระ (Character LCD Module)
2. LCD แบบกราฟิก (Graphics LCD Module)
3. LCD แบบเซกเมนต์ (Segment LCD Module)

LCD แบบอักขระ เป็นโมดูล LCD ที่สามารถแสดงตัวอักษร, ตัวเลข และเครื่องหมายต่างๆ ได้โดยสร้างจุดเล็กๆ หรือที่เรียกว่า คอสมेटริกซ์ ซึ่งก็จะมีขนาดความและสูงของอักขระแต่ละตัว โดยทั่วไปมี 2 ขนาดคือ 5x7 จุด นอกจากนั้น LCD แบบนี้สามารถแสดงผลมากกว่า 1 บรรทัดได้ โดยขึ้นอยู่กับรุ่นของ LCD ตัวนั้นๆ

LCD แบบกราฟิก สามารถแสดงข้อมูลเป็นตัวอักษร, ตัวเลข, เครื่องหมาย และรูปภาพได้ ความละเอียดของภาพของ LCD แบบนี้มีมากและมีหลายขนาดให้เลือกใช้ ในปัจจุบันนี้มีการพัฒนาให้มีการแสดงผลเป็นสีได้แล้วด้วย

LCD แบบเซกเมนต์ เป็นโมดูล LCD แบบที่เล็กที่สุด มีลักษณะการแสดงผลคล้ายกับ LED 7 Segment โดยปกติมักจะมีมากกว่า 1 หลัก พบเห็นทั่วไปในดิจิตอลมัลติมิเตอร์

ส่วนประกอบต่างๆของ LCD

1. Dot Matrix LCD เป็นตัวแสดงผลให้เรามองเห็นในลักษณะ การปิดและการเปิด ตัวอักษรกับแสงบนจอผลึก คือส่วนที่เป็นตัวกระจกบรจุผลึก
2. Driver เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก LCD อีกทีหนึ่ง โดยเบอร์ของ IC ที่นิยมใช้ เป็น LCD Module เช่น HD4400H, MSMS259
3. Controller เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาและจัดการควบคุม LCD Module ให้ทำงานแสดงผลต่างๆเช่น การลบจอภาพ, การแสดงตัวอักษรเป็นต้น โดยมีเบอร์ของ IC ที่นิยมใช้กัน คือ HD4478 ซึ่งจะใช้ในแบบ Character LCD Module เป็นส่วนใหญ่ และ HD61830 จะใช้ในแบบ Graphics LCD Module

คุณสมบัติของ DOT MATRIX LCD MODULE สามารถสรุปได้เป็นข้อๆ ดังนี้

1. มีให้เลือกหลายรุ่นตามความต้องการ โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงผลด้วย DOT MATRIX ขนาด 5x8 DOT
3. สามารถต่อเข้ากับระบบไมโครคอมพิวเตอร์ได้ 3 ลักษณะคือ
 - 3.1 แบบ Memory Map (20-pin LCD Bus)
 - 3.2 แบบผ่าน 8255 Port (20-pin 8255 Bus)
 - 3.2 การต่อผ่าน Port1 ของ MCS-51
4. การใช้งานง่ายและสะดวกในระบบไมโครคอมพิวเตอร์ เพียงแค่ส่งข้อมูลให้กับ LCD Module เท่านั้นข้อความก็จะปรากฏขึ้นที่จอแสดงผลและจะคงค้างไว้ตลอด ทำให้ไม่เสียเวลา CPU Times ของ CPU
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมายเช่น Clear Display, Home Cursor, On-Off Cursor, Blink Character และอื่นๆอีกมาก
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดคีย์ที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสและน้ำหนักรเบา ทำงานได้ในระดับไฟเลี้ยงแค่ 5 โวลท์เท่านั้น

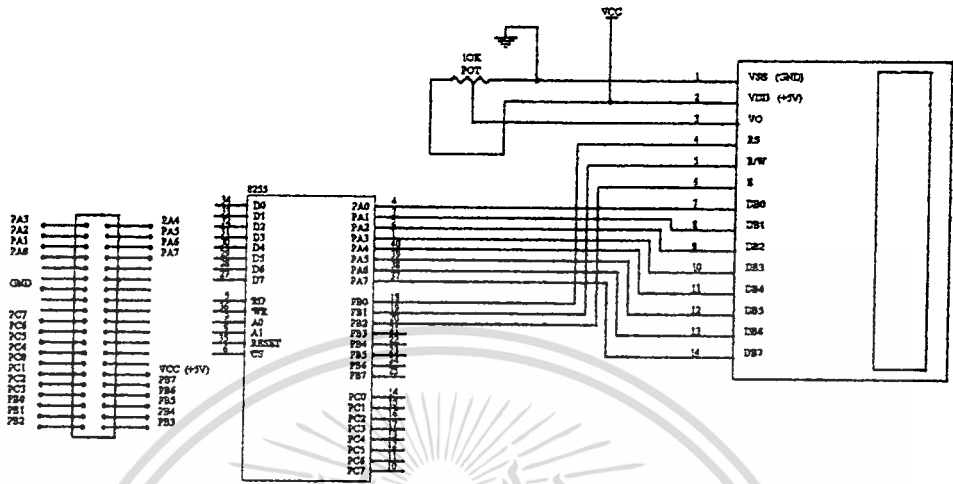
ขาสัญญาณของโมดูล LCD แบบอักษร

- Vss (ขา 1) : คอกราวด์
- Vdd (ขา 2) : ต่อไฟเลี้ยง +5V
- Vo (ขา 3) : เป็นขาอินพุตสำหรับป้อนแรงดันเพื่อปรับความเข้มของการแสดงผล
- RS (ขา 4) : เป็นขาอินพุต ใช้เลือกว่าข้อมูลที่ทำการส่งในขณะนั้นเป็นข้อมูลคำสั่งหรือข้อมูลแสดงผลสำหรับรีจิสเตอร์ของ LCD โดยถ้าขานี้มีสัญญาณเป็น "0" ข้อมูลที่ส่งมาจะเป็นข้อมูลคำสั่ง แต่ถ้าเป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล
- R/W (ขา 5) : เป็นขาที่เลือกไว้ว่าจะทำการอ่านหรือเขียนข้อมูลกับ LCD ถ้าหากเป็น "0" จะเป็นการเขียนข้อมูลที่จะส่งให้กับ LCD แต่ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจาก LCD
- E (ขา 6) : เป็นขาที่อีนามิเบิล(enable) ให้ LCD ทำงาน

DB0 - DB7 (ขา 7-14): เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายใน

นอกขนาด 8 บิต

อนึ่งขา RS, R/W, และ E จะใช้งานร่วมกัน โดยมีลักษณะความสลับพัลส์ดังรูป



รูปที่ 20

ลักษณะการต่อ LCD กับ 8255

รายละเอียดของคำสั่ง HD44780

1. CLEAR DISPLAY

CLEAR DISPLAY

RS	R/W	DB7	-----						DB0
0	0	0	0	0	0	0	0	0	1

Code

คำสั่งนี้จะเป็นการเขียนช่องว่างหรือ SPACE(ASCII 20H) เข้าไปใน DD RAM ทั้งหมด และทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพ SET I/D = 1, S ไม่มีการเปลี่ยนแปลง

2. RETURN HOME

RETURN HOME

RS	R/W	DB7	-----						DB0
0	0	0	0	0	0	0	0	1	0

Code No effect

คำสั่งนี้จะเป็นการ SET DDRAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพและข้อมูลในจอภาพไม่เปลี่ยน

3. ENTRY MODE SET

ENTRY MODE SET

RS	R/W	DB7	-----						DB0	
0	0	0	0	0	0	0	0	1	I/D	0

Code

BIT I/D : เป็นตัวกำหนดให้ว่าเมื่อทำการเขียนหรืออ่านข้อมูลแล้วจะทำให้ DD RAM ADDRESSER ค่าเพิ่มขึ้นหรือลดลงทีละหนึ่ง โดย

1 = เพิ่ม (Increment)

2 = ลด (Decrement)

BIT S : เป็นตัวกำหนดการแสดงผลโดยถ้า

S = 1 เป็นการใส่ข้อมูลแล้วตัว CURSOR อยู่ที่ที่และข้อมูลจะถูกดันให้ไปอยู่ด้านซ้ายมือ

S = 2 เป็นการใส่ข้อมูลแล้วข้อมูลจะอยู่ที่ที่และตัว CURSOR จะถูกดันให้ไปอยู่ด้านขวามือ

4. DISPLAY ON/OFF CONTROL

DISPLAY ON/OFF CONTROL

RS	R/W	DB7	-----						DB0	
0	0	0	0	0	0	0	1	D	C	B

Code

BIT D : เป็น BIT ให้เปิดปิดหน้าจอภาพโดยถ้า

D = 1 ON (เปิดหน้าจอ)

D = 2 OFF (ปิดหน้าจอ)

BIT C : ลักษณะการแสดงผลของ CURSOR

BIT C = 1 แสดง CURSOR

BIT C = 0 ซ่อน CURSOR โดยตัว CURSOR จะอยู่ที่ LINE ที่ 8 ในแบบ 5x7

DOT และจะอยู่ LINE ที่ 11 ในแบบ 5x10 DOT

BIT B : เป็น BIT SET การกระพริบของ CURSOR โดยถ้า

B=1 ให้CURSOR กระพริบ โดยมีระยะเวลาการกระพริบประมาณประมาณ
379.2 ms
B = 0 ให้CURSORไม่มีการกระพริบ

5. CURSOR OR DISPLAY SHIFT

CURSOR OR DISPLAY SHIFT

RS	R/W	DB7	-----							DB0
0	0	0	0	0	1	S/C	R/L	*	*	

Code No effect

เป็นคำสั่งกำหนดให้ตำแหน่ง CURSOR หรือข้อมูลไปเกิดทางซ้ายหรือขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่าน โดย

S/C	R/L	
0	0	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปซ้ายมือ 1 ตำแหน่ง
0	1	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปขวามือ 1 ตำแหน่ง
1	0	เป็นการค้นตัวอักษรที่เกิดไปทางซ้าย
1	1	เป็นการค้นตัวอักษรที่เกิดไปทางขวา

6. FUNCTION SET

FUNCTION SET

RS	R/W	DB7	-----							DB0
0	0	0	0	1	DL	N	F	*	*	

Code No effect

BIT DL : เป็นการ SET การคิดต่อว่าจะให้เป็นแบบ 8 บิตหรือ 4 บิต

DL = 0 เป็นการคิดต่อแบบ 4 บิต

DL = 1 เป็นการคิดต่อแบบ 8 บิต

N : เป็นการ SET บรรทัดการแสดงผล

N = 0 แสดง 1 บรรทัด

N = 1 แสดง 2 บรรทัด

กรณีมากกว่า 2 บรรทัดให้ SET N = 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F : เป็นการ SET ขนาดของ DOT การแสดงผล 5x7 หรือ 5x10

F = 0 เป็นแบบ 5x7

F = 1 เป็นแบบ 5x10

N F	No. of display lines	Character font	Duty factor	Remarkers
0 0	1	5x7 dots	$\frac{1}{8}$	
0 1	1	5x10 dots	$\frac{1}{11}$	
1 *	2	5x7 dots	$\frac{1}{16}$	Cannot display 2 lines with 5 x10 dot character font

* No Effect

7. SET CG RAM ADDRESS

SET CG RAM ADDRESS

RS	R/W	DB7	-----							DB0
0	0	0	A	A	A	A	A	A	A	

Code <----- Higher Order bits Lower Order bits ----->

ใน HD44780 นั้นจะมีหน่วยความจำอยู่ 2 ชุดคือ DISPLAY DATA RAM (DD RAM) จำนวน 80x8 BIT และ CHARACTER GENERATOR RAM จำนวน 512 BIT และ 7200 BIT คำสั่งนี้จะเป็นการ SET ADDRESS ใน CG RAM โดยต้องทำการ SET ADDRESS ก่อนเขียนหรืออ่านข้อมูลจาก CG RAM คว

8. SET DD RAM ADDRESS

SET DD RAM ADDRESS

RS	R/W	DB7	-----							DB0
0	0	1	A	A	A	A	A	A	A	

Code <----- Higher Order bits Lower Order bits ----->

เป็นคำสั่ง SET ค่า ADDRESS ใน DD RAM ในการเขียนหรืออ่านจาก DD RAM (DD RAM คือส่วนที่จะแสดงผลหน้าจอ LCD) โดยจำนวน ADDRESS ที่จะเกิดขึ้นบนหน้าจอ LCD จะอยู่กับการ SET ค่า N ค่าย

N = 0 (1 บรรทัด) ADDRESS อยู่ที่ 00H-4FH

N = 1 (2 บรรทัด) ADDRESS บรรทัดที่ 1 อยู่ที่ 00H-27H บรรทัดที่ 2 อยู่ที่ 40H-67H

แบบการจัด ADDRESS ของ DD RAM หน้าจอ LCD แบบ 16 ตัวอักษร 2 บรรทัด, 16 ตัวอักษร 4 บรรทัด, 20 ตัวอักษร 1 บรรทัด, 20 ตัวอักษร 2 บรรทัด และ 40 ตัวอักษร 2 บรรทัด

9. READ BUSY FLAG AND ADDRESS

READ BUSY FLAG AND ADDRESS

RS	R/W	DB7	-----							DB0
0	1	BF	A	A	A	A	A	A	A	

Code <----- Higher Order bits Lower Order bits ----->

เป็นคำสั่งอ่านค่า BUSY FLAG ซึ่งจะเป็นตัวบอกว่า IC เบอร์ HD44780 นี้อยู่ในขบวนการทำงานภายใน (ไม่รับข้อมูล) หรือมีสภาพพร้อมที่จะรับข้อมูล

BF = 1 อยู่ในขบวนการทำงานภายใน ไม่พร้อมที่จะรับข้อมูลหรือคำสั่ง

BF = 0 พร้อมที่จะรับข้อมูลหรือคำสั่งได้

และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูล ADDRESS ของ CG RAM หรือ DD RAM ค่าย

10. WRITE DATA TO CG หรือ DD RAM

WRITE DATA FROM CG หรือ DD RAM

RS	R/W	DB7	-----							DB0
0	1	BF	A	A	A	A	A	A	A	

Code <----- Higher Order bits Lower Order bits ----->

เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD RAM โดยเมื่อเขียนข้อมูลและ ADDRESS จะเพิ่มหรือลดโดยอัตโนมัติตามคำสั่งที่ SET ใน ENTRY MODE โดยเมื่อเขียนข้อมูล ADDRESS จะเพิ่มหรือลดโดยอัตโนมัติตาม FLAGS ที่ SET ใน ENTRY MODE ข้อกำหนดที่ทำให้ทราบว่า เป็นการเขียนข้อมูลของ CG RAM หรือ DD RAM มาจากการ SET ADDRESS ของ CG RAM ขึ้นมาก่อนจะเขียนข้อมูล

11. READ DATA FROM CG หรือ DD RAM

READ DATA FROM CG หรือ DD RAM

RS	R/W	DB7	-----						DB0
1	1	D	D	D	D	D	D	D	D

Code <----- Higher Order bits Lower Order bits ----->

เป็นคำสั่งอ่านข้อมูลจาก CG หรือ DD RAM โดยก่อนอ่านค่าจาก DD RAM หรือ CG RAM นี้ควรใช้คำสั่ง SET ADDRESS ของ CG RAM ก่อนเพื่อให้ทราบว่าข้อมูลที่อ่านได้นั้นเป็น DD RAM

จากตารางการทำงานจะพบว่าการใช้งาน LCD MODULE นั้นง่ายเพียงแต่เราส่งคำสั่งเริ่มต้นและ SET ความต้องการขนาดตัวอักษร, CURSOR หลังจากนั้นเราก็สามารถที่จะเขียนตัวอักษรเข้าไปใน DD RAM ตามตารางตัวอักษรที่อยู่ใน DATA SHEET หลังจากนั้นจะเกิดตัวอักษรปรากฏขึ้นบนจอภาพหลักของ LCD และเราก็ยังสามารถกำหนดตำแหน่งตัวอักษรที่เกิดขึ้นบนจอภาพได้โดยการ SET DD RAM ADDRESS ตามตารางที่ใหม่ ในหัวข้อ SET DD RAM ADDRESS

รุ่น	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	9	9	9	9	
DMC20	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	0	1	2	3
2	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	cf	d	d	d	d
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e		0	1	2	3

ISD 12xx /14xx SERIES (ไอซีบันทึกเสียง)

ไอซีในตระกูล ISD นี้อาศัยเทคโนโลยีการบันทึกเสียงทางอนาล็อกโดยตรงและภายในประกอบไปด้วยส่วนต่าง ๆ ที่ทำหน้าที่เก็บข้อมูลไว้ภายใน หรือเรียกว่า NV RAM (Nonvolatile RAM) ทำให้สามารถบันทึกเสียงได้นานตั้งแต่ 10-20 วินาที ตามเบอร์ไอซีในตระกูล ISD 12xx/14xx ซึ่งการใช้งานและการจัดขาของไอซีตระกูลนี้เหมือนกันหมดต่างกันตรงความสามารถในการบันทึกเสียงเท่านั้นเอง

คุณลักษณะที่แตกต่างไปจากไอซีเสียงทั่วไปก็คือระยะเวลาบันทึกเสียงด้วยตัวของไอซีเอง (ไม่มีหน่วยความจำมาต่อเพิ่มภายนอก) สามารถบันทึกเสียงได้นาน โดยจะกำหนดมาเป็นเบอร์ตั้งแต่เบอร์ ไอซีจะมีระยะเวลาต่างกันในการบันทึกเสียงและที่อยู่ในตระกูล ISD เช่นกัน

คุณสมบัติของ ISD 12xx/14xx

- เมื่อใช้ในฟังก์ชันการบันทึกและเล่นกลับด้วยตัวไอซีเองง่ายมาก
 - ไม่มีไอซีเบอร์อื่นๆ ประกอบเพิ่มเติมภายนอก
 - ต่ออุปกรณ์พาสซีฟภายนอกน้อยมาก
- ให้ระดับสัญญาณในการบันทึกที่มีประสิทธิภาพสูง
- สามารถต่อกับสวิทช์ควบคุมการบันทึกเล่นกลับ, หยุดชั่วคราวและปรับระดับสัญญาณต่างๆ ได้
- ข้อมูลที่ถูกบันทึกไว้ไม่สูญหายถึงแม้ว่าจะไม่มีแรงดันจ่ายให้กับไอซี และไม่ต้องการแบตเตอรี่สำรอง
- เก็บข้อมูลได้นานถึง 100 ปี แม้มันไม่มีแรงดันไฟเลี้ยง
- สามารถบันทึกใหม่ได้ 100,000 ครั้งปกติ
- มีวงจรฐานเวลาดำเนิน
- ไม่มีการโปรแกรมในตัวไอซีและไม่คงพัฒนาระบบเพิ่มเติมเพื่อให้ทำงานได้
- มีระบบสแตนด์บายเพื่อประหยัดพลังงานจากแหล่งจ่ายเมื่อไม่มีการบันทึกหรือเล่นกลับ
- ใช้แรงดันไฟเลี้ยงเดียว กินกระแสขณะสแตนด์บาย 0.5 ไมโครแอมป์

คุณสมบัติทางไฟฟ้าของ ISD 12xx /14xx

จากตารางนี้จะทำให้ทราบว่าค่าพารามิเตอร์บางอย่างที่แตกต่างกันของไอซีในแต่ละเบอร์ในตระกูล ISD 12xx/14xx นี้ ซึ่งบอกถึงความสามารถของระยะเวลาในการบันทึกที่แตกต่างกัน รวมทั้งความถี่ของการสุ่มสัญญาณก็ต่างกัน

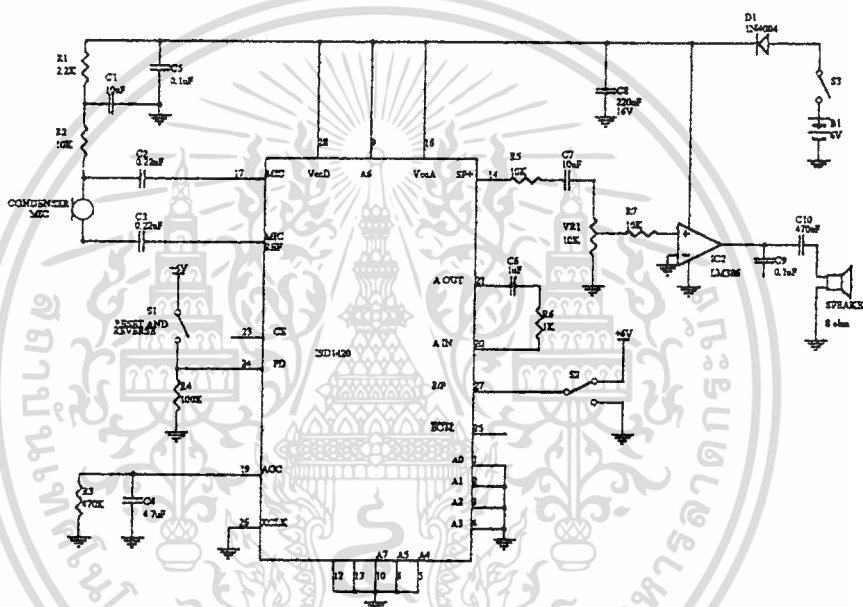
โดยทั่วไปของ ISD 12xx/14xx แล้ว ก็มีรูปร่างหน้าตาเหมือนกับไอซีโดยทั่วไป แต่ว่า ขาใช้งานและบล็อกไอเทมการทำงานภายในทำงานไม่เหมือนกัน โดยเฉพาะไอซีในตระกูล ISD 12xx/14xx นี้จะถูกผลิตขึ้นมาเป็น ไอซีที่ทำหน้าที่บันทึกและเล่นกลับ จนจบกระบวนการภายใน ไอซีเพียงตัวเดียว เพื่อประโยชน์และต้องการให้เป็นอุปกรณ์ ที่ทำหน้าที่บันทึกและเล่นกลับ ก็ ถือได้ว่าดีมากไม่แตกต่างจากการบันทึกลงดิสก์เทปคลาสเซต

ตารางที่ 1 คุณสมบัติทางไฟฟ้าของตระกูล ISD12xx/14xx

สัญลักษณ์	รายละเอียดคุณสมบัติ	ISD1210	ISD1212	ISD1416	ISD1420	หน่วย
Fs	อัตราความถี่ในการแซมปลิงสัญญาณ	6.4	5.8	8	6.4	kHz
BW	ความกว้างของแบนด์	2.7	2.8	3.4	2.7	kHz
Trpw	ความกว้างของพัลส์ขณะบันทึก	10	12	16	20	วินาที
Tplay	ระยะเวลาในการบันทึก(สูงสุด)	10	12	16	20	วินาที
Tled2	การหน่วงเวลาที่ขา RECLEDD ขณะเปิด	48.6	58.3	38.9	48.6	มิลลิวินาที
Trpad	การหน่วงเวลาขณะเริ่มบันทึก	32	39	26	32	มิลลิวินาที
Trpdd	การหน่วงเวลาขณะหยุดบันทึก	32	39	26	32	มิลลิวินาที
Tppud	การหน่วงเวลาขณะเล่นกลับ	32	39	26	32	มิลลิวินาที
Tppdd	การหน่วงเวลาขณะเล่นกลับสิ้นสุด	8.1	9.7	6.5	8.1	มิลลิวินาที
THD	ความเพี้ยนทางฮาร์โมนิกรวม	1	1	1	1	%
TledI	หน่วงเวลาขณะเริ่มบันทึกของ LED	5	5	5	5	ไมโครวินาที
Pout	กำลังขับลำโพงทางเอาต์พุต	12.2	12.2	12.2	12.2	มิลลิวัตต์
Vout	แรงดันตกคร่อมขาต่อลำโพง	2.5	2.5	2.5	2.5	โวลต์ P-P
Vin1	แรงดันอินพุตที่ไม่โครโฟน	20	20	20	20	มิลลิโวลต์
Vin2	แรงดันอินพุตอนุบาลอก	50	50	50	50	มิลลิโวลต์

ภายในตัวไอซีจะประกอบด้วยส่วนการทำงานที่สำคัญทุกส่วน โดยมีอุปกรณ์พาสซีฟต่อภายในก็มีวงจรกำเนิดความถี่ฐานเวลาชนิดซิมอส, วงจรขยายสัญญาณไมโครโฟน, วงจรควบคุมอัตราการขยายอัตโนมัติ, วงจรกรองความถี่และวงจรขยายสัญญาณออกสู่ลำโพงโดยตรงจากไอซีตัวนี้แต่ไม่ดังมากพอ ก็สามารถขับลำโพงให้ได้ยินเสียงดังมากขึ้น

สัญญาณที่จะทำการบันทึกจะถูกเก็บหรือบันทึกลงบนหน่วยความจำภายในไอซี ซึ่งเป็นหน่วยความจำที่ไม่ต้องการแรงดันไฟฟ้าสำรองขณะเก็บข้อมูลไว้ได้นานเป็น 100 ปี ลักษณะการบันทึกบนหน่วยความจำภายในไอซีตัวนี้จะทำการบันทึกสัญญาณ Analog โดยตรง (Direct Analog Storage Technology:DAST):ซึ่งสัญญาณ Analog นี้อาจจะเป็นสัญญาณย่านความถี่ 20 Hz. ถึง 22 kHz. การบันทึกนี้สัญญาณจะถูกเก็บไว้ในหน่วยความจำภายในชนิด EEPROM ซึ่ง เป็นอีพროมที่สามารถบันทึก 100,000 ครั้งตลอดอายุการใช้งาน



รูปที่ 21

แสดงวงจรบันทึกเสียงด้วยไอซีเบอร์ ISD 1420

การทำงานเบื้องต้น

การทำงานเบื้องต้นของ ISD 12xx/14xx จะเป็นชิป ไอซีที่เพียงตัวเดียวและมีสัญญาณแบบ เดี่ยว (single signal) เพื่อควบคุมที่ขา REC และสัญญาณควบคุมการเล่นกลับจะถูกควบคุมด้วย สวิตช์ควบคุมสองสวิตช์ คือควบคุมที่ขา PLAYE และ PLAYL นอกจากนั้นหากต้องการให้ สามารถควบคุมการบันทึกได้หลายๆ ลักษณะก็สามารถใช้ขาแอกแคเรสโตน มาทำการประยุกต์ใช้ งานควบคุมได้เช่นกัน และการทำงานของขาใช้งานแต่ละขาถืออธิบายสั้นๆ ถึงหน้าที่และลักษณะ การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพของเสียงที่บันทึก

ประสิทธิภาพของสัญญาณที่ทำการบันทึกและเล่นกลับจะมีคุณภาพของสัญญาณที่ทำการบันทึกและเล่นกลับจะมีคุณภาพดีมากขึ้นเนื่องจากการบันทึกและเล่นกลับจะใช้เทคโนโลยี DAST ดังกล่าวมาแล้ว โดยที่สัญญาณที่ทำการบันทึกหรือสัญญาณเสียงพูดที่เข้ามาทางอินพุตจะถูกบันทึกหรือเก็บเข้าไปไว้ใน EEPROM ซึ่งเป็นหน่วยความจำภายในโดยตรงด้วยสัญญาณแบบอนาล็อกและการเล่นกลับออกมาก็จะเหมือนกับสัญญาณที่ก่อนทำการบันทึกแน่นอน ปิดตัวเองเมื่อไม่มีการบันทึกหรือเล่นกลับ

ในขณะที่วงรอบหรือขั้นตอนการทำงานเล่นกลับหรือบันทึกสิ้นสุดลง ไอซีนี้ก็จะมีการทำงานให้ตัดเข้ามาสู่โหมดของการสแตนด์บายเพื่อให้ปริมาณการใช้กำลังต่ำเพื่อต้องการประหยัดแบตเตอรี่ ซึ่งกินกระแสเพียง 0.5 ไมโครแอมป์เท่านั้น ในช่วงที่ทำการเล่นกลับจบลงวงจรรภายในก็จะตัดกลับมาสู่สถานะสแตนด์บาย ในโหมดของการบันทึก หลังจากทำการบันทึกเสร็จสิ้นลง ก็จะกลับมาสู่โหมดสแตนด์บาย เมื่อเราควบคุม REC มีระดับลอจิกเป็น "1" การควบคุมการบันทึก(REC)

ที่เราควบคุมการบันทึกทางอินพุตนี้จะต้องการระดับลอจิก "0" เพื่อทำการบันทึกสัญญาณและจะเริ่มทำการบันทึกเมื่อระดับลอจิกที่เรา REC นี้เป็นลอจิก "0" และสถานะลอจิกที่ขา REC นี้จะต้องคงสถานะอยู่ที่ "0" ตลอดขณะที่ขา REC นี้จะต้องได้รับสัญญาณให้ทำการบันทึก และการบันทึกที่ขา REC จะต้องได้รับสัญญาณให้ทำการบันทึกก่อนที่จะทำการเล่นกลับหรือก่อนที่จะมีสัญญาณมาควบคุมที่ขา PLAYE หรือ ขา PLAYL ถ้าที่ขา REC มีระดับลอจิก "0" เพิ่มขึ้นไปเป็นค่าแรงดันบวก(ขึ้นไปเป็น "1") ก็จะเข้าสู่การทำงานของการเล่นกลับ การควบคุมการเล่นกลับ (PLAYE)

เมื่อเราควบคุมการเล่นกลับนี้ได้รับระดับลอจิกเป็น "0" หรือได้รับการกระตุ้นด้วยลอจิก "0" ที่อินพุตนี้วงจรถือจะเริ่มทำการเล่นกลับเพื่อนำข้อมูลที่ถูกบันทึกอยู่แสดงออกมาทางลำโพง การเล่นกลับฟังก์ชันนี้ จะเป็นการเล่นกลับอย่างต่อเนื่อง จนกว่าจะถึงข้อมูลสุดท้ายที่ทำการบันทึกตามเวลาที่กำหนดไว้ (10-20 วินาที) หรือการเล่นกลับจนกว่าข้อมูลที่บันทึกไว้ใน EEPROM ทุกข้อมูลจะถูกเล่นกลับออกมาทั้งหมด ซึ่งเป็นการเล่นกลับอย่างสมบูรณ์ หลังจากนั้นก็จะตัดเข้าสู่โหมดสแตนด์บาย ในระหว่างที่กำลังอยู่ในสถานะเล่นกลับนั้นทันทีที่ขา PLAYE มีสถานะเป็น "1" การเล่นกลับก็จะหยุดลงทันที การควบคุมการเล่นกลับ (PLAYL)

เมื่อขาอินพุตนี้มีการเปลี่ยนแปลงระดับลอจิกเป็น "1" ไปเป็น "0" จะเป็นการเล่นกลับแบบต่อเนื่องจนกระทั่งที่ขา (PLAYL) เพิ่มขึ้นเป็น "1" หมายถึง เกิดการตรวจจบการเล่นสิ้นสุดลงแล้ว หรือจบสิ้นข้อมูลที่เก็บไว้ใน EEPROM แล้ว และก็จะกลับมาสู่สถานะสแตนด์บาย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้โดยไม่ได้รับอนุญาต หากต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายบริการลูกค้า

DS1620 เทอร์โมมิเตอร์

ในบรรดาอุปกรณ์ตรวจจับอุณหภูมิโดยทั่วไปแล้ว จะให้เอาต์พุตออกมาเป็นสัญญาณอนาล็อกซึ่งอาจจะสามารถอ่านค่าได้เลยหรือใช้งานร่วมกับชุดแสดงผลอื่นๆเช่นดิจิตอลโวลท์มิเตอร์อนาล็อกมิเตอร์(มิเตอร์เข็ม)และก็สะดวกพอสมควรกับการใช้งาน แต่ข้อเสียก็คือการประยุกต์ใช้งานตรวจจับอุณหภูมิต่าง ๆ ก่อนข้างจะดูจำกัดทางด้านลักษณะการใช้งาน และความละเอียดของค่าที่แสดงผลออกมา บางครั้งไม่เป็นเชิงเส้น และยังรวมไปถึงการควบคุมการทำงานของอุปกรณ์ตรวจจับอุณหภูมิเองทำได้ลำบาก มีฟังก์ชันการทำงานที่เสียน้อยมากหรือไม่เลย

สำหรับอุปกรณ์ตรวจจับอุณหภูมินี้จะเป็นทางออกให้กับปัญหาหรืออุปสรรคที่เกิดขึ้นในอุปกรณ์ตัวเก่า นั่นคือไอซี DS1620 ทำหน้าที่เป็นตัวตรวจจับอุณหภูมิและฟังก์ชันเทอร์โมสตัดในตัวด้วย อาศัยตัวอุปกรณ์เองทำหน้าที่ตรวจจับการเปลี่ยนแปลงของอุณหภูมิในบริเวณโดยรอบตัวไอซีจากการที่สามารถเชื่อมต่อกับระบบ CPU ได้ จึงทำให้สามารถเขียนโปรแกรมควบคุมการทำงานและการแสดงผลได้อย่างมากมาย โดยเฉพาะฟังก์ชันที่แตกต่างไปจากอุปกรณ์ตัวเก่า ทำให้สามารถประยุกต์ใช้งานได้หลายลักษณะ ก็ขึ้นอยู่กับ การเขียน โปรแกรมควบคุมซึ่งรวมไปถึงการแสดงผลที่มีลูกเล่นมากกว่า

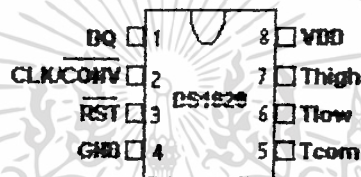
คุณสมบัติของ DS1620

หากจะกล่าวถึงคุณสมบัติทางการใช้งานและความสามารถที่เป็นจุดเด่นของไอซีตัวนี้แบ่งได้เป็นหัวข้อดังนี้

- ไม่มีอุปกรณ์ประกอบรวมภายนอก
- วัดอุณหภูมิได้ตั้งแต่ -55 องศาเซลเซียสถึง +125 องศาเซลเซียส ในทุกการเปลี่ยนแปลงเพิ่มขึ้น 0.5 องศาเซลเซียส หรือ -67 องศาฟาเรนไฮต์ทุกการเปลี่ยนแปลงเพิ่มขึ้น 0.9 องศา ฟาเรนไฮต์
- การอ่านข้อมูลของอุณหภูมิออกมาเป็นแบบ 9 บิต
- แปลงค่าอุณหภูมิไปเป็นสัญญาณดิจิตอลได้ภายใน 1 วินาที
- แบบเทอร์โมสตัดสามารถใหม่แบบจำกัดอุณหภูมิหรือแบบเปลี่ยนแปลงอุณหภูมิได้
- การอ่านและเขียนข้อมูลจากตัวตรวจจับผ่านสายนำสัญญาณ 3 เส้น อินเดอร์เฟสกับพอร์คอนุกรม (CLK , DQ , \overline{RST})
- สามารถประยุกต์ใช้งานได้กว้างเช่น วงจรควบคุมเทอร์โมสตัด, ระบบควบคุมอุณหภูมิในโรงงานอุตสาหกรรม, เทอร์โมมิเตอร์ และตัวตรวจจับการเปลี่ยนแปลงของอุณหภูมิทั่วไป เป็นต้น

จากคุณสมบัติดังกล่าวข้างต้น ก็ทำให้พอจะมองออกได้ว่าสามารถที่จะพัฒนาและประยุกต์ใช้งานในระบบอุตสาหกรรมโรงงานขนาดใหญ่ได้ รวมไปถึงการควบคุมการทำงานได้ง่ายจากจุดควบคุมจุดเดียวในกรณีที่ต้องการออกแบบใช้งานเป็นชุดตรวจจับอุณหภูมิหลาย ๆ จุดและกระจายอยู่ในบริเวณกว้างเนื่องจากการโปรแกรมผ่าน CPU เพื่อใช้ในการควบคุมและแสดงผล ดังนั้นจึงเป็นการง่ายที่จะต่อเป็นระบบตรวจจับอุณหภูมิหลายจุดโดยมีการเรียกตรวจสอบแต่ละจุดบนมอนิเตอร์ได้หรือเป็นแบบลุ่มแสดงผลก็ได้ ขึ้นอยู่กับการพัฒนาโปรแกรมควบคุมผ่าน CPU เท่านั้นเอง

หลักการการทำงานเบื้องต้น



รูปที่ 22

แสดงตำแหน่งและชื่อเรียกประจำขา

สามารถแยกการทำงานคร่าว ๆ อยู่ 2 ส่วน คือส่วนทำการวัดอุณหภูมิและส่วนควบคุมเทอโมสแตต นอกจากนั้นการทำงานส่วนประกอบอื่นๆ อีกที่บรรจุอยู่ภายในไอซีและที่จะต้องควบคุมจากภายนอกอีกทางหนึ่ง แต่ก่อนที่จะเข้าไปทำความเข้าใจกับการทำงานของไอซี มาทำความเข้าใจกับรูปร่างของไอซีและลักษณะชื่อเรียกขาใช้งานแต่ละขากันก่อน แสดงไว้ในรูปที่ 1 เป็นรูปแสดงลักษณะของ DS1620 ซึ่งเป็นไอซี DIP 8 ขาและมีชื่อเรียกประจำขาใช้งานแต่ละขาไว้ให้ทราบด้วย นอกจากนั้นหน้าที่การทำงานและความสำคัญของขาแต่ละขาจะได้อธิบายไว้ให้ทราบเพื่อง่ายแก่การนำไปใช้งาน

1. ขา DQ (ขา 1) เป็นขาข้อมูลอินพุต/เอาต์พุตของไอซีซึ่งจะให้ข้อมูลที่ผ่านการแปลงสัญญาณแรงดันที่เกิดจากการตรวจจับอุณหภูมิของไอซีแล้วแปลงเป็นสัญญาณทางดิจิทัลออกมาทางขา 1 นี้
2. ขา CLK / CONV (ขา 2) เป็นขารับสัญญาณนาฬิกาฐานเวลาและฟังก์ชันของการวัดอุณหภูมิขานี้จะทำหน้าที่ควบคุมการแปลงสัญญาณ แต่ถ้าหากใช้เป็นฟังก์ชันเทอโมสแตตอย่างเดียวนั้นไม่ใช้งานและจะไม่เป็นการอินเทอร์เฟสแบบ 3 เส้น
3. ขา RST (ขา 3) เป็นขาควบคุมการรีเซ็ตข้อมูลทางอินพุต/เอาต์พุตของไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ขาของ GND (ขา 4) ขาค้อกับไฟเลี้ยงค่านกราวด์
5. ขา Tcom (ขา 5) เป็นขาควบคุมการกระตุ้นรวมของฟังก์ชันเทอร์โมสตัต จะให้เอาต์พุตออกมาเป็น "0" เมื่ออุณหภูมิที่วัดมีค่าต่ำกว่าค่าอุณหภูมิต่ำสุดที่กำหนดไว้ (TL)
6. ขา Tlow (ขา 6) เป็นขากระตุ้นเมื่ออุณหภูมิลดลงกว่าค่าอุณหภูมิที่กำหนดไว้ต่ำสุด (TL) จะให้เอาต์พุตออกมาเป็น "1" เพื่อแสดงผลหรือกระตุ้นส่วนทำงานต่อไป
7. ขา Thigh (ขา 7) เป็นขากระตุ้นโดยเมื่อค่าอุณหภูมิที่วัดได้มีค่าสูงกว่าค่าอุณหภูมิที่กำหนดไว้สูงสุด (TH) เอาต์พุตจะกระตุ้นออกมาเป็น "1" เพื่อควบคุมส่วนการทำงานต่อไปหรือแสดงผล
8. ขา VDD (ขา 8) เป็นขารับแรงดันไฟเลี้ยงไอซี +5 V

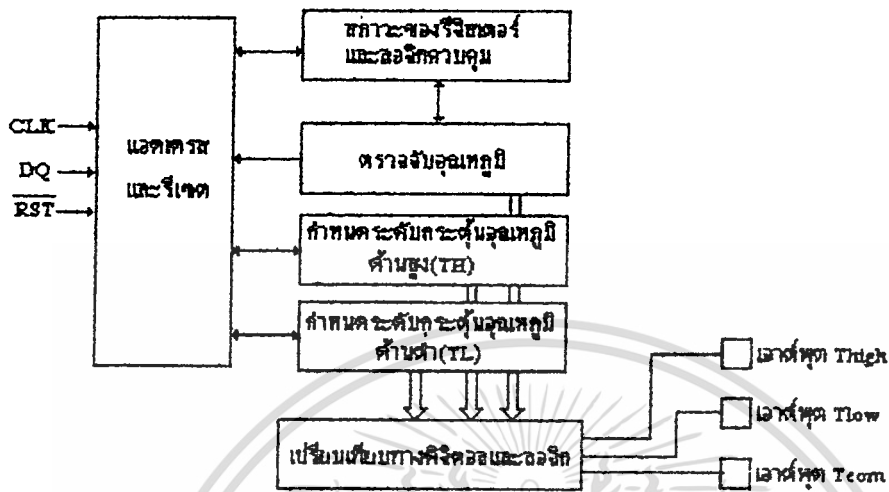
การวัดค่าอุณหภูมิ

ตารางที่ 2 ลักษณะข้อมูลที่ออกมาจากขา DQ ทั้งสองรูปแบบ

ค่าอุณหภูมิ	ดิจิตอลเอาต์พุต	
	ฐานสอง	ฐานสิบหก
+125 องศาเซลเซียส	0 11111010	00FAh
+25 องศาเซลเซียส	0 00110010	0082h
1/2 องศาเซลเซียส	0 00000001	0001h
0 องศาเซลเซียส	0 00000000	0000h
-1/2 องศาเซลเซียส	1 11111111	01FFh
-25 องศาเซลเซียส	1 11001110	01CEh
-55 องศาเซลเซียส	1 10010010	0192h

ตัวตรวจจับอุณหภูมิ DS1620 เมื่อใช้วัดอุณหภูมิแล้วมันสามารถวัดผ่านตัวมันเองได้โดยตรงอาจจะติดตั้งบนบอร์ดที่ต้องการวัดอุณหภูมิ หรือบอร์ดเล็กๆ สำหรับติดตั้ง DS1620 เพื่อสะดวกในการติดตั้งตำแหน่งต่างๆ ซึ่งก็สามารถดัดแปลงใช้งานตามเทคนิคของแต่ละโอกาสที่จะทำการวัด ค่าอุณหภูมิที่ทำการวัดจะถูกอ่านออกมาเป็นตัวเลขดิจิตอลขนาด 9 บิต ซึ่งก็มีสองรูปแบบของรหัสเอาต์พุตคือเป็นค่าตัวเลขฐานสอง กับค่าตัวเลขฐานสิบหก ดังแสดงไว้ตารางที่ 2 ซึ่งเป็นค่าของตัวเลขทั้งสองรูปแบบที่ค่าอุณหภูมิการตรวจจับแตกต่างกัน ข้อมูลที่ออกมาทางเอาต์พุตจะถูกต่ออินเทอร์เฟซกับระบบ CPU ผ่านทางพอร์ตอนุกรม ดูบล็อกไดอะแกรมการทำงาน

ภายในตัว DS1620 ได้ในรูปแบบที่ 2 ย่านการวัดอุณหภูมิ -55 ถึง +125 องศาเซลเซียสทุกการเปลี่ยนแปลงอุณหภูมิที่ 0.5 องศาเซลเซียส



รูปที่ 23

บล็อกไดอะแกรมภายในของ DS1620

ข้อมูลที่ถูกรับออก ไปทางบัสข้อมูลนั้นจะประกอบด้วยบิต LSB เป็นส่วนแรกที่ถูกส่งออกไปและข้อมูลของอุณหภูมิจะถูกเขียนและอ่านออกมาจาก DS1620 เป็นขนาด 9 บิต (ที่ขา RST จะเป็น "0" หลังจากบิตที่ 9 (MSB) ถูกส่งไปเป็นอันดับสุดท้าย ค่อยจากนั้นเมื่อสิ้นสุดแล้ว MSB จะเป็น "0")

การควบคุมเทอร์โมสแตต

การควบคุมเทอร์โมสแตตจะอาศัยขาเอาต์พุต 3 ขา คือ Thigh Flow และ Tcom ซึ่งลักษณะของสัญญาณเอาต์พุตแสดงไว้ในรูปที่ 4 โดยเมื่ออุณหภูมิที่ DS1620 ตรวจจับได้มีค่าสูงเกินกว่าค่าอุณหภูมิที่กำหนดไว้ในระดับสูงสุดไนรีจิสเตอร์ที่ขาเอาต์พุต Thigh จะกลับมามีสถานะเป็น "1" และจะคงสถานะอยู่เช่นนี้ไปเรื่อยๆ จนกว่าอุณหภูมิที่ตรวจจับด้วย DS1620 มีค่าลดลงต่ำกว่าค่าอุณหภูมิสูงสุดที่กำหนดไว้ในรีจิสเตอร์ สัญญาณเอาต์พุตที่ขา Thigh นี้สามารถนำไปขับเคลื่อนแสดงผลให้ทราบว่าอุณหภูมิขณะนั้นของอุปกรณ์ที่ตรวจจับมีสถานะเป็นเช่นไรหรือใช้ควบคุมการทำงานอย่างอื่น ได้อย่างกว้างขวางที่อยู่ในลักษณะควบคุมไม่ให้อุณหภูมิสูงเกิน เป็นต้น

การควบคุมเทอร์โมสแตคที่ขา Thow จะมีการทำงานที่ตรงข้ามกับขา Thigh โดยที่เมื่อ DS1620 วัดอุณหภูมิได้เท่ากับหรือต่ำกว่าค่าอุณหภูมิค่าสุดที่กำหนดไว้ในรีจิสเตอร์ เอาต์พุตของขา Thow นี้จะโชว์แสดงเป็นค่าอุณหภูมิต่ำกว่ากำหนดหรือใช้ควบคุมในลักษณะควบคุมการทำงานเมื่ออุณหภูมิ

ที่ขาควบคุม Tcom เอาต์พุตที่ขาควบคุมนี้จะให้เอาต์พุตออกมาเป็น “1” เมื่อทำการวัดอุณหภูมิที่สูงเกินกว่าค่าสูงสุดที่กำหนดไว้ (TH) และจะคงสถานะอยู่จนกว่าค่าอุณหภูมิจะลดลงมาต่ำกว่าค่าอุณหภูมิค่าต่ำสุดที่กำหนดไว้จึงจะให้สถานะออกมาเป็น “0” ซึ่งค่าการตรวจจับระหว่างค่าอุณหภูมิค่าต่ำสุดถึงค่าอุณหภูมิสูงสุดสามารถปรับเปลี่ยนได้และเอาต์พุตนี้สามารถนำไปควบคุมการคงที่ของอุณหภูมิได้ โดยจะครอบคลุมในลักษณะอุณหภูมิต่ำกว่าก็ตัดสูงกว่าก็ตัด เป็นต้น



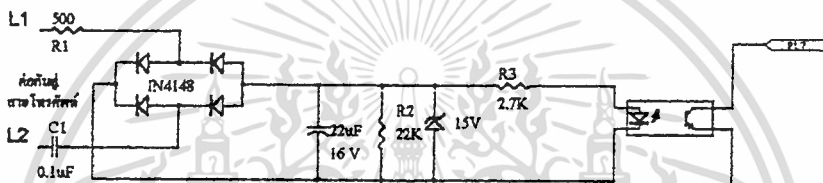
บทที่ 3

การคำนวณและการออกแบบวงจร

ส่วนประกอบของการทำงานทั้งหมด ซึ่งนำสัญญาณเข้ามาติดต่อกับ CPU ซึ่งเราสามารถแบ่งแยกออกได้เป็นส่วนต่างๆดังนี้

1. ส่วนตรวจจับสัญญาณกระดิ่ง (RINGING DETECTOR)

การตรวจจับสัญญาณกระดิ่งของสายโทรศัพท์ ซึ่งจากระดับแรงดันประมาณ 100 Vp-p ความถี่ประมาณ 25Hz



รูปที่ 24

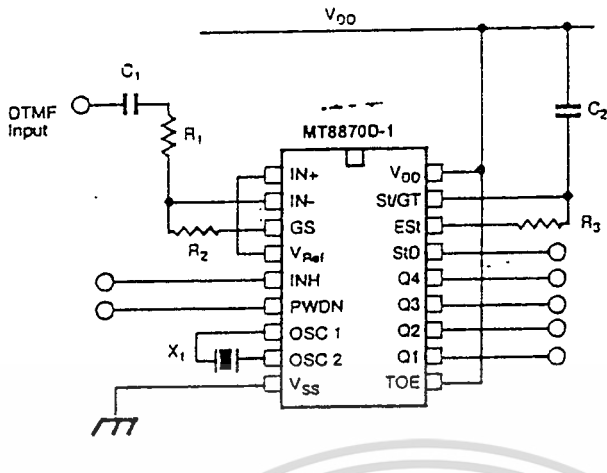
วงจรตรวจจับสัญญาณกระดิ่ง

จากรูปวงจรจะเห็นว่า แรงดันของสัญญาณกระดิ่งจะถูกแปลงเป็นแรงดัน DC ก่อนโดย C2 จะเป็นตัวกรองระดับแรงดันให้เรียบก่อน ขณะเดียวกันซีเนอร์ไดโอดจะปรับระดับแรงให้เหลือ 15 V เพื่อต่อเข้ากับตัวไอซีเบอร์ 4N26 แรงดันพัลส์ไฟตรงที่ตกคร่อม ซีเนอร์ไดโอดจะถูกจ่ายไบแอสให้กับ LED ภายในออปโต ไอซี 4N26 โดย R3 จำกัดกระแสให้ LED ใน ออปโต เมื่อมีสัญญาณกระดิ่งเข้ามาที่คู่สาย ขาคอลเล็กเตอร์ของโฟโตทรานซิสเตอร์ในออปโตจะต่อกับกราวด์เนื่องจาก ได้รับไบแอสตรง

2. ส่วนการถอดสัญญาณกดหมายเลข (DTMF DETECTOR)

MT8870 เป็นอุปกรณ์ถอดสัญญาณ โทนอนออกมาเป็น รหัส BCD โดยมีตัวเก็บประจุคัปปลิงสัญญาณ มาเข้าอินพุตร่วมกับตัวต้านทานเน็ตเวอร์ ไอซี MT8870 จะมี CRYSTAL เป็นตัวกำหนดความถี่แน่นอน ความต้านทาน R1 กับ R2 เป็นตัวกำหนดความแตกต่างของสัญญาณทางอินพุต การถอดรหัสออกมาทางเอาต์พุต (Q1-Q4) เอาต์พุตจะออกเป็น BCD และที่ขา STD เมื่อมีการออกหมายเลขหรือการถอดรหัส จะส่ง PULSE ออกมา 1 ลูกเพื่อแสดงว่ามีการถอดรหัสออกมาสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 25

วงจรการถอดรหัสสัญญาณหมายเลข

8. ส่วนประมวลผลกลางและหน่วยความจำ

ควบคุมด้วยไมโครคอนโทรลเลอร์ 80C32 ซึ่งเป็นตระกูลเดียวกับ MCS-51 มีหน้าที่ประมวลผลกลางขนาด 8 บิต สามารถอ้างตำแหน่งของหน่วยความจำโปรแกรมได้ 64 กิโลไบต์ และสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์ เบอร์ 80C32 จะมีหน่วยความจำแบบ RAM ภายในจำนวน 250 ไบต์ คือตำแหน่ง 00H - FFH การกำหนดให้ CPU ปฏิบัติตามโปรแกรมภายในหรือภายนอก โดยตัวใส่ JUMPPER (JP1) จะเป็นการ EXECUTE โปรแกรมภายนอก (EXTERNAL) ซึ่งหมายถึงการต่อขา EA ของ CPU ลงกราวด์

หน่วยความจำในไมโครคอนโทรลเลอร์ แบ่งออกเป็น 2 ส่วน คือ หน่วยความจำโปรแกรม

(program memory) และหน่วยความจำข้อมูล (data memory) เราได้ออกแบบหน่วยความจำโปรแกรม ดังรูปที่ 26 หน่วยความจำ U3 สามารถเลือกได้ 2 ขนาด คือ 8 kB โดยใช้ EPROM เบอร์ 2764 และขนาด 32 kB โดยใช้ EPROM เบอร์ 27256 โดยเลือก JP4

หน่วยความจำสำหรับข้อมูลในที่นี้เราหมายถึง RAM หรือ หน่วยความจำที่สามารถอ่านและเขียนข้อมูลลงได้ U4 และ U5 เป็นหน่วยความจำข้อมูลซึ่งมี Address ที่ 0000-7FFF (U4) ,8000-FFFF (U5) โดยสามารถอ่านและเขียนข้อมูลลงได้ U4 และ U5 เป็นหน่วยความจำข้อมูลโดยเราสามารถเลือกขนาดหน่วยความจำได้ โดยชุด Jumper

JP3 จะเป็นตัวเลือกตำแหน่งเริ่มต้นของหน่วยความจำ U4 โดยเริ่มตั้งแต่ 0000 - 6000H การใส่ JUMPPER 1 ตัว ที่ตำแหน่งใด U4 จะมีตำแหน่ง ADDRESS เริ่มต้นที่ตำแหน่งนั้น และมีความยาว 8 kB

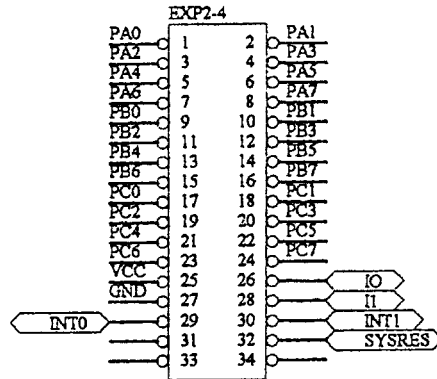
ตัวอย่างเช่น JP3 ใส่ JUMPPER ตรงตำแหน่ง 2000H เพียงตัวเดียวหมายถึง U4 มีตำแหน่ง ADDRESS ที่ 2000H - 3FFFH ดังนั้นการใส่ JUMPPER 4 ตัว ตั้งแต่ตำแหน่ง 0000 - 6000H ตำแหน่ง ADDRESS ของหน่วยความจำ U4 จะเริ่มที่ 0000H - 7FFFH

4. พอร์ตใช้งาน 8255

เป็นตัวขยายพอร์ตสำหรับการรับ/ส่งข้อมูลแบบขนานระหว่างอุปกรณ์ภายนอกกับไมโครคอนโทรลเลอร์ พอร์ตอินพุต/เอาต์พุต ทั้ง 3 ตัว โดยมีตำแหน่ง ADDRESS แต่ละตัวดังนี้

U6	PORTA	E080H
	PORTB	E081H
	PORTC	E082H
	CONTROL	E083H
U7	PORTA	E0A0H
	PORTB	E0A1H
	PORTC	E0A2H
	CONTROL	E0A3H
U8	PORTA	E0C0H
	PORTB	E0C1H
	PORTC	E0C2H
	CONTROL	E0C3H

พอร์ต 8255 แต่ละตัวจะมี CONECTOR



รูปที่ 27

แสดงตำแหน่ง I/O บน คอนเนคเตอร์

คอนเนคเตอร์ทั้ง 3 คิว มีการวางตำแหน่งต่างๆ เหมือนกัน

ตัวอย่างการเซตค่าเริ่มต้น ในการทำงานของ 8255

```
PORTAU6 EQU 0E0C0H ;PORTA 8255#U6
```

```
PORTBU6 EQU 0E0C1H ;PORTB 8255#U6
```

```
PORTCU6 EQU 0E0C2H ;PORTC 8255#U6
```

```
CONT_U6 EQU 0E0C3H ;PORTD 8255#U6
```

```
PORTAU8 EQU 0E0A0H ;PORTA 8255#U8
```

```
PORTBU8 EQU 0E0A1H ;PORTB 8255#U8
```

```
PORTCU8 EQU 0E0A2H ;PORTC 8255#U8
```

```
CONT_U8 EQU 0E0A3H ;PORTD 8255#U8
```

```
MOV DPTR,#CONT_U6
```

```
MOV A,#80H ;PA=0,PB=0,PC=0
```

```
MOVX @DPTR,A
```

```
LCALL DELAY
```

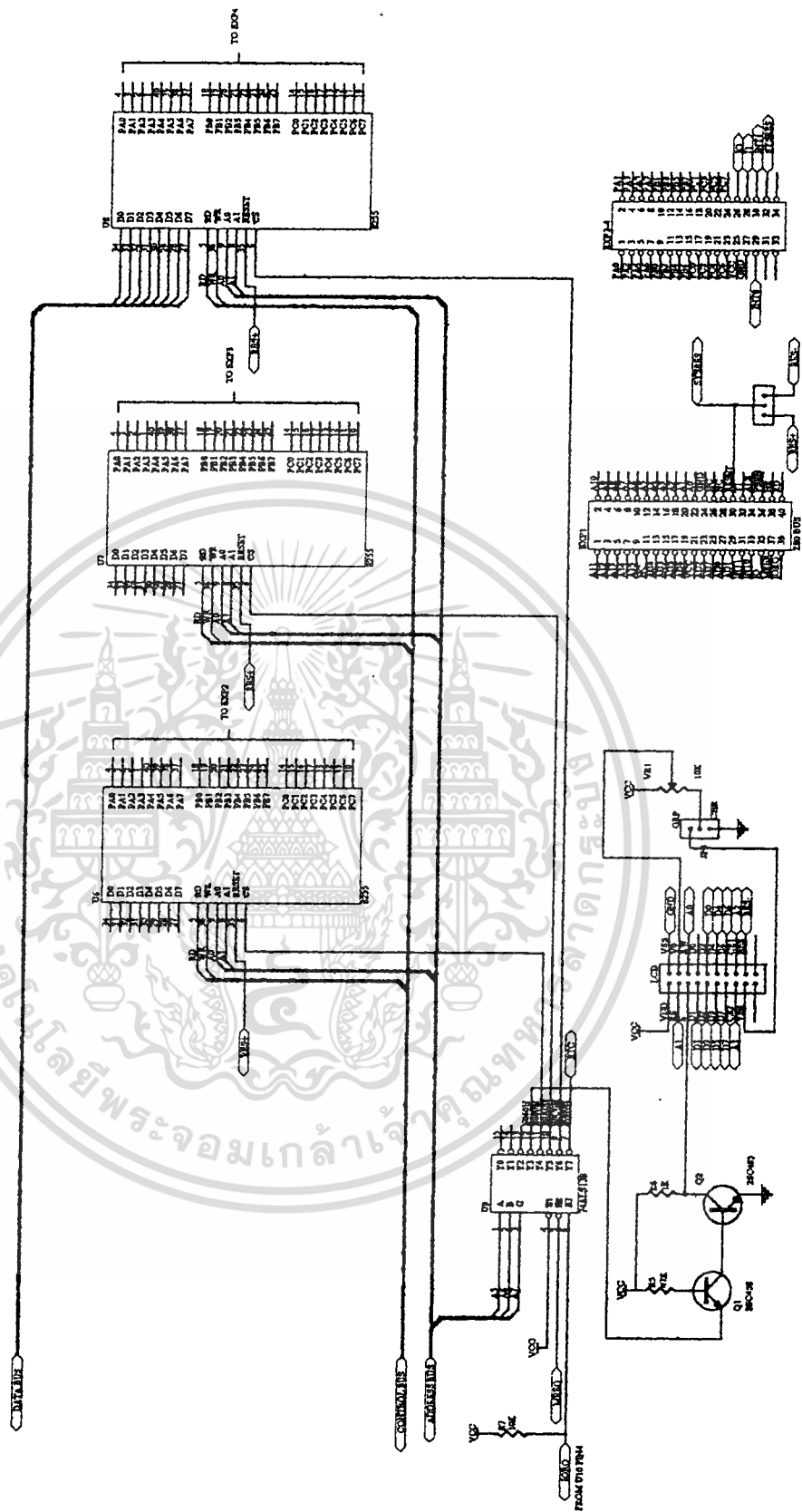
```
MOV DPTR,#CONT_U8
```

```
MOV A,#92H ;PA=1,PB=1,PCH=0,PCL=0
```

```
MOVX @DPTR,A
```

```
LCALL DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 28

แสดงการต่อวงจรของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเริ่ม SETUP ครั้งแรก

1. กำหนดให้ STOP และ RESET เท่ากับ 1
2. กำหนดการเดินเวลาแบบ 24 หรือ 12 ที่บิต 24/12
3. กำหนดเรื่องการ INTERRUPT ตามที่ CONTROL E
4. กำหนดให้ HOLE เท่ากับ 0
5. กำหนดให้ STOP และ RESET เท่ากับ 0 และพร้อมที่จะใช้งานได้อีกต่อไป

การอ่านหรือเขียนค่าเวลา

1. กำหนดให้ HOLD เท่ากับ 1 เพื่อค้างค่าเวลาเอาไว้
2. ตรวจสอบบิต BUSY โดยค่าเท่ากับ 1 หมายถึงยังไม่พร้อมที่จะอ่านหรือเขียนข้อมูลให้ กำหนด HOLD เท่ากับ 0 และกลับไปทำงานที่ข้อ 1 ใหม่ แต่ถ้า BUSY เท่ากับ 0 ให้ทำข้อต่อไปได้
3. ทำการอ่านหรือเขียนค่าเวลาได้ตามต้องการ โดยขบวนการให้ส่วนนี้ควรจะกระทำให้เสร็จเรียบร้อยภายในเวลา 1 วินาที
4. กำหนดให้ HOLD เท่ากับ 0 เป็นการสิ้นสุดขบวนการ

-ตัวอย่างโปรแกรมของการเซตค่า RTC MSN 6242B

```
RTCPOT EQU 0E0E0H ;RTC(M6242 OKI)
```

```
***** RTCINT SUB *****
```

```
;INITIAL 6242B OKI RTC
```

```
RTCINT:  MOV DPTR,#RTCPOT+0FH
        MOV A,#0000011B ;STOP=1,REST=1
        MOVX @DPTR,A
        ORL A,#00000100B ;24 HOUR
        MOVX @DPTR,A
        MOV DPTR,#RTCPOT+0EH
        MOV A,#00000100B ;SET STND 1 SEC
        MOVX @DPTR,A
        MOV DPTR,#RTCPOT+0DH
        MOV A,#00000000B ;HOLD = 0
        MOVX @DPTR,A
```

```
MOV DPTR,#RTCPOT+0FH
```

```

MOVX A,@DPTR
ANL A,#00000100B ;STOP=0,REST=0
MOVX @DPTR,A
RET

```

6. ส่วนการแสดงผล

- ตำแหน่งของพอร์ต LCD แบบ CHARACTER (พอร์ต E060H - E063H)
- BASE ADDRESS อยู่ที่ E060H และมีตำแหน่ง WRITE INSTRUCTION เขียนคำสั่งต่างๆ ให้กับ LCD MODULE เบอร์พอร์ตอยู่ที่ E060H
- READ BUSY FLAG & ADDRESS: อ่านสถานะและตำแหน่ง CURSOR ที่พอร์ต E061H
- WRITE DATA : เขียน CHARACTER ไปที่ LCD ตำแหน่งพอร์ตคือ E062H
- READ DATA : อ่านข้อมูลที่ตำแหน่ง CURSOR อยู่ตำแหน่งพอร์ตคือ E063H

โปรแกรม การกำหนดโหมดการทำงานและเซตค่าต่างๆของ LCD

```

COM_PORT EQU 0E060H ;READ-WRITE RES
RD_BUSY EQU 0E061H ;READ(BUSY FLAG)
DAT_PORT EQU 0E062H ;WRITE CHARACTER
RD_DATA EQU 0E063H ;READ DATA FROM DD RAM

MOV A,#00111000B ;FUNCTION SET
LCALL WR_COMM
LCALL DELAY
MOV A,#00001100B ;DISPLAY ON/OFF
LCALL WR_COMM
LCALL DELAY
MOV A,#01H ;CLEAR
LCALL WR_COMM
LCALL DELAY

```

```

;***** SUB WR_COMM *****

```

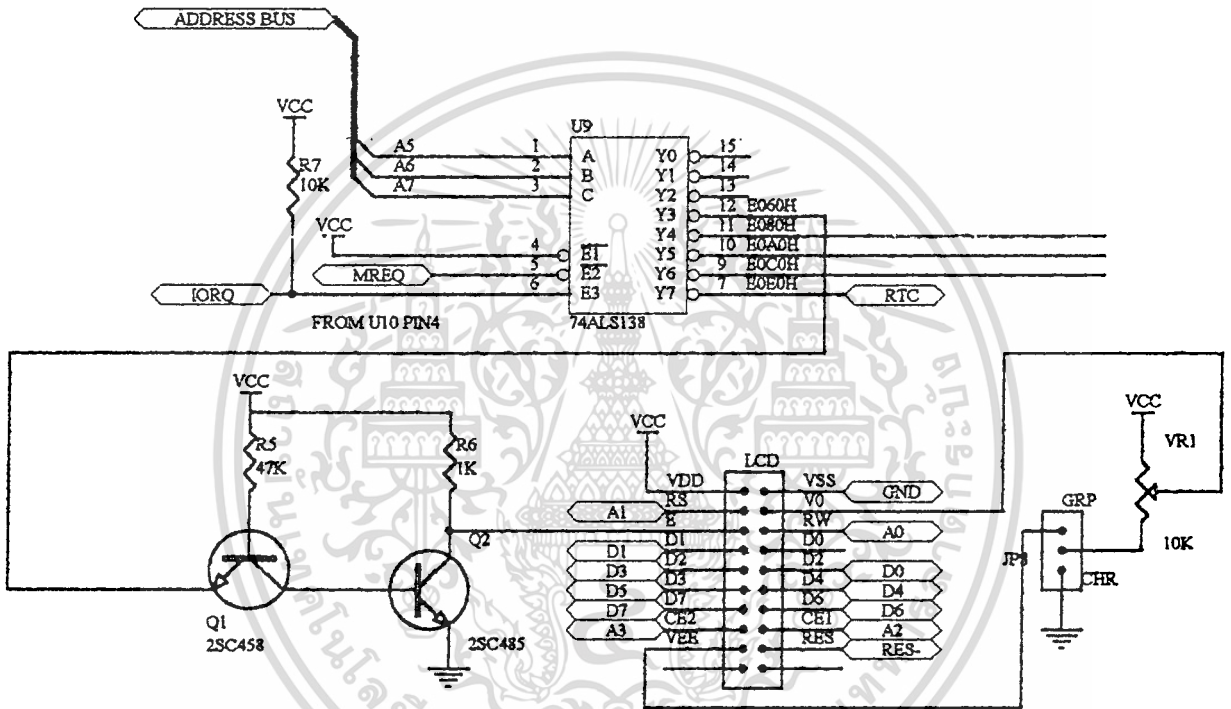
```

WR_COMM: PUSH DPH
        PUSH DPL
        MOV DPTR,#COM_PORT

```

```

MOVX @DPTR,A
MOV DPTR,#RD_BUSY
BUSYC: MOVX A,@DPTR
JB ACC.7,BUSYC
POP DPL
POP DPH
RET
    
```



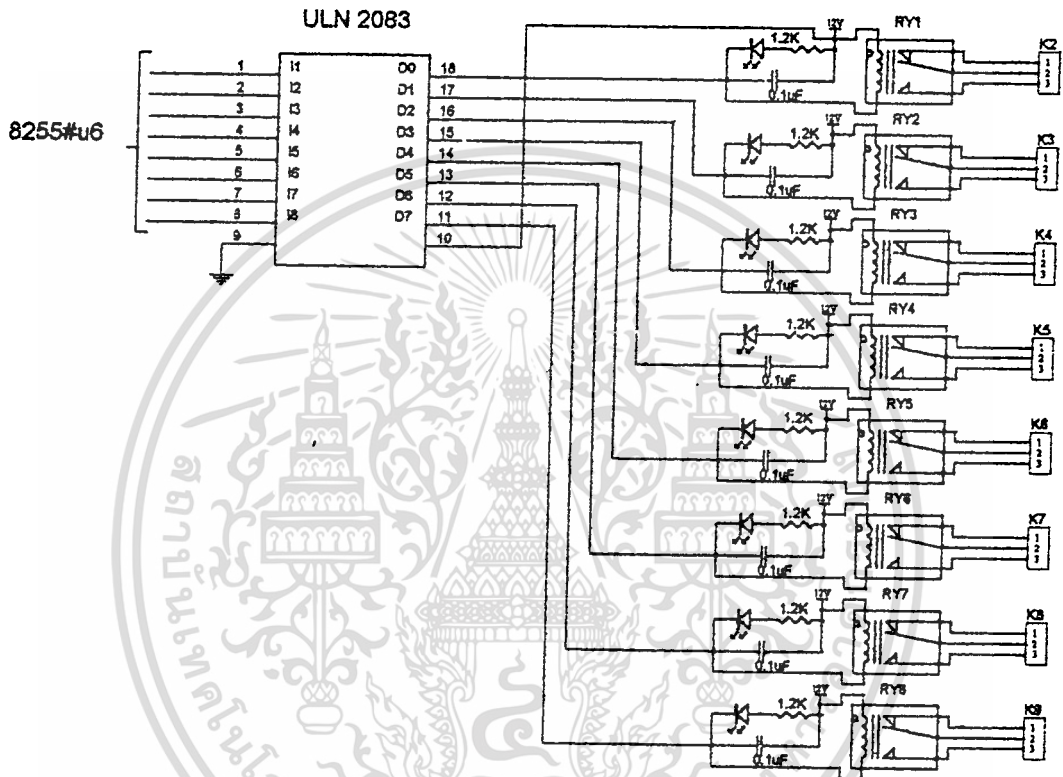
รูปที่ 30
แสดงการต่อ LCD

7. OUTPUT DRIVER

เป็นการต่อสัญญาณจาก IC 8255 จึงสามารถ LATCH ค่าได้นำมาขับ RELAY โดยต่อผ่าน IC ULN2803 ซึ่งมีเอาต์พุต 8 Channel โดยแต่ละเอาต์พุต จะรับกระแสได้ 500 mA 50 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณควบคุมจะส่งไปยังขา INPUT ของ ULN 2803 ซึ่งเบอร์ ULN 2803 ภายในประกอบด้วยชุด ทรานซิสเตอร์คาร์ริงตัน อาร์เรย์ 7 ชุด ที่มีตัวต้านทานค่ออนุกรมกับขาอินพุต รวมอยู่ภายในแล้วจึงทำให้คอใช้งานร่วมกับเอาต์พุต ของ TTL และ CMOS ได้โดยตรง



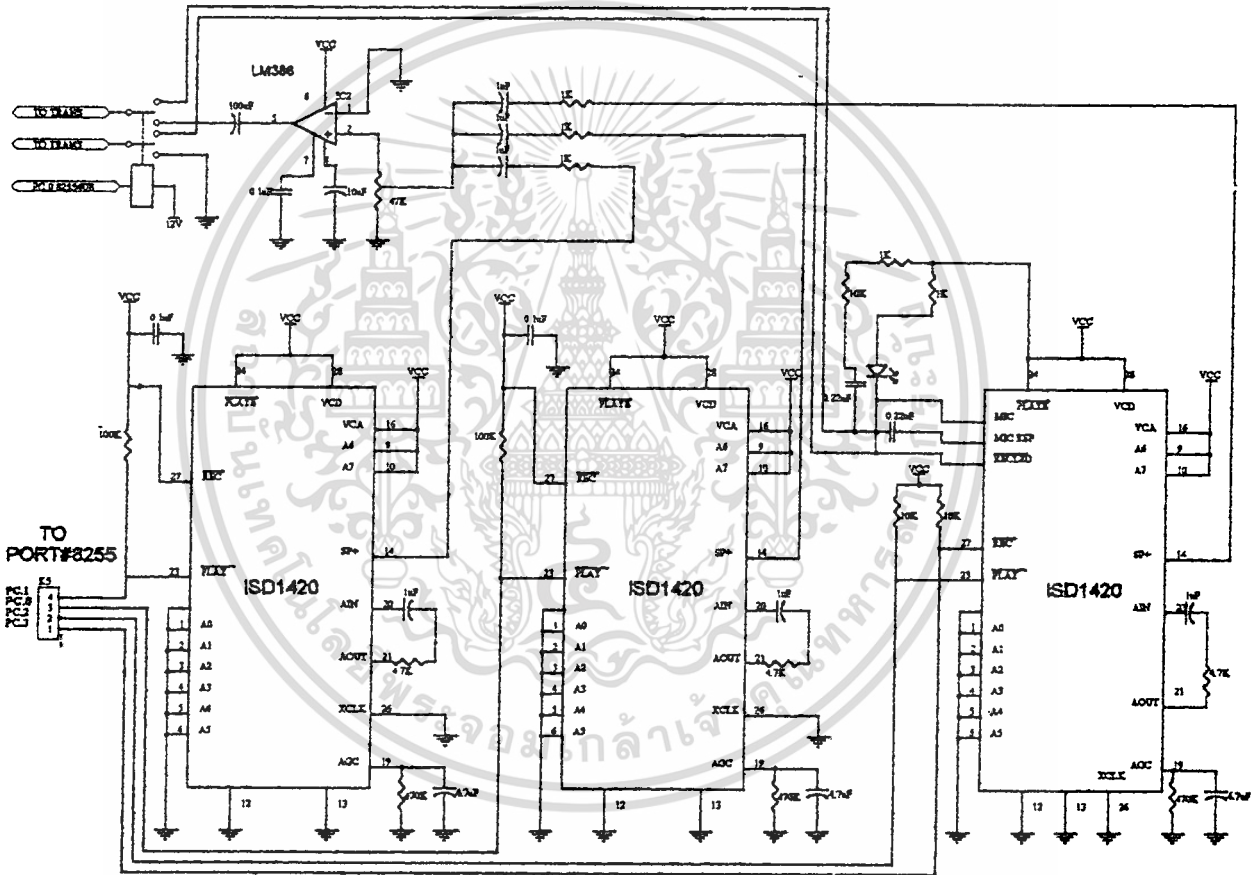
รูปที่ 31

แสดงภาคเอาต์พุต

จากวงจรเอาต์พุตของ ULN 2803 ทั้งหมดจะรีเลย์แต่ละ ตัวเฉพาะช่องของตนเองโดยในแต่ละช่องจะมี LED แสดงสถานะการทำงานของรีเลย์ นั้นมีหน้าที่สัมผัสรีเลย์จะเป็นแบบ 2 ทาง คือ ปกติปิด (NC) ,ปกติเปิด (NO) และขาคอมมอนด์ (COMMON)

8. การควบคุมเสียงและการอัดเสียงโดย ISD1420

ISD1420 จะมีหน่วยความจำขนาดความจุ 128,000 เซล การบันทึกไม่ได้ใช้หลักการทำงานแปลงสัญญาณจากสัญญาณอะนาลอกเป็นดิจิทัล (A/D) หรือจากดิจิทัลเป็นอะนาลอก (D/A) แต่อย่างไรก็ตามสามารถบันทึกสัญญาณอะนาลอกเข้าไปเก็บไว้ในหน่วยความจำได้โดยตรงจากรูปวงจรจะเห็นว่ามีการต่อ IC อยู่ 3 ตัวโดยนำสัญญาณทั้ง 3 ตัวมา MIX ต่อกันขยายสัญญาณเพิ่มโดยใช้ LM386 การควบคุมโดยต่อ CON4 เข้ากับไมโครคอนโทรลเลอร์



รูปที่ 32

แสดงการต่อ ISD1420 ร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น กรุณาอย่าให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. สรุปลินพุด/เอาต์พุด

PORT P1	FUNCTION
P1.0	- รับค่าDATA จาก DS1620
P1.1	- เป็นค่าสัญญาณนาฬิกาให้แก่ DS1620
P1.2	- เป็นการเริ่มทำงาน ของ IC
P1.3	- ขาที่ใส่เปรียบเทียบอุณหภูมิ
P1.4	- MT8870 จะส่งสัญญาณเมื่อมีการกดเบอร์
P1.5	- เป็นสัญญาณ RESET ไมโคร 4047 ทำงาน
P1.6	-
P1.7	- นับ Pulse ที่สำหรับรับสาย
8255 #U6	FUNCTION
PA.0 - PA.7	- เอาต์พุตขับ RELAY Channel 1-8
PB.0 - PB.7	- เอาต์พุตขับ RELAY Channel 9-16
PC.0	- PLAY เสียงที่ Channel B
PC.1	- PLAY เสียงที่ Channel C
PC.2	- PLAY เสียงที่ Channel A
PC.3	- REC. Channel A
8255 #U8	FUNCTION
PA.0 - PA.7	- อินพุตตรวจจับ SENSOR การทำงาน
PB.0-PB.3	- อินพุตที่ไต่จาก MT8870
PC.0	- การเล่นและการบันทึก
PC.1	- ขงหูโทรศัพท์
PC.2	- อัดเสียงเข้าเทป
PC.3	- คัด Tone ออกจากระบบเสียง
PC.7	- ขงหูโทรศัพท์ ที่ Board เสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ระบบควบคุมอัตโนมัติ ในช่วงที่ยังไม่มีการโทรศัพท์เข้ามา การทำงานของโปรแกรมจะทำการ Scan ค่าฐานเวลาโดยแสดงผลที่จอ LCD เป็นค่า วินาที นาที ชั่วโมง วัน เดือน ปี และจะตรวจสอบเวลาที่ตั้งไว้

หลังจากที่ได้รับสัญญาณการเรียกสายโทรศัพท์ โดยส่วนตรวจจับสัญญาณจนถึงการแปลงค่าข้อมูลออกเป็นค่าดิจิทัล ซึ่งนำค่าเหล่านั้นมาทำการสั่งงานให้ MCS-51 รับทราบ เป็นดังต่อไปนี้

1. ส่วนตรวจจับสัญญาณกระดิ่ง ซึ่งต่อเข้ากับขาพอร์ตที่ 1 บิทที่ 7 (P1.7) ลักษณะของสัญญาณทำการตรวจสอบ เมื่อผ่านการแปลงแล้วจนเข้าสู่ขา P1.7 จะมีลักษณะดังนี้
 - เมื่อมีสัญญาณกระดิ่งเข้ามาค่าเป็น High หรือ Logic "1"
 - เมื่อมีสัญญาณกระดิ่งหายไปค่าเป็น Low หรือ Logic "0"
2. การยกหูโทรศัพท์ เราสามารถตั้งโปรแกรมได้ ควรมีการรอสายนานเท่าไร ในที่นี้เราตั้งไว้ 6 ครั้ง เป็นการนับพัลส์ที่ขา P1.7 ว่ามีการตกเป็นค่า Low ครบ 6 ครั้งหรือเปล่า เมื่อตรวจสอบนับครบตามพัลส์ที่ตั้งไว้ MCS-51 จะส่ง Logic "1" มาไบแอสขาเบสของทรานซิสเตอร์ไปขับ Relay ให้ทำการยกหูโทรศัพท์ จึงทำให้สัญญาณจากคู่สายโทรศัพท์ต่อกับระบบ และระบบจะรับค่าข้อมูลเข้ามาทำการควบคุมอีกทีหนึ่ง
3. สำหรับการถอดรหัส DTMF ในจังหวะที่คู่สายโทรศัพท์ต่อเข้ากับชุดถอดรหัส DTMF ซึ่งเป็นอุปกรณ์ถอดรหัสสัญญาณโทนออกมาเป็นรหัส BCD สัญญาณทางอินพุตการถอดรหัส ออกมาทางเอาต์พุตจะเป็นรหัส BCD ทาง Q1-Q4 (ขา 11 - ขา 14) การถอดสัญญาณโทน หรือเมื่อมีการกดหมายเลขแต่ละครั้ง ไอซี MV8870 จะส่งสัญญาณออกมาที่ขา STD(ขา 15) หนึ่งพัลส์ มีลักษณะ Logic "1"
4. ส่วนการบันทึกและตอบรับโทรศัพท์ ซึ่งเป็นโหมดการควบคุมที่แยกอยู่ต่างหาก เรากำหนดเวลาในการบันทึกไว้ที่ 50 วินาที ต่อการโทร 1 ครั้ง เมื่อมีการโทรเข้ามาจะมีการพูดแนะนำการใช้งานโดยหลักการคือให้ตัดไปที่หมายเลข 9 เพื่อบันทึกข้อความ
5. โหมดการควบคุมอุปกรณ์ไฟฟ้า และตั้งเวลาการทำงาน เมื่อเข้าโหมดการทำงานนี้แล้ว ก็จะเป็นการสั่งงาน หรือตั้งค่าต่าง ๆ ให้กับระบบ ถ้าไม่มีการกดคีย์ หรือหมายเลขใด ๆ เป็นเวลานานกว่า 10 วินาที เครื่องควบคุมจะทำการวางหูโทรศัพท์ ดังนั้นการสั่งงานแต่ละครั้งจะต้องสั่งงานภายในเวลา 10 วินาที

สำหรับการตั้งเวลาทำงานของ Channel ต่าง ๆ นั้นโดยการกดหมายเลข Channel แล้วตามด้วย 3 เราสามารถตั้งเวลาเปิดปิดอุปกรณ์ต่าง ๆ ได้ ตลอด 24 ชั่วโมง ซึ่งค่าจะถูกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บไว้ใน NVSRAM ตลอดจนกว่าจะทำการเปลี่ยนค่าใหม่เราสามารถทำการยกเลิกการตั้งเวลาต่าง ๆ ได้โดยการกด # ตลอด

6. การวัดอุณหภูมิ ไอซีเบอร์ DS1620 เป็นการสื่อสารข้อมูลแบบอนุกรมในการอ่านหรือเขียนแต่ละครั้งของโปรแกรม จะต้องส่งค่า Config ของการรับส่งข้อมูล เช่น

- การอ่านค่าอุณหภูมิต่อ Config คือ AAh
- การเขียนค่าอุณหภูมิต่อ TH Config คือ 01h

การใช้งานและการควบคุม

สามารถแบ่งเป็นลำดับขั้นตอนได้ดังนี้

1. สภาวะปกติ ที่จอ LCD จะแสดงเวลาปัจจุบัน และ วัน เดือน ปี

Time is [22:01:32]

Date is [20:10:96]

ใช้ช่วงเวลาที่ไม่มีการโทรศัพท์เข้าหรือโทรออก

- เมื่อมีการใช้งานโดยมีสัญญาณกระดิ่งเข้ามาที่คู่สาย จะมีการรอเวลาขงู (สามารถตั้งโปรแกรมในระยะเวลาได้)
- เครื่องจะมีเสียงโทรตอบรับช่วงสั้นๆ 2 ครั้ง

2. หลังจากนั้นจะมีเสียงพูด เพื่อให้ทำการเลือกโหมดการทำงาน (แยกเป็น 2 กรณี)

2.1. กรณีการบันทึกฝากข้อความไว้ จะมีช่วงเวลาการบันทึก

2.2. กดรหัสผ่านเพื่อเข้าไปควบคุมอุปกรณ์ไฟฟ้า

- ทำการกดรหัสผ่านเพื่อเป็นเงื่อนไขเพื่อให้ไมโครคอนโทรลเลอร์ทำงาน โดยรหัสเริ่มต้นครั้งแรกคือ xxxxxx (สามารถตั้งรหัสผ่านใหม่ได้)
- กดหมายเลขรหัสให้ถูกต้องภายใน 10 วินาที เมื่อกดถูกต้องจะมีเสียงตอบกลับเช่นเดิมช่วงสั้นๆ 2 ครั้ง
- การตั้งรหัสผ่าน เพื่อป้องกันการสั่งงานอุปกรณ์ควบคุมจากบุคคลภายนอก เพื่อความปลอดภัยของอุปกรณ์ควบคุมและอุปกรณ์ที่ถูกควบคุม และเพื่อจำกัดสิทธิให้ใช้งานได้เฉพาะบุคคลที่ได้รับอนุญาตเท่านั้น

3. การเปลี่ยนรหัสใหม่ในการควบคุมอุปกรณ์ไฟฟ้า

3.1 การตั้งรหัสเดิม

- กด * ตามด้วยหมายเลข 1 เป็นการเข้าสู่โปรแกรมการตั้งรหัสผ่านจะมีเสียงสัญญาณตอบกลับสั้นๆ 2 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตามด้วยเครื่องหมาย # จะมีเสียงเตือนออกมา 1 ครั้งรหัสเดิมถูกลบออกเรียบร้อยแล้ว

3.2 การตั้งรหัสผ่านใหม่

- จำนวนหลักของรหัสสามารถกำหนดได้ 8 หลักและทุกครั้งที่มีการโปรแกรมเข้าไปหนึ่งหลักจะมีเสียงตอบกลับแบบสั้นๆ 1 ครั้ง
- สามารถกำหนดตัวเลขรหัสในแต่ละหลักได้ตั้งแต่เลข 0 ถึงเลข 9
- กรณีที่ไม่ต้องการตั้งหมายเลขรหัสให้ครบทั้งหมด 6 ตัวสามารถกระทำได้โดยกดตั้งรหัสไปตามที่ต้องการจนครบตามต้องการ หลังจากกดสุดท้ายให้กดเครื่องหมาย # ค่อยๆไปจนครบ 6 หลัก

4. ขั้นตอนต่อมา เป็นการเลือกโหมดการทำงานซึ่งแยกออกเป็นดังนี้

- 4.1 การตั้งงานควบคุมอุปกรณ์ไฟฟ้า Channel 1-8 (สามารถตั้งเวลาเปิด ปิดได้)
- 4.2 การตั้งงานควบคุมอุปกรณ์ไฟฟ้า Channel 9-16 (ไม่สามารถตั้งเวลาเปิด ปิดได้)
- 4.3 ตั้งระบบต่างๆ

5. การควบคุมอุปกรณ์ไฟฟ้า Channel 1-8

- 5.1 คีย์สวิตช์โทรศัพท์หมายเลข 1-8 เป็นหมายเลขแทนช่องสัญญาณที่ต้องการควบคุม

5.2 เมื่อตามด้วย

หมายเลข 1 หมายถึง เปิด (ON)

หมายเลข 0 หมายถึง ปิด (OFF)

หมายเลข 2 แสดงสถานะของช่วงสัญญาณนั้นๆ

หมายเลข 3 ตั้งเวลา เปิด หรือปิด แต่ละ Channel ช่องสัญญาณ

ตัวอย่างเช่น เมื่อต้องการควบคุม Channel 1 ทำการเปิดอุปกรณ์ไฟฟ้า เมื่อครบรหัสผ่านมาแล้ว กดหมายเลขดังต่อไปนี้

0 1 1

0 1 (2ตัวแรก) คือ Channel 01

1 (ตัวหลัง) คือ ON

ถ้าต้องการให้ Channel 1 ทำการปิดอุปกรณ์ไฟฟ้าให้กดหมายเลข

0 1 0

0 1 (2ตัวแรก) คือ Channel 01

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ (ตัวหลัง) คือ OFF เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการตรวจสอบสถานะการทำงานของ Channel(1-8) ทำการกดหมายเลข Channel (2 ตัว) แล้วตามด้วย เลข 2 คือการตรวจสอบสถานะการทำงาน หลังจากนั้นจะมีเสียงตอบกลับเป็นการบอกสถานะ

- ทุกครั้งที่ควบคุมให้ เปิด จะมีสัญญาณเสียงตอบกลับช่วงสั้นๆ 2 ครั้งเสมอ
- ทุกครั้งที่ควบคุมให้ ปิด จะมีสัญญาณเสียงตอบกลับช่วงยาว 1 ครั้งเสมอ

ตัวอย่าง เช่น ตรวจสอบว่า Channel 2 มีสถานะเปิดหรือปิด

0 2 (2ตัวแรก) คือ Channel 02

2 (ตัวหลัง) คือ ตรวจสอบสถานะของ Channel ที่(2ตัวแรก) คือ 02

เครื่องจะแสดงสถานะโดยส่งสัญญาณเสียงตอบกลับ

สำหรับการตั้งเวลาเปิด ปิด ทำการกดหมายเลข Channel (2 ตัว) แล้วตามด้วย เลข 3 การตั้งเวลาที่ดีเราควรตั้งเวลาภายใน (ตั้งที่เครื่อง ที่เครื่องจะมี LCD บอก) ซึ่งเราสามารถมองเห็นค่าที่เราตั้งใน LCD เช่น

CH 07 ON [09:30]

TIME OFF [10:45]

Channel 07 ทำการเปิดอุปกรณ์ เวลา 09.30 น.

ทำการปิดอุปกรณ์ เวลา 10.45 น.

- ตำแหน่ง Cursor จะเป็นตัวบ่งบอกให้เราทราบตำแหน่งหลักของค่าเวลา
- เมื่อไม่ต้องการตั้งเวลาใหม่ให้กดหมายเลขโทรศัพท์ 2 ครั้ง ก็จะหลุดออกจากโปรแกรมการตั้งเวลา
- ถ้าต้องการให้เปิด หรือ ปิดอย่างเดียว ก็สามารถทำได้โดยกดหมายเลขโทรศัพท์ ลงในตำแหน่งที่ไม่ใช่ เช่น

CH 07 ON [09:30]

TIME OFF [**:**]

Channel 07 ทำการเปิดอุปกรณ์ เวลา 09.30 น.

อย่างเคียว ไม่มีเวลาปิด

6. การควบคุมอุปกรณ์ไฟฟ้า Channel 9-16

6.1 คีย์สวิตช์โทรศัพท์หมายเลข 9-16 เป็นหมายเลขแทนช่องสัญญาณที่ต้องการควบคุม (ไม่สามารถตั้งเวลาได้)

6.2 เมื่อตามด้วย

หมายเลข 1 หมายถึง เปิด (ON)

หมายเลข 0 หมายถึง ปิด (OFF)

เอกสารนี้เป็นเอกสารที่สหมายเลข 2 แสดงสถานะของช่วงสัญญาณนั้นๆ กรุณาอย่าให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น เมื่อต้องการควบคุม Channel 15 ทำการเปิดอุปกรณ์ไฟฟ้า เมื่อครบหัทศผ่านมาแล้ว กดหมายเลขดังต่อไปนี้

- ทำการเปิดอุปกรณ์ไฟฟ้า Channel 15
 ทำการปิดอุปกรณ์ไฟฟ้า Channel 15
 ตรวจสอบสถานะการเปิดปิดอุปกรณ์ไฟฟ้า Channel 15

7. การเข้าโหมด Monitor หรือการเซตค่าต่างๆ โดยการกดหมายเลข แล้วตามด้วยหมายเลขต่อไปนี้

- หมายเลข การตั้งรหัสผ่าน
 หมายเลข การตั้งเวลารอกหูโทรศัพท์
 หมายเลข การทดสอบสัญญาณ DTMF
 หมายเลข การตั้ง ความยาวของเสียงในการตอบกลับ
 หมายเลข การตั้งเวลาปัจจุบัน (ชม.: นาที : วินาที)
 หมายเลข การตั้งเวลาปัจจุบัน (วัน : เดือน : ปี)
 หมายเลข การตั้งอุณหภูมิเปรียบเทียบ
 หมายเลข การอ่านค่าอุณหภูมิ
 หมายเลข ทดสอบ IC ISD 1420 (Play)
 หมายเลข การทดสอบการบันทึกเสียงของ ไอซี ISD1420 และการบันทึกเสียง(ฝาก

ข้อความ)

7.1 การตั้งรหัสใหม่

ในการเข้าโหมดการทำงานการตั้งรหัสใหม่ กดหมายเลข

- จำนวนหลักรหัสผ่านสามารถกำหนดได้ 8 หลักและทุกครั้งที่มีการป้อนค่า 1 หลักจะมีเสียงตอบกลับสั้นๆ 1 ครั้ง
- เราสามารถกดหมายเลขใดก็ได้ตั้งแต่ 0-9
- กรณีที่ไม่ต้องการจำนวนรหัสครบ 8 ตัว ก็สามารถทำได้โดยตั้งรหัสไป ตามต้องการแล้วหลังจากนั้น ให้กดตามด้วยหมายเลข ก็เป็นการสิ้นสุด

การตั้งรหัสใหม่

7.2 การกำหนดจำนวนสัญญาณเป็นการกำหนดเวลาในการตอบกลับ หรือเวลาในการรอ

เอกรับสายของเครื่องควบคุม สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กด * ตามด้วยหมายเลข 2 จะมีสัญญาณตอบกลับสั้นๆ 2 ครั้ง
- กดหมายเลข 0, 1 หรือ 2 ในการตั้งเวลาหลักสิบ
- เสร็จแล้วให้กดเลขหมาย 0-9 เพื่อตั้งหลักหน่วยต่อไป
- เมื่อกดเลขหมายครบแล้วจะมีเสียงสัญญาณดังยาว 1 ครั้งตอบกลับมา

ตัวอย่าง หากกด * และเลข 2
แล้วตามด้วยเลข 0 กับ เลข 6

แสดงว่าเป็นการตั้งเวลาการตรวจจับสัญญาณกระดิ่งที่เข้ามาทางคู่สายไว้ 6 พัลส์ (หรือประมาณ 5-6 วินาที)

7.3 การทดสอบสัญญาณ DTMF

- กดคีย์แต่ละคีย์จะต้องให้สัญญาณโทนได้อย่างถูกต้อง
- เริ่มจากกด * ตามด้วย 3
- ตามด้วยหมายเลขช่อง
- จะมีเสียงดังออกมาเท่ากับจำนวนของหมายเลขคีย์

7.4 การกำหนดเสียงสั้นหรือเสียงยาว

- การกำหนดความยาวเสียงตอบรับควรจะกำหนดให้เสียงตอบรับอยู่ในช่วงสั้นๆ (0 วินาที-3 วินาที) และทำตามขั้นตอนดังนี้
- กด * แล้วตามด้วย 4
- ตามด้วย 0, 1, 2 หรือ 3

7.5 การตั้งเวลาปัจจุบัน

- เป็นสิ่งที่จำเป็นมากที่ต้องการตั้งเวลาปัจจุบันเนื่องจากในโปรแกรมมีการนำ ค่าเวลามาเปรียบเทียบใช้ในการเปิด ปิด เอาดัตชุด

- เริ่มต้นจากการรหัสผ่านเข้ามาแล้ว กด * ตามด้วย 5 ก็เข้าสู่โหมดการตั้งเวลา จะ

มี จอ LCD แสดงค่าเวลาให้ดู

Time is [08:00:32]

Newtime [: :]

- บนจอ LCD จะแสดงค่าเวลาเดิมที่ตั้งไว้ใน บรรทัดบนของ LCD
- บรรทัดล่างของ LCD จะเป็นช่องที่ว่าง เราจะป้อนค่าเวลาใหม่ ตามตำแหน่ง CURSOR กระพริบ
- เมื่อป้อนค่าในหลัก ๆ หนึ่ง CURSOR จะเลื่อนไปรอรับค่าหลักถัดไปจนครบทั้งหมด 6 หลัก ให้กดเครื่องหมาย # เพื่อบันทึกค่าที่ตั้งไว้ในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.6 การตั้งวัน เดือน ปี

- ลักษณะการป้อนค่าวัน เดือน ปี คล้ายกับการตั้งค่าเวลาข้างต้น โดยมีมาตรฐาน [วัน / เดือน / ปี]

Date is [21/10/96]

Newdate [: :]

- เช่นเดียวกับบรรทัดบนทั้งหมดเป็นการแสดงค่า ปัจจุบันซึ่งบรรทัดล่างเป็นส่วนที่ว่าจะทำการป้อนค่าใหม่เข้าไปในหน่วยความจำต่อไป
- เมื่อเข้าสู่โหมดการตั้งเวลาหรือตั้ง วัน เดือน ปี ไม่ต้องการตั้งเวลาใหม่ ก็กระทำได้โดยการกด # จำนวนเจ็ดครั้ง(ยังไม่ได้ทำการป้อนค่าใหม่)

7.6 การตั้งอุณหภูมิเปรียบเทียบ

- เป็นการตั้งค่าให้กับ DS1620 ซึ่งเป็นไอซีวัดอุณหภูมิโดยการป้อนค่าอุณหภูมิ ที่เปรียบเทียบให้
- เมื่อเข้าสู่โหมดการทำงานการกดหมายเลข * ตามด้วย 7 บนจอจะแสดง ข้อให้เลือกโหมดการทำงานแต่ละข้อ

1. READ TH,TL 2. INTH
3. INTL # ENTER

จะมีตัวเลือกให้เลือกอยู่ 4 ตัวเลือกคือ

- เมื่อกด 1 คือการอ่านค่าอุณหภูมิเปรียบเทียบที่สูง และอุณหภูมิเปรียบเทียบที่ต่ำ

T-HIGH + 30.5°C
T-LOW + 18.5°C

- หรือเมื่อกด 2 คือการเลือกตั้งค่าอุณหภูมิเปรียบเทียบที่สูงเพื่อนำไปคำนวณในการตัดสินใจ บรรทัดบนคือค่า อุณหภูมิ ที่ตั้งไว้ครั้งก่อนโดยสามารถตั้งใหม่ได้ในบรรทัดที่ 2

T-HIGH + 30.5°C
New T-HIGH +°C

- หรือเมื่อกด 3 คือการเลือกตั้งค่าอุณหภูมิเปรียบเทียบที่ต่ำ เพื่อ นำไปคำนวณในการตัดสินใจ บรรทัดบนคือค่า อุณหภูมิ ที่ตั้งไว้ครั้งก่อนโดยสามารถตั้งใหม่ได้ใน

เอกสารนี้เป็นบรรทัดที่ 2 วนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

T-LOW + 18.5°C

New T-LOW +°C

- เมื่อทำโปรแกรมหรือป้อนค่าต่างๆ ทั้งหมดแล้ว กด # เพื่อออกจากการทำงานในโหมดนี้

7.8 การอ่านค่าการวัดอุณหภูมิ

- เมื่อเข้าสู่โหมดการทำงานโดยกด # ตามด้วย 8 จะแสดงผลอุณหภูมิที่อัปเดต

ทาง LCD

TEMPERATURE

+ 28.5°C

7.9 ทดสอบ IC ISD1420 (PLAY)

- เมื่อกด * ตามด้วย 9 จะเข้าสู่โหมดทดสอบการเล่นของไอซีอัดเสียงซึ่งมีอยู่ 3 ชุด โดยแต่ละชุดจะมีให้เลือก การเล่น

1. PLAY PA 2. PLAY PB

3. PLAY PC # ENTER

-เมื่อกด * 9 1 เป็นการทดสอบเสียงไอซีตัวที่ 1 (PA)

-เมื่อกด * 9 2 เป็นการทดสอบเสียงไอซีตัวที่ 1 (PB)

-เมื่อกด * 9 3 เป็นการทดสอบเสียงไอซีตัวที่ 1 (PC)

-เมื่อกด # เป็นการออกจากโหมดการทำงาน

บทสรุปและบทวิจารณ์

จากระยะเวลาที่ศึกษาและลงมือทำโครงการนี้มาทั้งหมด ได้ศึกษาการใช้งานของไอซีเบอร์ DS1620 ซึ่งเป็นไอซีวัดอุณหภูมิและอ่านเขียนค่าข้อมูลแบบอนุกรมได้ จึงใช้เวลาศึกษาพอสมควรและได้ผลเป็นที่น่าพอใจ ในส่วนอื่นๆ ที่ต้องสนใจคือ การใช้งานส่วนแสดงผล LCD และส่วนฐานเวลาซึ่งเราต้องสนใจเรื่องการกำหนด Config และ Scan Time เป็นอย่างมาก การพัฒนาโปรแกรมทั้งหมดเราใช้ EPROM EMULATOR เป็นตัวพัฒนาโปรแกรม และใช้ NVSRAM เป็นตัวเก็บข้อมูล

จากการศึกษาวงจรโปรแกรมการควบคุมส่วนต่าง ๆ ต่อจากนั้นเราจะต้องนำเอาทุกส่วนที่แยกศึกษามาต่อเพื่อทำงานร่วมกัน ซึ่งเป็นส่วนที่ต้องคำนึงเรื่องแหล่งจ่ายและความเข้ากันได้ มีปัญหาอะไรเกิดขึ้นบ้าง ในการศึกษาแยกแต่ละส่วนๆ ก่อนทำให้ทราบว่าจะต้องแก้จุดไหนมาก่อนและสามารถแบ่งงานให้ร่วมงานแต่ละคนใช้ร่วมกันทุกงานในโครงการนี้ ได้แบ่งแยกดังนี้

- Software(โปรแกรม)
- Hardware
- ถายวงจร
- เอกสารต่าง ๆ ฯลฯ

เมื่อส่วนของโปรแกรมจะทำความรู้เกี่ยวกับ Hardware คือ เราจะตัดแปลงในส่วน Hardware เป็นคีย์ในกรณีที่ Software เขียนได้ลำบาก เมื่อการพัฒนาโปรแกรมสมบูรณ์แล้วจึงนำโปรแกรมไปอัดใส่ EPROM โดยใช้ เครื่องอัด EPROM "SPEP PLUS"

ปัญหาและแนวทางแก้ไข ในการทำงานมีดังนี้

1. การต่อพอร์ตของ LCD ผิดขั้ว เนื่องจากมาตรฐานการเรียงขาของ BUS LCD ไม่เหมือนกับ Connectorทั่วไป จึงมีการสลับกันบ้าง และบางครั้ง Scan Time ของการรันโปรแกรม LCD มีปัญหา จึงทำให้โปรแกรมการทำงานทั้งหมดผิดพลาด เนื่องจากต้องรอให้การทำงานของ LCD ผ่านก่อนการแก้ปัญหาคือถอดส่วนโปรแกรมย่อยที่ต้องมาก่อนกระโดดไปทำงานให้น้อยลง ส่วนด้าน Connector ก็ทำเครื่องหมายไว้ป้องกันมิให้ทำการเสียบขั้วผิด
2. การเซตค่า 8255 ไม่ทันเนื่องจากไม่ได้ทำการหน่วงเวลาเปิดเครื่องจึงมีผลทำให้ 8255 รีเซตตัวเองไม่ทันได้ สามารถแก้ไขโดยหน่วงเวลา
3. การใส่ Battery Backup ฐานเวลาและ RAM ที่เก็บข้อมูลจะปัญหามีอยู่ที่คือ Battery ที่ใช้ Backup จะหมดเร็วทำให้ค่าที่ตั้งไว้หาย หรือผิดเพี้ยนไป จึงแก้ปัญหาโดยการใช้ NVSRAM มาแทน RAM ชرمตนาและตัดการ Backup จาก Battery ออกไปโดยจ่ายไฟให้

กับฐานเวลา MSN6242 อย่างเดียวซึ่งกระแสที่จ่ายออกไปน้อยลง ทำให้สามารถใช้งาน Battery ได้นานขึ้น

4. ทางด้านโปรแกรมเนื่องจากหน่วยความจำ Internal ของ 8051 ถูกใช้ไปอย่างมาก มีการโคคของโปรแกรม และเรียกโปรแกรมย่อยบ่อยมาก จึงมีปัญหาเรื่อง Stack Pointer โดยปกติจะเก็บไว้ที่หน่วยความจำ Internal ที่ 08H เป็นต้นไป การแก้ปัญหาคือเราเปลี่ยน CPU จากเบอร์ 80C31 เป็นเบอร์ 80C32 และเก็บ Stack Pointer ที่ตำแหน่งที่ไม่มีการใช้งาน คือที่ ตำแหน่ง 80H เป็นต้นไป เนื่องจาก 80C32 มีหน่วยความจำเพิ่มขึ้นอีก 125 Byte คือตั้งแต่ 80H ถึง FFH โดยเราต้องสั่งเก็บ ณ. หน่วยความจำเหล่านี้โดยอ้อมก็จะสามารถเข้าไปใช้งานได้ การเซต Stack Point ได้ดังนี้

```
MOV SP,#80H
```

5. แนวทางพัฒนาต่อไป จะมีการออกแบบวงจรและสายวงจร เราสามารถลดขนาดให้ เล็กลงได้โดยการทำ สายวงจรเป็นปริ้นท์ 2 หน้า ส่วน CPU หรือหน่วยความจำก็เปลี่ยนไปใช้ ชิปที่สามารถโปรแกรมได้ในตัว เช่นชิปเบอร์ 89C52 แทน ทางด้านของ MCS-51 ซึ่งมี Serial Port เราสามารถนำไปประยุกต์ติดต่อกับคอมพิวเตอร์ หรือต่อ Serial Port เครื่องควบคุมอีกตัวหนึ่งโดยการนำการสื่อสารแบบ RS485 มาใช้ซึ่งจะส่งได้ไกลมาก การวัดอุณหภูมิก็แยก หลายๆ ตัว โดยนำไปติดตั้งไว้วัดอุณหภูมิแต่ละห้อง ๆ แล้วนำค่าข้อมูลส่งมาให้ตัว MASTER

เอกสารอ้างอิง

- 1.SEMICONDUCTOR DATA BOOK HD74LS SERIES ,HITACHI
- 2.คู่มือ เทียบเบอร์ไอซี TTL , บริษัท ซีอีคยูเคชั่นจำกัด ,พ.ศ. 2534
- 3.ปรเมษฐ์ ประชนานันท์ บิชพอส์ เฒ่าวิช “คู่มือ และการประยุกต์ใช้งาน” ไมโครคอนโทรเลอร์ Mcs-51 บริษัท ซีอีคยูเคชั่น จำกัด , พ.ศ. 2536
- 4.DALLAS SEMICONDUCTOR , SILA RESEARCH CO,LTD,2539
- 5.สุทธินันท์ พรศิริกุล “ลึกอีกนิดกับโทรศัพท์” เซมิคอนดักเตอร์อิเล็กทรอนิกส์
- 6.SIGNATICS MICROCONTROLLER USER’S GUIDE,PHILIPS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ บิดามารดาที่ให้ความรักความห่วงใยและสนับสนุน

ขอขอบคุณ อาจารย์ที่ปรึกษาที่ให้คำแนะนำ

ขอขอบคุณ เพื่อน ๆ ในห้องทุกคนที่คอยเป็นทั้งที่ปรึกษาและให้คำที่
ปรึกษายาม มีปัญหาและ หยิบยื่นอุปกรณ์



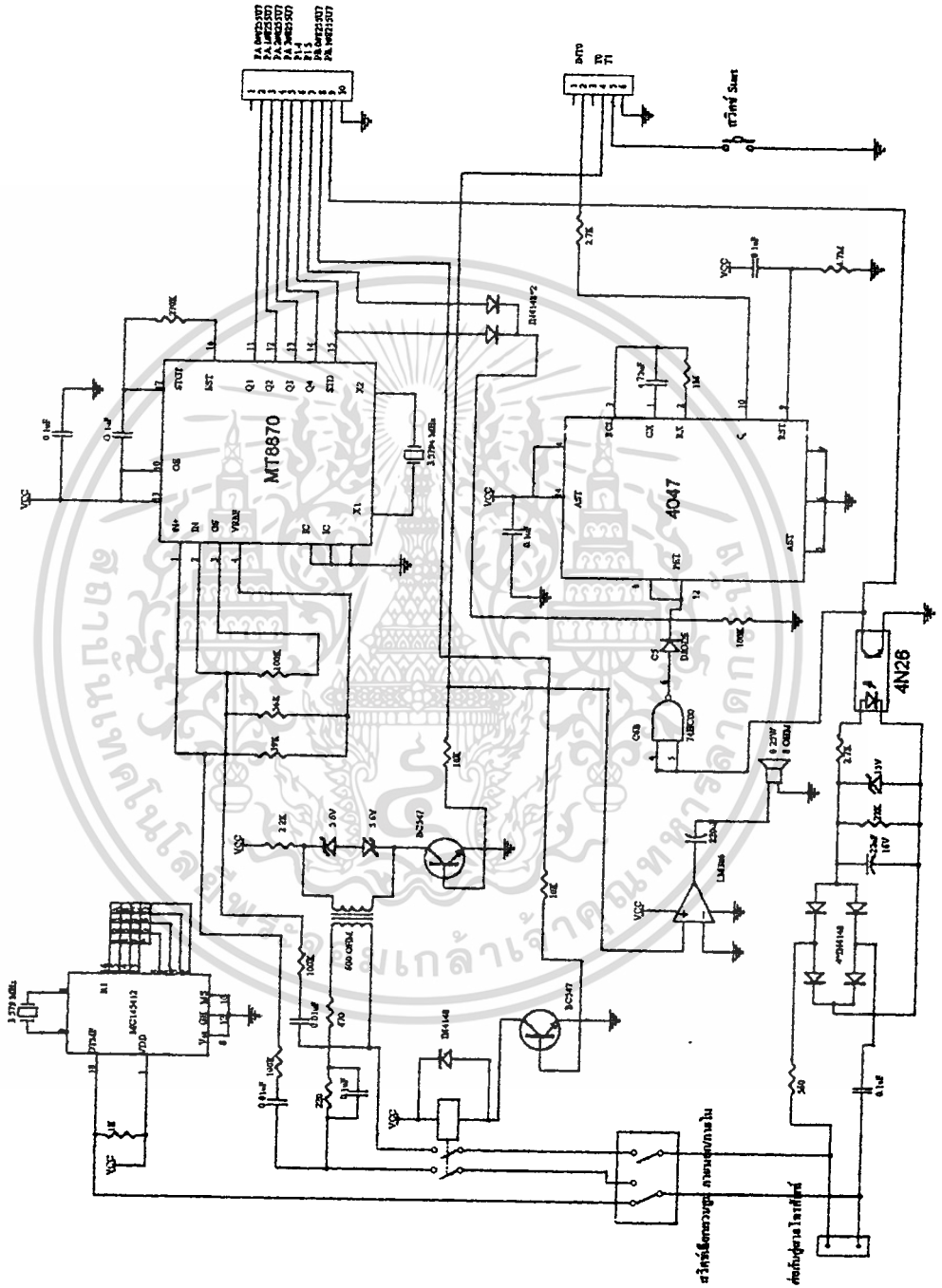
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก A

วงจรรวม

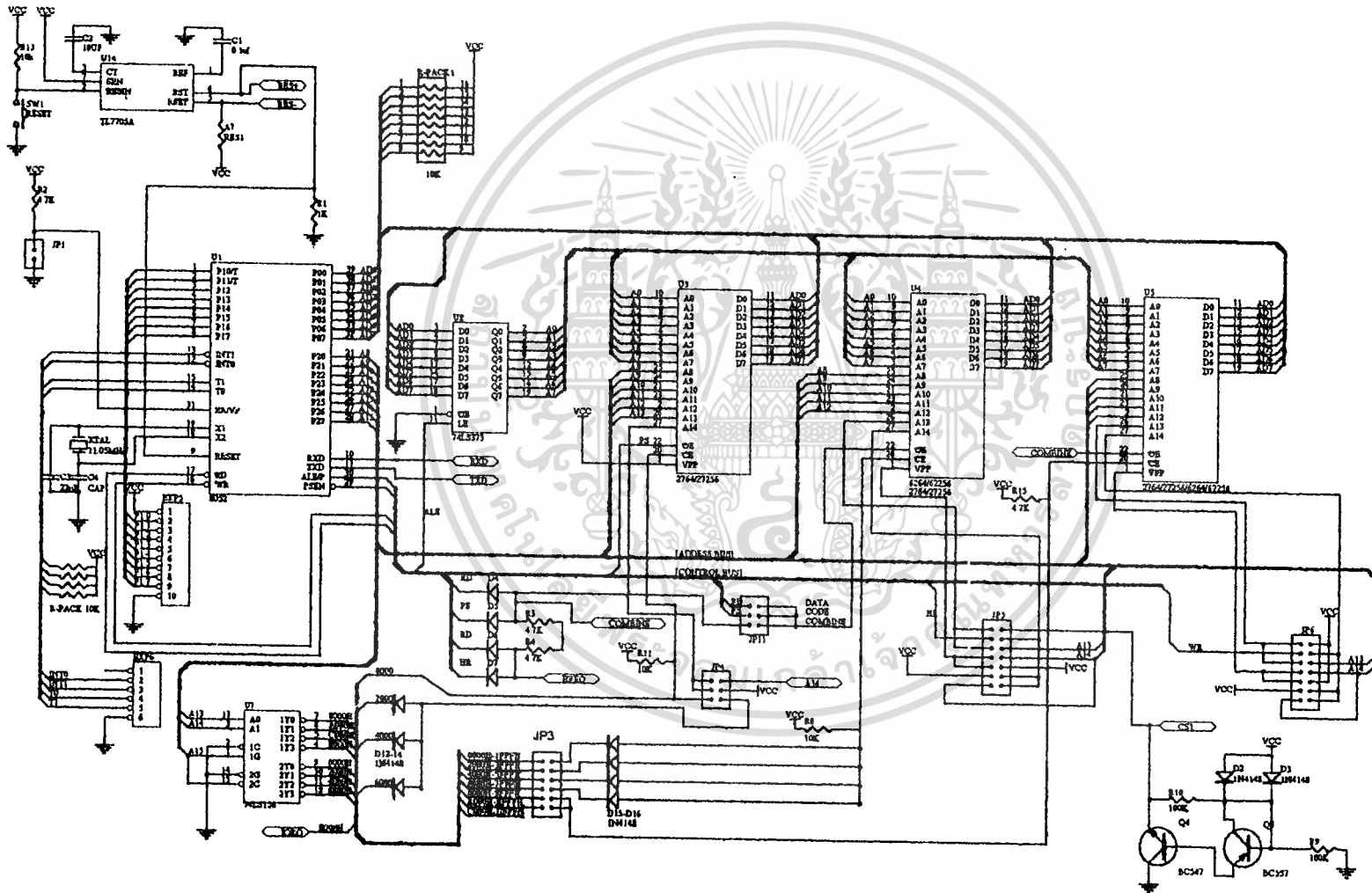


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

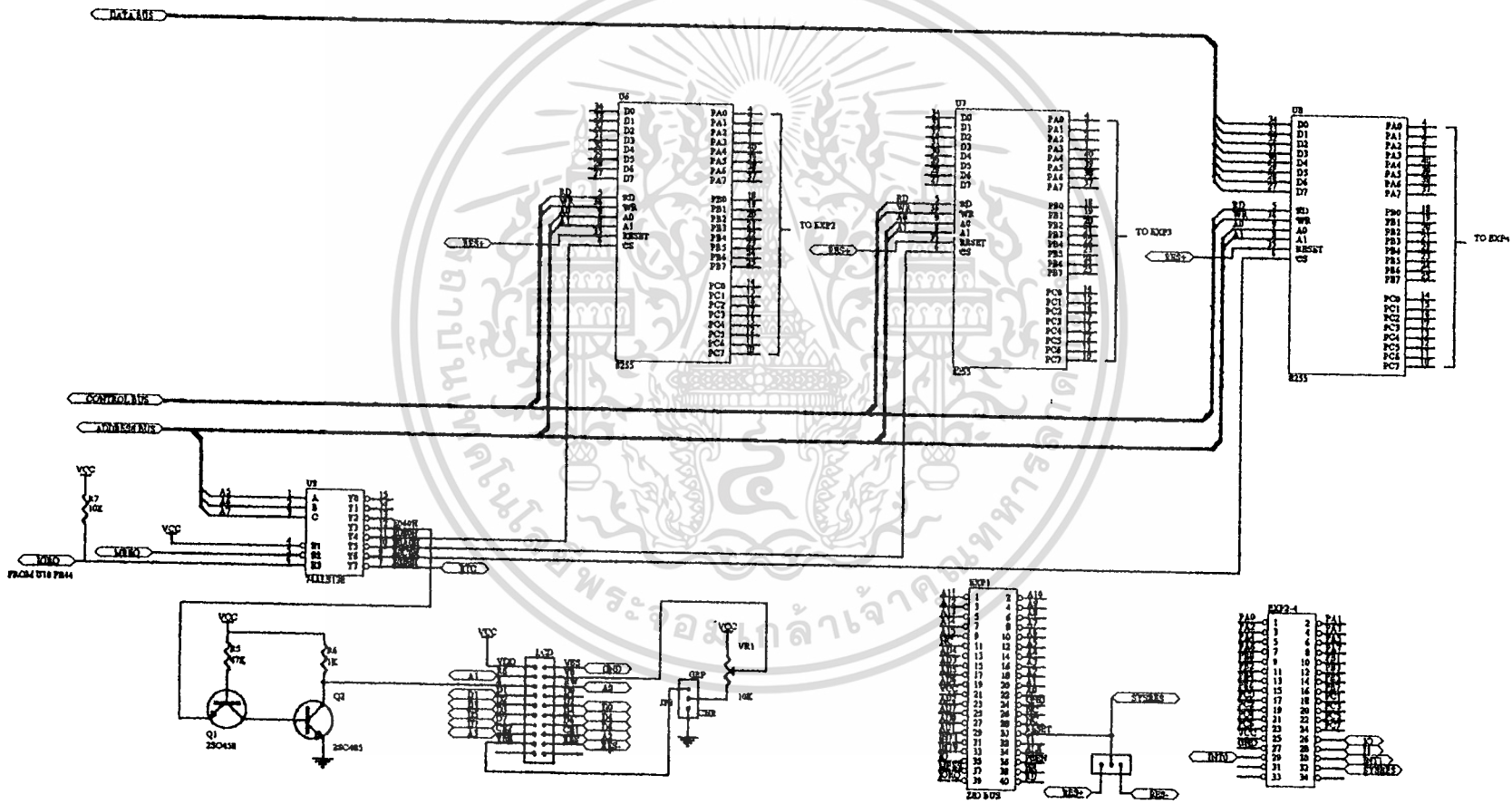


ภาคควบคุมและตรวจสอบไมโครสวิตช์

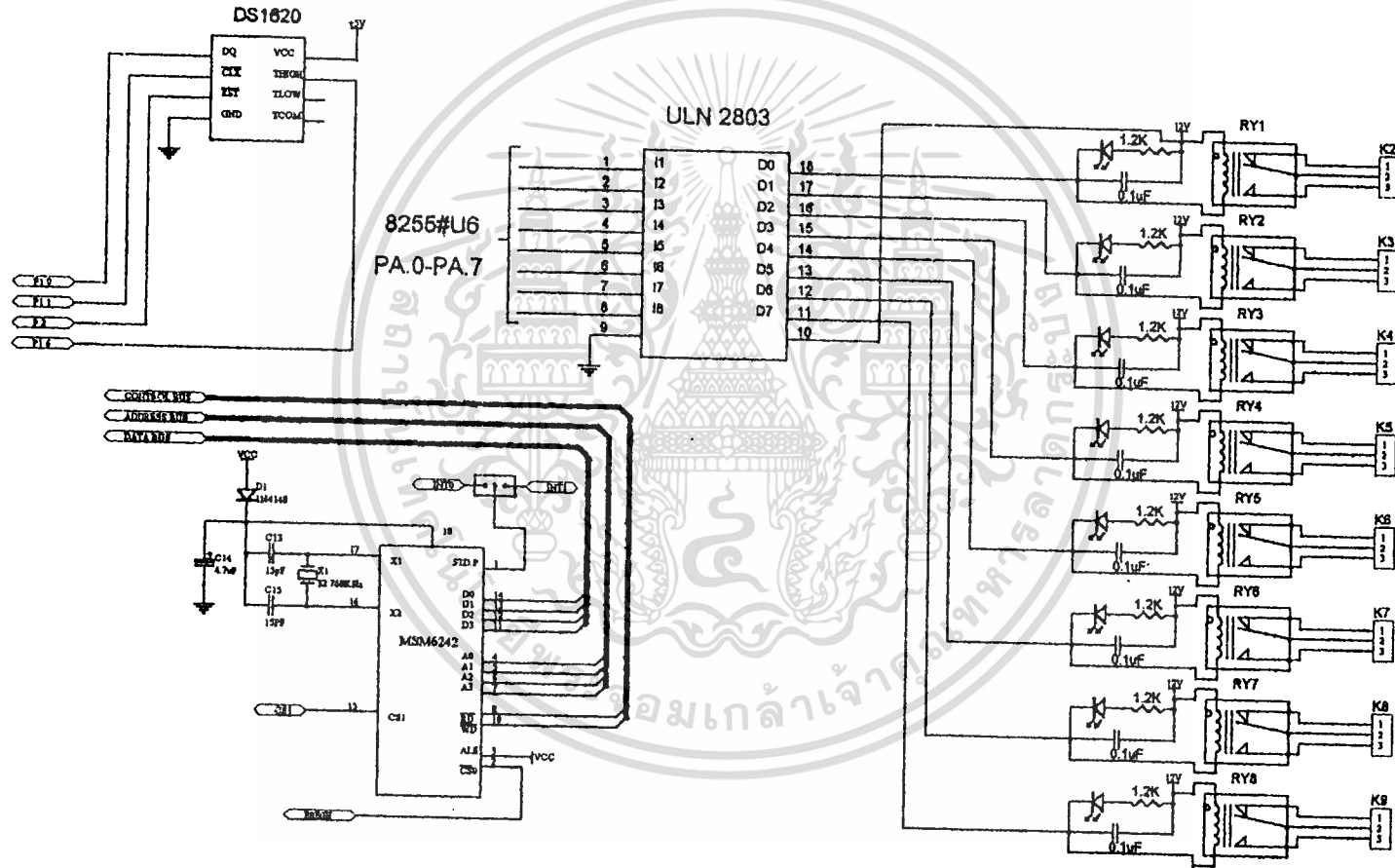
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคควบคุมส่วนกลาง MCS-51 และหน่วยความจำ



วงจรส่วนแสดงผล LCD และ พอร์ตต่อใช้งาน 8255



ภาคส่วน OUTPUT ฐานเวลา(MSM6242) และวงจรวัดพจนภูมิ(DS1620)

ภาคผนวก B

ASSEMBLER PROGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;FILENAME      NEW4.ASM
;DESCRIPTION   SYSTEM COMPUTER
;HARDWAVRE    SEKSAN CH.
;ASSEMBLER    SXA51
;START-DATE   10/05/1996
;***** VARIABLE *****
COM_PORT EQU 0E060H ;READ-WRITE RES
RD_BUSY EQU 0E061H ;READ(BUSY FLAG)
DAT_PORT EQU 0E062H ;WRITE CHARACTER
RD_DATA EQU 0E063H ;READ DATA FROM DD RAM
RTCPOT EQU 0E0E0H ;RTC(M6242 OKI)

PORTAU6 EQU 0E0C0H ;PORTA 8255#U6
PORTBU6 EQU 0E0C1H ;PORTB 8255#U6
PORTCU6 EQU 0E0C2H ;PORTC 8255#U6
CONT_U6 EQU 0E0C3H ;PORTD 8255#U6

PORTAU8 EQU 0E0A0H ;PORTA 8255#U8
PORTBU8 EQU 0E0A1H ;PORTB 8255#U8
PORTCU8 EQU 0E0A2H ;PORTC 8255#U8
CONT_U8 EQU 0E0A3H ;PORTD 8255#U8

DATAAPP EQU 9A00H
DATAFF EQU 9B00H
DATAOF EQU 9C00H
DATA_CH00 EQU 9000H
DATA_CH01 EQU 9010H
DATA_CH02 EQU 9020H
DATA_CH03 EQU 9030H
DATA_CH04 EQU 9040H
DATA_CH05 EQU 9050H
DATA_CH06 EQU 9060H
DATA_CH07 EQU 9070H
DATA_CH08 EQU 9080H
LOAD_D EQU 8000H
DATA_CHX1 EQU 8010H
DATA_CHY1 EQU 8020H
DATA_CHZ1 EQU 8030H

RING BIT P1.7
RET4047 BIT P1.5
INSTD BIT P1.4
THIGH BIT P1.3
TMDATA BIT P1.2
TMCLCK BIT P1.1
TMRSTT BIT P1.0

BUF_TEL EQU 10H
LCD_BUF EQU 68H ;LCD_BUF-LCD_BUF+4
TEMP_C EQU 58H ;TEMP_C-TEMP_C+2
TEMP_TH EQU 5BH ;TEMP_TH-TEMP_TH+2
TEMP_TL EQU 40H ;TEMP_TL-TEMP_TL+2
INTT1 EQU 25H
INTT2 EQU 26H
INTT3 EQU 27H
INHHH EQU 28H
INLLL EQU 29H

T_DELAYS EQU 05H
OUTP EQU 20H
OUTPB EQU 21H

OUTINT0 EQU 2AH
INMT_70 EQU 2BH
PLAYREC EQU 2CH
CON_PLAY EQU 2DH

CHK EQU 22H
RUN_STEP EQU 23H
TIM_SEC EQU 3FH
TIM_MIN EQU 3EH
TIM_HOUR EQU 3DH
BUF_TIM EQU 50H ;BUF_TIM-BUF_TIM+7

DATE_Y EQU 4FH
DATE_M EQU 4EH
DATE_D EQU 4DH
BUF_DATE EQU 60H ;BUF_DATE-BUF_DATE+7
DATE_WEEK EQU 4CH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนใช้งานเพื่อการศึกษเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BUF_INPUT EQU 3CH
COUNT EQU 30H ;COUNTER
CODE1 EQU 32H
CODE2 EQU 33H
CODE3 EQU 34H
CODE4 EQU 35H
CODE5 EQU 36H
CODE6 EQU 37H
CODE7 EQU 38H
CODE8 EQU 39H
CODE9 EQU 3AH

BUF_CODE EQU 40H ;BUF_CODE-BUF_CODE+9
BUF_COUNT EQU 70H
BUF_DTMF EQU 71H
L_SOUND EQU 72H
TCHXX EQU 73H
TCHZZ EQU 74H
B_HONOX EQU 75H
B_MONOX EQU 76H
B_HOFFOX EQU 77H
B_MOFFOX EQU 78H
BTTLX EQU 79H
BTTHX EQU 7AH
HOUR_ON EQU 7BH
HOUR_OFF EQU 7CH
MIN_ON EQU 7DH
MIN_OFF EQU 7EH

;***** PROGRAM *****
ORG 0000H
LJMP START

ORG 0003H ;INTERRUPT 0
INTOLOOP: LCALL INT_INT0
LJMP SYSTEM

ORG 0030H
INT_INT0: CLR IE.7
RETI

ORG 0050H
;*****
START: MOV SP,#90H
LCALL DELAY

MOV DPTR,#CONT_U6
MOV A,#80H ;PA=0,PB=0,PC=0
MOVX @DPTR,A
LCALL DELAY

MOV DPTR,#CONT_U8
MOV A,#92H ;PA=1,PB=1,PCH=0,PCL=0
MOVX @DPTR,A
LCALL DELAY

MOV A,#00111000B ;FUNCTION SET
LCALL WR_COMM
LCALL DELAY
MOV A,#00001100B ;DISPLAY ON/OFF
LCALL WR_COMM
LCALL DELAY
MOV A,#01H ;CLEAR
LCALL WR_COMM
LCALL DELAY
LCALL RTCINT
LCALL DELAY

LCALL SEND
MOV DPTR,#PORTAU6
MOV A,OUTP
MOVX @DPTR,A ;OFF PORTA

MOV DPTR,#PORTBU6
MOV A,OUTPB
MOVX @DPTR,A ;OFF PORTB

MOV CON_PLAY,#OFFH
LCALL OUTPUT_PC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    PLAYREC,#00H
LCALL  OUTU8_PC

LCALL  LSCHAR
MOV    CODE1,#06H
MOV    CODE2,#05H
MOV    CODE3,#04H
MOV    CODE4,#03H
MOV    CODE5,#02H
MOV    CODE6,#01H
MOV    CODE7,#0CH
MOV    COUNT,#06
MOV    L_SOUND,#02H

;***** SYSTEM *****
SYSTEM: MOV    DPTR,#TAB3
        LCALL  DISPLAY

        MOV    PLAYREC,#00H
        LCALL  OUTU8_PC
        MOV    CON_PLAY,#0FFH
        LCALL  OUTPUT_PC

        LCALL  C_CURSOR
        LCALL  SUBDATE
        LCALL  RESETP1 ;R0=COUNT,P1=#0BFH
        MOV    R2,#0CH ;WRITE CONFIG
        MOV    R3,#00000011B
        LCALL  DTWR8
CHECK1: LCALL  LEDINTO ;CHECK IE(INTO)
        LCALL  TIM_DATE
        MOV    R2,#0EEH ;START CONVERT
        LCALL  DTWR
        MOV    R2,#0ACH ;READ CONFIG (DONE BIT)
        LCALL  DTRD8
        LCALL  SCANTIM
        JNB   THIGH,TMPLOOP
        LCALL  TMPON16
        SJMP  CHECKL_P
TMPLOOP: LCALL  TMPOFF16
CHECKL_P: JNB   T1,CHECK3
        JB    RING,CHECK1
        MOV    R0,COUNT
RINGCALL: JB    RING,RINGCALL
        LCALL  DELAY
        JB    RING,RINGCALL
        JNB   RING,$
        LCALL  DELAY
        DJNZ  R0,RINGCALL
CHECK2: SETB   PLAYREC.7 ;UP RING
        LCALL  OUTU8_PC
        LCALL  DELAY

        SETB   PLAYREC.5 ;UP UBEEP
        LCALL  OUTU8_PC
        LCALL  UBEEP1 ;OK

        SETB   PLAYREC.1 ;UP TEL
        LCALL  OUTU8_PC
        LCALL  DELAY
        LCALL  DELAY
        CLR   CON_PLAY.3 ;PLAY SOUND PA
        LCALL  OUTPUT_PC
        MOV   IE,#00H ;NEW2
        MOV   R1,#15
        LCALL  DSEC
        LCALL  UBEEP1
        SETB  CON_PLAY.3 ;OFF SOUND PA
        LCALL  OUTPUT_PC
        LCALL  INPUTP1
        CJNE  A,#09H,CHECK4 ;*****
        CLR   EA
        LCALL  DELAY
        SETB  PLAYREC.2 ;TAPE (REC)
        LCALL  OUTU8_PC
        MOV   R1,#40
        LCALL  DSEC
        LCALL  DELAY
        LCALL  DELAY
        SJMP  CHECK4

```

```

CHECK3:  SETB  PLAYREC.7
          LCALL OUTU8_PC

          SETB  PLAYREC.5
          LCALL OUTU8_PC
          LCALL DELAY

          LCALL UBEEP1  ;OK

CHECK4:  LCALL LEDINT0
          LCALL DELAY
          LCALL DELAY

          MOV   A, #0CH
          CJNE  A, CODE1, $+9
          LCALL UBEEP1
          LJMP  PASS
          LCALL INPUTP1
          CJNE  A, CODE1, NOPASS

          MOV   A, #0CH
          CJNE  A, CODE2, $+9
          LCALL UBEEP1
          LJMP  PASS
          LCALL INPUTP1
          CJNE  A, CODE2, NOPASS

          MOV   A, #0CH
          CJNE  A, CODE3, $+9
          LCALL UBEEP1
          LJMP  PASS
          LCALL INPUTP1
          CJNE  A, CODE3, NOPASS

          MOV   A, #0CH
          CJNE  A, CODE4, $+15
          LCALL UBEEP1
          LJMP  PASS

NOPASS:  LCALL UBEEP3
          LCALL SYSTEM
          LCALL INPUTP1
          CJNE  A, CODE4, NOPASS

          MOV   A, #0CH
          CJNE  A, CODE5, $+9
          LCALL UBEEP1
          LJMP  PASS
          LCALL INPUTP1
          CJNE  A, CODE5, NOPASS

          MOV   A, #0CH
          CJNE  A, CODE6, $+9
          LCALL UBEEP1
          LJMP  PASS
          LCALL INPUTP1
          CJNE  A, CODE6, NOPASS

          MOV   A, #0CH
          CJNE  A, CODE7, $+9
          LCALL UBEEP1
          LJMP  PASS
          LCALL INPUTP1
          CJNE  A, CODE7, NOPASS

          MOV   A, #0CH
          CJNE  A, CODE8, $+9
          LCALL UBEEP1
          LJMP  PASS
          LCALL INPUTP1
          CJNE  A, CODE8, NOPASS

          MOV   A, #0CH
          CJNE  A, CODE9, NOPASS
          LCALL UBEEP1
          LJMP  PASS

```

```

COLO:    CJNE  A, #0AH, COL1
          LCALL INPUT1
          LJMP  CHANNAL1
          LJMP  PASS
COL1:    CJNE  A, #01H, COLW
          LCALL INPUT1
          LJMP  CHANNAL10
          LJMP  PASS
COLW:    CJNE  A, #0BH, COLX
          LJMP  MONITER
          LJMP  PASS
COLX:    CJNE  A, #0CH, COLZ
          LJMP  SYSTEM
COLZ:    LCALL ERROR
          LJMP  PASS

CHANNAL1: CJNE  A, #01H, CHANNAL2
          MOV   TCHXX, A
          LCALL INPUT1
          CJNE  A, #0AH, NAL1_1
          MOV   A, OUTP
          ANL   A, #0FEH
          MOV   OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_OFF
          LJMP  PASS
NAL1_1:  CJNE  A, #01H, NAL1_2
          MOV   A, OUTP
          ORL   A, #01H
          MOV   OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_ON
          LJMP  PASS
NAL1_2:  CJNE  A, #02H, NAL1_3
          MOV   A, OUTP
          ANL   A, #01H
          JNZ   CHAL1
          LCALL CHO_OFF
          LJMP  PASS
CHAL1:   LCALL CHO_ON
          LJMP  PASS
NAL1_3:  CJNE  A, #03H, NAL1_4
          LCALL D_TIME
          LCALL BTT1
          LJMP  PASS
NAL1_4:  CJNE  A, #04H, NAL1_5
          MOV   TCHZZ, #10H
          LCALL XX_TIME
          LCALL BXXX1
          LJMP  PASS
NAL1_5:  CJNE  A, #05H, NAL1_6
          MOV   TCHZZ, #20H
          LCALL XX_TIME
          LCALL BYY1
          LJMP  PASS
NAL1_6:  CJNE  A, #06H, NAL1_7
          MOV   TCHZZ, #30H
          LCALL XX_TIME
          LCALL BZZZ1
          LJMP  PASS
NAL1_7:  LCALL ERROR
          LJMP  PASS

CHANNAL2: CJNE  A, #02H, CHANNAL3
          MOV   TCHXX, A
          LCALL INPUT1
          CJNE  A, #0AH, NAL2_1
          MOV   A, OUTP
          ANL   A, #0FDH
          MOV   OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_OFF
          LJMP  PASS
NAL2_1:  CJNE  A, #01H, NAL2_2
          MOV   A, OUTP
          ORL   A, #02H
          MOV   OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_ON
          LJMP  PASS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในสถานศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NAL2_2: CJNE A, #02H, NAL2_3
        MOV A, OUTP
        ANL A, #02H
        JNZ CHAL2
        LCALL CHO_OFF
        LJMP PASS
CHAL2:  LCALL CHO_ON
        LJMP PASS
NAL2_3: CJNE A, #03H, NAL2_4
        LCALL D_TIME
        LCALL BTTT2
        LJMP PASS
NAL2_4: LCALL ERROR
        LJMP PASS

CHANNAL3: CJNE A, #03H, CHANNAL4
          MOV TCHXX, A
          LCALL INPUTP1
          CJNE A, #0AH, NAL3_1
          MOV A, OUTP
          ANL A, #0FBH
          MOV OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_OFF
          LJMP PASS
NAL3_1: CJNE A, #01H, NAL3_2
          MOV A, OUTP
          ORL A, #04H
          MOV OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_ON
          LJMP PASS
NAL3_2: CJNE A, #02H, NAL3_3
          MOV A, OUTP
          ANL A, #04H
          JNZ CHAL3
          LCALL CHO_OFF
          LJMP PASS
CHAL3:  LCALL CHO_ON
          LJMP PASS
NAL3_3: CJNE A, #03H, NAL3_4
          LCALL D_TIME
          LCALL BTTT3
          LJMP PASS
NAL3_4: LCALL ERROR
          LJMP PASS

CHANNAL4: CJNE A, #04H, CHANNAL5
          MOV TCHXX, A
          LCALL INPUTP1
          CJNE A, #0AH, NAL4_1
          MOV A, OUTP
          ANL A, #0F7H
          MOV OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_OFF
          LJMP PASS
NAL4_1: CJNE A, #01H, NAL4_2
          MOV A, OUTP
          ORL A, #08H
          MOV OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_ON
          LJMP PASS
NAL4_2: CJNE A, #02H, NAL4_3
          MOV A, OUTP
          ANL A, #08H
          JNZ CHAL4
          LCALL CHO_OFF
          LJMP PASS
CHAL4:  LCALL CHO_ON
          LJMP PASS
NAL4_3: CJNE A, #03H, NAL4_4
          LCALL D_TIME
          LCALL BTTT4
          LJMP PASS
NAL4_4: LCALL ERROR
          LJMP PASS

```

```

MOV    TCHXX,A
LCALL INPUTP1
CJNE  A,#0AH,NAL5_1
MOV   A,OUTP
ANL   A,#0EFH
MOV   OUTP,A
LCALL OUTPUT_PA
LCALL CHO_OFF
LJMP  PASS
NAL5_1: CJNE  A,#01H,NAL5_2
MOV   A,OUTP
ORL   A,#10H
MOV   OUTP,A
LCALL OUTPUT_PA
LCALL CHO_ON
LJMP  PASS
NAL5_2: CJNE  A,#02H,NAL5_3
MOV   A,OUTP
ANL   A,#10H
JNZ   CHAL5
LCALL CHO_OFF
LJMP  PASS
CHAL5: LCALL CHO_ON
LJMP  PASS
NAL5_3: CJNE  A,#03H,NAL5_4
LCALL D_TIME
LCALL BTTF5
LJMP  PASS
NAL5_4: LCALL ERROR
LJMP  PASS

CHANNAL6: CJNE  A,#06H,CHANNAL7
MOV   TCHXX,A
LCALL INPUTP1
CJNE  A,#0AH,NAL6_1
MOV   A,OUTP
ANL   A,#0DFH
MOV   OUTP,A
LCALL OUTPUT_PA
LCALL CHO_OFF
LJMP  PASS
NAL6_1: CJNE  A,#01H,NAL6_2
MOV   A,OUTP
ORL   A,#20H
MOV   OUTP,A
LCALL OUTPUT_PA
LCALL CHO_ON
LJMP  PASS
NAL6_2: CJNE  A,#02H,NAL6_3
MOV   A,OUTP
ANL   A,#20H
JNZ   CHAL6
LCALL CHO_OFF
LJMP  PASS
CHAL6: LCALL CHO_ON
LJMP  PASS
NAL6_3: CJNE  A,#03H,NAL6_4
LCALL D_TIME
LCALL BTTF6
LJMP  PASS
NAL6_4: LCALL ERROR
LJMP  PASS

CHANNAL7: CJNE  A,#07H,CHANNAL8
MOV   TCHXX,A
LCALL INPUTP1
CJNE  A,#0AH,NAL7_1
MOV   A,OUTP
ANL   A,#0BFH
MOV   OUTP,A
LCALL OUTPUT_PA
LCALL CHO_OFF
LJMP  PASS
NAL7_1: CJNE  A,#01H,NAL7_2
MOV   A,OUTP
ORL   A,#40H
MOV   OUTP,A
LCALL OUTPUT_PA
LCALL CHO_ON
LJMP  PASS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NAL7_2: CJNE A, #02H, NAL7_3
        MOV  A, OUTP
        ANL  A, #40H
        JNZ  CHAL7
        LCALL CHO_OFF
        LJMP PASS
CHAL7:  LCALL CHO_ON
        LJMP PASS
NAL7_3: CJNE A, #03H, NAL7_4
        LCALL D_TIME
        LCALL BTTT7
        LJMP PASS
NAL7_4: LCALL ERROR
        LJMP PASS

CHANNAL8: CJNE A, #08H, CHANNAL9
          MOV  TCHXX, A
          LCALL INPUTP1
          CJNE A, #0AH, NAL8_1
          MOV  A, OUTP
          ANL  A, #07FH
          MOV  OUTP, A
          LCALL OUTPUT_PA
          LCALL CHO_OFF
          LJMP PASS
NAL8_1: CJNE A, #01H, NAL8_2
        MOV  A, OUTP
        ORL  A, #80H
        MOV  OUTP, A
        LCALL OUTPUT_PA
        LCALL CHO_ON
        LJMP PASS
NAL8_2: CJNE A, #02H, NAL8_3
        MOV  A, OUTP
        ANL  A, #80H
        JNZ  CHAL8
        LCALL CHO_OFF
        LJMP PASS
CHAL8:  LCALL CHO_ON
        LJMP PASS
NAL8_3: CJNE A, #03H, NAL8_4
        LCALL D_TIME
        LCALL BTTT8
        LJMP PASS
NAL8_4: LCALL ERROR
        LJMP PASS

CHANNAL9: CJNE A, #09H, CHAN_10X
          MOV  TCHXX, A
          LCALL INPUTP1
          CJNE A, #0AH, NAL9_1
          MOV  A, OUTPB
          ANL  A, #0FEH
          MOV  OUTPB, A
          LCALL OUTPUT_PB
          LCALL CH09_OFF
          LJMP PASS
NAL9_1: CJNE A, #01H, NAL9_2
        MOV  A, OUTPB
        ORL  A, #01H
        MOV  OUTPB, A
        LCALL OUTPUT_PB
        LCALL CH09_ON
        LJMP PASS
NAL9_2: CJNE A, #02H, NAL9_3
        MOV  A, OUTPB
        ANL  A, #01H
        JNZ  CHAL9
        LCALL CH09_OFF
        LJMP PASS
CHAL9:  LCALL CH09_ON
        LJMP PASS
NAL9_3: LCALL ERROR
        LJMP PASS

CHAN_10X: LJMP PASS

CHANNAL10: CJNE A, #0AH, CHANNAL11
           CLR  A
           MOV  TCHXX, A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL INPUTP1
        CJNE A,#0AH,NAL10_1
        MOV A,OUTPB
        ANL A,#0FDH
        MOV OUTPB,A
        LCALL OUTPUT_PB
        LCALL CHI_OFF
        LJMP PASS
NAL10_1: CJNE A,#01H,NAL10_2
        MOV A,OUTPB
        ORL A,#02H
        MOV OUTPB,A
        LCALL OUTPUT_PB
        LCALL CHI_ON
        LJMP PASS
NAL10_2: CJNE A,#02H,NAL10_3
        MOV A,OUTPB
        ANL A,#02H
        JNZ CHAL10
        LCALL CHI_OFF
        LJMP PASS
CHAL10: LCALL CHI_ON
        LJMP PASS
NAL10_3: LCALL ERROR
        LJMP PASS

CHANNAL11:CJNE A,#01H,CHANNAL12
        MOV TCHXX,A
        LCALL INPUTP1
        CJNE A,#0AH,NAL11_1
        MOV A,OUTPB
        ANL A,#0FBH
        MOV OUTPB,A
        LCALL OUTPUT_PB
        LCALL CHI_OFF
        LJMP PASS
NAL11_1: CJNE A,#01H,NAL11_2
        MOV A,OUTPB
        ORL A,#04H
        MOV OUTPB,A
        LCALL OUTPUT_PB
        LCALL CHI_ON
        LJMP PASS
NAL11_2: CJNE A,#02H,NAL11_3
        MOV A,OUTPB
        ANL A,#04H
        JNZ CHAL11
        LCALL CHI_OFF
        LJMP PASS
CHAL11: LCALL CHI_ON
        LJMP PASS
NAL11_3: LCALL ERROR
        LJMP PASS

CHANNAL12:CJNE A,#02H,CHANNAL13
        MOV TCHXX,A
        LCALL INPUTP1
        CJNE A,#0AH,NAL12_1
        MOV A,OUTPB
        ANL A,#0F7H
        MOV OUTPB,A
        LCALL OUTPUT_PB
        LCALL CHI_OFF
        LJMP PASS
NAL12_1: CJNE A,#01H,NAL12_2
        MOV A,OUTPB
        ORL A,#08H
        MOV OUTPB,A
        LCALL OUTPUT_PB
        LCALL CHI_ON
        LJMP PASS
NAL12_2: CJNE A,#02H,NAL12_3
        MOV A,OUTPB
        ANL A,#08H
        JNZ CHAL12
        LCALL CHI_OFF
        LJMP PASS
CHAL12: LCALL CHI_ON
        LJMP PASS
NAL12_3: LCALL ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LJMP PASS

CHANNAL13:CJNE A,#03H,CHANNAL14
MOV TCHXX,A
LCALL INPUTP1
CJNE A,#0AH,NAL13_1
MOV A,OUTPB
ANL A,#0EFH
MOV OUTPB,A
LCALL OUTPUT_PB
LCALL CH1_OFF
LJMP PASS
NAL13_1: CJNE A,#01H,NAL13_2
MOV A,OUTPB
ORL A,#10H
MOV OUTPB,A
LCALL OUTPUT_PB
LCALL CH1_ON
LJMP PASS
NAL13_2: CJNE A,#02H,NAL13_3
MOV A,OUTPB
ANL A,#10H
JNZ CHAL13
LCALL CH1_OFF
LJMP PASS
CHAL13: LCALL CH1_ON
LJMP PASS
NAL13_3: LCALL ERROR
LJMP PASS

CHANNAL14:CJNE A,#04H,CHANNAL15
MOV TCHXX,A
LCALL INPUTP1
CJNE A,#0AH,NAL14_1
MOV A,OUTPB
ANL A,#0DFH
MOV OUTPB,A
LCALL OUTPUT_PB
LCALL CH1_OFF
LJMP PASS
NAL14_1: CJNE A,#01H,NAL14_2
MOV A,OUTPB
ORL A,#20H
MOV OUTPB,A
LCALL OUTPUT_PB
LCALL CH1_ON
LJMP PASS
NAL14_2: CJNE A,#02H,NAL14_3
MOV A,OUTPB
ANL A,#20H
JNZ CHAL14
LCALL CH1_OFF
LJMP PASS
CHAL14: LCALL CH1_ON
LJMP PASS
NAL14_3: LCALL ERROR
LJMP PASS

CHANNAL15:CJNE A,#05H,CHANNAL16
MOV TCHXX,A
LCALL INPUTP1
CJNE A,#0AH,NAL15_1
MOV A,OUTPB
ANL A,#0BFH
MOV OUTPB,A
LCALL OUTPUT_PB
LCALL CH1_OFF
LJMP PASS
NAL15_1: CJNE A,#01H,NAL15_2
MOV A,OUTPB
ORL A,#40H
MOV OUTPB,A
LCALL OUTPUT_PB
LCALL CH1_ON
LJMP PASS
NAL15_2: CJNE A,#02H,NAL15_3
MOV A,OUTPB
ANL A,#40H
JNZ CHAL15
LCALL CH1_OFF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LJMP PASS
CHAL15: LCALL CH1_ON
        LJMP PASS
NAL15_3: LCALL ERROR
        LJMP PASS

CHANNAL16: CJNE A, #06H, CHANNAL17
        MOV TCHXX, A
        LCALL INPUTP1
        CJNE A, #0AH, NAL16_1
        MOV A, OUTPB
        ANL A, #07FH
        MOV OUTPB, A
        LCALL OUTPUT_PB
        LCALL CH1_OFF
        LJMP PASS
NAL16_1: CJNE A, #01H, NAL16_2
        MOV A, OUTPB
        ORL A, #80H
        MOV OUTPB, A
        LCALL OUTPUT_PB
        LCALL CH1_ON
        LJMP PASS
NAL16_2: CJNE A, #02H, NAL16_3
        MOV A, OUTPB
        ANL A, #80H
        JNZ CHAL16
        LCALL CH1_OFF
        LJMP PASS
CHAL16: LCALL CH1_ON
        LJMP PASS
NAL16_3: LCALL ERROR
        LJMP PASS

CHANNAL17: CJNE A, #07H, CHAN_A1
        LCALL INPUTP1
NAL17_0: CJNE A, #0AH, NAL17_1
        MOV A, #00H
        MOV OUTP, A
        MOV OUTPB, A
        LCALL OUTPUT_PA
        LCALL OUTPUT_PB
        LCALL UBEEP3
        LJMP PASS
CHAN_A1: LCALL ERROR
        LJMP PASS
NAL17_1: CJNE A, #01H, CHAN_A1
        MOV A, OUTP ;OUT PA=01
        ORL A, #01H
        MOV OUTP, A
        LCALL OUTPUT_PA
        LCALL DELAY

        MOV A, OUTP ;OUT PA=02
        ORL A, #02H
        MOV OUTP, A
        LCALL OUTPUT_PA
        LCALL DELAY

        MOV A, OUTP ;OUT PA=04
        ORL A, #04H
        MOV OUTP, A
        LCALL OUTPUT_PA
        LCALL DELAY

        MOV A, OUTP ;OUT PA=08
        ORL A, #08H
        MOV OUTP, A
        LCALL OUTPUT_PA
        LCALL DELAY

        MOV A, OUTP ;OUT PA=10
        ORL A, #10H
        MOV OUTP, A
        LCALL OUTPUT_PA
        LCALL DELAY

        MOV A, OUTP ;OUT PA=20
        ORL A, #20H
        MOV OUTP, A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL OUTPUT_PA
LCALL DELAY

MOV  A,OUTP      ;OUT PA=40
ORL  A,#40H
MOV  OUTP,A
LCALL OUTPUT_PA
LCALL DELAY

MOV  A,OUTP      ;OUT PA=80
ORL  A,#80H
MOV  OUTP,A
LCALL OUTPUT_PA
LCALL DELAY

MOV  A,OUTPB     ;OUT PB=01
ORL  A,#01H
MOV  OUTPB,A
LCALL OUTPUT_PB
LCALL DELAY

MOV  A,OUTPB     ;OUT PB=02
ORL  A,#02H
MOV  OUTPB,A
LCALL OUTPUT_PB
LCALL DELAY

MOV  A,OUTPB     ;OUT PB=04
ORL  A,#04H
MOV  OUTPB,A
LCALL OUTPUT_PB
LCALL DELAY

MOV  A,OUTPB     ;OUT PB=08
ORL  A,#08H
MOV  OUTPB,A
LCALL OUTPUT_PB
LCALL DELAY

MOV  A,OUTPB     ;OUT PB=10
ORL  A,#10H
MOV  OUTPB,A
LCALL OUTPUT_PB
LCALL DELAY

MOV  A,OUTPB     ;OUT PB=20
ORL  A,#20H
MOV  OUTPB,A
LCALL OUTPUT_PB
LCALL DELAY

MOV  A,OUTPB     ;OUT PB=40
ORL  A,#40H
MOV  OUTPB,A
LCALL OUTPUT_PB
LCALL DELAY

MOV  A,OUTPB     ;OUT PB=80
ORL  A,#80H
MOV  OUTPB,A
LCALL OUTPUT_PB
LCALL DELAY
LCALL UBEEP1
LCALL LEDINT0
LJMP PASS

;***** MONITER *****
MONITER: LCALL INPUTP1      ;MONITER MODE
FUNC0:   CJNE  A,#0AH,FUNC2 ;FUNC 0
         LCALL UBEEP2
         LJMP RECSOUND
FUNC1:   CJNE  A,#01H,FUNC2 ;FUNC 1
         LCALL UBEEP1      ;SET PASSWORD
         LCALL FUNC_1
         LJMP PASS
FUNC2:   CJNE  A,#02H,FUNC3 ;FUNC 2
         LCALL UBEEP1      ;DELAY RING
         LCALL FUNC_2
LJMP PASS
FUNC3:   CJNE  A,#03H,FUNC4 ;FUNC 3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก
 LJMP PASS การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 FUNC3: CJNE A, #03H, FUNC4 ; FUNC 3
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL UBEEP1          ;TEST DMTF
LCALL FUNC_3
LJMP PASS_
FUNC4: CJNE A,#04H,FUNC5 ;FUNC 4
LCALL FUNC_4          ;SET SOUND
LJMP PASS_
FUNC5: CJNE A,#05H,FUNC6 ;FUNC 5
LCALL FUNC_5          ;SET TIME
LJMP PASS_
FUNC6: CJNE A,#06H,FUNC7 ;FUNC 6
LCALL FUNC_6          ;SET TIME
LJMP PASS_
FUNC7: CJNE A,#07H,FUNC8 ;FUNC 7
LJMP FUNC_7           ;READ TEMP & INPUT
FUNC8: CJNE A,#08H,FUNC9
LCALL TEMP            ;SET TEMP
LJMP PASS_
FUNC9: CJNE A,#09H,RETURE
LJMP TESTSOUND       ;TEST SOUND
RETURE: LCALL ERROR
LJMP SYSTEM

```

```

;TAB0: DB "TH . ",0," [ : : ] "
; DB "T . ",0," [ / / ] "
TAB1: DB "Time is [ : : ] "
DB "New time [ : : ] "
TAB2: DB "Date is [ / / ] "
DB "New date [ / / ] "
TAB3: DB "Time is [ : : ] "
DB "Date is [ / / ] "
TAB4: DB "CH 0_ON [ : ] "
DB "TIME OFF [ : ] "
TAB5: DB "MONITER MODE(FUNC 1)"
DB " SET PASSWORD "
TAB6: DB "MONITER MODE(FUNC 2)"
DB " SET DELAY RING "
TAB7: DB "MONITER MODE(FUNC 3)"
DB " SINGNAL TEST DMTF "
TAB8: DB "MONITER MODE(FUNC 4)"
DB "SET LONG SOUND (1-3)"
TAB9: DB "MCS-51 BY CPAT 32 "
DB "SEKSAN CHUENKIANKOT "
TAB10: DB " MODE OUTPUT DRIVER "
DB "CH 0_ON [ : ] "
TAB11: DB " MODE OUTPUT DRIVER "
DB "CH 0_OFF [ : ] "
TAB12: DB " MODE OUTPUT DRIVER "
DB "CH 1_ON [**:**] "
TAB13: DB " MODE OUTPUT DRIVER "
DB "CH 1_OFF [**:**] "
TAB14: DB " MODE OUTPUT DRIVER "
DB "CH 0_ON [**:**] "
TAB15: DB " MODE OUTPUT DRIVER "
DB "CH 0_OFF [**:**] "
TAB20: DB "!!!!!! ERROR !!!!!!"
DB "!!!!!! ERROR !!!!!!"
TAB21: DB " T-HIGH + . ",0,"C"
DB " T-LOW + . ",0,"C"
TAB22: DB "T-HIGH IS + . ",0,"C"
DB "NEW T-HIGH + . ",0,"C"
TAB23: DB "T-LOW IS + . ",0,"C"
DB "NEW T-LOW + . ",0,"C"
TAB24: DB "TEMPERATURE IN ROOM "
DB " + . ",0,"C "
TAB25: DB "1.READ TH,TL 2.IN TH"
DB "3.IN TL #.ENTER"
TAB26: DB "1.PLAY PA 2.PLAY PB"
DB "3.PLAY PC #.EXIT "
TAB27: DB "TEST SOUND CHANNAL A"
DB "ISD 1420P (PLAY/REC)"
TAB28: DB "TEST SOUND CHANNAL B"
DB "ISD 1420P (PLAY) "
TAB29: DB "TEST SOUND CHANNAL C"
DB "ISD 1420P (PLAY) "
TAB30: DB "1.REC PA 2.RECTAPE "
DB "#.EXIT "
TAB31: DB "REC SOUND CHANNAL A "
DB "ISD 1420P (REC) "
TAB32: DB "REC SOUND OF TAPEL C"
DB "LOW TIME 90 MIN "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกาใช้ภายในหน่วยงาน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAINTD:  DB    00001100B
          DB    00010010B
          DB    00001100B
          DB    00000000B
          DB    00000000B
          DB    00000000B
          DB    00000000B
          DB    00000000B

;***** SUB TMON16 *****
TMON16:  MOV    A,OUTPB
          ORL    A,#80H
          MOV    OUTPB,A
          LCALL OUTPUT_PB
          RET

;***** SUB TMOFF16 *****
TMOFF16: MOV    A,OUTPB
          ANL    A,#07FH
          MOV    OUTPB,A
          LCALL OUTPUT_PB
          RET

;***** SUB LOADDOWN *****
RECV:   MOV    DPTR,#TAB9
          LCALL DISPLAY
          MOV    DPTR,#LOAD_D
          MOV    A,OUTP
          MOVX   @DPTR,A
          MOV    DPTR,#LOAD_D+1
          MOV    A,OUTPB
          MOVX   @DPTR,A
          RET

;***** SUB SEND *****
SEND:   MOV    DPTR,#LOAD_D
          MOVX   A,@DPTR
          MOV    OUTP,A
          MOV    DPTR,#LOAD_D+1
          MOVX   A,@DPTR
          MOV    OUTPB,A
          RET

;***** SUB LEDINTO *****
LEDINTO: MOV    IE,#00H
          SETB  RET4047
          CLR   RET4047
          MOV    IE,#81H
          RET

;***** SUB DELAY *****
DELAY:  MOV    R6,#0B1H
DELAY_1: MOV    R7,#0FFH
          DJNZ  R7,$
          DJNZ  R6,DELAY_1
          RET

; ***** SUB DELAYS *****
; DELAY SUBROUTINE
; IN = R2
; REG = R2,R3,R4
DELAYS: MOV    R3,#0
DELAYS1: MOV    R4,#0
          DJNZ  R4,$
          DJNZ  R3,DELAYS1
          DJNZ  R2,DELAYS
          RET

;***** SUB DELAY1 *****
DELAY1:  MOV    R6,#50H
DELAY1_1: MOV    R7,#0FFH
          DJNZ  R7,$
          DJNZ  R6,DELAY1_1
          RET

;***** SUB DSEC *****
;DELAY_1 SECOND
;IN = R1
;REG = R1,R2,R3,R4

```

```

DSEC:    MOV     R2,#10
SDTSEC:  MOV     R3,#179      ;DELAY 1/10 SEC
SDTSEC1: MOV     R4,#00H
          DJNZ   R4,$
          NOP
          NOP
          DJNZ   R3,SDTSEC1
          DJNZ   R2,SDTSEC
          DJNZ   R1,DSEC
          RET

```

```

;***** SUB RESETP1 *****
RESETP1: MOV     IE,#00H
          CLR    PLAYREC.7
          LCALL  OUTPUT_PC
          MOVX   @DPTR,A
          MOV    R0,COUNT
          RET

```

```

;***** SUB SOUND *****
SOUND:   MOV     R5,L_SOUND
SOUND_1: MOV     R6,#0FFH
SOUND_2: MOV     R7,#0FFH
          SETB   P3.4
          DJNZ   R7,$
          MOV    R7,#0FFH
          CLR    P3.4
          DJNZ   R7,$
          DJNZ   R6,SOUND_2
          DJNZ   R5,SOUND_1
          RET

```

```

;***** SUB SOUND1 *****
SOUND1:  MOV     R5,#01H
SOUND1_1: MOV     R6,#0EEH
SOUND1_2: MOV     R7,#0EEH
          SETB   P3.4
          DJNZ   R7,$
          MOV    R7,#0EEH
          CLR    P3.4
          DJNZ   R7,$
          DJNZ   R6,SOUND1_2
          DJNZ   R5,SOUND1_1
          RET

```

```

;***** SUB OUTPUT_PA 8255#U6 *****
OUTPUT_PA:
          MOV    DPTR,#PORTAU6 ;DPTR=0E080H
          MOVX   @DPTR,A
          RET

```

```

;***** SUB OUTPUT_PB 8255#U6 *****
OUTPUT_PB:
          MOV    DPTR,#PORTBU6 ;DPTR=0E081H
          MOVX   @DPTR,A
          RET

```

```

;***** SUB OUTPUT_PC 8255#U6 *****
OUTPUT_PC:
          MOV    DPTR,#PORTCU6 ;DPTR=0E082H
          MOV    A,CON_PLAY
          MOVX   @DPTR,A
          RET

```

```

;***** SUB OUTPUT_PC 8255#U8 *****
OUTU8_PC: MOV    DPTR,#PORTCU8 ;DPTR=0E0C2H
          MOV    A,PLAYREC
          MOVX   @DPTR,A
          RET

```

```

;***** SUB INPUTP1 *****
INPUTP1: JNB     INSTD,$
          MOV    DPTR,#PORTBU8
          MOVX   A,@DPTR
          ANL   A,#0FH
          JB     INSTD,$
          LCALL LEDINT0
          RET

```

```

;***** SUB INPUTP2 *****
INPUTP2: JNB  INSTD,$
        MOV  DPTR,#PORTBUS
        MOVX A,@DPTR
        ANL  A,#0FH
        JB   INSTD,$
        LCALL LEDINT0
        CJNE A,#0AH,ZZ  ;0
        ANL  A,#00H
        RET

ZZ:     CJNE A,#0CH,ZZZ  ;#
        ANL  A,#00H
        RET

ZZZ:    CJNE A,#0BH,ZZZ2 ;*
        MOV  A,#''
ZZZ2:   RET

;***** SUB INPUTP3 *****
INPUTP3: JNB  INSTD,$
        MOV  DPTR,#PORTBUS
        MOVX A,@DPTR
        ANL  A,#0FH
        JB   INSTD,$
        LCALL LEDINT0
        CJNE A,#0BH,AAA  ;*
        MOV  A,#''
        RET

AAA:    CJNE A,#0AH,AAAA ;0
        ANL  A,#00H
AAAA:   RET

;***** SUB INPUTEL *****
INPUTEL: JNB  INSTD,$
        MOV  DPTR,#PORTBUS
        MOVX A,@DPTR
        ANL  A,#0FH
        JB   INSTD,$
        LCALL LEDINT0
        CJNE A,#0AH,OUT_INT ;0
        ANL  A,#00H
OUT_INT: RET

;***** SUB UBEEP1 *****
UBEEP1: LCALL SOUND  ;OK
        LCALL DELAY
        LCALL DELAY
        LCALL SOUND
        RET

;***** SUB UBEEP2 *****
UBEEP2: LCALL SOUND  ;TEST DTMF
        RET

;***** SUB UBEEP3 *****
UBEEP3: LCALL SOUND  ;NO OR OFF
        LCALL SOUND
        RET

;***** SUB ERROR SOUND *****
ERROR:  MOV  DPTR,#TAB20
        LCALL DISPLAY
        LCALL SOUND1  ;1
        LCALL DELAY1
        LCALL SOUND1  ;2
        LCALL DELAY1
        LCALL SOUND1  ;3
        LCALL DELAY1
        LCALL SOUND1  ;4
        RET

;***** SUB FUNC 0 *****
RECSOUND: MOV  DPTR,#TAB30
        LCALL DISPLAY
        LCALL UBEEP2
        SETB PLAYREC.1
        LCALL OUTU8_PC
        LCALL INPUTP1
RSOUND1: CJNE  A,#01H,RSOUND2
        LCALL REC_CHA

```

```

RSOUND2: SJMP RECSOUND
          CJNE A, #02H, RECEND
          LCALL REC_TAPE
          SJMP RECSOUND
RECEND:  CJNE A, #0CH, ERRORREC
          CLR PLAYREC.1
          LCALL OUTU8_PC
          SETB EA
          LJMP PASS
ERRORREC: LCALL ERROR
          CLR PLAYREC.1
          LCALL OUTU8_PC
          SETB EA
          LJMP PASS

;***** SUB REC_CHA *****
REC_CHA: CLR EA
          MOV DPTR, #TAB31
          LCALL DISPLAY

          LCALL DELAY
          SETB PLAYREC.0 ;ON RELAY>REC
          LCALL OUTU8_PC
          SETB PLAYREC.1
          LCALL OUTU8_PC ;UP TELEPHONE

          SETB PLAYREC.3 ;CUT BIT REC
          LCALL OUTU8_PC
          LCALL DELAY
          CLR CON_PLAY.2 ;ON ISD 1420(REC)
          LCALL OUTPUT_PC
          MOV R1, #20
          LCALL DSEC
          MOV PLAYREC, #00H ;OFF RELAY ALL
          LCALL OUTU8_PC
          SETB CON_PLAY.2 ;OFF REC ISD1420
          LCALL OUTPUT_PC
          SETB EA
          RET

;***** SUB REC_TAPE *****
REC_TAPE: CLR EA
          MOV DPTR, #TAB31
          LCALL DISPLAY

          LCALL DELAY
          CLR PLAYREC.0 ;OFF RELAY>REC
          LCALL OUTU8_PC
          SETB PLAYREC.2
          LCALL OUTU8_PC ;UP TELEPHONE

          MOV R1, #20
          LCALL DSEC
          MOV PLAYREC, #00H ;OFF RELAY ALL
          LCALL OUTU8_PC
          SETB EA
          RET

;***** SUB FUNC_1 *****
FUNC_1:  MOV DPTR, #TAB5
          LCALL DISPLAY
          MOV R1, #BUF_CODE+1
FUNC1_1: LCALL INPUTP1
          LCALL UBEEP2
          CJNE A, #0CH, SUB_FUN1
          MOV @R1, A
FUNC1_2: MOV CODE1, BUF_CODE+1
          MOV CODE2, BUF_CODE+2
          MOV CODE3, BUF_CODE+3
          MOV CODE4, BUF_CODE+4
          MOV CODE5, BUF_CODE+5
          MOV CODE6, BUF_CODE+6
          MOV CODE7, BUF_CODE+7
          MOV CODE8, BUF_CODE+8
          MOV CODE9, BUF_CODE+9
          LCALL UBEEP1
          RET
SUB_FUN1: MOV @R1, A
          INC R1
          CJNE R1, #BUF_CODE+9, FUNC1_1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    @R1,#0CH
LJMP  FUNC1_2

;***** SUB FUNC_2 *****
FUNC_2: MOV    DPTR,#TAB6
        LCALL DISPLAY
        LCALL INPUTP1
FUN2_1: CJNE  A,#01H,FUN2_2
        MOV    A,#10
        LJMP  SHIFT
FUN2_2: CJNE  A,#02H,FUN2_3
        MOV    A,#20
        LJMP  SHIFT
FUN2_3: CJNE  A,#0AH,SUB_FUN2
        CLR    A
SHIFT:  MOV    BUF_COUNT,A
        LCALL UBEEP2
        LCALL INPUTP2
        CJNE  A,#00H,$+6    ;0
        LJMP  SUB_ADD
        CJNE  A,#01H,$+6    ;1
        LJMP  SUB_ADD
        CJNE  A,#02H,$+6    ;2
        LJMP  SUB_ADD
        CJNE  A,#03H,$+6    ;3
        LJMP  SUB_ADD
        CJNE  A,#04H,$+6    ;4
        LJMP  SUB_ADD
        CJNE  A,#05H,$+6    ;5
        LJMP  SUB_ADD
        CJNE  A,#06H,$+6    ;6
        LJMP  SUB_ADD
        CJNE  A,#07H,$+6    ;7
        LJMP  SUB_ADD
        CJNE  A,#08H,$+6    ;8
        LJMP  SUB_ADD
        CJNE  A,#09H,SUB_FUN2 ;9
SUB_ADD: ADD    A,BUF_COUNT
        MOV    COUNT,A
        LCALL UBEEP1
        RET
SUB_FUN2: LCALL UBEEP3
        RET

;***** SUB FUNC3 *****
FUNC_3: MOV    DPTR,#TAB7
        LCALL DISPLAY
        LCALL INPUTP1
LDTMF:  MOV    BUF_DTMF,A
        LCALL UBEEP2
        LCALL SUB_FUN3
        MOV    A,BUF_DTMF
        DEC    A
        JNZ  LDTMF
        RET

SUB_FUN3: MOV    A,#05H
LOOPF3: LCALL DELAY
        DEC    A
        JNZ  LOOPF3
        RET

;***** SUB FUNC 4 *****
FUNC_4: MOV    DPTR,#TAB8
        LCALL DISPLAY
        LCALL INPUTP2
        ANL   A,#03H
        CJNE  A,#00H,OUTFUNC4
        MOV    A,#01H
OUTFUNC4: MOV    L_SOUND,A
        LCALL UBEEP1
        RET

;***** SUB FUNC 5 *****
FUNC_5: MOV    DPTR,#TAB1
        LCALL DISPLAY
        LCALL RDTIME
        LCALL DECODE
        MOV    R1,#BUF_TIM
        LCALL RUN_8R1

```

```

SFUNC_5: MOV R1,#BUF_TIM
MOV RUN_STEP,#0CAH
MOV @R1,#'[' ;[ = 5BH
LCALL STEP_RX
LCALL ON_CUR

LCALL INPUTP2 ;HOUR1 = ?
ANL A,#03H
MOV R4,A
LCALL HTOA1
MOV @R1,A
MOV RUN_STEP,#0CBH
LCALL STEP_RX

LCALL INPUTP2 ;HOUR2 = ?
MOV R5,A
LCALL HTOA1
MOV @R1,A
MOV RUN_STEP,#0CCH
LCALL STEP_RX
MOV A,R4
SWAP A
ORL A,R5
MOV TIM_HOUR,A ;
LCALL SHIFT_C

LCALL INPUTP2 ;MIN1 = ?
MOV R4,A
LCALL HTOA1
MOV @R1,A
MOV RUN_STEP,#0CEH
LCALL STEP_RX

LCALL INPUTP2 ;MIN2 = ?
MOV R5,A
LCALL HTOA1
MOV @R1,A
MOV RUN_STEP,#0CFH
LCALL STEP_RX
MOV A,R4
SWAP A
ORL A,R5
MOV TIM_MIN,A ;
LCALL SHIFT_C

LCALL INPUTP2 ;SEC1 = ?
MOV R4,A
LCALL HTOA1
MOV @R1,A
MOV RUN_STEP,#0D1H
LCALL STEP_RX

LCALL INPUTP2 ;SEC2 = ?
MOV R5,A
LCALL HTOA1
MOV @R1,A
MOV RUN_STEP,#0D2H
LCALL STEP_RX
MOV A,R4
SWAP A
ORL A,R5
MOV TIM_SEC,A ;
LCALL SHIFT_C
LCALL C_CURSOR
XXX: LCALL INPUTP1
CJNE A,#0CH,XXX
LCALL S_TIME
LCALL UBEEP1
RET

```

```

;***** SUB FUNC_6 *****

```

```

FUNC_6: MOV DPTR,#TAB2
LCALL DISPLAY
LCALL RDDATE
LCALL DECODE2
MOV R1,#BUF_DATE
LCALL RUN_8R1

```

```

SFUNC_6: MOV R1,#BUF_DATE
MOV RUN_STEP,#0CAH
MOV @R1,#'[' ;[ = 5BH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL STEP_RX
LCALL ON_CUR

LCALL INPUTP2      ;DAY1 = ?
ANL  A,#03H
MOV  R4,A
LCALL HTOA1
MOV  @R1,A
MOV  RUN_STEP,#0CBH
LCALL STEP_RX

LCALL INPUTP2      ;DAY2 = ?
MOV  R5,A
LCALL HTOA1
MOV  @R1,A
MOV  RUN_STEP,#0CCH
LCALL STEP_RX
MOV  A,R4
SWAP A
ORL  A,R5
MOV  DATE_D,A ;
LCALL SHIFT_C

LCALL INPUTP2      ;MONTH1 = ?
MOV  R4,A
LCALL HTOA1
MOV  @R1,A
MOV  RUN_STEP,#0CEH
LCALL STEP_RX

LCALL INPUTP2      ;MONTH2 = ?
MOV  R5,A
LCALL HTOA1
MOV  @R1,A
MOV  RUN_STEP,#0CFH
LCALL STEP_RX
MOV  A,R4
SWAP A
ORL  A,R5
MOV  DATE_M,A ;
LCALL SHIFT_C

LCALL INPUTP2      ;YEAR1 = ?
MOV  R4,A
LCALL HTOA1
MOV  @R1,A
MOV  RUN_STEP,#0D1H
LCALL STEP_RX

LCALL INPUTP2      ;YEAR2 = ?
MOV  R5,A
LCALL HTOA1
MOV  @R1,A
MOV  RUN_STEP,#0D2H
LCALL STEP_RX
MOV  A,R4
SWAP A
ORL  A,R5
MOV  DATE_Y,A ;
LCALL SHIFT_C
LCALL C_CURSOR
YYY:  LCALL INPUTP1
      CJNE  A,#0CH,YYY
      LCALL S_DATE
      LCALL UBEEP1
      RET

;***** SUB CHO ON *****
CHO_ON:  MOV  DPTR,#TAB10
         LCALL DISPLAY
         MOV  R1,#BUF_TIM
         MOV  RUN_STEP,#0C4H
         MOV  A,TCHXX
         LCALL HTOA1
         MOV  @R1,A
         LCALL STEP_RX

SCHO_XX: MOV  A,TCHXX
         SWAP A
         MOV  P2,#HIGH DATA_CHOO

```

```

MOV R0,A
MOVX A,@R0
LCALL HTOA3 ;1
MOV BUF_TIM,R2
MOV BUF_TIM+1,R3
INC R0
MOVX A,@R0 ;2
LCALL HTOA3
MOV BUF_TIM+3,R2
MOV BUF_TIM+4,R3
MOV BUF_TIM+2,#': '
MOV R1,#BUF_TIM
LCALL RUN_5R2
MOV R2,#T_DELAYS
LCALL DELAYS
LCALL UBEEP1
RET

;***** SUB CHO OFF *****
CHO_OFF: MOV DPTR,#TAB11
LCALL DISPLAY
MOV R1,#BUF_TIM
MOV RUN_STEP,#0C4H
MOV A,TCHXX
LCALL HTOA1
MOV @R1,A
LCALL STEP_RX

SCHO_YY: MOV A,TCHXX
SWAP A
ADD A,#02H
MOV P2,#HIGH_DATA_CH00
MOV R0,A
MOVX A,@R0
LCALL HTOA3 ;1
MOV BUF_TIM,R2
MOV BUF_TIM+1,R3
INC R0
MOVX A,@R0 ;2
LCALL HTOA3
MOV BUF_TIM+3,R2
MOV BUF_TIM+4,R3
MOV BUF_TIM+2,#': '
MOV R1,#BUF_TIM
LCALL RUN_5R2
MOV R2,#T_DELAYS
LCALL DELAYS
LCALL UBEEP3
RET

;***** SUB CH09_ON *****
CH09_ON: MOV DPTR,#TAB14
LCALL DISPLAY
MOV R1,#BUF_TIM
MOV RUN_STEP,#0C4H
MOV A,TCHXX
LCALL HTOA1
MOV @R1,A
LCALL STEP_RX
MOV R2,#T_DELAYS
LCALL DELAYS
LCALL UBEEP1
RET

;***** SUB CH09_ON *****
CH09_OFF: MOV DPTR,#TAB15
LCALL DISPLAY
MOV R1,#BUF_TIM
MOV RUN_STEP,#0C4H
MOV A,TCHXX
LCALL HTOA1
MOV @R1,A
LCALL STEP_RX
MOV R2,#T_DELAYS
LCALL DELAYS
LCALL UBEEP3
RET

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์;***** SUB CH1 OFF ***** ท่านนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 CH1_ON: MOV DPTR,#TAB12
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DISPLAY
MOV R1,#BUF_TIM
MOV RUN_STEP,#0C4H
MOV A,TCHXX
LCALL HTOA1
MOV @R1,A
LCALL STEP_RX
MOV R2,#T_DELAYS
LCALL DELAYS
LCALL UBEEP1
RET

;***** SUB CHI OFF *****
CHI_OFF: MOV DPTR,#TAB13
LCALL DISPLAY
MOV R1,#BUF_TIM
MOV RUN_STEP,#0C4H
MOV A,TCHXX
LCALL HTOA1
MOV @R1,A
LCALL STEP_RX
MOV R2,#T_DELAYS
LCALL DELAYS
LCALL UBEEP3
RET

;***** SUB XX TIME *****
XX_TIME: MOV DPTR,#TAB4
LCALL DISPLAY
MOV R1,#BUF_TIM
MOV RUN_STEP,#84H
MOV A,TCHXX
LCALL HTOA1
MOV @R1,A
LCALL STEP_RX

MOV A,TCHZZ
MOV P2,#HIGH_DATA_CHX1
MOV R0,A
MOVX A,@R0
LCALL DIS_TIM1
LCALL D_TIME1
RET

;***** SUB DIS TIME *****
DIS_TIME: MOV DPTR,#TAB4
LCALL DISPLAY
MOV R1,#BUF_TIM
MOV RUN_STEP,#84H
MOV A,TCHXX
LCALL HTOA1
MOV @R1,A
LCALL STEP_RX

MOV A,TCHXX
SWAP A
MOV P2,#HIGH_DATA_CH00
MOV R0,A
MOVX A,@R0

DIS_TIM1: LCALL HTOA3 ;1
MOV BUF_TIM,R2
MOV BUF_TIM+1,R3
INC R0
MOVX A,@R0 ;2
LCALL HTOA3
MOV BUF_TIM+3,R2
MOV BUF_TIM+4,R3
INC R0
MOVX A,@R0 ;3
LCALL HTOA3
MOV BUF_TIM+5,R2
MOV BUF_TIM+6,R3
INC R0
MOVX A,@R0 ;4
LCALL HTOA3
MOV BUF_TIM+8,R2
MOV BUF_TIM+9,R3
MOVX A,@R0 ;5
LCALL HTOA3
MOV BUF_TIM+7,R3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกองช่างเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV R1,#BUF_TIM
LCALL RUN_5R1
MOV R1,#BUF_TIM+5
LCALL RUN_5R2
RET

;***** SUB D_TIME *****
D_TIME: LCALL DELAY
        LCALL DIS_TIME

D_TIME1: MOV R1,#BUF_TIM
        MOV RUN_STEP,#8CH
        MOV @R1,'#[' ;[ = 5BH
        LCALL STEP_RX
        LCALL ON_CUR

        LCALL INPUTP3 ;HOUR1 = ?
        CJNE A,#0CH,HO_L
        LJMP OUT_LL
HO_L:   MOV R4,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#8DH
        LCALL STEP_RX

        LCALL INPUTP2 ;HOUR2 = ?
        MOV R5,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#8EH
        LCALL STEP_RX
        MOV A,R4
        SWAP A
        ORL A,R5
        MOV B_HONOX,A ;
        LCALL SHIFT_C

        LCALL INPUTP2 ;MIN1 = ?
        MOV R4,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#90H
        LCALL STEP_RX

        LCALL INPUTP2 ;MIN2 = ?
        MOV R5,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#91H
        LCALL STEP_RX
        MOV A,R4
        SWAP A
        ORL A,R5
        MOV B_MONOX,A ;
        LCALL SHIFT_C

MOV R1,#BUF_DATE
MOV RUN_STEP,#0CCH
MOV @R1,#5BH ;[ = 5BH
LCALL STEP_RX

        LCALL INPUTP2 ;HOUR1 = ?
        MOV R4,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#0CDH
        LCALL STEP_RX

        LCALL INPUTP2 ;HOUR2 = ?
        MOV R5,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#0CEH
        LCALL STEP_RX
        MOV A,R4
        SWAP A
        ORL A,R5
        MOV B_HOFFOX,A ;
        LCALL SHIFT_C

```

```

LCALL INPUTP2      ;MIN1 = ?
MOV  R4,A
LCALL HTOA2
MOV  @R1,A
MOV  RUN_STEP,#0DOH
LCALL STEP_RX

LCALL INPUTP2      ;MIN2 = ?
MOV  R5,A
LCALL HTOA2
MOV  @R1,A
MOV  RUN_STEP,#0DIH
LCALL STEP_RX
MOV  A,R4
SWAP A
ORL  A,R5
MOV  B MOFFOX,A ;
LCALL SHIFT_C
LCALL C_CURSOR
RET

OUT_LL:  MOV  A,TCHXX
        SWAP A
        MOV  P2,#HIGH DATA_CH00
        MOV  R0,A
        MOVX A,@R0      ;1
        MOV  B HONOX,A
        INC  R0
        MOVX A,@R0      ;2
        MOV  B MONOX,A
        INC  R0
        MOVX A,@R0      ;3
        MOV  B_HOFFOX,A
        INC  R0
        MOVX A,@R0      ;4
        MOV  B MOFFOX,A
        LCALL C_CURSOR
        LCALL UBEEP2
        RET

;***** BACK TIME CH 1 *****
BTTT1:  LCALL INPUTP1
        CJNE A,#0CH,BTTT1
        MOV  DPTR,#DATA_CH01
        MOV  A,B HONOX
        MOVX @DPTR,A
        MOV  DPTR,#DATA_CH01+1
        MOV  A,B MONOX
        MOVX @DPTR,A
        MOV  DPTR,#DATA_CH01+2
        MOV  A,B_HOFFOX
        MOVX @DPTR,A
        MOV  DPTR,#DATA_CH01+3
        MOV  A,B MOFFOX
        MOVX @DPTR,A
        LCALL UBEEP1
        RET

;***** BACK TIME CHX 1 *****
BXXX1:  LCALL INPUTP1
        CJNE A,#0CH,BXXX1
        MOV  DPTR,#DATA_CHX1
        MOV  A,B HONOX
        MOVX @DPTR,A
        MOV  DPTR,#DATA_CHX1+1
        MOV  A,B MONOX
        MOVX @DPTR,A
        MOV  DPTR,#DATA_CHX1+2
        MOV  A,B_HOFFOX
        MOVX @DPTR,A
        MOV  DPTR,#DATA_CHX1+3
        MOV  A,B MOFFOX
        MOVX @DPTR,A
        LCALL UBEEP1
        RET

;***** BACK TIME CHY 1 *****
BYYY1:  LCALL INPUTP1
        CJNE A,#0CH,BYYY1
        MOV  DPTR,#DATA_CHY1

```

```

MOV    A,B_HONOX
MOVX   @DPTR,A
MOV    DPTR,#DATA_CHY1+1
MOV    A,B_MONOX
MOVX   @DPTR,A
MOV    DPTR,#DATA_CHY1+2
MOV    A,B_HOFFOX
MOVX   @DPTR,A
MOV    DPTR,#DATA_CHY1+3
MOV    A,B_MOFFOX
MOVX   @DPTR,A
LCALL  UBEEP1
RET

;***** BACK TIME CHZ 1 *****
BZZZ1: LCALL  INPUTP1
        CJNE  A,#0CH,BZZZ1
        MOV   DPTR,#DATA_CHZ1
        MOV   A,B_HONOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHZ1+1
        MOV   A,B_MONOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHZ1+2
        MOV   A,B_HOFFOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHZ1+3
        MOV   A,B_MOFFOX
        MOVX  @DPTR,A
        LCALL UBEEP1
        RET

;***** BACK TIME CH2 *****
BTTT2: LCALL  INPUTP1
        CJNE  A,#0CH,BTTT2
        MOV   DPTR,#DATA_CHO2
        MOV   A,B_HONOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHO2+1
        MOV   A,B_MONOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHO2+2
        MOV   A,B_HOFFOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHO2+3
        MOV   A,B_MOFFOX
        MOVX  @DPTR,A
        LCALL UBEEP1
        RET

;***** BACK TIME CH3 *****
BTTT3: LCALL  INPUTP1
        CJNE  A,#0CH,BTTT3
        MOV   DPTR,#DATA_CHO3
        MOV   A,B_HONOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHO3+1
        MOV   A,B_MONOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHO3+2
        MOV   A,B_HOFFOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHO3+3
        MOV   A,B_MOFFOX
        MOVX  @DPTR,A
        LCALL UBEEP1
        RET

;***** BACK TIME CH4 *****
BTTT4: LCALL  INPUTP1
        CJNE  A,#0CH,BTTT4
        MOV   DPTR,#DATA_CHO4
        MOV   A,B_HONOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHO4+1
        MOV   A,B_MONOX
        MOVX  @DPTR,A
        MOV   DPTR,#DATA_CHO4+2
        MOV   A,B_HOFFOX
        MOVX  @DPTR,A

```

```

MOV DPTR,#DATA_CH04+3
MOV A,B_MOFFOX
MOVX @DPTR,A
LCALL UBEEP1
RET

```

```

;***** BACK TIME CH5 *****
BTTT5: LCALL INPUTP1
CJNE A,#0CH,BTTT5
MOV DPTR,#DATA_CH05
MOV A,B_HONOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH05+1
MOV A,B_MONOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH05+2
MOV A,B_HOFFOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH05+3
MOV A,B_MOFFOX
MOVX @DPTR,A
LCALL UBEEP1
RET

```

```

;***** BACK TIME CH6 *****
BTTT6: LCALL INPUTP1
CJNE A,#0CH,BTTT6
MOV DPTR,#DATA_CH06
MOV A,B_HONOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH06+1
MOV A,B_MONOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH06+2
MOV A,B_HOFFOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH06+3
MOV A,B_MOFFOX
MOVX @DPTR,A
LCALL UBEEP1
RET

```

```

;***** BACK TIME CH7 *****
BTTT7: LCALL INPUTP1
CJNE A,#0CH,BTTT7
MOV DPTR,#DATA_CH07
MOV A,B_HONOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH07+1
MOV A,B_MONOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH07+2
MOV A,B_HOFFOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH07+3
MOV A,B_MOFFOX
MOVX @DPTR,A
LCALL UBEEP1
RET

```

```

;***** BACK TIME CH8 *****
BTTT8: LCALL INPUTP1
CJNE A,#0CH,BTTT8
MOV DPTR,#DATA_CH08
MOV A,B_HONOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH08+1
MOV A,B_MONOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH08+2
MOV A,B_HOFFOX
MOVX @DPTR,A
MOV DPTR,#DATA_CH08+3
MOV A,B_MOFFOX
MOVX @DPTR,A
LCALL UBEEP1
RET

```

```

;***** SUB SCAN TIME *****
SCANTIM: PUSH A DPL

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                PUSH   DPH
                PUSH   ACC
SCAN1:         MOV     HOUR_ON,TIM_HOUR ;CH 01 TT
                MOV     DPTR,#DATA_CH01
                MOVX   A,@DPTR
                CJNE   A,HOUR_ON,S_1OFF ;
                MOV     MIN_ON,TIM_MIN
                MOV     DPTR,#DATA_CH01+1
                MOVX   A,@DPTR
                CJNE   A,MIN_ON,S_1OFF ;
                MOV     A,TIM_SEC
                CJNE   A,#00H,S_1OFF ;
                MOV     A,OUTP
                ORL    A,#01H
                MOV     OUTP,A
                LCALL  OUTPUT_PA
                ;LCALL UBEEP1

```

```

S_1OFF:        MOV     HOUR_OFF,TIM_HOUR
                MOV     DPTR,#DATA_CH01+2
                MOVX   A,@DPTR
S_1OFF1:       CJNE   A,HOUR_OFF,SCANX1 ;
                MOV     MIN_OFF,TIM_MIN
                MOV     DPTR,#DATA_CH01+3
                MOVX   A,@DPTR
                CJNE   A,MIN_OFF,SCANX1 ;
                MOV     A,TIM_SEC
                CJNE   A,#00H,SCANX1 ;
                MOV     A,OUTP
                ANL    A,#0FEH
                MOV     OUTP,A
                LCALL  OUTPUT_PA
                ;LCALL UBEEP3

```

```

SCANX1:        MOV     HOUR_ON,TIM_HOUR ;CH 01 XX
                MOV     DPTR,#DATA_CHX1
                MOVX   A,@DPTR
                CJNE   A,HOUR_ON,S_X1OFF ;
                MOV     MIN_ON,TIM_MIN
                MOV     DPTR,#DATA_CHX1+1
                MOVX   A,@DPTR
                CJNE   A,MIN_ON,S_X1OFF ;
                MOV     A,TIM_SEC
                CJNE   A,#00H,S_X1OFF ;
                MOV     A,OUTP
                ORL    A,#01H
                MOV     OUTP,A
                LCALL  OUTPUT_PA
                ;LCALL UBEEP1

```

```

S_X1OFF:       MOV     HOUR_OFF,TIM_HOUR
                MOV     DPTR,#DATA_CHX1+2
                MOVX   A,@DPTR
S_X1OFF1:      CJNE   A,HOUR_OFF,SCANY1 ;
                MOV     MIN_OFF,TIM_MIN
                MOV     DPTR,#DATA_CHX1+3
                MOVX   A,@DPTR
                CJNE   A,MIN_OFF,SCANY1 ;
                MOV     A,TIM_SEC
                CJNE   A,#00H,SCANY1 ;
                MOV     A,OUTP
                ANL    A,#0FEH
                MOV     OUTP,A
                LCALL  OUTPUT_PA
                ;LCALL UBEEP3

```

```

SCANY1:        MOV     HOUR_ON,TIM_HOUR ;CH 01 YY
                MOV     DPTR,#DATA_CHY1
                MOVX   A,@DPTR
                CJNE   A,HOUR_ON,S_Y1OFF ;
                MOV     MIN_ON,TIM_MIN
                MOV     DPTR,#DATA_CHY1+1
                MOVX   A,@DPTR
                CJNE   A,MIN_ON,S_Y1OFF ;
                MOV     A,TIM_SEC
                CJNE   A,#00H,S_Y1OFF ;
                MOV     A,OUTP
                ORL    A,#01H
                MOV     OUTP,A

```

```

        LCALL OUTPUT_PA
;LCALL UBEEP1

S_Y1OFF: MOV  HOUR_OFF,TIM_HOUR
        MOV  DPTR,#DATA_CHY1+2
        MOVX A,@DPTR
S_Y1OFF1: CJNE A,HOUR_OFF,SCANZ1 ;
        MOV  MIN_OFF,TIM_MIN
        MOV  DPTR,#DATA_CHY1+3
        MOVX A,@DPTR
        CJNE A,MIN_OFF,SCANZ1 ;
        MOV  A,TIM_SEC
        CJNE A,#00H,SCANZ1 ;
        MOV  A,OUTP
        ANL  A,#0FEH
        MOV  OUTP,A
        LCALL OUTPUT_PA
;LCALL UBEEP3

SCANZ1:  MOV  HOUR_ON,TIM_HOUR ;CH 01 ZZ
        MOV  DPTR,#DATA_CHZ1
        MOVX A,@DPTR
        CJNE A,HOUR_ON,S_Z1OFF ;
        MOV  MIN_ON,TIM_MIN
        MOV  DPTR,#DATA_CHZ1+1
        MOVX A,@DPTR
        CJNE A,MIN_ON,S_Z1OFF ;
        MOV  A,TIM_SEC
        CJNE A,#00H,S_Z1OFF ;
        MOV  A,OUTP
        ORL  A,#01H
        MOV  OUTP,A
        LCALL OUTPUT_PA
;LCALL UBEEP1

S_Z1OFF: MOV  HOUR_OFF,TIM_HOUR
        MOV  DPTR,#DATA_CHZ1+2
        MOVX A,@DPTR
S_Z1OFF1: CJNE A,HOUR_OFF,SCAN2 ;
        MOV  MIN_OFF,TIM_MIN
        MOV  DPTR,#DATA_CHZ1+3
        MOVX A,@DPTR
        CJNE A,MIN_OFF,SCAN2 ;
        MOV  A,TIM_SEC
        CJNE A,#00H,SCAN2 ;
        MOV  A,OUTP
        ANL  A,#0FEH
        MOV  OUTP,A
        LCALL OUTPUT_PA
;LCALL UBEEP3

SCAN2:  MOV  HOUR_ON,TIM_HOUR ;CH 02
        MOV  DPTR,#DATA_CH02
        MOVX A,@DPTR
        CJNE A,HOUR_ON,S_2OFF
        MOV  MIN_ON,TIM_MIN
        MOV  DPTR,#DATA_CH02+1
        MOVX A,@DPTR
        CJNE A,MIN_ON,S_2OFF
        MOV  A,TIM_SEC
        CJNE A,#00H,S_2OFF
        MOV  A,OUTP
        ORL  A,#02H
        MOV  OUTP,A
        LCALL OUTPUT_PA
;LCALL UBEEP1

S_2OFF:  MOV  HOUR_OFF,TIM_HOUR
        MOV  DPTR,#DATA_CH02+2
        MOVX A,@DPTR
S_2OFF1: CJNE A,HOUR_OFF,SCAN3 ;
        MOV  MIN_OFF,TIM_MIN
        MOV  DPTR,#DATA_CH02+3
        MOVX A,@DPTR
        CJNE A,MIN_OFF,SCAN3 ;
        MOV  A,TIM_SEC
        CJNE A,#00H,SCAN3 ;
        MOV  A,OUTP
        ANL  A,#0FDH
        MOV  OUTP,A

```

```

        LCALL OUTPUT_PA
        ;LCALL UBEEP3

SCAN3:  MOV    HOUR_ON,TIM_HOUR  ;CH 03
        MOV    DPTR,#DATA_CH03
        MOVX   A,@DPTR
        CJNE  A,HOUR_ON,S_3OFF
        MOV    MIN_ON,TIM_MIN
        MOV    DPTR,#DATA_CH03+1
        MOVX   A,@DPTR
        CJNE  A,MIN_ON,S_3OFF
        MOV    A,TIM_SEC
        CJNE  A,#00H,S_3OFF
        MOV    A,OUTP
        ORL   A,#04H
        MOV    OUTP,A
        LCALL OUTPUT_PA
        ;LCALL UBEEP1

S_3OFF:  MOV    HOUR_OFF,TIM_HOUR
        MOV    DPTR,#DATA_CH03+2
        MOVX   A,@DPTR

S_3OFF1: CJNE  A,HOUR_OFF,SCAN4  ;
        MOV    MIN_OFF,TIM_MIN
        MOV    DPTR,#DATA_CH03+3
        MOVX   A,@DPTR
        CJNE  A,MIN_OFF,SCAN4  ;
        MOV    A,TIM_SEC
        CJNE  A,#00H,SCAN4     ;
        MOV    A,OUTP
        ANL   A,#0FBH
        MOV    OUTP,A
        LCALL OUTPUT_PA
        ;LCALL UBEEP3

SCAN4:  MOV    HOUR_ON,TIM_HOUR  ;CH 04
        MOV    DPTR,#DATA_CH04
        MOVX   A,@DPTR
        CJNE  A,HOUR_ON,S_4OFF
        MOV    MIN_ON,TIM_MIN
        MOV    DPTR,#DATA_CH04+1
        MOVX   A,@DPTR
        CJNE  A,MIN_ON,S_4OFF
        MOV    A,TIM_SEC
        CJNE  A,#00H,S_4OFF
        MOV    A,OUTP
        ORL   A,#08H
        MOV    OUTP,A
        LCALL OUTPUT_PA
        ;LCALL UBEEP1

S_4OFF:  MOV    HOUR_OFF,TIM_HOUR
        MOV    DPTR,#DATA_CH04+2
        MOVX   A,@DPTR

S_4OFF1: CJNE  A,HOUR_OFF,SCAN5  ;
        MOV    MIN_OFF,TIM_MIN
        MOV    DPTR,#DATA_CH04+3
        MOVX   A,@DPTR
        CJNE  A,MIN_OFF,SCAN5  ;
        MOV    A,TIM_SEC
        CJNE  A,#00H,SCAN5     ;
        MOV    A,OUTP
        ANL   A,#0F7H
        MOV    OUTP,A
        LCALL OUTPUT_PA
        ;LCALL UBEEP3

SCAN5:  MOV    HOUR_ON,TIM_HOUR  ;CH 05
        MOV    DPTR,#DATA_CH05
        MOVX   A,@DPTR
        CJNE  A,HOUR_ON,S_5OFF
        MOV    MIN_ON,TIM_MIN
        MOV    DPTR,#DATA_CH05+1
        MOVX   A,@DPTR
        CJNE  A,MIN_ON,S_5OFF
        MOV    A,TIM_SEC
        CJNE  A,#00H,S_5OFF
        MOV    A,OUTP
        ORL   A,#10H
        MOV    OUTP,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL OUTPUT_PA
        ;LCALL UBEEP1

S_5OFF:  MOV    HOUR_OFF,TIM_HOUR
        MOV    DPTR,#DATA_CH05+2
        MOVX   A,@DPTR
S_5OFF1: CJNE   A,HOUR_OFF,SCAN6  ;
        MOV    MIN_OFF,TIM_MIN
        MOV    DPTR,#DATA_CH05+3
        MOVX   A,@DPTR
        CJNE   A,MIN_OFF,SCAN6  ;
        MOV    A,TIM_SEC
        CJNE   A,#00H,SCAN6    ;
        MOV    A,OUTP
        ANL    A,#0EFH
        MOV    OUTP,A
        LCALL OUTPUT_PA
        ;LCALL UBEEP3

SCAN6:   MOV    HOUR_ON,TIM_HOUR  ;CH 06
        MOV    DPTR,#DATA_CH06
        MOVX   A,@DPTR
        CJNE   A,HOUR_ON,S_6OFF
        MOV    MIN_ON,TIM_MIN
        MOV    DPTR,#DATA_CH06+1
        MOVX   A,@DPTR
        CJNE   A,MIN_ON,S_6OFF
        MOV    A,TIM_SEC
        CJNE   A,#00H,S_6OFF
        MOV    A,OUTP
        ORL    A,#20H
        MOV    OUTP,A
        LCALL OUTPUT_PA
        ;LCALL UBEEP1

S_6OFF:  MOV    HOUR_OFF,TIM_HOUR
        MOV    DPTR,#DATA_CH06+2
        MOVX   A,@DPTR
S_6OFF1: CJNE   A,HOUR_OFF,SCAN7  ;
        MOV    MIN_OFF,TIM_MIN
        MOV    DPTR,#DATA_CH06+3
        MOVX   A,@DPTR
        CJNE   A,MIN_OFF,SCAN7  ;
        MOV    A,TIM_SEC
        CJNE   A,#00H,SCAN7    ;
        MOV    A,OUTP
        ANL    A,#0DFH
        MOV    OUTP,A
        LCALL OUTPUT_PA
        ;LCALL UBEEP3

SCAN7:   MOV    HOUR_ON,TIM_HOUR  ;CH 07
        MOV    DPTR,#DATA_CH07
        MOVX   A,@DPTR
        CJNE   A,HOUR_ON,S_7OFF
        MOV    MIN_ON,TIM_MIN
        MOV    DPTR,#DATA_CH07+1
        MOVX   A,@DPTR
        CJNE   A,MIN_ON,S_7OFF
        MOV    A,TIM_SEC
        CJNE   A,#00H,S_7OFF
        MOV    A,OUTP
        ORL    A,#40H
        MOV    OUTP,A
        LCALL OUTPUT_PA
        ;LCALL UBEEP1

S_7OFF:  MOV    HOUR_OFF,TIM_HOUR
        MOV    DPTR,#DATA_CH07+2
        MOVX   A,@DPTR
S_7OFF1: CJNE   A,HOUR_OFF,SCAN8  ;
        MOV    MIN_OFF,TIM_MIN
        MOV    DPTR,#DATA_CH07+3
        MOVX   A,@DPTR
        CJNE   A,MIN_OFF,SCAN8  ;
        MOV    A,TIM_SEC
        CJNE   A,#00H,SCAN8    ;
        MOV    A,OUTP
        ANL    A,#0BFH
        MOV    OUTP,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL OUTPUT_PA
;LCALL UBEEP3

SCAN8:  MOV    HOUR_ON,TIM_HOUR ;CH 08
        MOV    DPTR,#DATA_CH08
        MOVX   A,@DPTR
        CJNE  A,HOUR_ON,S_8OFF
        MOV    MIN_ON,TIM_MIN
        MOV    DPTR,#DATA_CH08+1
        MOVX   A,@DPTR
        CJNE  A,MIN_ON,S_8OFF
        MOV    A,TIM_SEC
        CJNE  A,#00H,S_8OFF
        MOV    A,OUTP
        ORL   A,#80H
        MOV    OUTP,A
        LCALL OUTPUT_PA
;LCALL UBEEP1

S_8OFF:  MOV    HOUR_OFF,TIM_HOUR
        MOV    DPTR,#DATA_CH08+2
        MOVX   A,@DPTR
S_8OFF1: CJNE  A,HOUR_OFF,SCAN9 ;
        MOV    MIN_OFF,TIM_MIN
        MOV    DPTR,#DATA_CH08+3
        MOVX   A,@DPTR
        CJNE  A,MIN_OFF,SCAN9 ;
        MOV    A,TIM_SEC
        CJNE  A,#00H,SCAN9 ;
        MOV    A,OUTP
        ANL   A,#07FH
        MOV    OUTP,A
        LCALL OUTPUT_PA
;LCALL UBEEP3

SCAN9:   POP    ACC
        POP    DPH
        POP    DPL
        RET

;***** SUB ON CURSOR *****
ON_CUR:  MOV    A,#00001111B
        LCALL WR_COMM
        RET

;***** SUB CLEAR CURSOR *****
C_CURSOR: MOV  A,#00001100B
        LCALL WR_COMM
        RET

;***** SUB SHIFT CURSOR *****
SHIFT_C: MOV  A,#00010100B ;S/C=0,R/L=1
        LCALL WR_COMM
        RET

;***** SUB WR_COMM *****
WR_COMM: PUSH  DPH
        PUSH  DPL
        MOV   DPTR,#COM_PORT
        MOVX  @DPTR,A
        MOV   DPTR,#RD_BUSY
BUSYC:   MOVX  A,@DPTR
        JB   ACC.7,BUSYC
        POP  DPL
        POP  DPH
        RET

;***** SUB WRITE *****
WRITE:   PUSH  DPL
        PUSH  DPH
        MOV   DPTR,#DAT_PORT
        MOVX  @DPTR,A
        MOV   DPTR,#RD_BUSY
BUSY1:  MOVX  A,@DPTR
        JB   ACC.7,BUSY1
        POP  DPH
        POP  DPL
        RET

;***** SUB DISPLAY *****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DISPLAY:  MOV  A,#80H      ;SET ADD LINE 1
          LCALL LCD_RUN
          MOV  A,#0COH    ;SET ADD LINE 2
          LCALL LCD_RUN
          RET
LCD_RUN:  LCALL WR_COMM
          MOV  R2,#20      ;20 CHAR
LCD_RUN1: CLR  A
          MOVC A,@A+DPTR
          LCALL WRITE
          INC  DPTR
          DJNZ R2,LCD_RUN1
          RET

;***** SUB STEP_RX *****
STEP_RX:  MOV  A,RUN_STEP
          LCALL RUN_T1
          RET

RUN_T1:   LCALL WR_COMM
RUN_T2:   CLR  A
          MOV  A,@R1
          LCALL WRITE
          RET

;***** SUB FOR TEL *****
FOR_TEL:  MOV  A,RUN_STEP
          LCALL RUN_TEL1
          RET

RUN_TEL1: LCALL WR_COMM
RUN_TEL2: CLR  A
          MOV  A,R3
          LCALL WRITE
          RET

;***** SUB RUN_5R1 *****
RUN_5R1:  MOV  A,#08DH
          LCALL RUN_5C1
          RET

;***** SUB RUN_5R2 *****
RUN_5R2:  MOV  A,#0CDH
          LCALL RUN_5C1
          RET

;***** SUB RUN_5C *****
RUN_5C:   MOV  A,RUN_STEP ;L1=8A,L2CA
          LCALL RUN_5C1
          RET
RUN_5C1:  LCALL WR_COMM
          MOV  R2,#5      ;5 CHAR
RUN_5C2:  CLR  A
          MOV  A,@R1
          LCALL WRITE
          INC  R1
          DJNZ R2,RUN_5C2
          RET

;***** SUB RUN_8C *****
RUN_8C:   MOV  A,RUN_STEP ;L1=8A,L2CA
          LCALL RUN_8C1
          RET
RUN_8C1:  LCALL WR_COMM
          MOV  R2,#8      ;8 CHAR
RUN_8C2:  CLR  A
          MOV  A,@R1
          LCALL WRITE
          INC  R1
          DJNZ R2,RUN_8C2
          RET

;***** SUB RUN_8R1 *****
RUN_8R1:  MOV  A,#08BH
          LCALL RUN_8C1
          RET

;***** SUB RUN_8R2 *****
RUN_8R2:  MOV  A,#0CBH
          LCALL RUN_8C1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET

;***** SUB DECODE *****
DECODE:  MOV  BUF_TIM+2,#3AH
        MOV  BUF_TIM+5,#3AH
        MOV  A,TIM_HOUR
        LCALL HTOA
        MOV  BUF_TIM,R2
        MOV_ BUF_TIM+1,R3

        MOV  A,TIM_MIN
        LCALL HTOA
        MOV  BUF_TIM+3,R2
        MOV  BUF_TIM+4,R3

        MOV  A,TIM_SEC
        LCALL HTOA
        MOV  BUF_TIM+6,R2
        MOV  BUF_TIM+7,R3
        RET

;***** SUB DECODE2 *****
DECODE2: MOV  BUF_DATE+2,#2FH
        MOV  BUF_DATE+5,#2FH

        MOV  A,DATE_D
        LCALL HTOA
        MOV  BUF_DATE,R2
        MOV  BUF_DATE+1,R3

        MOV  A,DATE_M
        LCALL HTOA
        MOV  BUF_DATE+3,R2
        MOV  BUF_DATE+4,R3

        MOV  A,DATE_Y
        LCALL HTOA
        MOV  BUF_DATE+6,R2
        MOV  BUF_DATE+7,R3
        RET

;***** RTCINT SUB *****
;INITIAL 6242B OKI RTC
RTCINT:  MOV  DPTR,#RTCPOT+0FH
        MOV  A,#0000011B ;STOP=1,REST=1
        MOVX @DPTR,A
        ORL  A,#00000100B ;24 HOUR
        MOVX @DPTR,A
        MOV  DPTR,#RTCPOT+0EH
        MOV  A,#00000100B ;SET STND 1 SEC
        MOVX @DPTR,A
        MOV  DPTR,#RTCPOT+0DH
        MOV  A,#00000000B ;HOLD = 0
        MOVX @DPTR,A

        MOV  DPTR,#RTCPOT+0FH
        MOVX A,@DPTR
        ANL  A,#00000100B ;STOP=0,REST=0
        MOVX @DPTR,A
        RET

;***** RDDATE SUB *****
;READ DATE FROM RTC
;OUT = R5,R6,R7,R1
;   DAT,MONTH,YEAR,WEEK
;REG = A,R0,R5,R6,R7,DPTR
RDDATE:  MOV  DPTR,#RTCPOT+0DH
        MOVX A,@DPTR
        SETB ACC.0 ;HOLD=1
        MOVX @DPTR,A
        MOVX A,@DPTR ;READ BUSY
        JNB  ACC.1,RDDATE1
        CLR  ACC.0 ;HOLD=0
        MOVX @DPTR,A
        SJMP RDDATE

RDDATE1: MOV  DPTR,#RTCPOT+0AH ;YEAR
        MOVX A,@DPTR
        ANL  A,#0FH
        MOV  R0,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก... การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่วารณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    DPTR,#RTCPOT+0BH
MOVX   A,@DPTR
ANL    A,#0FH
SWAP   A
ORL    A,R0
MOV    DATE_Y,A

MOV    DPTR,#RTCPOT+8 ;MONTH
MOVX   A,@DPTR
ANL    A,#0FH
MOV    R0,A
MOV    DPTR,#RTCPOT+9
MOVX   A,@DPTR
ANL    A,#01H
SWAP   A
ORL    A,R0
MOV    DATE_M,A

MOV    DPTR,#RTCPOT+6 ;DAY
MOVX   A,@DPTR
ANL    A,#0FH
MOV    R0,A
MOV    DPTR,#RTCPOT+7
MOVX   A,@DPTR
ANL    A,#03H
SWAP   A
ORL    A,R0
MOV    DATE_D,A

MOV    DPTR,#RTCPOT+0DH
MOVX   A,@DPTR
CLR    ACC.0 ;HOLD=0
MOVX  @DPTR,A
RET

;***** RDTIME SUB *****
;READ TIME FROM RTC
;OUT = R2,R3,R4
; HOUR,MINUTE,SECOND
;REG = A,R0,DPTR
RDTIME: MOV    DPTR,#RTCPOT+0DH
MOVX   A,@DPTR
SETB  ACC.0 ;HOLD=1
MOVX  @DPTR,A
MOVX  A,@DPTR ;READ BUSY
JNB   ACC.1,RDTIME1
CLR   ACC.0
MOVX  @DPTR,A
SJMP  RDTIME

RDTIME1: MOV    DPTR,#RTCPOT+0 ;SECOND
MOVX   A,@DPTR
ANL    A,#0FH
MOV    R0,A
MOV    DPTR,#RTCPOT+1
MOVX   A,@DPTR
ANL    A,#07H
SWAP   A
ORL    A,R0
MOV    TIM_SEC,A

MOV    DPTR,#RTCPOT+2 ;MINUTE
MOVX   A,@DPTR
ANL    A,#0FH
MOV    R0,A
MOV    DPTR,#RTCPOT+3
MOVX   A,@DPTR
ANL    A,#07H
SWAP   A
ORL    A,R0
MOV    TIM_MIN,A

MOV    DPTR,#RTCPOT+4 ;HOUR
MOVX   A,@DPTR
ANL    A,#0FH
MOV    R0,A
MOV    DPTR,#RTCPOT+5
MOVX   A,@DPTR
ANL    A,#03H
SWAP   A

```

```

        ORL    A,R0
        MOV    TIM_HOUR,A

        MOV    DPTR,#RTCPOT+0DH
        MOVX   A,@DPTR
        CLR    ACC.0           ;HOLD=0
        MOVX  @DPTR,A
        RET

;***** WRDATE SUB *****
;WRITE DATE TO RTC
;IN = R2,R3,R4
;  DAY,MONTH,YEAR
;REG = A,DPTR
WRDATE:  MOV    DPTR,#RTCPOT+0DH
        MOVX   A,@DPTR
        SETB  ACC.0           ;HOLD=1
        MOVX  @DPTR,A
        MOVX  A,@DPTR        ;READ BUSY
        JNB  ACC.1,WRDATE1
        CLR   ACC.0           ;HOLD=0
        MOVX  @DPTR,A
        SJMP  WRDATE

WRDATE1: MOV    DPTR,#RTCPOT+6 ;DAY
        MOV    A,R5
        ANL   A,#0FH
        MOVX  @DPTR,A
        MOV    DPTR,#RTCPOT+7
        MOV    A,R5
        ANL   A,#30H
        SWAP  A
        MOVX  @DPTR,A

        MOV    DPTR,#RTCPOT+8 ;MONTH
        MOV    A,R6
        ANL   A,#0FH
        MOVX  @DPTR,A
        MOV    DPTR,#RTCPOT+9
        MOV    A,R6
        ANL   A,#10H
        SWAP  A
        MOVX  @DPTR,A

        MOV    DPTR,#RTCPOT+0AH ;YEAR
        MOV    A,R7
        ANL   A,#0FH
        MOVX  @DPTR,A
        MOV    DPTR,#RTCPOT+0BH
        MOV    A,R7
        ANL   A,#0F0H
        SWAP  A
        MOVX  @DPTR,A

        MOV    DPTR,#RTCPOT+0DH
        MOVX   A,@DPTR
        CLR    ACC.0           ;HOLD=0
        MOVX  @DPTR,A
        RET

;***** WRTIME SUB *****
;WRITE TIME TO RTC
;IN = R2,R3,R4
;  HOUR,MINUTE,SECOND
;REG = A,DPTR

WRTIME:  MOV    DPTR,#RTCPOT+0DH
        MOVX   A,@DPTR
        SETB  ACC.0           ;HOLD=1
        MOVX  @DPTR,A
        MOVX  A,@DPTR        ;READ BUSY
        JNB  ACC.1,WRTIME1
        CLR   ACC.0           ;HOLD=0
        MOVX  @DPTR,A
        SJMP  WRTIME

WRTIME1: MOV    DPTR,#RTCPOT+0 ;SECOND
        MOV    A,R4
        ANL   A,#0FH
        MOVX  @DPTR,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก... การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV   DPTR,#RTCPOT+1
MOV   A,R4
ANL   A,#70H
SWAP  A
MOVX  @DPTR,A

MOV   DPTR,#RTCPOT+2   ;MINUTE
MOV   A,R3
ANL   A,#0FH
MOVX  @DPTR,A
MOV   DPTR,#RTCPOT+3
MOV   A,R3
ANL   A,#70H
SWAP  A
MOVX  @DPTR,A

MOV   DPTR,#RTCPOT+4   ;HOUR
MOV   A,R2
ANL   A,#0FH
MOVX  @DPTR,A
MOV   DPTR,#RTCPOT+5
MOV   A,R2
ANL   A,#70H           ;12/24 HOUR
SWAP  A
MOVX  @DPTR,A

MOV   DPTR,#RTCPOT+0DH
MOVX  A,@DPTR
CLR   ACC.0           ;HOLD=0
MOVX  @DPTR,A
RET

;***** SUB SET TIME *****
S_TIME: MOV R2,TIM_HOUR
        MOV R3,TIM_MIN
        MOV R4,TIM_SEC
        LCALL WRTIME
        RET

;***** SUB SET DATE *****
S_DATE: MOV R5,DATE_D
        MOV R6,DATE_M
        MOV R7,DATE_Y
        LCALL WRDATE
        RET

;***** SUB TIM_DATE *****
TIM_DATE: LCALL RDTIME
          LCALL DECODE
          MOV R1,#BUF_TIM
          LCALL RUN_8R1

          MOV A,TIM_HOUR
          CJNE A,#00H,TIMLOOP
          MOV A,TIM_MIN
          CJNE A,#00H,TIMLOOP
          MOV A,TIM_SEC
          CJNE A,#00H,TIMLOOP
          LCALL RDDATE
          LCALL DECODE2
          MOV R1,#BUF_DATE
          LCALL RUN_8R2

TIMLOOP: RET

;***** SUB SUBDATE *****
SUBDATE: LCALL RDDATE
          LCALL DECODE2
          MOV R1,#BUF_DATE
          LCALL RUN_8R2
          RET

;***** HTOA SUB *****
;CONVERT HEX TO ASCII
;IN = A
;OUT = R2,R3
;REG = A,R2,R3
HTOA:  PUSH ACC
        SWAP A
        LCALL HTOAS
        MOV R2,A

```

```

POP ACC
LCALL HTOAS
MOV R3,A
RET
HTOAS: ANL A,#0FH
CJNE A,#0AH,$+3
JNC HTOAS1
ORL A,#30H
RET
HTOAS1: SUBB A,#9
ORL A,#40H
RET

;***** HTOA1 SUB *****
;CONVERT HEX TO ASCII
;IN = A
;OUT = A
HTOA1: ANL A,#0FH
CJNE A,#0AH,$+3
JNC HTOA12
ORL A,#30H
MOV R3,A
RET
HTOA12: SUBB A,#9
ORL A,#40H
MOV R3,A
RET

;***** HTOA2 SUB *****
;CONVERT HEX TO ASCII
;IN = A
;OUT = A
HTOA2: ANL A,#2FH
CJNE A,#2AH,LIHTOA ;*
MOV R3,A
RET
LIHTOA: ANL A,#0FH
ORL A,#30H
RET

;***** HTOA3 SUB *****
;CONVERT HEX TO ASCII
;IN = A
;OUT = A
HTOA3: PUSH DPL
PUSH DPH
PUSH ACC
SWAP A
LCALL HTOAS3
MOV R2,A
POP ACC
LCALL HTOAS3
MOV R3,A
POP DPH
POP DPL
RET

HTOAS3: ANL A,#0FFH
CJNE A,#2AH,HTOAS31 ;*
MOV A,#' '
RET
HTOAS31: CJNE A,#0A2H,HTOAS32 ;*
MOV A,#' '
RET
HTOAS32: CJNE A,#0AAH,HTOAS33 ;*
MOV A,#' '
RET
HTOAS33: ANL A,#0FH
ORL A,#30H
RET

;***** SUB TEMP *****
TEMP: MOV DPTR,#TAB24
LCALL DISPLAY ;READ TEMP & DISPLAY
MOV R2,#10
LCALL DELAY

;***** TEMP2 *****
TEMP2: MOV R2,#0CH ;WRITE CONFIG
MOV R3,#00000011B

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DTWR8

TEMP4:  MOV R2,#0EEH ;START CONVERT
        LCALL DTWR
TEMP42: MOV R2,#0ACH ;READ CONFIG (DONE BIT)
        LCALL DTRD8
        MOV A,R3
        JNB ACC.7,TEMP42 ;DONE=0

        MOV R2,#0AAH ;READ TEMP
        LCALL DTRD16
        LCALL DTCON
        PUSH ACC
        MOV TEMP_C,R2
        MOV TEMP_C+1,R3
        POP ACC
        MOV TEMP_C+2,#00H ;0.0

        JNB ACC.0,TEMP5
        MOV TEMP_C+2,#05H ;0.5

TEMP5:  LCALL DECODET
        MOV R1,#LCD_BUF
        MOV RUN_STEP,#0C6H
        LCALL RUN_5C
        JNB INSTD,TEMP4
        MOV DPTR,#PORTA08
        MOVX A,@DPTR
        JB INSTD,$
        CJNE A,#0CH,TEMP4

REPASS: SETB EA
        LCALL UBEEP2
        RET

;***** SUB INPUT T-HIGH,LOW ****
; SET FOR DS1620 STAND-ALONE MODE
IN_TH:  MOV DPTR,#TAB22 ;HIGH
        LCALL DISPLAY
        MOV RUN_STEP,#08DH
        LCALL READTH
        MOV R1,#BUF_TIM
        MOV RUN_STEP,#0CCH
        MOV @R1,#' '
        LCALL STEP_RX
        LCALL ON_CUR

        LCALL INPUTP3 ;TEMP_H1= ?
        CJNE A,#0CH,TH_1
        LJMP OUT_TH

TH_1:  MOV INTT1,A
        MOV R4,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#0CDH
        LCALL STEP_RX

        LCALL INPUTP2 ;TEMP_H2= ?
        MOV R4,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#0CEH
        LCALL STEP_RX

        LCALL INPUTP2 ;TEMP_H3= ?
        MOV R5,A
        LCALL HTOA2
        MOV @R1,A
        MOV RUN_STEP,#0CFH
        LCALL STEP_RX
        MOV A,R4
        SWAP A
        ORL A,R5
        MOV INTT2,A ;
        LCALL SHIFT_C

        LCALL INPUTP2 ;TEMP_H4= ?
        MOV INTT3,A
        MOV R4,A
        LCALL HTOA2

```

```

MOV @R1,A
MOV RUN_STEP,#0D1H
LCALL STEP_RX

MOV R1,#0
MOV R2,INTT1
MOV R3,INTT2
LCALL DTOH
MOV INHHH,A
MOV A,INTT3
JZ CONTD
SETB C
SJMP CONTDVV
CONTD: CLR C
CONTDVV: MOV A,INHHH
RLC A
MOV R4,A
MOV R3,#0
MOV R2,#01H
LCALL DTWR16
LCALL C_CURSOR
LCALL INPUTP1
CJNE A,#0CH,$
LCALL UBEEP1
RET
OUT_TH: LCALL C_CURSOR
RET

;***** SUB IN TL *****
IN_TL: MOV DPTR,#TAB23 ;LOW
LCALL DISPLAY
MOV RUN_STEP,#08DH
LCALL READTL
MOV R1,#BUF_TIM
MOV RUN_STEP,#0CCH
MOV @R1,'# '
LCALL STEP_RX
LCALL ON_CUR

LCALL INPUTP3 ;TEMP_L1= ?
CJNE A,#0CH,TL_1
LJMP OUT_TL
TL_1: MOV INTT1,A
MOV R4,A
LCALL HTOA2
MOV @R1,A
MOV RUN_STEP,#0CDH
LCALL STEP_RX

LCALL INPUTP2 ;TEMP_L2= ?
MOV R4,A
LCALL HTOA2
MOV @R1,A
MOV RUN_STEP,#0CEH
LCALL STEP_RX

LCALL INPUTP2 ;TEMP_L3= ?
MOV R5,A
LCALL HTOA2
MOV @R1,A
MOV RUN_STEP,#0CFH
LCALL STEP_RX
MOV A,R4
SWAP A
ORL A,R5
MOV INTT2,A ;
LCALL SHIFT_C

LCALL INPUTP2 ;TEMP_L4= ?
MOV INTT3,A
MOV R4,A ;0.0,0.5
LCALL HTOA2
MOV @R1,A
MOV RUN_STEP,#0D1H
LCALL STEP_RX

```

```

MOV R1,#0
MOV R2,INTT1
MOV R3,INTT2
LCALL DTOH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน R3,INTT2 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV INLL,A
MOV A,INT3
JZ CONDDL
SETB C
SJMP LLCONTD
CONDDL: CLR C
LLCONTD: MOV A,INLL
RLC A
MOV R4,A
MOV R3,#0
MOV R2,#02H
LCALL DTWR16
LCALL C_CURSOR
LCALL INPUP1
CJNE A,#0CH,$
LCALL UBEEP1
RET

OUT_TL: LCALL C_CURSOR
RET

;***** SUB RE_T *****
RE_T: MOV R2,#0CH ;WRITE CONFIG
MOV R3,#0000000B
LCALL DTWR8
LCALL UBEEP1
RET

;***** SUB FUNC_7 *****
FUNC_7: MOV DPTR,#TAB25
LCALL DISPLAY
LCALL INPUT1
TFUNC1: CJNE A,#01H,TFUNC2
LCALL RTHL
SJMP FUNC_7
TFUNC2: CJNE A,#02H,TFUNC3
LCALL IN_TH
SJMP FUNC_7
TFUNC3: CJNE A,#03H,ENDTFUNC
LCALL IN_TL
SJMP FUNC_7
ENDTFUNC: CJNE A,#0CH,ERRORT
LCALL RE_T
LJMP PASS
ERRORT: LCALL ERROR
LJMP PASS

;***** READ TEMP HIGH.LOW *****
RTHL: CLR EA
MOV DPTR,#TAB21
LCALL DISPLAY
MOV RUN_STEP,#8DH
LCALL READTH
MOV RUN_STEP,#0CDH
LCALL READTL
LCALL INPUT1
CJNE A,#0CH,$
LCALL UBEEP2
SETB EA
RET

;***** SUB READTH *****
; READ & DISPLAY TH,TL ON DS1620
READTH: MOV R2,#0A1H ;READ TH
LCALL DTRD16
LCALL DTCN
PUSH ACC
MOV TEMP_TH,R2
MOV TEMP_TH+1,R3
POP ACC
MOV TEMP_TH+2,#00H ;H.
JNB ACC.0,RHL1
MOV TEMP_TH+2,#05H
RHL1: LCALL DECODETH
MOV R1,#LCD_BUF
LCALL RUN_5C
RET

;***** SUB READTL *****
READTL: MOV R2,#0A2H ;READ TL

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DTRD16
LCALL DTCON
PUSH ACC
MOV TEMP_TL,R2
MOV TEMP_TL+1,R3
POP ACC
MOV TEMP_TL+2,#00H ;L.
JNB ACC.0,RHL2
MOV TEMP_TL+2,#05H
RHL2: LCALL DECODETL
MOV R1,#LCD_BUF
LCALL RUN_SC
RET

```

```

;***** SUB DECODE TEMP *****

```

```

DECODET: MOV A,TEMP_C
LCALL HTOA
MOV LCD_BUF,R3
MOV A,TEMP_C+1
LCALL HTOA
MOV LCD_BUF+1,R2
MOV LCD_BUF+2,R3
MOV A,TEMP_C+2
LCALL HTOA
MOV LCD_BUF+4,R3
MOV LCD_BUF+3,#'.'
RET

```

```

;**** SUB DECODE TEMP HIGH *****

```

```

DECODETH: MOV A,TEMP_TH
LCALL HTOA
MOV LCD_BUF,R3
MOV A,TEMP_TH+1
LCALL HTOA
MOV LCD_BUF+1,R2
MOV LCD_BUF+2,R3
MOV A,TEMP_TH+2
LCALL HTOA
MOV LCD_BUF+4,R3
MOV LCD_BUF+3,#'.'
RET

```

```

;**** SUB DECODE TEMP LOW *****

```

```

DECODETL: MOV A,TEMP_TL
LCALL HTOA
MOV LCD_BUF,R3
MOV A,TEMP_TL+1
LCALL HTOA
MOV LCD_BUF+1,R2
MOV LCD_BUF+2,R3
MOV A,TEMP_TL+2
LCALL HTOA
MOV LCD_BUF+4,R3
MOV LCD_BUF+3,#'.'
RET

```

```

;***** SUB FUNC_9 *****

```

```

TESTSOUND:
MOV DPTR,#TAB26
LCALL DISPLAY
LCALL UBEEP2
CLR EA
SETB PLAYREC.1
LCALL OUT8_PC
LCALL INPUTF1
TSOUND1: CJNE A,#01H,TSOUND2
LCALL UBEEP2
LCALL SOUNDCHA
SETB EA
SJMP TESTSOUND
TSOUND2: CJNE A,#02H,TSOUND3
LCALL UBEEP2
LCALL SOUNDCHB
SETB EA
SJMP TESTSOUND
TSOUND3: CJNE A,#03H,ENDSOUND
LCALL UBEEP2
LCALL SOUNDCHC
SETB EA
SJMP TESTSOUND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ENDSOUND: CJNE  A, #0CH, ERRORS
            CLR   PLAYREC.1
            LCALL OUTU8_PC
            SETB  EA
            MOV   IE, #81H
            LJMP  PASS
ERRORS:    LCALL  ERROR
            CLR   PLAYREC.1
            LCALL OUTU8_PC
            SETB  EA
            MOV   IE, #81H
            LJMP  PASS

;**** SUB SOUND CHANNAL1 *****
SOUNDCHA: LCALL  UBEEP2
            MOV   DPTR, #TAB27
            LCALL DISPLAY
            LCALL DELAY

            CLR   CON_PLAY.3
            LCALL OUTPUT_PC
            MOV   R1, #20
            LCALL DSEC

            SETB  CON_PLAY.3
            LCALL OUTPUT_PC
            RET

;***** SUB SOUND CHANNAL2 *****
SOUNDCHB: LCALL  UBEEP2
            MOV   DPTR, #TAB28
            LCALL DISPLAY
            LCALL DELAY

            CLR   CON_PLAY.0
            LCALL OUTPUT_PC
            MOV   R1, #20
            LCALL DSEC

            SETB  CON_PLAY.0
            LCALL OUTPUT_PC
            RET

;***** SUB SOUND CHANNAL3 *****
SOUNDCHC: LCALL  UBEEP2
            MOV   DPTR, #TAB29
            LCALL DISPLAY
            LCALL DELAY

            CLR   CON_PLAY.1
            LCALL OUTPUT_PC
            MOV   R1, #20
            LCALL DSEC

            SETB  CON_PLAY.1
            LCALL OUTPUT_PC
            RET

;***** LOAD SEK CHARACTER *****
LSCHAR:   MOV   A, #01000000B      ;CGRAM ADD.
            LCALL WR_COMM
            MOV   R7, #8*1        ;LOAD CHAR.
            MOV   DPTR, #MAINTD

LSCHAR1:  CLR   A
            MOVC  A, @A+DPTR
            PUSH  DPH
            PUSH  DPL
            LCALL WRITE
            POP   DPL
            POP   DPH
            INC  DPTR
            DJNZ R7, LSCHAR1
            RET

; ***** DTCON SUB *****
; CONVERT TO TEMPERATURE DATA
; IN = R3,R4 (9 BIT DATA)
; OUT = R2,R3,A (16 BIT DECIMAL DATA) (A 0=0.0 1=0.5)
; REG = ALL

```

```

DTCON:  MOV    A,R3
        JNB    ACC.0,DTCON2
        RET    ;??? (-)
DTCON2: MOV    R7,#0    ;0.0 (+)
        CLR    C
        MOV    A,R4
        RRC    A
        JNC    DTCON3
        MOV    R7,#1    ;0.5
DTCON3: MOV    DPL,A
        MOV    DPH,#0
        LCALL HTOD
        MOV    A,R7
        RET

```

```
; ***** DTWR,DTWR8,DTWR16 SUB *****
```

```
; WRITE COMMAND,DATA TO DS1620
```

```
; IN = R2  COMMAND
```

```
; R3,R4 DATA
```

```
; REG = A,R2,R3,R4,R5,R6
```

```
DTWR:   CLR    TMCLCK    ;CLK=0
        LCALL DTDEL
        SETB  TMRSTT    ;RST=1
        LCALL DTDEL
        MOV   A,R2      ;COMMAND
        LCALL DTWRS
        CLR   TMRSTT    ;RST=0
        LCALL DTDEL
        RET

```

```
DTWR8:  CLR    TMCLCK    ;CLK=0
        LCALL DTDEL
        SETB  TMRSTT    ;RST=1
        LCALL DTDEL
        MOV   A,R2      ;COMMAND
        LCALL DTWRS
        MOV   A,R3      ;DATA
        LCALL DTWRS
        CLR   TMRSTT    ;RST=0
        LCALL DTDEL
        LCALL DTDELE
        RET

```

```
DTWR16: CLR    TMCLCK    ;CLK=0
        LCALL DTDEL
        SETB  TMRSTT    ;RST=1
        LCALL DTDEL
        MOV   A,R2      ;COMMAND
        LCALL DTWRS
        MOV   A,R4      ;DATA
        LCALL DTWRS
        MOV   A,R3
        LCALL DTWRS
        CLR   TMRSTT    ;RST=0
        LCALL DTDEL
        LCALL DTDELE
        RET

```

```
DTWRS:  MOV    R5,#8 ;WRITE COMMAND/DATA
DTWRS1: RRC    A
        MOV    TMDATA,C
        NOP
        SETB  TMCLCK ;RISING EDGE CLOCK
        LCALL DTDEL
        CLR   TMCLCK
        LCALL DTDEL
        DJNZ  R5,DTWRS1
        RET

```

```
DTDEL:  MOV    R6,#4 ;DELAY
        DJNZ  R6,$
        RET

```

```
DTDELE: MOV    R5,#40H ;END
        ; DELAY(FOR EEPROM)

```

```
DTDELE1: MOV   R6,#0
        DJNZ  R6,$
        DJNZ  R5,DTDELE1
        RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ***** DTRD8,DTRD16 SUB *****
; WRITE COMMAND , READ DATA TO/FROM DS1620
; IN = R2  COMMAND
; OUT = R3,R4 DATA (9 BIT)
; REG = A,R2,R3,R4,R5,R6
```

```
DTRD8:  CLR    TMCLCK    ;CLK=0
        LCALL  DTDEL
        SETB   TMRSTT   ;RST=1
        LCALL  DTDEL
        MOV    A,R2      ;COMMAND
        LCALL  DTWRS
        LCALL  DTRDS
        MOV    R3,A
        CLR    TMRSTT   ;RST=0
        LCALL  DTDEL
        RET
```

```
DTRD16: CLR    TMCLCK    ;CLK=0
        LCALL  DTDEL
        SETB   TMRSTT   ;RST=1
        LCALL  DTDEL
        MOV    A,R2      ;COMMAND
        LCALL  DTWRS
        LCALL  DTRDS
        MOV    R4,A
        LCALL  DTRDS
        ANL   A,#0000001B
        MOV    R3,A
        CLR    TMRSTT   ;RST=0
        LCALL  DTDEL
        RET
```

```
DTRDS:  MOV    R5,#8     ;READ DATA SUB
        CLR    A
```

```
DTRDS1: CLR    TMCLCK
        LCALL  DTDEL
        MOV    C,TMDATA
        RRC    A
        SETB   TMCLCK
        LCALL  DTDEL
        DJNZ   R5,DTRDS1
        RET
```

```
; ***** HTOD SUB *****
```

```
; HEX TO DECIMAL
```

```
; IN = DPTR
```

```
; OUT = R1,R2,R3
```

```
; REG = A,R0,R1,R2,R3,R4,R5,DPTR
```

```
HTOD:  CLR    A          ;CLEAR OUTPUT
        MOV    R1,A
        MOV    R2,A
        MOV    R3,A
```

```
HTOD1: MOV    R4,#16     ;SHIFT 16 BIT
        MOV    A,DPL
        RLC    A
        MOV    DPL,A
        MOV    A,DPH
        RLC    A
        MOV    DPH,A
```

```
HTOD2: MOV    R5,#3     ;ADD DECIMAL
        MOV    R0,#3     ;INDEX TO R3
        MOV    A,@R0
        ADDC  A,ACC
        DA    A
        MOV    @R0,A
        DEC   R0
        DJNZ  R5,HTOD2
        DJNZ  R4,HTOD1
        RET
```

```
; ***** DTOH SUB *****
```

```
; DECIMAL TO HEX
```

```
; IN = R1,R2,R3
```

```
; OUT = DPTR
```

```
; REG = A,R0,R1,R2,R3,R4,R5,DPTR
```

```
DTOH:  MOV    R4,#16
```

```

D1OH1:  MOV    R5, #3           ;SHIFT & SUB
        MOV    R0, #1         ;INDEX TO R1
        CLR    C
D1OH2:  MOV    A, @R0
        RRC    A
        PUSH  PSW            ;-[
        JNB   ACC. 7, D1OH3
        CLR    C
        SUBB  A, #30H
D1OH3:  JNB   ACC. 3, D1OH4
        CLR    C
        SUBB  A, #03H
D1OH4:  MOV    @R0, A
        INC   R0
        POP   PSW            ;-]
        DJNZ  R5, D1OH2

        MOV   A, DPH
        RRC   A
        MOV   DPH, A
        MOV   A, DPL
        RRC   A
        MOV   DPL, A
        DJNZ  R4, D1OH1
        RET

        END

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้