



ระบบเก็บขมภัยไทยใช้ไมโครคอนโทรลเลอร์

SECURITY SYSTEM BY MICROCONTROLLER



ไทย

นางสาว ชนิตา เกียรติไวศยวรรณ

นางสาว ยุพาพร ขารุศิริวัฒน์

นางสาว รวีวรรณ สิริไวทย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของงานศึกษาระดับปริญญาตรี สาขาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2535

ภาควิชา ระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

เรื่อง ระบบเตือนภัยโดยใช้ไมโครคอนโทรลเลอร์

ผู้จัดทำ

1. นางสาว ชนิศา เกรือไวศยวรรณ

2. นางสาว ยุพาพร จารุศิริพจน์

3. นางสาว รวิวรรณ สิริไวทย

อ. เกียรติวรรณ ทรงสัจย์ อาจารย์ที่ปรึกษา

()

ปริญญาานิพนธ์ ปีการศึกษา 2535

ภาควิชา ระบบควบคุม

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

เรื่อง ระบบเตือนภัยโดยใช้ไมโครคอนโทรลเลอร์

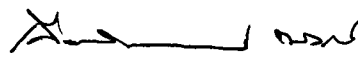
.Security System By Microcontroller

ผู้จัดทำ

1. นางสาว ธนิตา เครือไวศยวรรณ เลขประจำตัว 32.1115

2. นางสาว บุภาพร จารุศิริพจน์ เลขประจำตัว 32.1252

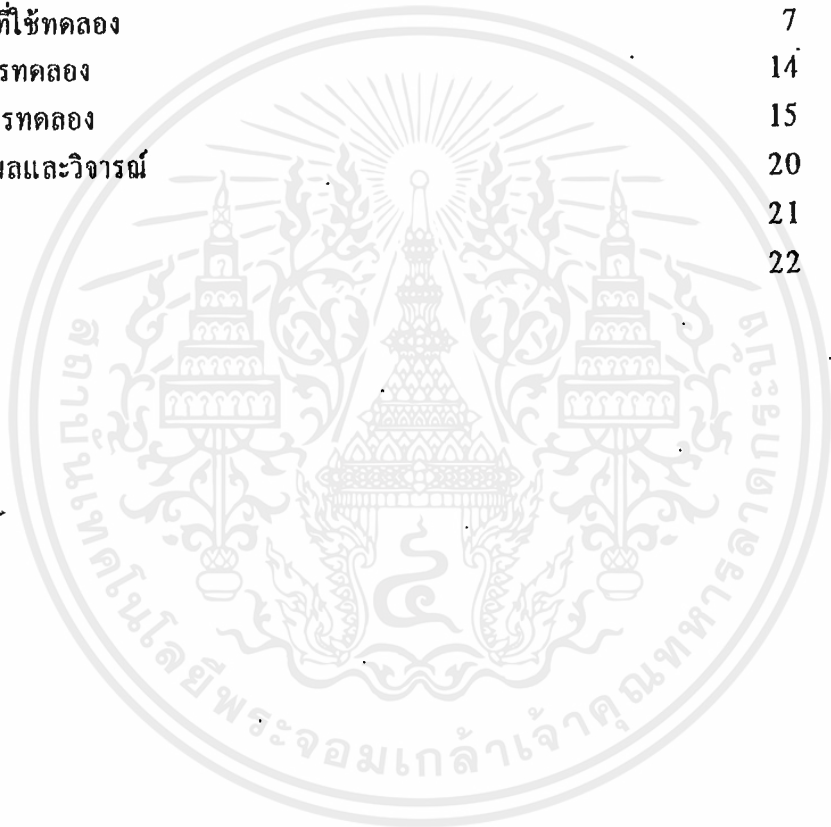
3. นางสาว รวิวรรณ สิริไวทย์ เลขประจำตัว 32.1254


..... (อาจารย์ที่ปรึกษา)

(อาจารย์ เกียรติวรรณ ทรงสัจย์)

สารบัญ

	หน้า
บทคัดย่อ	i
Abstract	ii
กิตติกรรมประกาศ	iii
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีของการสื่อสารข้อมูล	3
บทที่ 3 วงจรที่ใช้ทดลอง	7
บทที่ 4 วิธีการทดลอง	14
บทที่ 5 ผลการทดลอง	15
บทที่ 6 สรุปผลและวิจารณ์	20
เอกสารอ้างอิง	21
ภาคผนวก	22



สารบัญรูป

	หน้า
รูปที่ 1.1 รูปแบบการรับส่งข้อมูลด้วยเนกาทีฟฟิล์ม	1
รูปที่ 2.1 แสดงองค์ประกอบในการสื่อสาร	3
รูปที่ 2.2 การส่งข้อมูลแบบขนาน	3
รูปที่ 2.3 การส่งข้อมูลแบบอนุกรม	4
รูปที่ 2.4 รูปแบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม	6
รูปที่ 3.1 แสดงระบบเคเบิลใยแก้วนำแสง	7
รูปที่ 3.2 แสดงวงจรส่วนประกอบต่างๆ ของส่วนกลาง	8
รูปที่ 3.3 แสดงวงจรสวิตช์ซิง	9
รูปที่ 3.4 แสดงวงจรส่วนตรวจจับสัญญาณลบ	10
รูปที่ 3.5 แสดงวงจรส่วนประกอบต่างๆ ของส่วนลูกข่าย	10
รูปที่ 3.6 แสดงวงจรแยกสัญญาณซิงโครไนซ์	11
รูปที่ 3.7 แสดงวงจรสร้างสัญญาณช่วงแรงดันลบ	11
รูปที่ 3.8 แสดงวงจรส่วนประกอบต่าง ๆ ของส่วนแสดงผล	13

ระบบเตือนภัยโดยใช้ไมโครคอนโทรลเลอร์

นางสาว ธนิตา เครือไวศยวรรณ 32.1115

นางสาว ชุพาพร จารุศิริพจน์ 32.1252

นางสาว รวิวรรณ สิริไวทย์ 32.1254

อาจารย์ที่ปรึกษา

อาจารย์ เกียรติวรรณ ทรงสัจย์

บทคัดย่อ

โครงการนี้มีจุดมุ่งหมาย เพื่อนำไปใช้ในงานที่เกี่ยวข้องกับระบบเตือนภัย โดยใช้ไมโครคอนโทรลเลอร์ (8031) เป็นศูนย์กลางในการควบคุมการรับ-ส่ง สัญญาณกับอุปกรณ์ ณ ตำแหน่งต่าง ๆ ที่ต้องการตรวจจับความผิดปกติ เช่น การตรวจจับความผิดปกติ ตามประตู หน้าต่าง และ หลอดไฟ โดยหลักการคร่าว ๆ จะเริ่มจากศูนย์กลาง ส่งสัญญาณ แบบอนุกรมไปยังตำแหน่งที่ติดตั้งอุปกรณ์ไว้ และถ้าอุปกรณ์นั้นยังทำงานปกติคืออยู่ ก็จะส่ง สัญญาณเนกาทีฟฟลัสกลับมา แต่ถ้าผิดปกติ ก็จะไม่ส่งสัญญาณตอบรับกลับมา และเมื่อตรวจสอบจนครบทุกๆ ตำแหน่งแล้ว หลังจากนั้นศูนย์กลางก็จะทำ การแสดงตำแหน่ง ที่ผิดปกติทั้งหมด

Security System By Microcontroller

By :

Miss Thanisa Kruawaisayawan

Miss Yupaporn Jarusiriphot

Miss Rawiwan Siriwai

Advisor :

Mr. Keattiwat Songsat

Abstract

This project is about security system which has a microcontroller (8031) functioning as the center to control signal transmission between itself and substations at any positions , such as door detector or window detector etc. Its concept is the center sending signal in the form of serial data communication to all substations which have-sensors for detecting . If no error occurred, the sub - station will send a negative pulse to the center Otherwise, there is no negative pulse sent to it . Having scanned every substation , the center displays that detected error on LED 7 - segment.

กิตติกรรมประกาศ

โครงการนี้จะไม่สำเร็จลุล่วงไปได้ด้วยดี ถ้าขาดความช่วยเหลือจาก อาจารย์ เกียรติวรรณ ทรงสถิตย์ ซึ่งคอยดูแลช่วยเหลือและให้คำปรึกษา อีกทั้งสนับสนุนทางด้านเครื่องมือและวัสดุอุปกรณ์ ดังนั้นจึงขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ อาจารย์ วันชัย ธีรรุจา ที่คอยให้กำลังใจอย่างสม่ำเสมอตลอดมาและขอขอบคุณอาจารย์ทุกท่านที่ได้ให้ความรู้สำหรับการทำโครงการนี้

สุดท้ายขอขอบคุณ คุณเพื่อนๆที่คอยส่งเสบียงอาหาร , น้ำ และช่วยเหลือทางด้านอุปกรณ์ที่จำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

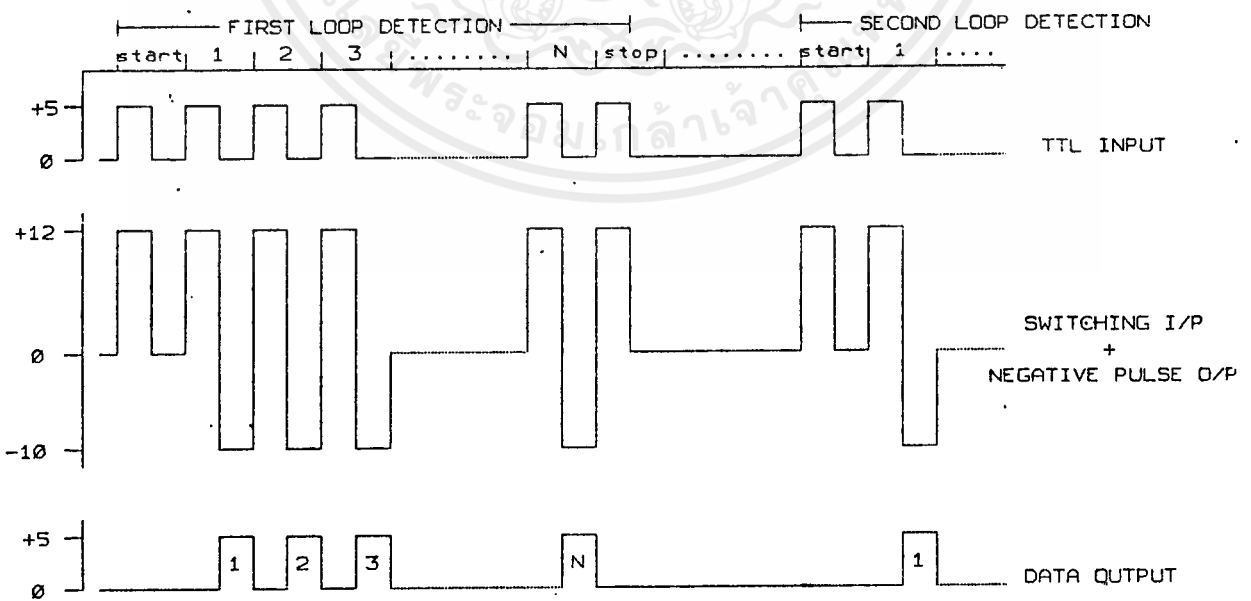
บทที่ 1

บทนำ

ในปัจจุบัน ความเจริญทางเทคโนโลยีได้ก้าวหน้าไปมากกว่าในอดีต มีตึกสูงผุดขึ้นมามากมายมาพร้อมกับดอกเห็ด ดังนั้นความปลอดภัยในอาคารจึงมีความจำเป็นต่อผู้อาศัยเป็นอย่างมาก เช่น ระบบเตือนภัยขณะเกิดเพลิงไหม้ หรือ ขณะมีผู้ไม่ประสงค์ดีบุกรุกเข้ามาในยามวิกาล เป็นต้น เนื่องจากมีเทคนิคต่าง ๆ มากมาย แต่ในโครงการนี้จะกล่าวถึง การตรวจจับสัญญาณผิดปกติโดยใช้การถ่ายโอนข้อมูลแบบอนุกรม อาศัยเทคนิคการรับส่งข้อมูลแบบเนกาทิฟพัลส์ ซึ่งสามารถรับส่งข้อมูลได้ระยะทางไกลด้วยความเร็วสูง และมีสัญญาณรบกวนต่ำ จึงทำให้ข้อมูลที่รับส่งนั้นเป็นไปได้อย่างถูกต้อง

การรับส่งข้อมูลแบบเนกาทิฟพัลส์ คือ การแปลงสัญญาณดิจิตอลแบบทีทีแอล (สัญญาณในระดับแรงดันช่วง 0 ถึง +5 โวลต์) ให้เป็นสัญญาณดิจิตอลในระดับแรงดันช่วงลบ โดยที่เราจะไม่ใช้แหล่งจ่ายไฟลบ ในการส่งสัญญาณเลข จุดประสงค์ก็เพื่อลดสัญญาณรบกวนที่จะเกิดจากแหล่งจ่ายไฟลบ และทำให้สามารถส่งสัญญาณดิจิตอลสองสัญญาณพร้อมกันในสายส่งคู่เดียวคือ ส่งสัญญาณหนึ่งในระดับแรงดันช่วงบวกในหนึ่งจะเป็นสัญญาณนาฬิกา และอีกสัญญาณหนึ่งในระดับแรงดันช่วงลบ ซึ่งจะเป็นสัญญาณตอบรับจากตำแหน่งใด ๆ

การรับส่งสัญญาณดิจิตอล สัญญาณนาฬิกาเป็นสิ่งสำคัญ เพื่อให้เครื่องรับซิงโครนัสกับเครื่องส่งสัญญาณนาฬิกาของเครื่องรับ และ เครื่องส่งจึงต้องตรงกัน เราแก้ปัญหานี้โดยส่งสัญญาณนาฬิกาเข้าวงจรสวิตซ์ซิง เพื่อขยายสัญญาณให้สวิตซ์ในช่วง 0 ถึง +12 โวลต์ ส่งเป็นสัญญาณช่วงแรงดันบวกเข้าไปในในสายส่ง ส่วนสัญญาณตอบรับที่ต้องการจะส่ง ต้องผ่านวงจรสร้างสัญญาณช่วงแรงดันลบกลับสัญญาณให้อยู่ในช่วง 0 ถึง -10 โวลต์ ส่งเป็นสัญญาณช่วงแรงดันลบเข้าไปในสายส่ง ซึ่งจะได้สัญญาณที่ผสมกันของสัญญาณในสายส่งคู่เดียว เมื่อถึงปลายเครื่องรับสัญญาณในสายส่งจะผ่านวงจรสองวงจร วงจรหนึ่งคือวงจรแยกสัญญาณ ซึ่งจะแยกสัญญาณช่วงแรงดันบวกออกมาเป็นสัญญาณนาฬิกาแบบทีทีแอล ส่วนอีกวงจรหนึ่งคือ วงจรตรวจจับสัญญาณช่วงแรงดันลบ และกลับให้เป็นสัญญาณช่วงแรงดันบวกแบบทีทีแอล ซึ่งจะอธิบายอธิบายวงจรที่กล่าวมาทั้งหมดอย่างละเอียดอีกครั้ง ในบทที่ 3



รูป 1.1 รูปแบบการรับส่งข้อมูลด้วยเนกาทิฟพัลส์

ระบบเตือนภัยโดยใช้ไมโครโปรเซสเซอร์ มีส่วนประกอบต่าง ๆ ที่สำคัญดังนี้ คือ

- ส่วนกลาง (Master)
- ส่วนลูกข่าย (Sub-station)
- ส่วนประมวลผล (Microcontroller)
- ส่วนแสดงผล (Display)

1. ส่วนกลาง เป็นส่วนส่งสัญญาณพัลส์ไปยังตัวลูกต่าง ๆ (ที่กำหนดค่าตำแหน่งสถานีย่อยไว้แล้ว) และคอยตรวจสอบความผิดปกติของตัวลูก อีกทั้งสามารถส่งสัญญาณควบคุมระยะไกล โดยใช้ 8031 ควบคุมการทำงานทั้งหมด
2. ส่วนลูกข่าย เป็นส่วนตรวจสอบความผิดปกติโดยใช้เซ็นเซอร์ (Sensor) ชนิดต่าง ๆ แก้วแค่สถานการณ์ที่จะตรวจสอบ ถ้าไม่มีความผิดปกติเกิดขึ้น ก็จะตอบรับเป็นสัญญาณช่วงแรงดันลบกลับมา แต่ถ้าตรวจสอบพบ ก็จะเลิกส่งสัญญาณช่วงแรงดันลบในส่วนของพัลส์ของตนเอง เพื่อบอกหน่วยประมวลผล (CPU) ให้รู้ว่าเกิดเหตุการณ์ผิดปกติขึ้น
3. ส่วนประมวลผล เป็นส่วนสำคัญในการควบคุมการทำงานทั้งหมดของระบบ ซึ่งโปรแกรมต่าง ๆ ได้ทำการบรรจุไว้ใน EPROM นอกจากนี้ยังใช้หน่วยความจำเพื่อเก็บข้อมูลขณะเมื่อมีความผิดปกติ ซึ่งจะบันทึกตำแหน่งของสถานีย่อยนั้นไว้เพื่อนำมาตรวจสอบในภายหลัง เนื่องจากจะต้องส่งพัลส์ไปจนครบทุกตำแหน่งของสถานีย่อยก่อน
4. ส่วนแสดงผล เป็นส่วนที่ผู้ใช้ติดต่อให้ทราบการทำงานของระบบ ประกอบด้วย
 - จอ LED 7 เซ็กเมนต์ (7-segment) จำนวน 4 หลัก ใช้แสดงข้อมูลต่างๆ ที่จอ
 - สวิตช์ (switch) 12 ตัว ใช้ในการกำหนดจำนวนสถานีย่อยที่จะใช้ตรวจสอบ รวมทั้งฟังก์ชันคีย์ (Function Key) ต่าง ๆ

ปกติการทำงานในระบบจะมีการส่งพัลส์ในการตรวจสอบความผิดปกติ จนกว่าจะมีการผิดปกติเกิดขึ้น ระบบจะทำงานจนครบจำนวนสถานีย่อยที่กำหนดไว้แล้วจะทำการแจ้งสถานีย่อยที่ผิดปกตินั้น ๆ บนจอแอลอีดี 7 เซ็กเมนต์ เพื่อรอการตรวจสอบและแก้ไข

ข้อดีของระบบ คือ ใช้สายสัญญาณเพียง 2 เส้นเท่านั้นทำให้สะดวกต่อการติดตั้งเดินสาย และสามารถนำไปใช้งานได้กว้างขวาง เนื่องจากสามารถต่อจำนวนสถานีย่อยได้ถึง 255 สถานีย่อยและผู้ใช้สามารถเลือกเซ็นเซอร์เพื่อมาตรวจสอบความผิดปกติได้มากตามจุดประสงค์การใช้งาน เนื่องจากตัวสถานีย่อยนั้นไม่ยึดติดกับชนิดของเซ็นเซอร์

สำหรับการนำระบบไปใช้งานควรเลือกใช้ในระบบที่มีขนาดใหญ่ เช่น โรงแรม, อาคารสำนักงาน, คอนโดมิเนียม หรือ อาคารต่าง ๆ เป็นต้น และควรเลือกใช้เซ็นเซอร์ที่เหมาะสมกับสภาพแวดล้อมและความต้องการ เพื่อให้ได้ระบบที่ดีมีประสิทธิภาพสูงสุด

บทที่ 2 ทฤษฎีของการสื่อสารข้อมูล

ในบทนี้จะกล่าวถึงหลักการเบื้องต้นของการสื่อสารข้อมูลโดยอาศัยไมโครคอมพิวเตอร์

2.1 องค์ประกอบเบื้องต้นในการสื่อสารข้อมูล

ในการสื่อสารข้อมูลต้องมีปัจจัยอย่างน้อย 3 ประการ คือ

- 1) แหล่งกำเนิด (source)
- 2) ตัวกลาง (medium)
- 3) ตัวรับ (receiver)

สามารถแสดงความสัมพันธ์ได้ดังรูป



รูป 2.1 แสดงองค์ประกอบในการสื่อสาร

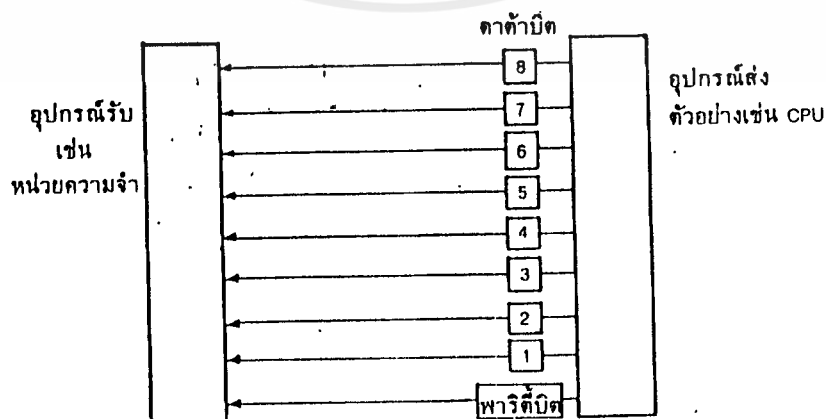
2.2 รูปแบบการถ่ายโอนข้อมูล

วิธีการถ่ายโอนข้อมูลระหว่างไมโครคอมพิวเตอร์ สามารถแบ่งออกเป็น 2 วิธี คือ

1) การถ่ายโอนแบบขนาน

ในระบบไมโครคอมพิวเตอร์ทั่วไป การส่งข้อมูลระหว่างอุปกรณ์ต่าง ๆ ที่อยู่บนแผงวงจรล้วนแล้วแต่เป็นการส่งแบบขนานทั้งสิ้น การส่งในลักษณะนี้ทำได้โดยส่งข้อมูลออกมาพร้อม ๆ กันทุกบิต ดังนั้นเส้นทางของข้อมูลจึงมีเท่ากับจำนวนบิต เช่น ถ้าเป็นไมโครคอมพิวเตอร์ 8 บิต สายส่งข้อมูลก็จะมี 8 เส้นด้วย

ลักษณะการถ่ายโอนข้อมูลแบบขนานจะเป็นดังรูป 2.2

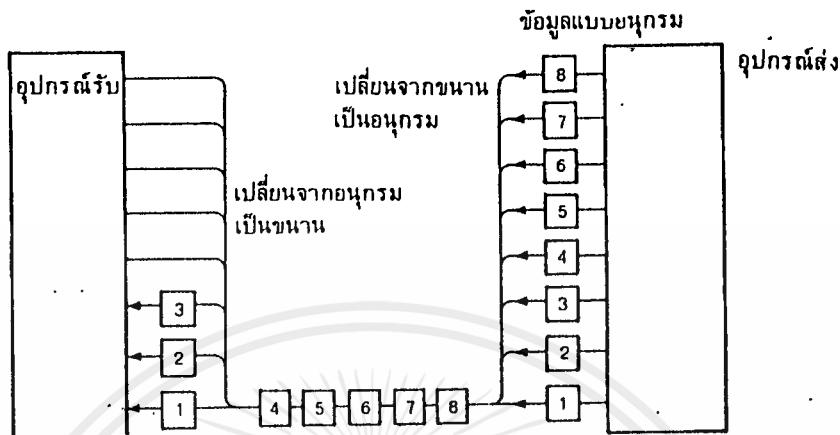


รูป 2.2 การถ่ายโอนข้อมูลแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การถ่ายโอนแบบอนุกรม

การส่งแบบอนุกรมนี้ ข้อมูลแต่ละบิตจะถูกส่งเรียงกันออกไปเป็นลำดับต่อเนื่องกันทีละบิตระหว่างจุดส่งและจุดรับ จากการส่งในลักษณะนี้จึงใช้สายส่งข้อมูลเพียงคู่เดียวเท่านั้น การถ่ายเทข้อมูลแบบอนุกรมนี้ ถึงแม้จะช้าแต่ก็เหมาะกับการส่งในระยะไกล ๆ ทั้งนี้เป็นเพราะประหยัดสายส่งได้เป็นจำนวนมาก ลักษณะการถ่ายโอนข้อมูลจะเป็นดังรูป 2.3



รูป 2.3 การถ่ายโอนข้อมูลแบบอนุกรม

ข้อเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานและแบบอนุกรม

	แบบขนาน	แบบอนุกรม
1. ระยะทาง	ปกติจะน้อยกว่า 100 ฟุต	ส่งได้ตั้งแต่ระยะสั้น ๆ จนถึงระยะทางเป็นไมล์
2. ความเร็ว	อัตราความเร็วสูงมากในระยะที่ไม่ไกลมากนัก กำหนดได้เป็นจำนวนบิตต่อวินาที	อัตราความเร็วของข้อมูลที่ใช้กันอยู่ทั่วไป จะอยู่ในช่วง 0 ถึง 2 ล้านบิตต่อวินาที
3. ระดับของสัญญาณ	ในการอินเทอร์เฟซจะใช้ระดับสัญญาณที่ใช้กับอุปกรณ์ TTL คือ สัญญาณลอจิก 1 และ 0 จะแทนด้วยระดับแรงดัน +5 V. และ 0 V. ตามลำดับ	ใช้มาตรฐานของ EIA-RS 232C คือมีระดับสัญญาณไฟฟ้าขนาด 12 V. หรืออาจจะใช้มาตรฐาน 20 mA. current loop หรืออาจจะใช้ระดับสัญญาณของ TTL ก็ได้
4. ความผิดพลาดของสัญญาณ	ถ้าส่งในระยะทางไกล ๆ ความผิดพลาดของข้อมูลจะเกิดขึ้นง่าย	การผิดพลาดของสัญญาณจะมีน้อยลง สั้นเปลืองน้อยกว่าหลายเท่า
5. ค่าใช้จ่าย	ถ้าส่งในระยะทางไกล ๆ จะสิ้นเปลืองค่าใช้จ่ายมาก เพราะต้องใช้สายส่งสัญญาณหลายเส้น	เท่า ถึงแม้ว่าจะใช้อุปกรณ์เปลี่ยนสัญญาณของข้อมูลแบบขนานไปเป็นแบบอนุกรมแล้วส่งผ่านสายส่งใช้อุปกรณ์ในการแปลงสัญญาณกลับมาเป็นแบบขนานอีก ก็ยังลงทุนน้อยกว่า

เนื่องจาก ความจริงแล้วภายในไมโครคอมพิวเตอร์มีการรับส่งข้อมูลแบบขนาน ดังนั้นจึงจำเป็นต้องมีการแปลงสัญญาณเข้าให้เหมาะสมเสียก่อน โดยการแปลงสัญญาณให้เป็นแบบขนานซึ่งมีรูปแบบในการรับ (protocol) เพื่อป้องกันการผิดพลาดในการรับ กลไกที่วางนี้แบ่งเป็น 2 แบบ คือ

- การรับส่งแบบอะซิงโครนัส (Asynchronous protocol)
- การรับส่งแบบซิงโครนัส (Synchronous protocol)

2.2.1 การรับส่งแบบอะซิงโครนัส (Asynchronous protocol)

การส่งข้อมูลแบบนี้ ข้อมูลที่ถูกส่งออกไปจะมีลักษณะเป็นบิตต่อก ๆ ซึ่งในแต่ละบิตก็จะมีประกอบด้วยบิตเริ่มต้น (start bit), ส่วนของข้อมูล และบิตสิ้นสุดของข้อมูล (stop bit) โดยบิตเริ่มต้นจะแสดงถึงการมาหรือการเริ่มต้นของข้อมูล แล้วตามด้วยส่วนของกลุ่มข้อมูล ในบางกรณีอาจจะมีการเพิ่มบิตพาริตีเข้าไปเพื่อใช้ตรวจสอบความถูกต้องของข้อมูล ส่วนในบิตสุดท้ายนั้นจะเป็นการบอกว่าข้อมูลได้สิ้นสุดเพียงเท่านี้

- บิตเริ่มต้น ในโปรโตคอลของการส่งข้อมูลอนุกรมแบบอะซิงโครนัส กำหนดให้สถานะมาร์ค (marking state) เป็นสัญญาณลอจิก 1 เมื่อทางด้านส่งจะทำการส่งข้อมูล ก็จะส่งบิตเริ่มต้นหรือสัญญาณลอจิก 0 ทำให้ทางด้านรับสามารถตรวจสอบสถานะของสายส่งได้ว่าในขณะนั้นกำลังจะมีข้อมูลส่งมา ปัญหาอันหนึ่งที่พบบ่อย ๆ ในการส่งสัญญาณข้อมูล คือ สัญญาณสไปร์ค (spike) ซึ่งทำให้สถานะลอจิกสั้นเกินไป ปัญหานี้แก้ได้โดยจะมีส่วนของวงจรสไปร์คดิเทกชัน (spike detection)

- บิตข้อมูล หลังจากที่ด้านรับสามารถตีเท็กสัญญาณบิตเริ่มต้นได้แล้ว ก็จะมีการเซ็ทสถานะของชิพรีจิสเตอร์ ให้พร้อมที่จะรับบิตข้อมูลได้ โดยบิตข้อมูลจะมีจำนวนบิตเป็น 5,6,7 หรือ 8 บิต ขึ้นกับจำนวนคาร์เรียร์เคอร์

- บิตพาริตี จะทำหน้าที่ในการบอกให้ส่วนรับข้อมูลทราบว่า ข้อมูลที่รับเข้ามาผิดหรือไม่ บิตพาริตีนี้จะถูกส่งออกมาพร้อมกับบิตข้อมูล ซึ่งบิตนี้จะเป็น 1 หรือ 0 ขึ้นอยู่กับข้อมูลที่ส่งออกมาว่ามีจำนวนบิต 1 เป็นจำนวนคี่หรือคู่ และยังขึ้นกับอุปกรณ์รับส่งข้อมูลด้วยว่าถูกออกแบบไว้ให้รับส่งบิตพาริตีในลักษณะพาริตีคู่หรือคี่อีกด้วย

2.2.2 การรับส่งข้อมูลแบบซิงโครนัส (Synchronous Protocol)

ในการส่งข้อมูลแบบซิงโครนัสนั้นต่างจากการส่งแบบอะซิงโครนัสคือ การส่งแบบนี้จะไม่มีการส่งสัญญาณ start bit หรือ stop bit ออกไปซิงค์กับตัวรับ แต่จะมีการส่งชุดสถานะของสัญญาณที่เรียกว่า sync character โดยจะส่งสัญญาณนี้ออกไปเรื่อย ๆ จนกว่าฝ่ายรับจะซิงค์กับตัวส่งได้

ดังนั้นจะเห็นได้ชัดว่า การส่งข้อมูลแบบซิงโครนัสจะมีความรวดเร็วกว่าการส่งข้อมูลแบบอะซิงโครนัส แต่อย่างไรก็ตามความผิดพลาดจากการส่งแบบอะซิงโครนัสย่อมต้องมีกว่า ทั้งนี้เนื่องมาจากไม่มีการตรวจสอบพาริตี สิ่งที่สำคัญมากสิ่งหนึ่งในการส่งข้อมูลแบบอนุกรมทั้งแบบซิงโครนัสและอะซิงโครนัสก็คือความถี่ที่ใช้ในการส่งข้อมูล ซึ่งจะต้องสัมพันธ์กันระหว่างอุปกรณ์ที่ทำการรับและส่งข้อมูล ความถี่นี้เราเรียกว่า baud rate หรือ อาจกล่าวได้ว่า

baud rate คือ " อัตราการรับส่งข้อมูลเป็นบิตต่อวินาที " ถ้าหากว่าเครื่องส่งใช้ baud rate ไม่สัมพันธ์กับเครื่องรับแล้ว ก็จะทำให้การรับส่งข้อมูลเกิดการผิดพลาดขึ้นได้

การติดต่อแบบอนุกรมอาจจะแบ่งตามรูปลักษณะได้ 3 แบบ ตามรูป 2.4

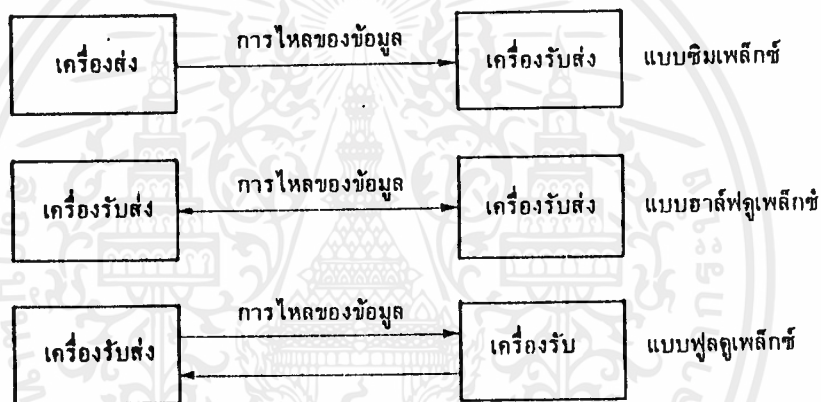
1. แบบซิมเพล็กซ์ (simplex) ข้อมูลส่งได้ในทางเดียวเท่านั้น บางครั้งก็เรียกว่า การส่งทิศทางเดียว (Unidirectional data bus)
2. แบบฮาล์ฟดูเพล็กซ์ (half duplex) ข้อมูลสามารถส่งได้ทั้งสองสถานี แต่จะต้องผลัดกันส่งและผลัดกันรับ จะส่งและรับพร้อมกันไม่ได้
3. แบบฟูลดูเพล็กซ์ (full duplex) ทั้งสองสถานีสามารถรับและส่งได้ในเวลาเดียวกัน

ตัวอย่างของระบบที่ใช้การติดต่อในลักษณะดังกล่าวนี้ได้แก่

ระบบซิมเพล็กซ์ : การส่งกระจายเสียงวิทยุหรือโทรทัศน์ หรือ การส่งข้อมูลจากคอมพิวเตอร์ไปให้เครื่องพิมพ์

ระบบฮาล์ฟดูเพล็กซ์ : ระบบวิทยุติดต่อ, ระบบ ATM

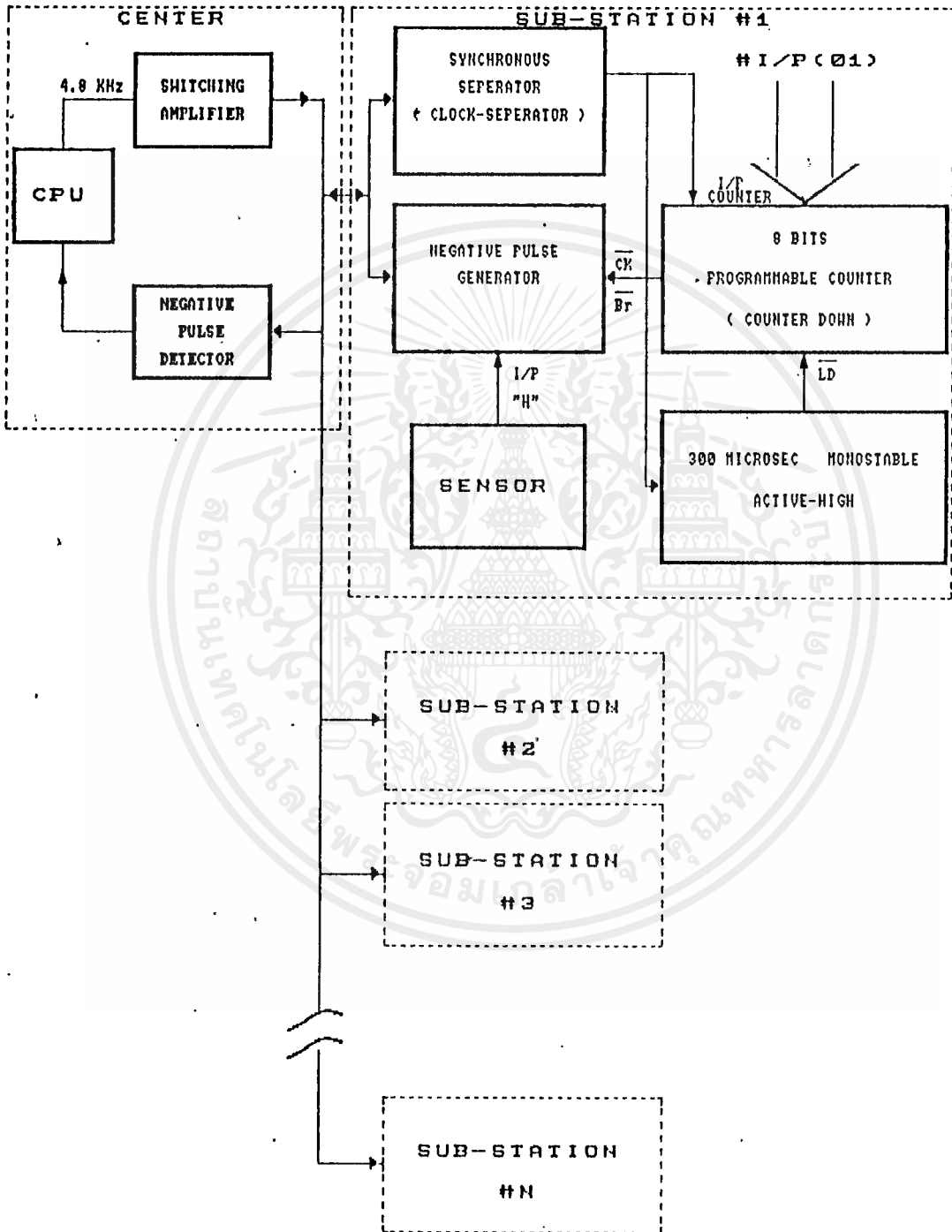
ระบบ ฟูลดูเพล็กซ์ : การสื่อสารทางโทรศัพท์ และการสื่อสารคอมพิวเตอร์



รูป 2.4 รูปแบบการติดต่อสื่อสารข้อมูลแบบอนุกรม

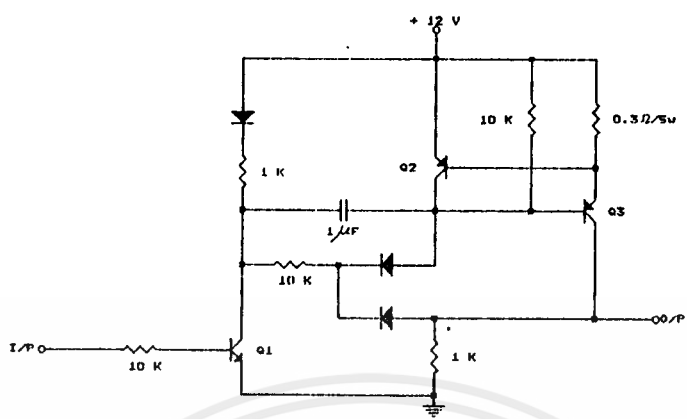
บทที่ 3
วงจรที่ใช้ในการทดลอง

ระบบเตือนภัยโดยใช้ไมโครคอมพิวเตอร์ สามารถแสดงส่วนประกอบต่างๆ ดังนี้



รูป 3.1 แสดงระบบเตือนภัยโดยใช้ไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.3 แสดงวงจรสวิทช์ซิง

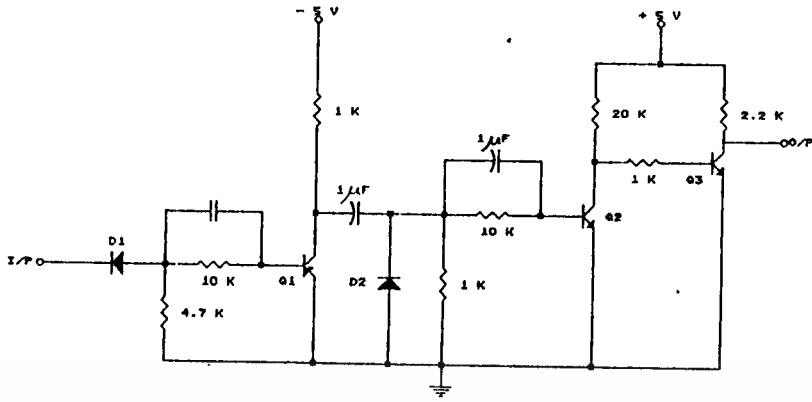
2) ส่วนตรวจจับสัญญาณลบ (Negative Pulse Detector)

วงจรนี้ทำหน้าที่รับสัญญาณพัลส์ลบที่ส่งมาจากสายจากสถานีย่อย และทำการแปลงส่วนพัลส์ลบนี้ให้กลายเป็นพัลส์บวก แล้วส่งต่อไปยังหน่วยประมวลผล เพื่อให้หน่วยประมวลผลถอดรหัสว่าสถานีย่อยตัวใดที่เกิดเหตุการณ์ผิดปกติขึ้น

กรณีที่ไม่มีสัญญาณพัลส์ลบที่อินพุทของวงจร ทรานซิสเตอร์ Q1 จะยังไม่ทำงาน จึงมีกระแสชาร์จตัวเก็บประจุ โดยผ่านทางไดโอด D2 และความต้านทาน 1 กิโลโอห์ม ซึ่งในขณะนี้โวลต์แดงที่ขาคาโอดของไดโอด D2 มีค่าประมาณ -0.65 V. ทำให้ทรานซิสเตอร์ Q2 ไม่ทำงาน ดังนั้นกระแสจากแหล่งจ่ายจะผ่านความต้านทาน 20 K, 1 K ohm และจะไปอัสทรานซิสเตอร์ Q3 ให้ทำงาน และเนื่องจากกระแสเบสมากพอที่จะทำให้ทรานซิสเตอร์ Q3 เกิดสถานะอิ่มตัว ดังนั้นที่เอาต์พุทของขาคอลเลกเตอร์ Q3 จึงมีค่าประมาณศูนย์โวลต์

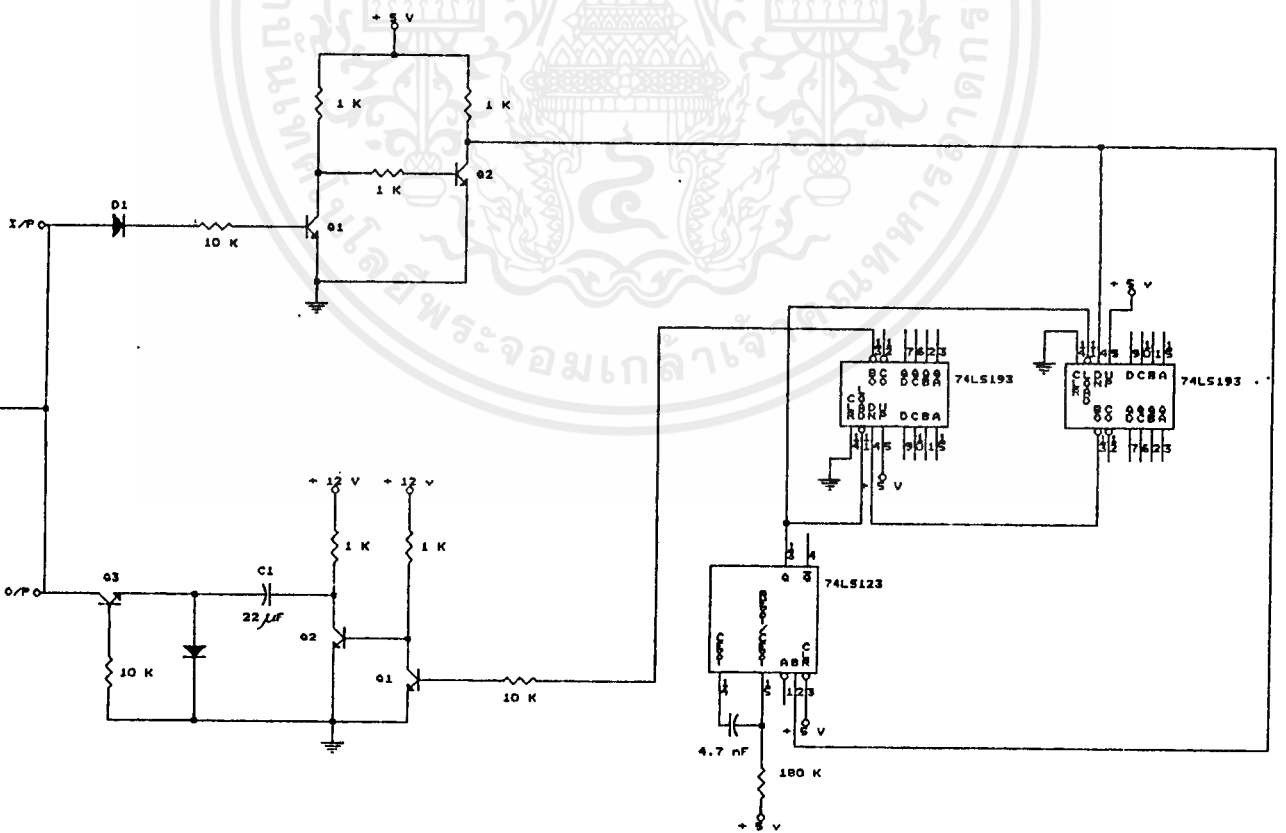
กรณีที่มีสัญญาณพัลส์ลบ เนื่องจากไดโอด D1 ทำหน้าที่กั้นไม่ให้สัญญาณพัลส์บวกผ่าน แต่ยอมให้สัญญาณพัลส์ลบผ่านเท่านั้น ดังนั้นที่ขาเบสของทรานซิสเตอร์ Q1 จึงมีโวลต์แดงเป็นลบ ซึ่งจะมีขนาดขึ้นกับขนาดสัญญาณพัลส์ลบ ดังนั้นค่าโวลต์แดงขาเบสจะน้อยกว่าที่ขาอีมิเตอร์ ซึ่งทำให้ทรานซิสเตอร์ Q1 ทำงาน ดังนั้นตัวเก็บประจุจะคายประจุเนื่องจากโวลต์แดงที่ขาคอลเลกเตอร์ของ Q1 ก่อนทำงานมีค่าประมาณ -5 V. เมื่อทรานซิสเตอร์ Q1 ทำงาน โวลต์แดงที่ขาคอลเลกเตอร์จะมีค่าประมาณศูนย์โวลต์ ดังนั้นโวลต์แดงที่ขาคาโอดของไดโอด D2 จะมีค่าประมาณ 5 โวลต์ ฉะนั้นจึงทำให้ทรานซิสเตอร์ Q2 ทำงาน โวลต์แดงที่ขาคอลเลกเตอร์จะประมาณศูนย์โวลต์ ดังนั้นทรานซิสเตอร์ Q3 จึงยังไม่ทำงาน ทำให้โวลต์แดงที่เอาต์พุทมีค่าประมาณ 5 โวลต์ ดังนั้นเมื่อมีสัญญาณพัลส์ลบเข้ามา ที่เอาต์พุทของส่วนตรวจจับสัญญาณลบนี้จะเป็นสัญญาณพัลส์บวก โดยมีขนาดของสัญญาณพัลส์บวกประมาณ 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.4 แสดงวงจรส่วนตรวจจับสัญญาณลบ

3.2 ส่วนลูกข่าย (Sub-Station)



รูป 3.5 แสดงส่วนประกอบต่างๆ ของส่วนลูกข่าย

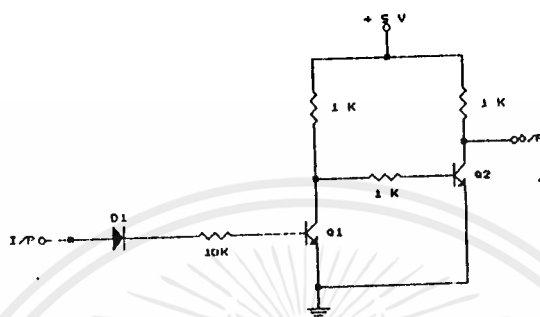
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงส่วนประกอบต่าง ๆ ของส่วนลูกข่าย ได้แก่

1) วงจรแยกสัญญาณซิงโครนัส (Synchronous Separator)

วงจรมีหน้าที่แยกสัญญาณซิงค์ แล้วนำสัญญาณซิงค์ที่ได้ (สัญญาณนาฬิกา) เป็นอินพุต countdown ของวงจรมับ และ ของโมโนสเตเบิล

เมื่อมีสัญญาณแรงดันบวกถูกส่งเข้ามา วงจรจะทำหน้าที่แยกสัญญาณที่ผสมกันอยู่ในสายส่ง โดยใช้ไดโอด D1 วงจรจะแปลงให้เป็นสัญญาณนาฬิกาในระดับที่ทีแอลและปรับแต่งสัญญาณโดยใช้ทรานซิสเตอร์ (common emitter) ในการกลับเฟส



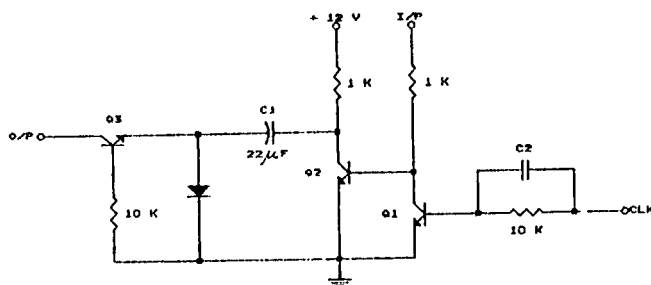
รูป 3.6 แสดงส่วนตรวจจับสัญญาณซิงค์โครนัส

2) วงจรสร้างสัญญาณช่วงแรงดันลบ

วงจรมีหน้าที่สร้างสัญญาณช่วงแรงดันลบ เมื่อไม่มีสิ่งใดผิดปกติ (sensor เป็น high) แต่ถ้ามีสิ่งผิดปกติเกิดขึ้น วงจรนี้จะไม่ทำงาน

ขณะที่มีสิ่งใดผิดปกติ กระแสจะไหลผ่านตัวต้านทาน 1 K ohm เข้าขั้วจตุรเกตุประจุ C1 และ Vc ของ Q2 ประมาณ 12 โวลต์ ทำให้ไดโอดถูกไบอัสตรง ซึ่งเปรียบเสมือนความต้านทานค่าต่ำ ๆ ดังนั้น Ve ของ Q3 ประมาณศูนย์โวลต์ แต่ Vb ของ Q2 เป็นศูนย์โวลต์ ดังนั้น Q2 จึงยังไม่ทำงาน

เมื่อสัญญาณนาฬิกาเป็น low จะทำให้ทรานซิสเตอร์ Q2, Q3 เริ่มทำงาน และ Vce ของ Q3 ประมาณศูนย์โวลต์ ตัวเก็บประจุ C จะคายประจุทำให้ไดโอดถูกไบอัสย้อนกลับ ทำให้ Q3 ทำงาน ทำให้สัญญาณเอาต์พุตมีพัลส์เป็นลบ



รูป 3.7 แสดงวงจรสร้างสัญญาณช่วงแรงดันลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) วงจรโมโนสเตเบิล (Monostable)

ในวงจรนี้ใช้เบอร์ 74HC123B ซึ่งเป็น Dual Retriggerable Monostable with จาก Function Table เมื่อให้อินพุทขา clear (ขา 11) มีสถานะเป็น High และมีอินพุทที่ขา A (ขา 9) มีสถานะเป็น Low ส่วนขา B (ขา 10) จะมีอินพุทเป็นสัญญาณนาฬิกา โดยพิจารณาที่ขอบขาขึ้น โมโนสเตเบิลจะให้เอาต์พุทออกมาที่ขา Q (ขา 5) ที่เปลี่ยนสถานะจาก 0 ไปเป็น 1 และจะคงสถานะสัญญาณไปจนกระทั่งสัญญาณนาฬิกาหยุดส่งเกินกว่า 300 s เอาต์พุทก็จะกลับลงมาเป็น 0 ส่วนขาที่ 7 เป็นขาที่ต่อกับค่า R, C ภายนอก เพื่อทำให้โมโนสเตเบิลแอกทิฟ High ที่ 300 s

4) วงจรนับ (Counter)

ในวงจรนี้ใช้เบอร์ 74HC193P ซึ่งเป็น Synchronous 4-bit Binary Up/Down Counter จากโครงงานนี้ต้องการวงจรนับ 8 บิต จึงใช้ตัวนับ 2 ตัวต่อกัน ซึ่งจะตั้งค่าตัวนับที่ขา อินพุทข้อมูล A B C D และวงจรนับจะเริ่มทำงานเมื่อขาอินพุท clear (ขา 14) มีสถานะเป็น Low และขา Load ได้มาจากโมโนสเตเบิล รวมทั้งมีสัญญาณนาฬิกาเข้ามาที่ขาอินพุท countdown (ขา 4) วงจรนับจะมีการนับค่าลงจากที่ได้ตั้งไว้จนมีค่าเป็น 0 และจะมีการเปลี่ยนสถานะของเอาต์พุทที่ขา Borrow (ขา 13) จาก High มาเป็น Low ซึ่งจะไปทำให้วงจรสร้างแรงดันลบทำงาน

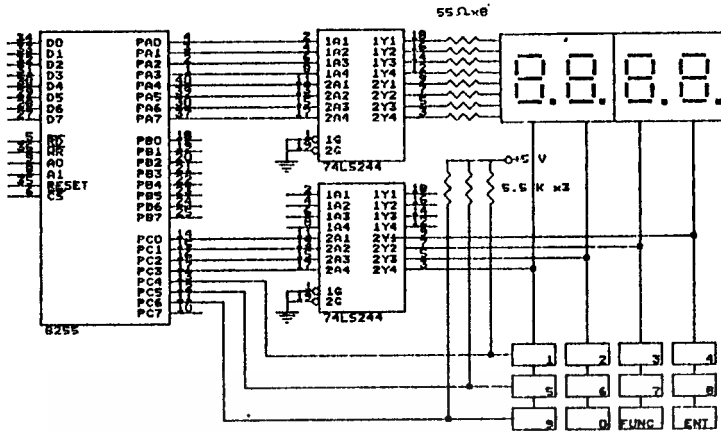
3.3 ส่วนแสดงผล (Display)

ส่วนแสดงผลนี้จะใช้ไมโครคอนโทรลเลอร์ 8031 ร่วมกับชิพเบอร์ 8255 เป็นส่วนสนับสนุน เพื่อแสดงผลออกทางแอลอีดี 7-เซกเมนต์ (LED 7-segment) โดยลักษณะของแอลอีดี 7 เซกเมนต์ ที่ใช้ในวงจรนี้ต้องใช้คอมมอนแอนอด (Common Anode) ทำงานควบคู่ไปกับไอซี 74LS244 ซึ่งทำหน้าที่เป็น Buffer และ Line driver

74LS244 จะทำหน้าที่เป็นตัวกันชน ระหว่างผู้ส่ง (ข้อมูล) กับผู้รับ (ข้อมูล) เนื่องจากบางครั้งการรับส่งข้อมูลนั้นอาจมีผู้ส่งหลายจุด แต่อาจมีผู้รับเพียงจุดเดียว ถ้าผู้ส่งหลายๆ จุดนั้นส่งข้อมูลที่มาถึงยังผู้รับนั้นปะปนกันจนเกิดความผิดพลาดขึ้น ดังนั้น ระหว่างผู้ส่งกับผู้รับจึงต้องถูกกันด้วยบัฟเฟอร์ ซึ่งจะทำหน้าที่จ่ายข้อมูลจากผู้ส่งให้แก่ผู้รับและกันข้อมูลจากผู้ส่งในจุดอื่นๆ ที่ยังไม่ต้องการส่ง หรือ ผู้รับยังไม่ต้องการจะรับ เอาไว้ไม่ให้ปะปนกัน

นอกจากทำห้สที่ เป็นบัฟเฟอร์แล้ว ยังสามารถนำมาเป็นไดรเวอร์ (driver) ได้ ในกรณีที่ผู้ส่งไม่สามารถตอบสนองเงื่อนไขบางอย่างให้แก่ผู้รับได้ เช่น เมื่อผู้ส่งต้องการส่งข้อมูลให้แก่ไอโอดเปล่งแสง (LED) เพื่อให้ติดหรือดับตามต้องการแต่ผู้ส่งเองไม่สามารถจ่ายกระแสได้มากพอที่จะขับแอลอีดี ให้สว่างได้ จึงต้องส่งข้อมูลนั้นผ่านไอซี 74LS244 ซึ่งสามารถจ่ายกระแสออก (Source) ได้ถึง 15 mA และรับกระแส (Sink) ได้ถึง 24 mA

ซึ่งจากวงจรของส่วนแสดงผล จะเป็นว่าใช้ พอร์ต A ของ 8255 เพื่อส่งข้อมูลที่ต้องการแสดงออกไปที่ แอลอีดี 7-เซกเมนต์ และใช้พอร์ต C ของ 8255 (PC0 - PC4) เพื่อเป็นพอร์ต เอาต์พุท เพื่อทำการเลือกหลักติด (Digit) และ PC5 - PC7 ของพอร์ต C เป็นพอร์ตอินพุท เพื่อรับการกดคีย์บอร์ด



รูป 3.8 แสดงส่วนประกอบต่างๆ ของส่วนแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
วิธีการทดลอง

1. เมื่อผู้ใช้เห็นคำว่า " LOAD " , " NO. " (STATION) และ " ____ " . ซึ่งหมายถึง ต้องการให้ผู้ใช้ใส่จำนวนตำแหน่งที่ต้องการตรวจสอบ (โดยกำหนดให้ไม่เกิน 254 ตำแหน่ง) ซึ่งถ้ากดเกินก็จะแสดงคำว่า " -E- " เพื่อบอกให้ผู้ใช้กดใหม่ และเมื่อได้จำนวนที่ต้องการแล้วให้กด key " ENTER "
2. หน้าจอก็จะแสดงตัวเลขที่ใส่ลงไปขึ้นมาอีกครั้ง เพื่อให้ผู้ใช้ตรวจสอบว่าถูกต้องหรือไม่ ถ้าสามารถตรวจจับความผิดพลาดได้หน้าจอก็จะแสดงคำว่า " -E- " ซึ่งหมายถึงกำลังจะแสดงจำนวนตำแหน่งที่ผิดพลาดทั้งหมด และเมื่อแสดงจำนวนตำแหน่งแล้วก็จะถามด้วยหมายเลขประจำตำแหน่งแรก ที่ผิดพลาด ถ้าต้องการดูตำแหน่งต่อไปให้กด key " ENTER " คุยไปเรื่อยๆ และเมื่อครบทุกตำแหน่งแล้วหน้าจอก็จะวนกลับไปแสดงตำแหน่งแรกใหม่



บทที่ 5
ผลการทดลอง

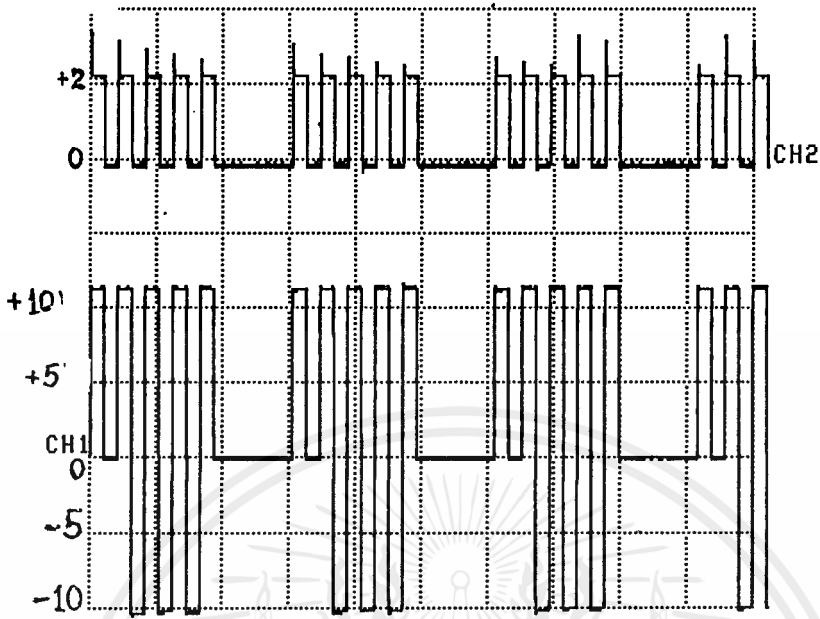
ในการทดลองนี้ ได้ทำการติดตั้งสถานีย่อยไว้ทดลอง 3 สถานี ซึ่งหลังจากที่ผู้ใช้ได้ป้อนจำนวนสถานีย่อยที่ต้องการตรวจสอบ เช่น จำนวน 3 สถานี แล้วผลการทดลองที่ได้จะเป็นไปดังตารางข้างล่างนี้

INPUT				ผลการทดลอง	
จำนวนสถานีย่อยที่ต้องการตรวจสอบ	สถานีย่อยหมายเลข 1	สถานีย่อยหมายเลข 2	สถานีย่อยหมายเลข 3	จำนวนสถานีย่อยที่ผิดปกติ	หมายเลขของสถานีย่อยที่ผิดปกติ
3	ปกติ	ปกติ	ปกติ	0	-
3	ปกติ	ผิดปกติ	ปกติ	1	2
3	ปกติ	ผิดปกติ	ผิดปกติ	2	2 , 3
3	ผิดปกติ	ผิดปกติ	ผิดปกติ	3	1 , 2 , 3

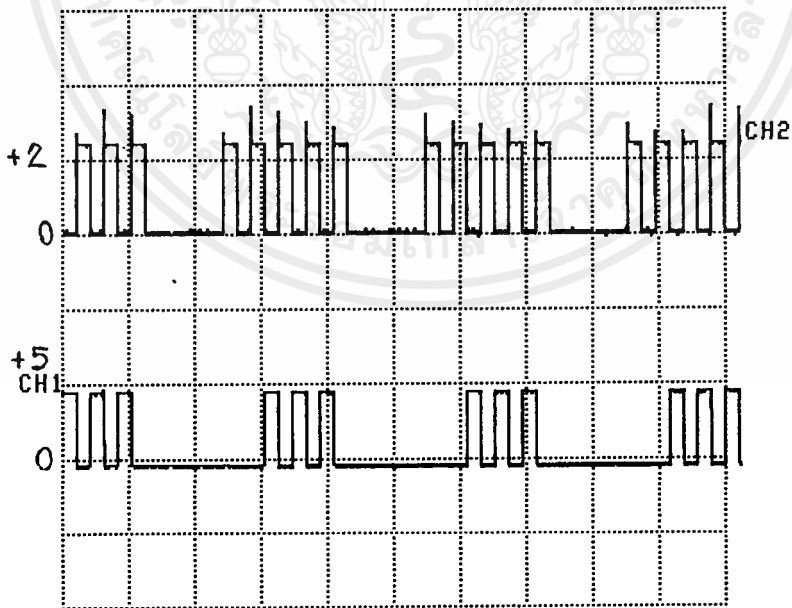
หมายเหตุ :

1. หมายเลขประจำตำแหน่ง และ จำนวนสถานีย่อยที่ต้องการตรวจสอบสามารถเปลี่ยนแปลงได้ตามต้องการ โดยจะถือว่าสถานีย่อยอื่นๆ นอกเหนือจาก 3 สถานีย่อยที่ใช้ทดลองอยู่ในสถานะ ผิดปกติเนื่องจากจะไม่มี การส่งสัญญาณตอบรับกลับมา
2. ถ้าสถานีย่อยที่ตรวจสอบปกติทุกสถานี หน้าจอจะแสดงคำว่า " RUN " เพื่อบอกให้รู้ว่ากำลังตรวจสอบอยู่

จากการทดลองจะได้กราฟรูปสัญญาณซึ่งจับจาก SCOPE รุ่น HM 205-3 โดยใช้ร่วมกับ HAMEQ GRAPHIC PRINTER HM 8148-2



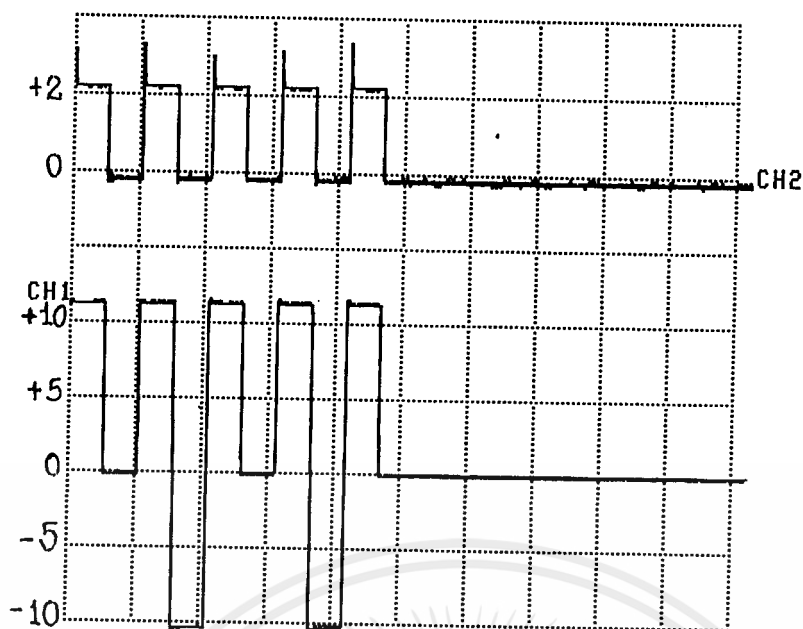
channel 2 (CH 2) จะแสดงที่ที่แวลอินพุท โดยพัลส์แรกคือ พัลส์ start bit ตามด้วยพัลส์ 3 พัลส์ เพื่อใช้ในการตรวจสอบตำแหน่งสถานีย่อย 3 สถานี และปิดท้ายด้วยพัลส์ stop bit โดยจะเปรียบเทียบกับรูปกราฟใน channel 1 (CH 1) ซึ่งแสดงถึงสัญญาณที่ตอบรับกลับมาจากสถานีย่อยที่ปกติ 3 สถานี โดยจะส่งมาเป็นสัญญาณเนกาทีฟพัลส์ โดยมีคาบของเวลาเป็น 0.5 ms



CH 1 แสดงสัญญาณที่มีการแปลงจากสัญญาณเนกาทีฟพัลส์เป็นสัญญาณที่แอตเพื่อส่งเข้าสู่ 8031 โดยมีคาบของเวลาเป็น 0.5 ms

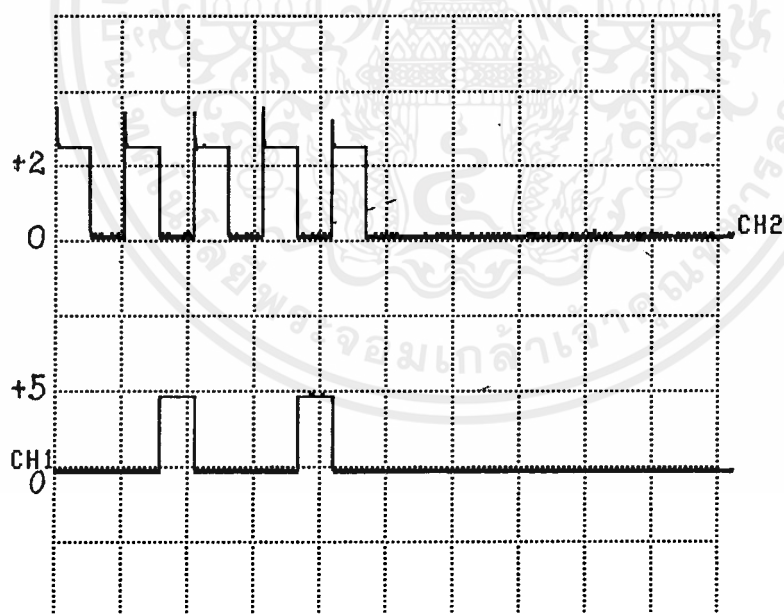
CH 2 แสดงสัญญาณที่ที่แวลอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CH 1 แสดงสัญญาณที่ไม่มีการตอบรับกลับมาจากสถานีย่อย หมายเลข 2 โดยจะส่งมาเป็นสัญญาณเนกาทีฟพัลส์โดยมีคาบของเวลาเป็น 0.2 ms

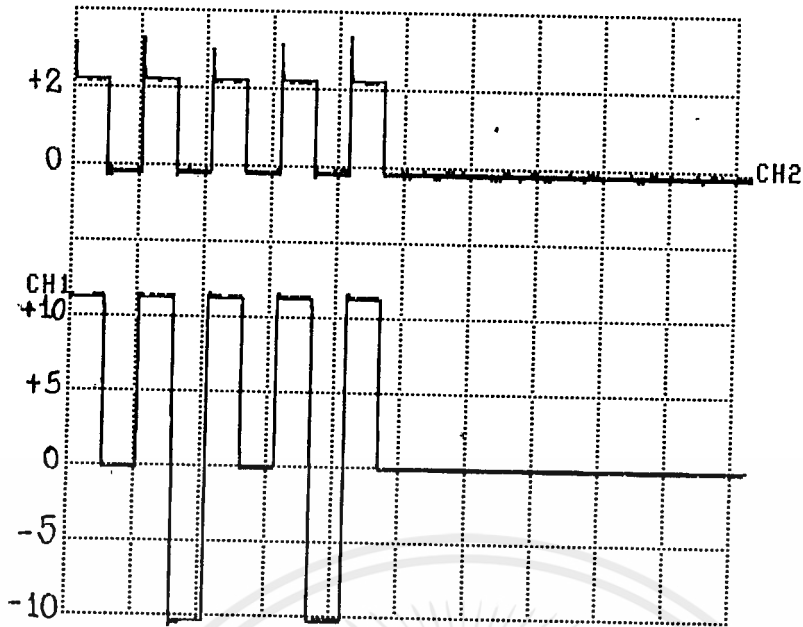
CH 2 แสดงสัญญาณที่ทีแอลอินพุท



CH 1 แสดงสัญญาณที่ไม่มีการตอบรับกลับมาจากสถานีย่อย หมายเลข 2 โดยมีการแปลงสัญญาณเนกาทีฟพัลส์เป็นสัญญาณที่ทีแอลเพื่อส่งเข้าสู่ 8031 โดยมีคาบของเวลาเป็น 0.2 ms

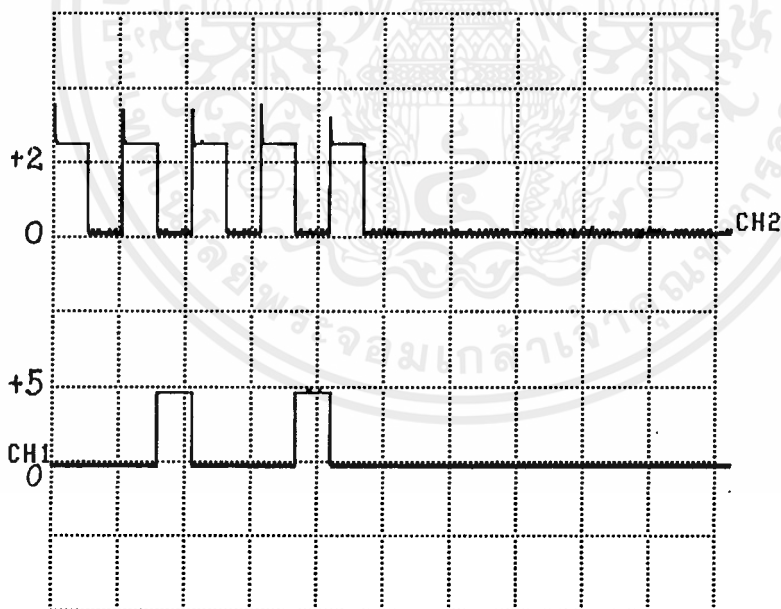
CH 2 แสดงสัญญาณที่ทีแอลอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CH 1 แสดงสัญญาณที่ไม่มีการตอบรับกลับมาจากสถานีย่อย หมายเลข 2 โดยจะส่งมาเป็นสัญญาณเนกาทีฟพัลส์โดยมีคาบของเวลาเป็น 0.2 ms

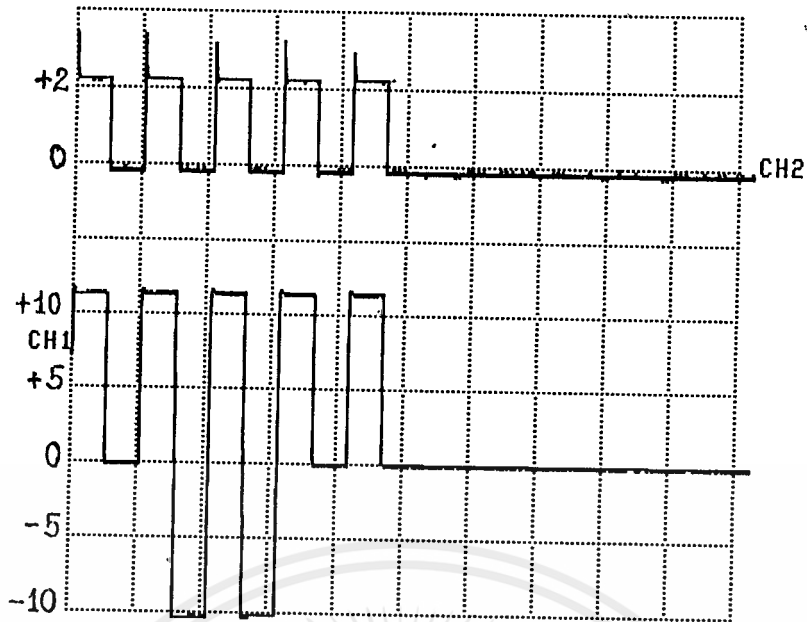
CH 2 แสดงสัญญาณที่ทีแอลอินพุท



CH 1 แสดงสัญญาณที่ไม่มีการตอบรับกลับมาจากสถานีย่อย หมายเลข 2 โดยมีการแปลงสัญญาณเนกาทีฟพัลส์เป็นสัญญาณที่ทีแอลเพื่อส่งเข้าสู่ 8031 โดยมีคาบของเวลาเป็น 0.2 ms

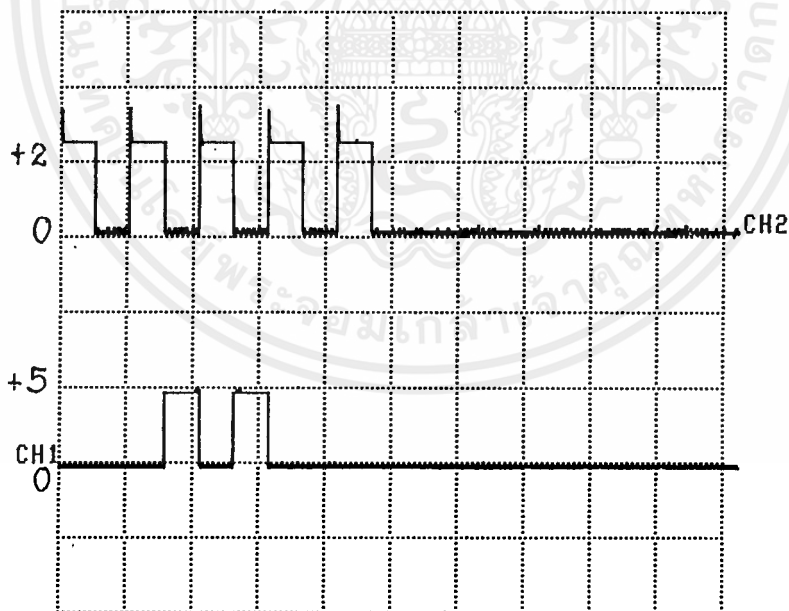
CH 2 แสดงสัญญาณที่ทีแอลอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CH 1 แสดงสัญญาณที่ไม่มีการตอบรับกลับมาจากสถานีย่อย หมายเลข 3 โดยจะส่งมาเป็นสัญญาณเนกาทีฟพัลส์โดยมีคาบของเวลาเป็น 0.2 ms

CH 2 แสดงสัญญาณที่ทีแอลอินพุท



CH 1 แสดงสัญญาณที่ไม่มีการตอบรับกลับมาจากสถานีย่อย หมายเลข 3 โดยมีการแปลงสัญญาณเนกาทีฟพัลส์เป็นสัญญาณที่ทีแอลเพื่อส่งเข้าสู่ 8031 โดยมีคาบของเวลาเป็น 0.2 ms

CH 2 แสดงสัญญาณที่ทีแอลอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 สรุปผลและวิจารณ์

โครงการนี้สามารถนำมาใช้ในระบบเตือนภัย โดยใช้ไมโครคอนโทรลเลอร์เข้ามาช่วยในการสร้างสัญญาณพิกาคความถี่ 4.8 KHz ส่งไปตามสถานีข่อยต่าง ๆ พร้อมทั้งใช้ตรวจสอบสัญญาณช่วงแรงดันลบที่คอยรับกลับมาจากสถานีข่อย ๆ นั้นด้วย ซึ่งจากผลการทดลองได้ตรงตามทฤษฎีที่ใช้ในการทดลอง

ข้อดีของระบบนี้ คือ

- สามารถรับส่งสัญญาณได้ในระยะไกลด้วยความเร็วสูง
- ในระบบนี้ทั้งระบบจะใช้สายส่งเพียง 2 เส้นเท่านั้น
- ในระบบนี้ศูนย์กลางจะสามารถติดต่อกับสถานีข่อยได้ตลอดเวลา ดังนั้นเมื่อสถานีข่อยใดผิดปกติก็จะทำให้ศูนย์กลางสามารถรับรู้ได้ทันที
- สถานีข่อยสามารถนำไปใช้งานได้กว้างขวาง เพราะต้องการเพียงแค่อุปกรณ์อะไรก็ได้ ที่ใช้เช่นเซอร์ในการตรวจจับความผิดปกติ
- ไฟเลี้ยงสำหรับสถานีข่อยทุกสถานีสามารถใช้จากศูนย์กลางไปเลี้ยงได้ โดยไม่ต้องใช้ไฟเลี้ยงแยกต่างหาก

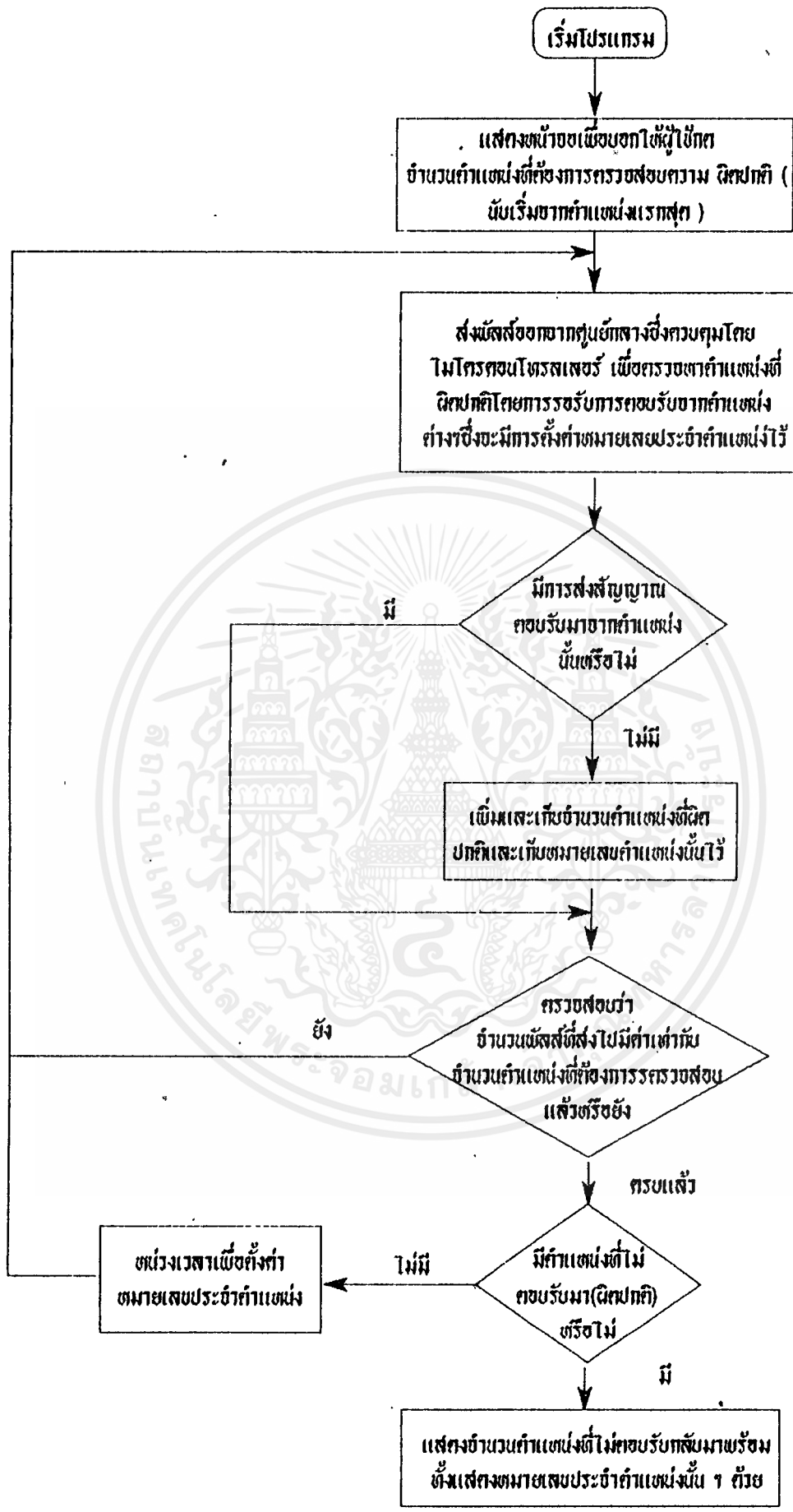
เอกสารอ้างอิง

- [1] น.ศ.ดร. ไพศาล สงวนหมู่ และ รศ. ยืน ภู่วรรณ , " การสื่อสารข้อมูล และไมโครคอมพิวเตอร์เน็ตเวิร์ค " บริษัท ซีเอ็ดดูเคชั่น จำกัด , กรุงเทพมหานคร 10400
- [2] บทความของ วีระ ธรรมจรัส , เรื่อง " การสื่อสารข้อมูลดิจิทัลโดยใช้เทคนิคการรับส่งเนกาทีฟพัลส์ " , จากหนังสือการประชุมวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 11 เล่มที่ 2 หน้า 3-1
- [3] Microprocessor DATA BOOK MCS-51 Microcontrollers
- [4] Kenneth J. Ajala " The 8051 Microcontroller Architecture , Programming , And Applications "

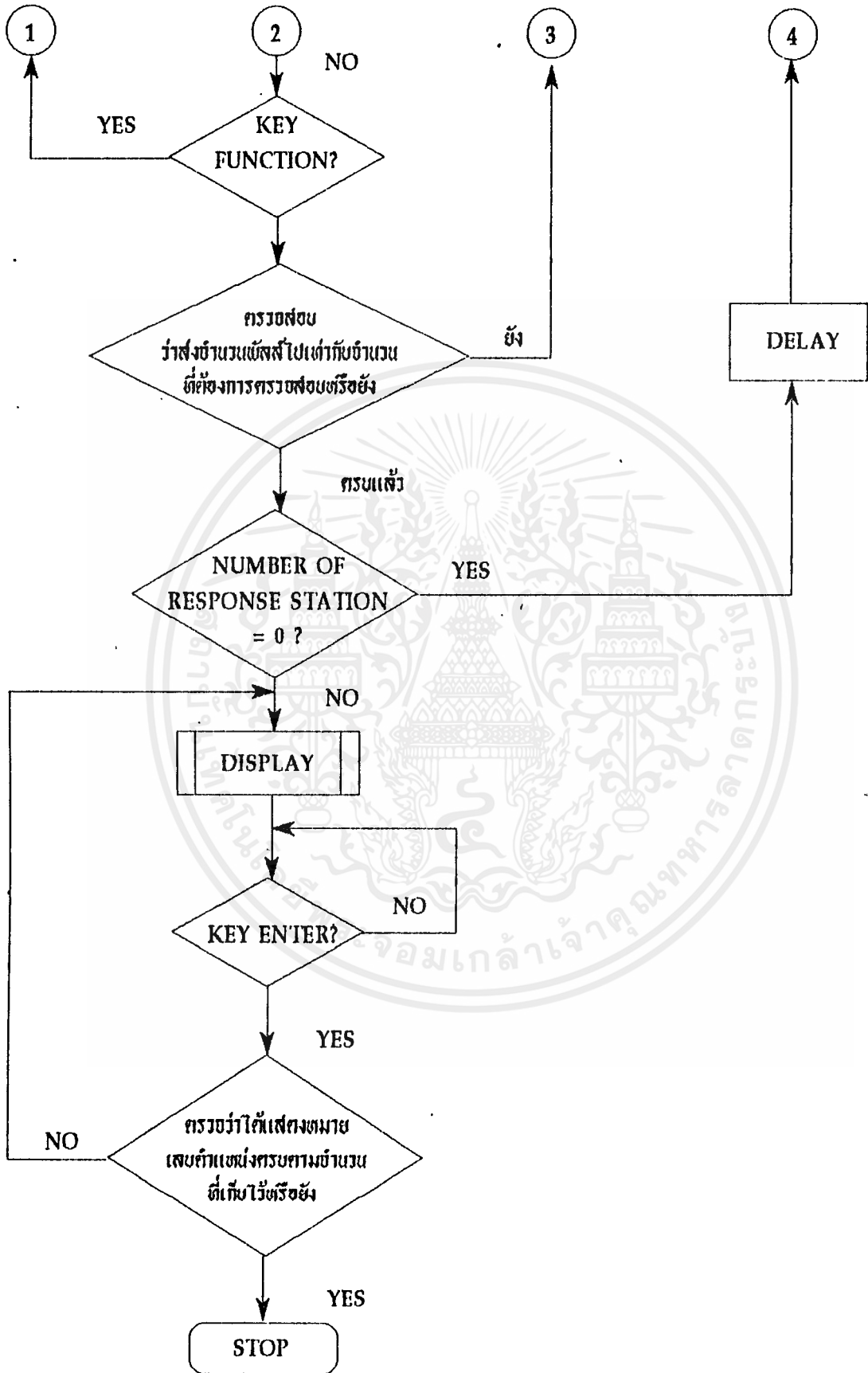
ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้นให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOAD

แสดงคำว่า "LOAD"

RET

NO

แสดงคำว่า " NO. "

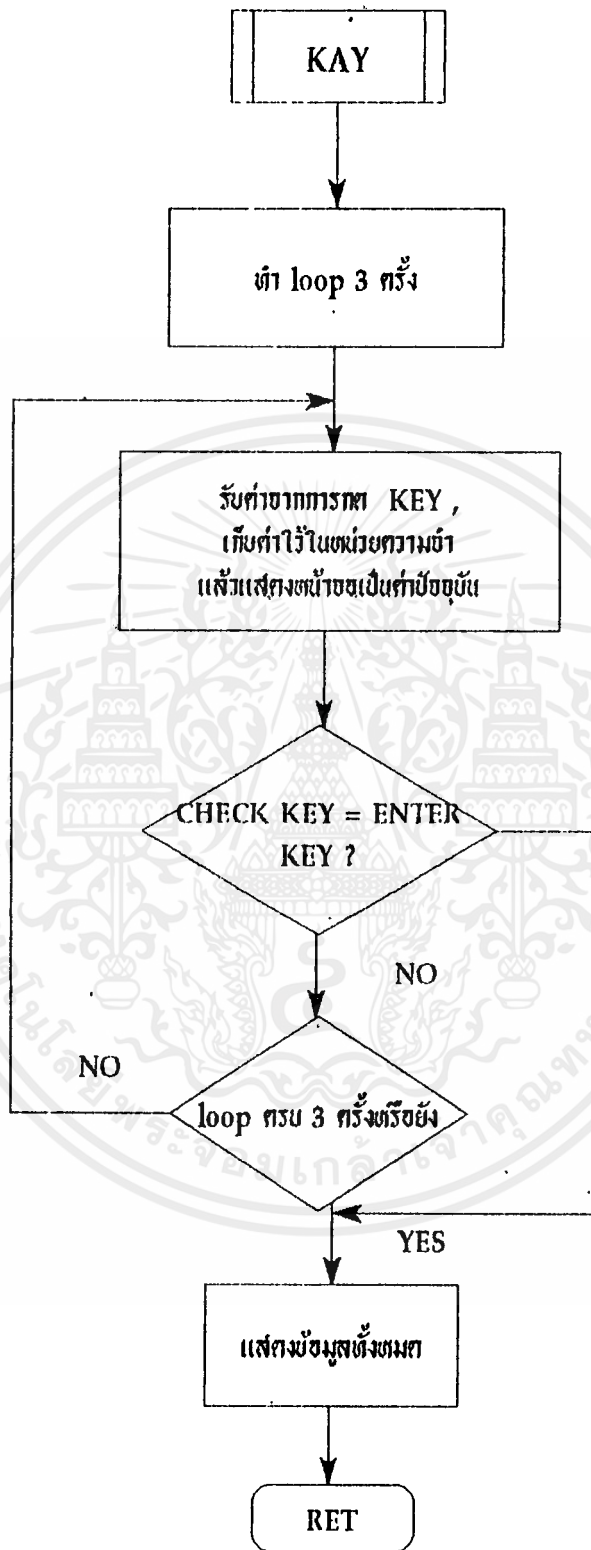
RET

LINE

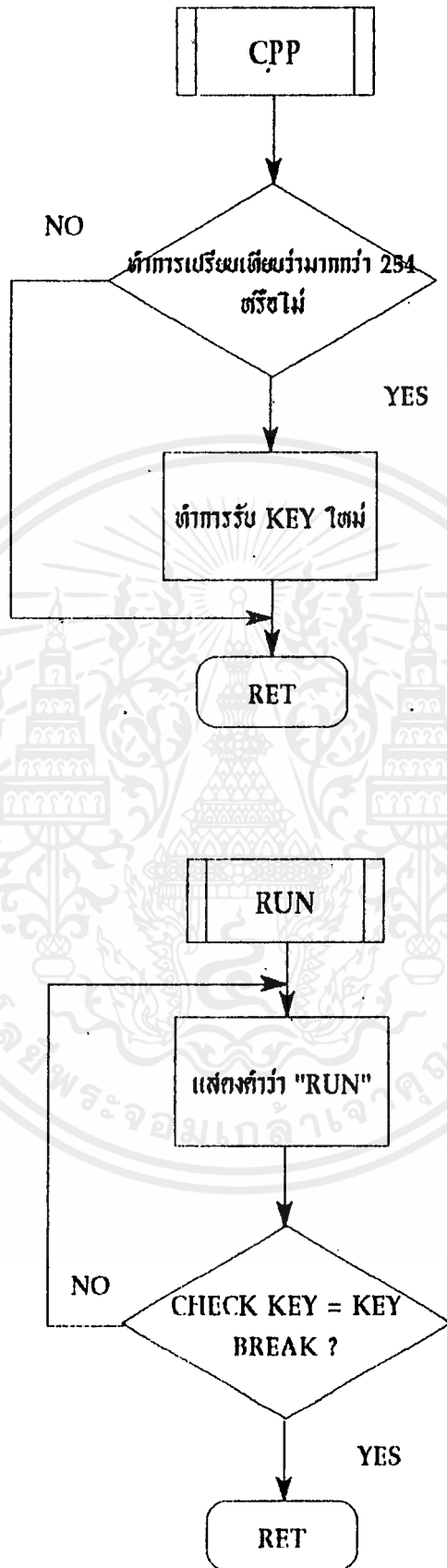
แสดงคำว่า " _ _ _ "

RET

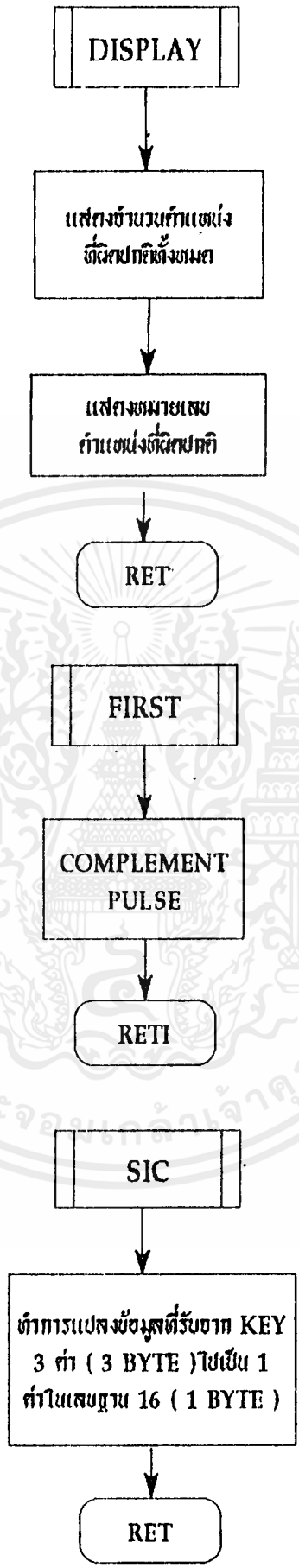
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UPIA      EQU      0F800H
UPIC      EQU      0F802H
UPIP      EQU      0F803H
BIT1      EQU      70H
BIT0      EQU      71H
NUM_PUL   EQU      73H           ;address for save input pulse
NUM_ERR   EQU      74H           ;address for save number error
NO_STAT   EQU      75H           ;address for save no. station
ME        EQU      76H
NAR       EQU      7AH
SHOW3     EQU      7FH           ;address for save bit 3
SHOW2     EQU      7EH           ;address for save bit 2
SHOW1     EQU      7DH           ;address for save bit 1
SHOW0     EQU      7CH           ;address for save bit 0

```

```

ORG      0000H
LJMP     MAIN

ORG      000BH
SJMP     FIRST           ;interrupt routine

```

```

FIRST:   CPL      P1.0           ;complement for generate pulse
         JB       P1.0,BACK       ;if pulse = 1 then return interrupt

         CLR      00H
         INC      NO_STAT         ;count number pulse
         MOV     A,NO_STAT
         CJNE    A,NUM_PUL,BACK   ;if number pulse <> input pulse
                                         ;then return interrupt

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับอาจารย์ผู้สอนเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;goto display

CLR    EA                ;stop generate pulse
MOV    TLO,#0H
MOV    THO,#01H
SETB   TRO                ;start delay 496 microsec
JNB    TFO,$
CLR    TFO

SETB   EA
SETB   ETO
SETB   OOH

MOV    R1,#30H           ;start internal ram for save error
MOV    NUM_ERR,#0FFH
MOV    NO_STAT,#0FFH

MOV    TLO,#0H
MOV    THO,#0A0H
SETB   TRO                ;start generate pulse

BACK:   RETI

DISPLAY1: LJMP    DISPLAY

```

```

MAIN:   MOV    DPTR,#UPIP
        MOV    A,#88H           ;data of control word
        MOVX   @DPTR,A

        MOV    RO,#30H         ;clear data in address

KIKI:   MOV    @RO,#0H
        INC    RO

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับอาจารย์และบุคลากรศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SHUN:      LCALL  LOAD           ;show display "LOAD"
           LCALL  NO           ;show display "NO"
           LCALL  LINE        ;show display "___"

TAP:       MOV    79H,#0H      ;clear memory at address 79H
           MOV    7CH,#3FH     ;data at address 7CH,7DH,7EH
           MOV    7DH,#3FH     ;equal 3FH (0 (segment code))
           MOV    7EH,#3FH
           MOV    7FH,#0H      ;data at address 7FH equal 0H

           LCALL  KAY          ;receive key and show display
           LCALL  CPP          ;compare ,must not greater than 254
           LCALL  SIC          ;change data 3 byte( receive 3 keys)
                               ;to 1 byte

           CLR    P1.0
           SETB   EA
           SETB   ETO
           SETB   OOH

           MOV    R1,#30H
           MOV    NUM_ERR,#0FFH
           MOV    NO_STAT,#0FFH
           INC    NUM_PUL

           MOV    TMOD,#02H     ;set timer0 mode 2
           MOV    TLO,#0H
           MOV    TH0,#0A0H     ;generate pulse frequency 4.8 KHz
           SETB   TRO           ;start generate pulse

RETURN:    JB     00H,*
           MOV    R4,#2H
           MOV    B,#2H

```

CHECK: LCALL DELAY

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JB      P1.1,NOSAVE      ;check response from stations
        DJNZ   B,CHECK          ;check 2 times for normal station

        MOV    @R1,NO_STAT      ;save error station
        INC    R1
        INC    NUM_ERR

        SJMP   CON

NOSAVE:      DEC    R4
             CJNE  R4,#0H,CHECK  ;check 2 times for error station

CON:         SETB   OOH
             SJMP  RETURN

;#####

DELAY:      MOV    R7,#0H      ;delay for check response
WAIT:       INC    R7
             CJNE  R7,#5H,WAIT
             RET

;*****
;PROCEDURE SHOW DISPLAY "LOAD"
;*****

LOAD:       MOV    7FH,#38H    ;data equal "L"
             MOV    7EH,#3FH    ;data equal "0"
             MOV    7DH,#77H    ;data equal "A"
             MOV    7CH,#5EH    ;data equal "d"
             LCALL  EEE          ;show display
             RET

;*****

```

๔๕;PROCEDURE SHOW DISPLAY "NO"ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NO:      MOV      7FH,#0H      ;clear data
        MOV      7EH,#37H     ;data equal "N"
        MOV      7DH,#0BFH    ;data equal "0"
        MOV      7CH,#0H      ;clear data
        LCALL   EEE           ;show display
        RET
```

;PROCEDURE SHOW DISPLAY "___"

```
LINE:   MOV      7FH,#0H      ;clear data
        MOV      7EH,#0H      ;data equal "_"
        MOV      7DH,#8H      ;data equal "_"
        MOV      7CH,#0H      ;data equal "_"
        LCALL   EEE           ;show display
        RET
```

;PROCEDURE RECEIVE KEY AND SHOW DISPLAY

```
KAY:    MOV      R4,#4H       ;receive 4 keys (total enter key)
KEYY1:  MOV      DPTR,#UPIC    ;port C
        MOV      A,#0EEH     ;scanner column begin at "11101110"
        MOV      R6,A        ;pick present data
        MOVX    @DPTR,A      ;output data to port C

        MOV      R0,#0FFH

JAIY:   MOVX    A,@DPTR      ;input data from port C
        MOV      77H,A       ;pick data of A into memory at 77H
        ANL     A,#070H      ;interest 3 bit( 6th, 5th, 4th)
```

เอกสารนี้เป็นเอกสารที่ CJNE สำหรับ A,#070H,SAN การคีย์เข้าไม่ส่งหรือ not การคีย์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OUU1:      MOV      R5,#01H
ERE1:      MOV      R1,#7CH
           MOV      R7,#0BEH

YUY1:      MOV      A,#0H
           MOV      DPTR,#UPIA
           MOVX     @DPTR,A           ;clear data in port A

           MOV      A,R7
           MOV      DPTR,#UPIC       ;select column that show display
           MOVX     @DPTR,A

ASSS1:     MOV      A,@R1
           MOV      DPTR,#UPIA       ;put data in port A
           MOVX     @DPTR,A
           DEC      R3
           CJNE     R3,#0H,ASSS1

           INC      R1                 ;increase pointer R1
           MOV      A,R7
           RL       A                 ;change column that show display
           MOV      R7,A
           CJNE     R1,#80H,YUY1      ;show data in display until R1=80H
           DEC      R5
           CJNE     R5,#0H,ERE1

           MOV      DPTR,#UPIC
           MOV      A,R6
           RL       A                 ;check next column that receive key
           MOV      R6,A
           MOVX     @DPTR,A           ;output data to port C
           LJMP     JAIY              ;out from jump when have key

```

SAN: นี่เป็นเอกสารที่ส่ง MOVX สำหรับ A, @DPTR เพื่อการศึกษา ; have key (sent key) ขั้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      78H,A           ;pick key in memory at address 78H
CJNE    A,#0E7H,P001
MOV      A,#66H          ;data of forth number
P001:    CJNE    A,#0EBH,P002
MOV      A,#4FH          ;data of third number
P002:    CJNE    A,#0EDH,P003
MOV      A,#5BH          ;data of second number
P003:    CJNE    A,#0EEH,P004
MOV      A,#6H           ;data of first number
P004:    CJNE    A,#0D7H,P005
MOV      A,#7FH          ;data of eighth number
P005:    CJNE    A,#0DBH,P006
MOV      A,#7H           ;data of seventh number
P006:    CJNE    A,#0DDH,P007
MOV      A,#7DH          ;data of sixth number
P007:    CJNE    A,#0DEH,P008
MOV      A,#6DH          ;data of fifth number
P008:    CJNE    A,#0B7H,P009
MOV      A,#0H           ;data of enter key
P009:    CJNE    A,#0BBH,P0010
MOV      A,#0F1H         ;data of function key
P0010:   CJNE    A,#0BDH,P0011
MOV      A,#3FH          ;data of zero number
P0011:   CJNE    A,#0BEH,ASS
MOV      A,#6FH          ;data of nineth number
ASS:     MOV      79H,A           ;pick in segment code
CJNE    R4,#1H,JERRY    ;loop equal 4th loop
CJNE    A,#0H,TOM       ;check enter key
LJMP    EEE             ;show display of key
TOM:     MOV      R4,#4H           ;set new loop
MOV      7CH,#3FH        ;set initial data
MOV      7DH,#3FH        ;set initial data
MOV      7EH,#3FH        ;set initial data
MOV      7FH,#0H         ;set initial data

```

เอกสารนี้เป็นเอกสารที่ LAMP JINN ใช้สำหรับทำงานเพื่อการวิจัยและพัฒนาโปรแกรมที่ใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JERRY:      CJNE    A,#0H,JINN      ;loop not equal 4th loop and check
                                           ;enter key
                LJMP   EEE          ;show display

JINN:       MOV     7FH,#0H
                MOV     7EH,7DH      ;transfer data in memory
                MOV     7DH,7CH
                MOV     7CH,79H      ;transfer new data

OUU:        MOV     R5,#0FH
EREE:       MOV     R1,#7CH
                MOV     R7,#0EEH

YUY:        MOV     A,#0H
                MOV     DPTR,#UPIA
                MOVX    @DPTR,A      ;clear data in port A
                MOV     A,R7
                MOV     DPTR,#UPIC
                MOVX    @DPTR,A      ;input data in port c

ASS1:       MOV     R3,#0FFH
                MOV     A,@R1
                MOV     DPTR,#UPIA   ;input data port A
                MOVX    @DPTR,A
                DEC     R3
                CJNE    R3,#0H,ASS1

                INC     R1           ;increase pointer
                MOV     A,R7
                RL      A            ;change column that show display
                MOV     R7,A
                CJNE    R1,#80H,YUY
                DEC     R5
                CJNE    R5,#0H,EREE

```

```

MOV     A,#0H
MOV     DPTR,#UPIA      ;clear port A
MOVX    @DPTR,A

MOV     A,78H
MOV     DPTR,#UPIC      ;pick column that find key put port C
MOVX    @DPTR,A        ;in order to check delete key

MOVX    A,@DPTR
CJNE    A,78H,JAN      ;check delete key
LJMP    OUU            ;show display that not change data
JAN:    DEC     R4
CJNE    R4,#0H,KEY    ;receive key equal 4 keys?
LJMP    EEE          ;equal 4 keys jump to show display
KEY:    LJMP    KEYY1

EEE:    MOV     B,#01H
EEE1:   MOV     R5,#0FFH
TEREE:  MOV     R1,#7CH
        MOV     R7,#0EEH

TYUY:   MOV     A,#0H
        MOV     DPTR,#UPIA
        MOVX    @DPTR,A      ;clear data port A

        MOV     A,R7
        MOV     DPTR,#UPIC
        MOVX    @DPTR,A      ;input data in port C

        MOV     R3,#0FFH
TASS1:  MOV     A,@R1
        MOV     DPTR,#UPIA
        MOVX    @DPTR,A      ;input data in port A (7 segment)
        DEC     R3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CJNE    R3,#0H,TASS1

INC     R1                ;increase pointer (index data port A)
MOV     A,R7
RL      A                 ;change column to show display
MOV     R7,A
CJNE    R1,#80H,TYUY     ;show 4 column
DEC     R5
CJNE    R5,#0H,TEREE
DJNZ    B,REE1
MOV     A,#0H
MOV     DPTR,#UPIA
MOVX    @DPTR,A          ;clear port A
RET

```

```

CPP:    MOV     A,7EH      ;at hundred position
        CJNE    A,#3FH,GTO ;compare data equal 0 ?
        LJMP    EEE      ;show display
GTO:    CJNE    A,#6H,GT1  ;compare data equal 1 ?
        LJMP    EEE      ;show display
GT1:    CJNE    A,#5BH,NWW ;compare data equal 2 ?

        MOV     A,7DH      ;at decimal position
        CJNE    A,#3FH,GOTO ;compare data equal 0 ?
        LJMP    EEE      ;show display
GOTO:   CJNE    A,#6H,GOT1  ;compare data equal 1 ?
        LJMP    EEE      ;show display
GOT1:   CJNE    A,#5BH,GOT2  ;compare data equal 2 ?
        LJMP    EEE      ;show display
GOT2:   CJNE    A,#4FH,GOT3  ;compare data equal 3 ?
        LJMP    EEE      ;show display
GOT3:   CJNE    A,#66H,GOT4  ;compare data equal 4 ?
        LJMP    EEE      ;show display

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์และบุคลากรในมหาวิทยาลัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GOT4:      CJNE    A,#6DH,NWW      ;compare data equal 5 ?

           MOV     A,7CH          ;at unit position
           CJNE    A,#3FH,GOTO0   ;compare data equal 0 ?
           LJMP    EEE           ;show display
GOTO0:     CJNE    A,#6H,GOOT1    ;compare data equal 1 ?
           LJMP    EEE           ;show display
GOOT1:     CJNE    A,#5BH,GOOT2   ;compare data equal 2 ?
           LJMP    EEE           ;show display
GOOT2:     CJNE    A,#4FH,GOOT3   ;compare data equal 3 ?
           LJMP    EEE           ;show display
GOOT3:     CJNE    A,#66H,NWW     ;compare data equal 4 ?
           LJMP    EEE           ;show display

NWW:       MOV     7FH,#0H        ;data equal display " "
           MOV     7EH,#40H       ;data equal display "- "
           MOV     7DH,#79H       ;data equal display "E"
           MOV     7CH,#40H       ;data equal display "- "
           LCALL   EEE           ;show display "-E-"
           AJMP   TAP            ;receive new total key

NUS:       RET

;*****
;PROCEDURE CALCULATE UNIT POSITION
;*****

SOT1:      MOV     NUM_PUL,#0H    ;clear memory at address 73H
SOS1:      MOV     R3,NUM_PUL
           INC     NUM_PUL
           MOV     A,NUM_PUL
           LCALL   WAN
           CJNE    A,7CH,SOS1     ;compare
           MOV     NAR,R3        ;unit position in heximal base
           RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;PROCEDURE CALCULATE DECIMAL POSITION
;*****

```

```

SOT2:      MOV     NUM_PUL,#0H
SOS2:      MOV     R3,NUM_PUL
           INC     NUM_PUL
           MOV     A,NUM_PUL
           LCALL  WAN
           CJNE   A,7DH,SOS2
           MOV     R4,#0H
SOTT2:     MOV     B,#0AH
STT2:      INC     R4           ;when procedure end ,R4 will pick
                               ;decimal in heximal base
           DJNZ   B,STT2
           DEC     R3
           CJNE   R3,#0H,SOTT2
           RET

```

```

;*****
;PROCEDURE CALCULATE HUNDRED POSITION
;*****

```

```

SOT3:      MOV     NUM_PUL,#0H
SOS3:      MOV     R3,NUM_PUL
           INC     NUM_PUL
           MOV     A,NUM_PUL
           LCALL  WAN
           CJNE   A,7EH,SOS3
           MOV     R1,#0H
SOTT3:     MOV     R6,#0AH
SOTT3:     MOV     B,#0AH
STT3:      INC     R1           ;when completely end ,R1 will pick
                               ;hundred position in heximal base

```

เอกสารนี้เป็นเอกสารที่ DJNZ สำหรับ B, STT3 งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEC     R6
CJNE   R6,#0H,SOTT3
DEC     R3
CJNE   R3,#0H,SOTTT3
RET

```

```

;*****
;PROCEDURE MAIN CALCULATE EVERY POSITION
;*****

```

```

SIC:    MOV     R4,#0H
        MOV     R1,#0H
        MOV     A,7EH
        CJNE   A,#3FH,G001
        MOV     A,7DH
        CJNE   A,#3FH,G002
        LCALL  SOT1
        LJMP   DDD

G002:   LCALL  SOT1
        LCALL  SOT2
        LJMP   DDD

G001:   LCALL  SOT1
        LCALL  SOT2
        LCALL  SOT3

DDD:    MOV     A,NAR
        CLR    C
        ADDC  A,R4
        ADDC  A,R1

        MOV     NUM_PUL,A      ;presently pick 1 byte in heximal
                                   ;base

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WAN:      MOV     A,@A+PC
          RET
          DB     3FH      ;0
          DB     06H      ;1
          DB     5BH      ;2
          DB     4FH      ;3
          DB     66H      ;4
          DB     6DH      ;5
          DB     7DH      ;6
          DB     07H      ;7
          DB     7FH      ;8
          DB     6FH      ;9
```

;PROCEDURE FOR DISPLAY ERROR

```
DISPLAY:  MOV     7AH,NUM_ERR    ;save number error in memory

          LCALL  SAVE          ;data for display number error bit2
          LCALL  SA_NUM        ;data for display bit0,1
          LCALL  SHOW          ;convert decimal to hex for bit0,1,2
          MOV    SHOW3,#79H    ;set data for display bit3

          LCALL  MOVEIN        ;save all register
          LCALL  ENTER         ;check enter and display data
          LCALL  MOVEOUT       ;moveout all register
```

```
AGAIN:   MOV     R0,#30H      ;start address for display
          MOV     R7,#0H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SEE:      INC      R7
          INC      R0
          MOV      7BH,@R0
          PUSH    7BH

          LCALL   KEEP           ;data for display no. station bit2
          LCALL   SA_STAT        ;data for display bit0,1
          LCALL   SHOW           ;convert decimal to hex for bit0,1,2
          MOV     SHOW3,#5CH     ;set data for display bit3

          LCALL   MOVEIN        ;save all register
          LCALL   ENTER         ;check enter and display data
          LCALL   MOVEOUT       ;moveout all register

          POP     7BH
          MOV     @R0,7BH

          MOV     A,R7
          CJNE   A,7AH,SEE      ;display = number error ?
          LJMP   AGAIN         ;goto loop display again

```

```

;*****
;PROCEDURE FOR SET DATA NUMBER ERROR BIT0,1
;*****

```

```

SA_NUM:   PUSH    7FH
          MOV     A,NUM_ERR
          SWAP   A           ;4 bit high <-> 4 bit low
          MOV     NUM_ERR,A
          MOV     A,#0FH
          ANL    A,NUM_ERR
          MOV     7FH,A

          MOV     A,NUM_ERR

```

เอกสารนี้เป็นเอกสารที่ SWAP ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                DA      A          ;convert binary to decimal
                MOV     NUM_ERR,A  ;save number error

                MOV     R2,#0H

PLUS:           MOV     A,R2
                CJNE   A,7FH,WELL
                SJMP   US

WELL:           MOV     A,#6H
                ADD    A,NUM_ERR
                DA      A          ;convert binary to decimal
                MOV     NUM_ERR,A
                INC    R2
                SJMP   PLUS

US:             MOV     BIT0,NUM_ERR ;set data for bit0,1
                POP    7FH
                RET

;*****
;PROCEDURE FOR SET DATA NUMBER ERROR BIT2
;*****

SAVE:           MOV     BIT1,#0H
                MOV     R6,#0H

LOOP:           INC    R6
                MOV     A,R6
                CJNE   A,7AH,LD     ;if number < 99 then set bit2 = 0

                ;+++++
                SJMP   TURN1
                ;+++++

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD:          CJNE    A,#064H,LOOP
             INC     BIT1          ;if 99< number <199 then set bit2 = 1

KNOW:       INC     R6
             MOV     A,R6
             CJNE   A,7AH,SAD

;+++++
             SJMP   TURN1
;+++++

SAD:        CJNE    A,#0C8H,KNOW
             INC     BIT1          ;if number >199 then set bit2 = 2

TURN1:     RET

;*****
;PROCEDURE FOR SET DATA NO. STATION BIT0,1
;*****

SA_STAT:   PUSH    7FH

             MOV     A,@R0
             SWAP   A             ;4 bit high <-> 4 bit low
             MOV     @R0,A
             MOV     A,#0FH
             ANL    A,@R0
             MOV     7FH,A

             MOV     A,@R0
             SWAP   A
             DA     A             ;conver binary to decimal
             MOV     @R0,A

```

เอกสารนี้เป็นเอกสารที่ **MOV** ไว้สำหรับ **R2, #0H** งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PP:      MOV      A,R2
        CJNE     A,7FH,WW
        SJMP     THEN
WW:      MOV      A,#6H
        ADD     A,@R0
        DA      A
        MOV     @R0,A
        INC     R2
        SJMP     PP

THEN:    MOV      BIT0,@R0      ;set data no.station bit0,i
        POP     7FH
        RET

;*****
;PROCEDURE FOR SET DATA NO.STATION ERROR BIT2
;*****

KEEP:    MOV      BIT1,#0H
        MOV     R6,#0H

LL:      INC     R6
        MOV     A,R6
        MOV     ME,@R0
        CJNE   A,ME,DD      ;if no.station < 99 then bit2 = 0

        ;+++++
        SJMP    TURN2
        ;+++++

DD:      CJNE   A,#064H,LL      ;if 99< no.station <199 then bit2 = 1
        INC     BIT1

OO:      INC     R6
        MOV     A,R6

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรในหน่วยงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     ME, @R0
CJNE   A, ME, BB

;+++++
S JMP   TURN2
;+++++

BB:     CJNE   A, #0C8H, 00    ;if no.station > 199 then bit2 = 2
INC     BIT1

TURN2:  RET

;*****
;PROCEDURE CONVERT DATA FOR DISPLAY
;*****

SHOW:   MOV     A, #0FH
        ANL     A, BIT0
        LCALL   CONVERT
        MOV     SHOW0, A      ;data for display bit0

        MOV     A, BIT0
        SWAP   A
        MOV     BIT0, A
        MOV     A, #0FH
        ANL     A, BIT0
        LCALL   CONVERT
        MOV     SHOW1, A      ;data for display bit1

        MOV     A, #0FH
        ANL     A, BIT1
        LCALL   CONVERT
        MOV     SHOW2, A      ;data for display bit2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;*****

;PROCEDURE CHECK ENTER KEY

;*****

```
ENTER:      MOV      DPTR,#UPIC      ;port C
            MOV      A,#0KEH      ;scanner column
            MOV      R6,A          ;pick present data
            MOVX     @DPTR,A       ;output data to port C

JAI2:      MOVX     A,@DPTR        ;input data from port C
            MOV      77H,A
            ANL      A,#070H
            CJNE     A,#070H,SN2   ;have input key that send or not

OOLO:      MOV      R4,#01FH
            LCALL    SCAN          ;display data
            DEC      R4
            CJNE     R4,#0H,OOLO   ;count for scan display

            MOV      DPTR,#UPIC
            MOV      A,R6
            RL       A              ;check next column
            MOV      R6,A
            MOVX     @DPTR,A       ;output data to port C (determine column)
            LJMP     JAI2

SN2:       MOVX     A,@DPTR        ;have key (sent key)
            MOV      78H,A
            CJNE     A,#0E7H,P012
            MOV      A,#66H        ;data of forth number

P012:      CJNE     A,#0EBH,P022
            MOV      A,#4FH        ;data of third number

P022:      CJNE     A,#0EDH,P032
            MOV      A,#5BH        ;data of second number
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

P032:      CJNE    A,#0EEH,P042
           MOV     A,#6H           ;data of first number
P042:      CJNE    A,#0D7H,P052
           MOV     A,#7FH           ;data of eighth number
P052:      CJNE    A,#0DBH,P062
           MOV     A,#7H           ;data of seventh number
P062:      CJNE    A,#0DDH,P072
           MOV     A,#7DH           ;data of sixth number
P072:      CJNE    A,#0DEH,P082
           MOV     A,#6DH           ;data of fifth number
P082:      CJNE    A,#0B7H,P092
           MOV     A,#0H           ;data of enter key
P092:      CJNE    A,#0BBH,P0102
           MOV     A,#0F1H          ;data of function key
P0102:     CJNE    A,#0BDH,P0112
           MOV     A,#3FH           ;data of zero number
P0112:     CJNE    A,#0BEH,SS
           MOV     A,#6FH           ;data of nineth number
SS:        CJNE    A,#0H,MA1       ;compare equal enter key ?
AI:        RET
MA1:      LJMP     ENTER

```

```

SCAN:      MOV     B,#050H
CO:        MOV     A,SHOWO
           MOV     DPTR,#UPIA
           MOVX    @DPTR,A         ;input data for bit0 in port a
           MOV     DPTR,#UPIC
           MOV     A,#0FEH         ;set port c for display bit0
           MOVX    @DPTR,A
           DJNZ   B,CO

```

เอกสารนี้เป็นเอกสารที่ **MOV** ไว้สำหรับ **B,#050H** งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

C1:      MOV     A,SHOW1
          MOV     DPTR,#UPIA
          MOVX    @DPTR,A      ;input data for bit1 in port a
          MOV     DPTR,#UPIC
          MOV     A,#0FDH      ;set port c for display bit1
          MOVX    @DPTR,A
          DJNZ    B,C1

          MOV     B,#050H

C2:      MOV     A,SHOW2
          MOV     DPTR,#UPIA
          MOVX    @DPTR,A      ;input data for bit2 in port a
          MOV     DPTR,#UPIC
          MOV     A,#0FBH      ;set port c for display bit2
          MOVX    @DPTR,A
          DJNZ    B,C2

          MOV     B,#050H

C3:      MOV     A,SHOW3
          MOV     DPTR,#UPIA
          MOVX    @DPTR,A      ;input data for bit3 in port a
          MOV     DPTR,#UPIC
          MOV     A,#0F7H      ;set port c for display bit3
          MOVX    @DPTR,A
          DJNZ    B,C3

          RET

CONVERT: INC     A              ;shift from "ret" = 1 address
          MOVC    A,@A+PC      ;for move 0-9 to display
          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ 3FH ซึ่งงานเพื่อการศึกษาเท่านั้น; 0 อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB	06H	;1
DB	5BH	;2
DB	4FH	;3
DB	66H	;4
DB	6DH	;5
DB	7DH	;6
DB	07H	;7
DB	7FH	;8
DB	6FH	;9

MOVEIN:

MOV	68H, R0
MOV	69H, R1
MOV	6AH, R2
MOV	6BH, R3
MOV	6CH, R4
MOV	6DH, R5
MOV	6EH, R6
MOV	6FH, R7

RET

MOVEOUT:

MOV	R0, 68H
MOV	R1, 69H
MOV	R2, 6AH
MOV	R3, 6BH
MOV	R4, 6CH
MOV	R5, 6DH
MOV	R6, 6EH
MOV	R7, 6FH

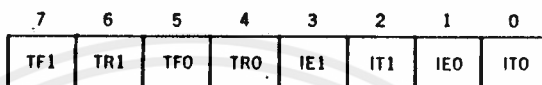
RET

END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COUNTER และ TIMER

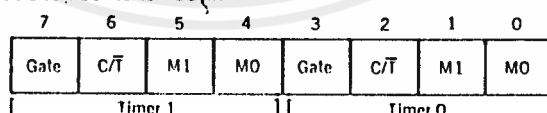
การประยุกต์ ใช้ไมโครคอนโทรลเลอร์มักจะใช้นับเหตุการณ์ภายนอก เช่น ความถี่ของ ขบวนการพัลส์ หรือกำเนิดเวลาหน่วงได้อย่างถูกต้องระหว่างการทำงานทางคอมพิวเตอร์ ซึ่งสามารถใช้เทคนิคทางซอฟต์แวร์ได้ แต่ในการควบคุมโดยใช้ซอฟต์แวร์ ก็จะทำให้กระบวนการส่วนอื่นทำงานไม่ได้ จึงแก้ปัญหาโดยใช้เคาน์เตอร์ขนาด 16 บิต 2 ตัว คือ T0 และ T1 ซึ่งสามารถโปรแกรมให้เป็นเคาน์เตอร์นับพัลส์ภายนอก หรือ เป็นไทม์เมอร์นับสัญญาณนาฬิกาภายใน



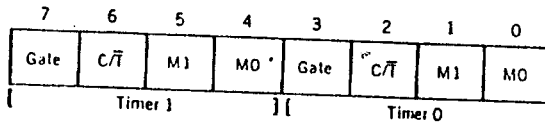
THE TIMER CONTROL (TCON) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7	TF1	Timer 1 Overflow flag. Set when timer rolls from all ones to zero. Cleared when processor vectors to execute interrupt service routine located at program address 0018h.
6	TR1	Timer 1 run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.
5	TFO	Timer 0 Overflow flag. Set when timer rolls from all ones to zero. Cleared when processor vectors to execute interrupt service routine located at program address 0008h.
4	TRO	Timer 0 run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.
3	IE1	External interrupt 1 edge flag. Set to 1 when a high to low edge signal is received on port 3 pin 3.3 (INT1). Cleared when processor vectors to interrupt service routine located at program address 0013h. Not related to timer operations.
2	IT1	External interrupt 1 signal type control bit. Set to 1 by program to enable external interrupt 1 to be triggered by a falling edge signal. Set to 0 by program to enable a low level signal on external interrupt 1 to generate an interrupt.
1	IE0	External interrupt 0 edge flag. Set to 1 when a high to low edge signal is received on port 3 pin 3.2 (INT0). Cleared when processor vectors to interrupt service routine located at program address 0003h. Not related to timer operations.
0	IT0	External interrupt 0 signal type control bit. Set to 1 by program to enable external interrupt 0 to be triggered by a falling edge signal. Set to 0 by program to enable a low level signal on external interrupt 0 to generate an interrupt.

Bit addressable as TCON.0 to TCON.7



รูป ก.1 TCON Function Register



THE TIMER MODE CONTROL (TMOD) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7/3	Gate	OR gate enable bit which controls RUN/STOP of timer 1/0. Set to 1 by program to enable timer to run if bit TR1/0 in TCON is set and signal on external interrupt INT1/0 pin is high. Cleared to 0 by program to enable timer to run if bit TR1/0 in TCON is set.
6/2	C/T	Set to 1 by program to make timer 1/0 act as a counter by counting pulses from external input pins 3.5 (T1) or 3.4 (T0). Cleared to 0 by program to make timer act as a timer by counting internal frequency.
5/1	M1	Timer/counter operating mode select bit 1. Set/cleared by program to select mode.
4/0	M0	Timer/counter operating mode select bit 0. Set/cleared by program to select mode.

M1	M0	Mode
0	0	0
0	1	1
1	0	2
1	1	3

TMOD is not bit addressable

TMOD Function Register

เคาน์เตอร์จะถูกแบ่งเป็นรีจิสเตอร์ขนาด 8 บิต 2 ตัว เป็น ไบท์ (TL0, TL1) และ ไบท์สูง (TH0, TH1) การทำงานของเคาน์เตอร์สามารถกำหนดได้จากบิตในรีจิสเตอร์ที่ใช้ควบคุมโหมด (TMOD) ควบคุมว่าเป็นเคาน์เตอร์หรือไทม์เมอร์ และคำสั่งของโปรแกรม

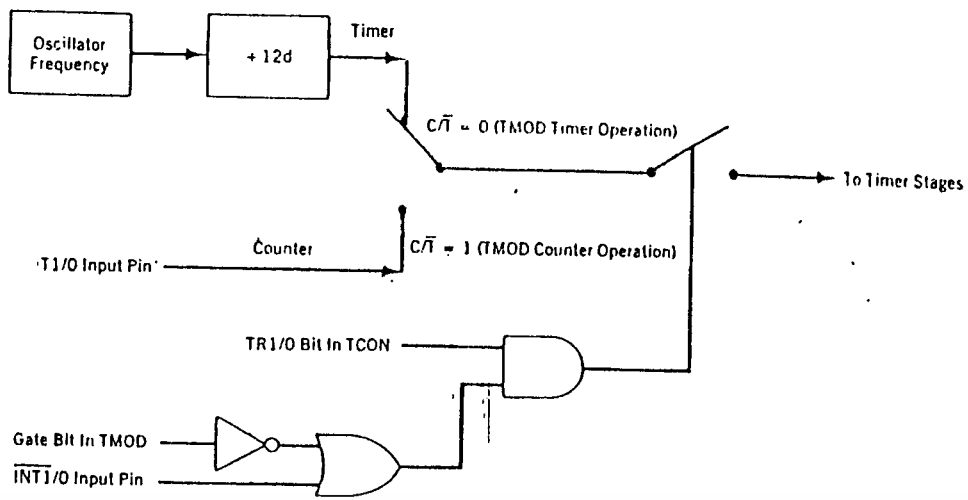
TMOD สามารถแบ่งเป็นคูรีจิสเตอร์ 4 บิต สำหรับ ไทม์เมอร์ 2 ตัว TCON จะมีบิตควบคุมและแฟลกของไทม์เมอร์ ใน 4 บิตบน และมีบิตควบคุมและแฟลกของอินเทอร์รัพท์ภายนอกใน 4 บิตล่าง รูป ก.1 แสดงลักษณะของ TMOD และ TCON

Timer Counter Interrupts

เคาน์เตอร์จะถูกรวมอยู่บนชิพเพื่อช่วยในกระบวนการนับ เมื่อต้องการให้โปรแกรมนับจำนวนพัลส์ภายในหรือเหตุการณ์ภายนอก ก็เขียนจำนวนนั้นไปที่เคาน์เตอร์ จำนวนนั้น ๆ จะแทนการนับที่มากที่สุดที่น้อยกว่าค่าที่ต้องการอยู่ 1 พัลส์ เคาน์เตอร์จะเพิ่มขึ้นจากค่าเริ่มต้นเป็นค่าที่มากที่สุดแล้วจะกลายเป็น 0 หมดที่พัลส์ลูกสุดท้าย และจะไปเซ็ตแฟลกของไทม์เมอร์ด้วย เงื่อนไขของแฟลกอาจทดสอบได้โดย คำสั่งที่ให้โปรแกรมทำการนับจนสำเร็จ หรืออาจใช้แฟลกในการอินเทอร์รัพท์โปรแกรมก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ก.2 Timer / Counter Control Logic

Timing

ถ้าเคาน์เตอร์ถูกโปรแกรมให้เป็นไทม์เมอร์ มันจะนับสัญญาณนาฬิกาภายในของที่ความถี่ วงจรออสซิลเลเตอร์หารด้วย 12d เช่นถ้าค่าความถี่คริสตอลเป็น 6 MHz แล้วสัญญาณนาฬิกาที่ใช้เป็น ไทม์เมอร์จะมีความถี่ 500 kHz

จากวงจรในรูปที่ ก.2 การที่จะให้สัญญาณนาฬิกาผ่านไปยังไทม์เมอร์ได้ จะต้องกำหนดให้ บิต C/T ในรีจิสเตอร์ TMOD เป็น 0 เพื่อให้เป็นไทม์เมอร์ แล้วเซตบิต TRX ใน TCON ให้เป็น 1 และบิตเกตใน TMOD เป็น 0 หรือภายนอก INTX เป็น 1 จะพูดอีกอย่างก็คือ ถ้าเคาน์เตอร์ ถูกทำให้เป็นไทม์เมอร์แล้ว สัญญาณพัลส์ผ่านไปที่เคาน์เตอร์ได้โดย บิตที่ใช้รัน และบิตเกต หรือบิต อินพุตภายนอก INTX

Timer Modes of Operation

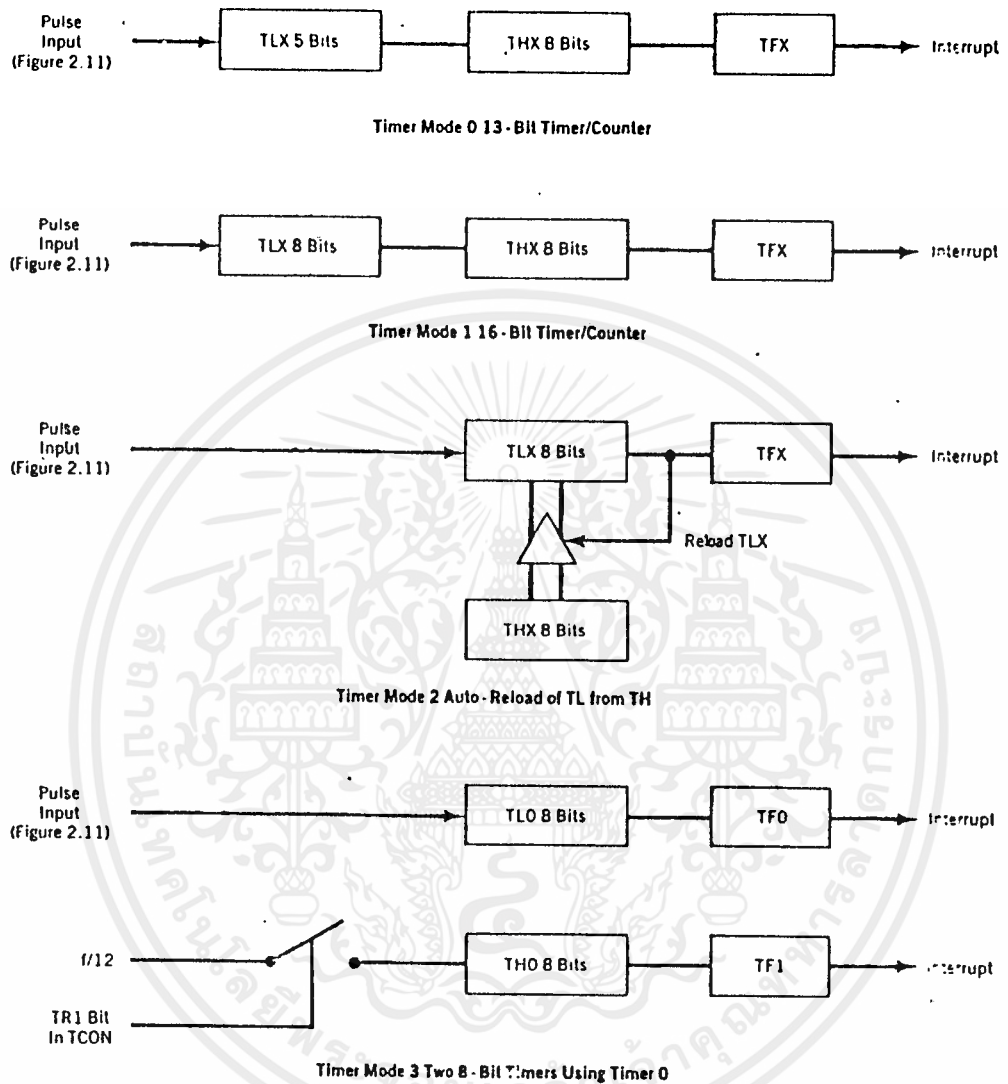
ไทม์เมอร์มีการทำงาน 4 โหมด ซึ่งกำหนดโดยบิต M1 และ M0 ใน TMOD รูป ก.3 แสดงถึงไทม์เมอร์ทั้ง 4 โหมด

Timer Mode 0

การเซตไทม์เมอร์โดยบิตโหมดใน TMOD เป็น 00b จะทำให้รีจิสเตอร์ THX เป็นเคาน์เตอร์ขนาด 8 บิต และ TLX เป็นเคาน์เตอร์ขนาด 5 บิต พัลส์อินพุตจะถูกหารด้วย 32d ใน TL

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ขออนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น TH จะนับความถี่ออสซิลเลเตอร์ที่ถูกหารด้วย 384d ตัวอย่างเช่นความถี่ออสซิลเลเตอร์ 6MHz ทำให้ความถี่สุดท้ายที่ TH เป็น 15625 Hz แพลกของไทม์เมอร์จะถูกเซตเมื่อ THX เปลี่ยนจาก FFh เป็น 00h หรือใน 0.0164 วินาที สำหรับคริสตอล 6MHz ที่ THX เริ่มนับจาก 00h



รูป ก.3 Timer 1 and Timer 0 Operation Mode

Timer Mode 1

โหมด 1 คล้ายกับโหมด 0 ,เพียงแต่ TLX ถูกทำให้นับได้ทั้ง 8 บิต เมื่อเซตบิตโหมดใน TMOD เป็น 01b แพลกของไทม์เมอร์จะถูกเซตในเวลา 0.1311 วินาที สำหรับคริสตอล 6 MHz

Timer Mode 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการเซ็ตบิตโหมดใน *TMOD* เป็น 10b แล้วไทม์เมอร์นี้จะใช้เพียง *TLX* เป็นเคาน์เตอร์ขนาด 8 บิต ส่วน *THX* จะใช้เก็บค่าที่จะไหลค้เข้าไปใน *TLX* เกิดโอเวอร์โฟลว์จาก *FFh* เป็น *00h* แพลกจะถูกเซ็ตด้วยโหมดนี้จะเป็นลักษณะของ *auto-reload* โดย *TLX* จะนับขึ้นจากค่าที่อยู่ใน *THX* จนเกิดโอเวอร์โฟลว์ แล้วเริ่มใหม่ด้วยค่าที่อยู่ใน *THX* เช่น เก็บค่า *9Ch* ไว้ใน *THX* และใช้คริสตอล 6 MHz จะกินเวลา 10.0002 วินาที แล้วโอเวอร์โฟลว์แพลกก็จะถูกเซ็ต

Timer Mode 3

Timer 0 และ 1 อาจจะถูกโปรแกรมให้ทำงานในโหมด 0, 1, หรือ 2 ได้โดยไม่ขึ้นกับไทม์เมอร์อีกตัวหนึ่ง แต่สำหรับโหมด 3 มันจะไม่เป็นเช่นนั้น ถ้าไทม์เมอร์ 0 ถูกเลือกให้ทำงานในโหมด 3 แล้วให้ไทม์เมอร์ 1 ทำงานในโหมด 3 ด้วย จะเป็นการหยุดนับซึ่งบิตควบคุม *TR1* และแพลกของไทม์เมอร์ 1 (*TF1*) จะถูกใช้โดยไทม์เมอร์ 0

Timer 0 ในโหมด 3 แยกเป็นเคาน์เตอร์ขนาด 8 บิตได้ 2 ตัว คือ *TL0* ที่ถูกควบคุมโดยเกตตามรูป ก.2 และจะเซ็ต *Timer Flag TF0* ทุกครั้งที่เกิดโอเวอร์โฟลว์จาก *FFh* เป็น *00h* ส่วน *TH0* จะรับสัญญาณนาฬิกา (ซึ่งถูกหารด้วย 12 แล้ว) ภายใต้การควบคุมของ *TR1* เท่านั้น แต่จะเซ็ต *TF1* เมื่อเกิดโอเวอร์โฟลว์ขึ้น

Timer 1 ซึ่งจะใช้ทำงานในโหมด 0, 1, และ 2 ขณะที่ Timer 0 อยู่ในโหมด 3 Timer 1 จะไม่สามารถสร้างสัญญาณอินเทอร์พท์ได้ ขณะที่ Timer 0 ใช้ *TF1* อยู่ ต้องทำการเลือกให้ Timer 1 ทำงานในโหมด 3 จึงจะหยุดการทำงานได้ (และจะยังจำค่าที่นับไว้ใน *Timer 1*) Timer 1 อาจใช้สร้าง *baudrate* ของพอร์ตอนุกรม หรือทำงานในโหมด 0, 1, และ 2 ที่ไม่ต้องการใช้อินเทอร์พท์ (ใช้ *TF1*) ในงานนั้น

Counting

ความแตกต่างของ *counting* และ *timing* คือแหล่งสัญญาณนาฬิกาที่ใช้ นับ เมื่อใช้ในงาน Timer จะได้สัญญาณนาฬิกาจากออสซิลเลเตอร์ผ่านวงจรถหาร 12 และเมื่อใช้เป็น *counter* ขา *T0 (P3.4)* จะจ่ายพัลส์ให้กับ Timer 0 และขา *T1 (P3.5)* จ่ายให้ timer 1 และต่อบิต *C/T* ใน *TMOD* ให้เป็น 1 เพื่อให้พัลส์จากขา *TX* ผ่านไปยังวงจรถควบคุมตามรูป ก.2 ได้

พัลส์อินพุทบนขา *TX* จะถูกแซมเปิลที่ช่วง *P2* ของ *state 5* ทุกๆ แมกซ์ซิมัซเซเคิล การ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนค่าบนขาอินพุทจากสูงไปต่ำ ระหว่างการแชนเบิ้ลจะไปเพิ่มค่าเคาน์เตอร์ ค่า High หรือ Low นี้ จะถูกคงค่าไปอีกอย่างน้อยหนึ่งแมชชีนไซเคิล เพื่อให้แน่ใจว่าการนับถูกต้อง เพราะเมื่อผ่านไป 24 พัลส์ ความถี่อินพุทที่มากที่สุด จะถูกนับอย่างถูกต้องคือเป็นความถี่ออสซิลเลเตอร์ที่ถูกหารด้วย 24 ถ้าใช้คริสตัล 6 MHz แต่ความถี่ภายนอกที่มากที่สุดที่นับได้คือ 250 KHz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERRUPTS

โปรแกรมคอมพิวเตอร์มีวิธีการจัดการจังหวะของวงจรภายในและภายนอก เพียง 2 วิธีเท่านั้น วิธีแรกคือการใช้คำสั่งซอฟต์แวร์ ซึ่งกระโดดไปที่สถานะของแฟล็กและขาพอร์ท วิธีที่ 2 เป็นการตอบสนองทางสัญญาณฮาร์ดแวร์ เรียกว่า อินเทอร์รัพท์ ซึ่งบังคับเรียกใช้โปรแกรมย่อย เทคนิคทางซอฟต์แวร์นั้นใช้เวลาในการประมวลผลมากกว่า แทนที่จะไปใช้ในงานอื่น แต่การอินเทอร์รัพท์จะใช้เวลาประมวลผลเท่าที่จำเป็น

7	6	5	4	3	2	1	0
EA	—	ET2	ES	ET1	EX1	ETO	EXO

THE INTERRUPT ENABLE (IE) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7	EA	Enable interrupts bit. Cleared to 0 by program to disable all interrupts; set to 1 to permit individual interrupts to be enabled by their enable bits.
6	—	Not implemented.
5	ET2	Reserved for future use.
4	ES	Enable serial port interrupt. Set to 1 by program to enable serial port interrupt; cleared to 0 to disable serial port interrupt.
3	ET1	Enable timer 1 overflow interrupt. Set to 1 by program to enable timer 1 overflow interrupt; cleared to 0 to disable timer 1 overflow interrupt.
2	EX1	Enable external interrupt 1. Set to 1 by program to enable $\overline{INT1}$ interrupt; cleared to 0 to disable $\overline{INT1}$ interrupt.
1	ETO	Enable timer 0 overflow interrupt. Set to 1 by program to enable timer 0 overflow interrupt; cleared to 0 to disable timer 0 overflow interrupt.
0	EXO	Enable external interrupt 0. Set to 1 by program to enable $\overline{INT0}$ interrupt; cleared to 0 to disable $\overline{INT0}$ interrupt.

Bit addressable as IE.0 to IE.7

7	6	5	4	3	2	1	0
—	—	PT2	PS	PT1	PX1	PT0	PX0

THE INTERRUPT PRIORITY (IP) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7	—	Not implemented.
6	—	Not implemented.
5	PT2	Reserved for future use.
4	PS	Priority of serial port interrupt. Set/cleared by program.
3	PT1	Priority of timer 1 overflow interrupt. Set/cleared by program.
2	PX1	Priority of external interrupt 1. Set/cleared by program.
1	PT0	Priority of timer 0 overflow interrupt. Set/cleared by program.
0	PX0	Priority of external interrupt 0. Set/cleared by program.

Note: Priority may be 1 (highest) or 0 (lowest)

Bit addressable as IP.0 to IP.7

รูป ข. 1

การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ส่วนใหญ่ รวมทั้งผลตอบสนองต่อเหตุการณ์ที่เกิดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นต้องเร็วพอที่จะควบคุมสภาพแวดล้อมที่สร้างเหตุการณ์นั้น (มักจะเรียกว่า "real time programming") ซึ่งมักจะใช้การอินเทอร์รัพท์ในการให้ real time programming ประสพผลสำเร็จ

อินเทอร์รัพท์อาจจะสร้างโดยการกระทำภายในชิพ หรือโดยแหล่งภายนอก อินเทอร์รัพท์ใดๆสามารถทำให้ฮาร์ดแวร์ของ 8051 เรียกโปรแกรมย่อย ที่ตำแหน่งที่กำหนดโดยผู้ออกแบบชิพ 8051 มีอินเทอร์รัพท์อยู่ 5 แห่ง ซึ่งมี 3 แหล่งที่สร้างอัตโนมัติโดยการกระทำภายใน คือ timer flag 0, timer flag 1, และ serial port interrupt (RI หรือ TI) อีก 2 แหล่งถูกทริกโดยสัญญาณ ภายนอกซึ่งสร้างโดยวงจรที่ต่อกับขา INT0 และ INT1 (ขาพอร์ต P3.2 และ P3.3)

หน้าที่ของอินเทอร์รัพท์ทั้งหมดจะถูกควบคุมโดยโปรแกรม ซึ่งเป็นโปรแกรมเมอร์สามารถเปลี่ยนแปลงบิตควบคุมนี้ได้ ใน interrupt enable register (IE), the interrupt priority register (IP) และ the timer control register (TCON) โปรแกรมสามารถป้องกันการอินเทอร์รัพท์ ใดๆจากโปรแกรมโดยการเซตหรือเคลียร์บิตในรีจิสเตอร์เหล่านี้ รูป ข.1 แสดงรีจิสเตอร์ IE และ IP

หลังจากเกิดอินเทอร์รัพท์ที่โปรแกรมเมอร์กำหนดตำแหน่งโดยโปรแกรมเมอร์ แล้วโปรแกรมที่ถูกอินเทอร์รัพท์จะทำงานที่คำสั่งที่ทำให้เกิดอินเทอร์รัพท์ โดยเก็บค่า PC เดิมไว้ในสแตคใน RAM แล้วทำการเปลี่ยนค่า PC ไปเป็นตำแหน่งโปรแกรมอินเทอร์รัพท์ใน ROM ตำแหน่ง PC เดิมจะถูกเรียกคืนมาจากสแตคหลังจากทำคำสั่ง RETI ที่ท้ายโปรแกรมย่อยแล้ว

Timer Flag Interrupt

เมื่อไทม์เมอร์ เคนต์เตอร์เกิดโอเวอร์โฟลว์ขึ้น แฟล็กที่ตอบสนอง TF0 และ TF1 จะถูกเซตเป็น "1" แล้วแฟล็กจะถูกเคลียร์เป็น "0" เพื่อโปรแกรมซึ่งที่ถูกเรียกไปที่โปรแกรมย่อยในหน่วยความจำแล้ว

Serial Flag Interrupt

ถ้าไบท์ข้อมูลถูกรับแล้ว บิต RI จะถูกเซตเป็น "1" ในรีจิสเตอร์ SCON หรือเมื่อไบท์ข้อมูลถูกส่งแล้วบิตอินเทอร์รัพท์ TI ถูกเซตใน SCON แล้วจะทำให้เกิดสัญญาณอินเทอร์รัพท์ 1 ครั้งไปที่กระบวนการอินเทอร์รัพท์การส่งแบบอนุกรม บิตเหล่านี้จะไม่ถูกเคลียร์เมื่อโปรแกรมอินเทอร์รัพท์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีเซ็ต ทำเสร็จแล้วต้องเขียนโปรแกรมรีเซ็ต *RI* หรือ *TI* ให้เป็น "0" เพื่อจะได้สื่อสารกับข้อมูลต่อไปได้

External Interrupt

ขา *INT0* และ *INT1* เป็นอินพุตที่มาจากวงจรมานอก สัญญาณอินเทอร์รัพท์ จะทำการรีเซ็ตแฟล็ก *IE0* หรือ *IE1* ใน *TCON* เป็น 1 ได้ 2 วิธี แฟล็ก *IEX* อาจจะถูกรีเซ็ตเมื่อสัญญาณขา *INTX* มีลอจิกต่ำ หรือ เกิดการเปลี่ยนจาก *High to Low* บนขา *INTX* เมื่อรีเซ็ตบิต *IT0* และ *IT1* เป็น "1" จะเป็นการตรวจจับสัญญาณอินเทอร์รัพท์ที่ขา *INTX* ที่มีการเปลี่ยนแปลงจาก *High To Low*

แฟล็ก *IEX* จะถูกรีเซ็ตโดยโปรแกรมหลังจากได้ย้ายไปที่โปรแกรมย่อย หรือเป็นการอินเทอร์รัพท์แบบ *Transition* สำหรับการรีเซ็ตอินเทอร์รัพท์จากภายนอก *Level* นั้นชนกับผู้ออกแบบ และโปรแกรมเมอร์จึงจะได้รับการรีเซ็ต หลังจากได้รับการบริการแล้วหรือไม่ วงจรมานอกต้องย้าย *low level* นั้นออกไปก่อนที่ *RETI* จะถูกกระทำ ถ้าย้ายออกไปไม่ทันก็จะเกิดการอินเทอร์รัพท์ขึ้นอีก

Reset

การรีเซ็ตก็เป็นอินเทอร์รัพท์แบบหนึ่งเพราะโปรแกรมไม่ได้ป้องกันแรงดันบนขา *RST* อินเทอร์รัพท์แบบนี้มักจะเรียกว่า "non maskable" เพราะไม่มีบิตใดในรีจิสเตอร์ที่สามารถหยุดหรือห้ามการรีเซ็ตนี้ ซึ่งต่างกับอินเทอร์รัพท์ธรรมดา ค่า *PC* จะไม่ถูกเก็บไว้สำหรับการทำโปรแกรมต่อไป การรีเซ็ตเป็นการกระโดดข้ามไปที่ตำแหน่ง *0000h* และ เริ่มทำงานจากตรงนั้น

เมื่อไรก็ตามที่มี *High Level* ถูกจ่ายให้กับขา *RST* ทำให้ *8051* เข้าสู่เงื่อนไขการรีเซ็ต หลังจากขา *RST* มีลอจิกต่ำแล้ว รีจิสเตอร์ภายในจะมีค่าต่างๆ ดังตารางต่อไปนี้

REGISTER	VALUE (HEX)
PC	0000
DPTR	0000
A	00
B	00
SP	07
PSW	00
PO-3	FF
IP	XXX00000b
IE	OXX00000b
TCON	00
TMOD	00
TH0	00
TLO	00
TH1	00
TL1	00
SCON	00
SBUF	XX
PCON	OXXXXXXXb

RAM ภายใน จะไม่เปลี่ยนแปลงเนื่องจากการรีเซ็ต สถานะของ RAM ภายในเมื่อเริ่มจ่ายพลังงานให้ 8051 จะถูกสุมค่าของรีจิสเตอร์ ใน Bank 0 จะกลายเป็น "0" หมด เช่นเดียวกับใน PSW

Interrupt Control

ในบางเวลาโปรแกรมอาจจะไม่ยอมให้มีการอินเทอร์รัพท์เกิดขึ้น ซึ่งบิตที่ควบคุมการเกิดอินเทอร์รัพท์ก็จะอยู่ในรีจิสเตอร์ IE ซึ่งสามารถโปรแกรมให้มีการตอบรับหรือไม่ตอบรับต่อการอินเทอร์รัพท์ทั้งหมดหรือบางส่วนก็ได้ แต่บ่อยครั้งที่มักจะมีการเปรียบเทียบความสำคัญระหว่างอินเทอร์รัพท์ที่เกิดขึ้นพร้อมกัน ซึ่งกำหนดไว้ในรีจิสเตอร์ IP โดยอินเทอร์รัพท์ที่มีความสำคัญกว่าจะได้รับการบริการก่อน

Interrupt Enable Disable

ถ้าเซตบิตในรีจิสเตอร์ *EA* เป็น "1" ก็จะสามารถตอบสนองอินเทอร์รัพท์ได้ แต่ถ้าเซตเป็น "0" ก็จะไม่สามารถตอบสนองต่ออินเทอร์รัพท์ได้ บิต *EA* จะเป็นตัวสำคัญที่จะยอมให้มีการอินเทอร์รัพท์ก็ได้หรือไม่ก็ได้

Interrupt Priority

รีจิสเตอร์ *IP* จะเป็นตัวบอกว่าอินเทอร์รัพท์ใดมีความสำคัญมากหรือน้อย ถ้าเซตเป็น "1" ก็จะมีค่าสำคัญมาก ถ้าเซตเป็น "0" ก็จะมีค่าสำคัญน้อย อินเทอร์รัพท์ที่มีความสำคัญมากสามารถอินเทอร์รัพท์ตัวอื่นที่มีความสำคัญน้อยกว่าได้ แล้วตัวที่มีความสำคัญต่ำกว่าก็จะขออินเทอร์รัพท์หลังจากตัวที่มีความสำคัญสูงกว่าทำเสร็จแล้ว

ถ้าอินเทอร์รัพท์ที่เกิดขึ้น 2 แหล่งมีความสำคัญเท่ากัน เกิดขึ้นพร้อมกันแล้วเราจะพิจารณาตามลำดับดังนี้

1. IEO
2. TFO
3. IE1
4. TF1
5. Serial = RI or TI

เราอาจจะกำหนดให้ *Serial Interrupt* มีความสำคัญสูงสุดโดยเซตบิต *PS* ใน *IP* เป็น "1" และบิตอื่นๆ เป็น "0"

Interrupt Destinations

แต่ละอินเทอร์รัพท์ทำให้โปรแกรมกระทำทางฮาร์ดแวร์ไปที่ตำแหน่งที่กำหนดไว้ ในหน่วยความจำซึ่งขึ้นกับโปรแกรมเมอร์ ว่าจะให้อินเทอร์รัพท์นั้นถูกบริการที่ตำแหน่งโปรแกรมย่อยใด

เมื่อเกิดการอินเทอร์รัพท์จะมีการเก็บค่า *PC* ของโปรแกรมที่ขณะเกิดอินเทอร์รัพท์ลงบนสแตคใน RAM ภายใน แล้วก็ย้ายไปทำงานที่ตำแหน่งที่เหมาะสมซึ่งกำหนดได้ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERRUPT	ADDRESS (HEX)
IE0	0003
TFO	000B
IE1	0013
TF1	001B
SERIAL	0023

คำสั่ง *RETI* ที่ตอนท้ายของโปรแกรมย่อยจะดึงค่า PC เดิมกลับมา และรีเซ็ตลอจิกอินเทอร์รัพท์เพื่อที่จะได้ให้บริการแก่อินเทอร์รัพท์ตัวอื่นๆได้ การที่มีอินเทอร์รัพท์เกิดขึ้นแต่ไม่ได้รับการตอบรับ เนื่องจากเงื่อนไขถูกปิดกั้น (บิต *IE* ไม่ถูกเซต หรืออินเทอร์รัพท์ที่มีความสำคัญกว่ากระทำเสร็จแล้ว) ต้องรอจนกว่าจะได้รับการบริการ มิฉะนั้นแล้วข้อมูลก็จะสูญหายไป ซึ่งตรงกับหลักการอินเทอร์รัพท์แบบ *INTX*

Software Generated Interrupts

เมื่ออินเทอร์รัพท์แฟลกใดๆ ถูกเซตเป็น "1" ไม่ว่าจะด้วยวิธีการใด แล้วอินเทอร์รัพท์ก็จะถูกทำให้อยู่ในสถานะที่ไม่มีกัป้องกัน ซึ่งก็หมายความว่า โปรแกรมจะสามารถอินเทอร์รัพท์ตัวเองได้ทุกชนิดโดยการเซตอินเทอร์รัพท์เป็น "1" ด้วยคำสั่งในโปรแกรม

54193/74193 Synchronous 4-Bit Binary Up/Down Counter (Dual clock with clear)

	Schottky TTL				High-Speed TTL				Low-Power-Schottky TTL				Standard TTL				Low-Power TTL						
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package					
T. I.									SN54LS193	J	Q		WD	SN54193	J	Q		WD	SN54L193	J	Q		WD
FAIRCHILD									SN74LS193	J	Q	N	Q	SN74193	J	Q	N	Q	SN74L193	J	Q	N	Q
MOTOROLA									FM54LS193/FM74LS193	WD	Q		Q	FM54193/FM74193	WD	Q		Q	FC74LS193/FC74LS193	WD	Q		Q
N. S. C.									SN74LS193	P	Q			MC74193	P	Q			DM54L193	J	Q	N	Q
PHILIPS									DM74LS193					DM74193					DM74L193	J	Q	N	Q
SIGNETICS									N74LS193					N74193									
SIEMENS														S54193	F	Q	B	Q	WD				
FUJITSU														N74193	F	Q	B	Q					
HITACHI									74LS193	M	Q			FLJ251									
MITSUBISHI									HD74LS193	P	Q			HD2542	D	P	Q						
NEC									M74LS193	P	Q			M53393	P	Q							
TOSHIBA									74LS193	C	Q			μPB2193	D	Q							
AMD									Am54LS193					TD74193	P	Q							
									Am74LS193														

Electrical Characteristics SN54LS193/SN74LS193

absolute maximum ratings over operating free-air temperature range

Supply voltage, V _{CC}	7V	Operating free-air temperature range	SN54LS	-55°C to 125°C
Input voltage	5.5V	SN74LS	0°C to 70°C	
		Storage temperature range		-65°C to 150°C

recommended operating conditions

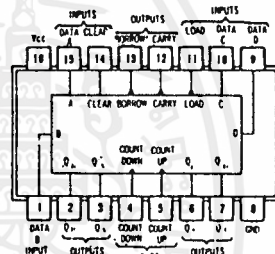
	SN54LS193			SN74LS193			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V _{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I _{OH}			-400			-400	μA
Low-level output current, I _{OL}			4			8	mA
Count frequency, f _{count}	0		25	0		25	MHz
Width of any input pulse, t _w	20		20				ns
Data setup time, t _{setup}	20		20				ns
Data hold time, t _{hold}	0		0				ns
Operating free-air temperature, T _A	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER*	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT	
V _{IH}	High-level input voltage		2		V	
V _{IL}	Low-level input voltage			0.8	V	
V _I	Input clamp voltage	V _{CC} = MIN, I _I = -18 mA		-1.5	V	
V _{OH}	High-level output voltage	V _{CC} = MIN, V _{IH} = 2V, V _{IL} = 0.8V, I _{OH} = -400 μA	2.7	3.4	V	
V _{OL}	Low-level output voltage	V _{CC} = MIN, V _{IH} = 2V, V _{IL} = 0.8V, I _{OL} = 8 mA	0.35	0.5	V	
I _I	Input current at maximum input voltage	V _{CC} = MAX, V _I = 7V		0.1	mA	
I _{IH}	High-level input current	V _{CC} = MAX, V _I = 2.4V		20	μA	
I _{IL}	Low-level input current	V _{CC} = MAX, V _I = 0.4V		0.4	mA	
I _{OS}	Short-circuit output current*	V _{CC} = MAX	SN54LS -20 SN74LS -20	100 100	mA	
I _{CC}	Supply current	V _{CC} = MAX, See Note 1	SN54LS 19 SN74LS 19	34 34	mA	
f _{max}	maximum clock frequency		25	32	MHz	
t _{PLH}	from Count-up to output	V _{CC} = 5V, T _A = 25°C, C _L = 15pF, R _L = 2k		17	26	ns
t _{PHL}	Carry			18	24	ns
t _{PLH}	from Count-down to output			16	24	ns
t _{PHL}	Borrow			15	24	ns
t _{PLH}	from Either Count to output 0			27	38	ns
t _{PHL}				30	47	ns
t _{PLH}	from Load to output 0			24	40	ns
t _{PHL}				25	40	ns
t _{PHL}	from Clear to output 0			23	35	ns

NOTES 1 I_{CC} is measured with all outputs open clear and load inputs grounded, and all other inputs at 4.5V
 A Clear overrides load, data, and count inputs
 B When counting up, count-down input must be high; when counting down, count-up input must be high

Pin Assignment (Top View)

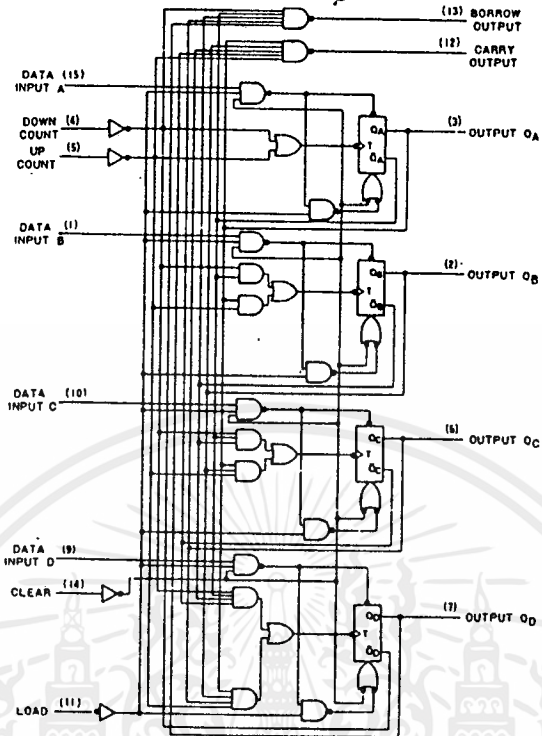


†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
 ‡All typical values are at V_{CC} = 5V, T_A = 25°C.
 *Not more than one output should be shorted at a time.
 †t_{PLH} = propagation delay time, low-to-high-level output
 †t_{PHL} = propagation delay time, high-to-low-level output

CONTINUED ON NEXT PAGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะวิธีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

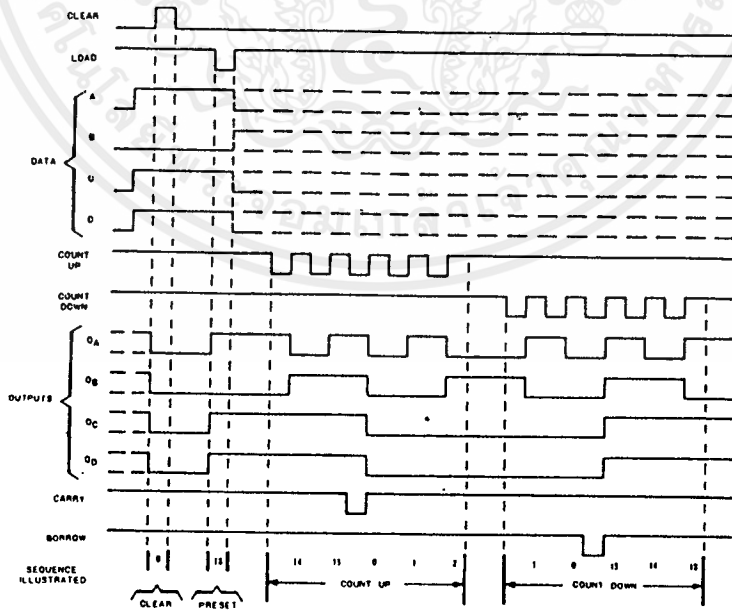
Functional Block Diagram



Dynamic input activated by a transition from a high level to a low level.
 193, L193, LS193 SYNCHRONOUS 4-BIT UP/DOWN COUNTER

typical clear, load, and count sequences

- Illustrated below is the following sequence (See Notes A and B)
1. Clear outputs to zero.
 2. Load (preset) is binary thirteen.
 3. Count up to fourteen, fifteen, carry, zero, one, and two.
 4. Count down to one, zero, borrow, fifteen, fourteen, and thirteen.



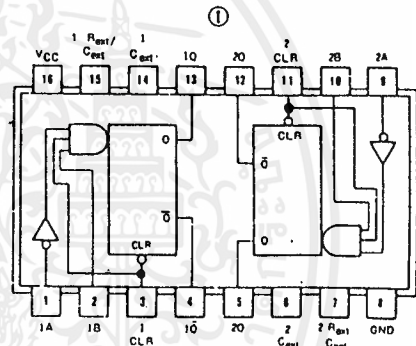
54123/74123 Dual Retriggerable Monostable Multivibrator with Clear

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL						
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package					
	C	P	M	CF	C	P	M	CF	C	P	M	CF	C	P	M	CF	C	P	M	CF			
TTL									SN54LS123	J	D		WD	SN54123	J	D		WD	SN54L123	J	D		
FAIRCHILD									SN74LS123	J	D	ND		SN74123	J	D	ND		SN74L123	J	D	ND	
MOTOROLA													MC74123					DM54L123A					
N.S.C.									DM54LS123					DM74123					DM74L123A				
PHILIPS									N74LS123					N74123									
SIGNETICS													SS4123	F	D	B	Q	WD					
SIEMENS													N74123	F	D	B	Q						
FUJITSU									74LS123	M	D			MB440									
HITACHI									HD74LS123	F	D			HD74123/HD2516									
MITSUBISHI									M74LS123	F	D			M53323									
NEC													μPB2123										
TOSHIBA																							

Electrical Characteristics SN54LS123/SN74LS123

absolute maximum ratings operating free-air temperature range							
Supply voltage, V _{CC}	7V	Operating free air temperature range	SN54LS	-55°C to 125°C			
Input voltage	7V		SN74LS	0°C to 70°C			
		Storage temperature range		-65°C to 150°C			
recommended operating conditions							
			SN54LS123	SN74LS123	UNIT		
Supply voltage, V _{CC}	MIN 4.5	NOM 5	MAX 5.5	MIN 4.75	NOM 5	MAX 5.25	V
High-level output current, I _{OH}			400			-400	μA
Low-level output current, I _{OL}			8			8	mA
Pulse width, t _p		40		40			ns
External timing resistance, R _{ext}		5	180	5	250		kΩ
External timing capacitance, C _{ext}			No restriction		No restriction		
Timing capacitance at Reset, C _{ext} terminal			50		50		pf
Operating free air temperature, T _a		-55	125	0	70		°C
electrical characteristics over recommended operating free-air temperature range							
PARAMETER*	TEST CONDITIONS†			MIN	TYP‡	MAX	UNIT
V _{IH}	High-level input voltage			2			V
V _{IL}	Low-level input voltage				0.8		V
V _I	input clamp voltage	V _{CC} = MIN, I _I = -18mA			-1.5		V
V _{OH}	High-level output voltage	V _{CC} = MIN, I _{OH} = MAX, See Note 1		2.7	3.5		V
V _{OL}	Low-level output voltage	V _{CC} = MIN, I _{OL} = 8mA, See Note 1		0.35	0.5		V
I _I	Input current at maximum input voltage	V _{CC} = MAX, V _I = 7V		0	1		mA
I _{IH}	High-level input current	V _{CC} = MAX, V _I = 2.7V			20		μA
I _{IL}	Low-level input current	V _{CC} = MAX, V _I = 0.4V			0.4		mA
I _{OS}	Short-circuit output current*	V _{CC} = MAX, See Note 1		20	100		mA
I _{CC}	Supply current (quiescent or triggered)	V _{CC} = MAX, See Notes 2 and 3		12	20		mA
t _{PLH}	from A to output Q	V _{CC} = 5V, I _A = 25°C, C _{ext} = 0, R _{ext} = 5kΩ		22	33		ns
t _{PHL}	from B to output Q			23	34		ns
t _{PHL}	from A to output Q			32	45		ns
t _{PHL}	from B to output Q			34	56		ns
t _{PLH}	from Clear to output Q	C _L = 15pF		20	27		ns
t _{PLH}	from Clear to output Q	R _L = 2kΩ		28	45		ns
t _{pQ} (min)	from A or B to output Q			116	200		ns
t _{wO}	from A or B to output Q	C _{ext} 1000pF, R _{ext} 10kΩ		4	4.5	5	μs

Pin Assignment (Top View)



FUNCTION TABLE

123 LS123, L123 (See Note 4)

CLEAR	INPUTS		OUTPUTS	
	A	B	Q	Q̄
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	L	L	H
H	X	H	L	H
X	L	H	L	H

- NOTES: 1: Ground C_{ext} to measure V_{OH} at Q, V_{OL} at Q or I_{OS} at Q
 C_{ext} is open to measure V_{OH} at Q, V_{OL} at Q or I_{OS} at Q
 2: Quiescent I_{CC} is measured (after clearing) with 2.4V applied to all clear and A inputs, B inputs grounded, all outputs open, C_{ext} = 0.02 μF, and R_{ext} = 25kΩ.
 3: I_{CC} is measured in the triggered state with 2.4V applied to all clear and B in inputs, A inputs grounded, all outputs open, C_{ext} = 0.02 μF, and R_{ext} = 25kΩ.
 4: A = high level (steady state), L = low level (steady state), ↑ = transition from low to high level, ↓ = transition from high to low level, ↑ = one high level pulse, ↓ = one low-level pulse, X = irrelevant (any input, including transitions)
 B An external timing capacitor may be connected between Rext and Cext (positive)
 C For accurate repeatable pulse widths, connect an external resistor between Rext/Cext and VCC with R₁₁ open-circuited
 D To obtain variable pulse widths, connect external resistance between R_{int} or Rext/Cext and VCC.

† For conditions shown as MIN or MAX, use the value specified under recommended operating conditions
 ‡ All typical values are at V_{CC} = 5V, T_a = 25°C
 * Not more than one output should be shorted at a time
 † t_{PLH} = propagation delay time, low-to-high-level output
 ‡ t_{PHL} = propagation delay time, high-to-low-level output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54244/74244 Octal Buffers/Line Drivers/Line Receivers

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL				
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			
		C	P	MCF		C	P	MCF		C	P	MCF		C	P	MCF		C	P	MCF	
T. I.									SN54LS244	J	D										
									SN74LS244	J	D										
FAIRCHILD																					
MOTOROLA																					
N. S. C.																					
PHILIPS																					
SIGNETICS																					
SIEMENS																					
FUJITSU																					
HITACHI																					
MITSUBISHI																					
NEC																					
TOSHIBA																					

Electrical Characteristics SN54LS244/SN74LS244

absolute maximum ratings over operating free-air temperature range			
Supply voltage, VCC	7V	Operating free-air temperature range	SN54LS 55°C to 125°C
Input voltage	5.5V	temperature range	SN74LS 0°C to 70°C
Intermittent voltage	5.5V	Storage temperature range	65°C to 150°C
recommended operating conditions			
	L554LS244		SN74LS244
	MIN	NOM	MAX
Supply voltage, VCC	4.5	5	5.5
High-level output current, IOH			12
Low-level output current, IOL			12
Operating free-air temperature, TA	55	125	0

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

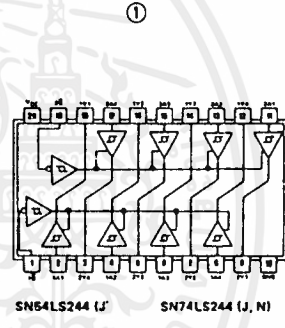
PARAMETER	TEST CONDITIONS †	SN74LS		UNIT
		MIN	TYP ‡	
V _{IH} High-level input voltage		2		V
V _{IL} Low-level input voltage			0.8	V
V _{IK} Input clamp voltage	VCC = MIN, I _I = -18mA		-1.5	V
Hysteresis (V _{T+} - V _{T-})	VCC = MIN	0.2	0.4	V
V _{OH} High-level output voltage	VCC = MIN, V _{IH} = 2V, V _{IL} = V _{ILmax} , I _{OH} = -3mA	2.4	3.4	V
V _{OL} Low-level output voltage	VCC = MIN, V _{IH} = 2V, V _{IL} = V _{ILmax} , I _{OL} = 12mA		0.4	V
I _{OZH} Off-state output current, high-level voltage applied	VCC = MAX, V _O = 2.7V		20	µA
I _{OZL} Off-state output current, low-level voltage applied	VCC = MAX, V _{IH} = 2V, V _{IL} = V _{ILmax} , V _O = 0.4V		-20	µA
I _I Input current at maximum input voltage	VCC = MAX, V _I = 7V		0.1	mA
I _{IH} High-level input current, any input	VCC = MAX, V _I = 2.7V		20	µA
I _{IL} Low-level input current	VCC = MAX, V _{IL} = 0.4V		-0.2	mA
I _{OS} Short-circuit output current †	VCC = MAX	-40	-225	mA
I _{CC} Supply current	Outputs high	VCC = MAX	All 13	23
	Outputs low		LS244	27
	All outputs disabled	Outputs open	LS244	32

switching characteristics, VCC 5V, TA 25°C

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t _{PLH} Propagation delay time, low-to-high-level output			9	14	ns
t _{PHL} Propagation delay time, high-to-low-level output	C _L = 45pF, R _L = 667Ω, See Note 2		12	18	ns
t _{PZL} Output enable time to low level			20	30	ns
t _{PZH} Output enable time to high level			15	23	ns
t _{PLZ} Output disable time from low level	C _L = 50F, R _L = 667Ω, See Note 2		15	25	ns
t _{PHZ} Output disable time from high level			10	18	ns

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions
 ‡ All typical values are at VCC 5V, TA 25°C
 † Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second
 NOTE 2 Load circuit and voltage wave forms are shown on page 3-11

Pin Assignment (Top View)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้