

ระบบรักษาความปลอดภัยผ่านทางกล้องวิดีโอ

VIDEO SECURITY SYSTEM



โดย

นาย จรัส ปัดดวงแก้ว

นาย ชัชวาล สายสมุทร

เลขหมู่.....
เลขทะเบียน..... 42649
วัน, เดือน, ปี- 5 ส.ย. 2545

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

ปฏิญานี้เป็นส่วนหนึ่งของหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยผ่านทางกล้องวิดีโอ
VIDEO SECURITY SYSTEM



โดย

นาย จรัส ปัดดวงแก้ว 40010111

นาย ชัชวาล สายสมุทร 40010171

อาจารย์ที่ปรึกษา

ดร.กิตติพล ชิตสกุล

ปริญญานิพนธ์นี้สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2543

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรักษาความปลอดภัยผ่านทางกล้องวิดีโอ

ผู้จัดทำ

นาย จรุต ปัดดวงแก้ว 40010111

นาย ชัชวาล สายสมุทร 40010171



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยผ่านทางกล้องวิดีโอ

นาย จรัส ปัดดวงแก้ว

นาย ชัชวาล สายสมุทร

ดร. กิตติพล ชิตสกุล อาจารย์ที่ปรึกษา

ปีการศึกษา 2543

บทคัดย่อ

โครงการนี้เป็นการออกแบบสร้างระบบรักษาความปลอดภัยผ่านทางกล้องวิดีโอ โดยใช้ไมโครคอมพิวเตอร์ในการควบคุมทิศทางและตำแหน่งการแพนกล้องในช่วง ± 90 องศาโดยมีโหมดการใช้งานสองโหมดคือกำหนดเองโดยผู้ควบคุมผ่านเมาส์หรือโหมดอัตโนมัติที่การกวาดของกล้องในแนวราบกลับไปมาอย่างต่อเนื่อง ภาพจากกล้องจะแสดงบนเมนูบนไมโครคอมพิวเตอร์อย่างต่อเนื่องทั้งสองโหมด ไมโครโปรเซสเซอร์ตระกูล MCS-51 ถูกใช้ในการควบคุมการหมุนของสเตปปีงมอเตอร์ การรับส่งข้อมูล แสดงตำแหน่งของกล้อง และอินเตอร์เฟซกับระบบโทรศัพท์ ผู้ใช้สามารถส่งสัญญาณเป็นเสียงเตือนที่บันทึกไว้ล่วงหน้าผ่านไปยังเครื่องรับ โดยการกดปุ่มสวิทช์เพียงปุ่มเดียว นอกจากนี้ผู้ใช้สามารถบันทึกภาพนิ่งในเฟรมที่เลือกได้อย่างต่อเนื่อง

VIDEO SECURITY SYSTEM

Mr. CHARUN PIDDOUNGKEAW

Mr. CHATCHAWAN SAISAMUT

Dr. KITTIPHON CHITSAKUL Advisor

Educational Year 2000

Abstract

This project is the design and implementation of a video security system. A microcomputer is employed to define the position and direction of panning of the video camera in ± 90 degree. Two modes of operation, manual and automatic mode is provided. In the manual mode, the operator can control the camera by clicking the mouse on the menu. The camera pans continuously when automatic mode has been selected. The image from the camera is continuously displayed on the monitor in both two modes of operation. The electronic system for controlling the camera, displaying the position of camera and interfacing to telephonic system is based on a microcontroller family MCS-51. The operator can send an record alarm sentence to a programmed phone number by one touch switch. Moreover, the image of selected frame can record on a file on the microcomputer.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีเป็นเพราะได้รับคำแนะนำและสั่งสอนมาจาก อาจารย์ ดร.กิตติพล ชิตสกุล อีกทั้งได้รับคำแนะนำ การแก้ปัญหา การระดมพลังทางความคิดจาก เพื่อน ๆ ที่ร่วมทำงานมาด้วยกัน ผู้จัดทำจึงขอขอบคุณ ทุก ๆ ท่านที่มีส่วนเกี่ยวข้องในความสำเร็จครั้งนี้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 ลักษณะของโครงการ	1
1.3 เนื้อหาของปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 สเต็ปเปอร์มอเตอร์	4
2.1.1 ชนิดของสเต็ปเปอร์มอเตอร์	4
2.1.2 การทำงานของสเต็ปเปอร์มอเตอร์	4
2.2 ไมโครคอนโทรลเลอร์ MCS-51	12
2.2.1 โครงสร้างของ 8051	12
2.2.2 สถาปัตยกรรมของ 8051	14
2.2.3 การทำงานของ 8051	22
2.3 มาตรฐานการเชื่อมต่อแบบ RS-232-C	26
2.4 การสื่อสารแบบ Synchronous และ Asynchronous	30
2.5 พอร์ตสื่อสารข้อมูลอนุกรม UART	31
2.5.1 พอร์ตสื่อสารข้อมูลอนุกรม UART ภายในชิพ MCS-51	31
2.5.2 ไทม์เมอร์ในการกำหนดอัตราเร็วในการรับส่งข้อมูล	36
2.6 สัญญาณโทรศัพท์	42
2.7 ตัวแสดงผลโคไดคอปเล้งแสง 7 ส่วน (Seven – Segment LED Displays)	44
2.8 ตัวถอดรหัส (Decoders)	47
2.9 ตัวถอดรหัสบีซีดีขับออกตัวแสดงผลตัวเลข 7 ส่วน (BCD-to-Seven Segment Decoder/Driver)	50
บทที่ 3 หลักการออกแบบ	55
3.1 การออกแบบในส่วนฮาร์ดแวร์ (Hardware)	56
3.1.1 วงจรไมโครคอนโทรลเลอร์	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.1.2 วงจรแปลงระดับแรงดันระหว่างคอมพิวเตอรื(+15v) กับไมโครคอนโทรลเลอร์(+5v)	58
3.1.3 สเต็ปมอเตอร์ และ วงจรขับสเต็ปมอเตอร์ (DRIVER)	59
3.1.4 วงจรแสดงผลทาง 7 –SEGMENT	61
3.1.5 วงจรส่วนเชื่อมต่อโทรศัพท์	62
3.1.6 ส่วนหม้อแปลง 600-600 โอห์ม	67
3.2 การออกแบบในส่วนซอฟต์แวร์ (Software)	68
3.2.1 การสื่อสารผ่านทาง Serial Port ด้วย Visual Basic	68
3.2.2 โพลีซาร์ทในส่วนของโปรแกรมภาษา Visual Basic 6.0	70
3.2.3 โพลีซาร์ทในส่วนของโปรแกรมภาษา Assembly	72
บทที่ 4 การทดสอบและผลการทดสอบ	78
4.1 การทดสอบการส่งค่าผ่านพอร์ตอนุกรม	78
4.2 ผลการทดสอบวงจรตรวจจับสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง	81
บทที่ 5 สรุปและวิจารณ์	84
บรรณานุกรม	
ภาคผนวก ก โปรแกรม	
ภาคผนวก ข คู่มือการใช้	
ภาคผนวก ค Data Cheet	

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 บล็อกไดอะแกรมของโครงการส่วนควบคุมกล้อง	2
รูปที่ 1.2 บล็อกไดอะแกรมของโครงการส่วนควบคุมโทรศัพท์เตือนภัย	2
รูปที่ 2.1 ลักษณะการพันขลวดบนสเตเตอร์	4
รูปที่ 2.2 ทิศทางการหมุนโรเตอร์ของสเต็ปเปอร์มอเตอร์ 4 เฟส	5
รูปที่ 2.3 ไบโพลาร์สเต็ปเปอร์มอเตอร์แบบ 2 เฟส สนามแม่เหล็กจะเปลี่ยนเมื่อ กลับทิศทางการไหล ของกระแส	6
รูปที่ 2.4 ยูนิโพลาร์สเต็ปเปอร์มอเตอร์การเปลี่ยนขั้วแม่เหล็กเกิดจาก กระแสไหลต่างขดกันกระแสจะไม่ไหลพร้อมกัน 2 ขดในสเตเตอร์เดียวกัน	6
รูปที่ 2.5 วงจรทดลองการทำงานของสเต็ปเปอร์	7
รูปที่ 2.6 บล็อกไดอะแกรมควบคุมด้วยสเต็ปเปอร์มอเตอร์	10
รูปที่ 2.7 การขับสเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์และไบโพลาร์	10
รูปที่ 2.8 การใช้กระแสขับเพื่อเพิ่มโมเมนต์เมื่อมีอัตราเร่งสูง	11
รูปที่ 2.9 สถาปัตยกรรมภายในของ 8051	14
รูปที่ 2.10 ไดอะแกรมขาของ 8051 แบบ DIP	15
รูปที่ 2.11 โครงสร้างของพอร์ท 0	16
รูปที่ 2.12 โครงสร้างของพอร์ท 1	17
รูปที่ 2.13 โครงสร้างของพอร์ท 2	18
รูปที่ 2.14 โครงสร้างของพอร์ท 3	19
รูปที่ 2.15 วงจรออสซิลเลเตอร์ภายใน 8051	22
รูปที่ 2.16 8051 ที่ทำงานโดยสัญญาณที่มาจากภายนอก	22
รูปที่ 2.17 ลำดับสถานะการทำงานใน MCS-51	24
รูปที่ 2.18 แสดงชุดข้อมูล UART ในโหมดที่ 1	32
รูปที่ 2.19 แสดงชุดข้อมูล UART ในโหมดที่ 2	33
รูปที่ 2.20 แสดงชุดข้อมูล UART ในโหมดที่ 2	33
รูปที่ 2.21 แสดงลักษณะของ LED ตัวเลข 7 ส่วน	44
รูปที่ 2.22 แสดงลักษณะภายนอกของตัวแสดงผลไดโอดเปล่งแสง 7 ส่วนแบบต่างๆ	45
รูปที่ 2.23 แสดงโครงสร้างภายในและการควบคุมตัวแสดงผล ไดโอดเปล่งแสง 7 ส่วนชนิดแอนโนด ร่วมและชนิดแคโทดร่วม	45
รูปที่ 2.24 แสดงการต่อตัวถอดรหัสกับตัวแสดงผลไดโอดเปล่งแสง 7 ส่วน	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 2.25 แสดงการวางขาของตัวถอดรหัสเบอร์ 7442 ,7443 และ 7444 และโครงสร้างภายในพร้อม ตารางความจริงแสดงการทำงานของตัวถอดรหัส	48
รูปที่ 2.26 แสดงลักษณะของตัวถอดรหัสเบอร์ 7447 และตารางแสดงการทำงานของตัวถอดรหัส และเข้าพุทที่ตัวถอดรหัสขั้วออกที่ตัวแสดง ผลไดโอดเปล่งแสง 7 ส่วน	51
รูปที่ 2.27 การต่อวงจรตัวถอดรหัสเบอร์ 7447 เพื่อขั้วตัว แสดงผลไดโอดเปล่งแสง 7 ส่วนชนิดแอนโนคร่วม	52
รูปที่ 2.28 แสดงโครงสร้างภายในของตัวถอดรหัสเบอร์ 7447	52
รูปที่ 2.29 แสดงโครงสร้างภายใน ลักษณะการจัดวางขา และตารางการทำงานของตัวถอดรหัสเบอร์ 7448	53
รูปที่ 3.1 บล็อกไดอะแกรมของโครงงาน (a) ส่วนควบคุมสแตมป์มอเตอร์ (b) ส่วนโทรศัพท์เตือนภัย	55
รูปที่ 3.2 แสดงวงจรไมโครคอนโทรลเลอร์ควบคุมสแตมป์มอเตอร์	56
รูปที่ 3.3 แสดงรูปวงจรไมโครคอนโทรลเลอร์ควบคุมสแตมป์มอเตอร์	57
รูปที่ 3.4 แสดงวงจรไมโครคอนโทรลเลอร์ควบคุมโทรศัพท์เตือนภัย	57
รูปที่ 3.5 แสดงวงจรไมโครคอนโทรลเลอร์ควบคุมโทรศัพท์เตือนภัย	58
รูปที่ 3.6 แสดงรูปวงจรแปลงระดับแรงดัน	58
รูปที่ 3.7 แสดงรูปวงจรแปลงระดับแรงดัน	59
รูปที่ 3.8 รูปสแตมป์มอเตอร์	59
รูปที่ 3.9 วงจรขั้วมอเตอร์	60
รูปที่ 3.10 วงจรขั้วมอเตอร์	60
รูปที่ 3.11 แสดงวงจรแสดงผลทาง 7-Segment	61
รูปที่ 3.12 แสดงรูปวงจรแสดงผลทาง 7-Segment	61
รูปที่ 3.13 รูปแสดงส่วนตรวจจับสัญญาณโทรศัพท์	62
รูปที่ 3.14 สัญญาณเรียกกลับ(บน)และสัญญาณสายไม่ว่าง(ล่าง)	63
รูปที่ 3.15 วงจรการตรวจจับสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง	64

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 3.16 รูปวงจรรองแรงดันและวงจรเปรียบเทียบแรงดัน	65
รูปที่ 3.17 วงจรรองแรงดันและวงจรเปรียบเทียบแรงดัน	65
รูปที่ 3.18 แสดงสัญญาณที่ได้ในการตรวจจับสัญญาณเรียกกลับและสายไม่ว่าง	66
รูปที่ 3.19 แสดงหม้อแปลง 600-600 โอห์ม	67
รูปที่ 3.20 Control MSCom (รูปโทรศัพท์)	68
รูปที่ 3.21 Control eZVidCap	69
รูปที่ 3.22 φόρμของ COMPUTER CONTROL VIDEO POSITION และ φόρμของ CAMERA SETTING	70
รูปที่ 3.23 โฟล์วชาร์ตของโปรแกรมวิชาลเบสิก	71
รูปที่ 3.24 โฟล์วชาร์ตการสั่งงานโทรศัพท์เตือนภัย	72
รูปที่ 3.25 โฟล์วชาร์ตส่วนควบคุมการหมุนกล็อง	73
รูปที่ 3.26 แสดงรูปวงจรส่วนควบคุมการหมุนกล็อง	74
รูปที่ 3.27 แสดงรูปวงจรส่วนการโทรศัพท์เตือนภัย	75
รูปที่ 3.28 แสดงรูปวงจรแปลงระดับแรงดัน	76
รูปที่ 3.29 แสดงรูปวงจรจับมอดเตอร์	77
รูปที่ 4.1 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Exit	78
รูปที่ 4.2 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Left	79
รูปที่ 4.3 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Right	79
รูปที่ 4.4 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Night	80
รูปที่ 4.5 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Stop	80
รูปที่ 4.6 วงจรตรวจจับสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง	81
รูปที่ 4.7 แสดงสัญญาณที่ขา 6 เอ้าท์พุทของ 74121 ของสัญญาณเรียกกลับ	82
รูปที่ 4.8 แสดงสัญญาณสายไม่ว่างที่วัดได้จากอินพุทขา 3 ของ LM567	82
รูปที่ 4.9 แสดงการเปรียบเทียบสัญญาณสายไม่ว่างอินพุทกับเอ้าท์พุทของ LM567	83
รูปที่ 5.1 แสดงรูปโครงการงานขณะใช้งาน	85
รูปที่ 5.2 แสดงรูปวงจрд้านข้าง	86
รูปที่ 5.3 แสดงรูปโครงการงานด้านบน	86

บทที่ 1

บทนำ

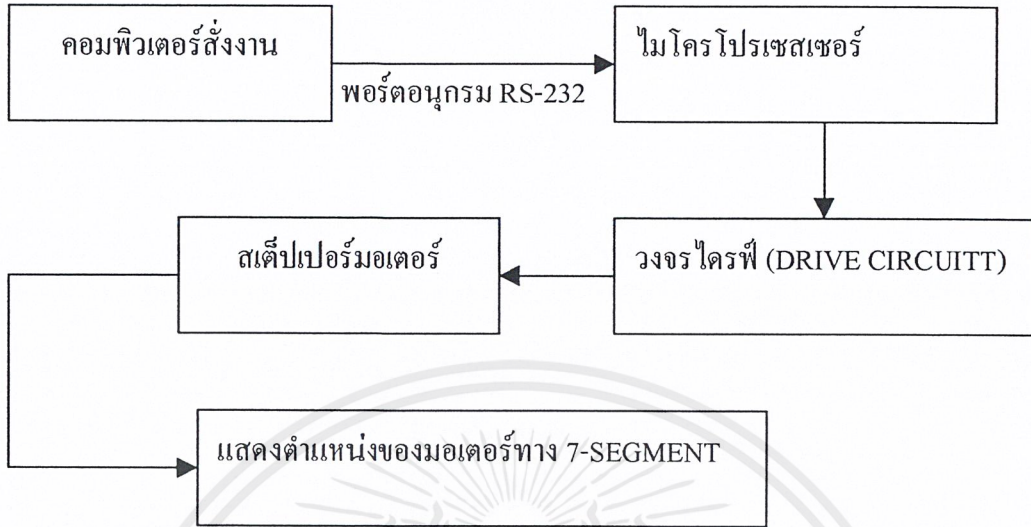
1.1 ความเป็นมาของโครงการ

สืบเนื่องจากปัญหาทางเศรษฐกิจและสังคม ทำให้ความปลอดภัยในทรัพย์สินตลอดจนกระทั่งของชีวิตลดน้อยลงไปตามลำดับ จึงมีแนวความคิดที่จะประยุกต์เทคโนโลยีใหม่ ๆ นำมาใช้ในการตรวจสอบและเตือนภัยที่อาจเกิดขึ้นจากอันตรายในรูปแบบต่าง ๆ จากการที่ราคาของไมโครคอมพิวเตอร์ลดลงมาก จนเกือบทุกสำนักงานหรือร้านค้านำมาใช้ในด้านบริหารจัดการอย่างแพร่หลาย นอกจากนี้อุปกรณ์ตรวจจับเช่นกล้องวิดีโอมีการพัฒนาให้มีขนาดเล็กและราคาถูกลงมาก ดังนั้นหากนำมาประกอบเข้ากับระบบอิเล็กทรอนิกส์ที่ออกแบบขึ้นมาเพื่อเชื่อมต่อระบบโทรศัพท์ สามารถจะพัฒนาเป็นระบบตรวจสอบและเตือนภัยผ่านทางระบบโทรศัพท์เมื่อตรวจจับได้ผ่านทางกล้องวิดีโอ

1.2 ลักษณะของโครงการ

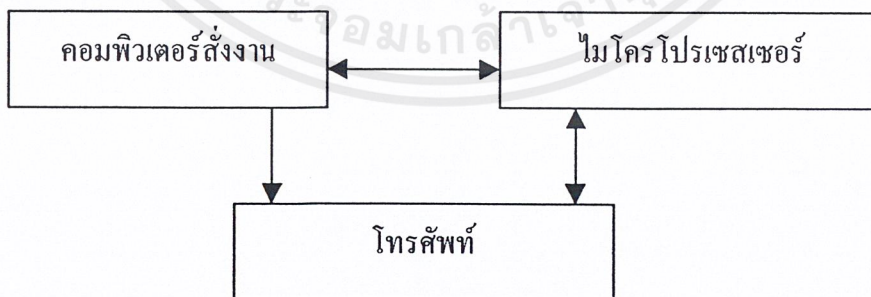
จากแนวคิดข้างต้นได้กำหนดโครงสร้างของระบบในเบื้องต้นดังนี้ กล้องวิดีโอขนาดเล็ก จะถูกติดตั้งบนแท่นสามารถหมุนรอบแกนไม่เกิน ± 90 องศา สามารถควบคุมการหมุนด้วยไมโครคอมพิวเตอร์ในสองโหมดคือ โหมดกำหนดเองด้วยการคลิกเมาส์บนเมนูหรือโหมดสแกนต่อเนื่องอัตโนมัติ ภาพจากกล้องจะเชื่อมต่อเข้ากับไมโครคอมพิวเตอร์ผ่านทางการ์ดวิดีโอและแสดงผลบนเมนูอย่างต่อเนื่อง ผู้ควบคุมสามารถคลิกเมาส์เพื่อบันทึกภาพนิ่งลงในไฟล์ได้ตลอดเวลา เมื่อต้องการแจ้งข่าวหรือข้อมูลใด ๆ ไปยังเลขหมายที่กำหนดไว้สามารถทำได้โดยกดปุ่มเพียงปุ่มเดียว ข้อความที่บันทึกในรูปแบบไฟล์เสียงจะถูกส่งผ่านการ์ดเสียงไปยังเบอร์ผู้รับเมื่อสัญญาณกระดิ่งดังครบสามครั้ง การติดต่อทางโทรศัพท์จะสิ้นสุดอัตโนมัติเมื่อสิ้นสุดข้อความ

โครงการระบบรักษาความปลอดภัยผ่านทางกล้องวิดีโอตามแนวคิดข้างต้น ใช้ไมโครคอมพิวเตอร์ (PC) มาใช้ส่งคำสั่งควบคุมไปยังไมโครโปรเซสเซอร์ แล้วไมโครโปรเซสเซอร์ก็ควบคุมการหมุนของสเต็ปเปอร์มอเตอร์มีส่วนประกอบหลักของโครงการดังแสดงในบล็อกไดอะแกรม



รูปที่ 1.1 บล็อกไดอะแกรมของโครงการส่วนควบคุมก่อกอง

จากรูปที่ 1.1 บล็อกไดอะแกรมนี้อธิบายได้คือโครงการนี้มีสเต็ปเปอร์มอเตอร์ 1 ตัวโดยควบคุมตำแหน่งการหมุนโดยไมโครโปรเซสเซอร์เบอร์ 8051 และไมโครโปรเซสเซอร์ก็จะถูกควบคุมโดยไมโครคอมพิวเตอร์อีกที โดยที่แกนหมุนของมอเตอร์จะมีกล้อวงรีติดอยู่โดยจุดประสงค์ในการหมุนของมอเตอร์ก็เพื่อให้กล้อสามารถเก็บภาพได้ทุกทิศทาง การกำหนดตำแหน่งของมอเตอร์ใช้การนับสเต็ปการหมุนของมอเตอร์เป็นตัวกำหนดตำแหน่งเป็นตัวเลข



รูปที่ 1.2 บล็อกไดอะแกรมของโครงการส่วนควบคุมโทรศัพท์เตือนภัย

จากรูปที่ 1.2 บล็อกไดอะแกรมนี้สามารถอธิบายได้ดังนี้คือ เมื่อมีการตรวจจับได้ว่ามีผู้บุกรุกจะมีการเตือนภัยโดยโทรศัพท์ไปยังหมายเลขที่ตั้งไว้ กล่าวคือเมื่อมีการตรวจจับได้ว่ามีผู้บุกรุก ไม่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โคร โพรเซสเซอร์จะส่งสัญญาณไปบอกคอมพิวเตอร์โดยการติดต่อระหว่างคอมพิวเตอร์กับ ไมโคร โพรเซสเซอร์ติดต่อผ่านทางพอร์ตอนุกรมRS232 จากนั้นคอมพิวเตอร์จะหมุนหมายเลขโทรศัพท์ที่ได้ตั้งไว้ และเพื่อให้แน่ใจได้ว่ามีผู้รับสายปลายทางแน่นอนมีการตรวจจับสัญญาณสายไม่ว่างและสายว่าง ในกรณีที่สัญญาณสายไม่ว่างจะมีการหมุนหมายเลขใหม่ จนกว่าจะมีสัญญาณสายว่าง และในกรณีที่สัญญาณสายว่างก็จะมีการตรวจจับอีกว่ามีคนรับสายที่ปลายทางหรือยัง ถ้าไม่มีคนรับก็จะมีการหมุนหมายเลขใหม่เช่นเดียวกับกรณีสายไม่ว่าง และถ้ามีคนรับสายแล้วคอมพิวเตอร์จะส่งวางหูโทรศัพท์สิ้นสุดการทำงานของส่วนโทรศัพท์เดือนถัด

โครงการนี้ได้เพิ่มส่วนตั้งงานปิดเปิดอุปกรณ์ไฟฟ้าผ่านทางพอร์ตอนุกรมด้วย การปิดเปิดอุปกรณ์ไฟฟ้าสามารถตั้งงานด้วยคอมพิวเตอร์ได้

1.3 เนื้อหาของปฏิญญาพันธ

ปฏิญญาพันธฉบับนี้เป็นผลจากการศึกษาและทดลองตลอดสองภาคการศึกษา เพื่อออกแบบสร้าง ระบบรักษาความปลอดภัยด้วยกล้องวิดีโอ ซึ่งเนื้อหาของปฏิญญาพันธจะประกอบด้วย ส่วนต่าง ๆ แยกเป็นบท ๆ ไปดังนี้

บทที่ 1 บทนำ กล่าวถึงความจำเป็นของโครงการ

บทที่ 2 กล่าวถึง ทฤษฎีและหลักการ โดยสังเขปของ อุปกรณ์ และระบบที่เกี่ยวข้อง ที่นำมาประกอบเป็นโครงการนี้

บทที่ 3 กล่าวถึง การออกแบบและการทำงานของวงจร โดยให้รายละเอียด แนวคิดของระบบ อิเล็กทรอนิกส์ที่พัฒนาขึ้นมา

บทที่ 4 กล่าวถึงการทดสอบและผลการทดสอบ แสดงรายละเอียด การปรับแต่งและการทดสอบระบบอิเล็กทรอนิกส์ และซอฟต์แวร์

บทที่ 5 บทสรุป

บทที่ 2

ทฤษฎีและหลักการ

2.1 สเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์เป็นมอเตอร์ประเภทหนึ่งที่ถูกนำไปใช้งานมาก ไม่ว่าจะเป็นส่วนประกอบในหุ่นยนต์ เครื่องจักรกลที่ควบคุมด้วยคอมพิวเตอร์ ต่างก็ใช้สเต็ปเปอร์มอเตอร์เป็นตัวขับเคลื่อน หัวข้อต่างๆที่จะกล่าวก็จะประกอบไปด้วยการทำงานของสเต็ปเปอร์มอเตอร์ ตลอดจนการควบคุมการหมุนและวงจรอิเล็กทรอนิกส์ที่นำมาขับสเต็ปเปอร์มอเตอร์

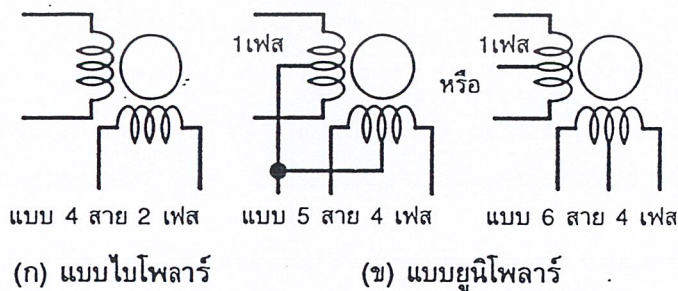
2.1.1 ชนิดของสเต็ปเปอร์มอเตอร์

การแบ่งชนิดของสเต็ปเปอร์มอเตอร์แบ่งได้เป็น 6 ชนิดตามโครงสร้างพื้นฐานหรือความแตกต่างของรูปแบบของโรเตอร์ดังนี้

1. ชนิดวาริเอเบิลรีลักแตนซ์ (Variable Reluctance: VR)
2. ชนิดไฮบริด(Hybrid)
3. ชนิดเพอร์มาเนนต์แม็กเน็ต(Permanent)
4. ชนิดแรเอิร์ธเพอร์มาเนนต์แม็กเน็ต(Rare Earth Permanent Magnet)
5. ชนิดไบโพลาร์
6. ชนิดยูนิโพลาร์

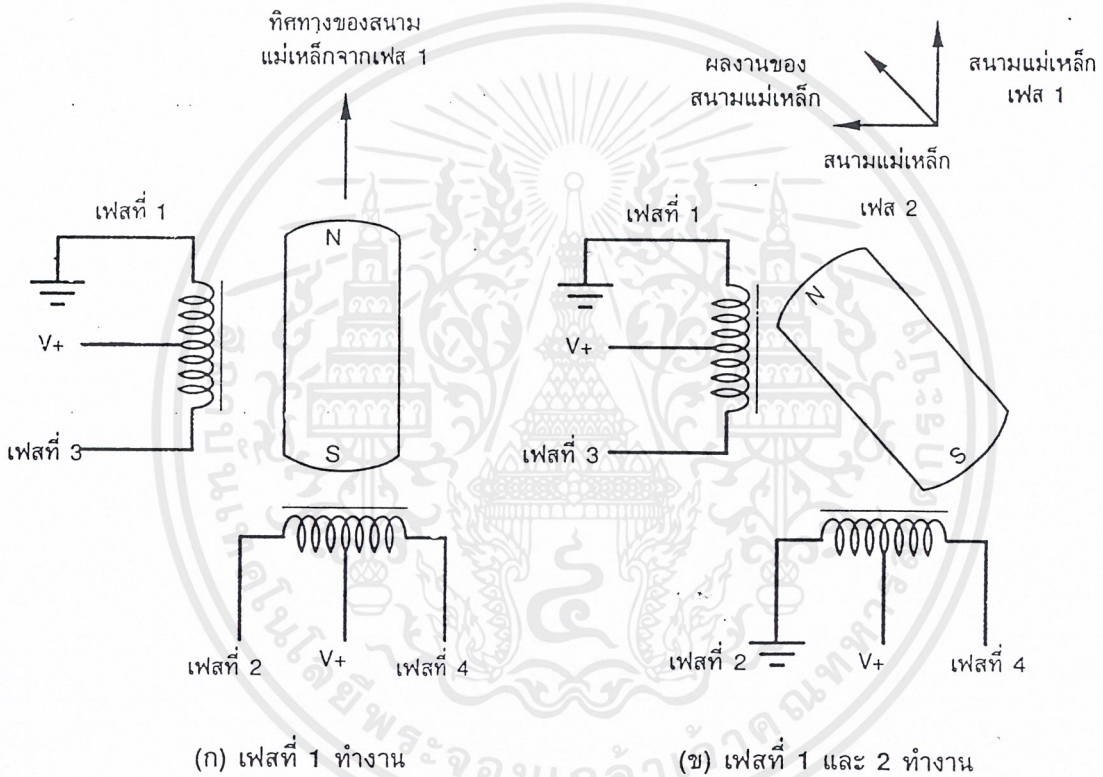
2.1.2 การทำงานของสเต็ปเปอร์มอเตอร์

การทำงานของสเต็ปเปอร์มอเตอร์จะมีความแตกต่างจากมอเตอร์ทั่วไปคือเมื่อป้อนกำลังไฟให้ตัวมัน ก็จะหมุนเพียงเล็กน้อยตามเส้นรอบวงแล้วหยุด สเต็ปเปอร์มอเตอร์สามารถกำหนดตำแหน่งการหมุนได้อย่างละเอียด โดยใช้คอมพิวเตอร์เป็นตัวกำหนดและจัดเก็บตัวเลขของตำแหน่งเหล่านั้นไว้



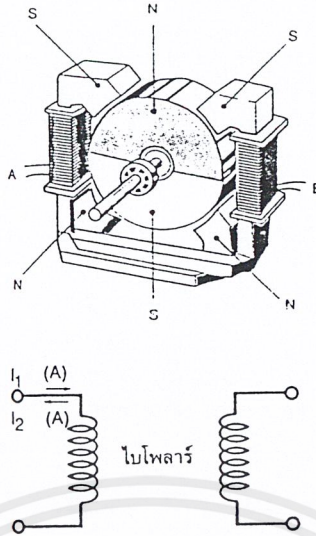
รูปที่ 2 .1 ลักษณะการพันขดลวดบนสเตเตอร์

หลักการทํางานอย่างง่ายของสเต็ปเปอร์มอเตอร์ แบบยูนิโพลาร์ 4 เฟส ตัวโรเตอร์เป็นแม่เหล็ก การให้พลังงานแก่ขดลวดใดขดลวดหนึ่ง โรเตอร์ก็จะหมุนไป 90 องศา ดังรูปที่ 2.2(ก) แต่ถ้าให้ที่เดียว 2 ขดลวดพร้อมกัน โรเตอร์ก็จะหมุนเพียง 45 องศา ดังรูปที่ 2.2(ข) ซึ่งแบบหลังจะสร้างแรงบิดได้มากกว่าแบบแรก สเต็ปเปอร์มอเตอร์จะมีมุมในการเคลื่อนที่แต่ละสเต็ปเป็น 1.8 องศา ฉะนั้นที่โรเตอร์จะต้องมีขั้วแม่เหล็ก 50 ขั้ว ($90/50 = 1.8$)



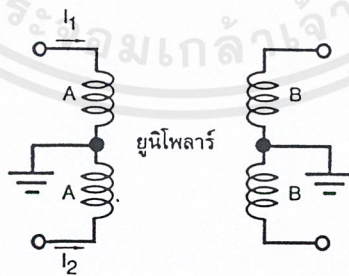
รูปที่ 2.2 ทิศทางการหมุนโรเตอร์ของสเต็ปเปอร์มอเตอร์ 4 เฟส

ส่วนรูปที่ 2.3 เป็นการพันขดลวดแบบไบโพลาร์ เมื่อขดลวด A และ B มีกระแสไหลผ่านเป็นผลให้โรเตอร์ที่มีขั้วแม่เหล็กคํากับสเต็ปเปอร์ถูกดูดคํามาเมื่อกระแสที่ไหลในขดลวด A เปลี่ยนทิศทางกลับ จึงผลให้ขั้วแม่เหล็กที่แกน A เปลี่ยนจากขั้ว S เป็นขั้ว N จากขั้ว N เป็นขั้ว S สลับกัน โรเตอร์จึงถูกผลักให้หมุนทวนเข็มนาฬิกาไป 90 องศา ถ้าคํับการหมุนใน 1 รอบเป็นสเต็ปดังนี้ AB, AB, AB, AB, AB มี 4 สเต็ปๆละ 90 องศา เครื่องหมายขีด (-) บนอักษร A,B แทนการกลับขั้ว



รูปที่ 2.3 ไบโพลาร์สเต็ปเปอร์มอเตอร์แบบ 2 เฟส สนามแม่เหล็กจะเปลี่ยนเมื่อกลับทิศทาง การไหลของกระแส

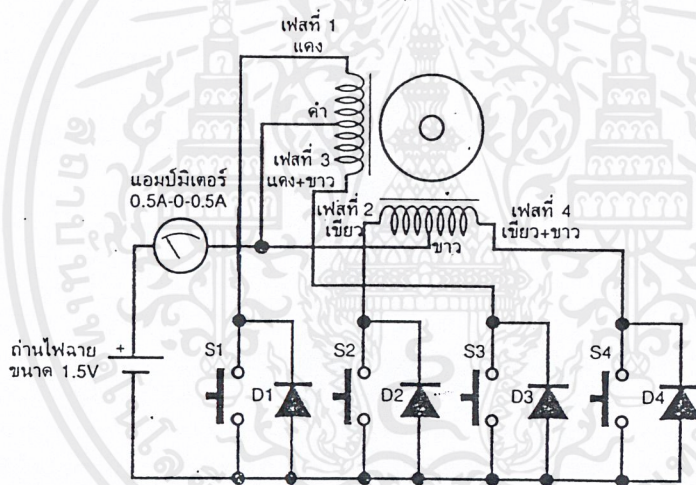
จะสังเกตได้ว่าเมื่อกลับขั้วแม่เหล็กในแต่ละเฟสจะต้องมีการหยุดกระแสก่อนแล้วกระแสจึงค่อยเปลี่ยนทิศทาง จึงสรุปเป็นสเต็ปได้ดังนี้ AB, B, AB, A, B, AB, A, AB การทำงานเป็นแบบครึ่งสเต็ปนี้ เป็นผลให้ค่าโมเมนต์มีค่าน้อยกว่าปรกติเพราะมีช่วงเวลาที่กระแสไหลแค่เฟสเดียว ส่วนแบบยูนิโพลาร์ก็คล้ายกับแบบไบโพลาร์โดยคิดขดเดียวในแต่ละเฟสของยูนิโพลาร์จะมีแทปกกลาง ซึ่งจะแบ่งเป็น 2 ขด ดังรูปที่ 2.4 เป็นผลให้ค่าฟลักซ์แม่เหล็กมีค่าน้อยกว่าไบโพลาร์ ดังนั้นเมื่อสนามแม่เหล็กเปลี่ยนแปลงกระแสจะไม่เปลี่ยนทิศทางไหล สนามแม่เหล็กที่ได้ก็น้อยตาม แรงบิดที่ขึ้นกับสนามแม่เหล็กก็น้อยกว่าด้วย



รูปที่ 2.4 ยูนิโพลาร์สเต็ปเปอร์มอเตอร์การเปลี่ยนขั้วแม่เหล็กเกิดจากกระแสไหลต่างขดกัน กระแสจะไม่ไหลพร้อมกัน 2 ขดในสเตเตอร์เดียวกัน

2.1.3 การควบคุมการหมุนของสเต็ปเปอร์ มอเตอร์

วิธีศึกษาการทำงานของสเต็ปเปอร์มอเตอร์ มีวิธีง่ายๆ ดังรูปที่ 2.5 โดยป้อนแรงดัน ขนาด 1.5 โวลต์ มาใช้ขับให้เกิดแรงบิดขึ้นที่ตัวมอเตอร์ใช้สวิทช์กด 4 ตัว ควบคุมมอเตอร์ให้ทำงานแบบเวฟหรือครึ่งสเต็ป ไดโอด D1 -D4 ป้องกันการสปาร์คที่หน้าคอนแท็คสวิทช์และแรงดันย้อนกลับ สมมติว่าขดลวดเฟสที่ 1 สวิทช์ S1 ปิดวงจรจะมีกระแสไหลผ่านขดลวดเฟสที่ 1 ลงกราวด์ และเมื่อสวิทช์ S1 เปิดวงจรก็จะทำให้เกิดกระแสไหลผ่านทางขดลวดเฟสที่ 3 โดยผ่านทางไดโอด D3 ในขณะที่เดียวกันก็จะมีแรงดันตกคร่อมขดลวด 2 ขณะเปิดวงจร(เป็นแรงดันตกคร่อมไดโอด) ซึ่งจะมีขนาดแรงดันเป็น 2 เท่าของแรงดันในตอนแรกเพราะฉะนั้นการเลือกใช้ทรานซิสเตอร์ต้องคำนึงถึงข้อนี้ด้วยเมื่อนำมาใช้ในการขับมอเตอร์แบบยูนิโพลาร์



รูปที่ 2.5 วงจรทดลองการทำงานของสเต็ปเปอร์

การกระตุ้นเพื่อควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปในการใช้งานจริงทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนแบบซีควเอนเชียลในรูปแบบที่ถูกต้อง แบ่งเป็น 3 รูปแบบคือ

แบบเวฟ (Wave)

เป็นการกระตุ้นรูปแบบที่ง่ายที่สุด โดยกระตุ้นขดลวดทีละขดในเวลาหนึ่งและเรียงถัดกันไป เช่น ขดที่ 1,2 ,3,4,1 หรือ 1,4,3,2,1 ขึ้นอยู่กับทิศทางที่ต้องการให้หมุน ดังนั้นจึงมีขดลวดเพียง 3

ขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น วงจรกระตุ้นแบบเวฟจึงมีราคาถูกและง่าย ขั้นตอนการทำงานต่างๆ แสดงดังตารางที่ 2.1

สเต็ปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

ตารางที่ 2.1 แสดงการกระตุ้นขดลวดแต่ละเฟสแบบเวฟ

แบบ 2 เฟส (Two Phase)

เป็นการกระตุ้นอีกรูปแบบหนึ่งซึ่งคล้ายกับแบบเวฟ แต่จ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และถัดเรียงไปเช่นเดียวกับแบบเวฟคือขดลวดที่ 12,23,34,41,12 หรือ 14,43,32,21,14 ขึ้นอยู่กับทิศทางที่ต้องการให้หมุนการเพิ่มจำนวนของขดลวดที่ที่ถูกกระตุ้นนี้ทำให้เพิ่มแรงบิดได้มากกว่าแบบเวฟ แต่ต้องใช้แหล่งจ่ายไฟกำลังไฟฟ้ามากขึ้น การทำงานแสดงที่ตารางที่ 2.2

แบบครึ่งสเต็ป (Half Step)

เป็นการผสมผสานกันระหว่างแบบเวฟและแบบ 2 เฟสเพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกหนึ่งเท่าตัว จะกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ ขดลวด 1, 12 ,23 , 3, 34 , 4,41, 1หรือในการหมุนอีกทิศทางหนึ่งจะได้เป็น 1, 14,4 ,43 ,3 ,32 ,2 ,21 ,1แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีกเพราะช่วงสเต็ปมีระยะสั้นลงและแต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขด ที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องระวังไว้ว่าเมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องการหมุนถึง 2 สเต็ป จึงจะได้เท่ากับ 1 สเต็ป เต็มเหมือนกับในการควบคุม 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้เทียบเท่ากับแบบ 2 เฟส จึงจะเพียงพอขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2.3

สเคิปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

ตารางที่ 2.2 แสดงการกระตุ้นขดลวดแต่ละเฟสแบบ 2 เฟส

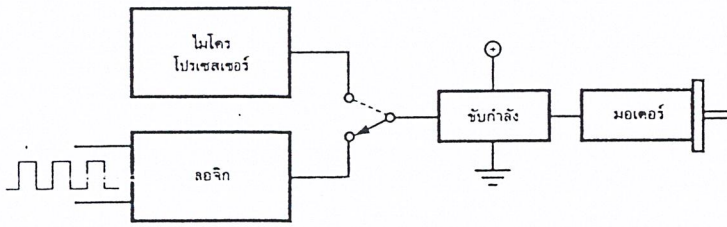
สเคิปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

ตารางที่ 2.3 แสดงการกระตุ้นขดลวดแต่ละเฟสแบบครึ่งสเคิป

วงจรมอเตอร์

การนำเอาวงจรมอเตอร์อิเล็กทรอนิกส์มาใช้กับสเคิปเปอร์มอเตอร์ แสดงดังบล็อกไดอะแกรมรูปที่ 2.6 การขับด้วยวงจรมอเตอร์อิเล็กทรอนิกส์ก็ต้องแยกด้วยว่าขั้วมอเตอร์แบบไบโพลาร์หรือยูนิโพลาร์ อีกทั้งยังขึ้นอยู่กับเฟสของมอเตอร์ด้วยว่ามีจำนวนเฟสอย่างน้อยแค่ไหนที่จะต้องควบคุมวงจรมอเตอร์แบบยูนิโพลาร์ นั้นแสดงไว้ในรูปที่ 2.7 จากวงจรจะเห็นว่าจะใช้ทรานซิสเตอร์ขับขดลวดมอเตอร์ในลักษณะ 1 ขดต่อทรานซิสเตอร์ 1 ตัวซึ่งจะแตกต่างจากขั้วมอเตอร์แบบไบโพลาร์ ขดลวดทางซ้ายมือต้องใช้ทรานซิสเตอร์ถึง 4 ตัว ต่อวงจรแบบบริดจ์มาขับขดลวด 1 ขด ส่วนขดลวดทางขวามือนั้นใช้ทรานซิสเตอร์ 2 ตัวมาขับขดลวด 1 ขด ดังรูปที่ 2.7 (ข) แต่แหล่งจ่ายไฟที่จ่ายให้กับทรานซิสเตอร์ 2 ชุดจะไม่เหมือนกัน

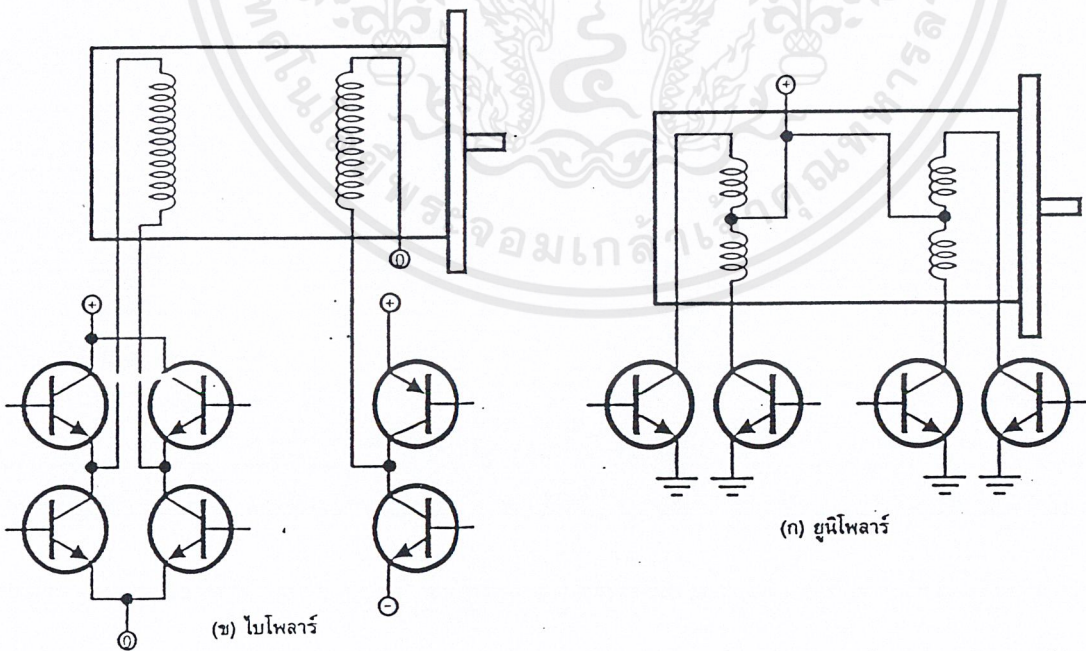
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 บล็อกไดอะแกรมควบคุมด้วยสแต็ปเปอร์มอเตอร์

ในรูปที่ 2.8 เป็นวงจรจับสแต็ปเปอร์มอเตอร์ โดยมีกราฟแสดงการเพิ่มขึ้นของกระแสที่มอเตอร์ ซึ่งในรูปที่ 2.8 (ก) ตัวต้านทานที่ต่ออนุกรมเข้าไปในนั้นจะปลดค่าคาบเวลาในการสวิตช์ของทรานซิสเตอร์ให้น้อยลงสำหรับใช้กับโหลดที่มีค่าเหนี่ยวนำน้อย ซึ่งก็แน่นอนว่าจะต้องมีการสูญเสียกำลังงานไปบางส่วน

ในรูปที่ 2.8(ข) เป็นวงจรที่ช่วยชดเชยค่าอินดักแตนซ์ โดยใช้วงจร RC ซึ่งวงจรนี้จะกำเนิดแอมป์ออซซิเลชันและทำให้ค่าคงที่ ในการแอมป์น้อยที่สุด กำหนดด้วยค่า R และ C คุณสมบัติเหล่านี้จะถูกกำหนดโดยผู้ผลิตสแต็ปเปอร์มอเตอร์



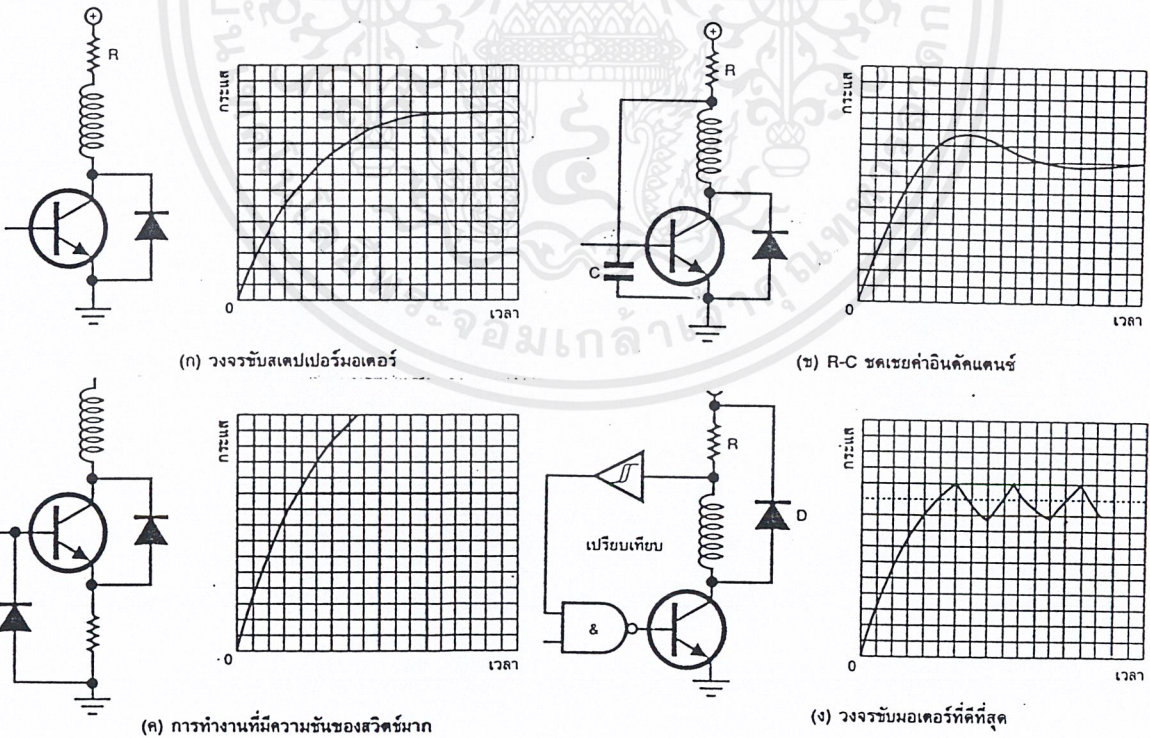
รูปที่ 2.7 การจับสแต็ปเปอร์มอเตอร์แบบยูนิโพลาร์และไบโพลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.8(ค) ใช้ทรานซิสเตอร์เป็นตัวขับเคลื่อน ซึ่งจะทำความชื้นของการสวิตช์ทำงานมีความชื้นมากและจะต้องป้อนแรงดันให้สูงเพียงพอด้วยขณะที่กระแสไหลผ่านทรานซิสเตอร์ในระดับค่าแล้วในรูปที่ 2.8(ค) จะไม่อยู่ในสภาวะอิ่มตัวนานนัก วงจรนี้จะสูญเสียพลังงานมาก จึงต้องระบายความร้อนออกมาในช่วงจังหวะนี้

ในรูปที่ 2.8(ง) เป็นวงจรขับเคลื่อนที่ลึกลับกว่าวงจรหนึ่ง เพราะเมื่อกระแสไหลจนถึงค่าๆหนึ่ง ทำให้ทรานซิสเตอร์หยุดทำงานและจะทำให้สนามแม่เหล็กลดลง โยผ่านไดโอดอย่างช้าๆแล้วเมื่อกระแสตกพร้อมมอเตอร์ลดลงต่ำกว่าค่าที่กำหนดไว้ ทรานซิสเตอร์ก็จะไม่สูญเสียพลังงานมากเหมือนในรูปที่ 2.8 (ค)

ถ้าต้องการควบคุมสเต็ปเปอร์มอเตอร์ด้วยคอมพิวเตอร์ เพื่อให้มอเตอร์ทำงานตามต้องการสามารถต่อเข้าที่พอร์ทโดยตรง และสามารถสั่งงานโดยการใช้ซอฟต์แวร์ เพื่อสั่งงานให้มอเตอร์หมุนเดินหน้าหรือถอยหลังได้ตามต้องการและนอกจากนี้ยังสามารถเลือกได้ว่าจะหมุนเป็นเวฟหรือครึ่งสเต็ปก็ได้ จากการปรับช่วงห่างระหว่างเวลาแต่ละสเต็ป วิธีการควบคุมก็โดยอาศัยวงจรลอจิกนำเอาที่พอร์ทมาควบคุมทรานซิสเตอร์ผ่านวงจร RS ไบสเตเบิล เพื่อป้องกันข้อผิดพลาดที่อาจเกิดขึ้น เช่นกรณีของการนำกระแสพร้อมๆกันของทรานซิสเตอร์ทั้ง 4 ตัวแบบบริดจ์ ในบางกรณีลอจิกเกตจะถูกใช้เป็นตัวรีเซ็ตวงจรไบสเตเบิลเพื่อกำหนดทิศทางการหมุนก็ได้



รูปที่ 2.8 การใช้กระแสขับเคลื่อนเพื่อเพิ่มโมเมนต์เมื่อมีอัตราเร่งสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีอุปกรณ์วงจรรวมหรือไอซีที่ผลิตขึ้นมาเพื่อใช้ในการออกแบบเป็นวงจรควบคุมสเต็ปเปอร์มอเตอร์ ที่ควบคุมความเร็วด้วยการเปลี่ยนแปลงความเร็วของพัลส์ และที่มีขายกันตามท้องตลาดก็อย่างเช่น เบอร์ SAA1027, L297, และ L298 หรือ TL376, ULN2002-UL2005, เป็นต้น

2.2 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ คือไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวมเพียงชิพเดียวเหมาะสำหรับงานควบคุมอุปกรณ์อื่นแบบอัตโนมัติ เพราะสามารถเขียนโปรแกรมควบคุมได้ตามต้องการ MCS-51 มีข้อดีดังนี้

1. สามารถนำข้อมูลมา AND, OR หรือทำการ COMPLEMENT ทั้งทีละ 8 บิต หรือ 1 บิต
2. สามารถใช้กับหน่วยความจำสำหรับเก็บโปรแกรม ซึ่งเป็นหน่วยความจำที่ใช้เก็บชุดคำสั่งที่จะทำให้ MCS-51 ทำงานได้สูงสุด 64 กิโลไบต์ทำให้เขียนโปรแกรมควบคุมได้มาก
3. สามารถติดต่อหน่วยความจำสำหรับข้อมูล ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการทำงานของโปรแกรมได้สูงสุด 64 กิโลไบต์
4. ใน 8051 และ 8751 มีหน่วยความจำสำหรับโปรแกรมจำนวน 4 กิโลไบต์ อยู่ภายในวงจรทำให้ไม่ต้องต่อหน่วยความจำสำหรับโปรแกรมภายนอก ระบบรวมทั้งหมดจึงมีขนาดเล็กและมีสัญญาณรบกวนภายนอกน้อย
5. มีพอร์ทแบบขนาน สำหรับข้อมูลเข้าและออกจำนวน 32 บิต ที่ข้อมูลแต่ละบิตเป็นอิสระต่อกัน
6. มีวงจร Timer/Counter ขนาด 16 บิต 2 ชุด ที่ทำงานได้ 4 โหมด
7. มีแหล่งกำเนิดสัญญาณขอ Interrupt 6 แหล่งซึ่งสามารถกระโดดไปทำงานได้ 5 ตำแหน่ง

ที่กล่าวมานี้ถูกบรรจุรวมอยู่ในวงจรรวมขนาด 40 ขา ซึ่งสามารถออกแบบระบบการทำงานให้มีขนาดเล็กและสามารถตรวจสอบข้อผิดพลาดได้ง่าย

2.2.1 โครงสร้างของ 8051

ภายใน 8051 ประกอบด้วย gate ต่างๆ ซึ่ง gate เหล่านี้ถูกนำมาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง วงจรสร้างสัญญาณนาฬิกา โครงสร้างภายในของ 8051 จะประกอบด้วย

1. CPU (Central Processor Unit) หรือตัวประมวลผล ส่วนนี้จะมีวงจรที่ทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่นๆ เรียกว่า วงจรควบคุม สัญญาณที่สร้างจากวงจรควบคุมได้แก่สัญญาณสำหรับการติดต่อกับหน่วยความจำ, อุปกรณ์ รับข้อมูลหรือส่งข้อมูลออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัว 8051 ซึ่งส่วนควบคุมการขัดจังหวะ(Interrupt Control) และส่วนควบคุมบัส (Bus Control) ก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณควบคุมจากส่วน CPU นี้ จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่งที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์เพื่อให้ทุกส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง ใน CPU นี้จะประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่าส่วนประมวลผล (ALU) ส่วนนี้จะทำหน้าที่ประมวลผลข้อมูล เช่น การบวก ลบ คูณ หรือหารแล้วนำผลที่ได้เก็บไว้ใน Register หรือหน่วยความจำที่ต้องการ

2. หน่วยความจำ มีไว้สำหรับจดจำข้อมูล ถ้าต้องการที่จะใช้หน่วยความจำ จำเป็นที่จะต้องรู้ตำแหน่งของหน่วยความจำ การนำเอาข้อมูลไปเก็บในหน่วยความจำเรียกว่าการเขียนข้อมูล และการนำเอาข้อมูลออกจากหน่วยความจำเรียกว่าการอ่านข้อมูล ซึ่งแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลได้เพียงค่าเดียว ในไมโครโปรเซสเซอร์ทั่วไปรวมทั้ง 8051 นั้นข้อมูลในแต่ละตำแหน่งของหน่วยความจำจะมีค่าได้เพียง 8 หลักของเลขฐาน 2 ดังนั้นในแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลมีค่าได้ตั้งแต่ 0-255 (00000000-11111111 ในเลขฐาน2) แต่จำนวนตำแหน่งที่จะเก็บข้อมูลได้ขึ้นกับไมโครโปรเซสเซอร์แต่ละเบอร์ การติดต่อกับหน่วยความจำจะต้องมีสัญญาณ 3 กลุ่ม คือ

1) ค่าตำแหน่งที่ต้องการติดต่อกับหน่วยความจำใน 8051 จะติดต่อกับหน่วยความจำประเภท Program Memory หรือ Data Memory ได้สูงสุดชนิดละ 65536 ตำแหน่ง ดังนั้นการอ้างอิงแต่ละตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น

2) ข้อมูลที่อ่านหรือเขียนกับหน่วยความจำที่ตำแหน่งในข้อที่ 1

3) สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำ เพื่อบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล สัญญาณเหล่านี้จะถูกรวบรวมภายใน 8051 สร้างมาจากวงจรถอดรหัสของคำสั่งที่ 8051 อ่านจากหน่วยความจำ Program Memory เข้าไปทำงานนั่นเอง

3. อุปกรณ์อินพุตและเอาต์พุต เป็นส่วนที่ใช้ส่งข้อมูลเข้าหรือออกจาก 8051 ทำให้ 8051 ติดต่อกับภายนอกได้ อุปกรณ์อินพุตและเอาต์พุตได้แก่ 4 I/O Port, Timer0, Timer1, Serial Port การทำงานของแต่ละส่วนมีดังนี้

1) 4 I/O Port คำว่า Port หมายถึงจุดที่จะติดต่อกับส่วนที่อยู่ภายนอก 4 I/O Port ของ 8051 เป็นที่ใช้สำหรับรับ-ส่งข้อมูล ซึ่งเป็นสัญญาณดิจิทัลเข้าหรือออกจากตัว MCS-51 พอร์ตมีทั้งหมด 4 พอร์ต โดยแต่ละพอร์ตจะรับส่งข้อมูลได้ 8 บิต มีพอร์ต P0, P1, P2 และ P3 บางพอร์ตจะทำงานมากกว่า 1 อย่างก็ได้ เช่น P0 และ P2 จะใช้รับส่งข้อมูลเมื่อติด

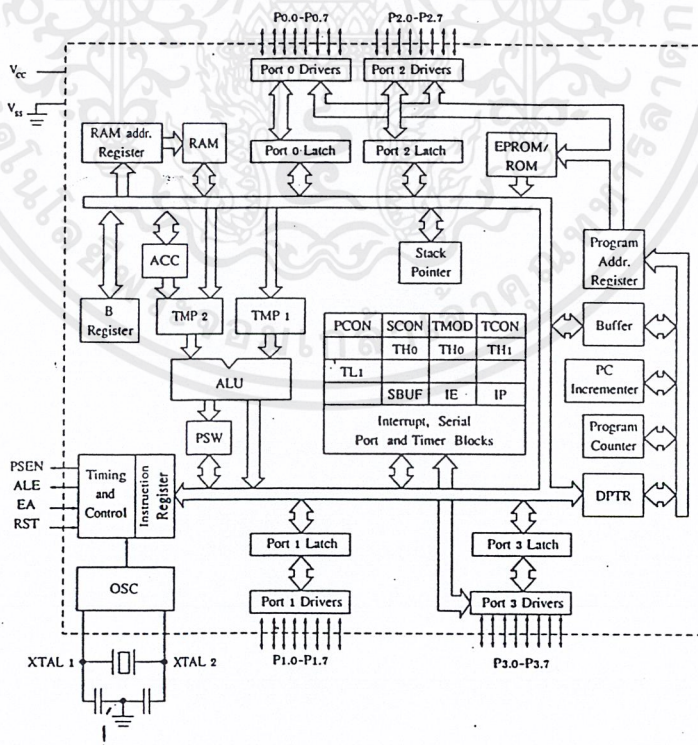
ต่อกับหน่วยความจำได้ด้วย แต่สิ่งเหล่านี้ไม่ได้เกิดขึ้นในเวลาเดียวกัน แต่จะทำงานตามลำดับ โดยควบคุมจากสัญญาณควบคุมที่ถอดรหัสมาจากแต่ละคำสั่งที่ให้คอมพิวเตอร์ทำงานนั่นเอง และสัญญาณทั้งหมดจะอ้างอิงกับสัญญาณนาฬิกา

2) Timer0 และ Timer1 เป็นวงจรมีหน้าที่สามารถกำหนดให้ทำการนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอก 8051 หรือจำนวนไซเคิลของสัญญาณนาฬิกาภายใน 8051 ก็ได้ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดย CPU

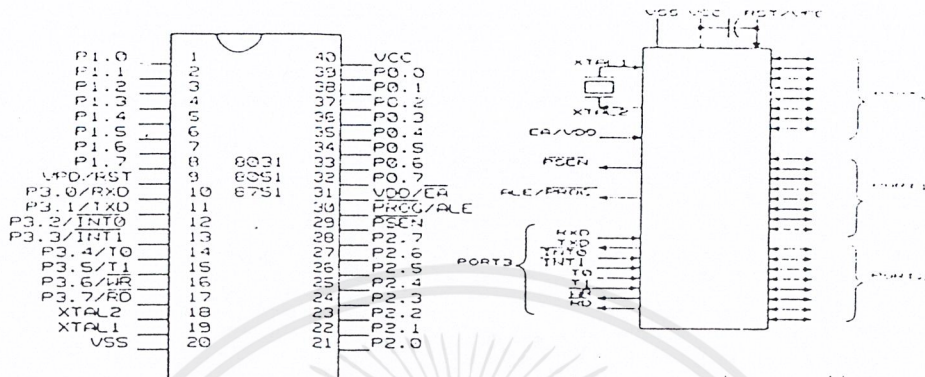
3) พอร์ตอนุกรมหรือ serial CPU จะอ่านหรือเขียนข้อมูลกับ Serial Port เป็นแบบ 8 บิต แต่ข้อมูลจะถูกส่งออกไปทีละบิตออกจกขา TXD และในการรับข้อมูลเข้าก็รับเข้ามาทีละบิตทางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิต เพื่อให้ CPU อ่านไปใช้งาน

2.2.2 สถาปัตยกรรมของ 8051

ในส่วนที่ผ่านมาได้กล่าวถึงโคแอสแกรมภายใน 8051 อย่างกว้างๆ ซึ่งพอจะบอกได้โดยสังเขปว่า ประกอบด้วยส่วนต่างๆ อะไรบ้างในรูปที่ 2.9 เป็นสถาปัตยกรรมของ 8051 ซึ่งเป็นการอธิบายถึงส่วนย่อยๆของภายใน 8051 เพียงชิพเดียว และสัญญาณจากภายใน จะต่อออกสู่ภายนอกทางขาของ 8051 ที่มีอยู่ 40 ขา ดังรูปที่ 2.10



รูป 2.9 สถาปัตยกรรมภายในของ 8051



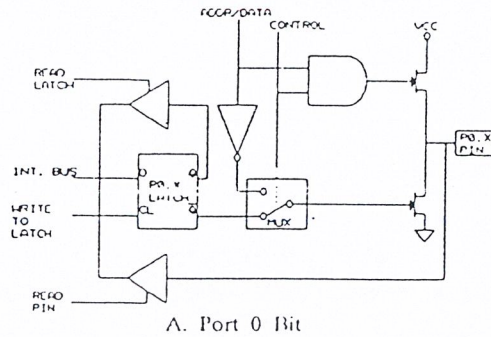
รูป 2.10 โค้ดแแกรมขาของ 8051 แบบ DIP

8051 ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 8051 มีอยู่ข้างละ 20 ขารวมทั้งหมด 40 ขา จะใช้งานต่างกัดังนี้

- Vcc ขา 40 เป็นขาที่ต้องไฟเลี้ยง +5 โวลต์เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับของโวลเตจของลอจิก 0 และ 1 ของ 8051 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL โดยตรง
- Vss ขา 20 เป็นขาที่ต่อกับกราวด์ ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน
- Port 0 เป็น พอร์ตขนานขนาด 8 บิต อยู่ที่ขา 32-39 เริ่มจากบิตที่ 7-0 ตามลำดับ พอร์ต 0 นี้ใช้ได้ทั้งการรับการส่งตำแหน่ง และข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ตรับส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกทางพอร์ต 0 จะถูก Latch ไว้ที่ขาของพอร์ต โครงสร้างของแต่ละบิตของพอร์ต 0 เป็นแบบ Open Drain Bidirectional ดังรูปที่ 2.1

ในรูปที่ 2.11 เมื่อเปรียบเทียบกับรูปที่ 2.9 ส่วนที่ 1 ของรูปที่ 2.11 ก็คือ Port 0 Driver ของรูปที่ 2.9 นั้นเอง จากโครงสร้างในรูปที่ 2.11 เมื่อมีคำสั่งการเขียนข้อมูลมายังพอร์ต 0 ข้อมูลจาก Internal Data Bus จะถูก Latch ไว้ที่ D-FF โดยสัญญาณ "Write to latch" ที่ถูกสร้างมาจากส่วน Timing and Control และในการอ่านข้อมูลจากพอร์ต 0 จะอ่านได้ 2 แบบคือการอ่านข้อมูลที่ส่งไปเก็บไว้ที่พอร์ตก็จะมีสัญญาณ Read Latch มาเพื่ออ่านข้อมูลจาก D-FF กลับเข้าไปยัง Internal Data Bus การอ่านข้อมูลอีกแบบก็คือการอ่านสถานะของสัญญาณที่เข้ามาทางพอร์ต 0 ก็จะมีสัญญาณ Read PIN มาควบคุมการอ่าน พอร์ต 0 จะใช้งานหลายอย่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



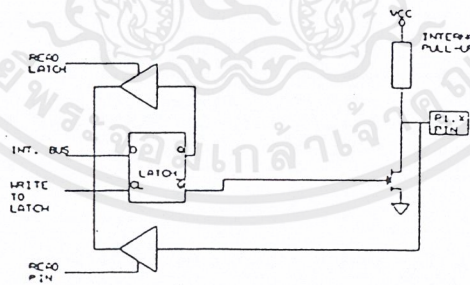
รูป 2.11 โครงสร้างของพอร์ต 0

1. ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อด้วย ตำแหน่งหน่วยความจำสูงสุดที่จะติดต่อได้คือ 64 กิโลไบต์ จึงมีค่าตำแหน่งหน่วยความจำ 16 บิต ของเลขฐาน 2 ค่าตำแหน่งหน่วยความจำ 8 บิตล่าง จะถูกส่งออกทางพอร์ต 0 และ 8 บิตบนจะส่งออกทางพอร์ต 2
2. ใช้รับ-ส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory
3. ใช้รับ-ส่งข้อมูลทางพอร์ตโดยตรง ในกรณีที่ไม่มีการใช้หน่วยความจำของ Program Memory หรือ Data Memory ภายนอก

วงจรภายในส่วน Timing and Control จะเป็นตัวสร้างสัญญาณมาควบคุมในวงจรเพื่อการทำงานแต่อย่างข้างต้น เมื่อแต่ละบิตของพอร์ต 0 ทำงานตามข้อ 1 และข้อ 2 ข้างต้น วงจร Timing and Control จะทำให้สถานะของลอจิกของขา Control เป็น 1 ซึ่งทำให้สวิตช์ MUX อยู่ในตำแหน่งข้างบน เมื่อพอร์ต 0 จะส่งข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำหรือข้อมูลที่จะเขียนออกไปยังหน่วยความจำภายนอกก็จะค่าดังกล่าวมายัง ADDR/DATA ถ้าข้อมูลที่ส่งมาเป็น 1 จะทำให้สัญญาณที่ส่งออกมาจาก AND GATE เป็น 1 และสัญญาณที่ออกมาจาก Inverter เป็น 0 ดังนั้น FET ตัวบน ON ส่วน FET ตัวล่าง OFF สถานะลอจิกที่ขา P0.X PIN จะเป็น 1 แต่ถ้าข้อมูลที่ส่งออกมาเป็น ADDR/DATA เป็น 0 ก็จะทำให้สัญญาณจาก AND GATE เป็น 0 และสัญญาณที่ออกมาจาก Inverter เป็น 1 นั้น FET ตัวบนจะ OFF FET ตัวล่างจะ ON ทำให้สถานะลอจิกที่ขา P0.X PIN เป็น 0 เมื่อ 8051 ต้องการใช้พอร์ต 0 สำหรับการอ่านข้อมูลจากหน่วยความจำภายนอก หรือใช้งานใน 3 ข้อข้างบนก็จะทำได้โดยวงจร Timing and Control ทำให้สถานะลอจิกของสัญญาณ Control เป็น 0 ทำให้เอาท์พุทจาก AND GATE เป็น 0 FET ตัวบนจะ OFF และสวิตช์ MUX จะอยู่ตำแหน่งข้างล่าง ดังนั้น FET ตัวข้างล่างจะ ON หรือ OFF ก็แล้วแต่ข้อมูลที่ขา Q ของ D-FF เมื่อมีการเขียนข้อมูล

จาก Internal Data Bus มายัง D-FF ก็จะมีสัญญาณ "Write to latch" มายัง D-FF ด้วย ถ้าข้อมูลที่เขียนมาเป็น 1 ก็จะทำให้ขา Q มีสถานะลอจิกเป็น 0 ทำให้ FET ตัวล่าง OFF ดังนั้นขา P0.X จะอยู่สถานะ High Impedance เพราะ FET ทั้ง 2 ตัว OFF แต่ถ้าข้อมูลที่เขียนมายัง D-FF เป็น 0 ทำให้ FET ตัวล่าง ON แต่ตัวบน OFF ทำให้สถานะลอจิกที่ขา P0.X เป็น 1 ดังนั้นพอร์ท 0 เมื่อให้ทำงานเป็นพอร์ทส่งข้อมูลจะไม่สามารถแสดงสถานะลอจิกเป็น 1 ได้ จึงต้องต่อตัวต้านทาน Pull Up ไว้ภายนอก ระหว่างขา P0.X กับไฟเลี้ยงวงจร ถ้าใช้พอร์ท 0 สำหรับรับข้อมูลเข้าจะต้องเขียน 1 เก็บไว้มายัง D-FF ก่อน เพื่อให้ขา P0.X อยู่ในสถานะ High Impedance แล้วจึงใช้คำสั่งอ่านสถานะลอจิกเข้าไปยัง Internal Data Bus ต่อไป โดยคำสั่งอ่านสถานะลอจิกทางพอร์ท 0 ก็จะทำให้วงจร Timing and Control สร้างสัญญาณ Read Pin สำหรับการสถานะลอจิกข้างต้น ถ้าไม่เขียน 1 มาเก็บไว้ที่ D-FF ก่อนที่จะอ่านข้อมูลแล้วอาจมีข้อมูลค้างอยู่ที่ D-FF ทำให้ Q เป็น 0 และ Q เป็น 1 ซึ่งทำให้ FET ตัวล่าง ON สัญญาณที่ต่อเข้ามาที่ขา P0.X ไม่ว่าจะมืสถานะลอจิกใดจะถูกดึงลงกราวนด์ ดังนั้นเมื่ออ่านข้อมูลเข้าไปก็พบว่า เป็น 0 เสมอ ในการอ่านข้อมูลจากหน่วยความจำภายนอกนั้นวงจร Timing and Control ก็จะเขียนข้อมูลมายัง D-FF ให้เป็น 1 และสร้างสัญญาณ Control ให้มีลอจิกเป็น 0 ก่อนอ่านข้อมูลเข้าไปด้วย

- Port1 เป็นพอร์ทขนานขนาด 8 บิต . ในรูปที่ 2.10 คือขา P1.0 ถึง P1.7 (ขา1-8) P1.0 หมายถึงบิต 0 พอร์ท 1 เป็นบิต Least Significant Bit และ P1.7 หมายถึงบิต 7 พอร์ท 1 เป็นบิต Most Significant Bit โครงสร้างของพอร์ท 1 แต่ละบิตมีดังรูปที่ 2.12

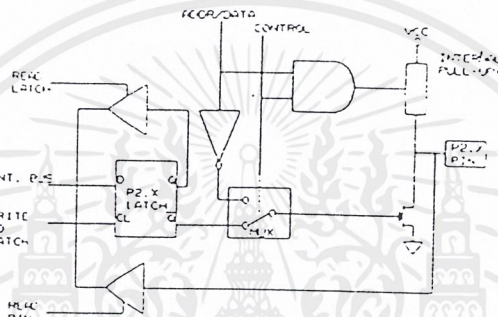


รูป 2.12 โครงสร้างของพอร์ท 1

ส่วนที่ 1 คือ Port 1 Latch ในรูปที่ 2.9 ซึ่งจะมีการทำงานเหมือนส่วนที่ 1 ของพอร์ท 0 ในรูปที่ 2.12 ส่วนที่ 2 คือ Port 1 Driver นี้จะมีตัวต้านทานค่ออยู่เป็น Internal Pull Up พอร์ท 1 นี้จะใช้ทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกทางพอร์ท 1 จะถูก Latch ไว้ก่อนแล้วจึง

ส่งออกไปแต่ละขา ก่อนที่จะอ่านข้อมูลเข้าไปทางพอร์ท 1 จะต้องเขียน 1 ไปยังทุกบิตของพอร์ท 1 เสียก่อนเพื่อให้ FET อยู่ในสภาวะ OFF ก่อน มิฉะนั้นถ้ามีข้อมูล 0 ส่งออกมาตามที่ D-FF จะทำให้ FET อยู่ในสภาวะ ON ดังนั้นถ้าสัญญาณภายนอกส่งเข้ามาที่ขานี้ก็จะถูกดึงวงจรลงกราวด์ โดยไม่สนใจว่าเป็นลอจิกอะไรข้อมูลที่ผ่านจึงเป็น 0

- Port 2 เป็นพอร์ทขนานขนาด 8 บิต คือขา P2.0 ถึง P2.7 โครงสร้างของพอร์ท 2 จะมีดังรูปที่ 2.13



รูป 2.13 โครงสร้างของพอร์ท 2

ลักษณะจะเหมือนกับ Port 0 แตกต่างกันใน พอร์ท 2 นั้น ภาค Driver จะใช้งานเพียง 2 ลักษณะคือ

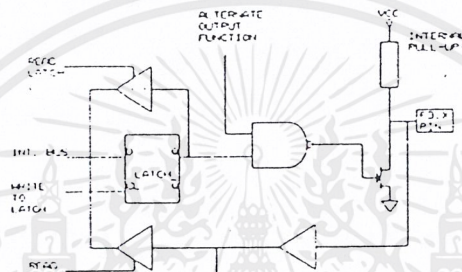
1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิตบนของค่าตำแหน่ง
2. ใช้เป็นพอร์ทรับส่งข้อมูลจากภายนอก

ดังนั้นภาค Driver ของ พอร์ท 2 จึงแตกต่างจาก Driver ของ พอร์ท 0 โดยที่พอร์ท 2 นั้นจะมีเฉพาะ ADDR(ตำแหน่งหน่วยความจำ) เข้ามาที่ MUX (Multiplexer) เท่านั้น นอกนั้นการทำงานจะเหมือนกันและที่เอาท์พุทของพอร์ท 2 จะมี Internal Pull Up ซึ่งเป็นตัวต้านทานและจะทำให้เอาท์พุทของพอร์ท 2 แสดงสภาวะลอจิกเป็น 1 ได้ ถ้า FET อยู่ในสภาวะ OFF บางครั้งเรียกว่า “Quasi-bidirectional” เมื่อใช้เป็นพอร์ทอินพุทก็สามารถทำได้ โดยการต่อสัญญาณจากภายนอกเข้ามาโดยตรง ถ้าสัญญาณภายนอกเป็น 0 ก็จะมีกระแสไหลออกจากพอร์ท (Source Current) ในการที่จะใช้พอร์ทนี้เป็นพอร์ทข้อมูลเข้าจะต้องเขียน 1 ไปยังแต่ละบิตก่อน

- Port 3 คือขา P3.0-P3.7 หรือขา 10-17 มีโครงสร้างดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 ในรูปที่ 2.14 เป็นส่วน Latch ข้อมูลที่เขียนมายังพอร์ท 3 ทาง Internal Data Bus เหมือนกับพอร์ทอื่นๆและพอร์ท 3 จะมี Internal Pull Up อยู่ทุกบิต แต่พอร์ท 3 นี้แต่ละบิตจะใช้ในการทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ Alternative Output Function ที่สร้างมาจากส่วน Timing and Control สัญญาณ Alternative Output Function เป็นสัญญาณที่ส่งออกในกรณีที่ใช้พอร์ท 3 ในการทำงานฟังก์ชันอื่น และ จุด Alternative Input Function เป็นจุดที่เอาสัญญาณไปเข้าส่วนอื่นตามการทำงานของบิตนั้น แต่ละบิตของพอร์ท 3 จะมีฟังก์ชันอื่นดังนี้



รูป 2.14 โครงสร้างของพอร์ท3

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/INT0 (External Interrupt) .ใช้รับสัญญาณ Interrupt ภายนอก

P3.3/INT1 (External Interrupt) .ใช้รับสัญญาณ Interrupt ภายนอก

P3.4/T0 (Timer / Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 0 ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ T0 นี้หรือสัญญาณนาฬิกาได้

P3.5/T1 (Timer / Counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 1 ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ T1 นี้หรือสัญญาณนาฬิกาได้

P3.6/WR (External Data Memory Write Strobe) ขารับสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

- RST ขา Reset จะใช้ทำการ Reset การทำงานของ 8051 ที่ขา Reset ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขา RST กับกราวด์ ถ้าป้อนสัญญาณที่มีสถานะลอจิกเป็น 1 ที่ขา RST จะเป็นการ Reset การทำงานของ 8051 ดังนั้นจึงสามารถต่อตัวเก็บประจุภายนอกระหว่างขา RST กับไฟเลี้ยง เพื่อให้เกิดการ Reset เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งเรียกว่า Power On Reset การ Reset จะทำให้ค่าใน Register ต่างๆ เปลี่ยนไปเป็นค่าหนึ่งดังตาราง 2.4

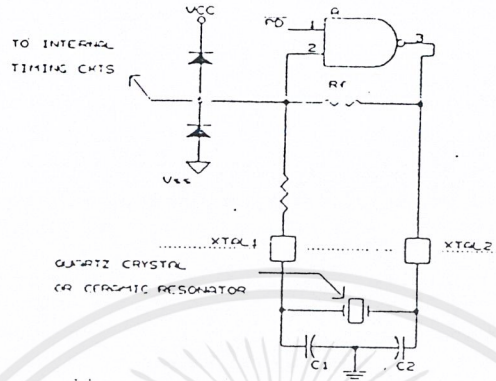
Register	มีค่าข้อมูลเป็น	Register	มีค่าข้อมูลเป็น
PC	00H	T2CON	00H
ACC	00H	TH0	00H
B	00H	TL0	00H
PSW	00H	TH1	00H
SP	07H	TL1	00H
DPTR	00H	TH2	00H
P0-P3	0FFH	TL2	00H
IP(8051)	XXX00000B	RLDH	00H
IP(8052)	XX000000B	RLDL	00H
IE(8051)	0XX00000B	SCAN	00H
IE(8052)	0X000000B	SBUF	INDETERMINATE
TMOD	00H	PCON(HMOS)	0XXXXXXXB
TCON	00H	PCON(CMOS)	0XXX0000B

ตารางที่ 2.4 ค่าของรีจิสเตอร์เมื่อเกิดการรีเซต 8051

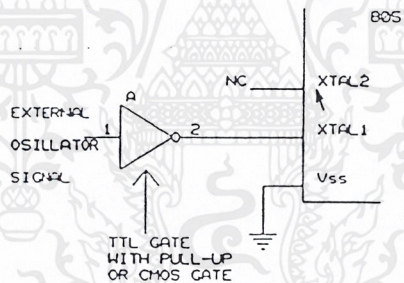
ในตาราง 2.4 ช่องทางขวาเป็นค่าของ Register ที่อยู่ทางซ้ายสุดเมื่อเกิดการ Reset ใน Register SBUF เมื่อสิ้นสุดการ Reset จะมีค่าที่ไม่แน่นอน และพอร์ทจะอยู่ในสถานะลอจิก 1 ทุกบิตตลอดเวลาที่สัญญาณของขา RST เป็น HIGH อยู่

เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็ออกจากการ Reset 8051 จะเริ่มทำงานจากคำสั่งที่อยู่ใน Program Memory ตำแหน่ง 00H เพราะค่าของ Register PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงานถูกเปลี่ยนให้เป็น 00H

- ALE Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกา จากออสซิลเลเตอร์ สัญญาณนี้จะส่งสัญญาณออกมาตลอดเวลาขั้วขาบางขั้วของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 8051 สัญญาณนี้จะบอกกับอุปกรณ์ภายนอก 8051 ว่าขณะนี้สัญญาณนี้ Active จะมีการส่งข้อมูลที่เป็น 8 บิตล่างของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อกออกไปทางพอร์ท 0 อุปกรณ์จะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุท
- PSEN Program Strobe Enable เป็นขาที่ 29 ขานี้ปกติจะให้ลอจิกเป็น 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่งที่จะนำไปทำงานจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 ในกรณีที่อ่านคำสั่ง ซึ่งเก็บอยู่ในหน่วยความจำสำหรับเก็บโปรแกรมภายใน 8051 แล้ว สัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา PSEN นี้สามารถต่อไปยังขาอินพุท TTL ชนิด LS ได้ถึง 8 อินพุท
- EA External Access เป็นขาที่ 31 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timming and Control เพื่อควบคุมการสร้างสัญญาณ เพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่สัญญาณที่ป้อนให้ขา EA เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H-0FFFH ถูกเก็บไว้ภายใน 8051 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่างๆจาก ROM ภายใน 8051
- XTAL 1 เป็นขาที่ 19 ขานี้จะต่อเข้ากับขาของ Inverting Amplifier ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ในรูปที่ 2.15 จะเห็นว่าวงจรภายในของออสซิลเลเตอร์ NAND GATE จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่ควบคุมให้มีการออสซิลเลตหรือไม่ ขึ้นกับสัญญาณ PD ซึ่งต่อมาจากบิต PD ของ Register PCON ถ้าต้องการใช้สัญญาณนาฬิกาจากภายนอกมาเป็นสัญญาณนาฬิกาควบคุมการทำงานของ 8051 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้ แต่ถ้าต้องการใช้วงจรออสซิลเลเตอร์ภายในก็ให้ต่อ Crystal หรือ เซรามิค โชนเตอร์ดังรูปที่ 2.15 ตัวเก็บประจุควรมีค่าประมาณ 20 pF
- XTAL 2 เป็นขาที่ 18 ขานี้เป็นจุดเอาต์พุทของวงจรขยายแบบกลับเฟสสัญญาณที่ประกอบเป็นวงจรออสซิลเลเตอร์ (อินพุทคือขา XTAL 1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกมาเป็นสัญญาณนาฬิกาของ 8051 แล้ว ให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้าที่ขา XTAL 1 ดังรูปที่ 2.16



รูป 2.15 วงจรออสซิลเลเตอร์ภายใน 8051



รูปที่ 2.16 8051 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

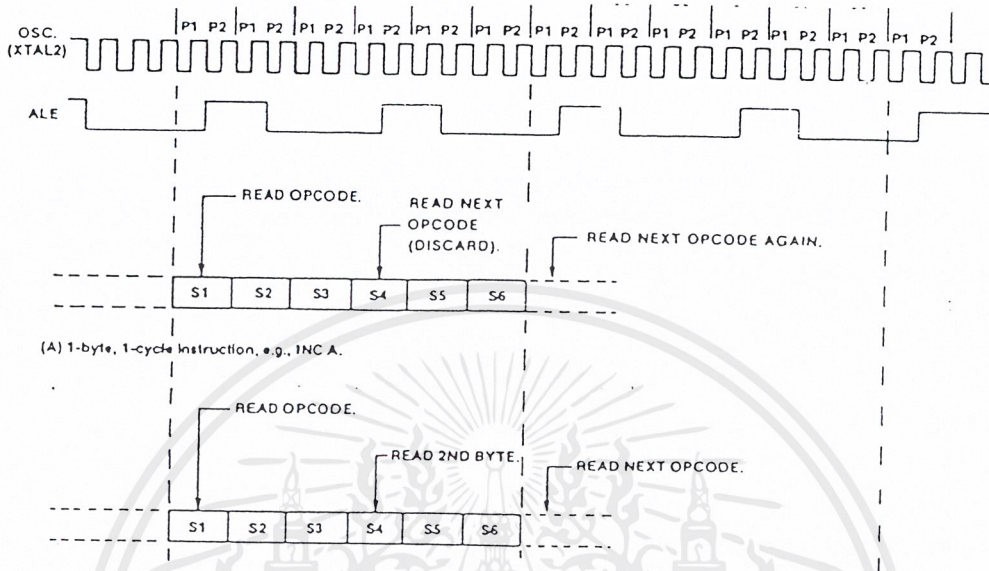
2.2.3 การทำงานของ 8051

จากรูปที่ 2.15 เมื่อป้อนไฟเลี้ยงให้กับ 8051 ซึ่งมีวงจร Power On Reset ต่ออยู่จะมีการ Reset เกิดขึ้น การทำงาน 8051 จะเริ่มจากบล็อก Program Counter ซึ่งเป็นวงจรนับชนิดหนึ่งส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงบนบัสหมายเลขหนึ่ง บัสนี้มีขนาด 16 บิต ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำจะปรากฏที่บัส 16 บิตหมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจาก Reset ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่าเป็น ROM ภายนอกหรือภายใน 8051 โดยการป้อนสถานะลอจิกเข้าไปที่ 8051 ขา EA ซึ่งอยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัสแล้วสร้างสัญญาณควบคุมต่อไป ถ้าป้อนสัญญาณลอจิก 0 เข้าที่ขา EA จะเป็นการเลือกใช้ ROM ภายใน 8051 โดยที่วงจร Timing and Control จะสร้างสัญญาณไปยัง ROM ภายในให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยตำแหน่งที่ส่งมาทางบัสหมายเลข 2 ข้อมูลจาก ROM จะถูกส่งไปยังบัสหมายเลข 3 ที่เรียกว่า Internal Data Bus แล้วไปเก็บไว้ที่ Instruction Register (เป็นวงจร Latch) เพื่อส่งต่อไปให้กับวงจร Timing and Control ทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่นๆ ต่อไปแล้วแต่จะทำให้ทำอะไรในกรณีที่เลือก ROM ภายนอก 8051 โดยการป้อนสัญญาณลอจิก 1 เข้าที่ขา EA จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ท 0 และพอร์ท 2 เพื่อส่งตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ท 0 ไปยัง Internal Data Bus แล้วไปเก็บไว้ที่ Instruction Register เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจาก ROM ภายในการทำงานในช่วงส่งตำแหน่งหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่เป็นคำสั่งกับเข้ามาเก็บไว้ใน Instruction Register เรียกว่าเป็นช่วงของการ Fetch Cycle ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle เช่นถ้าเป็นคำสั่งให้บวกข้อมูลใน Register Accumulator กับข้อมูลจากหน่วยความจำ Data Memory ภายใน RAM ตำแหน่ง 23H วงจร Timing and Control ก็จะส่งสัญญาณให้ Instruction Register ส่งตำแหน่งหน่วยความจำ 23H ไปยัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ RAM ADDR Register เพื่อชี้ตำแหน่งหน่วยความจำ RAM จากนั้น Timing and Control จะสั่งให้ RAM ส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ TMP1 (วงจร Latch) ขณะเดียวกันวงจร Timing and Control ก็จะส่งสัญญาณไปยัง ACC ให้ส่งข้อมูลไปยัง TMP 2 (วงจร Latch) วงจร ALU ซึ่งโครงสร้างเป็นวงจรทำการคำนวณทางคณิตศาสตร์ (บวก, ลบ, คูณ, หาร) และยังสามารถทำงานทางลอจิก (AND, OR, NOT, XOR) จะทำการบวกเลขจาก TMP1 และ TMP2 เข้าด้วยกันผลลัพธ์ที่ได้จะส่งผ่าน Internal Data Bus กลับไปเก็บยัง ACC PSW (Program Status Word) ซึ่งจะทำหน้าที่เก็บสถานะผลลัพธ์ของการทำงานใน ALU เช่นผลลัพธ์การบวกมีค่าเกิน 8 บิตก็จะทำให้บิตหนึ่งใน PSW ถูก SET เป็น 1

การทำงานที่กล่าวมาข้างต้นจะขึ้นกับสัญญาณควบคุมที่สร้างมาจากวงจร Timing and Control และสัญญาณที่สร้างขึ้นนี้จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างมาจากวงจร Oscillator ทำให้การทำงานต่างๆเป็นไปตามลำดับที่ผู้ผลิตได้ออกแบบไว้ดังรูปที่ 2.17



รูป 2.17 ลำดับสถานะการทำงานใน MCS-51

คำสั่งแต่ละคำสั่งใน 8051 จะใช้เวลาในการทำงาน 1,2 หรือ 3 ไชเคล็ดของเครื่อง (Machine Cycle) แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 Machine Cycle จะใช้เวลา 12 Cycles ของสัญญาณนาฬิกา ดังนั้นในแต่ละคำสั่งของ 8051 จะใช้เวลาทำงาน 12,24หรือ 36 ไชเคล็ดของสัญญาณนาฬิกานั้นเอง แต่ละ Machine Cycle จะถูกแบ่งออกเป็น 6 State คือ S1,S2,S3,S4,S5 และ S6 แต่ละ State จะประกอบด้วย 2 ไชเคล็ดของสัญญาณนาฬิกา ในไชเคล็ดแรกเรียกว่าเฟส 1 (P1) และ ไชเคล็ดที่ 2 เรียกว่าเฟส 2 (P2) ในแต่ละเฟสจะนับตั้งแต่ขอบขาของสัญญาณนาฬิกาจนถึงขอบขาของสัญญาณนาฬิกาถัดไปดังที่เห็นในรูปที่ 2.17 เมื่อ 8051 ทำงานเสร็จ 1 Machine Cycle ก็จะเริ่มทำงาน State1 Phase1(SIP1) ของไชเคล็ดต่อไปใน 1 ไชเคล็ดของวงจร Timing and Control จะสร้างสัญญาณ ALE ออกมา 2 ไชเคล็ด เพื่อ Fetch คำสั่งเข้าไป 2 ครั้งเสมอ ที่บริเวณขอบขาขึ้นของสัญญาณ ALE คำสั่งใดจะมีกี่ไบท์หรือใช้เวลาการทำงานกี่ไชเคล็ดสามารถดูได้จากตารางชุดคำสั่ง 8051

คำสั่งประเภท 1 ไบท์ 1 ไชเคล็ดของเครื่องได้แก่ คำสั่ง INCA จะมีการอ่านคำสั่งจากหน่วยความจำสำหรับ โปรแกรม 2 ครั้งที่เวลาประมาณขอบขาขึ้นของสัญญาณ ALE เมื่อคำสั่งแรกถูกอ่าน

เข้าไปที่เวลาขอบขึ้นของสัญญาณ ALE แรก แล้วนำไปเก็บที่ Instruction Register เพื่อให้วงจร Timing and Control ถอดรหัสและเข้าสู่การ Execute ขณะเดียวกันก็จะเริ่มการ Fetch คำสั่งที่อยู่ในหน่วยความจำตำแหน่งถัดไปเข้ามาและคำสั่งที่ 2 จะถูกอ่านเข้ามาที่เวลาขอบขึ้นของสัญญาณ ALE ถัดไป วงจร Timing and Control เมื่อถอดรหัสคำสั่งแรกก็จะทราบว่าการทำงานคำสั่งนี้ให้สิ้นสุดจะใช้คำสั่งเพียง 1 ไบต์ ดังนั้นคำสั่งที่ถูกอ่านมาในไบต์ที่ 2 จะไม่ถูกนำมาทำงานแต่จะอ่านเข้ามาแล้วทิ้งไปในรูปที่ 2.17a

คำสั่งประเภท 2 ไบต์ 1 ไชเคล็ดของเครื่องได้แก่คำสั่ง ADD A,#Data ใน 1 Machine Cycle นี้ จะมีการคำสั่งเข้ามา 2 ไบต์ เหมือนกับคำสั่งประเภท 1 ไบต์ 1 ไชเคล็ดของเครื่อง แตกต่างกันที่ ไบต์ที่ 2 จะถูกนำมาใช้งานด้วยไม่ได้ถูกทิ้งไป ตัวอย่าง ADD A,#33H จะเขียนเป็นภาษาเครื่องได้ 2 ไบต์คือ 24 33 เมื่ออ่านคำสั่งไบต์แรกคือ 24 ไปเก็บไว้ที่ Instruction Register แล้ว Timing and Control จะถอดรหัสพบว่าเป็นคำสั่งบวกเลขก็จะส่งสัญญาณไปที่ Accumulator ให้เอาข้อมูลไปไว้ที่ TMP1 เมื่อคำสั่งที่ 2 ถูกเข้ามาไว้ที่ Instruction Register แล้ว Timing and Control จะสั่งให้เอาข้อมูลไบต์ที่ 2 ส่งไปยัง Internal Data Bus ไปเก็บยัง TMP1 จากนั้นวงจร ALU จะนำเอาข้อมูล TMP1 และ TMP2 มาบวกกันผลที่ได้จาก ALU ไปยัง Internal Data Bus แล้วไปเก็บไว้ที่ Accumulator

คำสั่งประเภท 1,2 และ 3 ไบต์ที่ใช้เวลาการทำงาน 2 ไชเคล็ดของเครื่อง เช่นคำสั่ง INC DPTR จะมีการคำสั่งเข้าไป 4 ครั้งทุกๆขอบขึ้นของสัญญาณ ALE ที่มี 2 ครั้ง ต่อ 1 ไชเคล็ดของเครื่อง ถ้าเป็นคำสั่งประเภท 1,2 และ 3 ไบต์ วงจร Timing and Control จะเอาคำสั่งประเภท 1,2 และ 3 ไบต์แรกเท่านั้นไปทำงานส่วนคำสั่งที่เหลือทิ้งไป คำสั่ง 1 ไบต์ที่ใช้เวลาการทำงาน 2 ไชเคล็ดของเครื่องที่กล่าวมาแล้วจะไม่รวมถึงคำสั่ง MOVX ซึ่งใช้ในการอ่านหรือเขียนข้อมูลกับหน่วยความจำ Data Memory ภายนอก การทำงานของคำสั่งนี้จะมีการ fetch คำสั่งเข้าไปแต่จะเป็นช่วงคำสั่งนั้นเสร็จสิ้นก่อนจึงจะเริ่มการ Reset ได้ โดย 8051 จะดูสถานะของสัญญาณที่ขา RST ของ S5P2 ในไชเคล็ดของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ของไชเคล็ดเครื่องแรกๆ ในคำสั่งอาจมีสถานะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไชเคล็ดเครื่องสุดท้ายมีสถานะลอจิกที่ขา RST เป็น 0 ก็จะไม่เกิดการ Reset ขึ้นที่เวลา S5P2 เมื่อตรวจสอบสถานะสัญญาณที่ขา RST แล้วพบว่าเป็น 1 จะต้องรอจนถึงเวลา S4P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนถึง S4P2 ของ Machine Cycle ถัดไปจะยังคงมีการ Fetch คำสั่งเข้าไปทำงานอีก 2 คำสั่งเมื่อสัญญาณ Reset ภายในเปลี่ยนเป็น 1 ก็เริ่ม Reset โดยการเขียนข้อมูล 0. ไปยัง Special Function Register ทุกตัว ยกเว้นพอร์ท 0 ถึงพอร์ท 3 Stack Pointer และ Register SBUF ระหว่างนี้ข้อมูลใน RAM ภายใน 8051 จะไม่เปลี่ยนแปลงข้อมูลในระหว่างการเขียนข้อมูลลงไปยัง SFR จะมีการ Fetch คำสั่งเข้ามาทำงานอีก คำสั่งจนกว่าจะถึง S3P1 ของไชเคล็ดที่ 2 (นับตั้งแต่ไชเคล็ดของเครื่องที่ตรวจพบลอจิก 1 ที่ขา RST) ก็จะทำให้

สภาวะลอจิกที่ขา ALE และ PSEN ค้างอยู่ที่สภาวะลอจิก 1 และจะเป็นอย่างนี้ไปจนกว่าสภาวะลอจิกที่ขา RST เป็น 0 เวลานั้นตั้งแต่พบสัญญาณลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลาที่ ALE และ PSEN ค้างอยู่ที่ 1 จะเท่ากับ 19 ไซเคลของ Oscillator เมื่อสัญญาณที่ขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 8051 จะรออีก 1- 2 ไซเคลของเครื่องสัญญาณ ALE และ PSEN จะเริ่มเปลี่ยนแปลงเพื่อเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานรับจากคำสั่งในหน่วยความจำสำหรับโปรแกรม ตำแหน่ง 0000H

2.3 มาตรฐานการเชื่อมต่อแบบ RS-232-C

การเชื่อมต่อแบบอนุกรมนั้นอาจจะกล่าวได้ว่าเป็นวิธีการติดต่อสื่อสารทั่วไปของอุปกรณ์ทางอิเล็กทรอนิกส์ต่าง ๆ ไม่ว่าจะมาจากผู้ผลิตรายใดก็ตาม แต่เนื่องจากผู้ผลิตสินค้าทางอิเล็กทรอนิกส์ต่างก็ออกแบบพอร์ตติดต่อสื่อสารอนุกรมของตนขึ้นมา ทำให้มีความแตกต่างกันในรายละเอียดของพอร์ตหรือวิธีในการรับส่งข้อมูล เพื่อให้แน่ใจว่าพอร์ตอนุกรมหนึ่ง ๆ นั้นสามารถที่จะติดต่อกับพอร์ตอนุกรมอื่น ๆ ได้โดยไม่มีปัญหาทางด้านเทคนิค องค์กรอุตสาหกรรมอิเล็กทรอนิกส์ของสหรัฐ (EIA) ก็ได้กำหนดมาตรฐานของการเชื่อมต่อและคุณสมบัติของพอร์ตติดต่อสื่อสารอนุกรมขึ้นมาในปี ค.ศ. 1969

ตัวองค์กร EIA เองนั้นนับได้ว่าเกี่ยวข้องโดยตรงกับอุตสาหกรรมอิเล็กทรอนิกส์ทั่วโลก ดังนั้นมาตรฐานดังกล่าวจึงกลายเป็นมาตรฐานสากลไปโดยปริยาย มาตรฐานอันหนึ่งที่ได้รับการรับรองโดย EIA แล้วก็คือมาตรฐานที่ 232 แบบ C หรือเป็นที่รู้จักกันในชื่อของ RS-232-C (Regulation Standard No. 232 Type C) มาตรฐานแบบนี้มีความคล้ายคลึงกับมาตรฐาน CCITT V.24 และมีคุณสมบัติทางไฟฟ้าใกล้เคียงกับมาตรฐาน CCITT V.28 และนอกจากนี้ก็มีความใกล้เคียงกับมาตรฐาน Mil-Std-188C ของกระทรวงกลาโหมสหรัฐ ๆ ด้วย

หลังจากนั้นในปี ค.ศ. 1970 EIA ได้กำหนดมาตรฐานใหม่ที่มีมาตรฐานสูงกว่ามาตรฐาน RS-232-C ออกมาใหม่ ซึ่งก็ได้แก่มาตรฐาน RS-499 , RS-422 และ RS-423 โดยที่มาตรฐานใหม่เหล่านี้จะมีคุณสมบัติทางไฟฟ้าและคุณสมบัติทางกายภาพคล้าย ๆ กัน ดังนั้นจึงถูกเรียกรวม ๆ ไปว่าเป็นกลุ่มมาตรฐาน RS-499 แต่อย่างไรก็ตามทางบริษัท IBM และบริษัทผู้ผลิตคอมพิวเตอร์ตระกูล PC (Personal Computer) อื่น ๆ ก็ไม่ได้ให้ความสนใจมาตรฐานใหม่นี้เท่าไรนักถึงแม้ว่าจะสามารถส่งข้อมูลได้เร็วและระยะทางไกลกว่ามาตรฐาน RS-232-C ก็ตาม มีเพียงบริษัท Apple Computer เท่านั้นที่ได้นำเอามาตรฐานชุด RS-449 นี้ไปใช้กับผลิตภัณฑ์ Macintosh ของตน

มาตรฐาน RS-232-C นั้นแบ่งการเชื่อมต่อเป็น 2 ลักษณะคือ การต่อกับเทอร์มินอล (DTE : Data Terminal Equipment) และการต่อกับอุปกรณ์สื่อสารข้อมูล (DCE : Data Communication

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Equipment) ซึ่งในกรณีปกติ DCE จะต้องต่อเข้ากับ DTE เสมอ ยกตัวอย่างเช่น การต่อโมเด็มเข้ากับเครื่อง PC จะเป็นอุปกรณ์ DTE และ โมเด็มก็จะเป็นอุปกรณ์ DCE

การเชื่อมต่อตามมาตรฐาน RS-232-C นั้น โดยปกติจะใช้คอนเน็กเตอร์รูปตัว D ชนิด 25 ขา กำหนดให้ปลายสายสัญญาณด้านหนึ่งเป็นคอนเน็กเตอร์ตัวผู้ใช้ต่ออยู่กับอุปกรณ์ DCE และปลายของสัญญาณอีกด้านหนึ่งจะต้องเป็นคอนเน็กเตอร์ตัวเมียใช้ต่ออยู่กับอุปกรณ์ DTE แต่ว่ากฎและมาตรฐานนั้นมักถูกเปลี่ยนแปลงอยู่ตลอดเสมอ ดังนั้นผู้ผลิตอุปกรณ์คอมพิวเตอร์อื่น ๆ ก็ได้ออกแบบมาตรฐานคอนเน็กเตอร์แบบใหม่ออกมาเพื่อใช้แทนมาตรฐาน RS-232-C เดิม อย่างเช่นบริษัท IBM นั้นก็ได้ออกแบบคอนเน็กเตอร์แบบ 9 ขา ขึ้นมาใช้แทนคอนเน็กเตอร์แบบ 25 ขา และได้กำหนดให้เป็นมาตรฐานบนเครื่อง IBM AT

ตารางที่ 2.5 รายละเอียดการต่อคอนเน็กเตอร์แบบ DB9 ตามมาตรฐาน RS-232-C

หมายเลขขาสัญญาณ	ชื่อของสายสัญญาณ
1	Carrier Detect
2	Received Data
3	Transmitted Data
4	Data Terminal Ready
5	Signal Common
6	Data Ser Ready
7	Request To Send
8	Clear To Send
9	Ring Indicator

จากตารางที่ 2.5 จะเห็นว่าสายนำสัญญาณที่ใช้มาตรฐาน RS-232-C นั้นจะมีทั้งหมด 9 เส้น ประกอบด้วย วงจรข้อมูล 2 วงจร คือ ขานำสัญญาณหมายเลข 2 และ 3 ซึ่งใช้ในการส่งผ่านข้อมูลระหว่างเครื่อง PC และ โมเด็ม ขานำสัญญาณหมายเลข 5 เป็นกราวด์ ส่วนวงจรชุดที่ 2 จะประกอบไปด้วยขานำสัญญาณหมายเลข 7 และ 8 เป็นวงจรควบคุมสัญญาณที่เรียกว่า RTS และ CTS ซึ่งจะใช้ในการควบคุมการไหลของข้อมูลระหว่าง PC และ โมเด็ม (Hardware Flow Control) ขานำสัญญาณหมายเลข 6 ใ้บอก PC ว่าในขณะที่นั้นโมเด็มได้ทำการต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้วและพร้อมที่จะส่ง ขานำสัญญาณหมายเลข 1 เป็นสัญญาณ Carrier Detect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(CD) ใช้บอก PC ว่าในขณะที่โมเด็มได้รับสัญญาณพาหะ (Carrier) อยู่หรือไม่ ขานำสัญญาณหมายเลข 4 (DTR : Data Terminal Ready) ใช้บอกโมเด็มว่าในขณะที่เทอร์มินอล (หรือ PC) พร้อมทั้งจะติดต่อกับโมเด็มหรือไม่ ส่วนขานำสัญญาณหมายเลข 9 จะใช้เป็นสัญญาณแสดงว่าในขณะที่โมเด็มได้รับสัญญาณกระดิ่งหรือไม่ ซึ่งมักจะไม่ได้ถูกนำมาใช้งานมากนัก

ในการสื่อสารแบบอนุกรมนั้น การที่จะทำให้อุปกรณ์อิเล็กทรอนิกส์สื่อสารกันได้ อย่างน้อยก็จำเป็นต้องใช้สายไฟฟ้าเชื่อมต่อกันระหว่างอุปกรณ์สองตัวไม่ต่ำกว่าสองเส้น แต่จะเห็นว่ามีมาตรฐาน RS-232-C นั้นกำหนดให้คอนเน็คเตอร์มีขานำสัญญาณถึง 25 ขาซึ่งอาจจะดูเหมือนว่าจะพุ่มเพื่อยเกินกว่าความจำเป็นจนเกินไป เหตุผลที่ต้องใช้คอนเน็คเตอร์ 25 ขานี้ก็เพราะว่าอุปกรณ์สื่อสารแบบอนุกรมตามมาตรฐานนั้น จะต้องมียังวงจรไฟฟ้าที่เกี่ยวข้องกับการรับส่งข้อมูล อยู่สามวงจรร่วมกันก็คือ วงจรรับส่งข้อมูล วงจรส่งข้อมูล และวงจรควบคุม ซึ่งมีสัญญาณที่เกี่ยวข้องกับการสื่อสารแบบอนุกรมอยู่เป็นจำนวนมาก ดังที่จะกล่าวถึงรายละเอียดแยกเป็นประเภทดังต่อไปนี้

1. สัญญาณข้อมูล

ส่วนที่นับได้ว่าเป็นส่วนที่สำคัญที่สุดของมาตรฐาน RS-232-C ก็คือวงจรสำหรับส่งผ่านข้อมูลซึ่งมีอยู่สองวงจรถือคือ วงจรที่ทำหน้าที่ส่งข้อมูลจาก DTE ไป DCE และอีกวงจรหนึ่งก็จะทำหน้าที่ตรงกันข้ามกับวงจรแรกก็คือ วงจรที่มีหน้าที่รับข้อมูลจาก DCE เพื่อส่งไปให้ DTE ซึ่งเครื่อง PC นั้นจะส่งข้อมูลไปให้โมเด็มทางด้านขานำสัญญาณหมายเลข 3 และรับข้อมูลกลับจากโมเด็มทางด้านขานำสัญญาณหมายเลข 2 โดยใช้ขานำสัญญาณหมายเลข 5 เป็นกราวด์รวมทั้งสองวงจร

สัญญาณทางไฟฟ้าที่ขานำสัญญาณหมายเลข 2, 3 และ 5

มาตรฐาน RS-232-C นั้นจะให้ความสำคัญกับวิธีการส่งผ่านข้อมูลลงบนสายนำสัญญาณเป็นอย่างมาก โดยรูปแบบสัญญาณทางไฟฟ้าที่ใช้แทนข้อมูลนั้นก็ในรูปแบบสัญญาณง่าย ๆ คือสัญญาณสี่เหลี่ยมที่ถูกสร้างจากไฟฟ้ากระแสตรงระดับแรงดันประมาณ 3 ถึง 25 โวลต์ แทนข้อมูล "0" และ -3 ถึง -25 โวลต์แทนข้อมูล "1" สำหรับช่วงระดับแรงดัน -3 ถึง 3 โวลต์ นั้นจะเป็นช่วงระดับแรงดันที่ใช้ในการแบ่งแยกระดับสถานะของสัญญาณระหว่างสถานะ "0" และสถานะ "1"

คุณสมบัติของสัญญาณแบบสี่เหลี่ยมนี้เมื่อถูกส่งออกมาในสายไฟฟ้าแบบธรรมดา มักจะเกิดความผิดเพี้ยนได้ง่าย เนื่องจากข้อจำกัดของสายนำสัญญาณหรืออาจมีสาเหตุมาจากสัญญาณรบกวนภายนอกอื่น ๆ ทำให้การรับส่งข้อมูลมีระยะช่วงใช้งานอยู่ในช่วงจำกัด โดยเฉพาะการรับ

ส่งข้อมูลที่มีความเร็วสูงนั้น คลื่นที่เหลื่อมจะเรียงชิดติดกัน จึงยิ่งทำให้มีโอกาสเกิดความผิดพลาดของการรับข้อมูลได้ง่าย ดังนั้นมาตรฐาน RS-232-C จึงได้กำหนดความยาวของสายนำสัญญาณสำหรับการใช้งานที่ความเร็วต่าง ๆ ไว้ ยกตัวอย่างเช่น ถ้าหากจะใช้การเชื่อมต่อตามมาตรฐาน RS-232-C ที่ความเร็วในการส่งข้อมูล 9600 BTS ก็จะมีระยะใช้งานได้ที่ระยะทางไม่เกินกว่า 50 ฟุต เป็นต้น แต่อย่างไรก็ตามคุณภาพของสัญญาณก็จะขึ้นอยู่กับคุณภาพของสายนำสัญญาณด้วยเช่นกัน

2. วงจรควบคุม

วงจรควบคุมของ RS-232-C นั้นมีอยู่ทั้งหมด 5 วงจร ซึ่งแต่ละวงจรจะมีหน้าที่ในการสร้างสัญญาณควบคุมต่าง ๆ ขึ้นมาเพื่อให้เครื่อง PC และ โมเด็มทราบสถานะในการทำงานของกันและกันว่าอยู่ในสถานะใด โดยที่สัญญาณในวงจรควบคุมนั้นจะมีลักษณะทางกายภาพเช่นเดียวกับสัญญาณที่ปรากฏอยู่ในวงจรข้อมูล แต่โมเด็มส่วนใหญ่ไม่ได้ใช้วงจรควบคุมครบทุกวงจรมอกจากนี้ผู้ใช้อาจสามารถที่จะกำหนดให้โมเด็ม หรือ PC ไม่ตอบสนองต่อสัญญาณควบคุมบางอย่างได้ตามต้องการ

ในการติดตั้งโมเด็มความเร็วสูงตั้งแต่ 9600 BPS ขึ้นไปนั้น การกำหนดพารามิเตอร์เกี่ยวกับสัญญาณควบคุมต่าง ๆ ให้สอดคล้องกับการทำงานนั้น จึงควรจะมี ความเข้าใจพื้นฐานเกี่ยวกับสัญญาณควบคุมต่าง ๆ เสียก่อน ซึ่งมีรายละเอียดดังต่อไปนี้

Request To Send และ Clear To Send (ขานำสัญญาณหมายเลข 7 และ 8)

วงจร RTS และ CTS นั้นเป็นวงจรที่ใช้สำหรับการควบคุมการส่งข้อมูลระหว่างเครื่อง PC และ โมเด็ม โดยโมเด็มจะส่งสัญญาณ CTS สถานะ "ON" ให้แก่ PC เมื่อโมเด็มพร้อมที่จะรับข้อมูล และส่งสัญญาณ CTS สถานะ "OOF" เมื่อโมเด็มนั้นไม่พร้อมที่จะรับข้อมูลจากเครื่อง PC ได้ ในทำนองเดียวกันเครื่อง PC ก็จะทำให้สัญญาณ RTS สถานะ "ON" เมื่อ PC พร้อมที่จะรับข้อมูล และส่งสัญญาณ RTS สถานะ "OOF" เมื่อ PC ไม่พร้อมที่จะรับข้อมูลจากโมเด็ม ซึ่งสัญญาณ RTS และ CTS นี้มีประโยชน์มากในการสื่อสารข้อมูลด้วยความเร็วสูง โดยสามารถป้องกันการล้นและสูญหายของข้อมูลได้เป็นอย่างดี

Data Terminal Ready (ขานำสัญญาณหมายเลข 4) และ Data Set Ready(ขานำสัญญาณหมายเลข 6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ DTR จะใช้เป็นที่บอกโมเด็มทราบว่าเครื่อง PC นั้นกำลังอยู่ในสถานะที่พร้อมจะติดต่อกับโมเด็มหรือไม่ และในกรณีเดียวกันสัญญาณ DSR นั้นก็จะใช้เป็นที่บอกให้เครื่อง PC ทราบว่าโมเด็มก็พร้อมที่จะติดต่อกับ PC หรือไม่ โดยที่สัญญาณ DSR นั้นจะอยู่ในสถานะ “ON” ก็ต่อเมื่อโมเด็มได้รับสัญญาณ DTR แล้ว

Carrier Detect (ขาน้ำสัญญาณหมายเลข 1)และRing Indicator(ขาน้ำสัญญาณหมายเลข 9)

สัญญาณ CD นี้จะใช้เป็นที่บอก PC ให้ทราบว่าโมเด็มนั้นกำลังเชื่อมต่อกับโมเด็มเครื่องอื่น ๆ และกำลังได้รับสัญญาณพาหะจากโมเด็มปลายทาง ส่วนสัญญาณ RI นั้นก็จะเป็นการบอกเครื่อง PC ให้ทราบว่ามียุสัญญาณกระดิ่งโทรศัพท์เรียกเข้ามาที่โมเด็ม ซึ่งโมเด็มส่วนใหญ่ในปัจจุบันก็มีวงจรที่ใช้ตอบรับโทรศัพท์โดยอัตโนมัติ(Auto-answer)อยู่ภายในตัวเองแล้ว จึงไม่จำเป็นต้องใช้สัญญาณ RI นี้แต่ในบางกรณี โปรแกรมบางประเภทก็อาจจะใช้สัญญาณ RI นี้เป็นตัวกำหนดให้เริ่มทำการรันโปรแกรมอื่น ๆ ได้เช่นกัน

2.4 การสื่อสารแบบ Synchronous และ Asynchronous

การสื่อสารของเครื่อง PC ส่วนใหญ่นั้นเป็นแบบ Asynchronous ซึ่งก็หมายความว่าไม่มีการกำหนดช่วงเวลาตายตัวที่ใช้ส่งข้อมูลหนึ่งตัวอักษรออกไปที่พอร์ตนุกรม นั่นก็หมายความว่า การส่งข้อมูลแต่ละครั้งนั้นจะต้องมีบิตเริ่มต้น(Start bit) และบิตสิ้นสุด(Stop bit) เป็นตัวกำหนดขอบเขตของข้อมูล บิตที่แสดงขอบเขตนี้จะถูกสร้างขึ้นโดยอัตโนมัติจากวงจรสื่อสาร แต่สำหรับการสื่อสารแบบ Synchronous นั้นบิตเริ่มและบิตหยุดไม่มีความจำเป็นเนื่องจากว่าได้กำหนดช่วงเวลาในการส่งข้อมูลต่อหนึ่งตัวอักษรไว้แล้ว

เมื่อ PC ได้ทำการเชื่อมต่อเข้าเครื่องคอมพิวเตอร์โฮสต์อื่น ๆ หรือศูนย์บริการข้อมูลอื่น ๆ โมเด็มจะใช้การติดต่อแบบ Asynchronous เสมอ ส่วนการสื่อสารแบบ Synchronous นั้นจะใช้เฉพาะในงานพิเศษ ๆ ที่ต้องการความเร็วในการส่งข้อมูลสูงมาก เช่นการส่งข้อมูลระหว่าง LAN ซึ่งวงจรสื่อสารที่พบในเครื่อง PC ทั่ว ๆ ไปนั้นไม่สามารถทำงานในแบบ Synchronous ได้

การสื่อสารแบบ Synchronous นั้นมีประสิทธิภาพการทำงานสูงกว่าแบบ Asynchronous มาก แต่จะต้องการวงจรพิเศษสำหรับ การสื่อสารแบบ Synchronous โดยข้อมูลที่ส่งแบบ

Synchronous นั้นจะไม่มี การขาดคอน โดยข้อมูลจะถูกแยกเป็นชุด ๆ จากสัญญาณนาฬิกาที่ส่งไปพร้อม ๆ กัน ในสายนำสัญญาณอีกเส้นหนึ่ง ดังนั้นจึงมีความเร็วในการส่งข้อมูลที่สูงมาก

หน้าที่ที่สำคัญของสื่อสารแบบ Asynchronous คือ

ส่งสัญญาณ

1. เปลี่ยนสัญญาณแบบขนานจาก PC ค่อยทยอยส่งออกเป็นแบบอนุกรม
2. เพิ่มบิตเริ่มและบิตสิ้นสุด
3. เพิ่มสัญญาณควบคุม โมเด็มที่ต่อเชื่อม

รับสัญญาณ

1. เปลี่ยนสัญญาณเข้ามาอนุกรมให้เป็นแบบขนาน
2. ตรวจสอบความผิดพลาดของสัญญาณที่รับ
3. คัดบิตเริ่มและบิตสิ้นสุดออก
4. ส่งสัญญาณให้ว่า PC รู้ว่ารับสัญญาณไว้แล้ว

2.5 พอร์ตสื่อสารข้อมูลอนุกรม UART

จากหัวข้อที่ผ่านมาได้อธิบายถึงการจัด โครงสร้างตลอดจนคุณสมบัติของชิปไมโครคอนโทรลเลอร์ MCS-51 จะพบว่าไมโครคอนโทรลเลอร์ตระกูล 8051 นี้จะมีพอร์ตสื่อสารข้อมูลอนุกรม UART (Universal Asynchronous Receiver Transmitter) อยู่ภายในชิพของ MCU จึงทำให้การนำ MCU ไปประยุกต์ใช้งานในการติดต่อสื่อสารข้อมูลหรือสามารถใช้งานควบคุมการทำงานของอุปกรณ์ภายนอกได้สะดวกและมีประสิทธิภาพมากยิ่งขึ้น

2.5.1 พอร์ตสื่อสารข้อมูลอนุกรม UART ภายในชิพ MCS-51

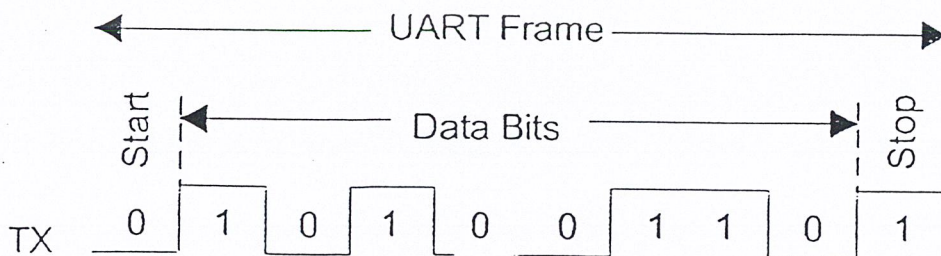
โดยในการใช้งานพอร์ตสื่อสารข้อมูลอนุกรมของ ไมโครคอนโทรลเลอร์ MCS-51 นั้น MCS-51 สามารถที่จะให้มีการเซตหรือกำหนดอัตราในการติดต่อหรือกำหนดอัตราเร็วในการติดต่อหรือรับส่งข้อมูล (Baud rate) ได้ เพื่อยืดหยุ่นในการใช้งานในการติดต่อสื่อสารข้อมูลได้ครอบคลุมหลายรูปแบบ ทั้งการติดต่อระหว่าง MCU ด้วยกันเองหรือจะติดต่อสื่อสารผ่านพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ PC กับ MCU ก็สามารถกระทำได้ โดยรูปแบบในการใช้งานพอร์ตสื่อสารข้อมูลอนุกรมของ MCS-51 นั้นสามารถที่จะจัดรูปแบบการใช้งานได้ถึง 4 รูปแบบ โดยแต่ละรูปแบบดังกล่าวจะสามารถควบคุมเลือกการใช้งานได้โดยรีจิสเตอร์ใช้งานเฉพาะ SCON ภายในชิป

ซึ่งรูปแบบในการใช้งานพอร์ตสื่อสารข้อมูลอนุกรม UART ภายในชิพจะมีลักษณะเป็นโหมด ๆ ดังนี้

โหมด 0 การทำงานของพอร์ตอนุกรม UART ในโหมด 0 นี้จะเป็นการใช้งานขา P3.0 (RXD) ของตัว MCU สำหรับการรับส่งข้อมูลและใช้ขา P3.1 (TXD) ในการผลิตสัญญาณนาฬิกา (Clock) เพื่อให้อุปกรณ์ภายนอกรับรู้จังหวะการรับส่งข้อมูลที่ขา RXD นั่นคือสัญญาณนาฬิกาที่ขา TXD จะเป็นตัวกำหนดการเลื่อน (Shift) ของข้อมูลที่ขา RXD นั่นเอง (ในการรับส่งข้อมูลที่ขา RXD จะใช้สัญญาณนาฬิกาที่ขา TXD เป็นตัวจังหวะรับส่งข้อมูล) โดยข้อมูลที่ขา RXD จะเป็นลักษณะของข้อมูล 8 บิต ซึ่งจะมีการส่งหรือรับข้อมูลที่บิตต่ำสุด (Least Significant Bit ; LSB) ของชุดข้อมูลก่อน แล้วจึงเรียงลำดับข้อมูลจนถึงบิตสูงสุด (Most Significant Bit ; MSB) จึงจะเป็นการสิ้นสุดของข้อมูล ในโหมด 0 นี้จะไม่สามารถกำหนดอัตราความเร็วในการรับส่งข้อมูล (Baud Rate) ได้ ซึ่งอัตราความเร็วในการรับส่งข้อมูลที่โหมด 0 นี้ จะกำหนดไว้ที่ 1/12 ของความถี่ X-TAL Oscillator ที่ใช้ โดยสามารถคำนวณได้ดังสมการ

$$\text{Baud Rate โหมด 0} = X\text{-TAL}/12$$

โหมด 1 การทำงานของพอร์ตอนุกรม UART ในโหมดที่ 1 นี้จะมีลักษณะของชุดข้อมูลจำนวน 10 บิต ดังแสดงในรูปที่ 2.12 โดยจะมีบิตเริ่มต้นของชุดข้อมูล (Start Bit มีค่าเป็น 0 เสมอ) ตามมาด้วยข้อมูลจำนวน 8 บิต และบิตสุดท้ายของข้อมูล (Stop Bit มีค่าเป็น 1 เสมอ) ซึ่งชุดข้อมูลจะถูกส่งออกที่ขา TXD ของตัว MCU และรับชุดข้อมูลเข้ามาภายใน MCU โดยขา RTX อัตราเร็วในการรับส่งข้อมูล (Baud Rate) ของการทำงานในโหมด 1 นี้ สามารถกำหนดหรือเปลี่ยนแปลงได้ โดยใช้ Timer ภายในตัว MCU เป็นตัวควบคุม ซึ่งรายละเอียดในการใช้งาน Timer ในการควบคุมอัตราเร็วในการรับส่งข้อมูลจะกล่าวในหัวข้อการประยุกต์ใช้งานพอร์ตสื่อสารข้อมูลอนุกรมต่อไป

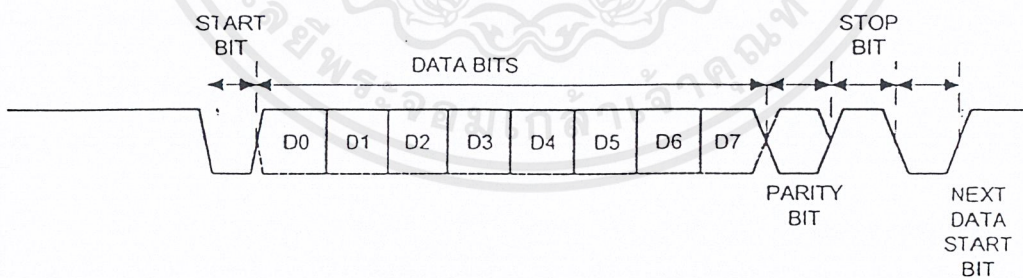


รูปที่ 2.18 แสดงชุดข้อมูล UART ในโหมดที่ 1

โหมด 2 ในโหมด 2 นี้จะมีชุดของข้อมูลที่ใช้ในการรับส่งแตกต่างจากข้อมูลในโหมดที่ 1 คือ จะมีชุดข้อมูลจำนวน 11 บิต ดังแสดงในรูปที่ 2.19 ซึ่งจะมีวิธีการในการรับและส่งข้อมูลผ่านทางขาของ MCU เช่นเดียวกับในโหมดที่ 1 โดยชุดข้อมูลในโหมดที่ 2 นั้นจะประกอบไปด้วยบิตแรก (Start Bit มีค่าเป็น 0 เสมอ) และตามมาด้วยข้อมูลจำนวน 8 บิต จากนั้นจะมีบิตที่ 9 เพิ่มขึ้น โดยจะเป็นบิตที่สามารถจะเซตค่าให้เป็น 0 หรือ 1 ได้ ซึ่งใช้ในการตรวจสอบความผิดพลาดของ ข้อมูลในการรับและส่งนั่นเอง (Parity Bit) จากนั้นจะเป็นบิตสุดท้ายของชุดข้อมูล (Stop Bit มีค่าเป็น 1 เสมอ) โดยโหมด 2 นี้จะไม่สามารถกำหนดอัตราเร็วในการรับส่งข้อมูล (Baud Rate) ได้เช่นเดียวกับในโหมด 0 ซึ่งอัตราเร็วในการรับส่งข้อมูลในโหมด 2 นี้จะถูกกำหนดไว้ที่ 1/32 หรือ 1/64 ของความถี่ X-TAL Oscillator ที่ใช้ โดยจะมีบิต SMOD ภายในรีจิสเตอร์ใช้งานเฉพาะ PCON เป็นตัวกำหนดดังนี้

- ถ้าค่าบิต SMOD เป็น 0 ค่า Baud Rate จะถูกกำหนดเป็น 1/64 ของความถี่ X-TAL Oscillator ที่ใช้
 - ถ้าค่าบิต SMOD เป็น 1 ค่า Baud Rate จะถูกกำหนดเป็น 1/32 ของความถี่ X-TAL Oscillator ที่ใช้
- ซึ่งมีสมการในการคำนวณดังนี้

$$\text{Baud Rate Mode 2} = \text{X-TAL}/32\text{or}64, \text{SMOD} = 1 \text{ OR } 0$$



รูปที่ 2.19 แสดงชุดข้อมูล UART ในโหมดที่ 2

โหมด 3 จะเป็นโหมดในการใช้งานโหมดสุดท้ายของการใช้งานพอร์ตสื่อสารข้อมูลอนุกรม UAPT ภายในชิป MSC-51 ในโหมด 3 นี้ จะมีชุดข้อมูลในการรับและส่งจำนวน 11 บิต ซึ่งจะมีการรับและส่งภาพทางขาของ MCU เช่นเดียวกับในโหมดที่ 1 โดยองค์ประกอบของชุดข้อมูลจะ

เหมือนกับชุดข้อมูลในโหมดที่ 1 ทุกประการ แต่โหมดที่ 3 นี้สามารถกำหนดอัตราความเร็วในการรับส่งข้อมูลได้เช่นเดียวกันกับในโหมดที่ 1 โดยใช้ Timer ภายใน MCU เป็นตัวควบคุม

โดยในการใช้งานโหมดการทำงานข้างต้นดังที่กล่าวมาจะนิยมใช้พอร์ตสื่อสารข้อมูลอนุกรม UART ในโหมดที่ 1 ในการใช้งานเนื่องจากในโหมดที่ 1 นั้นจะมีลักษณะของชุดข้อมูลที่เป็นมาตรฐานทั่วไป และสามารถเช็คหรือกำหนดอัตราเร็วในการสื่อสารข้อมูลได้จากคุณสมบัติหรือเหตุผลดังกล่าวเนื้อหาในการประยุกต์ใช้งานพอร์ตสื่อสารข้อมูลที่จะกล่าวถึงต่อไป จะอธิบายถึงการใช้งานพอร์ตสื่อสารอนุกรมในโหมดที่ 1 เพื่อเป็นพื้นฐานในการประยุกต์ใช้งานพอร์ตสื่อสารข้อมูลอนุกรมของไมโครคอนโทรลเลอร์ MSC-51 ในเบื้องต้น

ก่อนที่เข้าถึงการประยุกต์ใช้งานพอร์ตสื่อสารข้อมูลอนุกรมในโหมดที่ 1 ต่อไปนั้น จะขอแนะนำรีจิสเตอร์ใช้งานเฉพาะต่างๆ เพื่อการใช้งานพอร์ตสื่อสารข้อมูลอนุกรมในโหมดที่ 1 และ ไทม์เมอร์ (Timer) เพื่อกำหนดอัตราเร็วในการรับส่งข้อมูลอนุกรม (Baud Rate) ดังนี้

รีจิสเตอร์ใช้งานเฉพาะ SBUF: Serial Data Buffer (Not Bit Addressable)

เป็นรีจิสเตอร์ที่ใช้สำหรับการรับส่งสื่อสารชุดข้อมูลอนุกรม โดยจะทำหน้าที่เป็นบัฟเฟอร์ (Buffer) ขนาด 8 บิตไม่สามารถเข้าถึงในระดับบิต ซึ่งรีจิสเตอร์ SBUF นี้จะมีหน้าที่ ทั้งในการรับและส่งชุดข้อมูลอนุกรมโดยตัว CPU จะทำการควบคุมการใช้งานรีจิสเตอร์ SBUF ทั้งในการรับและส่งข้อมูลโดยอัตโนมัติ

รีจิสเตอร์ใช้งานเฉพาะ SCON :Serial Port Control Register (Bit Addressable)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

ภายในรีจิสเตอร์ต่างๆ เกี่ยวกับไมโครคอนโทรลเลอร์ MSC-51 นั้นผู้อ่านคงจะพอรู้แล้วว่าภายใน 1 รีจิสเตอร์ที่สามารถเข้าถึงข้อมูลได้ในระดับบิตนั้น นั่นก็คือ 1 ไบต์ซึ่งจะประกอบไปด้วยบิตต่างๆ จำนวน 8 บิต รีจิสเตอร์ใช้งานเฉพาะ SCON นั้นก็เช่นกัน คือจะมีบิตต่างๆ จำนวน 8 บิต ที่สามารถเข้าถึงได้ระดับบิตที่ใช้ควบคุมหรือกำหนดค่าต่างๆในการใช้งานพอร์ตสื่อสารข้อมูลอนุกรม โดยบิตที่เริ่มจำเป็นจะต้องรู้จักเพื่อการใช้งานพอร์ตสื่อสารข้อมูลอนุกรมในโหมดที่ 1 ภายในรีจิสเตอร์ SCONจะแสดงไว้ในตารางที่ 2.5

โดยจากตารางที่ 2.5 จะพบว่าบิตTI และRI จะสามารถทำการเช็คหรือเคลียร์ได้ทั้งทางซอฟต์แวร์หรือตัวของ CPU ภายใน MCS-51 เอง จากนั้นจะเห็นว่าบิต SM0และSM1 นั้นจะเป็นบิต

ควบคุมในการเลือกโหมดการใช้งานพอร์ตสื่อสารข้อมูลอนุกรม โดยรายละเอียดการใช้งานหรือตารางความจริงของบิต SM0 และ SM1 แสดงดังตารางที่ 2.6

ชื่อบิต	รายละเอียด
SM0	บิตเลือกโหมดการใช้งานพอร์ตสื่อสารข้อมูลอนุกรม “ต่ำ”
SM1	บิตเลือกโหมดการใช้งานพอร์ตสื่อสารข้อมูลอนุกรม “สูง”
SM2	บิตเลือกโหมดการใช้งานพอร์ตสื่อสารข้อมูลอนุกรม โดย $SM2 = 0$ จะเป็นการใช้งานโหมด 1 และ 3 ปกติและบิต R1 จะเป็น 0 จนกว่าจะรับบิตสุดท้ายของชุดข้อมูลเป็นที่เรียบร้อยแล้ว
REN	REN=1เป็นการเซตให้ตัว CPU รับข้อมูลจากภายนอกได้ REN=0เป็นการเซตให้ตัว CPU ไม่รับข้อมูลจากภายนอก
TI	บิตบอกสถานะในการส่งข้อมูลอนุกรมทำงาน โดยอัตโนมัติ TI=0 จะเห็นว่าเป็นช่วงจังหวะในขั้นตอนในการส่งชุดข้อมูลออกจากพอร์ตสื่อสารข้อมูลอนุกรม TI=1 จะแสดงว่าชุดข้อมูลถูกส่งออกไปเป็นที่เรียบร้อยแล้วพร้อมที่จะส่งข้อมูลชุดต่อไป โดยก่อนการที่จะส่งข้อมูลชุดต่อไปนั้นต้องทำการเคลียร์บิต TI ใหม่ก่อนทุกครั้ง
RI	บิตบอกสถานะในการรับข้อมูลและถ้าข้อมูลถูกรับเรียบร้อยแล้ว บิต RB8 ของรีจิสเตอร์ SCONจะมีสถานะ 1 พร้อมๆกับชุดข้อมูล 8 บิตจะถูกเก็บไว้ในรีจิสเตอร์ SBUFและบิต RIจะกลับเป็น 1 โดยอัตโนมัติ จากนั้นต้องเคลียร์บิต RI เพื่อที่จะรับข้อมูลชุดใหม่ต่อไป RI = 1 เป็นการแสดงว่าให้รับข้อมูลที่ส่งมาไปเก็บไว้ในรีจิสเตอร์ SBUF โดยก่อนหน้านี้นี้ต้องเคลียร์บิต RI ก่อน

ตารางที่ 2.6 แสดงบิตภายในรีจิสเตอร์ SCON เพื่อการใช้งานพอร์ตสื่อสารข้อมูลอนุกรมโหมดที่ 1

SM0	SM1	Mode	Description	Baud Rate	
0	0	0	SHIFT REGISTER	Fosc./12	
0	1	1	8-Bit UART	Variable	1
0	2		9-Bit UART	Fosc./64OR Fosc./32	1
1	3		9-Bit UART	Variable	

ตารางที่ 2.7 แสดงบิตในการควบคุมโหมดการใช้งานด้วยบิต SM0 และ SM1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ไทเมอร์ในการกำหนดอัตราเร็วในการรับส่งข้อมูล

ไทเมอร์ภายในตัวไมโครคอนโทรลเลอร์ MSC-51 ทุกเบอร์จะประกอบไปด้วยไทเมอร์ 0 และ ไทเมอร์ 1 และอาจจะมีไทเมอร์ 2 ในบางเบอร์ของ MSC-51 โดยการกำหนดอัตราเร็วในการรับส่งข้อมูลอนุกรมในหมวดที่ 1 นั้นจะใช้ไทเมอร์ 1 เป็นตัวกำหนด Baud Rate ซึ่งการใช้งานไทเมอร์นั้นจะถูกควบคุมด้วยรีจิสเตอร์ใช้เฉพาะ TMOD โดยมีรายละเอียดดังต่อไปนี้

รีจิสเตอร์ใช้งานเฉพาะ TMOD: Timer/counter Mode Control Register (NotBit Addressable)

Timer 1				Timer 0			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

ภายในรีจิสเตอร์ TMOD ประกอบไปด้วยการควบคุมการใช้งานต่างๆ จำนวน 8 บิตจากข้างต้นจะพบว่าจะมีบิตในการใช้งานไทเมอร์ 1 และ ไทเมอร์ 0 โดยบิต 4 บนจะควบคุมการใช้งานไทเมอร์ 1 และ 4 บิตล่างจะเป็นชุดบิตที่ควบคุมไทเมอร์ 0 ซึ่งทั้ง 4 บิตบนและล่างจะมีลักษณะการใช้งานเช่นเดียวกันรายละเอียดของชุดบิตดังที่กล่าวมาเพื่อการใช้งานไทเมอร์ในการกำหนดอัตราเร็วในการรับส่งข้อมูลผ่านพอร์ตอนุกรมดังแสดงในตารางที่ 2.8

ชื่อบิต	รายละเอียด
GATE	เท่ากับ 0 จะสามารถเลือกใช้งานไทเมอร์หรือเคาน์เตอร์ได้
C/T	บิตเลือกการใช้งานไทเมอร์หรือเคาน์เตอร์ได้ เท่ากับ 1 หมายถึงใช้งานเป็นเคาน์เตอร์นับพัลส์ภายนอกที่ขา TX (T0,T1) เท่ากับ 0 หมายถึงใช้งานเป็นไทเมอร์นับการทำงานของแมชชีนไซเคิล
M0	บิตเลือกการใช้งานไทเมอร์ 0 หรือ ไทเมอร์ 1
M1	บิตเลือกการใช้งานไทเมอร์ 0 หรือ ไทเมอร์ 1

ตารางที่ 2.8 แสดงบิตในการควบคุมการใช้งานไทเมอร์ 1

การควบคุมบิต TRX (TR0,TR1) ทางซอฟต์แวร์นั้น บิต TRX จะอยู่ภายในรีจิสเตอร์ TCON (รีจิสเตอร์ TCON) ซึ่งสามารถเข้าถึงได้ในระดับบิตโดยถ้า TR1 = 1 จะเป็นการใช้งานไทเมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมอร์ 1 และถ้า $TR0 = 1$ จะเป็นการใช้งานไทม์เมอร์ 0 ซึ่งนั่นก็คือบิต $TR1$ และ $TR0$ ภายในรีจิสเตอร์ใช้งานเฉพาะ $TCON$ จะเป็นตัวควบคุมการใช้งานไทม์เมอร์ 1 และ ไทม์เมอร์ 0 ตามลำดับ

โดยจากตาราง 2.8 จะเห็นว่าบิต $M0$ และ $M1$ นั้นจะใช้เป็นการใช้เป็นบิตในการควบคุมหรือเลือกโหมดการทำงานของไทม์เมอร์ ซึ่งสามารถสรุปตารางความจริงของบิต $M0$ และ $M1$ ได้ดังแสดงในตารางที่ 2.8

การใช้งานไทม์เมอร์ 1 เพื่อการกำหนดอัตราเร็วในการสื่อสารข้อมูลอนุกรมนั้นจะใช้งานไทม์เมอร์ 1 ในโหมดที่ 2 โดยจากตารางที่ 2.8 นั่นก็คือบิต $M1$ ถูกเซตให้เป็น 1 และ บิต $M0$ ให้เป็น 0 ในส่วนของรายละเอียดของการทำงานภายในไทม์เมอร์ 1 ในโหมดที่ 2 หรือในโหมดอื่นๆ นั้นจะไม่ขอแสดงรายละเอียดเนื่องจากเป็นขั้นตอนการทำงานภายในโดยตัว CPU เป็นตัวควบคุมการทำงานทั้งหมด

จากที่กล่าวมาแล้วการใช้งานไทม์เมอร์ 1 ในโหมดที่ 2 เพื่อการกำหนดอัตราเร็วในการสื่อสารข้อมูลอนุกรมนั้น จะมีบิต $SMOD$ ภายในรีจิสเตอร์ใช้งานเฉพาะ $PCON$ เป็นตัวกำหนดอัตราการเกิดโอเวอร์โฟลว์ (Over Flow) ให้กับไทม์เมอร์ 1 โดยลักษณะชุดข้อมูลของรีจิสเตอร์ $PCON$ มีดังนี้

รีจิสเตอร์ $PCON$: Power Control Register (Not Bit Addressable)

SMOD	-	-	-	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

ข้างต้นเป็นลักษณะรูปแบบชุดข้อมูลของรีจิสเตอร์ $PCON$ จะเห็นว่าบิต $SMOD$ จะอยู่ในตำแหน่งบิตสูงสุด (Most Significant Bit ; MSB) ของชุดข้อมูล 8 บิตและรีจิสเตอร์ $PCON$ นี้ จะไม่สามารถเข้าถึงข้อมูลในระดับบิตได้ ฉะนั้นการเซตค่า $SMOD$ ในการควบคุมการเกิดโอเวอร์โฟลว์ให้ไทม์เมอร์ 1 นั้นจะสามารถกระทำได้โดยการส่งชุดข้อมูลครั้งละ 8 บิตเข้าไปในรีจิสเตอร์ $PCON$ ดังตัวอย่าง

`MOV PCON, # 80H ; บิต SMOD เท่ากับ 1`

`MOV PCON, # 00H ; บิต SMOD เท่ากับ 0`

รายละเอียดการใช้งานบิต $SMOD$ นั้นจะเป็นบิตในการควบคุมอัตราความเร็วในการรับส่งข้อมูล (Baud Rate) โดยถ้าหากถูกเซตหรือบิต $SMOD$ มีค่าเป็น 1 จะทำให้มีการเพิ่มอัตราความเร็วในการรับส่งข้อมูลในการรับส่งข้อมูลเป็น 2 เท่านั่นเอง

จากที่กล่าวมาแล้วว่ารูปแบบในการใช้งานพอร์ตสื่อสารข้อมูลอนุกรมของ MCS-51 นั้นสามารถที่จะจัดรูปแบบการใช้งานได้ถึง 4 โหมดโดยจะมีเพียงโหมดที่ 1 และ 3 เท่านั้นที่สามารถ

กำหนดอัตราความเร็วในการรับส่งข้อมูลได้ ซึ่งสามารถกำหนดอัตราความเร็วในการรับส่งข้อมูลได้ ซึ่งสามารถสรุปได้ดังสมการที่ 1

$$\text{Baud Rate} = \frac{K \times \text{Oscillator Frequency}}{32 \times 12 \times [256 - (\text{TH1})]} \quad (\text{สมการที่ 1})$$

จากสมการที่ 1 ค่า K จะมีค่าเท่ากับ 2^{SMOD} โดยค่าอัตราความเร็วในการรับส่งข้อมูลนั้นเราจะทำการกำหนดขึ้นมาได้ตามความต้องการแต่โดยมาตรฐานทั่วไปจะใช้อัตราความเร็ว 4,800 หรือ 9,600 บิตต่อวินาที (bps) ฉะนั้นตัวแปรที่เราต้องการหาจากสมการที่ 1 ก็คือค่าของ TH1 ซึ่งสามารถเปลี่ยนสมการที่ 1 ได้ใหม่เพื่อการคำนวณค่าของ TH1 ให้ตามค่า Baud Rate ตามที่เราได้ใหม่ดังสมการที่ 2

$$\text{Baud Rate} = \frac{K \times \text{Oscillator Frequency}}{328 \times \text{Baud Rate}} \quad (\text{สมการที่ 2})$$

จากสมการที่ 2 สามารถสรุป และแสดงตัวอย่างค่าต่างๆ ของ TH1 ที่สามารถกำหนดอัตราความเร็วในการรับส่งข้อมูลมาตรฐาน (Baud Rate) ดังแสดงในตารางที่ 2.9

Baud Rate	X-tal Frequency	SMOD	TH1 Reload Value	Actual Baud Rate	Error
9600	12.000 MHz	1	-7(F9H)	8923	7 %
2400	12.000 MHz	0	-13(F3H)	2404	0.16 %
1200	12.000 MHz	0	-26(E6H)	1202	0.16 %
9300	12.000 MHz	1	-3(FDH)	19200	0
9600	11.059 MHz	0	-3(FDH)	9600	0
2400	11.059 MHz	0	-12(F4H)	2400	0
1200	11.059 MHz	0	-24(E8H)	1200	0

ตารางที่ 2.9 แสดงค่าพารามิเตอร์ต่างๆ ในการทำงานพอร์ตสื่อสารอนุกรมโมโหมคที่ 1

จากตารางที่ 2.9 จะเห็นว่าในการใช้งาน X-TAL Oscillator ที่ความถี่ 11.059 MHz นั้นจะสามารถกำหนดอัตราความเร็วในการรับส่งข้อมูล (Baud Rate) ได้ค่า Actual Baud Rate ที่ลงตัวในการใช้งานพอร์ตอนุกรมสื่อสารข้อมูลอนุกรมในโมโหมคที่ 1 โดยการใช้งานไทม์เมอร์ 1 ในโมโหมคที่ 2 เป็นตัวควบคุมหรือกำหนดอัตราความเร็ว (Baud Rate) ที่ไม่มีความผิดพลาด (Error) เกิดขึ้นเลยหรือเป็นไปตามทฤษฎีนั่นเอง โดยจะเห็นได้ว่าเมื่อต้องการกำหนดอัตราความเร็วในการรับส่งข้อมูล

มูลที่อัตราความเร็ว 9,600 บิตต่อวินาที (bps) นั้นต้องทำการเคลียร์บิต SMOD (ถูกเซ็ตเป็น 0) และทำการเซ็ตค่าของรีจิสเตอร์ใช้งานเฉพาะ TH1 ให้มีค่าเป็น FDH (253 = FDH จากการคำนวณในสมการที่ 2) เพื่อเริ่มต้นการใช้งานไทม์เมอร์ 1 ก็จะได้อัตราความเร็วในการรับส่งข้อมูล 9,600 บิตต่อวินาที (bps) ตามต้องการ

ซึ่งอัตราความเร็วในการรับส่งข้อมูลที่กล่าวมาข้างต้น จะใช้กับข้อมูลที่มีขนาด 10 บิต (ประกอบด้วย Start Bit ,ข้อมูล 8 บิตและ Stop Bit) ซึ่งเป็นชุดข้อมูลมาตรฐานในการคำนวณ Baud Rate ดังที่กล่าวมา

โดยต่อไปจะขอยกตัวอย่างการเขียนโปรแกรมในการใช้งานพอร์ตสื่อสารอนุกรมในโหมดที่ 1 โดยใช้งานไทม์เมอร์ 1 ในโหมดที่ 2 กำหนดค่า Baud Rate เท่ากับ 9,600 บิตต่อวินาที (bps) ดังต่อไปนี้

ตัวอย่างโปรแกรมที่ 2-1 การใช้งานพอร์ตสื่อสารข้อมูลอนุกรมโหมดที่ 1

```
MOV SCON,#XXH ; XX=40 For TXD Serial Port Mode 1 , REN = 0,SM2=0
```

```
      ; XX=50 For RXD Serial Port Mode 1,REN = 1,SM2 =0
```

```
MOV PCON,#00H ; SMOD = 0 กรณีไม่มีการเซ็ตค่าบิต SMOD ก่อนหน้านี้ก็ไม่จำเป็นต้องใช้คำสั่งนี้เพราะปกติ SMOD จะมีค่าเป็น 0 อยู่แล้ว(หลังจากรีเซ็ต CPU เพื่อเริ่มต้นการใช้งาน)
```

```
MOV TMOD,#20H ; เซ็ตเลือกการใช้งาน ไทม์เมอร์ 1 โหมดที่ 2 (0010 0000B)
```

```
MOV TH1,#0FDH ; Baud Rate 9,600 bit/s ,X-TAL 11.069 MHz
```

```
SETB TR1 ; เริ่มต้นการใช้งานไทม์เมอร์ 1 (หรือใช้คำสั่ง MOV TCON, #40H)
```

จากนั้นเมื่อทำการกำหนดค่าต่างๆเพื่อใช้งานพอร์ตสื่อสารข้อมูลอนุกรมตามตัวอย่างที่ 2-1 เป็นที่เรียบร้อยแล้ว เราลองมาดูตัวอย่างการส่งข้อมูลออกไปทางพอร์ตอนุกรมกันบ้างโดยจะยกตัวอย่างโปรแกรมทั้งส่งและรับข้อมูลที่ละไบต์ เพื่อเป็นพื้นฐานในการใช้งานต่อไป

ตัวอย่างโปรแกรมที่ 2-2 การส่งข้อมูลที่ละไบต์

```
MOV A,#0FH ; เก็บค่าคงที่ #0FH ไว้ในAccumulator (1 ไบต์)
```

```
CLR TI ; เคลียร์บิต TI เพื่อทำขั้นตอนในการส่งข้อมูล
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV SBUF,A ; ส่งข้อมูลภายใน Accumulator ไปยัง Serial Port (TX)
 LOOP: JNB TI,LOOP ; รอจนกระทั่งส่งข้อมูลเสร็จค่า TI จะเป็น 1 โดยอัตโนมัติหรือใช้คำสั่ง
 JNB TI,\$ ก็จะมีผลเช่นเดียวกัน โดยจะลดจำนวนไบต์ในการเขียนโปรแกรมลงแล้ว จึงจะทำคำสั่งต่อไป (ครบ 1 ไบต์)

ตัวอย่างโปรแกรมที่ 2-3 การรับข้อมูลที่ละไบต์

CLR RI ; เคลียร์บิต TI เพื่อทำขั้นตอนในการรับข้อมูล
 JNB RI,NEXT ; รอจนกระทั่งรับข้อมูลเสร็จค่า RI จะเป็น 1 โดยอัตโนมัติและจะทำการ
 เคลียร์บิต RI โดยอัตโนมัติ หรือจะใช้คำสั่ง JNB RI,\$ ก็ได้เช่นกัน แต่จะไม่ทำการเคลียร์บิต RI
 ให้อัตโนมัติ ซึ่งจะต้องใช้คำสั่ง CLR RI อีกครั้งเพื่อจะรับชุดข้อมูลต่อไป
 NEXT;
 MOV A,SBUF ; นำข้อมูลภายในรีจิสเตอร์ SBUF เก็บไว้ใน Accumulator

ตัวอย่างโปรแกรมที่ 2-4 Program Transmit Data 1 Byte

```

ORG      0000H
MOV      SCON,#40H      ; Set Serial - port Mode 1 ,REN=0 , SM2=0
MOV      PCON,#00H     ; Set Bit SMOD=0
MOV      TMOD,#20H     ; Set Timer 1 Mode 2 (0010 0000B)
MOV      TH1,#0FDH     ; Baud Rate 9,600 Bit/s ,X-TAL 11.069 MHz
SETB     TRI           ; Start Timer 1
MOV      A,#0FH        ; A=0FH
CLR      TI            ; Ready To Transmit
MOV      SBUF,A        ; Transmit Data in A(HEX) out The TXD Line
LOOP:    JNB          TI,LOOP ; Wait until Transmit completed (Data 1 Byte)
END
  
```

ตัวอย่างโปรแกรมที่ 2-5 Program Receive Data 1 Byte (Not used Serial interrupt)

```

ORG      0000H

MOV      SCON,#50H      ; Used Serial Port Mode 1, REN=1 ,SM2
MOV      PCON,#00H     ; Set Bit SMOD=0
MOV      TMOD,#20H     ; Set Timer 1 Mode 2 (0010 0000B)
MOV      TH1,#0FDH     ; Baud Rate 9,600 Bit/s ,X-TAL 11.069 MHz
CLR      TI             ; Ready To Transmit
JNC      RI,NEXT        ; Receive Data from Serial Port save in SBUF
NEXT: MOV      A,SBUF   ; Loop for send Data 1 Byte
END

```

ตัวอย่างโปรแกรมที่ 2-6 Program Receive Data 1 Byte (used serial interrupt)

```

ORG      0000H
JMP      MAIN
ORG      0023H      ; Starting Address of Serial Interrupt
JMP      SERIAL_INT
MAIN: MOV      SCON,#50H      ; Set Serial Port Mode 1 ,REN=1 SM2=0
                                with Recieve
MOV      PCON,#00H      ; Set Bit SMOD=0
MOV      TMOD,#20H     ; Set Timer 1 Mode 2 (0010 0000B)
SETB    EA             ; Enable The Serial Port&Global Interrupt Bits
CLR      RI             ;Ready to Receive
LOOP: JMP      LOOP      ;Endless Loop (Unless Interrupt occurs)

SERIAL_INT:                ; Serial Interrupt Routine
CLR      RI
MOV      A,SBUF           ; Move the Receive Data to Accumulator
RETI                    ; Exit The Serial Interrupt Routine
END

```

จากตัวอย่างโปรแกรมที่ 2-5 เป็นโปรแกรมในการรับข้อมูล 1 ไบต์โดยใช้วิธีการตรวจเช็ค บิต R1 ถ้าบิต R1 มีสถานะเป็น 1 นั้นแสดงว่าได้รับข้อมูลครบ 1 ไบต์ และเก็บไว้ในรีจิสเตอร์ SBUF เป็นที่เรียบร้อยแล้วนั่นเอง แต่ในบางครั้งในการใช้งานไมโครคอนโทรลเลอร์นั้นอาจจะใช้ การ อินเทอร์รัพท์ เข้ามามีส่วนช่วยในการรับข้อมูลในระหว่างที่ CPU กำลังประมวลผลการทำงาน อื่น ๆ อยู่ก็จะช่วยให้การรับข้อมูลจากพอร์ตสื่อสารข้อมูลอนุกรมมีความแน่นอนและมีประสิทธิภาพ มากยิ่งขึ้น

โดยการใช้งานอินเทอร์รัพท์การรับข้อมูลจากพอร์ตสื่อสารข้อมูลอนุกรมนั้นเราจะใช้งาน Serial Interrupt ในการควบคุมการรับข้อมูลโดยมีบิต PS ภายในรีจิสเตอร์ IP ในการเช็คระดับ ความสำคัญการใช้งาน Serial Interrupt การใช้งานอินเทอร์รัพท์ที่เกิดจากสัญญาณภายนอกนั้นจะมี บิต EA ภายในรีจิสเตอร์ IE เป็นตัวควบคุมการใช้งาน

การทำงานของ Serial Interrupt นั้นเมื่อมีชุดข้อมูลเข้ามาทางขา RXD ของไมโคร คอนโทรลเลอร์ เมื่อชุดข้อมูลถูกนำไปเก็บไว้ในรีจิสเตอร์ SBUF เป็นที่เรียบร้อยแล้ว(Stop Bit) CPU ก็จะถูกอินเทอร์รัพท์ ซึ่งตำแหน่งแอดเดรสของหน่วยความจำหลังจากถูก อินเทอร์รัพท์ Serial Interrupt นั้นจะอยู่ที่ตำแหน่ง 0023H ของหน่วยความจำ ตัวอย่างโปรแกรมการรับข้อมูลผ่านทาง พอร์ตสื่อสารข้อมูลอนุกรม โดยการใช้งาน Serial Interrupt แสดงในตัวอย่าง โปรแกรมที่ 2-6

2.6 สัญญาณโทรศัพท์

เครื่องโทรศัพท์ที่เห็นกันอยู่ทั่วไป โดยมากจะเป็นแบบกดปุ่มซึ่งนิยมใช้ในปัจจุบัน เครื่อง โทรศัพท์ชนิดนี้จะมีการส่งสัญญาณที่มีค่าความถี่ที่แตกต่างกันออกไปในแต่ละหมายเลขที่มีอยู่ 10 ตัว โดยความถี่ที่ส่งออกไปเป็นความถี่ที่อยู่ในย่านความถี่เสียง ในการกดแต่ละหมายเลขครั้งหนึ่งจะมีสัญญาณเสียงที่ถูกมอดูเลต แล้วส่งออกไปยังชุมสายโทรศัพท์ 2 ความถี่ หรือที่เรียกว่าสัญญาณดี ทวีเอ็มเอฟ (Dual Tone Multi-Frequency : DTMF)

สัญญาณต่างๆที่ทางชุมสายโทรศัพท์ส่งมาตามคู่สายโทรศัพท์นั้น จะเป็นสัญญาณที่แจ้ง สถานะการใช้งานโทรศัพท์ ซึ่งแบ่งออกเป็น 4 ประเภท คือ

- 1 สัญญาณพร้อมหมუნ(Dial Tone) เป็นสัญญาณที่ทางชุมสายโทรศัพท์ว่าอุปกรณ์ต่างๆใน ชุมสายพร้อมที่จะทำการต่อ โทรศัพท์ให้กับผู้ใช้โทรศัพท์
- 2 สัญญาณเรียกกลับหรือสัญญาณแจ้งว่าสายว่าง (Ringback Tone) เป็นสัญญาณที่บอกผู้เรียก ใช้ทราบว่าทางสายของผู้ที่ถูกเรียกว่างและกำลังทำการเรียกอยู่

- 3 สัญญาณกระดิ่ง (Ringing Tone) เป็นสัญญาณที่ทางชุมสายส่งไปยังเครื่องผู้รับเพื่อบอกให้ทราบว่ามี การติดต่อมา
- 4 สัญญาณสายไม่ว่าง (Busy Tone) เป็นสัญญาณที่บอกให้ผู้เรียกใช้ทราบที่ไม่สามารถติดต่อกับเครื่องรับ โทรศัพท์เลขหมายนั้นในเวลานั้นได้

ตารางที่ 2.10 แสดงลักษณะของสัญญาณต่างๆ ที่ใช้ในการแจ้งสภาวะการใช้งานทางโทรศัพท์

ชนิดของสัญญาณ	การส่งสัญญาณ	ความถี่ (Hz)
สัญญาณพร้อมหมวน	ต่อเนื่อง ไม่ว่าง	350 มอดูเลทกับ 440
สัญญาณเรียกกลับ	ดิ่ง 2 วินาที เียบ 4 วินาที	440 มอดูเลทกับ 480
สัญญาณกระดิ่ง	ดิ่ง 2 วินาที เียบ 4 วินาที	25
สัญญาณแจ้งว่าสายไม่ว่าง	- ขาดหาย 30 ครั้งต่อวินาที เมื่อสายในชุมสายไม่ว่าง - ขาดหาย 60 ครั้งต่อนาทีเมื่อ เครื่องรับโทรศัพท์ที่ต้องการ ติดต่อถูกใช้งานอยู่ - ขาดหาย 120 ครั้งต่อนาที เมื่อทร้งค์ไม่ว่าง	440 มอดูเลทกับ 480

*หมายเหตุ ทร้งค์ หมายถึงอุปกรณ์ที่ใช้ในการสับสวิตช์ระหว่างคู่สายโทรศัพท์ ระดับสัญญาณระหว่างคู่สายโทรศัพท์นั้น มีทั้งสัญญาณที่เป็นไฟกระแสดตรงและไฟกระแสสลับ ซึ่งระดับแรงดันไฟฟ้าของสัญญาณระหว่างคู่สายจะแตกต่างกันไป

ตารางที่ 2.11 แสดงระดับสัญญาณระหว่างคู่สายโทรศัพท์ในการใช้งานต่างๆ

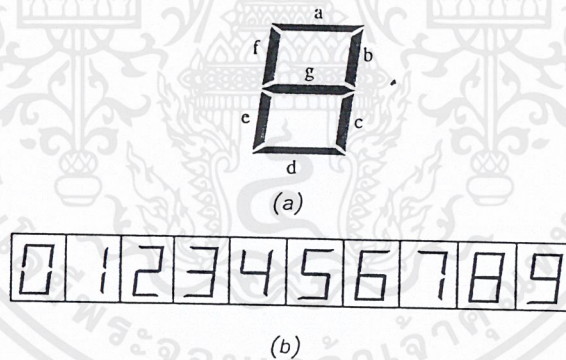
ช่วงเวลาการใช้งาน	ระดับสัญญาณ ไฟกระแสดตรง	ระดับสัญญาณ ไฟกระแสสลับ
ไม่ได้ใช้งาน(ไม่ได้ยกหูฟังขึ้น)	48 โวลต์	
ยกหูฟังขึ้น มีสัญญาณพร้อมหมวน	10 โวลต์	600 มิลลิโวลต์
ขณะกดหมายเลข	10 โวลต์	ไม่เกิน 0.5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีสัญญาณสายไม่ว่าง	10 โวลต์	400 มิลลิโวลต์
มีสัญญาณเรียกกลับ	10 โวลต์	400 มิลลิโวลต์
มีสัญญาณกระดิ่ง	48 โวลต์	110 มิลลิโวลต์
มีการพูดคุยระหว่างคู่สาย	10 โวลต์	ไม่เกิน 1 โวลต์

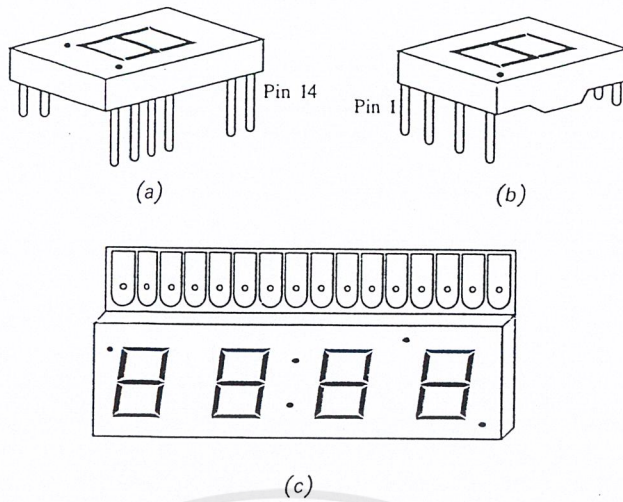
2.7 ตัวแสดงผลไดโอดเปล่งแสง 7 ส่วน (Seven – Segment LED Displays)

LED ตัวเลข 7 ส่วน ประกอบขึ้นจาก LED 7 ตัว บรรจุในถังตัวเดียวกันและได้รับการจัดเรียงเป็นรูปตัวเลข ถูกเรียกว่า ส่วน หรือ เซ็กเมนต์ (Segment) แต่ละส่วนหรือเซ็กเมนต์มีชื่อเรียกแตกต่างกันตามตำแหน่งที่ได้รับการจัดวางคือ a,b,c,d,e,f และ g ดังรูป ส่วน dp เป็น LED 1 ตัวที่บรรจุใน LED ตัวเลข 7 ส่วนใช้เป็นตัวแสดงจุดทศนิยมในกรณีที่มีการแสดงผลในลักษณะเลขที่มีทศนิยม



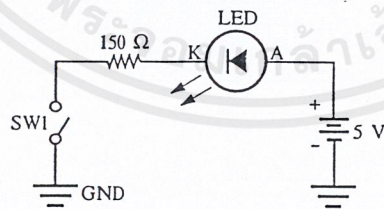
รูปที่ 2.21 แสดงลักษณะของ LED ตัวเลข 7 ส่วน

ลักษณะภายนอกของตัวแสดงผลไดโอดเปล่งแสง 7 ส่วนมีหลายแบบ เช่นในรูปที่ 2.22(a) เป็นตัวแสดงผล 1 หลัก ตัวถังพลาสติกแบบ DIP ปกติมี 14 ขา แบบที่สองแสดงในรูปที่ 2.22(b) เป็นตัวแสดงผล 1 หลักเช่นกัน ตัวถังแบบ 10 ขา (10 – pin single digit package) และแบบหลายหลัก(multidigit package) แสดงในรูปที่ 2.22(c)



รูปที่ 2.22 แสดงลักษณะภายนอกของตัวแสดงผลไดโอดเปล่งแสง 7 ส่วนแบบต่างๆ

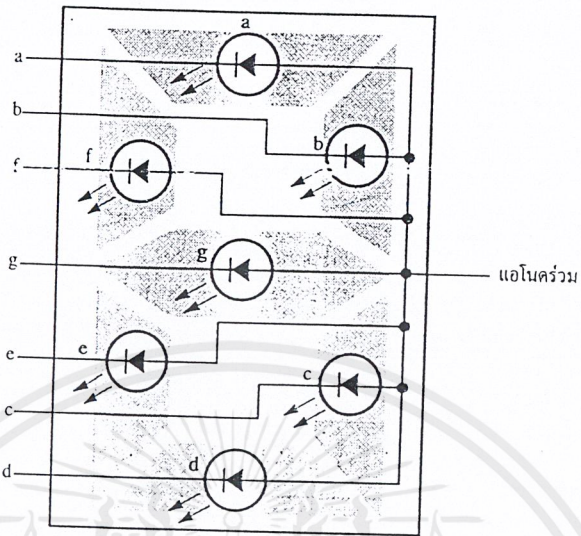
โครงสร้างภายในของไดโอดเปล่งแสง 7 ส่วนคืออุปกรณ์รอยต่อ พีเอ็น โดยปกติแต่ละส่วนจะกินกระแสไม่เกิน 20 mA เมื่อต่อกับแหล่งจ่ายไฟ 5 V จะต้องมีตัวจำกัดกระแส 150 โอห์ม ต่ออนุกรมอยู่ด้วย ดังรูปที่ 2.23(a) เพราะปกติมันจะทำงานที่แรงดันเพียง 1.7 - 2.1 V เท่านั้น เนื่องจากไดโอดเปล่งแสงมี 2 ขั้วคือ แอนโนด(A) และแคโทด (K) ดังนั้นจึงมีตัวแสดงผลไดโอดเปล่งแสง 7 ส่วน 2 ชนิด คือ ชนิด แอนโนดรวมดังรูปที่ 2.23(b) และชนิดแคโทดรวมดังรูปที่ 2.23(c) เมื่อต้องการให้ตัวแสดงผลไดโอดเปล่งแสง 7 ส่วนชนิดแอนโนดรวมทำงาน ต้องป้อนลอจิก 0 เข้าที่อินพุทของมัน และที่จุกร่วมต้องต่อกับแรงดันไฟ 5 V ดังรูปที่ 2.23(d) ในทำนองเดียวกัน เมื่อต้องการให้ตัวแสดงผลไดโอดเปล่งแสง 7 ส่วนชนิดแคโทดรวมทำงาน ต้องป้อนลอจิก 1 เข้าที่อินพุทของมัน และที่จุกร่วมต้องต่อร่วมกับกราวด์ (GND) ดังแสดงในรูปที่ 2.23(e)



(a)

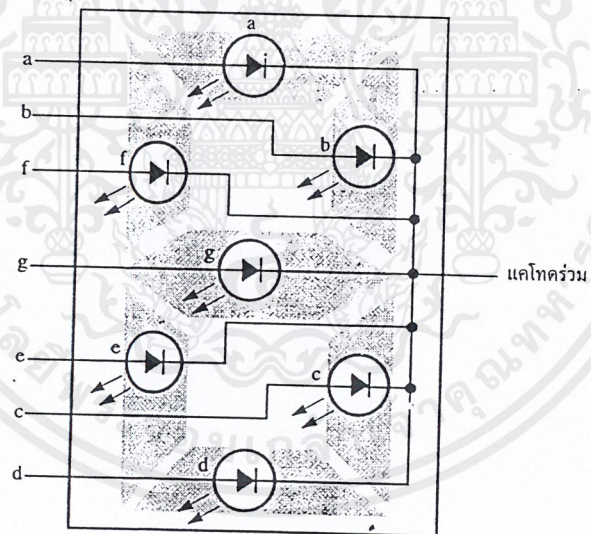
รูปที่ 2.23 แสดงโครงสร้างภายในและการควบคุมตัวแสดงผลไดโอดเปล่งแสง 7 ส่วนชนิดแอนโนดรวมและชนิดแคโทดรวม

แคโทดอินพุต



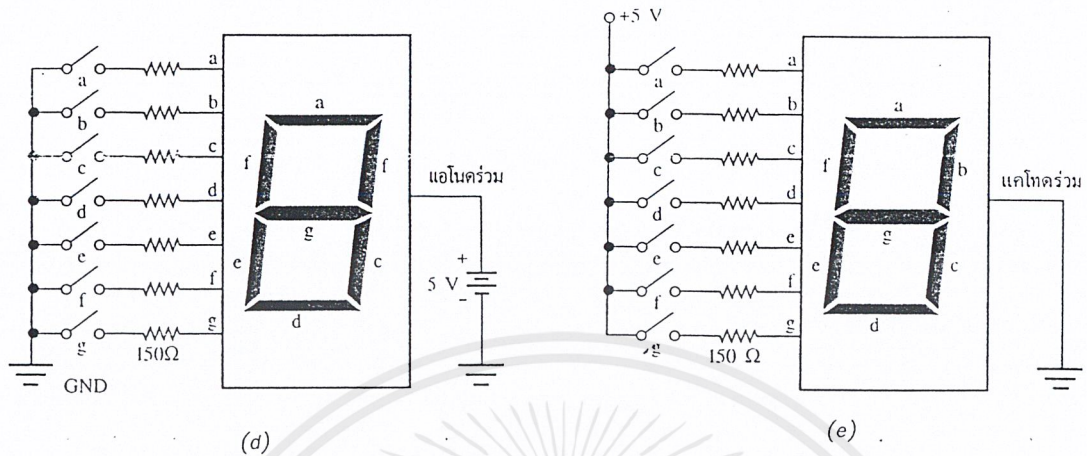
(b)

แอโนดอินพุต



(c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

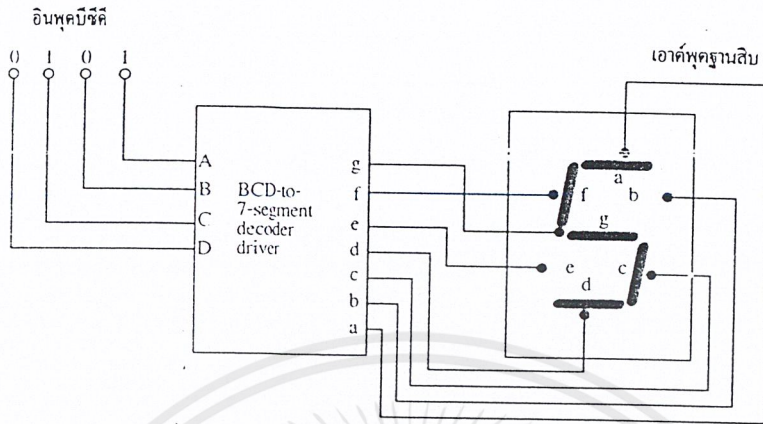


รูปที่ 2.23 ต่อ

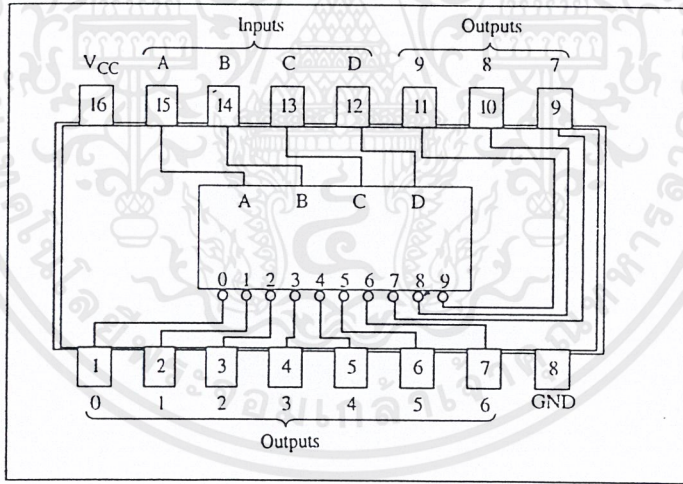
2.8 ตัวถอดรหัส (Decoders)

ตัวถอดรหัสจะรับข้อมูลหลังจากที่หน่วยประมวลผลและหน่วยความจำทำงานเสร็จแล้ว ข้อมูลดังกล่าวจะเป็นรหัสไบนารีหนึ่งในตัวเลขฐานสอง ข้อมูลที่เข้าสู่ตัวถอดรหัสบิตชีดี 8421 ตัวถอดรหัสจะแปลงรหัสดังกล่าวให้เป็นเลขฐานสิบ และแสดงผลที่ตัวแสดงผลไดโอดเปล่งแสง 7 ส่วน ตัวถอดรหัสพวกนี้อาจเรียกได้ว่าเป็น BCD to Seven Segment Decoder/Driver ลักษณะของตัวถอดรหัสบิตชีดี 8421 ให้ไปแสดงผลที่ตัวเลข 7 ส่วนแสดงได้ในรูปที่ 2.24 เมื่อนำข้อมูลรหัสบิตชีดี 8421 4 บิต ที่มีค่า 0101 เข้าที่อินพุทของตัวถอดรหัส (อินพุท A, B, C และ D) ทางด้านเข้าที่พุทของตัวถอดรหัสจะส่งสัญญาณลจิกไปควบคุมตัวแสดงผลไดโอดเปล่งแสง 7 ส่วน ให้ตัดเป็นเลขฐานสิบเท่ากับ 5 จะสังเกตเห็นว่าเข้าที่พุทของตัวถอดรหัสนี้จะมี 7 ขาคือ a-g เท่ากับจำนวนอินพุทของตัวแสดงผลไดโอดเปล่งแสง 7 ส่วน

ตัวถอดรหัสนอกจากที่แสดงในรูปที่ 2.24 แล้วยังมีอีกหลายชนิด เช่น ตัวถอดรหัสเบอร์ 7442 ถอดรหัสบิตชีดีเป็นเลขฐานสิบ (BCD-to-Decimal) หรือตัวถอดรหัสเบอร์ 7443 ถอดรหัสเกิน 3 เป็นเลขฐานสิบ (Excess-3-to-Decimal) หรือตัวถอดรหัสเกิน 3 และรหัสเกรย์เป็นเลขฐานสิบ (Excess-3-Gray-to-Decimal) มีลักษณะการจัดวางขาตั้งรูปที่ 2.25(a) และตารางความจริงแสดงการทำงานของตัวถอดรหัสดังกล่าวตั้งรูปที่ 2.25(b) และมีลักษณะโครงสร้างภายในตัวถอดรหัสเบอร์ 7442, 7443 และ 7444 ตั้งรูปที่ 2.25(c)



รูปที่ 2.24 แสดงการต่อตัวถอดรหัสกับตัวแสดงผล ไดโอดเปล่งแสง 7 ส่วน



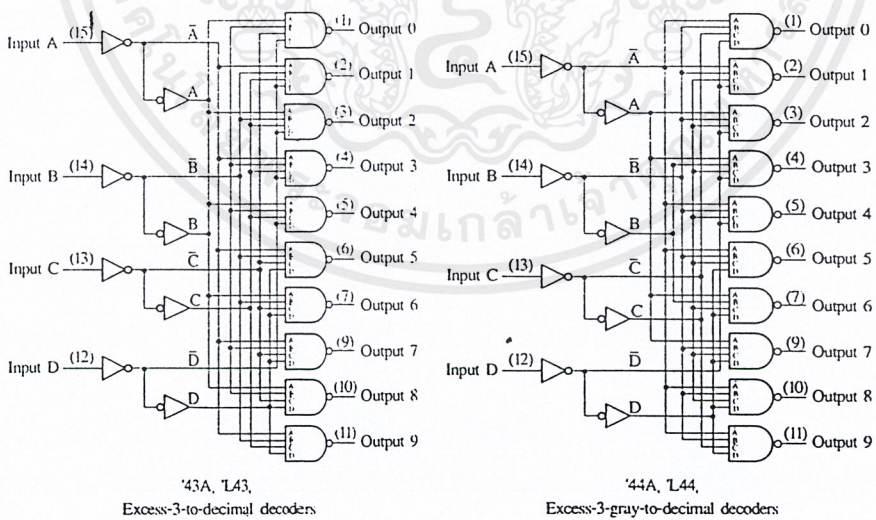
(a)

รูปที่ 2.25 แสดงการวางขาของตัวถอดรหัสเบอร์ 7442 ,7443 และ 7444 และ โครงสร้างภายในพร้อมตารางความจริงแสดงการทำงานของตัวถอดรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NO.	'42A, 'L42, LS42 BCD Input				'43A, 'L43 Excess-3- Input				'44A, 'L44 Excess-3-Gray Input				ALL TYPES Decimal Output									
	D	C	B	A	D	C	B	A	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	L	L	L	L	L	L	H	H	L	L	H	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	L	H	L	L	L	H	L	L	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	L	H	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	L	H	H	L	L	H	L	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	L	H	H	H	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	L	L	L	H	H	L	L	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	L	L	H	H	H	L	H	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H
8	H	L	L	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	L	H	H
9	H	L	L	H	H	H	L	L	H	L	H	L	H	H	H	H	H	H	H	H	L	H
INVALID	H	L	H	L	H	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
	H	L	H	H	H	H	H	L	H	L	L	H	H	H	H	H	H	H	H	H	H	H
	H	H	L	L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	L	H	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	H	L	L	L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H
H	H	H	H	L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H

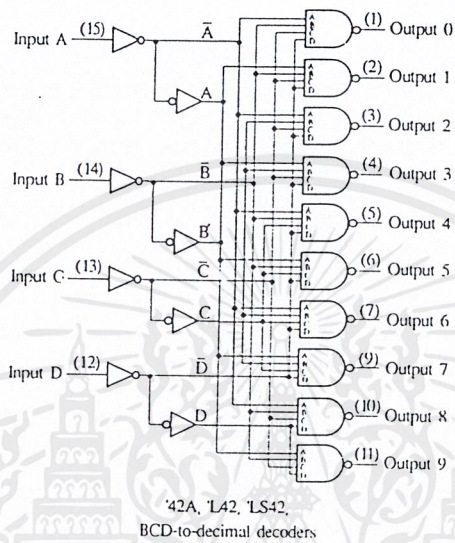
(b)



(c) แผนภาพกรอบแสดงรายละเอียดของวงจรภายในด้านอินพุตและเอาต์พุต

รูปที่ 2.25 ต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

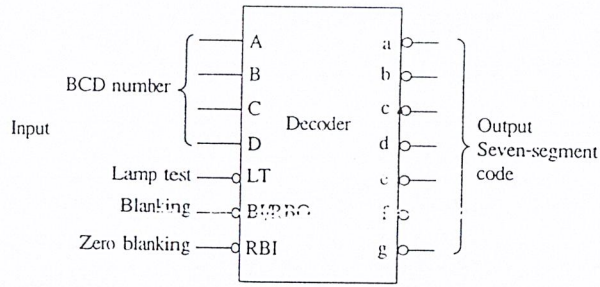


รูปที่ 2.25 ต่อ

2.9 ตัวถอดรหัสบีซีดีขับขั้วออกตัวแสดงผลตัวเลข 7 ส่วน

(BCD-to-Seven Segment Decoder/Driver)

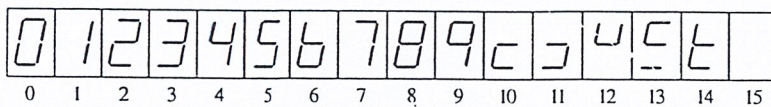
ตัวถอดรหัสที่นิยมใช้กันอย่างมากคือ ตัวถอดรหัสบีซีดี 8421 ให้เป็นเลขฐานสิบโดยขับสัญญาณออกที่ตัวแสดงผลไดโอดเปล่งแสง 7 ส่วน คือวงจรรวมที่แอล เบอร์ 7447 และเบอร์ 7448 ดังแสดงในรูปที่ 2.26(a) คือเบอร์ 7447 มีขาอินพุตรับข้อมูลรหัสบีซีดี 8421 จำนวน 4 บิต (คือ A, B, C, D) และมีขาอินพุตควบคุมอีก 3 ขา คือขา Lamp test (LT) และขา Blanking (BI/RBO) และขา Zero Blanking (RBI) สำหรับขาเข้าที่พ่วงของมันมี 7 ขาคือขา a ถึง g ที่จะนำไปต่อขับตัวเลข 7 ส่วน เพื่อแสดงผลเลขฐานสิบได้สำหรับเบอร์ 7447 ใช้สำหรับขับตัวเลข 7 ส่วน ชนิดแอมโนคร่วมเท่านั้น ตารางการทำงานของตัวถอดรหัสเบอร์ 7447 แสดงในรูปที่ 2.26(b) และรหัสเลขฐานสิบที่ตัวถอดรหัสนี้ถอดได้แสดงได้ในรูปที่ 2.26(c)



(a)

Decimal or Function	Inputs						BI/RBO	Outputs							Note
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	On	On	On	On	On	On	Off	1
1	H	X	L	L	L	H	H	Off	On	On	Off	Off	Off	Off	
2	H	X	L	L	H	L	H	On	On	Off	On	On	Off	On	
3	H	X	L	L	H	H	H	On	On	On	On	Off	Off	On	
4	H	X	L	H	L	L	H	Off	On	On	Off	Off	On	On	
5	H	X	L	H	L	H	H	On	Off	On	On	Off	On	On	
6	H	X	L	H	H	L	H	Off	Off	On	On	On	On	On	
7	H	X	L	H	H	H	H	On	On	On	Off	Off	Off	Off	
8	H	X	H	L	L	L	H	On	On	On	On	On	On	On	
9	H	X	H	L	L	H	H	On	On	On	Off	Off	On	On	
10	H	X	H	L	H	L	H	Off	Off	Off	On	On	Off	On	
11	H	X	H	L	H	H	H	Off	Off	On	On	Off	Off	On	
12	H	X	H	H	L	L	H	Off	On	Off	Off	Off	On	On	
13	H	X	H	H	L	H	H	On	Off	Off	On	Off	On	On	
14	H	X	H	H	H	L	H	Off	Off	Off	On	On	On	On	
15	H	X	H	H	H	H	H	Off	Off	Off	Off	Off	Off	Off	
BI	X	X	X	X	X	X	L	Off	Off	Off	Off	Off	Off	Off	2
RBI	H	L	L	L	L	L	L	Off	Off	Off	Off	Off	Off	Off	3
LT	L	X	X	X	X	X	H	On	On	On	On	On	On	On	4

(b)

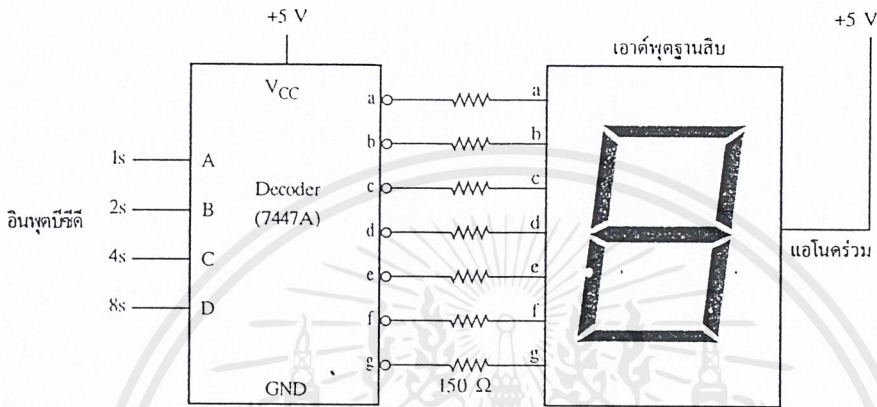


(c)

รูปที่ 2.26 แสดงลักษณะของตัวถอดรหัสเบอร์ 7447 และตารางแสดงการทำงานของตัวถอดรหัส และเข้าพุทที่ตัวถอดรหัสขั้วออกที่ตัวแสดงผลไดโอดเปล่งแสง 7 ส่วน

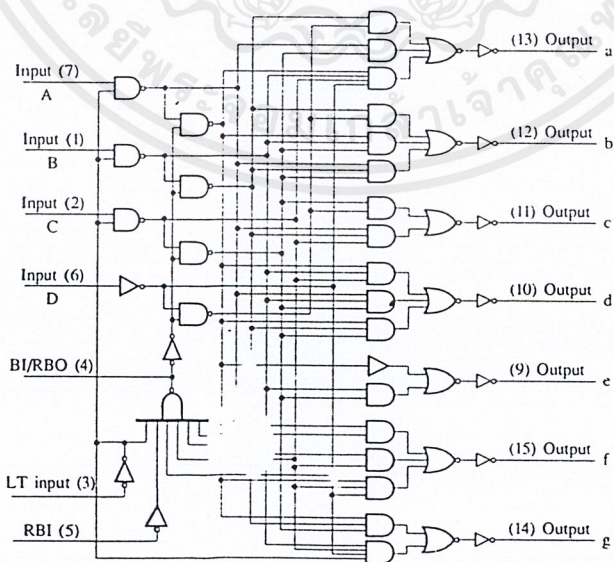
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวถอดรหัสตัวนี้เป็นชนิดการทำงานด้วยลอจิก 0 และแสดงผลแบบ Active Low output หมายความว่า เมื่อบีซีไดอินพุตเป็น 0001 เอ้าท์พุทที่ b และ c จะเป็นศูนย์ทำให้ตัวเลข 7 ส่วนชนิด แอนโนคร่วมติดสว่างเฉพาะส่วน b และ c ปรากฏเป็นตัวเลข 1 ตามรูปที่ 2.26(c) ดังนั้นการต่อตัวถอดรหัสเบอร์ 7447 เพื่อขังหมายเลข 7 ส่วน ชนิดแอนโนคร่วมที่ถูกต้องดังแสดงในรูปที่ 2.27



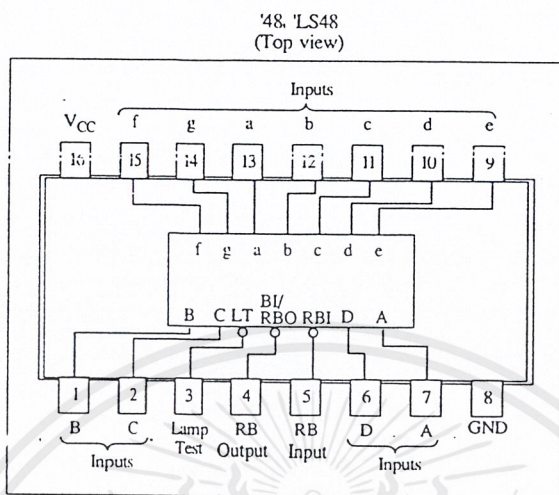
รูปที่ 2.27 การต่อวงจรตัวถอดรหัสเบอร์ 7447 เพื่อขังตัวเลขแสดงผล ไดโอดเปล่งแสง 7 ส่วนชนิดแอนโนคร่วม

โครงสร้างภายในของตัวถอดรหัสเบอร์ 7447 ในรูปที่ 2.28 จะเห็นว่าประกอบไปด้วยบีซีไดอินพุต 4 อินพุตและอินพุตควบคุม 3 อินพุตคือ LT ,BI/RBO ,RBI และเอ้าท์พุท 7 เอ้าท์พุทคือ a- g



รูปที่ 2.28 แสดง โครงสร้างภายในของตัวถอดรหัสเบอร์ 7447

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

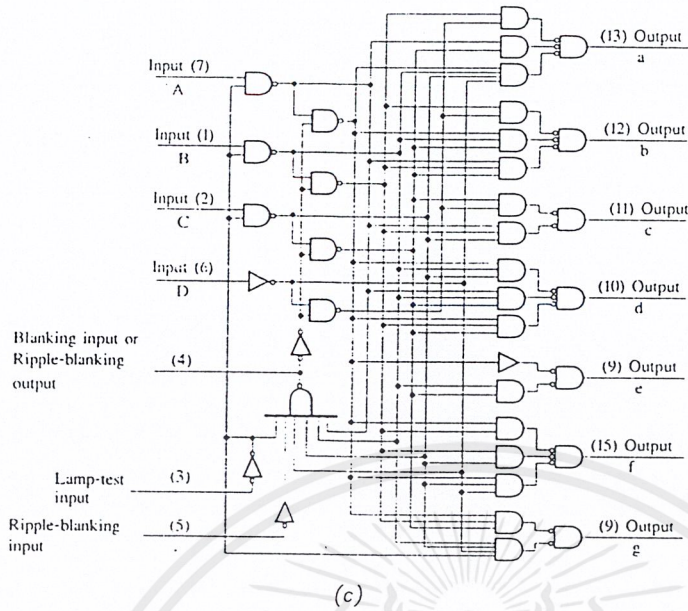
Decimal or Function	Inputs						BI / RBO [†]	Outputs							Note
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	H	H	H	H	H	L	1	
1	H	X	L	L	L	H	H	L	H	H	L	L	L		
2	H	X	L	L	H	L	H	H	H	L	H	H	L		
3	H	X	L	L	H	H	H	H	H	H	H	L	L		
4	H	X	L	H	L	L	H	L	H	H	L	L	H		
5	H	X	L	H	L	H	H	H	L	H	H	L	H		
6	H	X	L	H	H	L	H	L	L	H	H	H	H		
7	H	X	L	H	H	H	H	H	H	L	L	L	L		
8	H	X	H	L	L	L	H	H	H	H	H	H	H		
9	H	X	H	L	L	H	H	H	H	L	L	H	H		
10	H	X	H	L	H	L	H	L	L	L	H	L	H		
11	H	X	H	L	H	H	H	L	L	H	H	L	H		
12	H	X	H	H	L	L	H	L	H	L	L	H	H		
13	H	X	H	H	L	H	H	H	L	L	H	L	H		
14	H	X	H	H	H	L	H	L	L	L	H	H	H		
15	H	X	H	H	H	H	H	L	L	L	L	L	L		
BI	X	X	X	X	X	X	L	L	L	L	L	L	L	2	
RBI	H	L	L	L	L	L	L	L	L	L	L	L	L	3	
LT	L	X	X	X	X	X	H	H	H	H	H	H	H	4	

(b)

รูปที่ 2.29 แสดง โครงสร้างภายใน ลักษณะการจัดวงขา และตารางการทำงานของตัวถอดรหัสเบอร์

7448

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



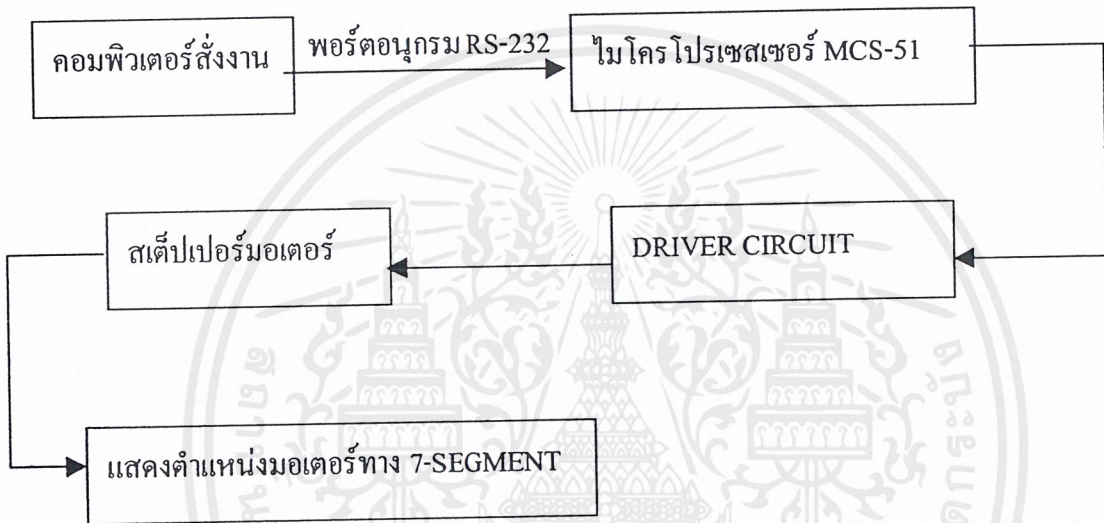
รูปที่ 2.29 ต่อ

สำหรับตัวถอดรหัสเบอร์ 7448 เป็นตัวถอดรหัสที่ออกมาแบบมาเพื่อขับตัวแสดงผลไดโอด เปล่งแสง 7 ส่วน ชนิดแคโทดรวมเท่านั้น เป็นชนิด Active Low input แต่ Active High output ซึ่งเป็นเข้าพุทชนิดตรงกันข้ามกับเบอร์ 7447 แต่รับสัญญาณแบบเดียวกัน สำหรับการจัดวางขาเหมือนกับเบอร์ 7447 ในรูปที่ 2.29(a) และตารางการทำงานแสดงได้ในรูปที่ 2.29 (b) สำหรับ โครงสร้างภายในตัวถอดรหัสเบอร์ 7448 แสดงในรูปที่ 2.29(c)

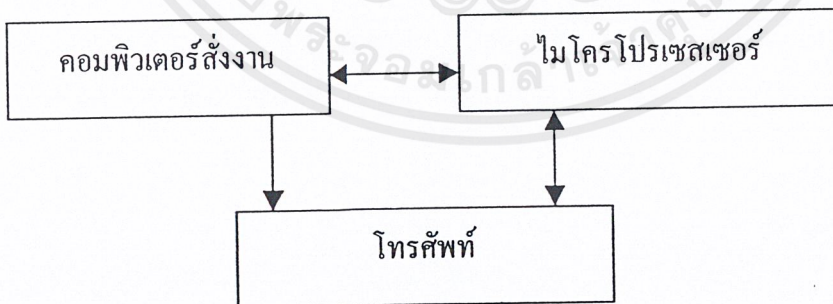
บทที่ 3

หลักการออกแบบ

โครงการนี้ประกอบด้วยส่วนของฮาร์ดแวร์และซอฟต์แวร์ซึ่งมีบล็อกไดอะแกรมของการทำงานโครงการดังนี้คือ



(a)



(b)

รูปที่ 3.1 บล็อกไดอะแกรมของโครงการ

(a) ส่วนควบคุมสเต็ปมอเตอร์ (b) ส่วนโทรศัพท์เตือนภัย

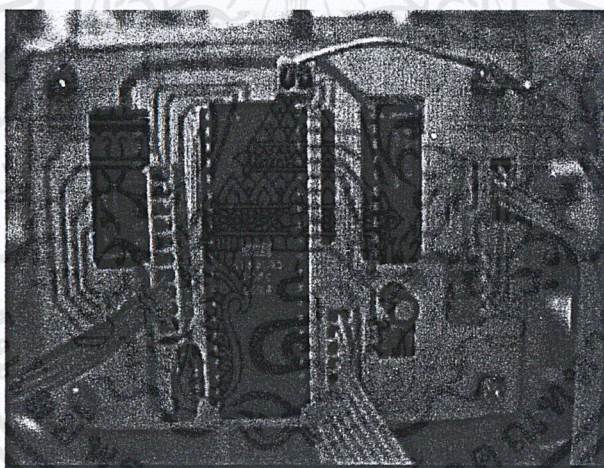
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การออกแบบในส่วนฮาร์ดแวร์ (Hardware)

โครงการนี้มีการแบ่งฮาร์ดแวร์ (Hardware) เป็น 2 ส่วนหลัก ๆ ด้วยกันคือ ส่วนที่สั่งงานและควบคุมการทำงานของสเต็ปมอเตอร์ทางคอมพิวเตอร์ และส่วนการเตือนภัยทางโทรศัพท์อัตโนมัติ โดยฮาร์ดแวร์ของแต่ละส่วนมีรายละเอียดดังนี้

3.1.1 วงจรไมโครคอนโทรลเลอร์

โครงการนี้ได้ทำการออกแบบและใช้ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ซึ่งจะใช้ชิปไอซีเบอร์ 89C51 เป็นส่วนประกอบหลักในการควบคุมการทำงานของสเต็ปมอเตอร์ 1 ตัวและควบคุมส่วนโทรศัพท์เตือนภัยอีก 1 ตัว



รูปที่ 3.2 แสดงวงจรไมโครคอนโทรลเลอร์ควบคุมสเต็ปมอเตอร์

ไมโครคอนโทรลเลอร์ตัวที่ 1 ต่อกับวงจรส่วนต่าง ๆ ดังนี้

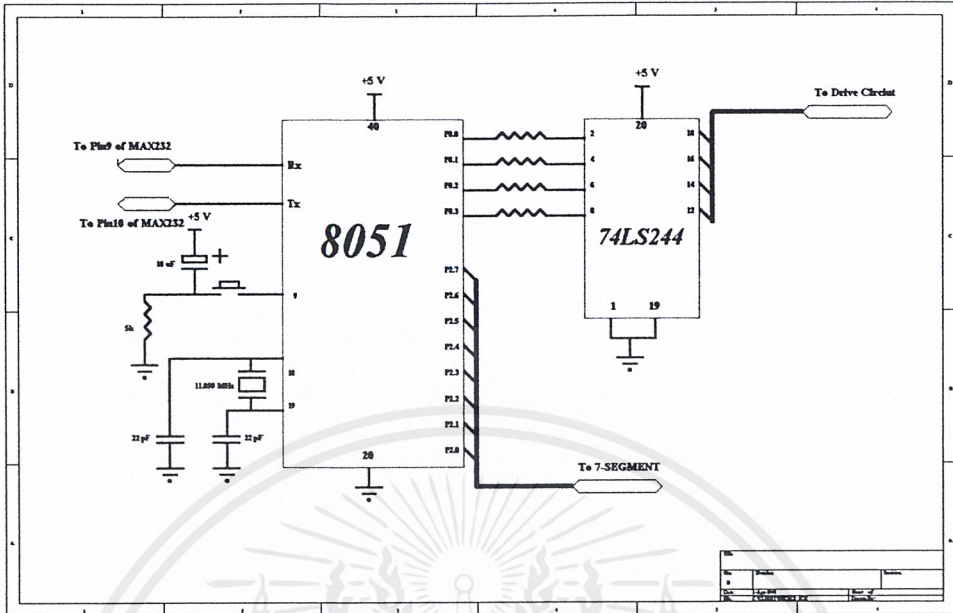
ที่ P0.0 - P0.3 (ขา 37 - ขา40) ต่อกับ บัฟเฟอร์เบอร์ 74LS244 ซึ่งจะต่อเชื่อมกับวงจรขับสเต็ปมอเตอร์

ที่ P2.0 - P2.7 (ขา 21- ขา 28) ต่อกับส่วนแสดงผล 7-Segment 2 หลัก

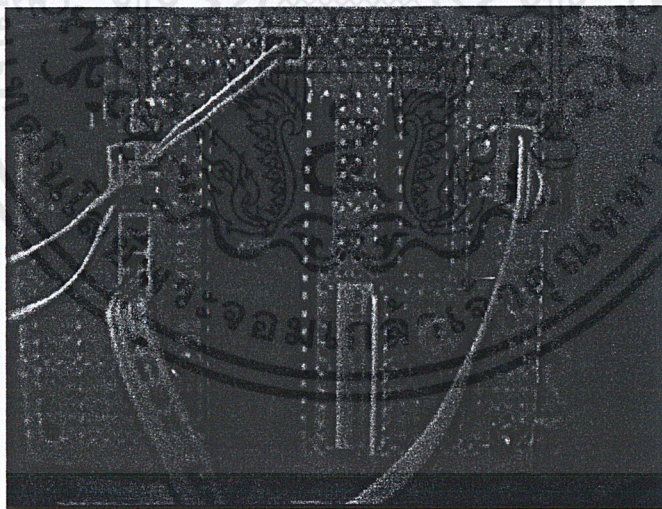
ที่ ขา Rx (ขา 10) ต่อกับ ขา TTL /CMOS Output (ขา 9) ของ IC MAX 232

ที่ ขา Tx (ขา 11) ต่อกับ ขา TTL/CMOS Input (ขา 10) ของ IC MAX 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงรูปวงจรไมโครคอนโทรลเลอร์ควบคุมสตีปมอเตอร์



รูปที่ 3.4 แสดงวงจรไมโครคอนโทรลเลอร์ควบคุมโทรศัพท์เตือนภัย

ไมโครคอนโทรลเลอร์ตัวที่ 2 ต่อกับวงจรส่วนต่าง ๆ ดังนี้

ที่ P0.0 (ขา 37)ต่อกับ บัฟเฟอร์เบอร์ 74LS244 ซึ่งจะต่อเชื่อมกับวงจรขับรีเลย์

ที่ P1.0 (ขา 1) ต่อกับ สวิตช์เตือนภัยทางโทรศัพท์

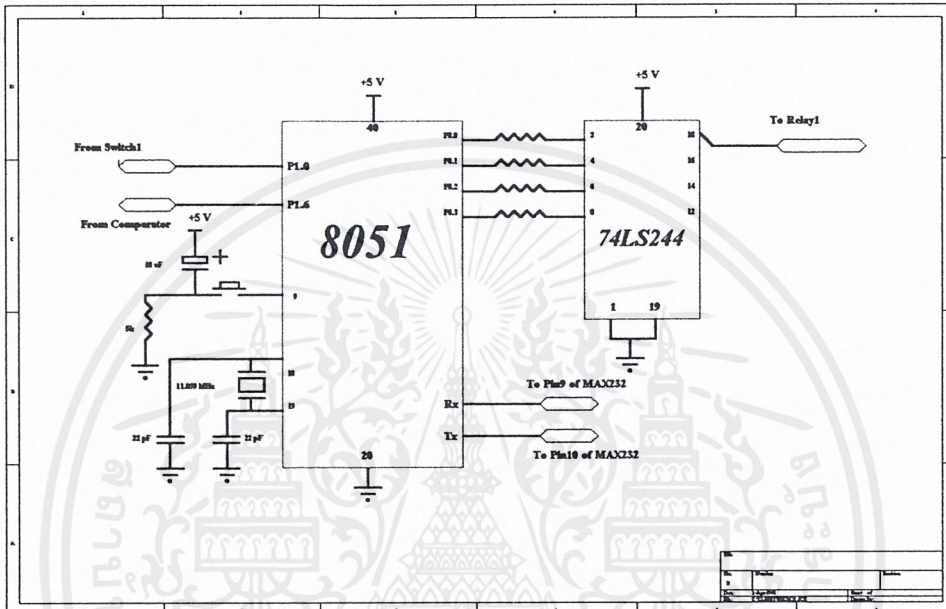
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ P1.6 (ขา 7) ต่อกับ Output ของ Comparator สายว่างหรือไม่ว่าง (ขา 1) ของ IC

LM324

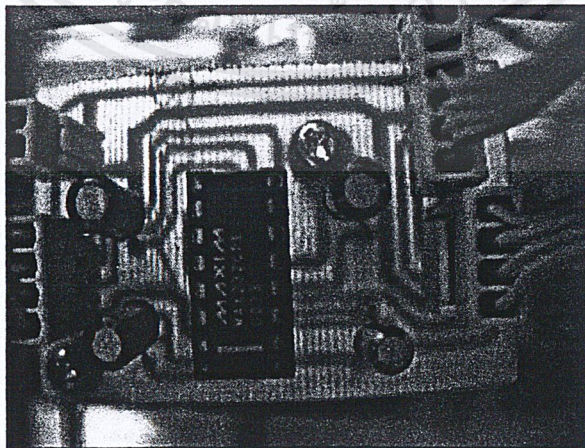
ที่ ขา Rx (ขา 10) ต่อกับ ขา TTL /CMOS Output (ขา 9) ของ IC MAX 232

ที่ ขา Tx (ขา 11) ต่อกับ ขา TTL/CMOS Input (ขา 10) ของ IC MAX 232



รูปที่ 3.5 แสดงวงจรไมโครคอนโทรลเลอร์ควบคุมโทรศัพท์เคลื่อนที่

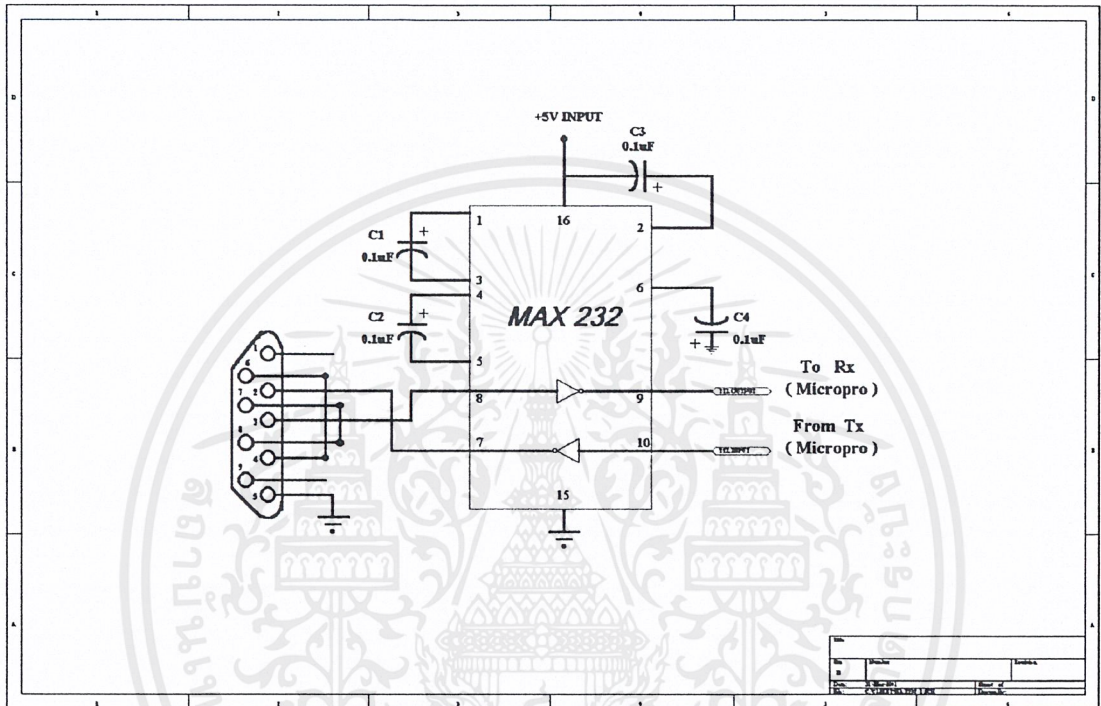
3.1.2 วงจรแปลงระดับแรงดันระหว่างคอมพิวเตอร์(+15v)กับไมโครคอนโทรลเลอร์(+5v)



รูปที่ 3.6 แสดงรูปวงจรแปลงระดับแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

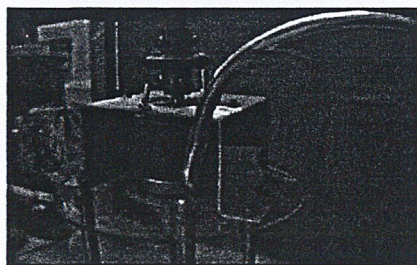
วงจรจะเปลี่ยนสัญญาณดิจิทัลจากลอจิก “0” จากไมโครคอนโทรลเลอร์ให้เป็น “+3V ถึง +15V” และระดับลอจิก “1” ให้เป็น “-3V ถึง -15V” แล้วส่งให้คอมพิวเตอรื และจะเปลี่ยนสัญญาณ “+5V ถึง +15V” จากคอมพิวเตอรืให้เป็นลอจิก “0” และเปลี่ยนสัญญาณ “-5V ถึง -15V” ให้เป็นลอจิก “1” เพื่อส่งให้ไมโครคอนโทรลเลอร์ประมวลผลต่อไป



รูปที่ 3.7 แสดงรูปวงจรแปลงระดับแรงดัน

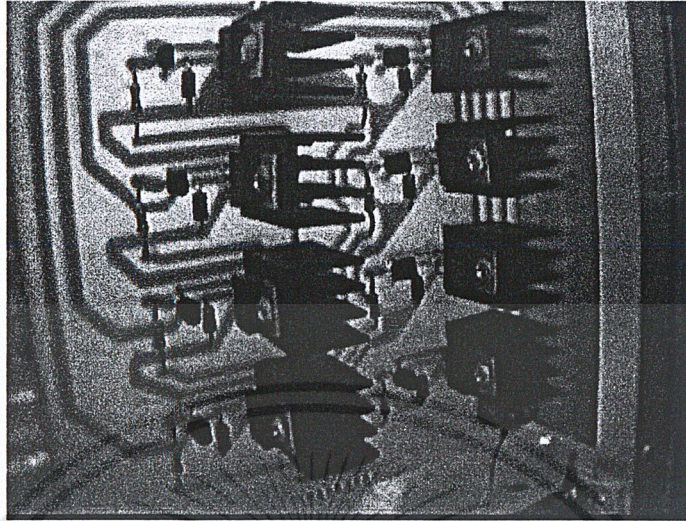
3.1.3 สเต็ปมอเตอร์ และ วงจรขับสเต็ปมอเตอร์ (DRIVER)

สเต็ปมอเตอร์ใช้สำหรับหมุนกลิ้งไปรอบๆ โดยที่การออกแบบให้กลิ้งสามารถหมุนได้ประมาณ 108 องศาในแนวอนโดยการนับสเต็ปการหมุนซึ่ง 1 สเต็ปของการหมุนมีค่าประมาณ 1.8 องศาต่อกลิ้งไป 108 องศาถือว่าการหมุนของสเต็ปมอเตอร์ 60 สเต็ปนั่นเอง

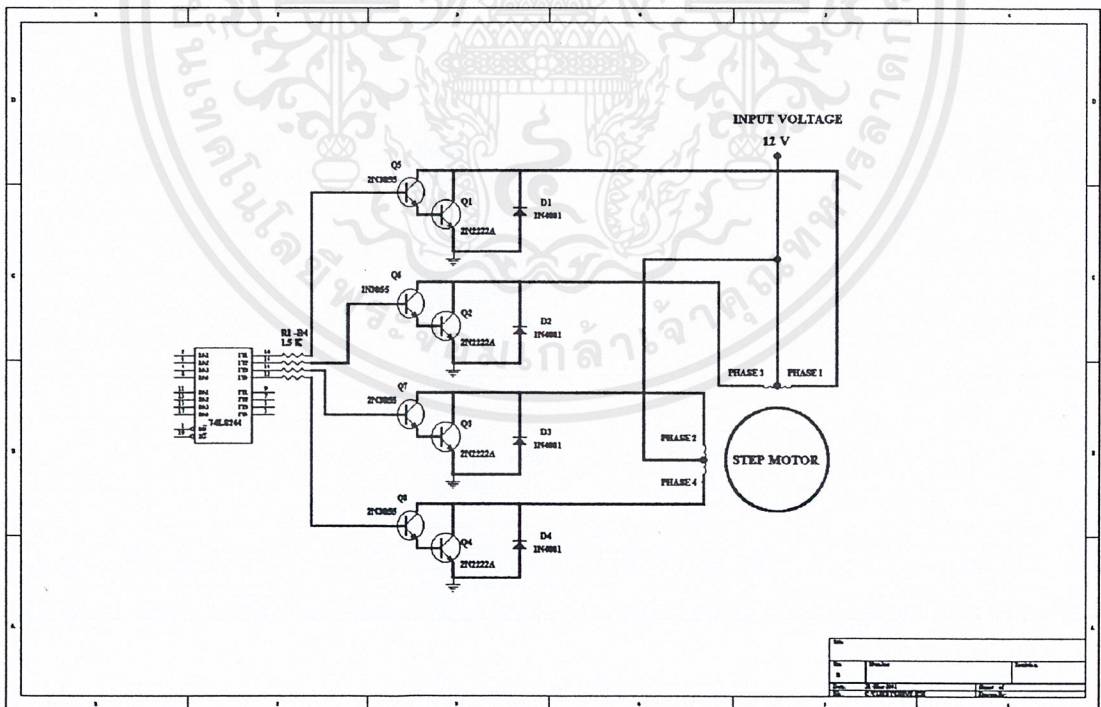


รูปที่ 3.8 รูปสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 วงจรขับมอเตอร์

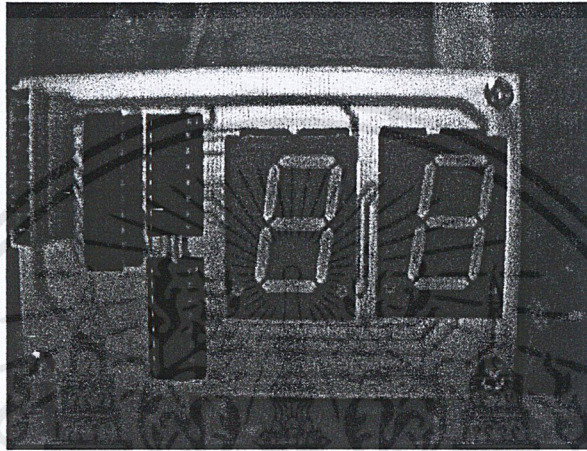


รูปที่ 3.10 วงจรขับมอเตอร์

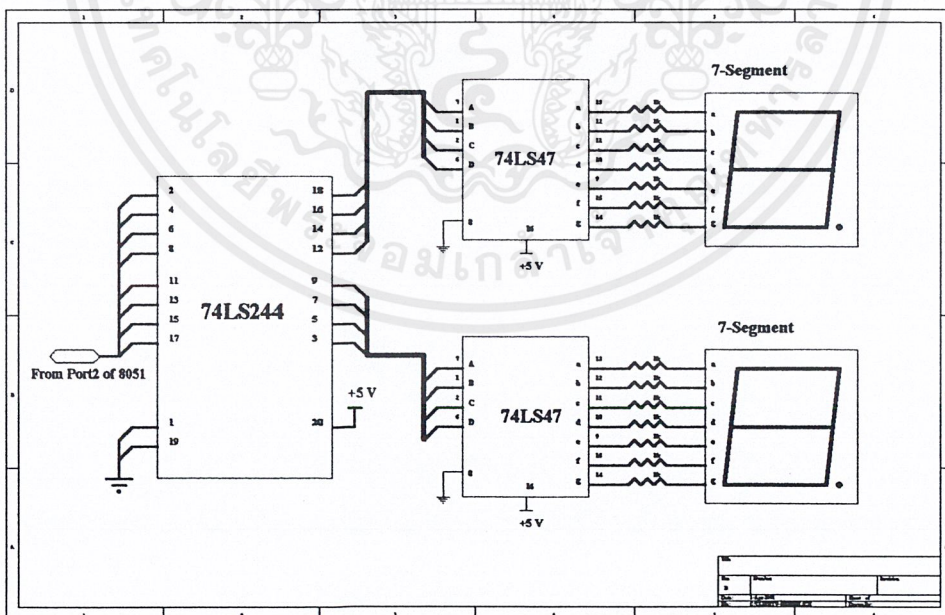
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 วงจรแสดงผลทาง 7 –SEGMENT

วงจรแสดงผลทาง 7 –SEGMENT ประกอบด้วยไอซี 74LS244 เป็นบัฟเฟอร์รับข้อมูลจากเอาต์พุตพอร์ท 2 ของไอซี 89C51 ซึ่งเป็นข้อมูลบ่งบอกถึงตำแหน่งของมอเตอร์ หลังจากที่ได้รับข้อมูลแล้วนำมาแปลงจาก BCD เป็น 7 –SEGMENT ด้วยไอซี 74LS47



รูปที่ 3.11 แสดงวงจรแสดงผลทาง 7-Segment

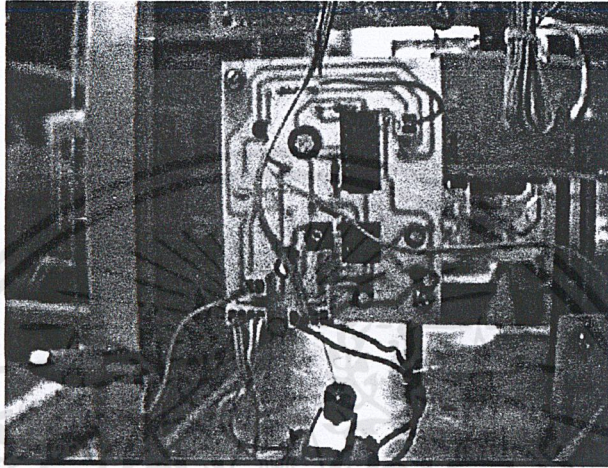


รูปที่ 3.12 แสดงรูปวงจแสดงผลทาง 7-Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 วงจรส่วนเชื่อมต่อโทรศัพท์

วงจรประกอบด้วยสองส่วนคือส่วนที่เป็นส่วนหมุนหมายเลขโทรศัพท์และส่วนตรวจจับสัญญาณโทรศัพท์ ทั้งสองส่วนแสดงดังรูปที่ 3.13



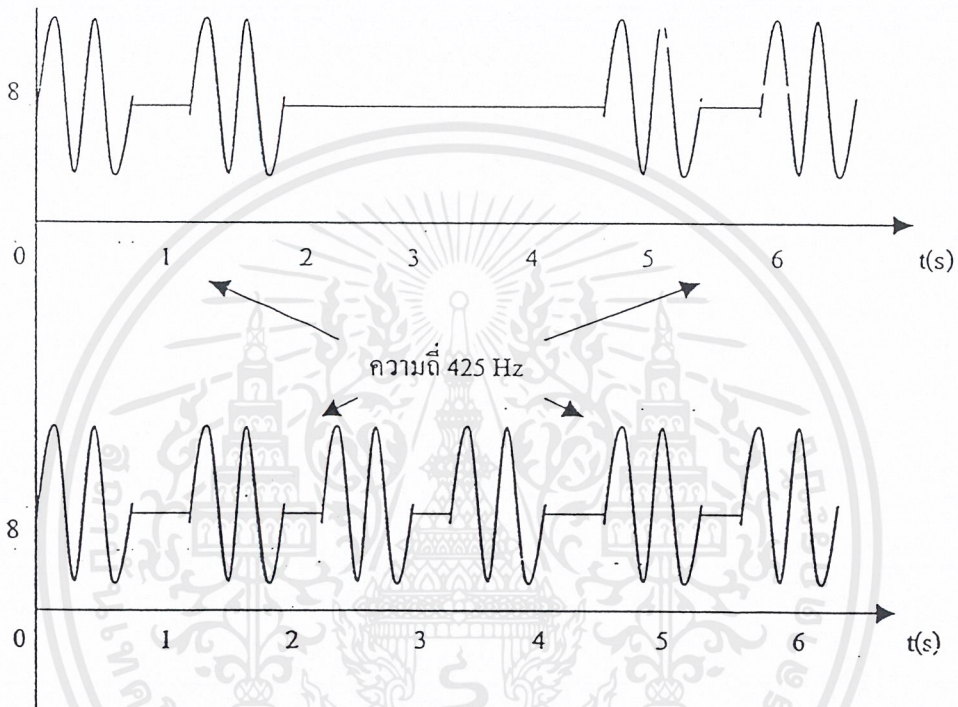
รูปที่ 3.13 รูปแสดงส่วนตรวจจับสัญญาณโทรศัพท์

ส่วนหมุนโทรศัพท์ประกอบด้วยแจ็กโฟนที่รับสัญญาณคิตีเอ็มเอฟจากชาวค์การ์ดของคอมพิวเตอร์ คือการหมุนโทรศัพท์ที่จะใช้สัญญาณเสียงส่งผ่านไปยังชุมสายโทรศัพท์เพื่อต่อสายปลายทาง

ส่วนตรวจจับสัญญาณโทรศัพท์เมื่อมีการตรวจจับได้ว่ามีผู้บุกรุกโดยตรวจจับด้วยไมโครโปรเซสเซอร์โดยการสมมุติโดยการกดปุ่ม ซึ่งต่อไว้ที่ Port 1.0 ไมโครโปรเซสเซอร์จะส่งสัญญาณไปบอกคอมพิวเตอร์โดยผ่านทางพอร์ตอนุกรม คอมพิวเตอร์ก็จะส่งสัญญาณให้ไมโครโปรเซสเซอร์สั่งงานให้มีการยกหูโทรศัพท์โดยการส่งคิตีเอ็มเอฟตัวที่ 1 และหมุนหมายเลขโทรศัพท์เพื่อเตือนภัยโดยอัตโนมัติ เมื่อมีการหมุนหมายเลขโทรศัพท์แล้วเพื่อความมั่นใจได้ว่าสายปลายทางได้รับเสียงเตือนแน่นอน จึงต้องมีการตรวจจับสัญญาณว่าคู่สายที่ต้องการติดต่อนั้นมีสภาวะอย่างไร ระหว่างสัญญาณสายว่างหรือสัญญาณเรียกกลับกับสายไม่ว่าง(Busy)ในกรณีที่สายว่างหรือเรียกว่ามีสัญญาณเรียกกลับ คอมพิวเตอร์จะนับสัญญาณเรียกกลับ ถ้าสัญญาณเรียกกลับหายไปแสดงว่ามีผู้รับสายแล้ว คอมพิวเตอร์ก็จะส่งเสียงเตือนภัยไปยังสายปลายทางและวางหู แต่เมื่อไม่มีผู้รับสาย คอมพิวเตอร์จะนับจำนวนสัญญาณเรียกกลับ เมื่อครบจำนวนที่ตั้งไว้แล้วคอมพิวเตอร์ จะสั่งไมโครโปรเซสเซอร์วางหูแล้วยกหูและหมุนหมายเลขใหม่ เช่นเดียวกับกรณีที่ไมโครโปรเซสเซอร์ตรวจ

จับได้ว่าสายไม่ว่างก็จะทำการวางหูแล้วยกหูและหมุนหมายเลขใหม่จนกว่าจะมีสัญญาณสายว่างหรือสัญญาณเรียกกลับ

สัญญาณเรียกกลับ และสัญญาณสายไม่ว่าง มีลักษณะเป็นความถี่ประมาณ 425 Hz โดยสัญญาณเรียกกลับจะคั่ง 0.5 วินาที หยุด 0.25 วินาที และคั่ง 0.5 วินาที ส่วนสัญญาณสายไม่ว่างจะคั่งและหยุดทุกๆ 0.5 วินาทีซึ่งสัญญาณเรียกกลับและสัญญาณสายไม่ว่างนั้นแสดงได้ในรูปที่ 3.14



รูปที่ 3.14 สัญญาณเรียกกลับ(บน)และสัญญาณสายไม่ว่าง(ล่าง)

วงจรการตรวจจับสัญญาณเรียกกลับและสัญญาณสายไม่ว่างนั้นอาศัยการทำงานของไอซี Tone Decoder เบอร์ LM567 ซึ่งสามารถปรับจูนความถี่ ในช่วงความถี่ประมาณ 425 Hz ของสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง โดยหาความถี่ได้จาก

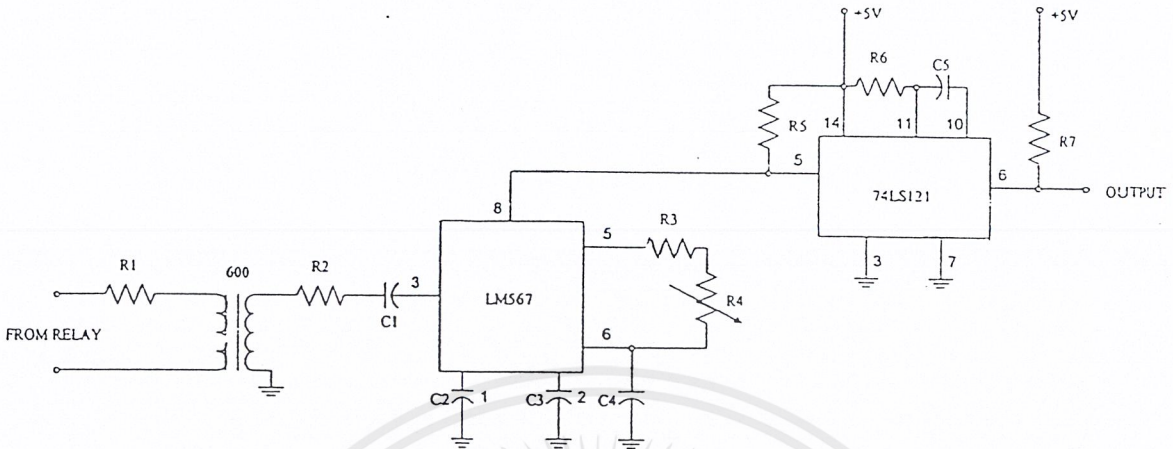
$$f_0 = 1.1/RC$$

เมื่อ f_0 เป็นค่าความถี่ที่ต้องการเลือก

R ความต้านทานระหว่างขา 5 กับ ขา 6

C ตัวเก็บประจุที่ขา 6

ดังนั้นเมื่อความถี่ที่ขา 3 มีค่า 425 Hz และตัวเก็บประจุมีค่า 2.2 ไมโครฟารัดจะได้ความต้านทานที่ขา 5 และขา 6 มีค่าประมาณ 1.3 กิโล โอห์ม ดังแสดงในรูปที่ 3.15



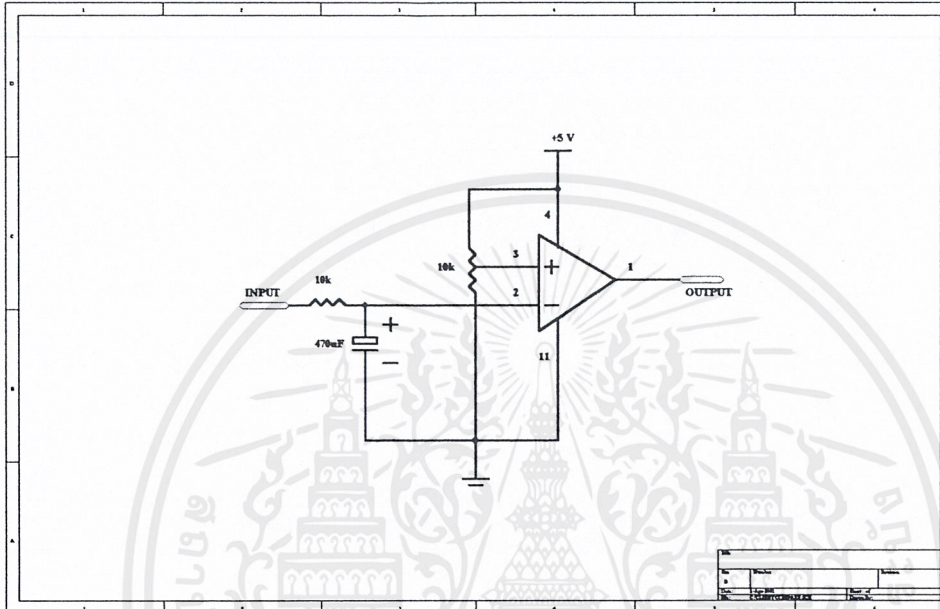
รูปที่ 3.15 วงจรการตรวจจับสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง

เมื่อมีสัญญาณความถี่ 425 Hz ตามที่ปรับจูนไว้เข้ามาที่ขา 3 ของไอซี LM567 จะทำให้เอาต์พุตที่ขา 8 มีสถานะต่ำหรือเป็น 0 โวลต์ แต่ถ้าป้อนความถี่อื่นที่นอกจากความถี่ที่ปรับจูนเอาไว้ หรือไม่ป้อนสัญญาณใดๆ สถานะที่ขา 8 เป็นสถานะสูงหรือ +5 โวลต์ จะเห็นว่าเราสามารถนำเอาไอซี LM567 มาประยุกต์ใช้ในการตรวจจับสัญญาณสายไม่ว่างได้ โดยการปรับจูนความถี่สามารถทำได้ โดยการปรับความต้านทานและตัวเก็บประจุจากภายนอก ต่อเข้ากับขาต่างๆของไอซีจากนั้นอาศัยการจูนค่าโดยปรับความต้านทานไฟฟ้าระหว่างขา 5 กับ 6 ของไอซี LM567 นั่นเอง

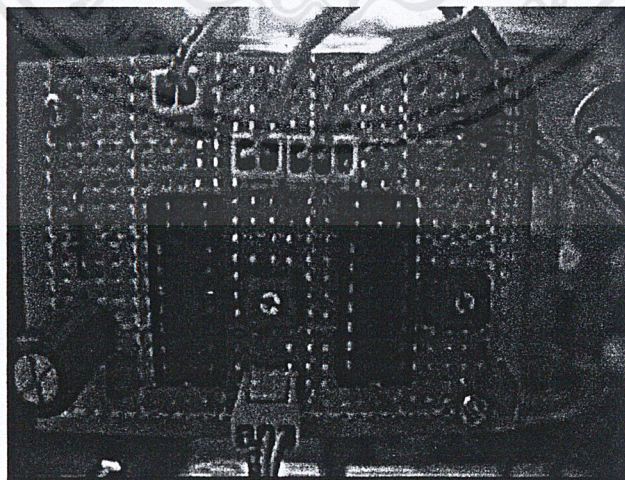
หลังจากที่เราจูนความถี่ได้ตรงกับความถี่ของสัญญาณสายไม่ว่างแล้ว จะพบว่าเมื่อไอซี LM567 ได้รับสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง สถานะของเอาต์พุตขาที่ 8 ของไอซีดังกล่าวจะเป็น 0 โวลต์ ตามเวลาของสัญญาณความถี่ทั้งสอง แต่เนื่องจากสัญญาณที่ได้จากไอซี LM567 ขาดความคมชัดของสัญญาณจึงต้องนำเข้าสู่วงจรหน่วง ซึ่งเราสามารถนำเอาเอาต์พุตที่ได้ไปตรวจสอบช่วงเวลาอีกทีเพื่อแยกความแตกต่างระหว่างสัญญาณเรียกกลับและสัญญาณสายไม่ว่างเพื่อประมวลผลในส่วนต่อไป

เมื่อได้สัญญาณที่เอาต์พุตของขา 6 ของไอซี 74121 แล้วจะทำการประมวลผลโดยนำสัญญาณที่ได้ไปแปลงเป็นไฟตรงโดยผ่านวงจรกรองแรงดัน RC ไฟตรงที่ได้จะมีลักษณะกระเพื่อมซึ่งเอาต์พุตที่แสดงว่าสายไม่ว่างและสายว่างจากไอซี 74121 เมื่อผ่านวงจรกรองแรงดัน RC จะมีระดับแรงดันไฟตรงที่ต่างกัน กล่าวคือไฟตรงของสัญญาณสายไม่ว่างจะมีระดับที่สูงกว่าสัญญาณสายไม่ว่าง ทำให้สามารถตรวจสอบสัญญาณสายโทรศัพท์ได้โดยการใช้วงจรเปรียบเทียบแรงดันและเมื่อมีสัญญาณสายว่างแล้วการตรวจจับการรับสายปลายทางก็ทำเช่นเดียวกับการตรวจจับ

สัญญาณสายไม่ว่าง โดยการเปรียบเทียบแรงดัน โดยแรงดันที่ได้จากสัญญาณสายว่างจะมีระดับที่สูงกว่าสัญญาณที่มีผู้รับสายปลายทางแล้ว รูปวงจรรองแรงดันและวงจรเปรียบเทียบแรงดันแสดงดังรูปที่ 3.16

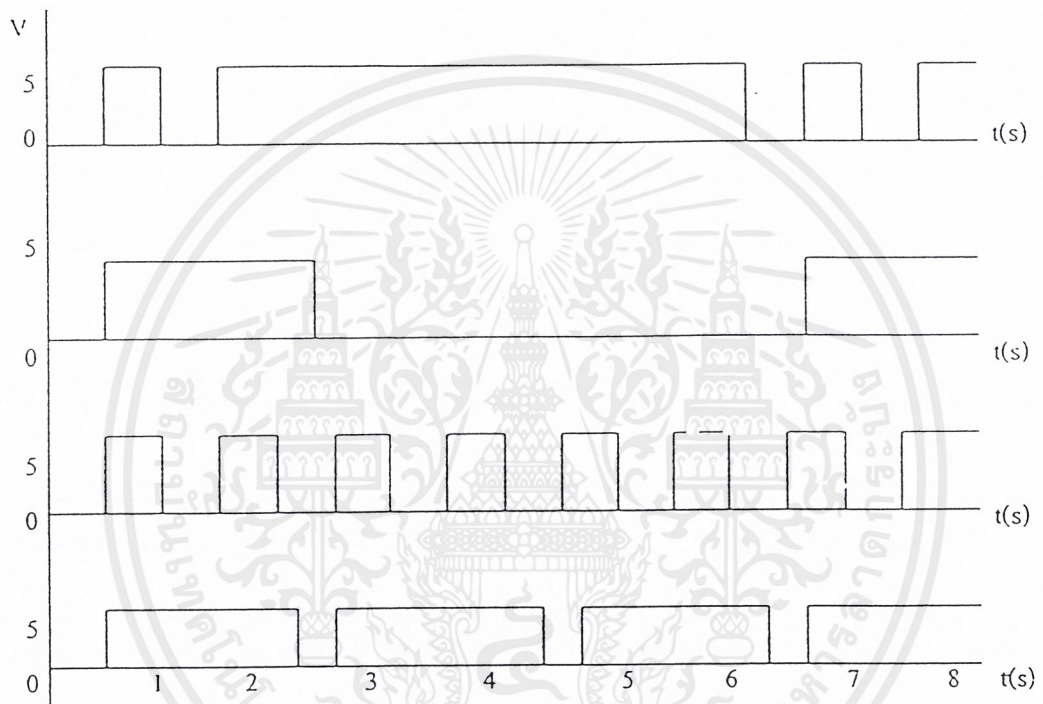


รูปที่ 3.16 รูปวงจรรองแรงดันและวงจรเปรียบเทียบแรงดัน



รูปที่ 3.17 วงจรรองแรงดันและวงจรเปรียบเทียบแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



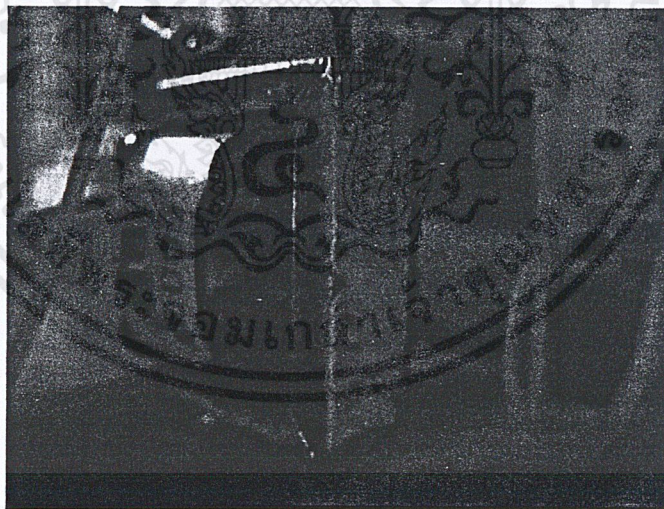
รูปที่ 3.18 แสดงสัญญาณที่ได้ในการตรวจจับสัญญาณเรียกกลับและสายไม่ว่าง

- (ก) เอ้าท์พุทที่ขา 8 ของ LM567 เมื่อได้รับสัญญาณเรียกกลับ
- (ข) เอ้าท์พุทที่ขา 6 ของ ไอซี 74121 เมื่อได้รับสัญญาณเรียกกลับ
- (ค) เอ้าท์พุทที่ขา 8 ของ LM567 เมื่อได้รับสัญญาณสายไม่ว่าง
- (ง) เอ้าท์พุทที่ขา 6 ของ ไอซี 74121 เมื่อได้รับสัญญาณสายไม่ว่าง

3.1.6 ส่วนหม้อแปลง 600-600 โห้หม

ในการที่จะทำให้ระบบสามารถส่งเบอร์โทรศัพท์หรือสัญญาณดีทีเอ็มเอฟ ออกไปแจ้งเหตุ โดยอัตโนมัติได้ตลอดเวลาที่เกิดภัยไม่ว่าเจ้าของอยู่บ้านหรือสำนักงานอยู่หรือไม่นั้น เครื่องโทรศัพท์ จะต้องสามารถส่งสัญญาณดีทีเอ็มเอฟออกไปยังชุมสายโทรศัพท์ได้โดยไม่ต้องยกหูโทรศัพท์

เมื่อยกหูโทรศัพท์ ชุมสายจะมองเห็นเครื่องโทรศัพท์นั้นมีความต้านทานอยู่ 600 โห้หม จากนั้นชุมสายจะจ่ายกระแสให้แก่เครื่องโทรศัพท์ ทำให้สามารถกำเนิดสัญญาณดีทีเอ็มเอฟ และส่งหมายเลขโทรศัพท์ติดต่อสื่อสารกันได้ จากหลักการเบื้องต้นนี้ ในโครงการนี้จึงได้ใช้หม้อแปลง ขนาด 600-600 โห้หม มาต่อในวงจร โดยปกติเมื่อระบบยังไม่เข้าสู่โหมดการทำงานเพื่อแจ้งเหตุและเตือนภัยโดยอัตโนมัติ นั้น ระบบจะต่อคู่สายระบบจะต่อคู่สายโทรศัพท์เข้าสู่ชุมสายซึ่งจะสามารถใช้โทรศัพท์ได้ตามปกติ แต่เมื่อระบบได้รับสัญญาณให้ส่งหมายเลขโทรศัพท์เพื่อเตือนภัย หม้อแปลง จะถูกต่อขนานเข้ามาในระบบ โดยที่เครื่องโทรศัพท์ไม่ได้ยกหู แต่ชุมสายจะมองเห็นเสมือนว่าได้ยกหูขึ้น ดังนั้นชุมสายจะจ่ายกระแสไปยังเครื่องโทรศัพท์และรอรับสัญญาณดีทีเอ็มเอฟจากเครื่องโทรศัพท์



รูปที่ 3.19 แสดงหม้อแปลง 600-600 โห้หม

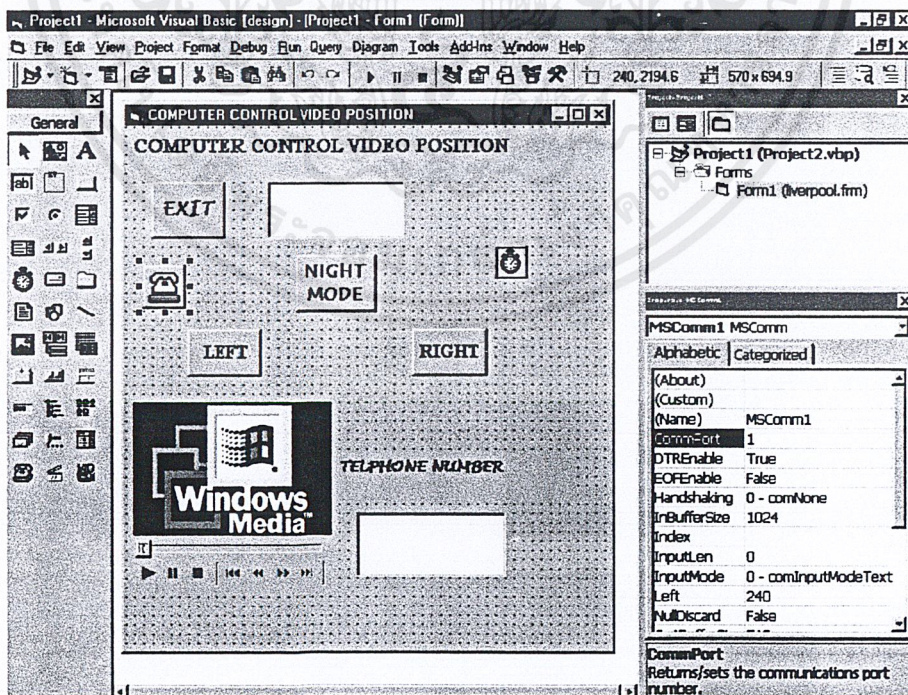
3.2 การออกแบบในส่วนซอฟต์แวร์ (Software)

ในการควบคุมการทำงานของส่วนฮาร์ดแวร์ทั้งหมดจะต้องอาศัยการเขียนโปรแกรมใน 2 ส่วนคือ ส่วนโปรแกรมภาษา Visual Basic 6.0 ซึ่งนำมาใช้เป็นฟอร์มของส่วนควบคุมการทำงาน จากคอมพิวเตอร์แล้วส่งข้อมูลต่อมาให้ไมโครโปรเซสเซอร์ และ ส่วนภาษา Assembly ที่ควบคุมการทำงานของไมโครโปรเซสเซอร์เพื่อรับหรือส่งข้อมูลกับคอมพิวเตอร์ในการประมวลผลแล้วส่งให้ส่วน ฮาร์ดแวร์ทำงานตามที่ประมวลผลได้

3.2.1 การสื่อสารผ่านทาง Serial Port ด้วย Visual Basic

การโปรแกรมสื่อสารผ่านทาง Serial Port ด้วย Visual Basic นั้น สิ่งที่ต้องเรียนรู้ประกอบไปด้วยรายละเอียดดังต่อไปนี้

1. ต้องมี MsCom Control ซึ่งก็มีอยู่แล้วเมื่อทำการติดตั้ง Visual Basic
2. กำหนด Custom Control MSCOMM.OCX โดยการเลือก MSCOMM.OCX จากเมนู **Project** → **Component** → **Microsoft Comm Control 6.0** (โครงการนี้ใช้ Visual Basic 6.0) จะได้ Control MSCom (รูปโทรศัพท์) ปรากฏในส่วนของ Toolbox
3. นำ Control MSCOMM ไปไว้ใน Form ที่ต้องการ
4. กำหนด Properties ให้กับ Control เขียน Function ที่เกี่ยวข้องกับ MSCOMM

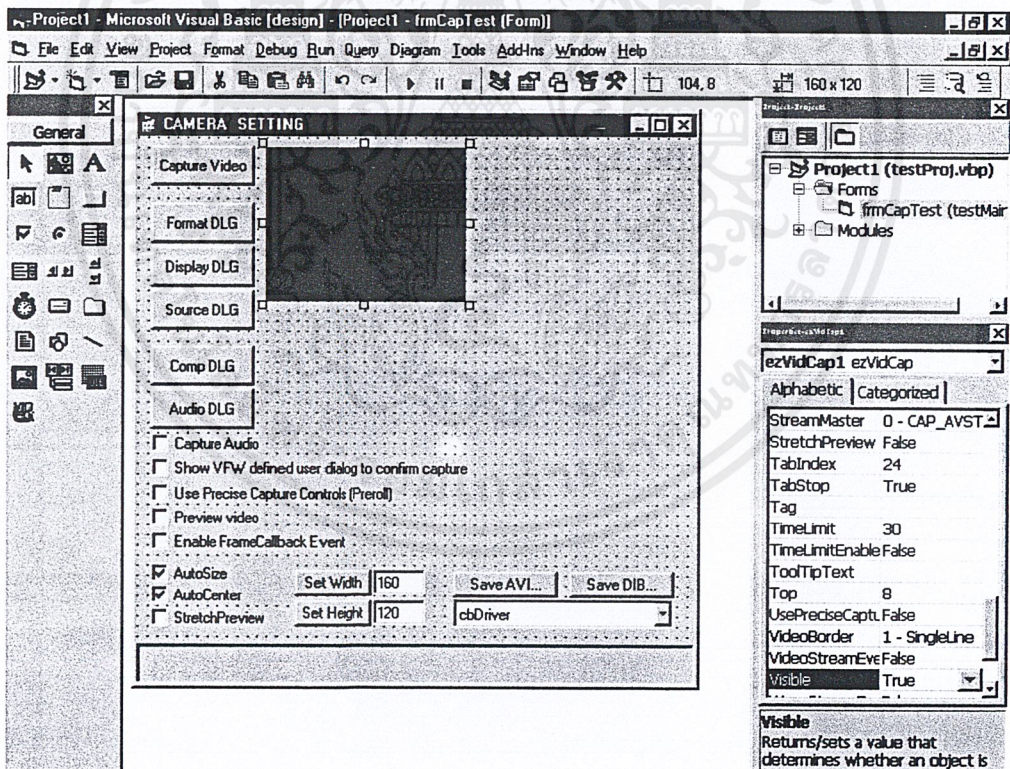


รูปที่ 3.20 Control MSCom (รูปโทรศัพท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งสำคัญที่ต้องกำหนดเพื่อให้ MSCOMM Control ทำงานได้คือ

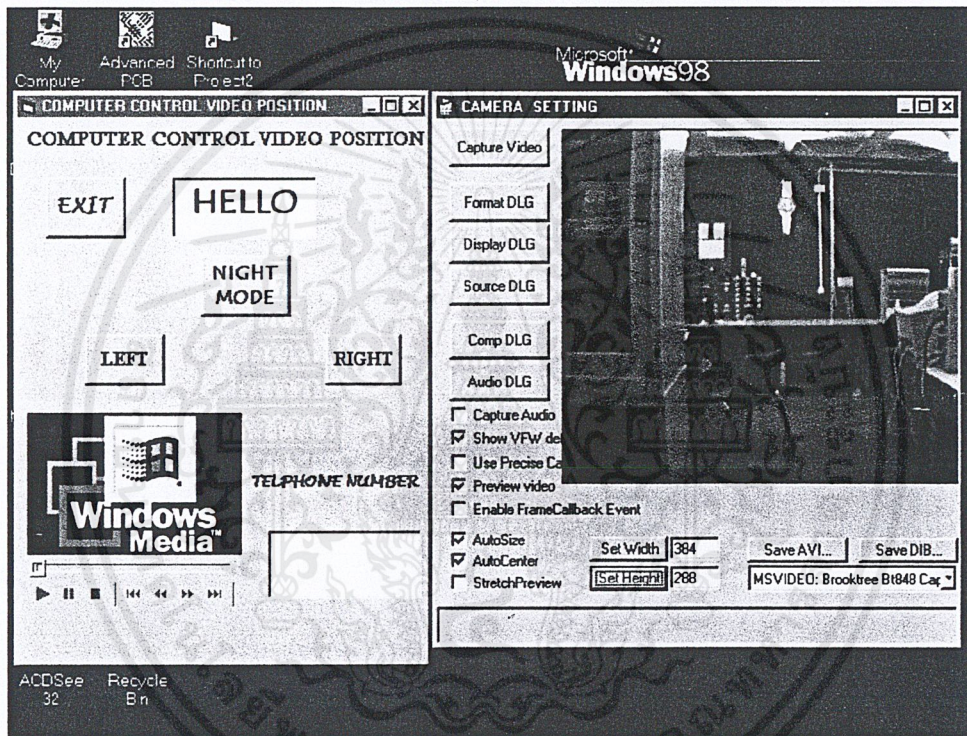
1. กำหนด ComPort โดยใช้ CommPort Property เช่น MSComm1.CommPort = 1 คือ เลือกใช้พอร์ต COM 1
2. กำหนด Setting โดยใช้ Setting Property เช่น MSComm1.Setting = "9600,N,8,1" คือกำหนดให้ 9600 baud , No Parity , 8 data , and 1 stop bit
3. กำหนด InputLen โดยใช้ InputLen Property เช่น MSCom1.InputLen = 0 คือ การให้มีการอ่านข้อมูลทั้งหมดในบัฟเฟอร์เมื่อมีการใช้ Input Propoty
4. กำหนด PortOpen โดยใช้ PortOpen Property เช่น MSCom1.PortOpen = True เป็นหาการเปิด Port ใช้งาน
5. กำหนด Rthreshold Property เมื่อต้องการ ใช้งาน Event – Driven เช่น MSComm1.Rthershold = 1 คือ ให้เกิด Event – Driven ทุกครั้งเมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล



รูปที่ 3.21 Control ezVidCap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

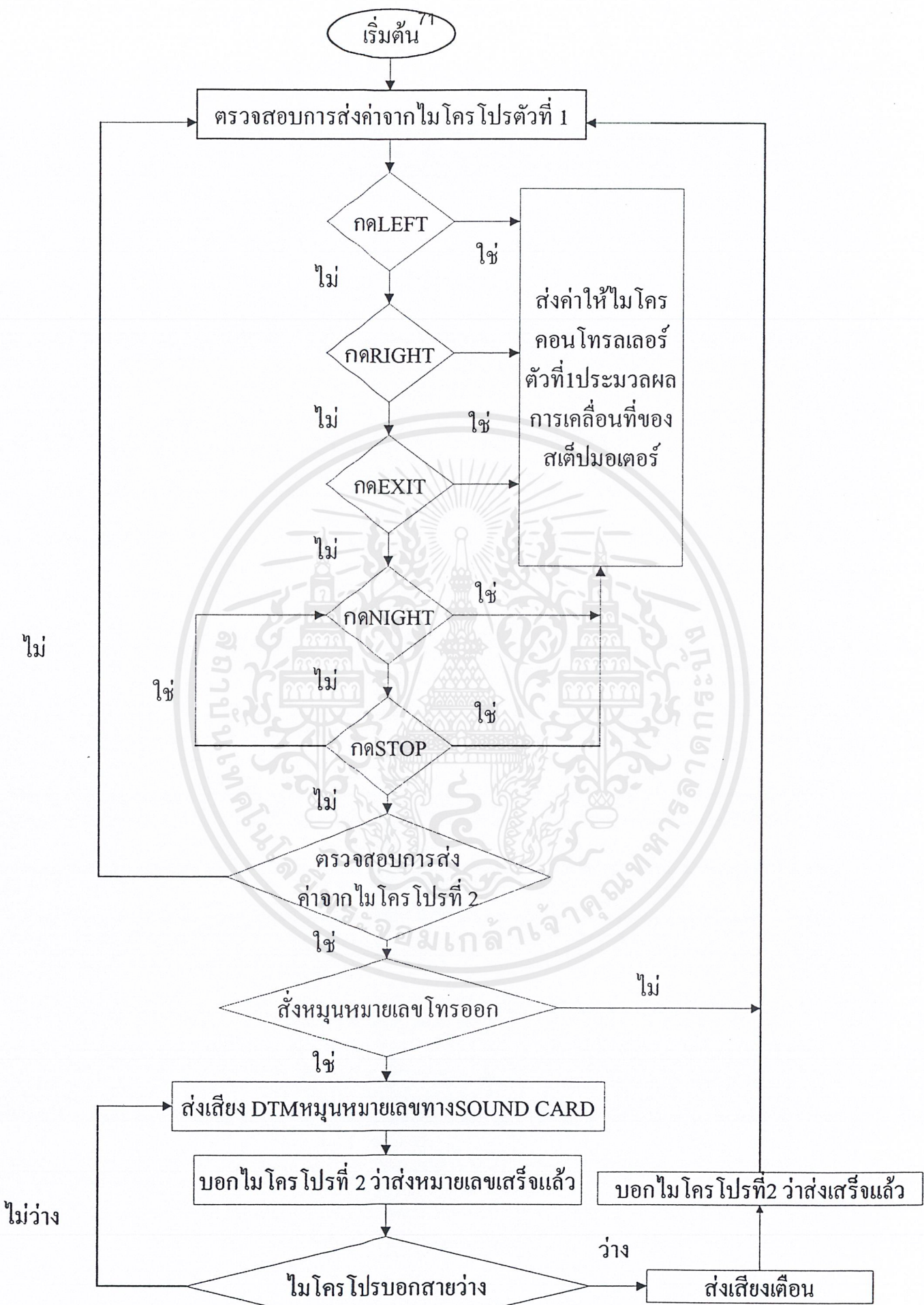
ในส่วนภาพจากกล้องที่แสดงผ่านทาง Visual Basic ก็ต้องทำการ Add ตัว Component ตัวหนึ่งลงไป คือ ตัว eZVidCap Component By Ray Merser (VB6) จากนั้นก็นำ Component ดังกล่าวไปไว้ใน Form ที่ต้องการ ดังรูปที่ 3.20



รูปที่ 3.22 ฟอรัมของ COMPUTER CONTROL VIDEO POSITION และ ฟอรัมของ CAMERA SETTING

3.2.2 โพล์ซาร์ทในส่วนของโปรแกรมภาษา Visual Basic 6.0

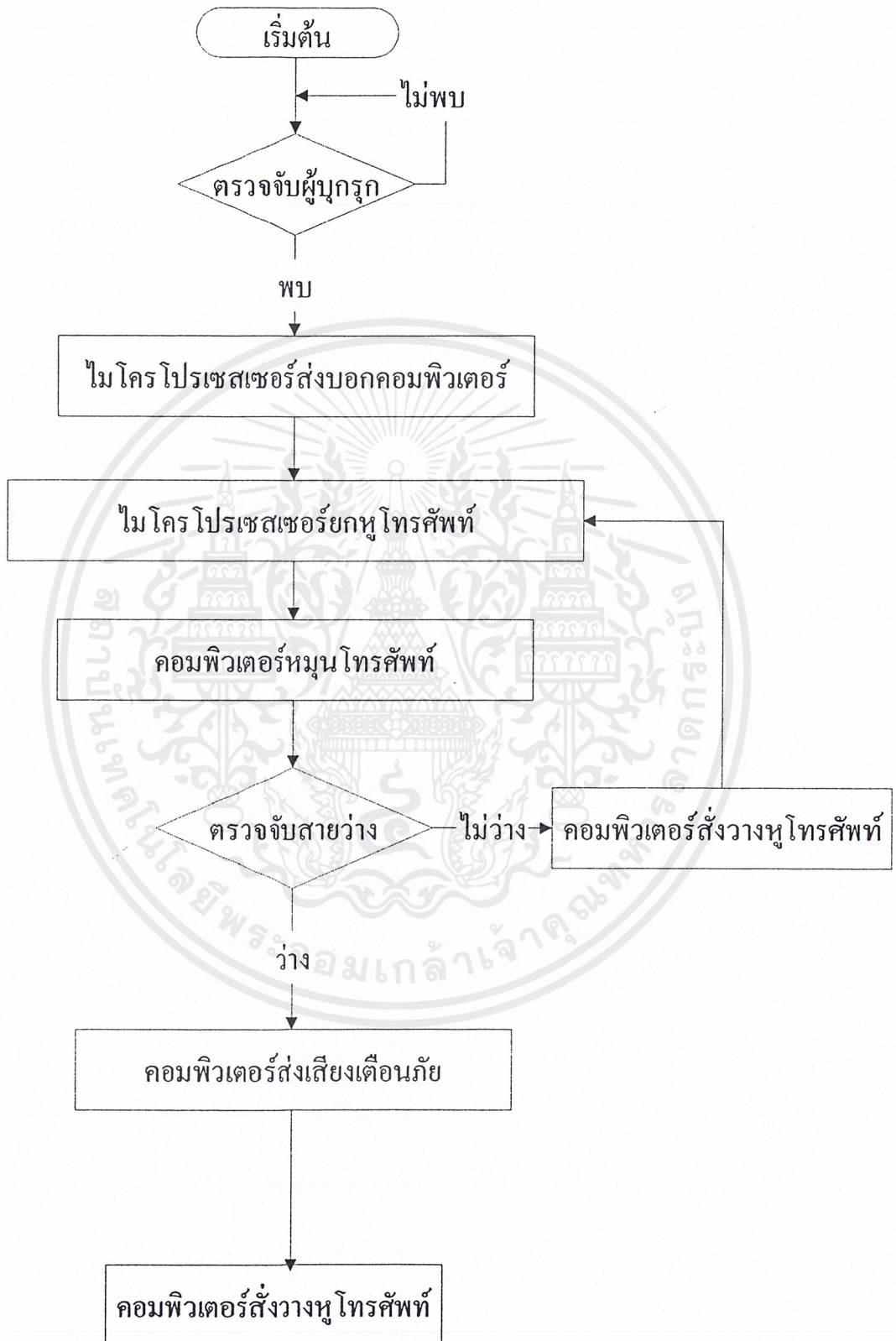
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



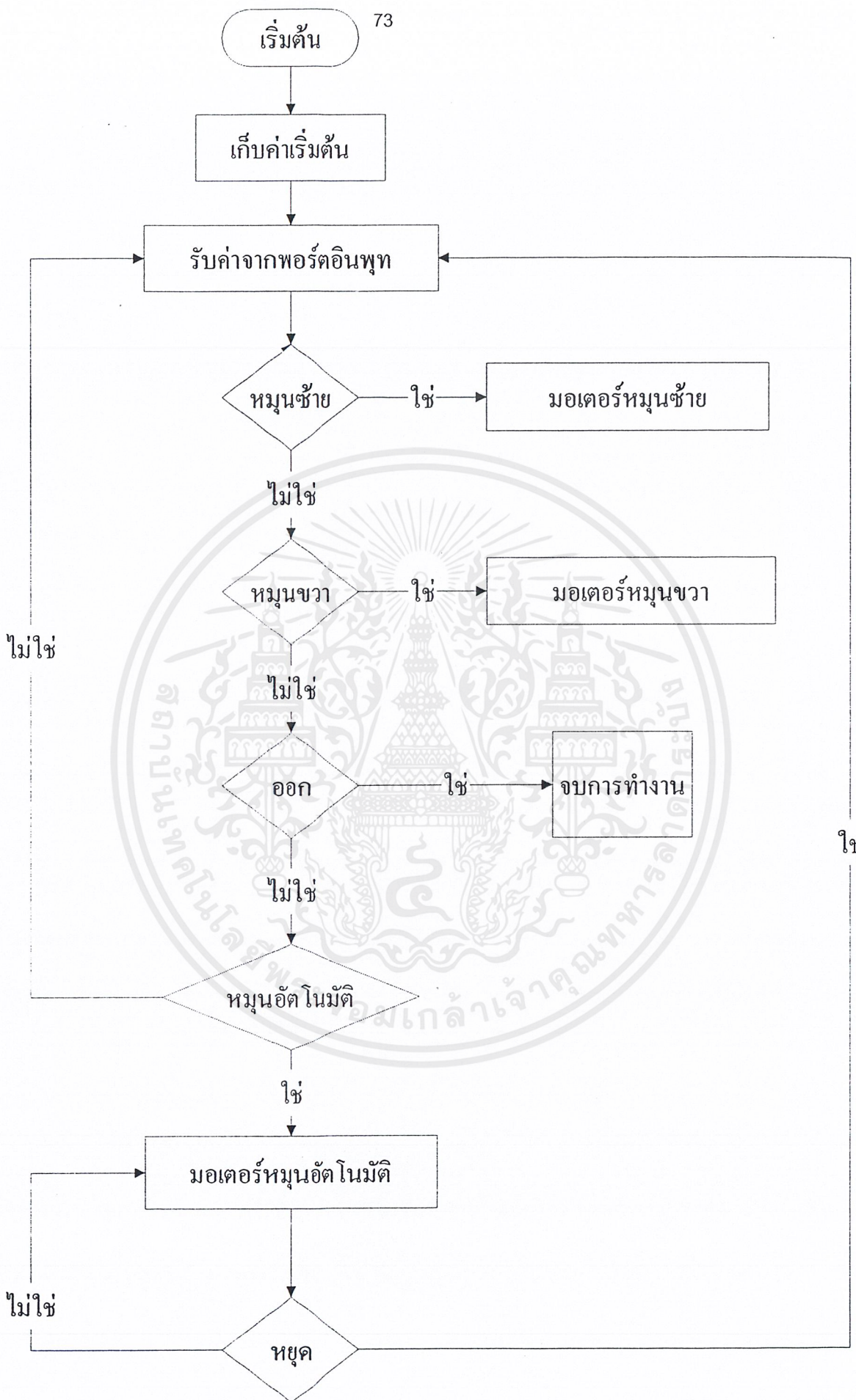
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังขอสงวนสิทธิ์ในเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.23 โฟลว์ชาร์ตของโปรแกรมวิซวลเบสิก

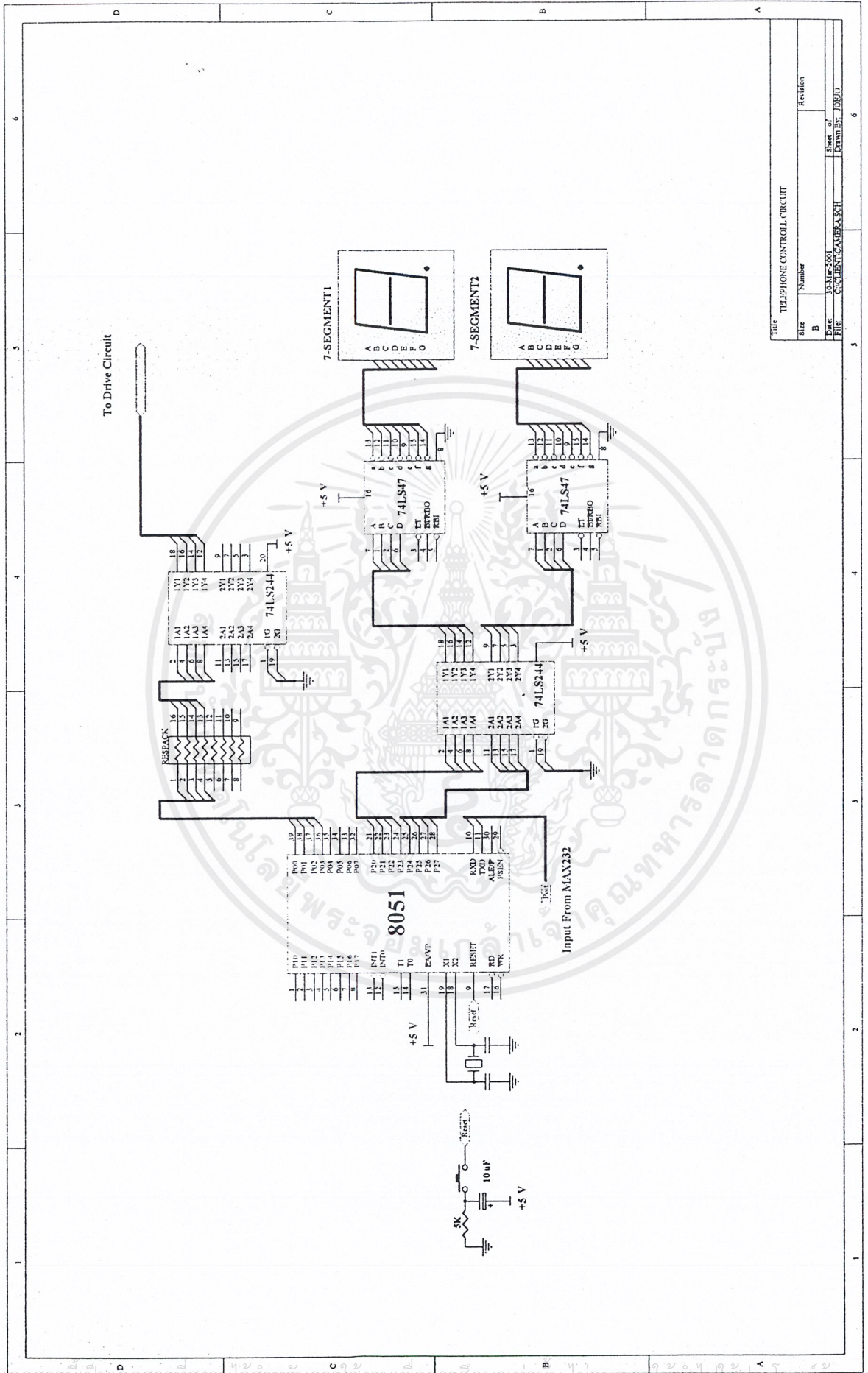
3.2.2 โฟลว์ชาร์ตส่วนของโปรแกรมภาษาแอสเซมบลี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.24 โฟลว์ชาร์ตการสั่งงานโทรศัพท์เตือนภัย
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



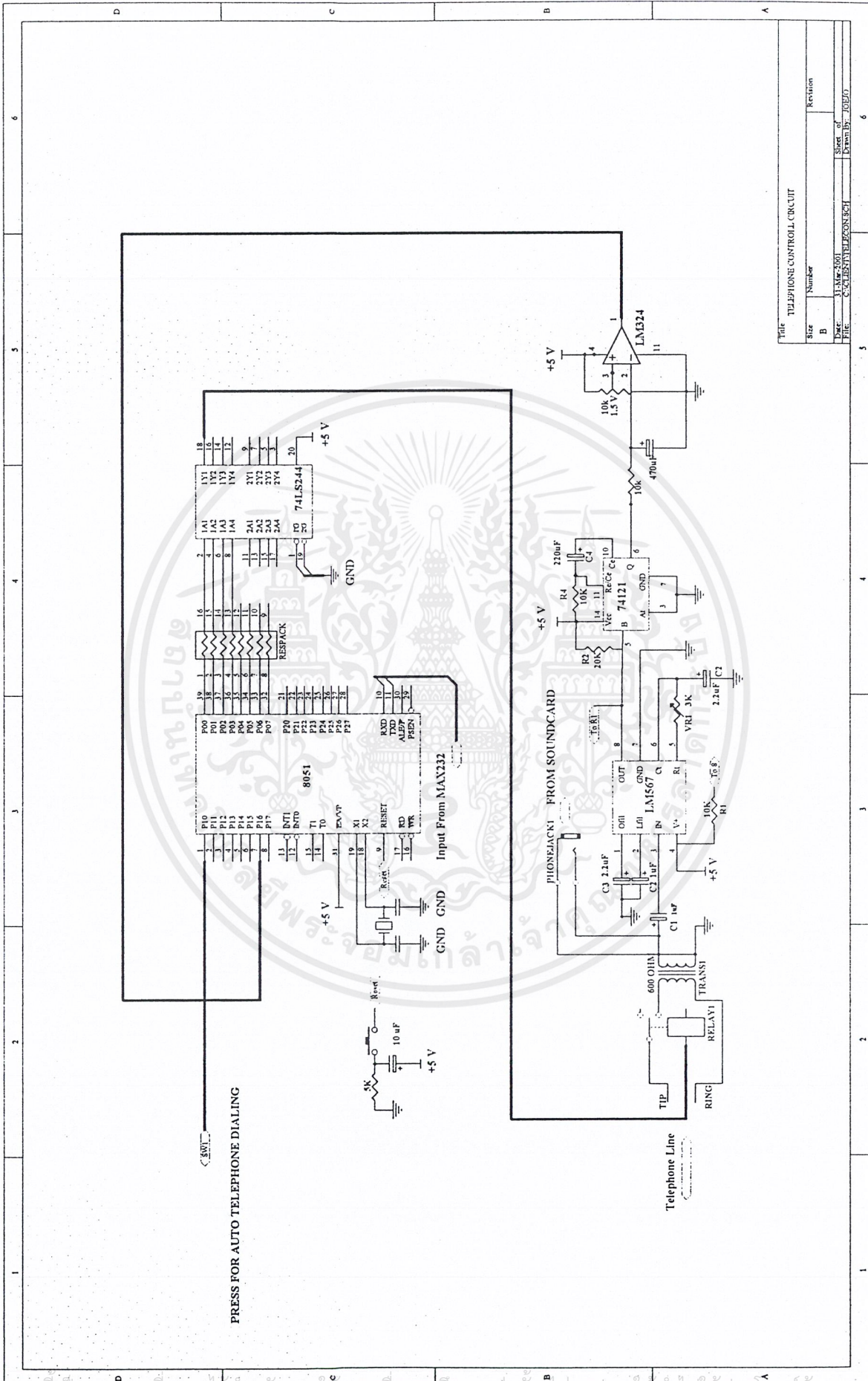
เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title			TELEPHONE CONTROL CIRCUIT		
Size	Number		Revision		
B					
Date	30-Mar-2001		Sheet of		
File	C:\MICRO\LABEASSET1		Drawn By: DDEBU		

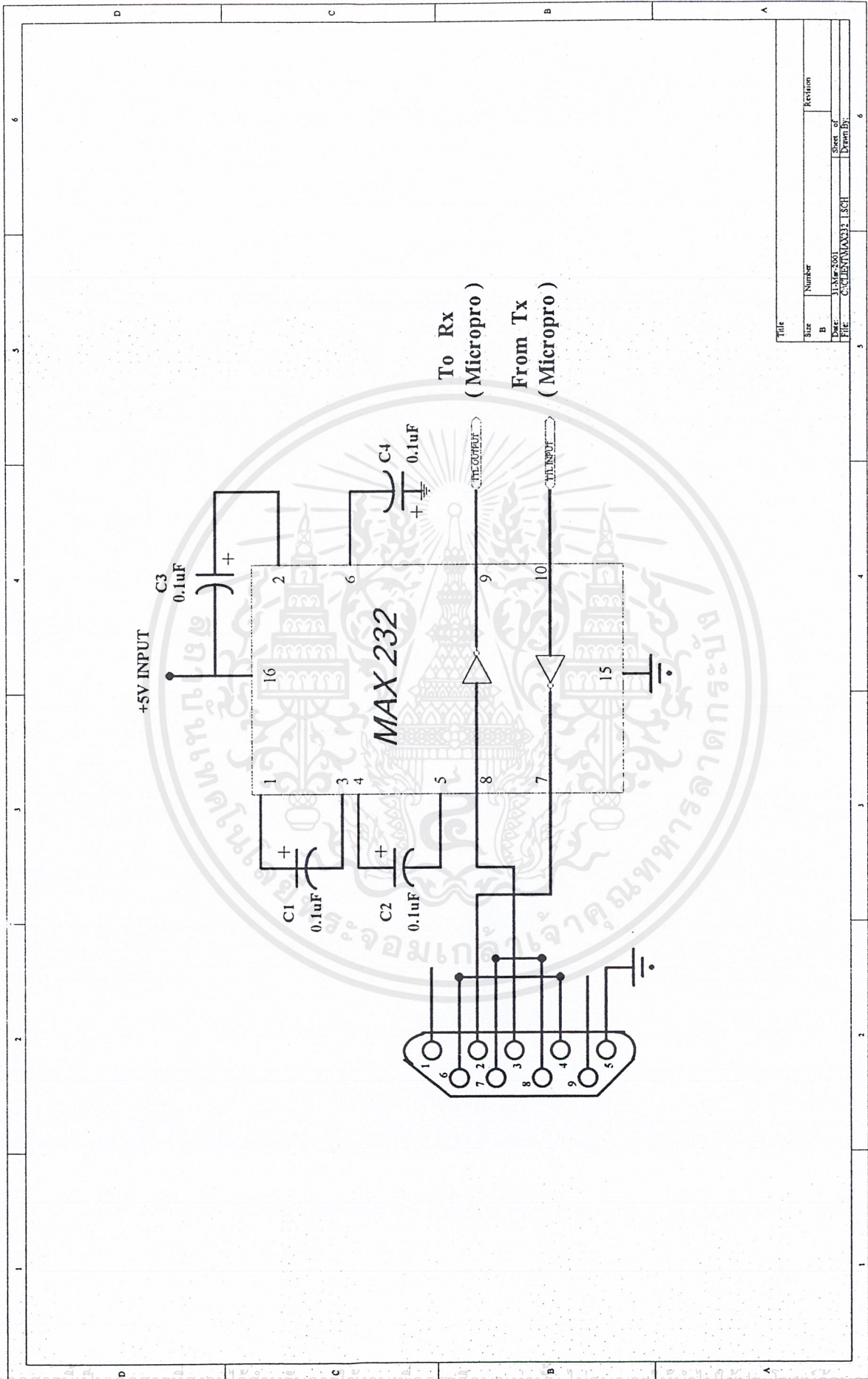
รูปที่ 3.26 แสดงรูปวงจรส่วนควบคุมการหมุนกลิ้ง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ที่นอกเหนือจากนี้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



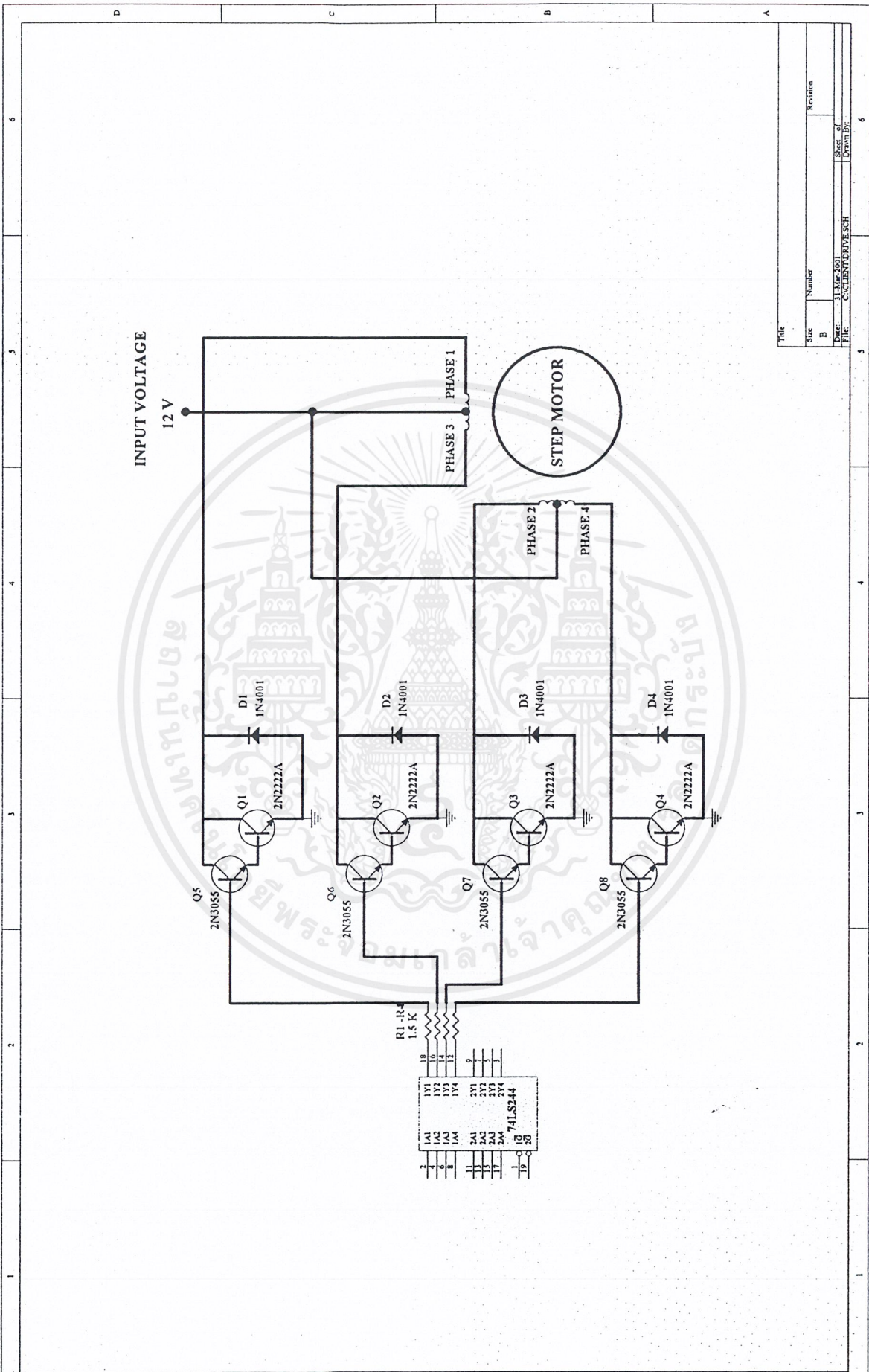
Title		TELEPHONE CONTROL CIRCUIT	
Size	Number	Revision	
B			
Doc. No.	CU-ENG-2001	Sheet of	6
File	C:\SUBSYSTEM\TELEPHONE	Drawn by	JOBFO

รูปที่ 3.27 แสดงรูปวงจรส่วนการโทรศัพท์ที่เตือนภัย



Title		Revision	
Size	Number		
B	1	Date	13 Mar 2001
File	C:\CLIENT\MAX321.1SCH	Sheet of	6
		Drawn By:	

รูปที่ 3.28 แสดงรูปวงจรแปลงระดับแรงดัน



Title	
Size	Number
B	
Date:	31 Mar 2001
FILE:	C:\CLIENT\KIVE\SCH
Sheet of	6
Drawn By:	

รูปที่ 3.29 แสดงรูปวงจรขับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

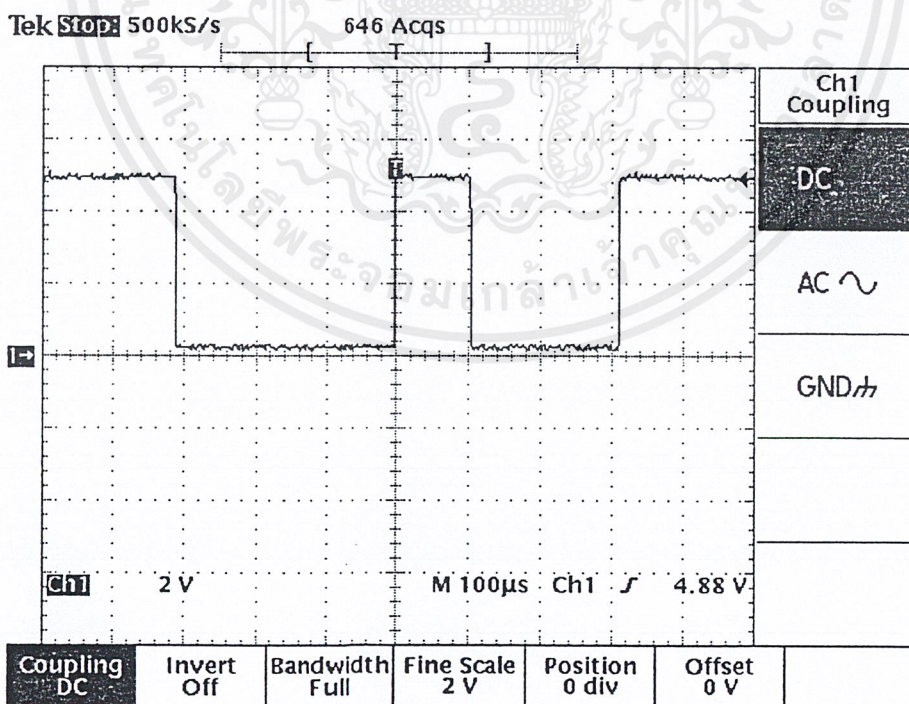
การทดสอบและผลการทดสอบ

4.1 การทดสอบการส่งค่าผ่านพอร์ตอนุกรม

เพื่อสังเกตสัญญาณที่ออกจากพอร์ตอนุกรม และที่ขา RXD ของไมโครโปรเซสเซอร์ดู ลักษณะสัญญาณรูปคลื่นเมื่อใช้คำสั่งจากคอมพิวเตอร์ดังต่อไปนี้

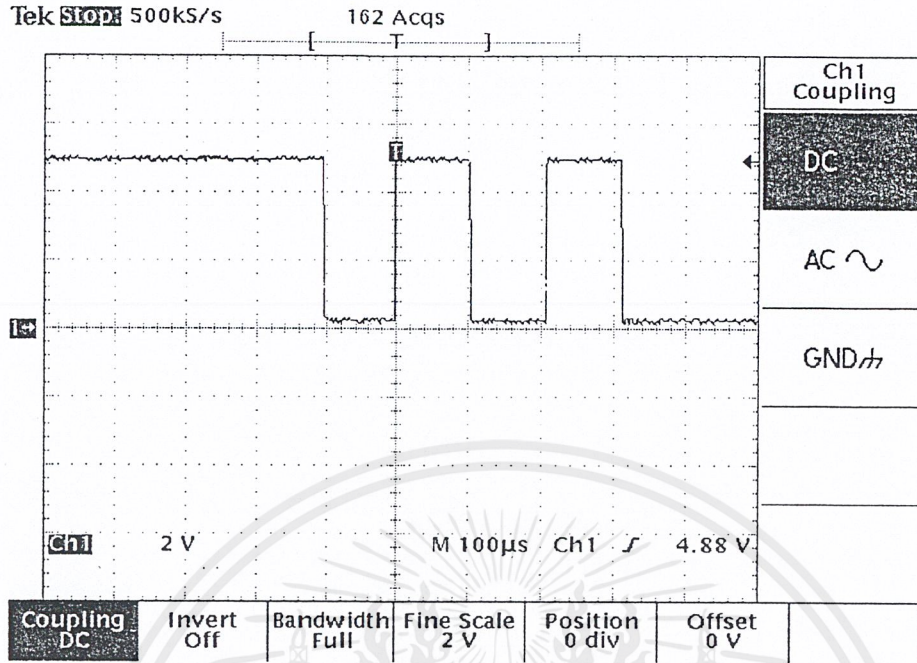
1. ใช้คำสั่ง Exit ส่งออกทางพอร์ตอนุกรม com1
2. ใช้คำสั่ง Left ส่งออกทางพอร์ตอนุกรม com1
3. ใช้คำสั่ง Right ส่งออกทางพอร์ตอนุกรม com1
4. ใช้คำสั่ง Night ส่งออกทางพอร์ตอนุกรม com1
5. ใช้คำสั่ง Stop ส่งออกทางพอร์ตอนุกรม com1

รูปสัญญาณที่วัดได้เป็นดังต่อไปนี้

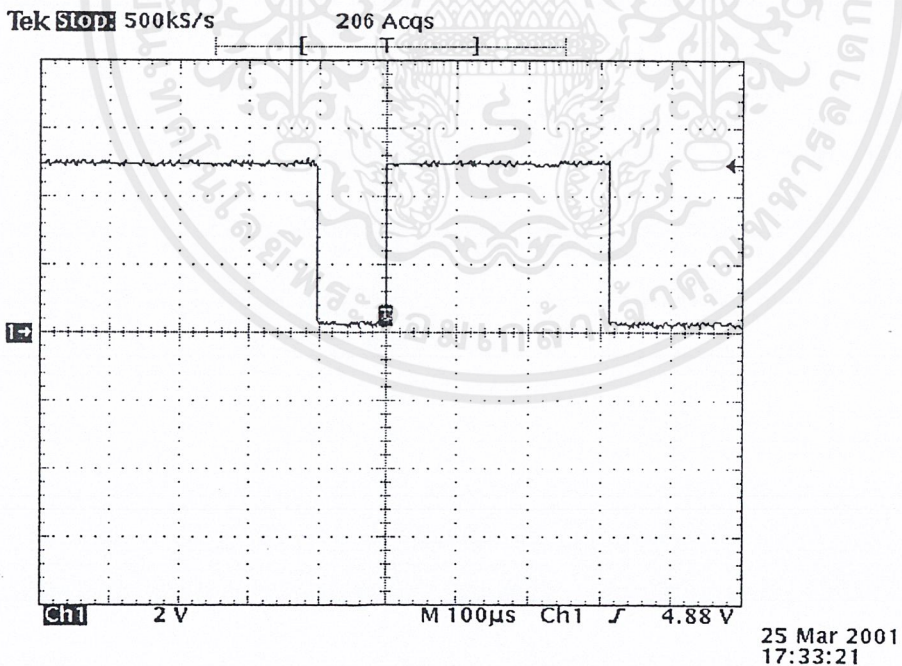


รูปที่ 4.1 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Exit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

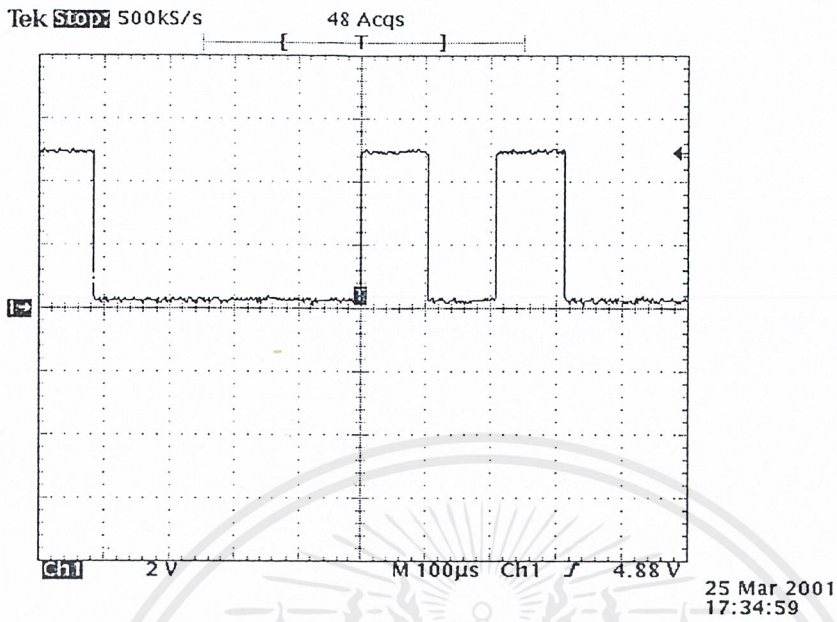


รูปที่ 4.2 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Left

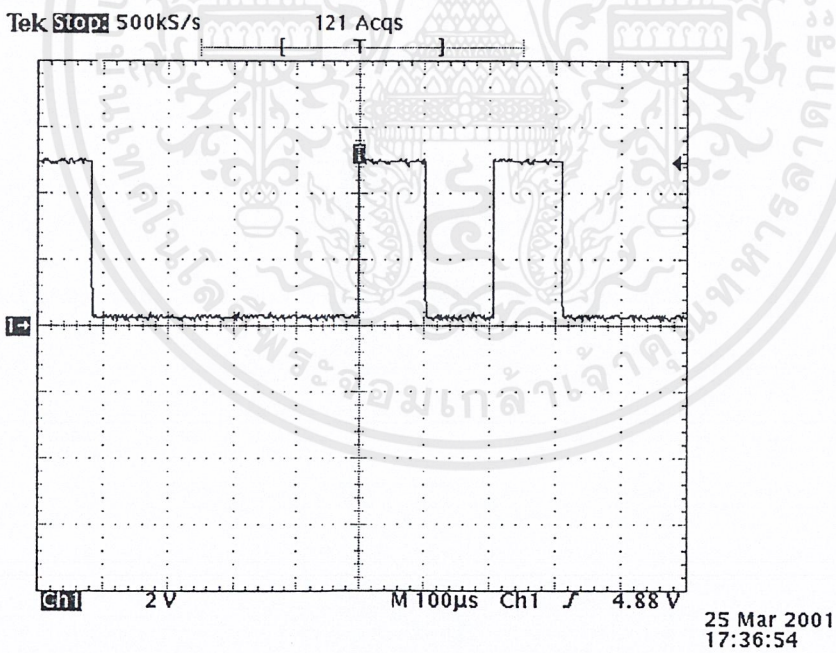


รูปที่ 4.3 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Right

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Night



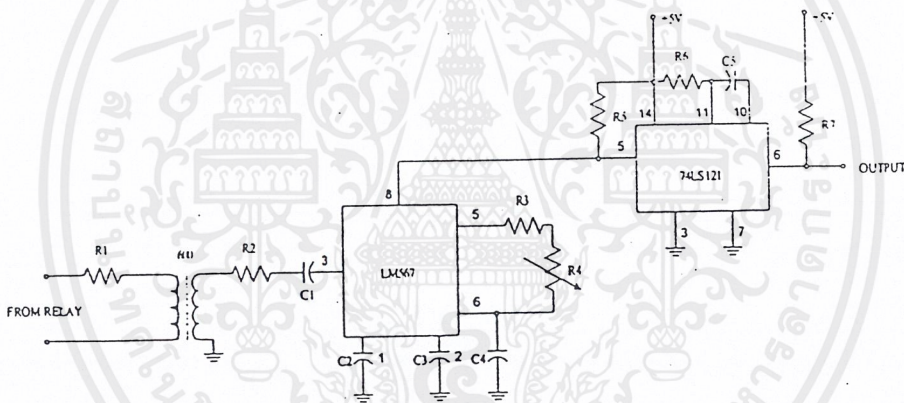
รูปที่ 4.5 แสดงรูปสัญญาณจากพอร์ตอนุกรมของคำสั่ง Stop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

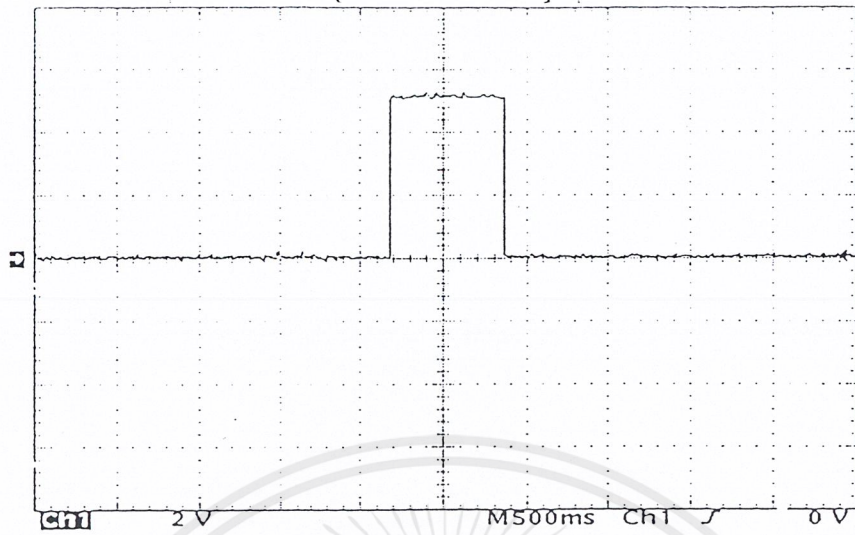
4.2 ผลการทดสอบวงจรตรวจจับสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง

ใช้วงจรในรูปที่ 4.6 เมื่อมีสัญญาณเรียกกลับเข้ามาแล้วสัญญาณที่ขา 8 ของไอซี 74121 ทำการหน่วงเวลาไปอีก 1.55 วินาที ดังแสดงในรูปที่ 4.7 ซึ่งจะเป็นได้ว่าระดับสัญญาณอินพุทของ LM567 ที่เกิดการผิดเพี้ยน ทำให้ไอซี LM567 ไม่สามารถตรวจจับได้เต็มช่วงเวลาของสัญญาณเรียกกลับ ดังนั้นการใช้วงจรหน่วงเวลาเข้ามาช่วย จะทำให้สามารถกำหนดช่วงเวลาที่แน่นอนของสัญญาณเรียกกลับได้

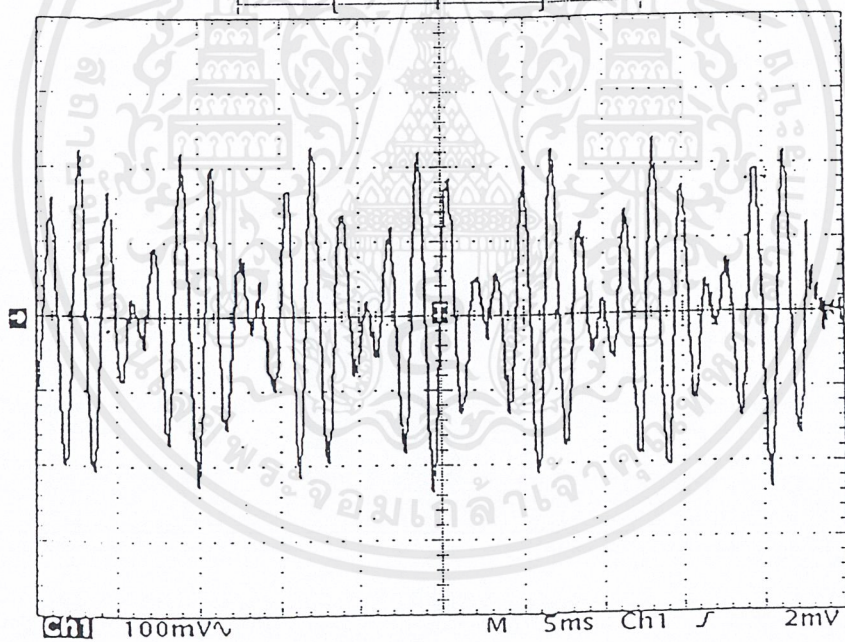
ส่วนสัญญาณสายไม่ว่างที่เข้ามาในวงจร จะมีลักษณะอินพุทและเอาต์พุตดังแสดงในรูปที่ 4.8 และ 4.9 ซึ่งหลังจากที่ผ่านไอซี 74121 แล้ว จะสามารถแยกความแตกต่างระหว่างสัญญาณเรียกกลับและสัญญาณสายไม่ว่างได้ โดยการผ่านวงจรกรองแรงดันและวงจรเปรียบเทียบแรงดัน



รูปที่ 4.6 วงจรตรวจจับสัญญาณเรียกกลับและสัญญาณสายไม่ว่าง

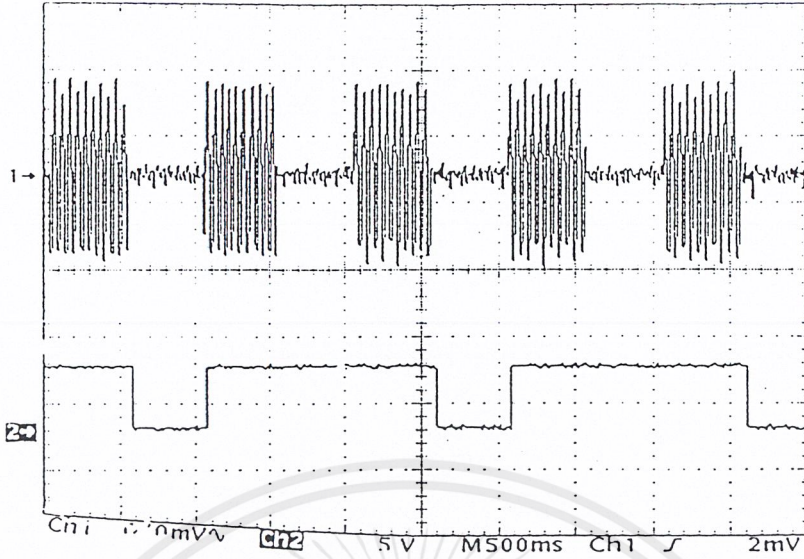


รูปที่ 4.7 แสดงสัญญาณที่ขา 6 เอ้าท์พุทของ 74121 ของสัญญาณเรียกกลับ
 10 Acqs



รูปที่ 4.8 แสดงสัญญาณสายไม่ว่างที่วัดได้จากอินพุทขา 3 ของ LM567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงการเปรียบเทียบสัญญาณสายไม่ว่างอินพุทกับเอาต์พุทของ LM567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์

จากการทำโครงการชิ้นนี้ พบว่าเกิดปัญหาหลายประการทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์
ปัญหาที่เกิดขึ้นในส่วนที่เป็นฮาร์ดแวร์

1. อุปกรณ์ที่ใช้ในวงจรมักจะเล็บบ่อย เช่น 7- SEGMENT , LED
2. มอเตอร์ เป็นสแต็ปเปอร์มอเตอร์ที่ใช้เป็นตัวขับเคลื่อนมีกำลังไม่มากพอที่จะใช้ขับเคลื่อนและวงจรขับเคลื่อนต้องจ่ายให้มากพอที่มอเตอร์ต้องการ
3. การตรวจจับสัญญาณโทรศัพท์ทำได้ยากเพราะเมื่อเปลี่ยน PABX ย่อยทำให้ความถี่ของสัญญาณเปลี่ยน ทำให้ยากมากต่อการตรวจจับ
4. การเชื่อมต่อระหว่างวงจรรย่อยซึ่งแยกกันอยู่ด้วยอุปกรณ์เชื่อมต่อเช่น CONNECTOR อาจจะหลวม ไม่แน่น ตรวจสอบให้แน่ใจ
5. แผ่นวงจรที่ใช้แผ่นอนุกรมประตงค์มีการขาดของสายหรือ short กับขาอื่น

ปัญหาที่เกิดขึ้นในส่วนที่เป็นซอฟต์แวร์ เกี่ยวกับการหนดเวลาต้องให้เหมาะสมเพื่อให้มอเตอร์ทำงานได้ และการเขียนโปรแกรมที่ยาวๆ จะทำให้การประมวลผลทำได้ช้า ต้องศึกษาการเขียนโปรแกรมให้ต้องแท้ การเขียนโปรแกรมลงบนชิปไอซีบ่อยๆ การถอดเข้าออกอาจทำให้ขาไอซีหักได้ต้องระมัดระวัง

แนวทางในการพัฒนาโครงการ

ขอบเขตการทำงานของโครงการนี้คือ

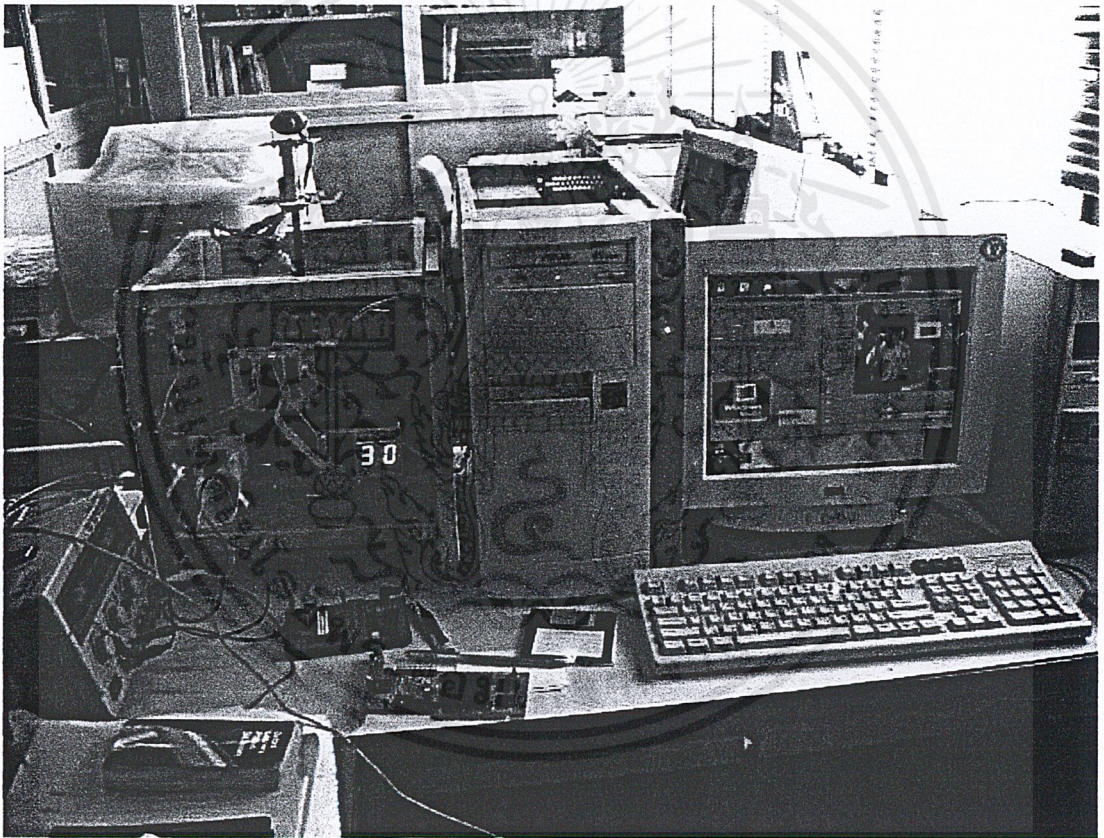
1. สามารถควบคุมการหมุนของกลิ้งได้โดยการตั้งงานทางคอมพิวเตอร์ โดยมีคำสั่งคือให้หมุน ไปด้านซ้าย หมุนไปด้านขวาและหมุนอัตโนมัติ
2. สามารถใช้เตือนภัยทางโทรศัพท์โดยอัตโนมัติได้ โดยการตั้งงานทางคอมพิวเตอร์ ในโครงการนี้สามารถตรวจจับสายว่างและสายไม่ว่างของโทรศัพท์ได้ แต่ไม่สามารถตรวจจับการยกหูโทรศัพท์ของคู่สายปลายทางได้

สำหรับการพัฒนาโครงการนี้อาจใช้การตั้งงานและตั้งภาพมายังเข้าของอาคารผ่านเครือข่ายอินเทอร์เน็ต มีการตรวจจับการรับสายของคู่สายปลายทางเพื่อให้แน่ใจว่าข้อความได้ส่งถึงคู่สายปลายทางแล้ว หรืออาจมีการเตือนภัยด้วยวิทยุติดตามตัว เตือนภัยเมื่อเกิดเหตุไฟไหม้ หรือมีการตรวจสอบความเรียบร้อยของอาคารผ่านทางโทรศัพท์หรือเครือข่ายอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

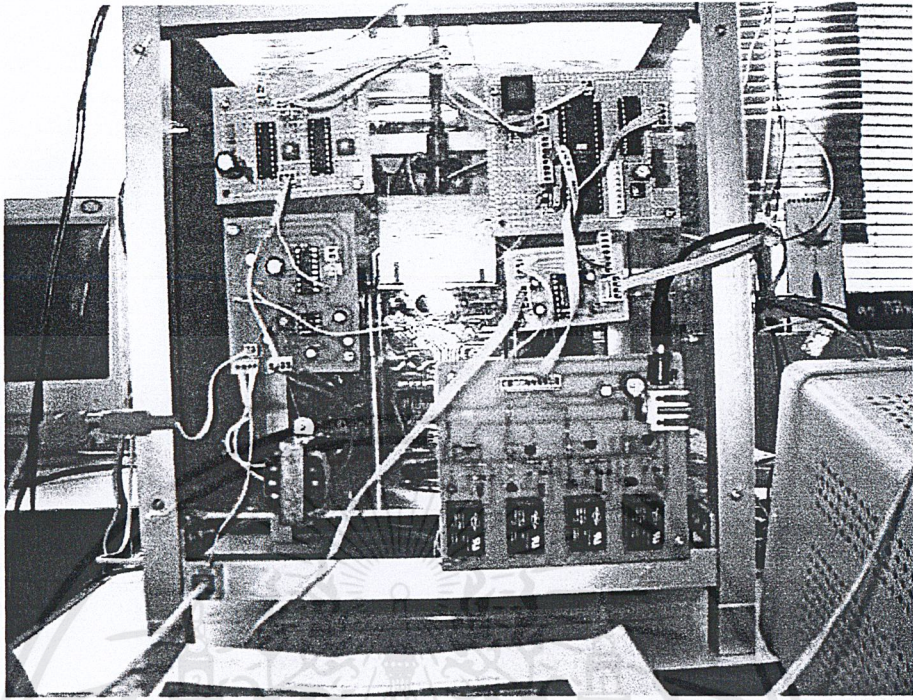
สรุปและวิจารณ์

กล่าวโดยสรุป โครงการนี้ต้องใช้ความรู้ด้านฮาร์ดแวร์ทั้งการเขียนโปรแกรมวิซวลเบสิก และ การเขียนภาษาแอสเซมบลี รวมทั้งความรู้ด้านฮาร์ดแวร์ต้องมีการศึกษาทดลอง โดยรวมโครงการชิ้นนี้สำเร็จในระดับที่น่าพอใจในระดับหนึ่ง และยังมีหลายส่วนที่ยังต้องเพิ่มเติมเพื่อนำไปใช้งานได้จริง

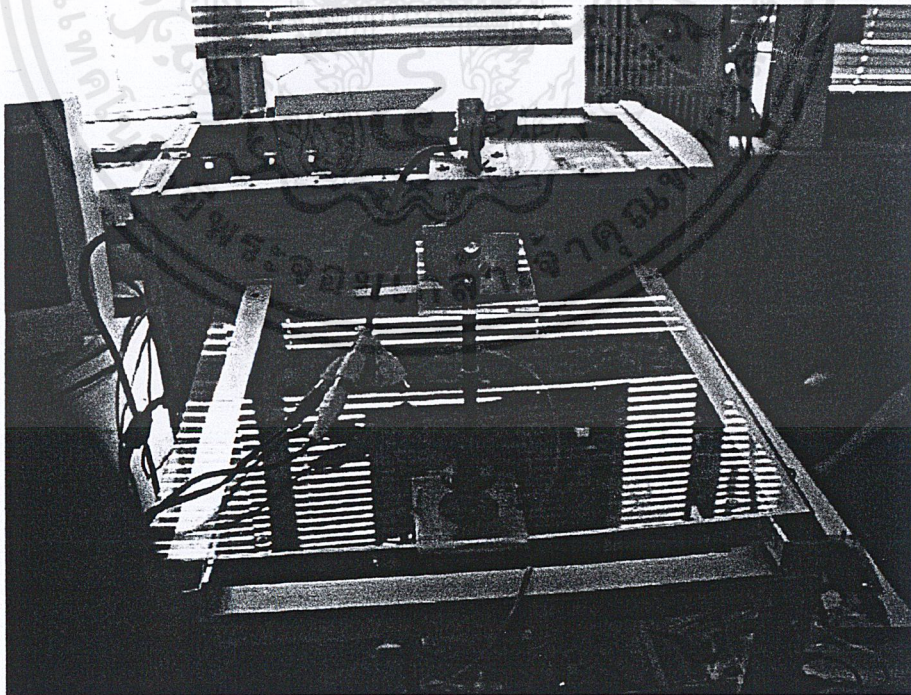


รูปที่ 5.1 แสดงรูปโครงการขณะใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แสดงรูปวงจรด้านข้าง



รูปที่ 5.3 แสดงรูปโครงงานด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. รศ.สมยศ จุณณะปิยะ , การประยุกต์ใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51, พิมพ์ครั้งที่ 2 พ.ศ. 2541
2. *Hobby Electronics* , บริษัทซีเอ็ดยูเคชั่น จำกัด.
3. วันสุระ ศรีใสดี , ไมโครคอนโทรลเลอร์ภาคปฏิบัติ MCS-51 , พิมพ์ครั้งแรก พ.ศ. 2542
4. *Microcomputer* , เรื่อง *Micosoft Access* กับการติดต่อ *Serial Port* , หน้าที่ 159-163 เดือน พฤษภาคม 2542
5. คู่มือเทียบเบอร์ไอซี TTL , บริษัทซีเอ็ดยูเคชั่น จำกัด.
6. จิระวัฒน์ ใจอ่อนนุ่ม - นภัทร วัจนเทพินทร์ , วงจรดิจิทัล 1 ลอจิกเชิงจัดหมู่ (*Digital circuit Combination Logic*), บริษัท สกายบุ๊กส์ จำกัด.
7. ชีรวัดน์ ประกอบผล , การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ , สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)
8. กิตติ ภัคดีวัฒนกุล – จำลองครูอุตสาหะ , *Visual Basic 6 ฉบับโปรแกรมเมอร์* , บริษัท เคทีพี แอนด์ คอนซัลท์ จำกัด
9. *National Semiconductor DATA CONVERSION/ACQUISTION DATABOOK* , 1993.
10. *National Operational Amplifiers Databook*, National Semiconductor , 1995.



ภาคผนวก ก

โปรแกรมวิชาพลเบติค และแอสเซมบลี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาษา Visual Basic ของ ฟอรัม Computer Control Video Position

Option Explicit

Dim I As Integer

Dim back As Boolean

Dim ch, RUN

Dim TEMP

Private Sub command1_Click()

MSComm1.Output = "d"

MSComm1.Output = Chr(0)

Me.Text1.Text = "EXIT"

End Sub

Private Sub Form_Load()

RUN = 1

Me.MSComm1.PortOpen = True

Me.Text1.Text = "HELLO"

End Sub

Private Sub MSComm1_OnComm()

Select Case MSComm1.CommEvent

Case comEvReceive

TEMP = MSComm1.Input

If TEMP = "A" Then Timer1.Interval = 250

If TEMP = "B" Then

mm.FileName = "c:\wave\wave4.wav"

mm.Play

Timer2.Interval = 5000

End If

End Select

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub night_Click()
    MSComm1.Output = "("
    MSComm1.Output = Chr(0)
    command1.Enabled = False
    leff.Enabled = False
    rit.Enabled = False
    Text1.Text = "NIGHT"
    If live.Enabled = True Then
        live.Enabled = False
        night.Caption = "STOP"
    Else
        MSComm1.Output = "b"
        Me.Text1.Text = "STOP"
        command1.Enabled = True
        leff.Enabled = True
        rit.Enabled = True
        live.Enabled = True
        night.Caption = "NIGHT MODE"
    End If
End Sub

```

```

Private Sub leff_Click()
    MSComm1.Output = "%"
    MSComm1.Output = Chr(0)
    Text1.Text = "LEFT"
End Sub

```

```

Private Sub live_Click()
    Me.Text1.Text = "LIVE"
End Sub

```

```
Private Sub rit_Click()
MSComm1.Output = ""
    MSComm1.Output = Chr(0)
    Text1.Text = "RIGHT"
End Sub
```

```
Private Sub Timer1_Timer()
    ch = Mid(T.Text, RUN, 1)
    Choose
If RUN > Len(T.Text) Then
Timer1.Interval = 0
RUN = 0
Timer2.Interval = 15000
End If
RUN = RUN + 1
End Sub
Sub Choose()
Select Case ch
    Case "0"
        mm.FileName = "c:\wave\DTMF0.wav"
        mm.Play
    Case "1"
        mm.FileName = "c:\wave\DTMF1.wav"
        mm.Play
    Case "2"
        mm.FileName = "c:\wave\DTMF2.wav"
        mm.Play
    Case "3"
        mm.FileName = "c:\wave\DTMF3.wav"
        mm.Play
    Case "4"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mm.FileName = "c:\wave\DTMF4.wav"  
mm.Play  
Case "5"  
mm.FileName = "c:\wave\DTMF5.wav"  
mm.Play  
Case "6"  
mm.FileName = "c:\wave\DTMF6.wav"  
mm.Play  
Case "7"  
mm.FileName = "c:\wave\DTMF7.wav"  
mm.Play  
Case "8"  
mm.FileName = "c:\wave\DTMF8.wav"  
mm.Play  
Case "9"  
mm.FileName = "c:\wave\DTMF9.wav"  
mm.Play  
End Select  
End Sub  
  
Private Sub Timer2_Timer()  
Timer2.Interval = 0  
MSComm1.Output = TEMP  
Text1.Text = TEMP  
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาษา Visual Basic ของ ฟอรัม Telephone Warning

```
Private Sub cbDriver_Click()  
    Dim oldDriver As Long  
    oldDriver = ezVidCap1.DriverIndex  
  
    On Error Resume Next  
    ezVidCap1.DriverIndex = cbDriver.ListIndex  
    If Err Then  
        'restore old settings  
        ezVidCap1.DriverIndex = oldDriver  
        cbDriver.ListIndex = oldDriver  
        lblStatusString = "Could not connect!"  
    End If  
End Sub  
  
Private Sub chkAudio_Click()  
    If chkAudio.Value = 1 Then  
        ezVidCap1.CaptureAudio = True  
    Else  
        ezVidCap1.CaptureAudio = False  
    End If  
End Sub  
  
Private Sub chkAutoSize_Click()  
    If chkAutoSize.Value = 1 Then  
        ezVidCap1.AutoSize = True  
    Else  
        ezVidCap1.AutoSize = False  
    End If  
End Sub  
  
Private Sub chkCenter_Click()  
    If chkCenter.Value = 1 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ezVidCap1.CenterVideo = True
Else
ezVidCap1.CenterVideo = False
End If
End Sub
```

```
Private Sub chkFrameCallback_Click()
If chkFrameCallback.Value = 1 Then
ezVidCap1.FrameEventEnabled = True
Else
ezVidCap1.FrameEventEnabled = False
End If
End Sub
```

```
Private Sub chkPreRoll_Click()
If chkPreRoll.Value = 1 Then
ezVidCap1.UsePreciseCaptureControls = True
Else
ezVidCap1.UsePreciseCaptureControls = False
End If
End Sub
```

```
Private Sub chkPreview_Click()
If chkPreview.Value = 1 Then
ezVidCap1.Preview = True
Else
ezVidCap1.Preview = False
End If
End Sub
```

```
Private Sub chkStretch_Click()
If chkStretch.Value = 1 Then
ezVidCap1.StretchPreview = True
Else
ezVidCap1.StretchPreview = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
End Sub

Private Sub chkUserConfirm_Click()
    If chkUserConfirm.Value = 1 Then
        ezVidCap1.MakeUserConfirmCapture = True
    Else
        ezVidCap1.MakeUserConfirmCapture = False
    End If
End Sub
```

```
Private Sub cmdAudioDlg_Click()
    'From Beta2 the syntax has changed here
    'ezVidCap1.ShowDlgAudioFormat = True
    ezVidCap1.ShowDlgAudioFormat
End Sub
```

```
Private Sub cmdCapture_Click()
    Call ezVidCap1.CaptureVideo
End Sub
```

```
Private Sub cmdCompDlg_Click()
    'From Beta2 the syntax has changed here
    'ezVidCap1.ShowDlgCompressionOptions = True
    ezVidCap1.ShowDlgCompressionOptions
End Sub
```

```
Private Sub cmdDisplayDlg_Click()
    'From Beta2 the syntax has changed here
    'ezVidCap1.ShowDlgVideoDisplay = True
    ezVidCap1.ShowDlgVideoDisplay
End Sub
```

```
Private Sub cmdFormatDlg_Click()
    'From Beta2 the syntax has changed here
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'ezVidCap1.ShowDlgVideoFormat = True
ezVidCap1.ShowDlgVideoFormat
End Sub

Private Sub cmdHeight_Click()
ezVidCap1.Height = txtHeight.Text
'show actual size (in case auto size is turned on)
txtHeight.Text = ezVidCap1.Height
End Sub

```

```

Private Sub cmdSaveAs_Click()
Dim filename As String
If mCmnDlg.VBGetSaveFileNamePreview(filename, _
FileMustExist:=False, _
filter:="AVI files (*.avi)|*.avi", _
InitDir:=App.Path, _
DlgTitle:="Save AVI File", _
DefaultExt:="avi", _
Owner:=Me.hWnd) _
Then
On Error Resume Next
Call ezVidCap1.SaveAs(filename)
If Err Then
MsgBox Err.Description, vbInformation, App.Title
End If
End If

```

```
End Sub
```

```

Private Sub cmdSaveDIB_Click()
Dim filename As String
Dim run
' If mCmnDlg.VBGetSaveFileName(filename, _
' filter:="Bitmap files (*.bmp)|*.bmp", _
' InitDir:=App.Path, _

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'           DlgTitle:="Save Frame As Bitmap File", _
'           DefaultExt:="bmp", _
'           Owner:=Me.hWnd) _
'           Then
'       On Error Resume Next
run = Mid(Time$, 1, 2) & Mid(Time$, 4, 2)
filename = "c:" & run & ".bmp"
Call ezVidCap1.SaveDIB(filename)
'       If Err Then
'           MsgBox Err.Description, vbInformation, App.Title
'       End If
'   End If
End Sub

Private Sub cmdSourceDlg_Click()
'From Beta2 the syntax has changed here
ezVidCap1.ShowDlgVideoSource = True
ezVidCap1.ShowDlgVideoSource
End Sub

Private Sub cmdWidth_Click()
ezVidCap1.Width = txtWidth.Text
'show actual size (in case auto size is turned on)
txtWidth.Text = ezVidCap1.Width
End Sub

Private Sub ezVidCap1_StatusClear()
lblStatusCode.Caption = ""
lblStatusString.Caption = ""
End Sub

Private Sub ezVidCap1_CaptureYield()
'Setting Yield = True will allow this event to be generated
'but will slow down performance
Debug.Print "yield"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DoEvents
End Sub

Private Sub ezVidCap1_ErrorMessage(ByVal ErrCode As Long, ByVal ErrString As
String)

    If ErrCode <> 0 Then
        'Debug.Print ErrString
        lblStatusString = "Error " & ErrString
        lblStatusString.Refresh
    End If
End Sub

Private Sub ezVidCap1_FrameCallback(ByVal lpVHdr As Long)
Debug.Print "Video frame: " & lpVHdr

Call MessWithVidBits(lpVHdr)
End Sub

Private Sub ezVidCap1_PreRollComplete()
Dim userRet As Long

userRet = MsgBox("Using precise capture controls." & vbCrLf & _
    "PreRoll complete - Click OK to start capture immediately." & _
    , vbOKCancel, App.Title)

If userRet = vbOK Then
    ezVidCap1.PreciseCaptureStart
Else
    ezVidCap1.PreciseCaptureCancel
End If
End Sub

Private Sub ezVidCap1_StatusMessage(ByVal StatCode As Long, ByVal StatString
As String)

    lblStatusCode.Caption = "StatusCode: " & StatCode

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lblStatusCode.Refresh
If StatCode <> 0 Then
    'Debug.Print StatString
    lblStatusString.Caption = StatString
    lblStatusString.Refresh
End If
End Sub

Private Sub EnableButtons()
    cmdAudioDlg.Enabled = False
    cmdFormatDlg.Enabled = False
    cmdDisplayDlg.Enabled = False
    cmdSourceDlg.Enabled = False
    cmdCapture.Enabled = False
    cmdCompDlg.Enabled = False

    With ezVidCap1
        If .NumCapDevs > 0 Then
            cmdCapture.Enabled = True
            cmdCompDlg.Enabled = True
        End If
        If .HasAudio Then cmdAudioDlg.Enabled = True
        If .HasDlgFormat Then cmdFormatDlg.Enabled = True
        If .HasDlgDisplay Then cmdDisplayDlg.Enabled = True
        If .HasDlgSource Then cmdSourceDlg.Enabled = True
    End With
End Sub

Private Sub ezVidCap1_VideoStreamCallback(ByVal lpVHdr As Long)
    Debug.Print "Video stream: " & lpVHdr
End Sub

Private Sub ezVidCap1_WaveStreamCallback(ByVal lpWHdr As Long)
    Debug.Print "Wave stream: " & lpWHdr
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Load()
```

```
'THE FOLLOWING 2 LINES ARE UNNECESSARY AFTER BETA2
```

```
'Me.Show 'control will not connect to capdevice until it is shown
```

```
' 'it must be initialized by being shown before you can read some of the
```

properties

```
'DoEvents 'allows driver to connect
```

```
Dim i As Long
```

```
Call EnableButtons 'check device caps and enable appropriate btns
```

```
Me.Show 'show form
```

```
Me.Refresh
```

```
If 0 < ezVidCap1.NumCapDevs Then
```

```
For i = 0 To ezVidCap1.NumCapDevs - 1
```

```
cbDriver.AddItem (ezVidCap1.GetDriverName(i))
```

```
Next
```

```
cbDriver.ListIndex = ezVidCap1.DriverIndex
```

```
Else
```

```
cbDriver.AddItem ("<none>")
```

```
cbDriver.ListIndex = 0
```

```
MsgBox "No Video Capture Device!", vbInformation, App.Title
```

```
End If
```

```
'init form with current properties
```

```
lblStatusCode = "Status Panel"
```

```
lblStatusString = ezVidCap1.GetDriverVersion()
```

```
txtWidth = ezVidCap1.Width
```

```
txtHeight = ezVidCap1.Height
```

```
chkAutoSize.Value = -(ezVidCap1.AutoSize)
```

```
chkCenter.Value = -(ezVidCap1.CenterVideo)
```

```
chkStretch.Value = -(ezVidCap1.StretchPreview)
```

```
chkAudio.Value = -(ezVidCap1.CaptureAudio)
```

```
chkPreRoll.Value = -(ezVidCap1.UsePreciseCaptureControls)
```

```
chkUserConfirm.Value = -(ezVidCap1.MakeUserConfirmCapture)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
chkPreview.Value = -(ezVidCap1.Preview)
chkFrameCallback.Value = -(ezVidCap1.FrameEventEnabled)
End Sub
```

```
Private Sub Form_Resize()
'this is just to provide a nice status bar with no control
With picStatus
    lblStatusCode.Move 0, 0, .Width * 0.25, .Height
    lblStatusString.Move .Width * 0.25, 0, .Width * 0.75, .Height
End With
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาษาแอสเซมบลี

; Computer Control Video Position

```
RUN EQU 30H
CARRY EQU 31H
ORG 0000H
SJMP START
```

```
ORG 0023H
LJMP SERIAL
```

```
START: MOV TMOD,#20H
MOV SCON,#52H
MOV TH1,#0FDH
SETB TR
MOV IE,#10010000B

MOV 21H,#10001000B
MOV 22H,#30
MOV 25H,#0AH
MOV 32H,#1
```

```
; *****
```

```
LEFT: MOV A,22H
JZ LIVE
MOV A,21H
RL A
MOV 21H,A
MOV P0,A
ACALL DELAY1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV R6,22H
DEC R6
MOV 22H,R6
MOV A,22H
MOV B,25H
DIV AB
JZ OUT1
SWAP A
MOV R4,B
ORL A,R4
SWAP A
MOV P2,A
AJMP NEW1
OUT1: MOV A,B
SWAP A
MOV P2,A
NEW1: AJMP LIVE

RIGHT: mov a,run
cjne a,#39,night
MOV A,22H
CJNE A,#60,CON
AJMP LIVE
CON: MOV A,21H
RR A
MOV 21H,A
MOV P0,A
ACALL DELAY1
MOV R6,22H
INC R6
MOV 22H,R6
MOV A,22H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV B,25H
DIV AB
JZ OUT2
SWAP A
MOV R4,B
ORL A,R4
SWAP A
MOV P2,A
AJMP NEW2
OUT2: MOV A,B
SWAP A
MOV P2,A
NEW2: AJMP LIVE
NIGHT: mov a,run
cjne a,#40,exitl
MOV 27H,#60
MOV 24H,#00
AJMP SET
exitl: MOV 26H,#30
ajmp ex
SET: MOV A,22H
JZ LEFT1
MOV R6,22H
MOV A,21H
RL A
MOV P0,A
ACALL DELAY
MOV 21H,A
DEC R6
MOV 22H,R6
MOV A,22H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV B,25H
DIV AB
JZ OUT3
SWAP A
MOV R4,B
ORL A,R4
SWAP A
MOV P2,A
AJMP NEW3
OUT3: MOV A,B
SWAP A
MOV P2,A

NEW3: ACALL STOP
MOV A,22H
JZ LEFT1
JNZ SET

RIGHT1: MOV A,27H
JZ,LEFT1
MOV R7,27H
MOV R6,24H
MOV A,21H
RR A
MOV P0,A
ACALL DELAY
MOV 21H,A
DEC R7
INC R6
MOV 27H,R7
MOV 22H,R6
MOV A,22H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV B,25H
DIV AB
JZ OUT4
SWAP A
MOV R4,B
ORL A,R4
SWAP A
MOV P2,A
AJMP NEW4
OUT4: MOV A,B
SWAP A
MOV P2,A
NEW4: MOV 24H,R6
ACALL STOP
MOV A,27H
JZ LEFT1
JNZ RIGHT1

LEFT1: MOV A,24H
JZ RIGHT1
MOV R7,27H
MOV R6,24H
MOV A,21H
RL A
MOV P0,A
ACALL DELAY
MOV 21H,A
DEC R6
INC R7
MOV 27H,R7
MOV 24H,R6
MOV 22H,R6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,22H
MOV B,25H
DIV AB
JZ OUT5
SWAP A
MOV R4,B
ORL A,R4
SWAP A
MOV P2,A
AJMP NEW5
OUT5: MOV A,B
SWAP A
MOV P2,A
NEW5: ACALL STOP
MOV A,24H
JZ RIGHT1
JNZ LEFT1
; *****
STOP: mov a,run
cjne a,#98,speed1
ajmp live

SPEED1: mov a,run
cjne a,#101,speed2
MOV A,32H
CJNE A,#30,REE
RET
REE: MOV A,32H
INC A
MOV 32H,A
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SPEED2: mov a,run
cjne a,#102,wait
MOV A,32H
CJNE A,#1,REE1
RET
REE1: MOV A,32H
DEC A
MOV 32H,A
RET

```

```

WAIT: ret
waitt: ajmp live
;*****
ex:   mov a,run
cjne a,#100,waitt
ajmp exx

exx:  MOV A,22H
JZ EXITC
MOV R6,22H
MOV A,21H
RL A
MOV P0,A
ACALL DELAY
MOV 21H,A
DEC R6
MOV 22H,R6
MOV A,22H
MOV B,25H
DIV AB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JZ OUT6
SWAP A
MOV R4,B
ORL A,R4
SWAP A
MOV P2,A
AJMP NEW6
OUT6:  MOV A,B
SWAP A
MOV P2,A
NEW6:  MOV A,22H
JZ EXITC
JNZ EXX

EXITC:
MOV R6,22H
MOV R7,26H
MOV A,26H
JZ EXITT
MOV A,21H
RR A
MOV P0,A
MOV 21H,A
ACALL DELAY
INC R6
DEC R7
MOV 22H,R6
MOV A,22H
MOV B,25H
DIV AB
JZ OUT7
SWAP A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV R4,B
ORL A,R4
SWAP A
MOV P2,A
AJMP NEW7
OUT7: MOV A,B
SWAP A
MOV P2,A
NEW7: MOV 26H,R7
MOV A,26H
JZ EXITT
JNZ EXITC
```

```
EXITT: AJMP LIVE
```

```
;*****DELAY*****
```

```
DELAY: MOV R3,#15
```

```
DERAY: MOV R4,#01FH
```

```
DEL: MOV R5,#04FH
```

```
DJNZ R5,$
```

```
DJNZ R4,DEL
```

```
DJNZ R3,DERAY
```

```
RET
```

```
;*****DELAY1*****
```

```
DELAY1: MOV R3,#01H
```

```
DERAY1: MOV R4,#01FH
```

```
DEL1: MOV R5,#06FH
```

```
DJNZ R5,$
```

```
DJNZ R4,DEL1
```

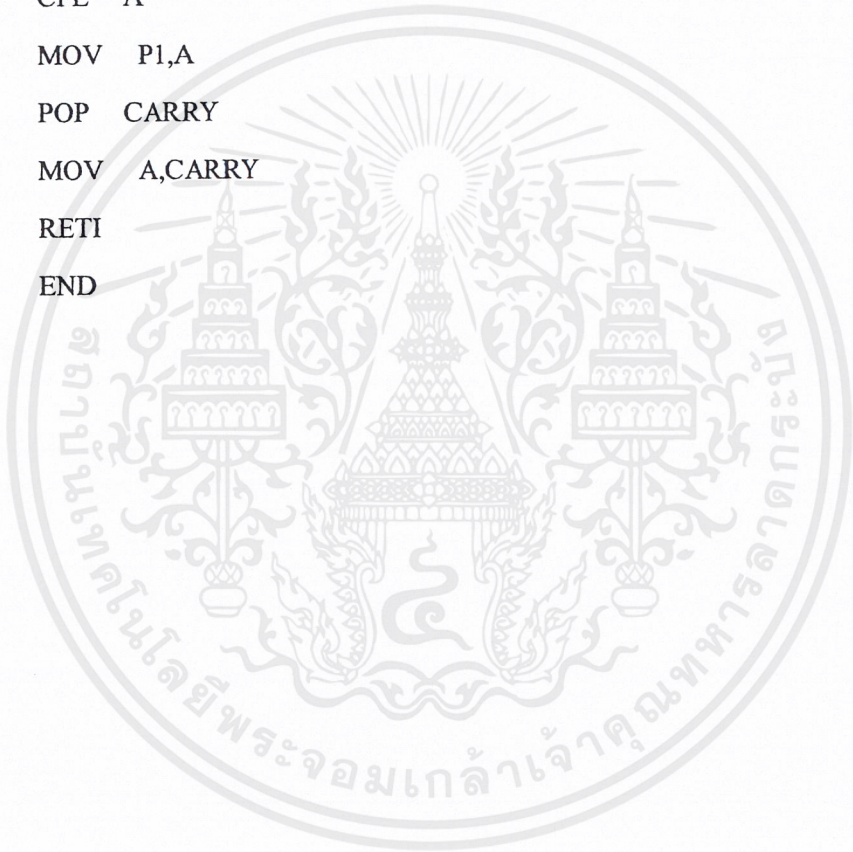
```
DJNZ R3,DERAY1
```

```
RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*****SERIAL*****

```
SERIAL: CLR RI
MOV CARRY,A
PUSH CARRY
MOV A,SBUF
MOV run,A
CPL A
MOV P1,A
POP CARRY
MOV A,CARRY
RETI
END
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาษาแอสเซมบลี

; Telephone Warning

```
ORG 0000h  
RUN EQU 30H  
CARRY EQU 31H
```

```
SJMP START
```

```
ORG 0023H  
LJMP SERIAL
```

```
MOV A,#0FFH  
MOV P0,A
```

```
START: JNB P1.0,JO3  
JMP START
```

```
JO3: CLR P0.7
```

```
jo1: CALL DELAY  
jnb p0.7,jo  
jmp jo1
```

```
jo: call delay
```

```
CLR P0.0
```

```
call delay
```

```
JB p0.0,jo
```

```
MOV TMOD,#20H
```

```
MOV SCON,#40H
```

```
MOV TH1,#0FDH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV PCON,#00H
SETB TR1
MOV A,#41H
CLR TI
MOV SBUF,A
JNB TI,$
```

```
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
CALL DELAY
```

```
START1: MOV TMOD,#20H
MOV SCON,#52H
MOV TH1,#0FDH
SETB TR1
MOV IE,#10010000B
```

```
UP: mov a,run
cjne a,#41H,UP
JNB P1.6,RECALL
```

```
AG: MOV TMOD,#20H
MOV SCON,#40H
MOV TH1,#0FDH
MOV PCON,#00H
MOV R7,#02H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AAA: SETB TR1
MOV A,#42H
CLR TI
MOV SBUF,A
JNB TI,$
```

```
START2: MOV TMOD,#20H
```

```
MOV SCON,#52H
```

```
MOV TH1,#0FDH
```

```
SETB TR1
```

```
MOV IE,#10010000B
```

```
CALL DELAY
```

```
CALL DELAY
```

```
CALL DELAY
```

```
CALL DELAY
```

```
UP1: mov a,run
```

```
cjne a,#42H,UP1
```

```
DJNZ R7,AAA
```

```
mov run,#00h
```

```
DOWN: SETB P0.0
```

```
setb p0.7
```

```
CALL DELAY
```

```
CALL DELAY
```

```
JMP jo1
```

```
recall: SETB P0.0
```

```
CALL DELAY
```

```
JMP jo
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;*****DELAY*****

```
DELAY: MOV R3,#1FH
DERAY: MOV R4,#01FH
DEL: MOV R5,#04FH
DJNZ R5,$
DJNZ R4,DEL
DJNZ R3,DERAY
RET
```

;*****DELAY1*****

```
DELAY1: MOV R3,#01H
DERAY1: MOV R4,#01FH
DEL1: MOV R5,#06FH
DJNZ R5,$
DJNZ R4,DEL1
DJNZ R3,DERAY1
RET
```

;*****SERIAL*****

```
SERIAL: CLR RI
MOV CARRY,A
PUSH CARRY
MOV A,SBUF
MOV run,A
CPL A
MOV P1,A
POP CARRY
MOV A,CARRY
RETI
END
```

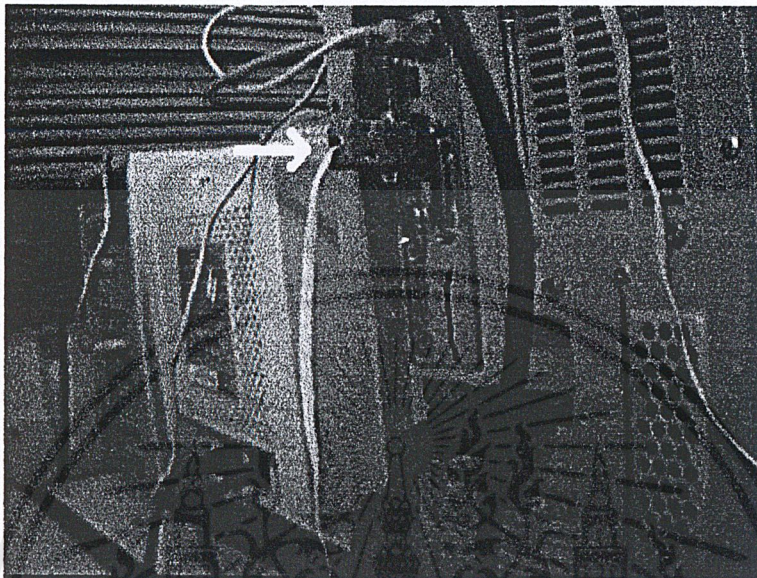
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

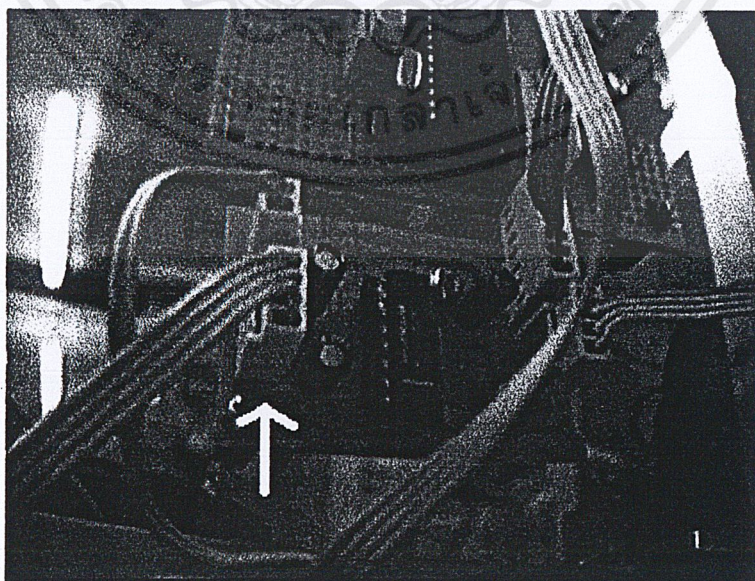
การเชื่อมต่ออุปกรณ์ต่างๆ เข้ากับกล้องวิดีโอ

1. การเชื่อมต่อกับพอร์ตอนุกรม



รูปที่ 1 การเชื่อมต่อเข้าพอร์ตอนุกรม

จากรูปที่ 1 ที่ถูกสร้างขึ้นเป็นการเชื่อมต่อกับคอมพิวเตอร์พอร์ตที่ใช้เป็นแบบ DB9 ต่อเข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ส่วนนี้จะทำหน้าที่ส่งคำสั่งรับส่งจากคอมพิวเตอร์กับ ไมโครโปรเซสเซอร์ การต่อให้ต่อเข้ากับ พอร์ต COM1 ส่วนอีกฝั่งหนึ่งของสายเชื่อมต่อพอร์ตอนุกรมนี้ต่อเข้ากับวงจรแปลงไฟ MAX232 ดังรูปที่ 2



รูปที่ 2 วงจรแปลงไฟ MAX232

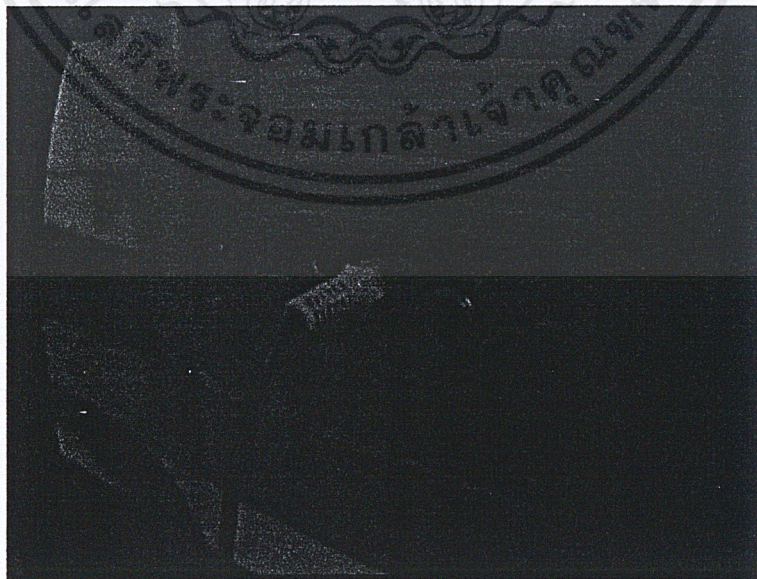
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.การเชื่อมต่อกับกล้องวิดีโอ



รูปที่ 3 สายเชื่อมต่อจากกล้องวิดีโอ

จากรูปที่ 3 มีสายเชื่อมต่อสองสายจากกล้องวิดีโอมีช่องเสียบสีแดงต่อเข้ากับหม้อแปลงไฟตรง (Adapter) 6 V ส่วนช่องเสียบสีเหลืองเป็นสายสัญญาณภาพต่อเข้ากับการ์ดทีวีที่ใช้เสียบช่อง VIDEO IN ดังรูปที่ 4

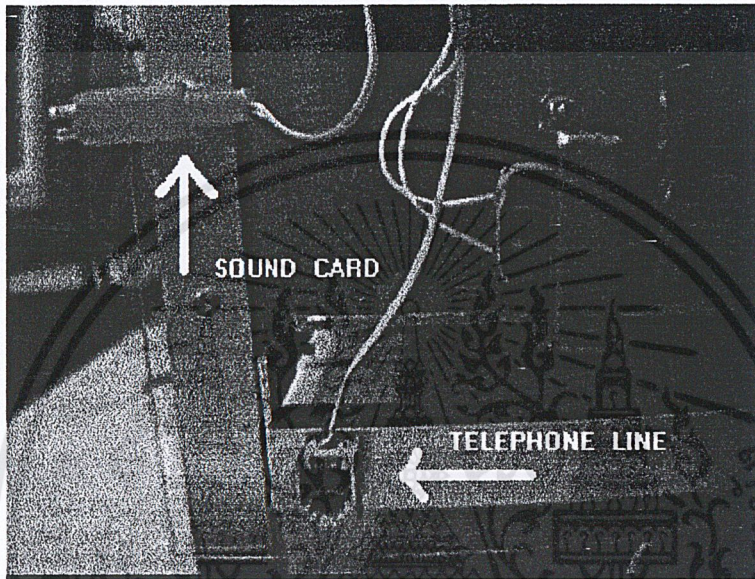


รูปที่ 4 การเชื่อมต่อสายสัญญาณเข้าการ์ดทีวี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การเชื่อมต่อกับการ์ดเสียงและคู่สายโทรศัพท์

การเชื่อมต่อเข้าการ์ดเสียงทำเพื่อใช้ส่งเสียงจากคอมพิวเตอร์หมายเลขโทรศัพท์และส่งเสียงเตือนภัยสู่คู่สายโทรศัพท์ การเชื่อมต่อแสดงดังรูปที่ 5

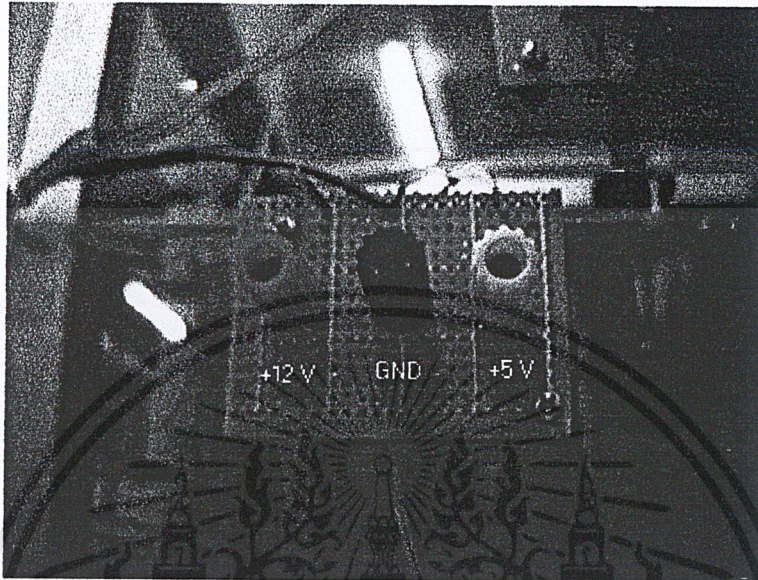


รูปที่ 5 การเชื่อมต่อเข้าสู่คู่สายโทรศัพท์และการ์ดเสียงของคอมพิวเตอร์

จากรูปที่ 5 แสดงการเชื่อมต่อเข้าสู่คู่สายโทรศัพท์และการ์ดเสียงของคอมพิวเตอร์ ช่องเสียบสีแดงต่อกับการ์ดเสียงของคอมพิวเตอร์และช่องเสียบโทรศัพท์เสียบเข้ากับคู่สายโทรศัพท์

5. การเชื่อมต่อกับแหล่งจ่ายไฟ

มีการจ่ายไฟให้กับวงจร 2 ระดับ คือระดับแรงดันไฟตรง 12 V และระดับแรงดันไฟตรง 5 V การเชื่อมต่อไฟเลี้ยงให้กับวงจรแสดงดังรูปที่ 6

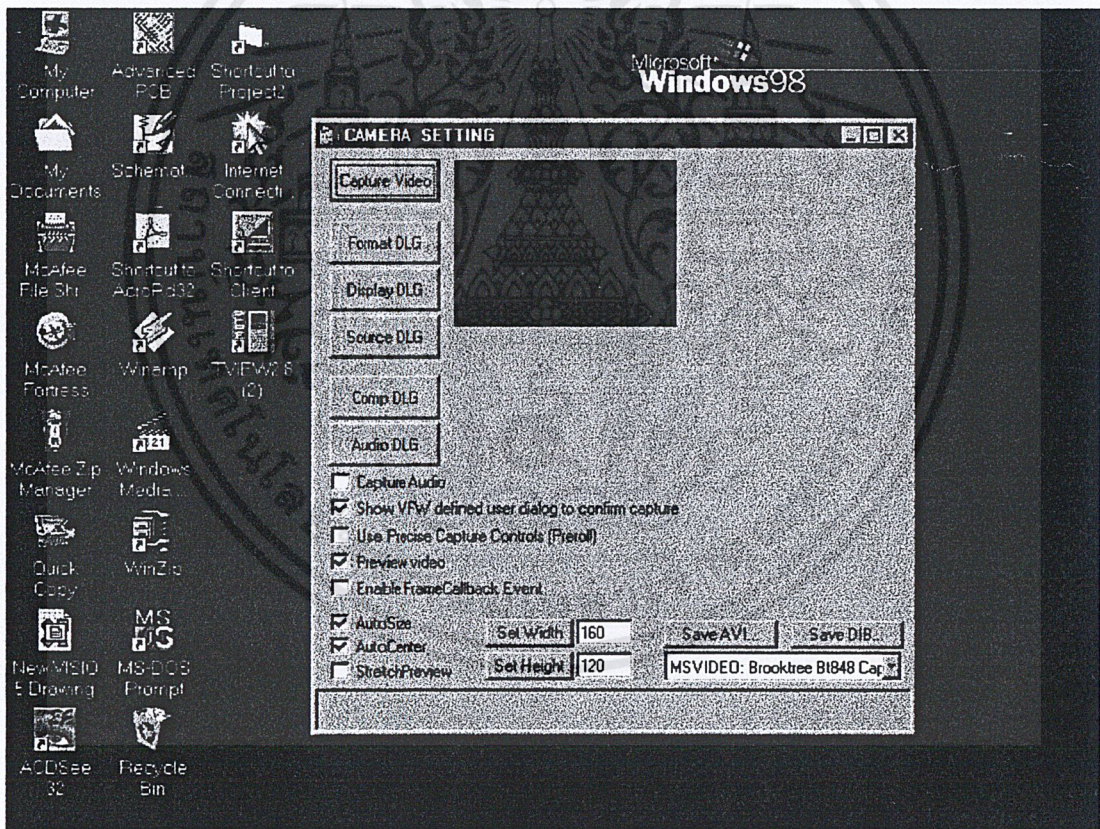


รูปที่ 6 แสดงช่องเสียบเข้ากับแหล่งจ่ายไฟตรง

จากรูปที่ 6 ช่องเสียบสีแดงต่อเข้ากับไฟตรง +12 V ช่องเสียบสีเหลืองต่อเข้ากับไฟตรง +5V ส่วนช่องเสียบสีดำต่อเข้ากับ กราวนด์ของแหล่งจ่ายทั้ง 2 ระดับ คือต้องเชื่อมกราวนด์ของแหล่งจ่ายทั้งสองด้วย

วิธีการใช้กล้องวิดีโอ

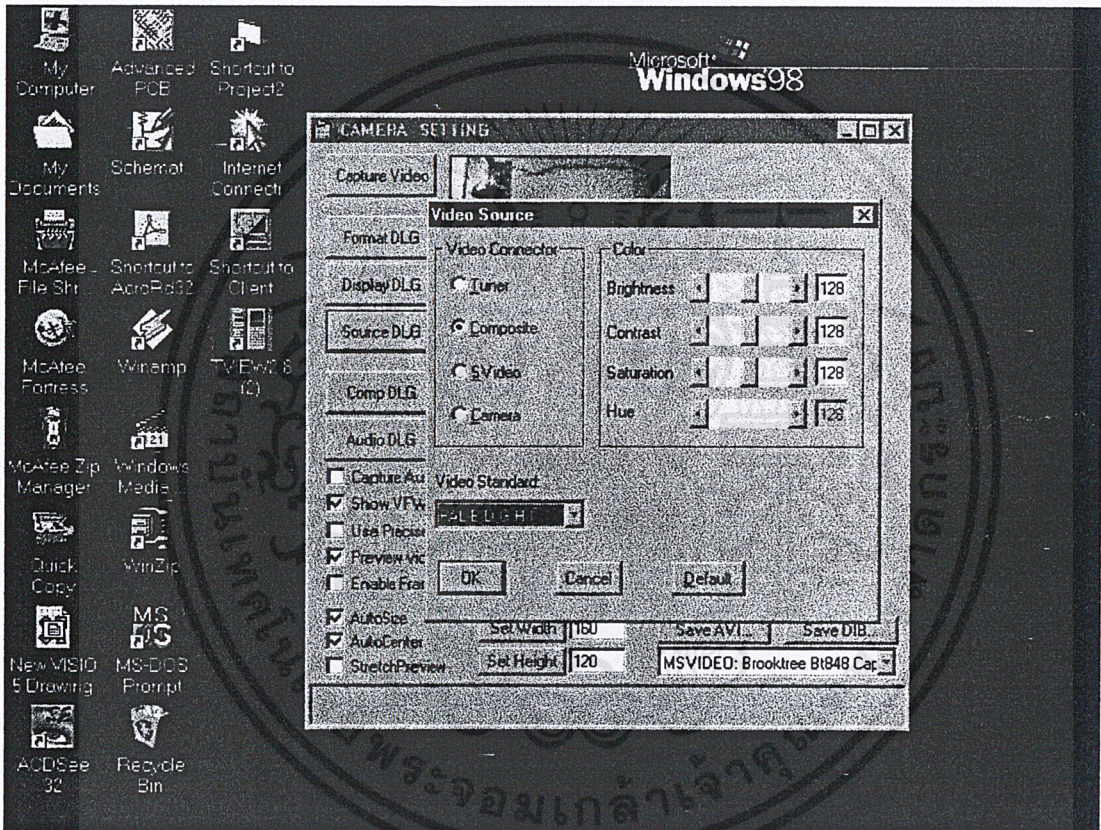
1. เปิดคอมพิวเตอร์แล้วเรียกโปรแกรม COMPUTER CONTROL VIDEO POSITION และโปรแกรม CAMERA SETTING ออกมา ที่โปรแกรม COMPUTER CONTROL VIDEO POSITION จะมีช่องใส่หมายเลขโทรศัพท์ ให้ใส่หมายเลขโทรศัพท์ที่ต้องการให้เครื่องโทรออกอัตโนมัติไว้
2. ต่อสายเชื่อมต่อต่างๆ ตรวจสอบดูให้แน่ใจว่าได้เชื่อมต่อแล้วทุกจุด
3. จัดตำแหน่งกล้องให้อยู่ตำแหน่งตรงกลางที่ต้องการ
4. การใช้งาน โปรแกรมมี 2 ฟอรัมด้วยกัน ดังนี้
 - 4.1 ฟอรัมของส่วน CAMERA SETTING มีรูปแบบของ ฟอรัมดังนี้



รูปที่ 7 ฟอรัมของส่วน CAMERA SETTING เมื่อทำการรันโปรแกรมขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการรันโปรแกรม CAMERA SETTING ขึ้นมาครั้งแรกจะปรากฏฟอร์มดังรูปที่ 7 จากนั้น ให้ Click ที่ Source DLG จะปรากฏฟอร์มดังรูปที่ 8

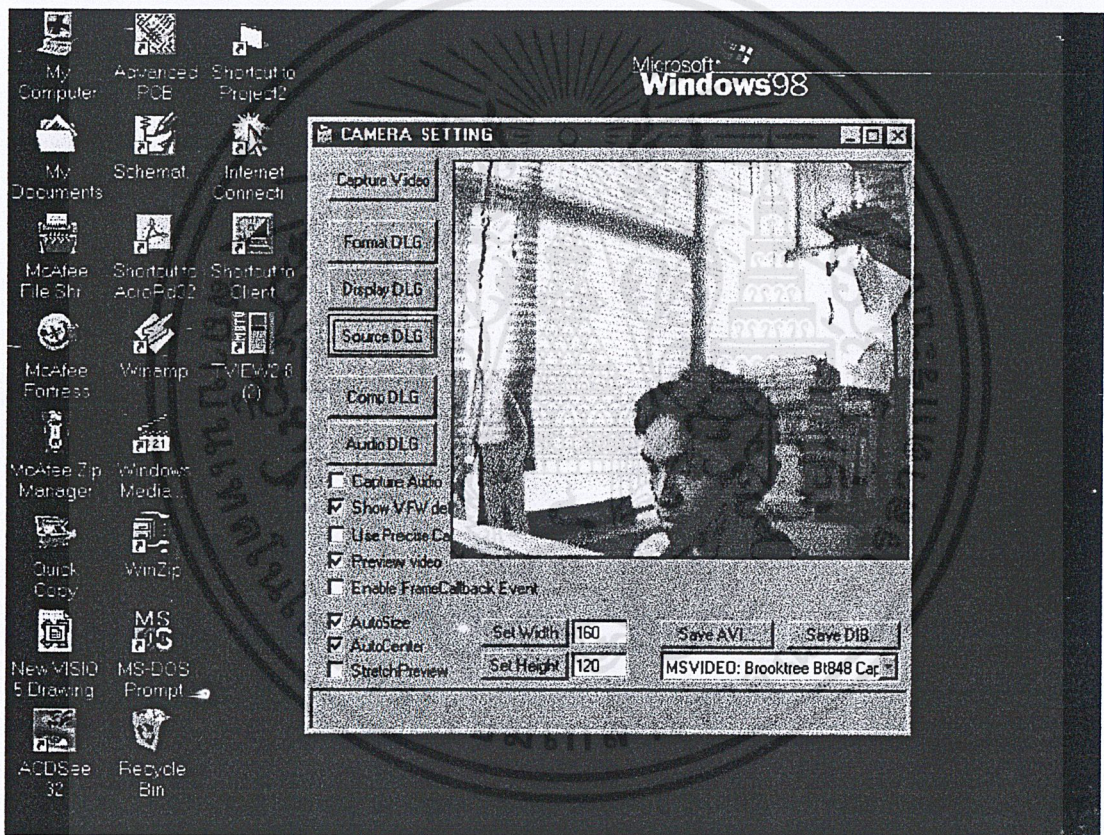


รูปที่ 8 ฟอร์มเมื่อ Click ที่ Source DLG

ในส่วนของ Video Connector ให้เลือก Composite และในส่วนของ Video Standard ให้เลือก PAL-B,D,G,H,I แล้ว Click OK จะปรากฏภาพที่ถ่ายได้จากกล้อง ดังรูปที่ 9 แต่ถ้าไม่ปรากฏภาพอาจเกิดจากยังไม่เลือก Driver ที่ถูกต้องให้กับกล้อง ในที่นี้ให้เลือกที่ MSVIDEO; Brooktree Bt848 Capture ซึ่งเป็น การ์ด TV Tuner ที่ ใช้กับกล้อง ในส่วนการปรับความคมชัดของภาพท่านก็สามารถเลือกปรับได้ตามต้องการ ที่กรอบ Color โดยสามารถปรับได้ทั้ง Brightness , Contrast , Saturation และ Hue ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนอื่น ๆ ของฟอร์มสามารถเซตค่าได้ตามต้องการ ไม่ว่าจะ เป็น Format DLG จะเป็น ส่วนของ Video Format สามารถเลือกขนาดของจอภาพได้ แต่โปรเจกนี้เลือกที่ ขนาด 1/2 หรือ 384 x 288 จะได้ภาพขนาดที่เห็นในรูปที่ 9



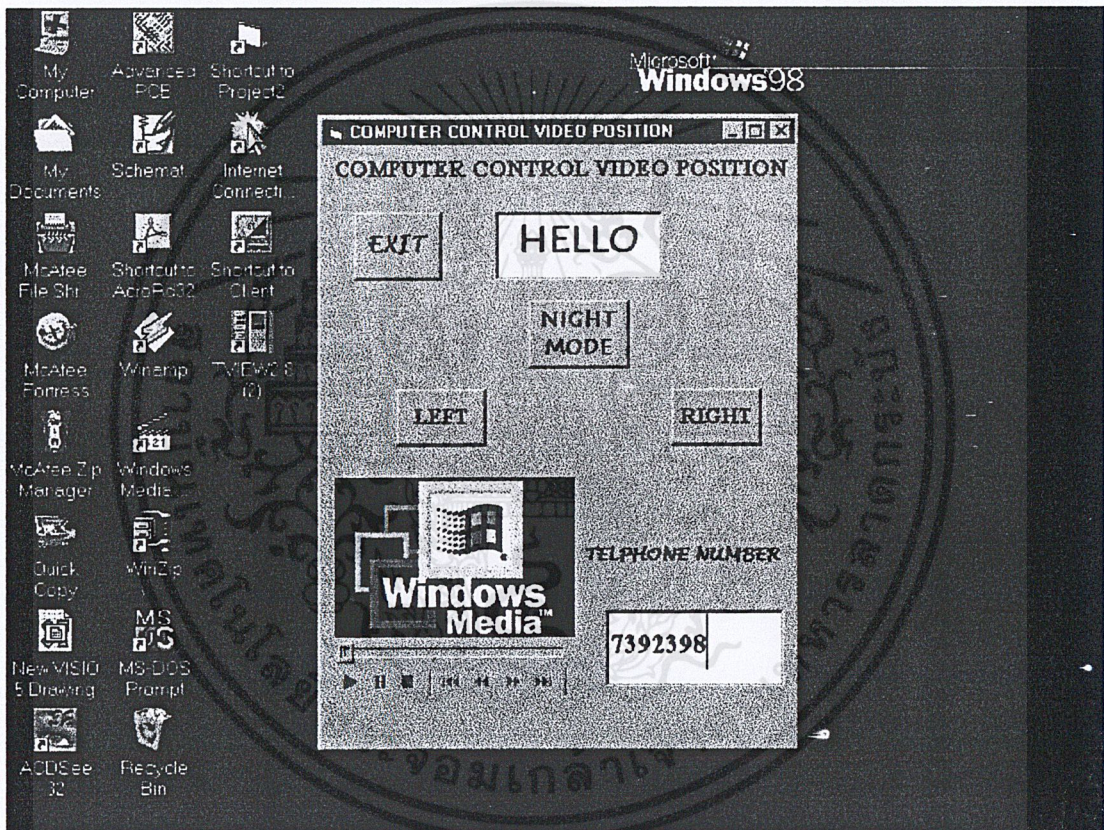
รูปที่ 9 แสดงภาพจากกล้องวีดีโอเมื่อเซตค่าใน Source DLG ถูกต้อง

4.2 ฟอร์มในส่วน COMPUTER CONTROL VIDEO POSITION มีรูปแบบของฟอร์มดังรูปที่ 10

เมื่อทำการรัน COMPUTER CONTROL VIDEO POSITION ออกมาจะปรากฏ Text box 2 Text box ด้วยกัน Text box อันบนจะใช้แสดงข้อความเมื่อทำการตั้งงานผ่าน Botton ต่าง ๆ คือ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มต้นรัน COMPUTER CONTROL VIDEO POSITION จะปรากฏ Text box แสดงข้อความว่า “HELLO” ขึ้นมาดังรูปที่ 10 ข้อความใน Text box จะเปลี่ยนแปลงตาม การสั่งงานที่ Botton ต่าง ๆ เช่น เมื่อ Click ให้กล่องหมุนซ้ายจะปรากฏข้อความ “LEFT” เป็นต้น

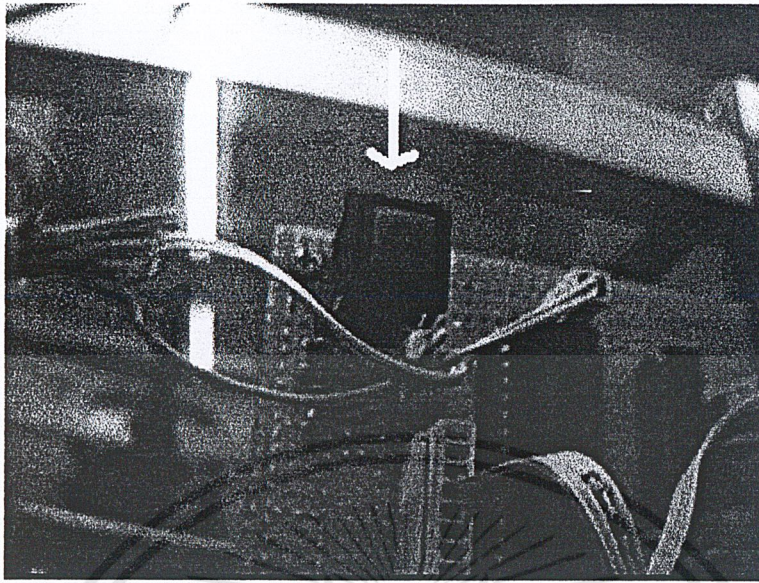
ส่วน Text box อื่นต่างจะมีช่องใส่หมายเลขโทรศัพท์ ให้ใส่หมายเลขโทรศัพท์ที่ต้องการให้เครื่องโทรออกในการเตือนภัยอัตโนมัติ



รูปที่ 10 แสดงฟอร์ม COMPUTER CONTROL VIDEO POSITION

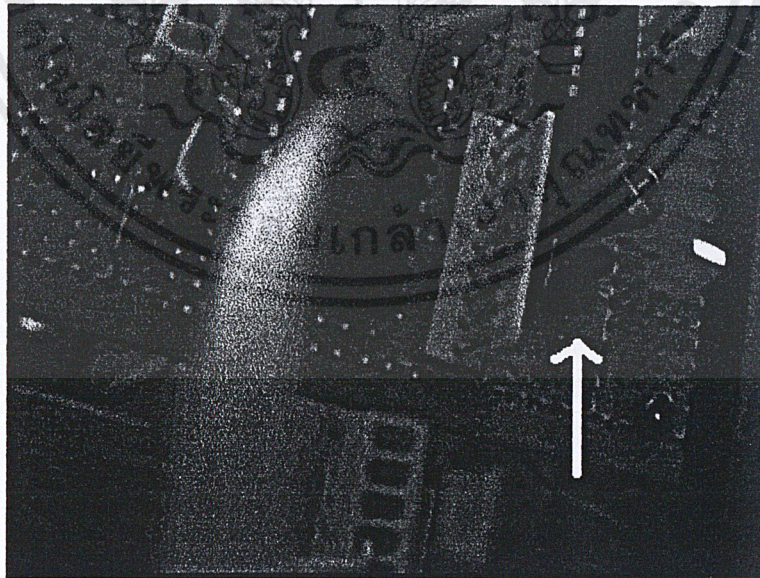
5. ในกรณีที่ต้องการให้โทรออกอัตโนมัติให้คปุมสี่เหลี่ยมที่วงจรวงดังแสดงในรูปที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 ปุ่มสั่งงานโทรออกอัตโนมัติ

6. เมื่อได้มีการโทรศัพท์เรียบร้อยแล้วต้องทำการรีเซ็ตเครื่องใหม่โดยการกดปุ่มรีเซ็ตทุกครั้ง ไม่เช่นนั้นแล้วเครื่องจะไม่ตอบสนองต่อการกดครั้งต่อไป นอกจากกดปุ่มรีเซ็ตแล้ว การรีเซ็ตก็สามารถทำได้โดยการปิดแล้วเปิดเครื่องใหม่อีกครั้ง ปุ่มรีเซ็ตแสดงดังรูปที่ 8



รูปที่ 8 ปุ่มรีเซ็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค

Data Sheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BCD TO 7-SEGMENT DECODER

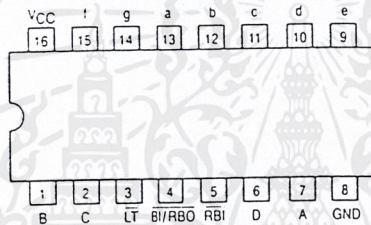
The SN54/74LS48 is a BCD to 7-Segment Decoder consisting of NAND gates, input buffers and seven AND-OR-INVERT gates. Seven NAND gates and one driver are connected in pairs to make BCD data and its complement available to the seven decoding AND-OR-INVERT gates. The remaining NAND gate and three input buffers provide lamp test, blanking input/ripple-blanking input for the LS48.

The circuit accepts 4-bit binary-coded-decimal (BCD) and, depending on the state of the auxiliary inputs, decodes this data to drive other components. The relative positive logic output levels, as well as conditions required at the auxiliary inputs, are shown in the truth tables.

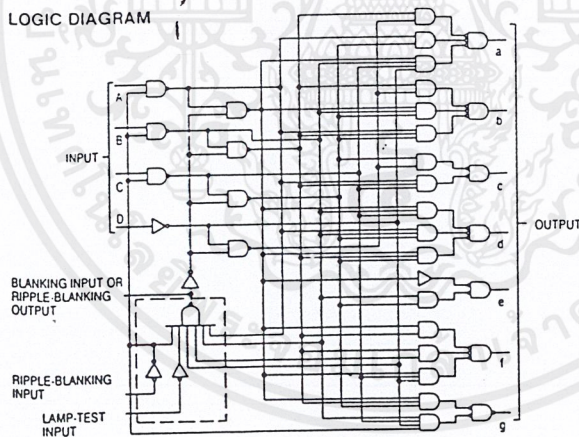
The LS48 circuit incorporates automatic leading and/or trailing edge zero-blanking control (RBI and RBO). Lamp Test (LT) may be activated any time when the BI/RBO node is HIGH. Both devices contain an overriding blanking input (BI) which can be used to control the lamp intensity by varying the frequency and duty cycle of the BI input signal or to inhibit the outputs.

- Lamp Intensity Modulation Capability (BI/RBO)
- Internal Pull-Ups Eliminate Need for External Resistors
- Input Clamp Diodes Eliminate High-Speed Termination Effects

CONNECTION DIAGRAM DIP (TOP VIEW)

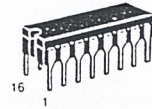


LOGIC DIAGRAM

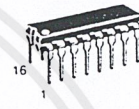


SN54/74LS48

BCD TO 7-SEGMENT DECODER
LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 620-09



N SUFFIX
PLASTIC
CASE 648-08

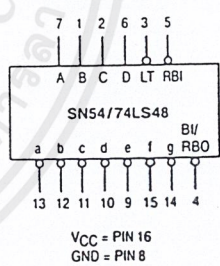


D SUFFIX
SOIC
CASE 751B-03

ORDERING INFORMATION

- SN54LSXXJ Ceramic
- SN74LSXXN Plastic
- SN74LSXXD SOIC

LOGIC SYMBOL



+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

MAX220-MAX249

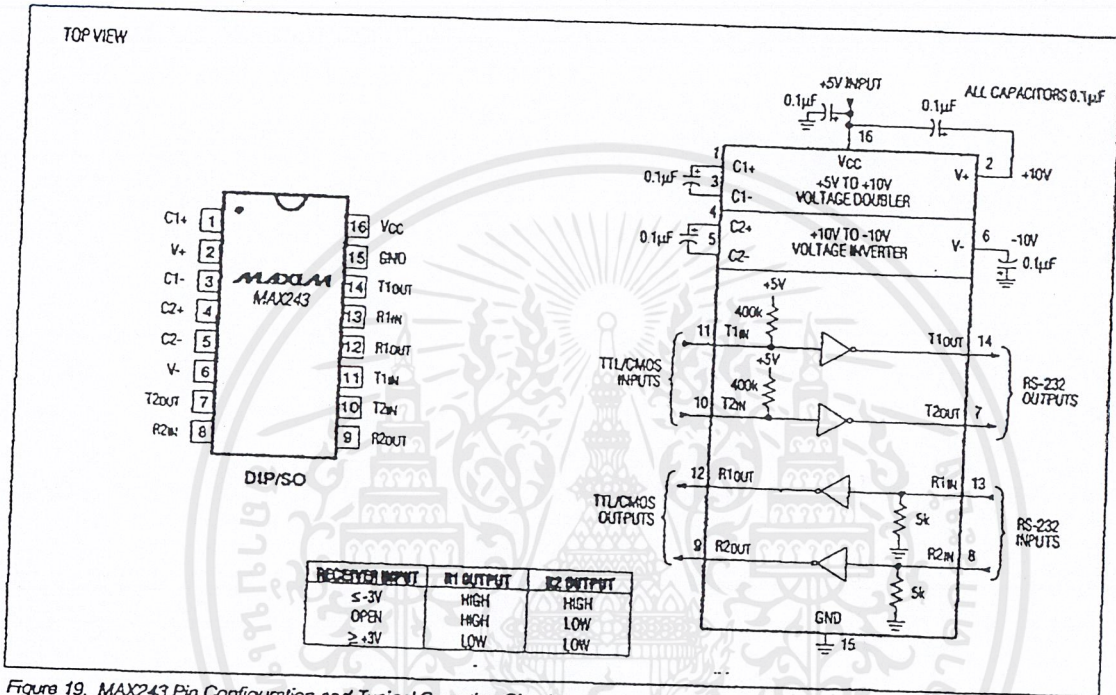


Figure 19. MAX243 Pin Configuration and Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้