สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ป้ายแสดงราคาน้ำมัน

(Display for show gasoline prices)

โดย

นาย นิพิฐ ก้าวสัมพันธ์

นาย ปฐมพงศ์ วิริยะอุดมชาติ

ป/ว
ม61926
2549

เลขหมู่.....
เลขทะเบียน 72844
วัน,เดือน,ปี 23 มิ.ย. 2550

b 11773200
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้ายแสดงราคาน้ำมัน

(Display for show gasoline prices)

โดย

นาย นิพิฐ ก้าวสัมพันธ์ รหัส 46010372

นาย ปฐมพงศ์ วิริยะอุดมชาติ รหัส 46010404

อาจารย์ที่ปรึกษา

ผศ. พลผดุง ผดุงกุล

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2549

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ป้ายแสดงราคาน้ำมัน

ผู้จัดทำ

1. **นาย นิพิฐ ก้าวสัมพันธ์**
2. **นาย ปฐมพงศ์ วิริยะอุดมชาติ**

.....อาจารย์ที่ปรึกษา

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้ายแสดงราคาน้ำมัน

นาย นิพิฐ ก้าวสัมพันธ์
นาย ปฐมพงศ์ วิริยะอุดมชาติ
อาจารย์ที่ปรึกษา : อ.พลผดุง ผดุงกุล
ภาคเรียนที่ 2 ปีการศึกษา 2549

บทคัดย่อ

รายงานฉบับนี้เป็นรายงานประกอบโครงงานป้ายแสดงราคาน้ำมัน แสดงผลด้วยชุดแสดงผล LED ตัวเลขเจ็ดส่วน ซึ่งรับค่าราคาน้ำมันที่ต้องการแสดงผลจากคอมพิวเตอร์ซึ่งใช้โปรแกรม Visual Basic ในการสร้างหน้าต่างรับค่าราคาน้ำมันที่ต้องการแสดงผล ค่าที่ป้อนเข้าไปจะส่งไปแสดงผลโดยการส่งผ่านพอร์ตอนุกรมของคอมพิวเตอร์ไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผลและสั่งให้ชุดแสดงผล LED ตัวเลขเจ็ดส่วน แสดงราคาน้ำมันที่ต้องการแสดง ซึ่งการส่งข้อมูลนั้นจะส่งแบบไร้สายโดยใช้คลื่นความถี่วิทยุ และชุดแสดงผล LED ตัวเลขเจ็ดส่วนซึ่งใช้ในการแสดงราคาน้ำมันสามารถควบคุมด้วยรีโมทซึ่งเลือกได้ว่าจะส่งข้อมูลโดยผ่านคลื่นความถี่วิทยุ หรือจะต่อสายตรงก็ได้โครงงานนี้สามารถนำไปเป็นแนวทางในการศึกษา ออกแบบ อีกทั้งยังสามารถนำไปประยุกต์ใช้งานในอนาคตต่อไปได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dot matrix 7-Segment for show gasoline prices

Mr. Nipit Kaosamphan

Mr. Prathompong Viriyaudomchat

Advisor: Mr. Polpadung Padungkul

2nd semester, Education year 2006

Abstract

This report proposes a method for constructing a display to show gasoline prices. Gasoline prices can be shown with dot matrix 7-segment. The price wanted show can receive from window in program visual basic on computer. The gasoline price is transmitting from serial port on computer to microcontroller. It compiles and control dot matrix 7-segment for show gasoline prices. The transmission use wireless in radio frequency. Dot matrix 7-segment for show gasoline prices can control with RF remote control whose transmission can use wire and wireless.This report can use as a guide line for study, design and improve other work in the future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จรุร่วงได้ดีด้วยความอนุเคราะห์จากผู้ทรงคุณวุฒิหลายท่าน โดยเฉพาะอาจารย์ที่ปรึกษา ผศ. พลผดุง ผดุงกุล และอาจารย์ประจำภาควิชาอิเล็กทรอนิกส์นับตั้งแต่เริ่มต้นจนเสร็จสิ้นสมบูรณ์จึงใคร่ขอกราบขอบพระคุณด้วยความเคารพอย่างสูง

ขอขอบพระคุณคณะกรรมการทุกท่านที่กรุณาเป็นคณะกรรมการสอบปริญญานิพนธ์เพื่อให้ปริญญานิพนธ์ฉบับนี้มีความสมบูรณ์

สุดท้ายนี้ขอขอบพระคุณบิดามารดาที่ได้ให้ความช่วยเหลือด่างๆ และเป็นกำลังใจให้ผู้จัดทำมาโดยตลอด

อนึ่งหากปริญญานิพนธ์นี้มีคุณค่าและมีประโยชน์ต่อการศึกษาค้นคว้าของผู้ที่สนใจผู้จัดทำใคร่ขออุทิศให้แก่บิดาครูอาจารย์ตลอดจนผู้มีพระคุณทุกท่าน ส่วนข้อผิดพลาดและข้อบกพร่องใดๆ ผู้เขียนใคร่ขอน้อมรับไว้แต่ผู้เดียว

นาย ปฐมพงศ์ วิริยะอุดมชาติ

มีนาคม 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงงาน	1
1.2 ขั้นตอนการทำงาน	2
1.3 ขอบเขตการทำงาน	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 หลักการและทฤษฎี	3
2.1 ไดโอดเปล่งแสง (Light Emitting Diode: LED)	3
2.2 โปรแกรม Visual Basic	6
2.3 ระบบสื่อสารอิเล็กทรอนิกส์ (Electronic Communication Systems)	7
2.4 การมอดูเลต	13
2.5 การสื่อสารผ่านพอร์ต RS-232C	23
2.6 ไมโครคอนโทลเลอร์ (Microcontroller)	27
บทที่ 3 การออกแบบ	44
3.1 การสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์	44
3.2 การสื่อสารระบบไร้สายระหว่างไมโครคอนโทรลเลอร์ 2 ตัว	45
บทที่ 4 การทดลอง	54
4.1 การทดสอบสัญญาณที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์	54
4.2 สัญญาณหลังจากผ่าน RS-232	60
4.3 สัญญาณหลังจากการเข้ารหัส	66
บทที่ 5 สรุปและวิจารณ์โครงการ	72
5.1 สรุปผลการปฏิบัติงาน	72
5.2 ปัญหาและอุปสรรค	72
5.3 แนวทางในการแก้ไข	72
5.4 ประโยชน์และการประยุกต์ใช้งาน	73
5.5 สิ่งที่ได้จากการทำโครงงานนี้	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

เรื่อง	หน้า
รูปที่ 2.1 โครงสร้างพื้นฐาน และรูปร่างของไดโอดเปล่งแสงแบบรอยต่อ P-N ของ GaAs	3
รูปที่ 2.2 แสดงลักษณะของ LED	4
รูปที่ 2.3 แสดงโครงสร้างภายในของ LED	4
รูปที่ 2.4 รูปร่างลักษณะภายนอกของไดโอดเปล่งแสงที่ผลิตจำหน่ายในท้องตลาดปัจจุบัน	5
รูปที่ 2.5 แสดงการมองเห็นแสง ของ LED ในย่านความยาวคลื่นต่างๆ	5
รูปที่ 2.6 แบบจำลองระบบสื่อสาร	7
รูปที่ 2.7 การสื่อสารแบบทิศทางเดียว	12
รูปที่ 2.8 การสื่อสารแบบกึ่งสองทิศทาง	12
รูปที่ 2.9 การสื่อสารแบบสองทิศทาง	13
รูปที่ 2.10 การมอดูเลตทางความถี่	15
รูปที่ 2.11 ความแตกต่างระหว่างคลื่น PM กับ FM	17
รูปที่ 2.12 การแปลงระหว่างคลื่น PM กับ FM	19
รูปที่ 2.13 การมอดูเลตสัญญาณแบบ Amplitude Shift Keying	20
รูปที่ 2.14 การมอดูเลตสัญญาณแบบ Frequency Shift Keying	21
รูปที่ 2.15 การมอดูเลตสัญญาณแบบ Phase Shift Keying	21
รูปที่ 2.16 การใช้ RS-232C เชื่อมต่ออุปกรณ์	23
รูปที่ 2.17 การกำหนดข้อต่อของ RS-232C	24
รูปที่ 2.18 โครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51	27
รูปที่ 2.19 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51	29
รูปที่ 2.20 รายละเอียดของรีจิสเตอร์แสดงสถานะของโปรแกรมหรือ PSW	33
รูปที่ 3.1 บล็อกไดอะแกรมของการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์	44
รูปที่ 3.2 วงจรที่ใช้ในการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์	45
รูปที่ 3.3 บล็อกไดอะแกรมของภาคส่งสัญญาณ	45
รูปที่ 3.4 Timing Diagram ของ Encoder	46
รูปที่ 3.5 วงจรภาคส่งสัญญาณ	47
รูปที่ 3.6 บล็อกไดอะแกรมภาครับสัญญาณ	48
รูปที่ 3.7 Timing Diagram ของ Decoder	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 วงจรภาครับสัญญาณ 50
รูปที่ 3.9 วงจรภายในของ ULN2803A 51
รูปที่ 3.10 บล๊อกไดอะแกรมภาครีโมทควบคุม 51
รูปที่ 3.11 วงจรภาครีโมทควบคุม 53
รูปที่ 4.1 สัญญาณเลข 0 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 54
รูปที่ 4.2 สัญญาณเลข 1 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 55
รูปที่ 4.3 สัญญาณเลข 2 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 55
รูปที่ 4.4 สัญญาณเลข 3 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 56
รูปที่ 4.5 สัญญาณเลข 4 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 56
รูปที่ 4.6 สัญญาณเลข 5 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 57
รูปที่ 4.7 สัญญาณเลข 6 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 57
รูปที่ 4.8 สัญญาณเลข 7 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 58
รูปที่ 4.9 สัญญาณเลข 8 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 58
รูปที่ 4.10 สัญญาณเลข 9 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 59
รูปที่ 4.11 สัญญาณเลข 48 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์ 59
รูปที่ 4.12 สัญญาณเลข 0 หลังจากผ่าน RS-232 60
รูปที่ 4.13 สัญญาณเลข 1 หลังจากผ่าน RS-232 61
รูปที่ 4.14 สัญญาณเลข 2 หลังจากผ่าน RS-232 61
รูปที่ 4.15 สัญญาณเลข 3 หลังจากผ่าน RS-232 62
รูปที่ 4.16 สัญญาณเลข 4 หลังจากผ่าน RS-232 62
รูปที่ 4.17 สัญญาณเลข 5 หลังจากผ่าน RS-232 63
รูปที่ 4.18 สัญญาณเลข 6 หลังจากผ่าน RS-232 63
รูปที่ 4.19 สัญญาณเลข 7 หลังจากผ่าน RS-232 64
รูปที่ 4.20 สัญญาณเลข 8 หลังจากผ่าน RS-232 64
รูปที่ 4.21 สัญญาณเลข 9 หลังจากผ่าน RS-232 65
รูปที่ 4.22 สัญญาณเลข 48 หลังจากผ่าน RS-232 65
รูปที่ 4.23 สัญญาณเลข 0 หลังจากการเข้ารหัส 66
รูปที่ 4.24 สัญญาณเลข 1 หลังจากการเข้ารหัส 67
รูปที่ 4.25 สัญญาณเลข 2 หลังจากการเข้ารหัส 67
รูปที่ 4.26 สัญญาณเลข 3 หลังจากการเข้ารหัส 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.27 สัญญาณเลข 4 หลังจากการเข้ารหัส 68
รูปที่ 4.28 สัญญาณเลข 5 หลังจากการเข้ารหัส 69
รูปที่ 4.29 สัญญาณเลข 6 หลังจากการเข้ารหัส 69
รูปที่ 4.30 สัญญาณเลข 7 หลังจากการเข้ารหัส 70
รูปที่ 4.31 สัญญาณเลข 8 หลังจากการเข้ารหัส 70
รูปที่ 4.32 สัญญาณเลข 9 หลังจากการเข้ารหัส 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 แถบความถี่และการใช้งาน	10
ตารางที่ 2.2 เปรียบเทียบค่าความถี่เบี่ยงเบนและเฟสเบี่ยงเบนในระบบ FM และ PM	18
ตารางที่ 2.3 การกำหนดย่านของแรงดันฟ้าในสัญญาณตามมาตรฐาน	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันการติดต่อสื่อสารทำได้ง่ายโดยอาศัยเทคโนโลยีของอุปกรณ์การสื่อสารที่พัฒนาขึ้นเพื่ออำนวยความสะดวกในการติดต่อสื่อสาร สิ่งเหล่านี้ได้เข้ามามีบทบาทในชีวิตประจำวันของผู้คนในปัจจุบันมากขึ้น โดยเฉพาะด้านธุรกิจโฆษณา ที่ต้องการสื่อที่สามารถแสดงข่าวสารได้อย่างครบถ้วนและมีรูปแบบที่ดึงดูดความสนใจพร้อมทั้งสื่อสารข้อมูลให้ทุกคนเข้าใจได้อย่างชัดเจน

ป้ายแสดงราคาน้ำมัน LED 7-Segment เป็นอีกสื่อหนึ่งที่ได้รับความนิยมใช้กันมากในปัจจุบัน เนื่องจากราคาน้ำมันในปัจจุบันมีการเปลี่ยนแปลงอย่างรวดเร็ว ซึ่งจะเห็นได้จากสถานบริการน้ำมันใหญ่ๆทั่วประเทศ

รายงานฉบับนี้ได้อธิบายขั้นตอน และวิธีการในการออกแบบ รวมทั้งวงจร และผลการทดลองทดสอบคุณสมบัติต่างๆของป้ายแสดงราคาน้ำมัน โดยจะมีเนื้อหาแบ่งออกเป็นบทต่างๆ ดังต่อไปนี้

บทที่ 2 กล่าวถึงทฤษฎี และหลักการพื้นฐานของการสื่อสารตลอดจนทฤษฎีที่เกี่ยวข้องเพื่อที่จะได้นำไปใช้ในการออกแบบป้ายแสดงราคาน้ำมัน

บทที่ 3 กล่าวถึงการออกแบบ โดยจะกล่าวถึงขั้นตอนในการออกแบบส่วนต่างๆของป้ายราคาน้ำมัน

บทที่ 4 กล่าวถึงการทดลอง และผลการทดลองคุณสมบัติต่างๆของป้ายแสดงราคาน้ำมัน

บทที่ 5 เป็นการสรุป และวิจารณ์ผลการออกแบบป้ายราคาน้ำมันที่เราได้สร้างขึ้น ตลอดจนสิ่งที่ได้ในการศึกษาวิชานี้

1.1 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการรับส่งสัญญาณความถี่วิทยุ
2. เพื่อศึกษาโปรแกรม Visual Basic 6
3. เพื่อศึกษาการส่งข้อมูลออกทางพอร์ตอนุกรมของคอมพิวเตอร์
4. เพื่อศึกษาโปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์ MCS-51
5. เพื่อศึกษาการติดต่อระหว่างไมโครคอนโทรลเลอร์ กับ ไอซีภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ขั้นตอนการทำงาน

1. ศึกษาหลักการทำงานและส่วนประกอบของอุปกรณ์และวงจรรับส่งสัญญาณ
2. ศึกษาโปรแกรม Visual Basic 6
3. ศึกษาการส่งข้อมูลออกทางพอร์ตอนุกรมของคอมพิวเตอร์
4. ออกแบบ Page หน้าจอสำหรับรับค่าราคาน้ำมันที่ต้องการส่งออกพอร์ตอนุกรมเพื่อแสดงผลออกทาง LED 7-Segment
5. ศึกษาโปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์ MCS-51
6. ศึกษาการติดต่อระหว่างไมโครคอนโทรลเลอร์ กับ ไอซีภายนอก
7. เขียนโปรแกรมภาษา C เพื่อควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์
8. ออกแบบวงจร
9. นำวงจรที่ได้ไปออกแบบลายวงจรที่จะลงแผ่นปริ้นท์ โดยใช้โปรแกรมโพรเทล พีซีบี (Protel PCB) แล้วกัดปริ้นท์
10. นำอุปกรณ์ประกอบลงบนแผ่นปริ้นท์

1.3 ขอบเขตการทำงาน

โครงงานป้ายแสดงราคาน้ำมัน ประกอบด้วยไมโครคอนโทรลเลอร์ตัวหนึ่งรับค่าจากพอร์ตอนุกรมของคอมพิวเตอร์และแปลงเป็นรหัสส่งผ่านตัวส่งไร้สายไปยังตัวรับ และทำการแปลงรหัสเพื่อแสดงผลด้วย LED 7-Segment พร้อมทั้งมีรีโมทควบคุมการแสดงผลโดยสามารถควบคุมได้ทั้งแบบมีสายและแบบไร้สาย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจถึงหลักการทำงานและทฤษฎีต่างๆที่เกี่ยวข้องที่ใช้ในการออกแบบการรับส่งสัญญาณ
2. สามารถนำความรู้ที่ได้จากการศึกษาโปรแกรม Visual Basic 6 และโปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์ MCS-51 ไปประยุกต์ใช้งานต่อไปได้ในอนาคตได้
3. ทำให้ทราบถึงอุปสรรคและแนวทางการพัฒนา
4. สามารถนำความรู้ไปใช้ในการออกแบบผลิตภัณฑ์ต่อไปในอนาคตได้
5. สามารถนำความรู้ไปประยุกต์ใช้ในการส่งสัญญาณในรูปแบบอื่นๆได้

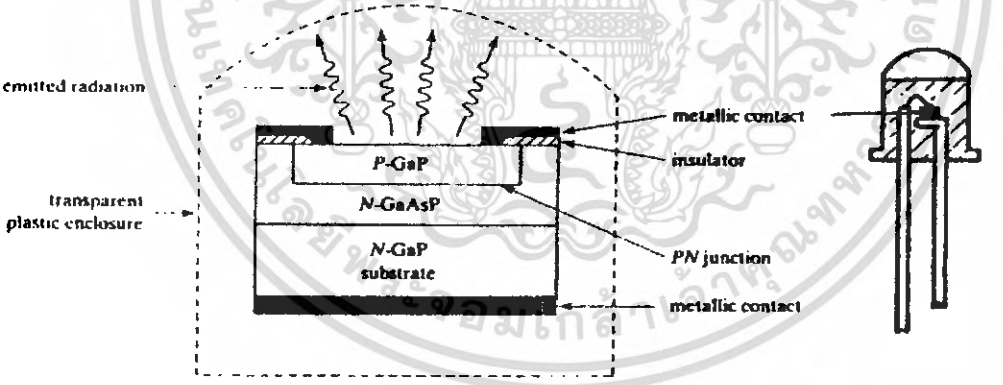
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎี

2.1 ไดโอดเปล่งแสง (Light Emitting Diode: LED)

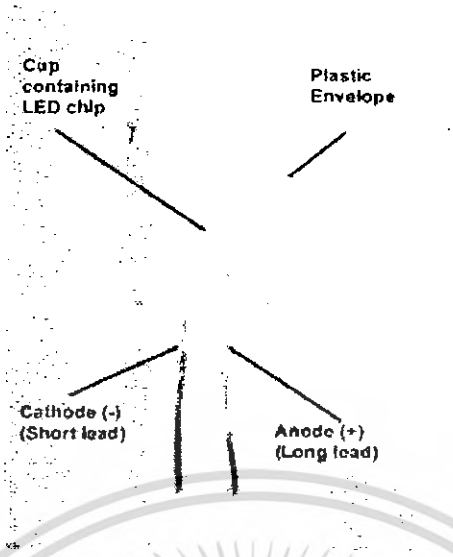
LED ย่อจาก Light Emitting Diodes มีให้เห็นได้ทั่วไปในอุปกรณ์อิเล็กทรอนิกส์ เช่น นาฬิกาดิจิตอล รีโมทคอนโทรล หน้าปัดอุปกรณ์ไฟฟ้า โทรทัศน์ ไฟจราจร เป็นต้น ที่จริงแล้ว หลอด LED คือหลอดไฟขนาดเล็กแต่มีหลักการทำงานแตกต่างจากหลอดไฟมีไส้ เพราะ LED ไม่มีการเผาไส้หลอด ดังนั้นหลอด LED จึงไม่เกิดความร้อน แสงสว่างเกิดขึ้นจากการเคลื่อนของอิเล็กตรอนภายในสารกึ่งตัวนำ

ไดโอดเปล่งแสงเป็นสิ่งประดิษฐ์สารกึ่งตัวนำชนิดหนึ่ง ที่สามารถแปลงพลังงานไฟฟ้าเป็นพลังงานแสง การเปล่งแสงเกิดขึ้นเมื่อพาหะดายหรือปลดปล่อยพลังงานออกมาเพื่อลดระดับลงไปอยู่ที่ระดับพลังงานต่ำกว่า พลังงานที่ปลดปล่อยออกมาจะอยู่ในรูปของคลื่นแม่เหล็กไฟฟ้า และเมื่อโฟตอนของคลื่นนี้มีความยาวคลื่นในช่วงที่สายตาเรามองเห็นได้ เราก็จะเห็นว่าไดโอดสามารถเปล่งแสงออกมาได้ การปลดปล่อยพลังงานนี้เรียกว่า "Band to band Transition" จะมีโอกาสเกิดได้สูงในกรณีที่สารกึ่งตัวนำนั้นเป็น สารกึ่งตัวนำชนิด "Direct band gap semiconductor" เช่น GaAs



รูปที่ 2.1 โครงสร้างพื้นฐาน และรูปร่างของไดโอดเปล่งแสงแบบรอยต่อ P-N ของ GaAs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงลักษณะของ LED

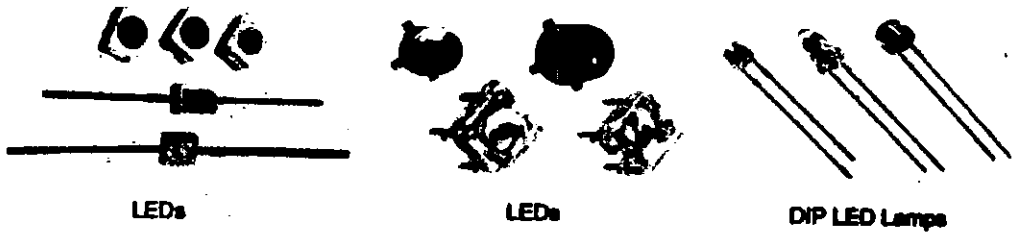
รูปที่ 2.3 แสดงโครงสร้างภายในของ LED

เงื่อนไขที่สำคัญของสารกึ่งตัวนำที่จะมาทำเป็น LED ที่เปล่งแสงในย่านที่ตามองเห็นมี 3 ประการ คือ

1. ต้องเป็นสารกึ่งตัวนำประเภท Direct band gap
2. ขนาดของ E_G ต้องอยู่ระหว่าง 1.77eV - 3.10eV
3. สามารถทำเป็นโครงสร้างของรอยต่อ P-N ได้

จากเงื่อนไขข้างต้น LED ที่ผลิตขายกันมักสร้างมาจากสารกึ่งตัวนำประเภทอัลลอย โดยการนำสารกึ่งตัวนำประเภทสารประกอบมารวมกันเพื่อให้ได้แสงสีต่างๆ และมีคุณสมบัติตามต้องการประสิทธิภาพของไดโอดเปล่งแสงด้วยค่า η ซึ่งเรียกว่า "External efficiency" ถูกนิยามด้วยอัตราส่วนระหว่าง กำลังงานของแสงด้านเอาท์พุตกับกำลังงานไฟฟ้าทางด้านอินพุต

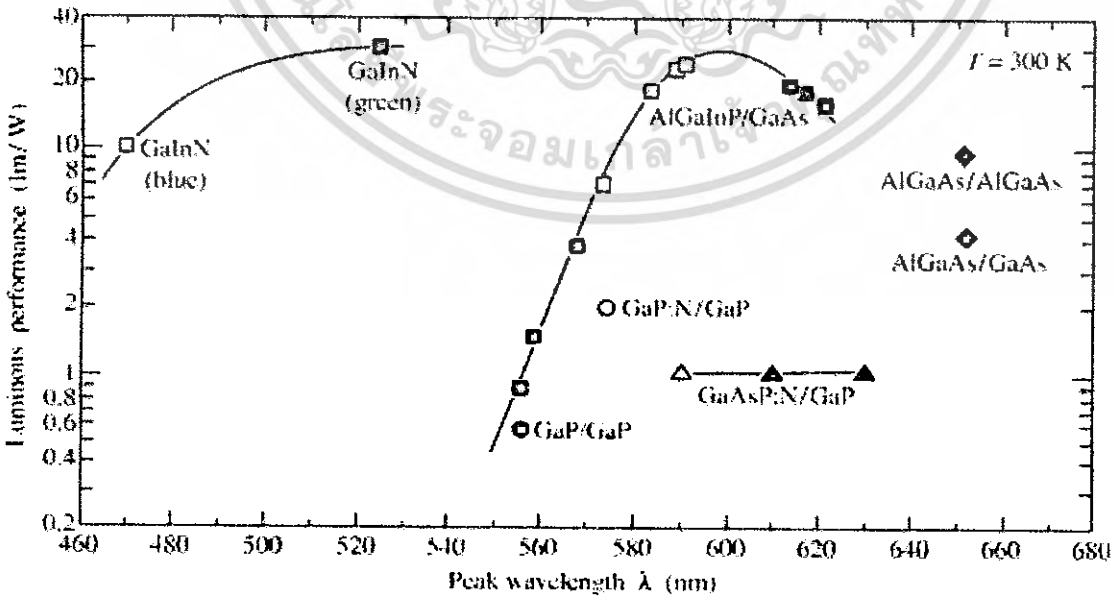
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 รูปร่างลักษณะภายนอกของไดโอดเปล่งแสงที่ผลิตจำหน่ายในท้องตลาดปัจจุบัน

การวัดความเข้มแสงแบ่งได้เป็น 2 ประเภทใหญ่ๆ คือ แบบเรดิโอมีทรี (Radiometry) และแบบโฟโต้มีทรี (Photometry) โดยการวัดแบบเรดิโอมีทรีเป็นการวัดพลังงานที่เปล่งออกมาทุกความยาวคลื่นแสงในขณะที่การวัดแบบโฟโต้มีทรีจะวัดความสว่างที่ปรากฏที่มนุษย์สามารถมองเห็นได้ด้วยตาเท่านั้น ทั่วไปตาคนเราจะมองเห็นได้ดีในช่วงความยาวคลื่นแสง (light wavelengths) จาก 380nm – 740nm

หน่วยวัดความเข้มแสงนั้นในปัจจุบันมีอยู่ด้วยกันหลายหน่วย แต่ที่พบมากคือ หน่วยลูเมนส์ (Lumens) โดยหน่วยนี้จะวัดคลื่นแสงทุกความยาวคลื่น อีกหน่วยคือ หน่วยแคนเดลา (Candelas) หรือ cd หน่วยนี้วัดเฉพาะความเข้มแสงที่ตามองเห็นได้เท่านั้น ดังนั้นโดยทั่วไปสำหรับการวัดความเข้มแสงของจอแสดงผลใดๆจึงมักใช้หน่วย cd หรือ mcd ในการบอกถึงปริมาณของแสงที่ผู้ใช้หรือผู้มองภาพจะสามารถมองเห็น



รูปที่ 2.5 แสดงการมองเห็นแสง ของ LED ในย่านความยาวคลื่นต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โปรแกรม Visual Basic

โปรแกรม Visual Basic เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่างๆ ที่เรียกกันว่า คอนโทรล(Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น

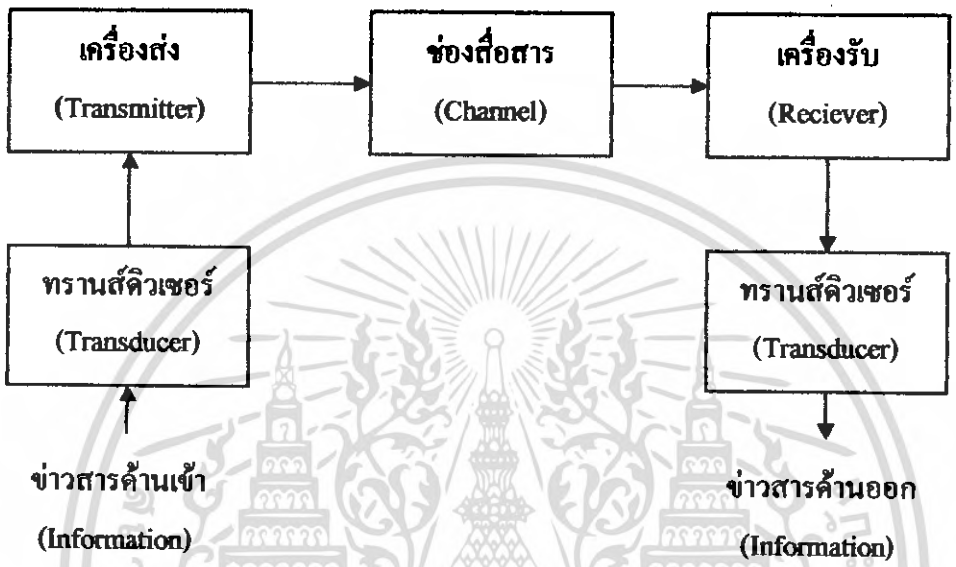
เครื่องมือ หรือ คอนโทรลต่างๆที่ Visual Basic ได้เตรียมไว้ให้ ไม่ว่าจะเป็น Form, TextBox, Label ฯลฯ ถือว่าเป็นออบเจ็กต์ (Object) นั่นหมายความว่า ไม่ว่าจะเป็นเครื่องมือใด ๆ ใน Visual Basic จะเป็นออบเจ็กต์ทั้งสิ้น สามารถที่จะควบคุมการทำงาน แก้ไขคุณสมบัติของออบเจ็กต์นั้นได้โดยตรง ในทุกๆ ออบเจ็กต์จะมีคุณสมบัติ (properties) และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจ็กต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือต่างกันก็ได้ ขึ้นอยู่กับชนิดของออบเจ็กต์

ในการพัฒนาโปรแกรมประยุกต์ด้วย Visual Basic การเขียนโค้ดจะถูกแบ่งออกเป็นส่วนๆ เรียกว่า โพรซีเดอร์ (procedure) แต่ละโพรซีเดอร์จะประกอบไปด้วย ชุดคำสั่งที่พิมพ์เข้าไปแล้ว ทำให้คอนโทรลหรือออบเจ็กต์นั้นๆ ตอบสนองการกระทำของผู้ใช้ ซึ่งเรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) แต่ตัวภาษา Visual Basic ยังไม่ถือว่าเป็นการเขียนโปรแกรมแบบ OOP อย่างแท้จริง เนื่องจากข้อจำกัดหลายๆ อย่างที่ Visual Basic ไม่สามารถทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิเล็กทรอนิกส์ (Electronic Communication Systems)

การสื่อสารอิเล็กทรอนิกส์ หมายถึง การส่ง (Transmission), การรับ (Reception) และการประมวลผลของข้อมูลหรือข่าวสาร (Information) ระหว่าง 2 จุดหรือมากกว่าด้วยการใช้อิเล็กทรอนิกส์



รูปที่ 2.6 แบบจำลองระบบสื่อสาร

จากแบบจำลองระบบสื่อสารดังรูปที่ 2.6 พิจารณาเริ่มจากทางซ้ายมือ ข่าวสารด้านเข้าที่ป้อนเข้าสู่ระบบมีลักษณะได้หลายรูปแบบ ทั้งที่เป็นสัญญาณ Analog หรือ Digital เช่น เสียง ดนตรี ความดัน อุณหภูมิ ภาพ ข้อความตัวอักษร เป็นต้น สัญญาณเหล่านี้จะถูกแปลงให้อยู่ในรูปสัญญาณที่เหมาะสมในการส่งผ่านระบบสื่อสาร โดยผ่านระบบสื่อสาร โดยอาศัยอุปกรณ์ที่เรียกว่า ทรานส์ดิวเซอร์ (Transducer) ยกตัวอย่างเช่นระบบโทรศัพท์ สัญญาณเสียงจะถูกแปลงให้อยู่ในรูปของสัญญาณไฟฟ้าโดยใช้ไมโครโฟน ส่วนถัดมาของแบบจำลองระบบสื่อสารคือเครื่องส่ง (Transmitter) อุปกรณ์ส่วนนี้มีหน้าที่ในการแปลงสัญญาณที่ออกจากทรานส์ดิวเซอร์ให้เหมาะสมกับช่องสื่อสารที่จะส่งออก กระบวนการที่ใช้ในการแปลงสัญญาณนี้เรียกว่า การมอดูเลต (modulation) การมอดูเลตสัญญาณมีได้หลายรูปแบบ เช่น การมอดูเลตเชิงแอมพลิจูด การมอดูเลตเชิงความถี่ และการมอดูเลตเชิงเฟส ส่วนถัดมาของแบบจำลอง คือ ช่องสื่อสาร (Channel) ในส่วนนี้ก็คือสื่อนำสัญญาณที่พาข่าวสารจากแหล่งกำเนิดไปสู่ปลายทางภาครับ สื่อที่ใช้มีตั้งแต่แบบที่มีสายนำสัญญาณ เช่น สายใยแก้วนำแสง สายคู่ตีเกลียว สายโคแอกซ์ ไปจนถึงการส่งสัญญาณประเภทไม่มีสายสัญญาณ แต่การแพร่ของคลื่นแม่เหล็กไฟฟ้าออกจากสายอากาศไปในบรรยากาศ ช่องสัญญาณส่วนใหญ่จะได้รับผลระทบจาสัญญาณรบกวน และในสภาพที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณรบกวนมีความรุนแรงมากอาจจะส่งผลให้ข่าวสารที่ส่งผ่านในช่องสัญญาณเกิดความผิดเพี้ยนไปจากเดิม เมื่อสัญญาณได้รับการส่งจนถึงจุดหมายปลายทาง สัญญาณเหล่าจะได้รับการแปลงกลับโดยเครื่องรับ (reciever) กระบวนการแปลงสัญญาณขั้นตอนเรียกว่า กระบวนการดีมอดูเลต (demodulation) จากสัญญาณที่ได้จะถูกแปลงกลับคืนให้กลายเป็นข่าวสารที่มีรูปแบบตามต้องการ โดยใช้อุปกรณ์ทรานส์ดิวเซอร์ ยกตัวอย่างเช่น ลำโพงซึ่งทำหน้าที่แปลงสัญญาณไฟฟ้าให้กลายเป็นสัญญาณเสียง

เครื่องส่ง (Transmitter)

เป็นอุปกรณ์หรือวงจรทางอิเล็กทรอนิกส์ที่ถูกออกแบบสำหรับแปลงสัญญาณจากแหล่งกำเนิดสัญญาณที่จะสื่อสารให้กลายเป็นสัญญาณที่มีรูปแบบ และระดับพลังงานที่เหมาะสมกับตัวกลางการสื่อสารของแต่ละระบบ

เครื่องส่งอาจเป็นคีย์สวิตช์ของระบบโทรเลขแบบใช้สายหรืออาจจะเป็นวงจรอิเล็กทรอนิกส์สลับซับซ้อนของระบบสื่อสารดาวเทียม สัญญาณที่จะสื่อสารอาจอยู่ในรูปของสัญญาณเสียงหรือภาพหรือข้อมูลในรูปของสัญญาณดิจิตอล ซึ่งสัญญาณแต่ละชนิดจะมีค่าความกว้างของแถบความถี่ (Bandwidth) แตกต่างกัน ซึ่งความกว้างของแถบความถี่ของสัญญาณนี้เป็นพารามิเตอร์ที่สำคัญที่สุดในการพิจารณาเลือกใช้หรือออกแบบระบบสื่อสาร

เครื่องรับ (Receiver)

จะเป็นอุปกรณ์และวงจรอิเล็กทรอนิกส์อีกชุดหนึ่ง ซึ่งจะทำหน้าที่แปลงสัญญาณที่รับมาได้จากตัวกลาง ให้กลายเป็นสัญญาณที่มีรูปแบบและระดับพลังงานที่เหมาะสมกับอุปกรณ์ปลายทางด้านรับ เช่น เครื่องรับของระบบโทรเลขใช้สายเป็นเพียงขดลวดโซลีนอยด์ (Solenoid) หรือวงจรอิเล็กทรอนิกส์ที่ซับซ้อนของการรับสัญญาณโทรทัศน์ผ่านดาวเทียม

สัญญาณทางอิเล็กทรอนิกส์เกือบทุกสัญญาณจะเกิดจากผลรวมของคลื่นซายน์หลาย ๆ ความถี่ การเดินทางของสัญญาณในวงจรจากจุดหนึ่งไปยังอีกจุดหนึ่ง หรือการเดินทางของสัญญาณจากวงจรหนึ่งไปยังอีกวงจรหนึ่ง หรือการเดินทางของสัญญาณจากระบบหนึ่งไปยังอีกระบบหนึ่ง สามารถพิจารณาเป็นการเดินทางของคลื่นซายน์ทุกความถี่ที่มีอยู่ในสัญญาณนั้นจากที่หนึ่งไปยังอีกที่หนึ่งในรูปแบบของคลื่นแม่เหล็กไฟฟ้า (Electromagnetic waves) ซึ่งอาจจะเป็นการเคลื่อนที่ผ่านตัวนำไฟฟ้าหรือตัวอุปกรณ์ต่าง ๆ และการเคลื่อนที่ผ่านอากาศหรืออวกาศในรูปของการแผ่รังสี (Radiation)

ความกว้างแถบความถี่ของสัญญาณ คือ ค่าแถบความถี่ของคลื่นซายน์หลายความถี่ที่รวมกันเป็นสัญญาณ ซึ่งจะมีค่าเท่ากับผลต่างของความถี่สูงสุดกับความถี่ต่ำสุดที่มีอยู่ในสัญญาณ

ถ้ามีสัญญาณมากกว่าหนึ่งสัญญาณในอากาบริเวณเดียวกัน และสัญญาณเหล่านั้นมีค่าแถบความถี่ที่ทับซ้อนกัน จะทำให้เกิดการรบกวนซึ่งกันและกัน ถ้าระดับกำลังของสัญญาณต่าง ๆ ที่มีแถบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ซ้อนทับกันมีค่าใกล้เคียงกัน ก็จะเกิดการรบกวนซึ่งกันและกัน แต่ถ้าระดับกำลังของสัญญาณแตกต่างกันมาก สัญญาณที่มีกำลังมากแทบไม่ถูกรบกวนจากสัญญาณที่มีกำลังต่ำกว่ามาก

ในทางตรงกันข้ามสัญญาณที่มีกำลังต่ำจะถูกสัญญาณที่มีกำลังสูงกว่ากลบจนหมดในระบบสื่อสาร เราไม่สามารถส่งสัญญาณที่มีแถบความถี่ซ้อนทับกันผ่านตัวกลางของการสื่อสารเดียวกันภายในเวลาเดียวกันได้ สัญญาณใดก็ตามที่มีแถบความถี่ซ้อนทับกับสัญญาณที่เราต้องการสื่อสาร จะถูกเรียกว่า "สัญญาณรบกวน (NOISE)"

การเดินทางของสัญญาณจากเครื่องส่งไปยังเครื่องรับจะถูลดทอนให้มีกำลังต่ำลงในขณะที่ผ่านตัวกลาง เมื่อสัญญาณมีกำลังไฟฟ้าลดลง จะมีโอกาสถูกรบกวนจากสัญญาณรบกวนที่อยู่ระหว่างเส้นทาง ยิ่งไปกว่านั้นภายในตัวอุปกรณ์อิเล็กทรอนิกส์ก็เป็นแหล่งกำเนิดสัญญาณรบกวนอีกด้วย ถ้าความกว้างแถบความถี่ของสัญญาณยิ่งกว้างก็จะยิ่งเพิ่มโอกาสที่จะถูกรบกวนมากยิ่งขึ้น

แถบความถี่ของคลื่นแม่เหล็กไฟฟ้า

จุดประสงค์ของ Electronic Communication คือ การติดต่อสื่อสารระหว่างจุด 2 จุด ข้อมูลข่าวสารของการสื่อสารจะต้องอยู่ในรูปของคลื่นแม่เหล็กไฟฟ้า ซึ่งคลื่นแม่เหล็กไฟฟ้านี้จะสามารถแผ่กระจายผ่านตัวนำไฟฟ้า อากาศ หรืออวกาศได้

แถบความถี่ของสัญญาณ (Signal Bandwidth) คือ แถบความถี่ของคลื่นแม่เหล็กไฟฟ้าที่สัญญาณครอบครองอยู่ ซึ่งจะมีค่าเท่ากับผลต่างของความถี่สูงสุดกับความถี่ต่ำสุดของสัญญาณ

แถบความถี่ทั้งหมดของคลื่นแม่เหล็กไฟฟ้าที่สามารถนำมาใช้ประโยชน์ในการสื่อสาร ได้ถูกแบ่งเป็นแถบความถี่ต่างๆ และตั้งชื่อโดย CCIR (The Internatoinal Radio Consulative Committee) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แถบความถี่และการใช้งาน

แถบความถี่	ชื่อแถบความถี่	การใช้งาน
3 - 30 kHz	ความถี่ต่ำมาก (Very low frequency : VLF)	ระบบนำวิถีระยะไกล
30 - 300 kHz	ความถี่ต่ำ (Low frequency : LF)	ระบบนำวิถี
300 - 3,000 kHz	ความถี่ปานกลาง (Medium frequency : MF)	ระบบวิทยุ
3 - 30 MHz	ความถี่สูง (High frequency : HF)	ระบบโทรเลข ระบบโทรศัพท์ ระบบสื่อสารการเดินเรือ
30 - 300 MHz	ความถี่สูงมาก (Very High frequency : VHF)	ระบบแพร่ภาพโทรทัศน์ VHF ระบบวิทยุ FM ระบบควบคุมการจราจรทางอากาศ ระบบสื่อสารของตำรวจ
0.3 - 3 GHz	ความถี่สูงยิ่ง (Ultrahigh frequency : UHF)	ระบบแพร่ภาพโทรทัศน์ UHF ระบบโทรศัพท์เคลื่อนที่ ระบบเรดาร์ความปลอดภัย ระบบสื่อสาร LAN ชนิดไร้สาย
3 - 30 GHz	ความถี่สูงยวดยิ่ง (Superhigh frequency : SHF)	ระบบสื่อสารไมโครเวฟ ระบบสื่อสารดาวเทียม ระบบเรดาร์พยากรณ์อากาศ
30 - 300 GHz	ความถี่สูงสุดโต่ง (Extremely high frequency : EHF)	ระบบเรดาร์ ระบบการลงจอดของเครื่องบิน
300 GHz – 3 THz	(0.1 – 1 mm)	
43 - 430 THz	อินฟราเรด (Infrared 7 – 0.7 μm)	ระบบสื่อสารทางแสง
430 – 750 THz	แสงที่ตามองเห็น (Visible light 0.7 – 0.4 μm)	ระบบสื่อสารทางแสง
750 – 3,000 THz	แสงอัลตราไวโอเลต (Ultraviolet light 0.4 – 0.1 μm)	ระบบสื่อสารทางแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความกว้างของแถบความถี่ของช่องสื่อสาร (Communication Channel Bandwidth) จะต้องมีความกว้างพอที่จะให้ความถี่ที่มีนัยสำคัญทั้งหมดของสัญญาณข้อมูลข่าวสารผ่านไปได้ทั้งหมด โดยที่ความกว้างของแถบความถี่ของช่องสื่อสาร : BW_{CH} จะต้องมีค่าเท่ากับหรือมากกว่าความกว้างของแถบความถี่ของช่องสัญญาณ :BW_{signal}

$$BW_{CH} \geq BW_{signal}$$

เช่น แถบความถี่ของเสียงมนุษย์ คือ 300 Hz – 3 kHz ความกว้างของแถบความถี่ของช่องสื่อสารสำหรับเสียงมนุษย์ จะต้องมีความกว้างอย่างน้อยที่สุดเป็น 2.7 kHz (3 kHz – 300 Hz = 700 Hz)

ความจุของข้อมูลข่าวสาร (Information Capacity) ของระบบสื่อสารเป็นจำนวนที่ใช้วัดความสามารถของระบบสื่อสารว่า สามารถจะส่งข้อมูลข่าวสารผ่านระบบได้เป็นประมาณเท่าใดต่อหนึ่งหน่วยของเวลา จะเห็นได้อย่างชัดเจนว่า ระบบที่มีความกว้างของแถบความถี่ของช่องสื่อสารกว้าง ก็จะยิ่งมีความจุของข้อมูลข่าวสารสูง

กฎของ "ฮาร์ทเลย์" (Hartley's law) กล่าวว่า

$$I \propto BW_{CH} \times t$$

โดยที่ I คือ ความจุของข้อมูลข่าวสาร (Information Capacity)

BW_{CH} คือ ความกว้างของแถบความถี่ของช่องสื่อสาร

t คือ เวลาที่ใช้ในการส่ง

ถึงแม้ว่าระบบสื่อสารที่มีความกว้างของแถบความถี่ของช่องสื่อสารกว้าง จะมีความจุของข้อมูลข่าวสารมาก อย่างไรก็ตาม กฎพื้นฐานในการออกแบบระบบสื่อสารที่วิศวกรผู้ออกแบบพึงระลึกอยู่ตลอดเวลา คือ จะต้องพยายามทำให้ความกว้างของแถบความถี่ของช่องสื่อสารมีความกว้างให้น้อยที่สุดเท่าที่จะเป็นไปได้ โดยไม่ทำให้สูญเสียข้อมูลข่าวสารที่มีนัยสำคัญ เพื่อให้มีจำนวนของช่องสื่อสารให้มากที่สุดเท่าที่จะเป็นไปได้ ซึ่งจะเป็นการเปิดโอกาสให้คนจำนวนมากได้ใช้ประโยชน์จากแถบความถี่ของคลื่นแม่เหล็กไฟฟ้าซึ่งเป็นทรัพยากรที่มีจำกัดร่วมกัน

ความถี่บางความถี่ เช่น ความถี่ที่สูงกว่าย่าน HF (VHF, UHF, ...) จะสามารถแพร่กระจายได้ในขอบเขตจำกัดเพราะการแพร่กระจายของคลื่นย่านความถี่ VHF หรือ สูงกว่าจะเป็นลักษณะเส้นตรง หรือเรียกว่าระดับเส้นสายตา การแพร่กระจายคลื่นแบบนี้จะถูกส่วนโค้งของโลกบัง เราจึงสามารถใช้ความถี่ของระบบสื่อสารซ้อนทับกันในบริเวณที่ห่างไกลกัน สัญญาณของระบบสื่อสารก็จะไม่รบกวนกัน การจำกัดบริเวณของการแพร่กระจายคลื่นอาจทำได้โดยไม่ใช้เครื่องส่งที่มีกำลังเกินกว่าความจำเป็น

แต่ความถี่บางความถี่ เช่น ในย่าน HF จะมีความสามารถสะท้อนกับบรรยากาศของโลก (ชั้น Ionozsphere) ได้ดี จะสามารถแพร่กระจายคลื่นไปได้ไกลมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

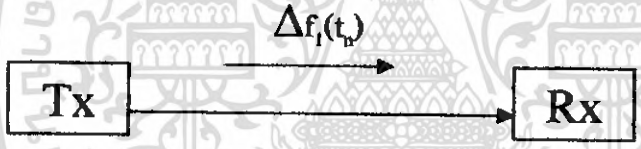
รูปแบบของการสื่อสาร

เราสามารถแบ่งรูปแบบของการสื่อสารได้หลายรูปแบบหรือหลายชนิด เช่น แบ่งตามลักษณะของตัวกลางก็จะสามารถแบ่งได้เป็น 2 แบบ คือ แบบมีสายกับแบบไร้สาย ถ้าแบ่งตามรูปแบบของสัญญาณก็จะสามารถแบ่งได้ 2 แบบ คือ การสื่อสารแบบแถบฐานกับการสื่อสารแบบมอดดูเลชั่น

ไม่ว่าจะเป็นแบบใดในการส่งสัญญาณจำนวน n สัญญาณ ผ่านระบบสื่อสารในเวลาเดียวกันและผ่านตัวกลางเดียวกัน จะต้องใช้ช่องสื่อสารจำนวน n ช่องสื่อสาร แถบความถี่ของช่องสื่อสารแต่ละช่องจะซ้อนทับกันไม่ได้ และความกว้างของแถบความถี่ของช่องสื่อสารแต่ละช่องควรมีความกว้างให้น้อยที่สุดเท่าที่จะทำได้ โดยต้องไม่สูญเสียข้อมูลที่มีนัยสำคัญ

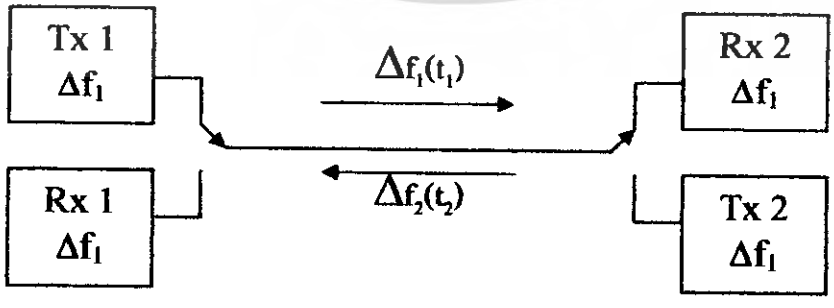
เราอาจแบ่งรูปแบบของการสื่อสารตามทิศทางของการสื่อสารหรือจำนวนของช่องสื่อสารได้ 3 แบบ คือ

1. **แบบทิศทางเดียว (SIMPLEX ; SPX, XS)** คือ ฝ่ายหนึ่งจะส่งเพียงอย่างเดียวและอีกฝ่ายหนึ่งจะเป็นฝ่ายรับเพียงอย่างเดียว การสื่อสารแบบทิศทางเดียวนี้ต้องการช่องสื่อสารเพียง 1 ช่อง ตัวอย่าง เช่น การส่งวิทยุกระจายเสียง (Broad casting)



รูปที่ 2.7 การสื่อสารแบบทิศทางเดียว

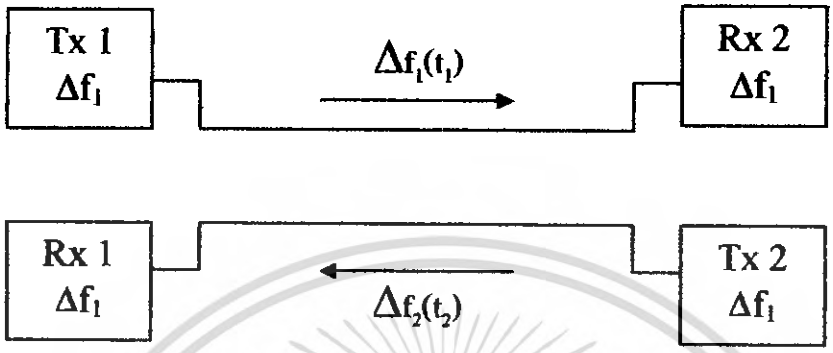
2. **แบบกึ่งสองทาง (HALF DUPLEX: HDX, HX)** เป็นการสื่อสารแบบ 2 ทางแต่สลับเวลากันในขณะที่ฝ่ายหนึ่งเป็นฝ่ายส่ง อีกฝ่ายหนึ่งจะเป็นฝ่ายรับ การสื่อสารแบบกึ่งสองทางนี้ต้องการช่องสื่อสารเพียงช่องเดียว ตัวอย่าง เช่น วิทยุโทรคมนาคม หรือ Walkie – Talkie



รูปที่ 2.8 การสื่อสารแบบกึ่งสองทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แบบสองทาง (FULL DUPLEX; FDX, FX) เป็นการสื่อสารแบบ 2 ทิศทาง โดยที่ทั้ง 2 ฝ่าย สามารถรับและส่งได้พร้อมกันในเวลาเดียวกัน การสื่อสารแบบ 2 ทิศทางนี้ต้องการช่องสื่อสาร 2 ช่องในเวลาเดียวกัน ตัวอย่างเช่น โทรศัพท์ และ โทรศัพท์เคลื่อนที่



รูปที่ 2.9 การสื่อสารแบบสองทิศทาง

2.4 การมอดูเลต

การส่งสัญญาณในระบบไฟฟ้าสื่อสารมักจะมีกระบวนการเลื่อนหรือแปลงความถี่ของสัญญาณข้อมูลให้อยู่ในช่วงความถี่ที่เหมาะสมก่อนการส่งออกผ่านสื่อนำสัญญาณ กระบวนการนี้เรียกโดยรวมว่า การมอดูเลต (modulation) เหตุผลหลักๆที่เราต้องมอดูเลตสัญญาณมีอยู่หลายประการพอจะสรุปได้ดังนี้ คือ เนื่องจากช่วงความถี่หรือสเปกตรัมของสัญญาณข้อมูลแต่ละประเภทที่มนุษย์ต้องการส่งนั้นทับซ้อนกันอยู่ เช่น สัญญาณเสียงพูดมีความถี่อยู่ระหว่าง 0 – 4 kHz สัญญาณเสียงดนตรีประกอบด้วยความถี่ในย่าน 0 – 20 kHz สัญญาณโทรทัศน์ครอบคลุมความถี่ตั้งแต่ 0 – 5 MHz ดังนั้นหากต้องการส่งสัญญาณทั้งหมดนี้ผ่านสื่อสัญญาณเดียวกันโดยไม่ผ่านกระบวนการมอดูเลต สัญญาณเหล่านี้จะทับกันและรบกวนกันอย่างหลีกเลี่ยงไม่ได้ นอกจากนี้ในระบบแพร่สัญญาณโทรทัศน์หรือวิทยุโดยทั่วไปจะมีการส่งสัญญาณออกมากกว่าหนึ่งช่องในเวลาเดียวกัน ดังนั้นการเลื่อนความถี่ของสัญญาณข้อมูลจากแต่ละแหล่งให้ไปอยู่ในช่วงความถี่ที่สูงขึ้นและไม่ทับซ้อนกันก็จะช่วยแก้ปัญหาที่กล่าวมานี้ได้

เหตุประการที่สอง ในการแผ่กระจายของสัญญาณคลื่นแม่เหล็กไฟฟ้า สายอากาศที่ใช้ในการแพร่กระจายสัญญาณมักจะต้องมีขนาดที่ใกล้เคียงกับขนาดของความยาวคลื่นของสัญญาณที่ส่งออก ยกตัวอย่างเช่น หากเราส่งสัญญาณที่ความถี่ 4,000 Hz ซึ่งเมื่อคิดเป็นความยาวคลื่นได้เท่ากับ 75,000 เมตร จะเห็นว่าสายอากาศที่ใช้มีขนาดใหญ่มากเกินกว่าที่จะเป็นประโยชน์ในการนำมาใช้งานในทางปฏิบัติ แต่หากเราเลื่อนความถี่ให้เพิ่มขึ้นเป็น 101.5 MHz ก่อนการแพร่สัญญาณ สายอากาศที่ใช้จะมีขนาดที่เหมาะสมในการนำมาใช้งาน เพราะความยาวคลื่นที่ความถี่นี้มีค่าเพียง 2.95 เมตร นอกจากนี้การมอดูเลตสัญญาณยังสามารถช่วยลดหรือหลีกเลี่ยงผลกระทบของสัญญาณรบกวนได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การมอดดูเลตสามารถแบ่งได้เป็น 2 ประเภทใหญ่ๆ คือ

1. การมอดูเลตแบบ Analog
2. การมอดูเลตแบบ Digital

การมอดูเลตสัญญาณแบบ Analog

การมอดูเลตสัญญาณแบบ Analog สามารถแบ่งออกได้เป็น 2 ประเภทหลัก คือ การมอดูเลตเชิงแอมพลิจูด (Amplitude modulation) และการมอดูเลตเชิงมุม (Angle modulation)

1. การมอดูเลตเชิงแอมพลิจูด คือการแปรเปลี่ยนแอมพลิจูดของสัญญาณคลื่นพาห์ไปตามขนาดของสัญญาณข้อมูลที่จะส่ง การมอดูเลตประเภทนี้สามารถแบ่งแยกเป็นวิธีย่อยได้ 4 วิธี คือ

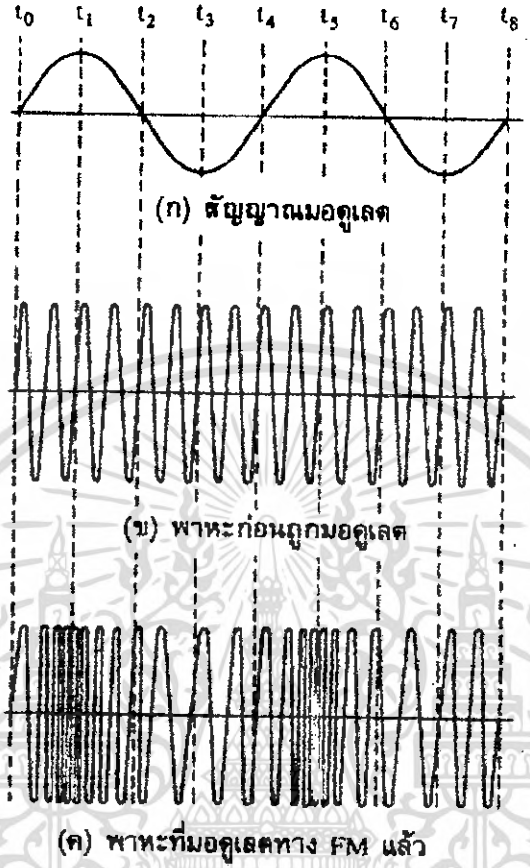
1. การมอดูเลตแบบ AM
2. การมอดูเลตแบบ DSB-SC (Double sideband-suppressed carrier)
3. การมอดูเลตแบบ SSB (Single sideband)
4. การมอดูเลตแบบ VSB (Vestigial sideband)

2. การมอดูเลตเชิงมุม คือแปรเปลี่ยนมุมของสัญญาณคลื่นพาห์ (carrier) ไปตามขนาดของสัญญาณข้อมูลที่จะส่ง ซึ่งประกอบด้วยการมอดูเลตแบบ FM (Frequency modulstion) และการมอดูเลตแบบ PM (Phase modulation)

การมอดูเลตเชิงความถี่

รูปคลื่นของสัญญาณ FM เกิดจากสัญญาณมอดูเลตดังรูปที่ 2.10 (ก) เช่น สัญญาณเสียงซึ่งเป็นข่าวสารเข้าไปมอดูเลตลงบนสัญญาณคลื่นพาห์ดังรูปที่ 2.10 (ข) สัญญาณคลื่นพาห์หลังจากถูกมอดูเลตแล้วดังรูปที่ 2.10 (ค) เป็นสัญญาณ FM จะเห็นว่าที่เวลา t_0 สัญญาณ FM อยู่ที่ความถี่กลาง เมื่อสัญญาณที่เข้ามามอดูเลตมีค่าทางบวกสูงสุด ความถี่คลื่นพาห์จะเพิ่มขึ้นสูงสุดด้วย นั่นคือสัญญาณมอดูเลตถึงจุดสุดยอด (สัญญาณมอดูเลตมีขนาดสูงสุดนั่นเอง) ที่เวลา t_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 การมอดูเลตทางความถี่

ที่เวลา t_2 สัญญาณมอดูเลตลดลงมีค่าเป็นศูนย์ ความถี่ของคลื่นพาห์ก็จะลดลงมาที่ความถี่กลางดังเดิมหลังจากเวลาของสัญญาณมอดูเลตมีค่าตกลงต่ำกว่าศูนย์จนกลายเป็นลบ พาหะจะมีความถี่ลดลงต่ำกว่าความถี่กลาง และเมื่อเวลาสัญญาณมอดูเลตกลับเป็นศูนย์อีกครั้งหนึ่ง ความถี่ของคลื่นพาห์ก็จะกลับมายังความถี่กลางดังเดิม เช่นกัน ในช่วงเวลา t_4 ถึง t_8 ก็จะซ้ำแบบเดิมเรื่อยๆ ไป สรุปแล้วความถี่ของคลื่นพาห์จะเปลี่ยนแปลงไปตามแอมพลิจูดของสัญญาณมอดูเลต และสัญญาณคลื่นพาห์ยังคงอยู่ที่ความถี่กลางเมื่อสัญญาณมอดูเลตเป็นศูนย์

ช่วงความถี่คลื่นพาห์เบี่ยงเบนไปจากความถี่กลางเรียกว่า ความถี่เบี่ยงเบน (frequency deviation) หรือ "ดีวิเอชั่น" ตัวอย่างเช่น คลื่นพาห์มีความถี่ 100 MHz ลดลงต่ำสุดเป็น 99MHz และเพิ่มขึ้นสูงสุดเป็น 101 MHz สลับไปมาเช่นนี้ หมายความว่าช่วงความถี่เบี่ยงเบนเท่ากับ $\pm$ 0.1 MHz หรือ $\pm$ 100 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการเบี่ยงเบนความถี่ของสัญญาณ FM ขึ้นอยู่กับความถี่ของสัญญาณที่เข้ามามอดูเลต ตัวอย่างเช่น ถ้าสัญญาณที่เข้ามามอดูเลตเป็นโทน (สัญญาณเสียง) ความถี่ 1 kHz อัตราการเบี่ยงเบนความถี่ของสัญญาณ FM จะเท่ากับ 1,000 ครั้งต่อวินาที ถ้าสัญญาณที่เข้ามามอดูเลตเพิ่มความถี่เป็น 10 kHz โดยคงค่าแอมพลิจูดเท่าเดิม ช่วงความถี่เบี่ยงเบนก็จะยังคงเท่าเดิม คือเท่า +100 kHz หรือ –100 kHz แต่อัตราการเบี่ยงเบนจะเพิ่มเป็น 1,000 ครั้งต่อวินาที นั่นคือ ความถี่ของสัญญาณที่เข้ามามอดูเลตเป็นตัวกำหนดอัตราการเบี่ยงเบนความถี่

สำหรับแอมพลิจูดของสัญญาณมอดูเลต จะเป็นตัวกำหนดช่วงความถี่เบี่ยงเบน ตัวอย่างเช่น สัญญาณโทนที่มีแอมพลิจูดสูงสุดจะทำให้ความถี่เบี่ยงเบนไป +100 kHz หรือ –100 kHz สัญญาณโทนที่มีแอมพลิจูดน้อยลงจะทำให้ความถี่เบี่ยงเบนไป +50 kHz หรือ -50 kHz

กล่าวโดยสรุป สัญญาณ FM มีคุณสมบัติที่สำคัญดังนี้

1. มีแอมพลิจูดคงที่ตลอด แต่ความถี่เปลี่ยนตามสัญญาณที่เข้ามามอดูเลต
2. อัตราการเบี่ยงเบนความถี่ของสัญญาณพาหะจะมีค่าเท่ากับความถี่ของสัญญาณที่เข้ามามอดูเลต
3. ช่วงความถี่ที่เบี่ยงเบน (หรือดีวิเอชั่น) เป็นสัดส่วนกับแอมพลิจูดของสัญญาณที่เข้ามามอดูเลต

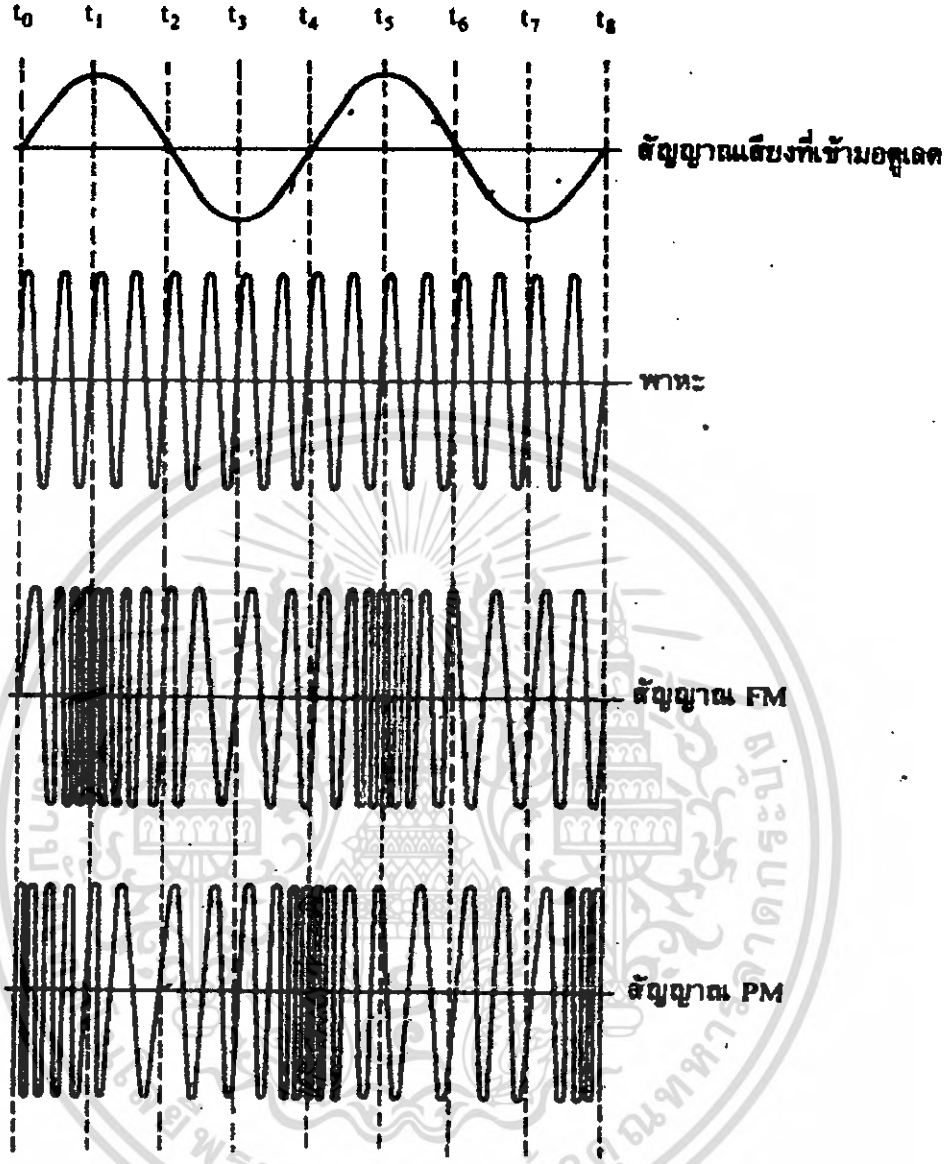
การมอดูเลตทางเฟส

ในการมอดูเลตทางเฟส เฟสของพาหะย่อมเปลี่ยนแปลงตามสัญญาณที่เข้ามามอดูเลต เมื่อเฟสของสัญญาณพาหะเปลี่ยนแปลง ความถี่ของคลื่นพาห์ก็จะเปลี่ยนแปลงตามไปด้วย ฉะนั้นการมอดูเลตทางเฟสจะทำให้เกิดความถี่เบี่ยงเบนด้วยเสมอ จริงๆแล้วในการมอดูเลตทางความถี่ เฟสก็จะเปลี่ยนแปลงตามไปด้วย สรุปแล้วการเปลี่ยนแปลงทั้งสองแบบนั้นจะเกิดขึ้นพร้อมๆ กัน

รูปที่ 2.11 แสดงการเปรียบเทียบสัญญาณ FM กับ PM จะเห็นว่าสัญญาณทั้งสองจะมีลักษณะเหมือนกันทุกประการ เว้นแต่ในการมอดูเลตทางเฟส ความถี่เบี่ยงเบนมีค่าเป็นสัดส่วนโดยตรงกับการเปลี่ยนเฟสและแอมพลิจูดของสัญญาณที่เข้ามามอดูเลต ด้วยเหตุนี้การมอดูเลตทางเฟส ความถี่ที่เบี่ยงเบนมากที่สุดในขณะที่สัญญาณมอดูเลตผ่านศูนย์ (แกนศูนย์) กล่าวคือ เฟสของพาหะมีการเปลี่ยนแปลงไปมากที่สุดในขณะที่สัญญาณมอดูเลทเปลี่ยนจากบวกเป็นลบหรือลบเป็นบวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.11 ความแตกต่างระหว่างคลื่น PM กับ FM

สังเกตว่าที่เวลา t_2 สัญญาณที่เข้ามามอดูเลตผ่านศูนย์ สัญญาณ PM เบี่ยงเบนไปยังความถี่ต่ำสุด ที่เวลา t_4 สัญญาณที่เข้ามามอดูเลตผ่านศูนย์อีกครั้ง สัญญาณ PM เบี่ยงเบนไปยังความถี่สูงสุด ฉะนั้น การมอดูเลตทางเฟสจึงทำให้เกิดสัญญาณ FM ด้วยเช่นกัน บางทีเราเรียกว่าการมอดูเลตทางเฟสนี้ว่าเป็น FM โดยอ้อม (indirect FM)

ดังที่ได้กล่าวมาแล้วนั้นว่าในการมอดูเลตทางเฟส ปริมาณความถี่เบี่ยงเบนมีค่าเป็นสัดส่วนโดยตรงกับการเปลี่ยนแปลงของสัญญาณที่เข้ามามอดูเลต ด้วยเหตุนี้ความถี่จะเบี่ยงเบนไปมากที่สุดเมื่อสัญญาณที่เข้ามอดูเลตผ่านแกนศูนย์ สมมติว่าความถี่ของสัญญาณที่เข้ามามอดูเลต (คือสัญญาณเสียง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
72844

เพิ่มจาก 100 Hz เป็น 1 kHz คือเพิ่มเป็น 10 เท่าที่ความถี่ 100 Hz สัญญาณเสียงจะเปลี่ยนเฟสเท่ากับ 100 Hz × 360 องศา = 36,000 องศาต่อวินาที (เนื่องจาก 1 ไซเกิลเท่ากับ 360 องศา) ที่ความถี่ 1,000 Hz เสียงจะเปลี่ยนเฟสเพิ่มขึ้นเป็น 360,000 องศาต่อวินาที เนื่องจากการเบี่ยงเบนความถี่เป็นสัดส่วนโดยตรงกับการเปลี่ยนเฟส ฉะนั้นความถี่คลื่นพาห์ย่อมเบี่ยงเบนมากขึ้น เมื่อความถี่ของสัญญาณเสียงที่เข้ามามอดูเลตสูงขึ้น

ข้อแตกต่างระหว่างสัญญาณ PM กับ FM ได้แก่

1.สัญญาณ PM มีความถี่เบี่ยงเบนเป็นสัดส่วนโดยตรงกับทั้งความถี่และแอมพลิจูดของสัญญาณที่เข้ามามอดูเลต

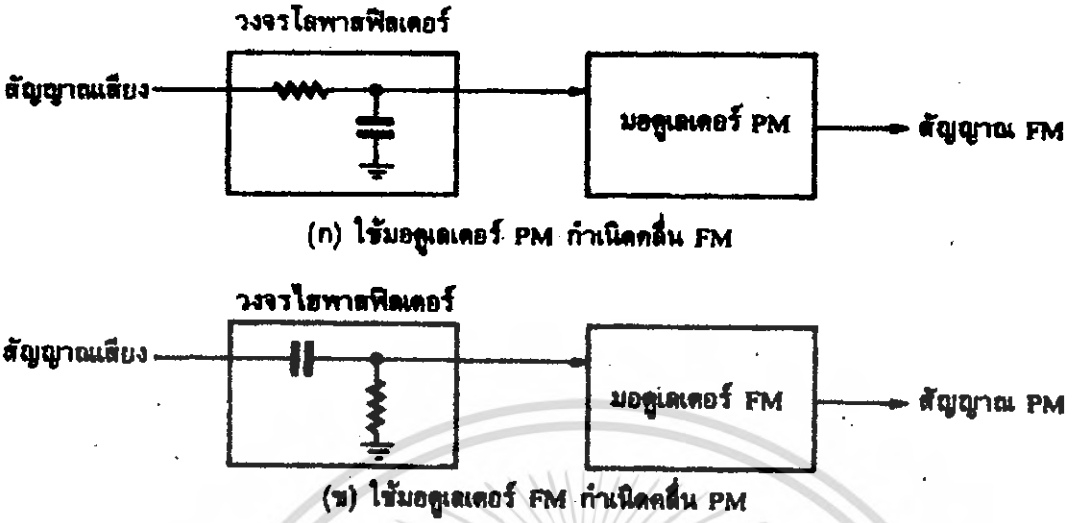
2.สัญญาณ FM มีความถี่เบี่ยงเบนที่เข้ามาเป็นสัดส่วนโดยตรงกับแอมพลิจูดของสัญญาณที่เข้ามามอดูเลตโดยไม่ขึ้นกับความถี่ของสัญญาณที่เข้ามามอดูเลต

ตารางที่ 2.2 เปรียบเทียบค่าความถี่เบี่ยงเบนและเฟสเบี่ยงเบนในระบบ FM และ PM

การมอดูเลท	ความถี่เบี่ยงเบน	เฟสเบี่ยงเบน
ระบบ FM	เป็นสัดส่วนกับแรงดันของสัญญาณมอดูเลต	เป็นสัดส่วนกับแรงดันและผกผันกับความถี่ของสัญญาณมอดูเลต
ระบบ PM	เป็นสัดส่วนกันทั้งแรงดันและสัญญาณของสัญญาณมอดูเลต	เป็นสัดส่วนกับแรงดันของสัญญาณมอดูเลต

ในความเป็นจริงแล้ว ถ้าเราใช้สัญญาณความถี่เดียวกันมอดูเลตให้กับสัญญาณคลื่นพาห์ให้มีความถี่เบี่ยงเบนเท่ากัน ทั้งสัญญาณ FM และสัญญาณ PM ก็จะมีรูปคลื่นเหมือนกันทุกประการ แต่เมื่อเปลี่ยนแปลงความถี่ของสัญญาณมอดูเลต เราจะเห็นความแตกต่างระหว่างสัญญาณ FM และ PM เพราะว่าสัญญาณ PM จะมีความถี่เบี่ยงเบนสูงขึ้น แต่สัญญาณ FM จะเบี่ยงเบนความถี่เท่าเดิม (อัตราการเบี่ยงเบนเท่านั้นที่เพิ่มขึ้น) ด้วยเหตุนี้การเปลี่ยนแปลงวงจรมอดูเลตแบบ FM เป็น PM หรือแบบ PM เป็น FM เราสามารถใช้วงจรฟิลเตอร์ RC ธรรมดาแปลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 การแปลงระหว่างคลื่น PM กับ FM

ในรูปที่ 2.12 เราใช้วงจรฟิลเตอร์ชนิด (lowpass filter) ทำหน้าที่ลดทอนแอมพลิจูดของสัญญาณที่เข้ามามอดูเลตเมื่อความถี่สูงขึ้น สัญญาณที่ได้เมื่อป้อนให้แก่วงจรเฟสมอดูเลเตอร์ก็จะกลายเป็นสัญญาณ FM ทั้งนี้ก็เพราะว่า เมื่อความถี่ของสัญญาณที่เข้ามามอดูเลตสูงขึ้น การลดทอนแอมพลิจูดก็จะทำให้ความถี่เบี่ยงเบนลดลงไปชดเชยกับความถี่เบี่ยงเบนซึ่งเพิ่มขึ้นเนื่องจากความถี่ของสัญญาณที่เข้ามามอดูเลตสูงขึ้น ทำให้คุณสมบัตินี้ก็กลายเป็นคุณสมบัติของสัญญาณ FM นั่นเอง

ในทำนองเดียวกัน ถ้าเราต้องการเปลี่ยนแปลงการมอดูเลตแบบ FM ให้ได้เป็นสัญญาณ PM เราสามารถใช้วงจรฟิลเตอร์ชนิดไฮพาส (highpass filter) เพื่อลดทอนสัญญาณความถี่ต่ำ ทำให้ความเบี่ยงเบนของสัญญาณลดลง เมื่อความถี่สัญญาณที่เข้ามามอดูเลตลดลง ก็จะได้ความถี่ซึ่งมีคุณสมบัติเหมือนกับสัญญาณ PM

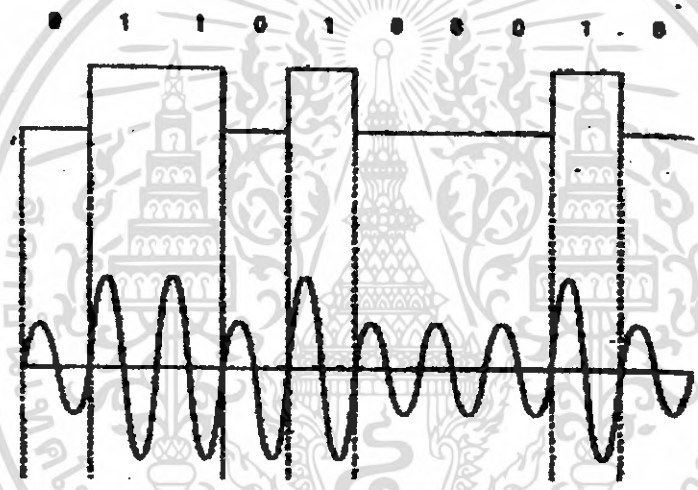
การมอดูเลตสัญญาณแบบ Digital

การมอดูเลตสัญญาณแบบ Digital คือ กระบวนการแปลงข้อมูลดิจิตอลที่อยู่ในรูป 0 และ 1 ให้ได้เป็นสัญญาณที่มีรูปลักษณ์เหมาะสมกับการส่งผ่านช่องสัญญาณโดยอาศัยคลื่นแม่เหล็กไฟฟ้า คลื่นสัญญาณที่ใช้ในการส่งผ่านช่องสัญญาณที่ใช้กันทั่วไปจะเป็นสัญญาณซายนูซอยด์ และเราเรียกสัญญาณที่เป็นเหมือนพาหนะในการนำข้อมูลข่าวสารดิจิตอลจากแหล่งกำเนิดไปถึงภาครับนี้ว่า คลื่นพาห์ (carrier) การมอดูเลตสัญญาณข่าวสารดิจิตอลลงบนคลื่นพาห์สามารถทำได้หลายลักษณะวิธีการพื้นฐานที่สำคัญและเป็นประโยชน์ในการสำมาใช้งานได้แก่ แอมพลิจูดชิฟต์คีย์อิง (ASK : Amplitude Shift Keying) ฟรีเควนซีชิฟต์คีย์อิง (FSK : Frequency Shift Keying) และเฟสชิฟต์คีย์อิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(PSK : Phase Shift Keying)

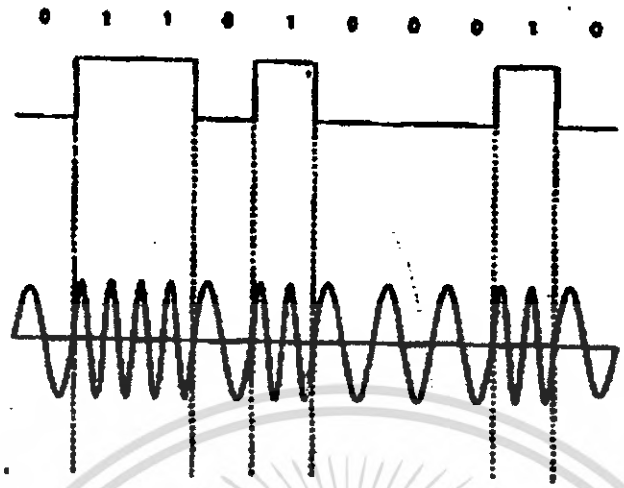
พิจารณาจากชื่อของแต่ละวิธีก็พอจะเห็นถึงความแตกต่างอย่างกว้างๆได้ดังนี้ คือ การมอดูเลต แบบ ASK อาศัยการเปลี่ยนขนาดแอมพลิจูดของคลื่นพาห์ไปตามค่าของข้อมูลดิจิตอลที่ป้อนเข้ามาดู ตัวอย่างรูปสัญญาณ ASK สำหรับข้อมูลดิจิตอลแบบไบนารี 2 ระดับในรูปที่ 2.13 สังเกตว่าในตัวอย่างนี้ ได้กำหนดให้แอมพลิจูดของคลื่นพาห์มีค่าเป็นศูนย์เพื่อแทนบิตข้อมูลที่เป็น 0 ซึ่งเทียบได้กับการไม่ส่ง สัญญาณเลย และจะส่งสัญญาณซายนูซอยด์ที่มีแอมพลิจูดไม่เป็นศูนย์เพื่อแทนบิตข้อมูลที่มีค่าเป็น 1 ในกรณีเฉพาะแบบนี้จะเรียกกรรมวิธีการมอดูเลตแบบนี้ในอีกชื่อหนึ่งว่า ออนออฟคีย์อิง (OOK : on-off keying)



รูปที่ 2.13 การมอดูเลตสัญญาณแบบ Amplitude Shift Keying

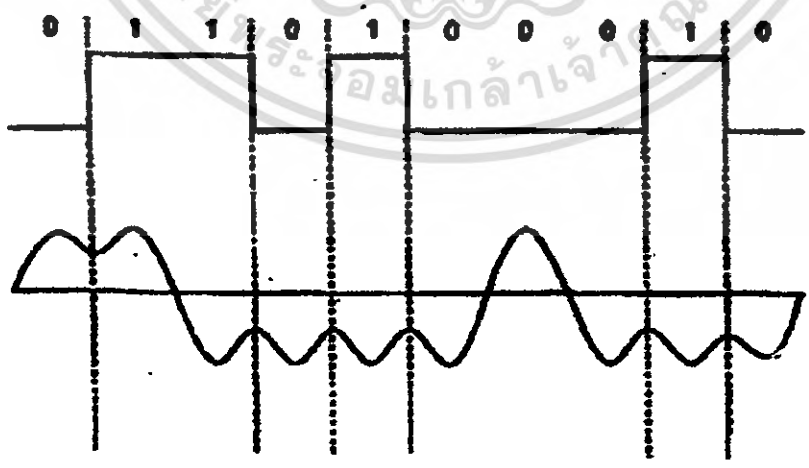
กรณีการมอดูเลตแบบ FSK แอมพลิจูดของสัญญาณคลื่นพาห์ที่ใช้จะคงที่ไม่เปลี่ยนแปลง มีเพียงความถี่ของคลื่นพาห์เท่านั้นที่เปลี่ยนแปลงไปตามข้อมูล โดยความถี่สูงใช้แทนข้อมูลดิจิตอลที่มีค่าเป็น 1 และความถี่ต่ำใช้แทนข้อมูลที่เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การมอดูเลตสัญญาณแบบ Frequency Shift Keying

ส่วนกรณีการมอดูเลตแบบ PSK แอมพลิจูดและความถี่ของคลื่นพาห์จะกำหนดให้เป็นค่าคงที่ค่าหนึ่ง โดยไม่มีการเปลี่ยนแปลงใดๆ ส่วนที่เปลี่ยนไปตามข้อมูลดิจิตอล 0 หรือ 1 คือเฟส กล่าวคือถ้าบิตข้อมูลที่จะส่งมีค่าเป็น 0 ระบบจะส่งคลื่นพาห์ที่มีเฟสตามที่กำหนดค่าหนึ่ง และถ้าบิตข้อมูลมีค่าเป็น 1 ระจะส่งคลื่นพาห์ที่มีเฟสอีกค่าหนึ่งที่ต่างไปจากกรณีแรก โดยทั่วไปถ้าต้องการให้ได้ระบบที่มีสมรรถนะดี ก็มักจะเลือกให้เฟสทั้งสองมีค่าต่างกัน 180 องศา



รูปที่ 2.15 การมอดูเลตสัญญาณแบบ Phase Shift Keying

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับตัวอย่างที่ได้ยกมานี้เป็นกรณีที่เรียนง่าย กล่าวคือรูปสัญญาณที่ใช้ในการส่งออกมีเพียง 2 รูปแบบ โดยใช้ในการแทนข้อมูลดิจิตอล 0 และ 1 แต่ในระบบสื่อสารที่ต้องการประสิทธิภาพที่ดีขึ้น มักจะมีรูปสัญญาณมากกว่า 2 แบบ ให้ใช้ และจะพิจารณาบิตข้อมูลครั้งละมากกว่าหนึ่งบิต ยกตัวอย่าง เช่น ถ้าเราพิจารณาบิตข้อมูลในการส่งคราวละ 3 บิต ระบบจะต้องมีรูปสัญญาณที่แตกต่างกันถึง $2^3 = 8$ รูปแบบ เพื่อแทนสัญลักษณ์ที่แตกต่างกันดังนี้ 000, 001, 010, 011, 100, 101, 110 และ 111

การมอดูเลตสัญญาณแบบ Digital มีหลากหลายรูปแบบ แต่ละแบบก็มีคุณลักษณะที่แตกต่างกันไป ฉะนั้นการเลือกกรรมวิธีที่เหมาะสมกับการใช้งานในทางปฏิบัติ มีองค์ประกอบที่จะต้องนำมาพิจารณาหลายประการ แต่โดยทั่วไปวิธีการมอดูเลตที่จัดว่าดีมีประสิทธิภาพควรจะมีคุณสมบัติดังต่อไปนี้

1. สามารถให้อัตราความผิดพลาดบิตที่ต่ำ (bit error rate) แม้ว่าอัตราส่วนกำลังของสัญญาณต่อสัญญาณรบกวน (signal to noise ratio) ของสัญญาณที่รับได้มีค่าไม่สูงนัก

2. มีความต้องการแบนด์วิดท์ขนาดเล็ก และสามารถใช้ส่งข้อมูลด้วยอัตราบิตที่สูง นั่นคือสามารถใช้งานสเปกตรัมความถี่ที่มีอยู่จำกัดได้อย่างมีประสิทธิภาพ

3. สามารถทำงานได้ดีกับช่องสัญญาณที่มีปัญหาของมัลติพาทเฟดดิง (multipath fading)

4. วงจรภาครับและภาคส่งมีโครงสร้างเรียบง่าย ไม่ซับซ้อน สามารถสร้างจริงได้ด้วยต้นทุนที่ประหยัด

ในสภาพความเป็นจริงวิธีการมอดูเลตที่มีอยู่ในทางปฏิบัตินิได้มีคุณสมบัติที่มีครบพร้อมกันทุกข้อ การเลือกใช้วิธีการที่เหมาะสมจึงขึ้นอยู่กับความต้องการของระบบและทรัพยากรที่มีอยู่เป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การสื่อสารผ่านพอร์ต RS-232C

พอร์ต RS-232C นี้ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรมเรียกว่า Universal Asynchronous Adaptor สาเหตุที่เรียกชื่อว่า RS-232 C เนื่องมาจากสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์แห่งอเมริกา หรือ EIA ได้กำหนดมาตรฐานของอุปกรณ์การสื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อ RS-232C ความจริงมาตรฐานการส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่ที่นิยมกันมากที่สุดสำหรับไมโครคอมพิวเตอร์คือ RS-232C

มาตรฐาน RS-232C

ได้จัดพิมพ์ขึ้นเมื่อ ปี ค.ศ. 1969 โดยสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์แห่งสหรัฐอเมริกา RS ย่อมาจาก Recommend Standard ส่วน 232 เป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ C เป็นหมายเลขของฉบับสุดท้ายของมาตรฐานตัวนี้ จุดประสงค์ของมาตรฐานตัวนี้ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment, DTE) กับอุปกรณ์สื่อสารข้อมูล (Data Communication Equipment : DEC) สำหรับผู้ใช้ไมโครคอมพิวเตอร์ DTE ก็หมายถึงตัวไมโครคอมพิวเตอร์ และ DEC ก็หมายถึงตัวโมเด็ม อุปกรณ์อื่นๆเช่น เครื่องพิมพ์ที่รับสัญญาณแบบอนุกรม อาจเป็นได้ทั้ง DTE และ DEC ขึ้นอยู่กับผู้ผลิต ข้อแตกต่างของ DTE และ DEC จะเห็นได้จากรูปที่ 2.16 จากรูปนี้เราจะเห็นได้ว่า RS-232C มีส่วนสำคัญอย่างใหญ่หลวงสำหรับการสื่อสารข้อมูลระหว่างไมโครคอมพิวเตอร์

รูปที่ 2.16 การใช้ RS-232C เชื่อมต่ออุปกรณ์

ลักษณะของสัญญาณ RS-232C

เพื่อเป็นหลักประกันว่าข้อมูลถูกส่งออกไปอย่างถูกต้อง และอุปกรณ์ถูกควบคุมอย่างถูกต้อง จำเป็นต้องมีข้อตกลงกันในเรื่องของสัญญาณที่ใช้มาตรฐาน RS-232C กำหนดย่านของแรงดันไฟฟ้าของสัญญาณเพื่อสนองจุดประสงค์ข้างต้น ดังแสดงในตารางที่ 2.3

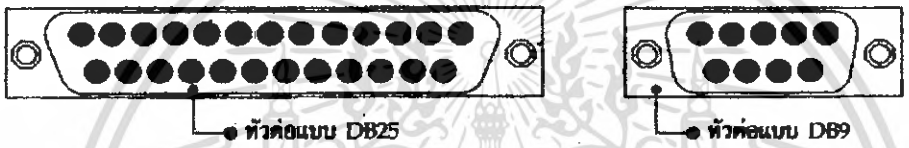
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 การกำหนดย่านของแรงดันฟ้าในสัญญาณตามมาตรฐาน

มาตรฐานของการใช้แรงดันไฟฟ้า			
แรงดันไฟฟ้า	สถานภาพของลอจิก	สถานภาพของลอจิก	ฟังก์ชันในการควบคุม
+3 V ถึง +15 V	0	สเปซ	ON
-3 V ถึง -15 V	1	มาร์ค	OFF

ลักษณะของ Connector แบบ D-type

ในทางฟิสิกส์แล้ว มาตรฐาน RS-232C กำหนดข้อต่อแบบ DB-9 แต่ละขาของข้อต่อกำหนดดังในรูปที่ 2.17



DB-25	DB-9	สัญลักษณ์	ชื่อสัญญาณ	ลักษณะสัญญาณ
Pin 2	Pin 3	TD	Transmit Data	Output
Pin 3	Pin 2	RD	Receive Data	Input
Pin 4	Pin 7	RTS	Request To Send	Output
Pin 5	Pin 8	CTS	Clear To Send	Input
Pin 6	Pin 6	DSR	Data Set Ready	Input
Pin 7	Pin 5	SG	Signal Ground	Ground
Pin 8	Pin 1	CD	Carrier Detect	Input
Pin 20	Pin 4	DTR	Data Terminal Ready	Output
Pin 22	Pin 9	RI	Ring Indicator	Input

รูปที่ 2.17 การกำหนดข้อต่อของ RS-232C

สัญญาณต่างๆของพอร์ตอนุกรมมีหน้าที่ดังนี้

Transmit Data

เป็นสัญญาณที่ส่งออกจาก DTE (หรือตัวไมโครคอมพิวเตอร์) ไปยังโมเด็มหรือต่อเข้าโดยตรงกับไมโครคอมพิวเตอร์ตัวอื่นหรือเครื่องพิมพ์ เมื่อไม่มีสัญญาณส่งออก สถานภาพของลอจิกที่ขานี้จะมีค่าเท่ากับ "1" หรือเทียบเท่ากับสต๊อปบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Receive Data

เป็นทางของสัญญาณเข้าไปยัง DTE หรือไมโครคอมพิวเตอร์เมื่อไม่มีสัญญาณรับเข้ามาขานี้จะมีสถานภาพทางลอจิกเป็น "1"

Request To sent

ใช้สำหรับส่งสัญญาณไปยังโมเด็มหรือเครื่องพิมพ์เป็นการเรียกร้องที่จะส่งสัญญาณมาทาง ขา 2 สัญญาณนี้ใช้คู่กับ CTS หรือ Clear To Send อุปกรณ์รับหากได้รับสัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณได้หรือยัง หากพร้อมที่จะได้รับสัญญาณออกไปที่สาย CTS

Clear To Send

ดังอธิบายใน RTS เมื่อสัญญาณนี้อยู่ในสภาวะ OFF (Negative Voltage หรือลอจิก "1") หมายความว่า อุปกรณ์รับกำลังบอกว่าพร้อมที่จะรับข้อมูลแล้ว

Data Set Ready

เมื่อสัญญาณสายนี้อยู่ในสถานะ ON (Positive Voltage หรือลอจิก "0") เป็นการบอกไมโครคอมพิวเตอร์หรือฝ่ายส่งว่า โมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้วพร้อมที่จะส่งได้แล้ว โมเด็มที่มีการหมุนหมายเลขอัตโนมัติส่งสัญญาณสายนี้ไปบอกให้คอมพิวเตอร์รู้ว่าได้ต่อโทรศัพท์เสร็จแล้ว

Signal Ground

SG ทำหน้าที่เป็นระดับแรงดันอ้างอิงสำหรับทุกๆ สายของสัญญาณ จะมีแรงดันเป็น "0" เมื่อเทียบกับสัญญาณตัวอื่น

Carrier Detect

โมเด็มจะส่งสัญญาณที่อยู่ในสถานะ ON (ลอจิก "0") ไปบอกไมโครคอมพิวเตอร์ เมื่อได้รับสัญญาณจากโมเด็มของอีกฝ่ายหนึ่ง สัญญาณนี้จะนำไปจุด LED บอกว่าได้รับสัญญาณจากโมเด็มอีกฝ่ายหนึ่งแล้ว ไฟ LED จะอยู่บนหน้าปัดของโมเด็มเอง

Data Terminal Ready

คอมพิวเตอร์เปิดสัญญาณสายนี้ให้ ON (ลอจิก "0") เมื่อพร้อมที่จะติดต่อกับโมเด็ม โมเด็มส่วนมากจะไม่รายงานสถานภาพของตัวเอง (CD, DSR และ CTS)ให้คอมพิวเตอร์รู้ หากคอมพิวเตอร์ไม่เปิดสัญญาณ DTR

Ring Indicator

สัญญาณนี้ใช้ในโมเด็มที่เป็นระบบตอบได้อัตโนมัติ (Auto-answer) สัญญาณนี้จะ ON เมื่อมีกระดิ่งมาและ OFF ระหว่างเสียงของกระดิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องค์ประกอบในการรับ-ส่งข้อมูลแบบอนุกรม

การสื่อสารแบบอนุกรมสำหรับเครื่องคอมพิวเตอร์นั้น เป็นการสื่อสารข้อมูลแบบอะซิงโครนัส ซึ่งประกอบไปด้วย 4 ส่วนด้วยกันคือ

Start Bit

มีขนาด 1 บิต จะใส่ที่จุดเริ่มต้นเสมอ เพื่อเตือนอุปกรณ์ฝ่ายรับว่าข้อมูลกำลังจะมาถึง

บิตข้อมูล (Data Bit)

มีขนาด 7 หรือ 8 บิต ซึ่งบิตข้อมูลจะส่งเป็นกลุ่มๆ โดยทั่งไปจะส่งเป็น 7 หรือ 8 บิต ซึ่งเพียงพอต่อการส่ง Ascii Word

Parity Bit

มีขนาด 1 บิต ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ส่ง เราจะใส่บิตพาริตี้เข้าไป แต่ทั้งตัวรับและตัวส่งจะต้องรู้กันว่าใช้พาริตี้แบบไหนในการส่งข้อมูล ซึ่งหลักในการกำหนดบิตพาริตี้มีหลายแบบดังนี้

1. พาริตี้คู่ (Even Parity)

ค่าของบิตพาริตี้นี้เมื่อรวมกับทุกๆบิตของข้อมูลแล้ว จะต้องมีจำนวนบิตที่เป็นเลข 1 เป็นเลขคู่ ตัวอย่างเช่น ข้อมูล 1000101 มีเลข 1 ทั้งหมด 3 ตัว ดังนั้น บิตพาริตี้จะเป็น 0

2. พาริตี้คี่ (Odd Parity)

ค่าของบิตพาริตี้นี้เมื่อรวมกับทุกๆบิตของข้อมูลแล้ว จะต้องมีจำนวนบิตที่เป็นเลข 1 เป็นเลขคี่ ตัวอย่างเช่น ข้อมูล 1000101 มีเลข 1 ทั้งหมด 3 ตัว ดังนั้น บิตพาริตี้จะเป็น 1

3. ไม่มีพาริตี้ (None)

ถ้าตั้งบิตพาริตี้เป็น None ทั้งภาครับและภาคส่งจะไม่มีการตรวจสอบบิตพาริตี้

Stop Bit

มีขนาด 1 หรือ 2 บิต เป็นบิตที่ส่งมาปิดท้ายข้อมูล

อัตราความเร็วในการรับส่งข้อมูลอนุกรม

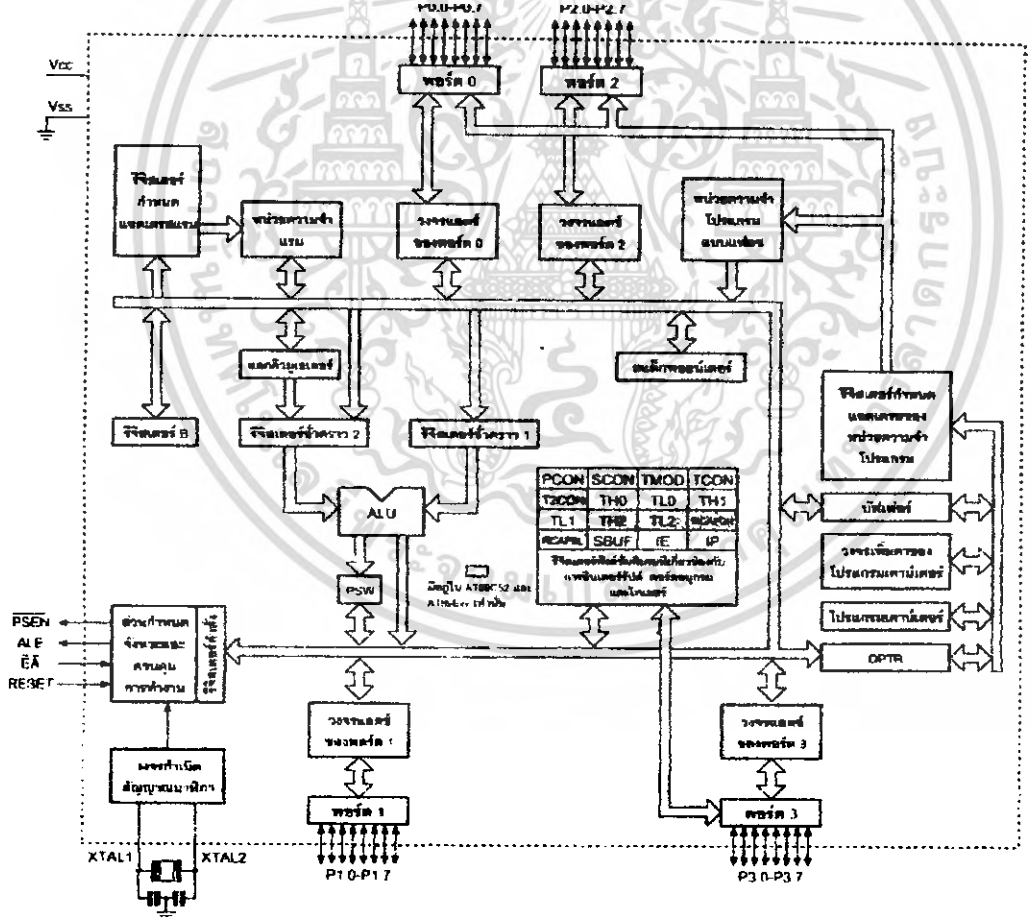
การที่อุปกรณ์ 2 อย่างจะติดต่อสื่อสารกันได้นั้น จะต้องทำงานด้วยอัตราเร็วเท่ากัน ซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสคือ ค่าบอดเรต มีหน่วยเป็นบิตต่อวินาที ซึ่งค่าอัตราเร็วในการสื่อสารแบบอนุกรมสำหรับมาตรฐาน RS-232C นั้นมีใช้ดังนี้ 110, 150, 300, 600, 120, 2400, 4800, 9600 และ 19200 บิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ไมโครคอนโทลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ คือ อุปกรณ์ประเภทสารกึ่งตัวนำที่รวบรวมฟังก์ชั่นการทำงานต่างๆ ไว้ภายในตัวของมันเอง โดยมีโครงสร้างใกล้เคียงกับคอมพิวเตอร์ คือ ภายในประกอบด้วยหน่วยรับข้อมูลและโปรแกรม หน่วยประมวลผล หน่วยความจำ หน่วยแสดงผล ซึ่งส่วนประกอบเหล่านี้มีความสมบูรณ์ในตัวของมันเอง ทำให้มีขนาดเล็ก และสามารถเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ต่างๆ ที่เชื่อมต่อกับตัวมัน ง่ายต่อการนำไปประยุกต์ใช้งาน

ความแตกต่างของ Microcontroller กับ Microcomputer คือ Microcontroller นั้นมีสมบูรณ์ภายในตัวของมันเอง คือ มีส่วนประกอบต่างๆ ครบถ้วน ส่วน Microcomputer นั้นต้องทำงานร่วมกับอุปกรณ์ข้างเคียงที่เชื่อมต่อจากภายนอก เช่น แป้นพิมพ์ เครื่องอ่านเขียนแผ่นบันทึก หน่วยความจำ I/O ฯลฯ



รูปที่ 2.18 โครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ MCS-51

1. ต้องการแหล่งจ่ายไฟ 5 Volt เพียงชุดเดียว

2. มีหน่วยความจำสำหรับการเก็บโปรแกรมควบคุมการทำงานอยู่ภายในชิปจำนวน 4 กิโลไบต์

3. มีหน่วยความจำสำหรับเก็บข้อมูลทั่วไป(RAM) อยู่ภายในจำนวน 128 ไบต์

4. สามารถใช้หน่วยความจำสำหรับโปรแกรมข้อมูลที่อยู่ภายนอกชิปได้อย่างละ 64 กิโลไบต์ แยกจากกัน

5. คำสั่งส่วนใหญ่ใช้เวลาการทำงาน 1 ไมโครวินาที เมื่อใช้ คริสตัลความถี่ 12 เมกะเฮริตซ์

6. มีวงจรรับส่งข้อมูลอนุกรมได้ 2 ทิศทาง

7. รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดอัตราเร็วในการรับและส่งข้อมูล (Baud rate) ได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที

8. จัดลำดับความสำคัญของสัญญาณ Interrupt ได้ 2 ระดับ

9. มี Register สำหรับใช้งานเป็น Timer หรือ Counter เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิปหรือนับการเปลี่ยนสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัวเพื่อใช้สำหรับนับจำนวน Pulse หรือใช้วัดช่วงเวลา

10. หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และระดับบิต เพื่อให้การออกแบบโปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น

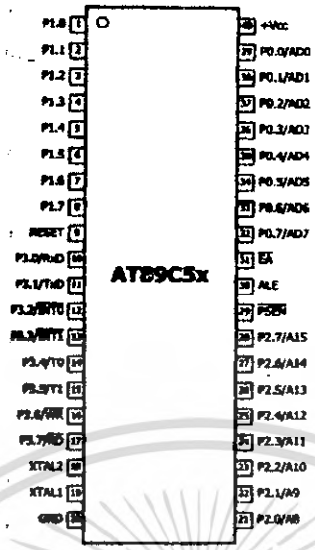
11. มีคำสั่ง คูณและหาร เลขขนาด 8 บิตในตัวเอง

12. สามารถประมวลผลแบบบูลีนเพื่อใช้ในงานควบคุมโดยเฉพาะ

โครงสร้างภายนอกของ MCS-51

ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51

หน้าที่การใช้งานแต่ละขาของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 มีดังนี้

ขา Vss (ขา20)สำหรับต่อลงกราวด์

ขา Vcc (ขา40)สำหรับต่อแหล่งจ่ายแรงดันไฟตรงขนาด 5 Volt

ขาพอร์ต 0 (ขา 32-39) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 0 ขนาด 8 บิต (P0.0-P0.7) แบบ Open Drain Bidirectional พอร์ตนี้สามารถใช้งานเป็น Input Output พอร์ต ทั่วไปได้ โดยหากต้องการใช้งานเป็น Input ทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาอยู่ในสถานะถูกปล่อยลอย (มีสถานะ High impedance) นอกจากใช้งานเป็น Input Output พอร์ตแล้ว พอร์ต 0 ยังใช้ในการติดต่อกับขาแอดเดรสไบต์ต่ำ (A0-A7) และการรับส่งข้อมูล (D0-D7) โดยการมัลติเพล็กซ์ เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรส และขาข้อมูล

ขาพอร์ต1 (ขา1-8) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 1 (P1.0-P1.7) สามารถใช้งานเป็นอินพุตหรือเอาท์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็น Input ทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาอยู่ในสถานะถูกปล่อยลอย (มีสถานะ High impedance) นอกจากนี้ยังใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูล

ขาพอร์ต2 (ขา21-28) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 2 (P2.0-P2.7) ขนาด 8 บิต แบบ Open Drain Bidirectional สามารถใช้งานเป็นอินพุตหรือเอาท์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็น Input ทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาอยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะถูกปล่อยลอย (มีสถานะ High impedance) นอกจากนี้พอร์ต 2 ยังใช้ในการติดต่อหน่วยความจำสำหรับจัดเก็บโปรแกรมและข้อมูลภายนอกด้วย โดยใช้สำหรับส่งค่าแอดเดรสบัสไบต์สูง (A8-A15)

ขาพอร์ต3 (ขา10-17) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 3 (P3.0-P3.7) สามารถใช้งานเป็นอินพุตหรือเอาท์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็น Input ทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาอยู่ในสถานะถูกปล่อยลอย (มีสถานะ High impedance)

นอกจากนี้พอร์ต3ยังใช้งานในหน้าที่พิเศษ ดังมีรายละเอียดดังต่อไปนี้ดังนี้

ขา P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

ขา P3.1 ใช้เป็นขาเอาท์พุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

ขา P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา $\overline{INT0}$

ขา P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา $\overline{INT1}$

ขา P3.4 ใช้เป็นขาอินพุตรับสัญญาณไทม์เมอร์จากภายนอกช่อง 0 หรือขา T0

ขา P3.5 ใช้เป็นขาอินพุตรับสัญญาณไทม์เมอร์จากภายนอกช่อง 1 หรือขา T1

ขา P3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับการจัดเก็บข้อมูลภายนอก หรือขา $\overline{WR}$

ขา P3.7 ใช้เป็นสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับการจัดเก็บข้อมูลภายนอก หรือขา $\overline{RD}$

ขา RST (ขา 9) ใช้สำหรับการรีเซตการทำงานทุกอย่างภายในชิปเพื่อเริ่มต้นการทำงานใหม่ การรีเซตใช้เมื่อเริ่มจ่ายพลังงานหรือเมื่อโปรแกรมเกิดการทำงานผิดพลาด เมื่อต้องการรีเซต MCS-51 ขานี้ต้องมีสถานะ "1" อย่างน้อย 2 Machine cycle โดยที่สัญญาณนาฬิกายังคงทำงานอยู่อย่างต่อเนื่อง

ขา ALE/PROG (ขา30) เป็นขาสำหรับใช้ส่งสัญญาณออกไปภายนอก เพื่อควบคุมการแลตช์ค่าแอดเตรสไบต์ต่ำ (Address Latch Enable) จากพอร์ต 0 ในระหว่างการติดต่อหน่วยความจำสำหรับเก็บข้อมูลภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์

ขา PSEN (ขา 29) ใช้ส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำนอก (Program Strobe Enable) เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกชิปตัวไมโครคอนโทรลเลอร์นี้จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีน-ไซเคิล แต่ในช่วงการเขียนหรืออ่านข้อมูลกับหน่วยความจำภายนอก หรือเมื่อใช้โปรมแกรมจากหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปจะไม่มีสัญญาณออกมาจากขานี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา EA/Vpp (ขา 31) เป็นขาสำหรับใช้เลือกให้ MCS-51 ทำงานจากโปรแกรมที่อยู่ภายในหรือภายนอก โดยหากขานี้มีสถานะเป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมจากภายนอก หากขานี้มีสถานะเป็น 1 หมายถึงบังคับไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายใน และสำหรับ MCS-51 ที่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในสามารถเลือกให้ทำงานได้ทั้งจากโปรกรมที่เก็บไว้ในหน่วยความจำภายนอกชิป ด้วยการต่อขา EA กับไฟเลี้ยงหรือ กราวด์ตามลำดับ ส่วนใน MCS-51 ที่ไม่มีหน่วยความจำสำหรับจัดเก็บโปรกรมภายใน ให้ต่อขานี้ลงกราวด์เสมอ

ขา XTAL 1(ขา 19) และ **XTAL 2**(ขา 18) เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

โครงสร้างภายในของ Microcontroller MCS-51 Series ในไมโครคอลโทรลเลอร์ตระกูล MCS-51 แบ่งชนิดหรือหน้าที่ของหน่วยความจำออกเป็น 2 ส่วนคือ

- หน่วยความจำโปรแกรม (Program memory)
- หน่วยความจำข้อมูล (Data memory)

หน่วยความจำสำหรับเก็บโปรแกรม หน่วยความจำสำหรับเก็บโปรแกรมใน MCS-51 จะแบ่งออกเป็น 2 ส่วนคือ หน่วยความจำสำหรับเก็บโปรแกรมภายใน (Internal program memory) และหน่วยความจำสำหรับเก็บโปรแกรมภายนอก (External program memory) ขนาดของหน่วยความจำสำหรับเก็บโปรแกรมภายในมีได้ตั้งแต่ 0,4,8,16 กิโลไบต์ ขึ้นอยู่กับเบอร์ของไมโครคอนโทรลเลอร์

หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บข้อมูลของ MCS-51 จะแบ่งออกเป็น 2 ส่วนคือ หน่วยความจำสำหรับเก็บข้อมูลภายใน 128 ไบต์จะเป็นหน่วยความจำที่ใช้งานทั่วไป (Internal Ram) อยู่ที่ตำแหน่งแอดเดรส 00H-7FH และหน่วยความจำในตำแหน่งแอดเดรส 80H-FFH ซึ่งจะเป็นส่วนของรีจิสเตอร์ที่ใช้งานเฉพาะ (Special Function Register)

โดยหน่วยความจำข้อมูลภายในที่ตำแหน่งแอดเดรส 00H-7FH ยังสามารถแบ่งออกเป็นส่วนย่อยได้ดังนี้

1. **พื้นที่ในหน่วยความจำข้อมูลตำแหน่งที่ 00H-1FH จำนวน 32 ไบต์ จะถูกแบ่งเป็น 4 กลุ่ม เรียกว่า แบงค์ (Bank) และในแต่ละแบงค์จะมี 8 ไบต์ พื้นที่ในแต่ละแบงค์จะถูกใช้งานเป็นรีจิสเตอร์ที่ใช้งานทั่วไป (R0-R7) โดยที่รีจิสเตอร์ R0 จะอยู่ในตำแหน่งแรกของแต่ละแบงค์ และ รีจิสเตอร์ R7 จะอยู่ในตำแหน่งสุดท้ายของแต่ละแบงค์ ในการนำไปใช้งานจะเลือกใช้รีจิสเตอร์ได้เพียงแบงค์เดียว และเลือกใช้พื้นที่ของรีจิสเตอร์ R0-R7 ในแบงค์ใดๆก็ได้ โดยการกำหนดค่าของข้อมูลที่รีจิสเตอร์ PSW ในส่วนของรีจิสเตอร์เฉพาะ หากไม่มีการกำหนดใดๆเลย เมื่อทำการรีเซ็ตให้กับไมโครคอนโทรลเลอร์จะถูกกำหนดให้เริ่มต้นใช้งานที่รีจิสเตอร์ R0-R7 ในหน่วยความจำตำแหน่งแบงค์ 0 ให้เอง**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. พื้นที่ในหน่วยความจำข้อมูลตำแหน่งที่ 20H-0FH จำนวน 16 ไบต์ เป็นส่วนที่ใช้งานในลักษณะการเข้าข้อมูลแบบไบต์หรือแบบบิตได้ และสามารถอ้างตำแหน่งแบบบิตได้โดยตรง เพียงแต่ระบุตำแหน่งหรือชื่อของบิตนั้นๆได้ ซึ่งจะมีอยู่จำนวน 128 บิต แต่ละบิตจะมีหมายเลขตำแหน่งของบิต คือ 00H-7FH โดยตำแหน่งบิตที่ 00H ก็คือข้อมูลบิตที่ต่ำที่สุดในตำแหน่งที่แอดเดรสที่ 20H หรืออาจเรียกว่า (20H.1) และตำแหน่งของบิตที่ 7FH คือข้อมูลบิตที่สูงสุดในตำแหน่งแอดเดรสที่ 2FH หรืออาจเรียกว่า (20H.7) การอ้างตำแหน่งบิตจะทำให้โปรแกรมทำงานได้เร็วขึ้น

รีจิสเตอร์ใช้งานเฉพาะ เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ โดยมีแอดเดรสอยู่ระหว่าง 80H-FFH ในพื้นที่ของหน่วยความจำข้อมูลส่วนบน สามารถเข้าถึงได้โดยตรง

1. รีจิสเตอร์แสดงสถานะของโปรแกรม (Program Status Word : PSW)

เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต จึงสามารถกำหนดค่าในแต่ละบิตของรีจิสเตอร์ตัวนี้ได้อย่างอิสระ มีแอดเดรสอยู่ที่ D0H ทำหน้าที่เก็บสถานะของการทำงานของโปรแกรมในขณะนั้นจะเรียกสถานะต่างๆของโปรแกรมว่า แฟลก(flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วเกิดการเปลี่ยนแปลงสถานะขึ้น ผลของการเปลี่ยนแปลงนั้นจะมาปรากฏที่บิตต่างๆของรีจิสเตอร์ PSW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
CY	AC	F0	RS1	RS0	OV	-	P

CY . แฟลกทด (Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิก แล้วค่าของแอกคิวมูลเลเตอร์เกิน 255 (ฐานสิบ) หรือ FFH

AC . แฟลกทดเสริม (Auxiliary Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์แล้วทำให้เกิดการทดข้ามจากบิต 3 มายังบิต 4

F0 แฟลกใช้งานทั่วไป เมื่อผู้เขียนโปรแกรมกำหนดค่าที่บิตนี้แล้ว ไม่ว่าจะกระทำคำสั่งใด ๆ ที่บิตนี้จะไม่มีการเปลี่ยนแปลง

RS1 . บิตเลือกรีจิสเตอร์แบงก์ (Register Select1) ใช้งานร่วมกับบิต RS0 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

RS0 . บิตเลือกรีจิสเตอร์แบงก์ (Register Select0) ใช้งานร่วมกับบิต RS1 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

OV . บิตเกิน (Overflow) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว ทำให้เกิดการทดข้ามจากบิต 6 มายังบิต 7 ของแอกคิวมูเลเตอร์ หรือแอกคิวมูเลเตอร์มีค่าเกิน 127 (ฐานสิบ) นอกจากนั้นยังใช้เป็นการแสดงค่าลบด้วย

- . บิตนี้ผู้ใช้งานสามารถกำหนดใช้งานได้อย่างอิสระ

P บิตพาริตี้ (Parity) ใช้ในการตรวจสอบจำนวนค่า "1" ภายในแอกคิวมูเลเตอร์ ถ้าหากในแอกคิวมูเลเตอร์มีจำนวนบิตที่เป็น "1" รวมกันเป็นเลขคู่ บิตนี้จะเป็น "0" ถ้ารวมกันเป็นเลขคี่ บิตนี้จะเป็น "1"

รูปที่ 2.20 รายละเอียดของรีจิสเตอร์แสดงสถานะของโปรแกรมหรือ PSW

2. แอกคิวมูเลเตอร์ (Accumulator : ACC)

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะอย่างยิ่งในการคำนวณทางคณิตศาสตร์และลอจิก ก่อนที่จะส่งข้อมูลหรือผลลัพธ์ที่ได้ให้แก่ซีพียูเพื่อทำการประมวลผลต่อไป อาจเรียกสั้นๆว่า รีจิสเตอร์ A หรือ ACC รีจิสเตอร์นี้สามารถเข้าถึงระดับบิตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. รีจิสเตอร์ B

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง E0H มีหน้าที่พิเศษคือ หากต้องการคูณหรือหารทางคณิตศาสตร์ต้องนำข้อมูลที่ต้องการหารหรือคูณมาเก็บไว้ในรีจิสเตอร์ B แล้วจึงกระทำคำสั่งการคูณหรือหารกับค่ารีจิสเตอร์ A ต่อไป ในกรณีที่ไม่ได้มีความต้องการคูณหรือหารข้อมูล สามารถใช้รีจิสเตอร์ B นี้ในการเก็บข้อมูลทั่วไปได้เหมือนกับรีจิสเตอร์ปกติ และสามารถเข้าถึงในระดับบิตได้เช่นเดียวกับรีจิสเตอร์ A

4. โปรแกรมเคาน์เตอร์ (Program Counter : PC)

มีขนาด 16 บิต มีหน้าที่แจ้งแอดเดรสของหน่วยความจำโปรแกรมในตำแหน่งถัดไปที่ซีพียูจะต้องไปทำงาน รีจิสเตอร์ PC เป็นรีจิสเตอร์ตัวเดียวที่ไม่ได้จัดสรรไว้รวมกับรีจิสเตอร์ SFR ตัวอื่นๆ การเปลี่ยนแปลงค่าของรีจิสเตอร์ PC จะขึ้นอยู่กับผลของการกระทำคำสั่งภายในหน่วยความจำโปรแกรมกำหนด

5. สแต็กพอยน์เตอร์ (Stack Pointer : SP)

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 81H ใช้ในการเก็บค่าตำแหน่งของตัวชี้สแต็ก ซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียูมีการกระโดดไปทำงานที่โปรแกรมย่อย หรือกระโดดโปรแกรมย่อยกลับมายังโปรแกรมหลัก เมื่อมีการรีเซตเกิดขึ้น

6. รีจิสเตอร์ชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (Data Pointer : DPTR)

มีขนาด 16 บิต โดยแบ่งเป็นรีจิสเตอร์ชี้ข้อมูลไบต์สูง (DPH) และรีจิสเตอร์ชี้ข้อมูลไบต์ต่ำ (DPL) แต่ละตัวมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 82H สำหรับ DPL และ 83H สำหรับ DPH รีจิสเตอร์ DPTR นี้ใช้ในการเก็บค่าแอดเดรสของหน่วยความจำหรืออุปกรณ์ภายนอกที่ไมโครคอนโทรลเลอร์ต้องการติดต่อด้วย

7. รีจิสเตอร์พอร์ต (Port Register)

มีขนาด 8 บิต ที่ใช้เก็บข้อมูลของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ มี 4 ตัว คือ รีจิสเตอร์พอร์ต 0 หรือ P0 มีรีจิสเตอร์อยู่ที่ 80H, รีจิสเตอร์พอร์ต 1 หรือ P1 มีรีจิสเตอร์อยู่ที่ 90H, รีจิสเตอร์พอร์ต 2 หรือ P2 มีรีจิสเตอร์อยู่ที่ A0H, รีจิสเตอร์พอร์ต 3 หรือ P3 มีรีจิสเตอร์อยู่ที่ B0H รีจิสเตอร์ทุกตัวสามารถเข้าถึงได้ในระดับบิต เมื่อต้องการอ่านหรือเขียนข้อมูลไปยังพอร์ตของไมโครคอนโทรลเลอร์จะต้องกระทำผ่านรีจิสเตอร์นี้ทุกครั้ง

8. รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer : SBUF)

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 99H ใช้ในการเก็บข้อมูลที่ส่งออกหรือรับเข้าของวงจรสื่อสารอนุกรมที่มีอยู่ในไมโครคอนโทรลเลอร์โดยแบ่งออกเป็น 2 ส่วน คือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (Transmit Buffer Register)และรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (Recieve Buffer Register) เมื่อมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูลเพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อผ่านไปยังบัฟเฟอร์สำหรับรับข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือขา P3.0 ของไมโครคอนโทรลเลอร์

9. รีจิสเตอร์ไทเมอร์ (Timer Register)

มีขนาด 16 บิต ใช้ในการเก็บค่าของตัวนับหรือเคาน์เตอร์ (Counter) ภายในไมโครคอนโทรลเลอร์ เพื่อใช้ในการสร้างฐานเวลา, จับเวลา หรือนับจำนวนพัลส์สัญญาณนาฬิกาภายใน บางทีเรียกรีจิสเตอร์ตัวนี้ว่า รีจิสเตอร์ไทเมอร์/เคาน์เตอร์

รีจิสเตอร์ไทเมอร์/เคาน์เตอร์ แบ่งเป็น T0 หรือ Timer 0 และ T1 หรือ Timer 1 ในรีจิสเตอร์ยังแบ่งเป็นรีจิสเตอร์ไทเมอร์ไบต์ต่ำ (TL) และรีจิสเตอร์ไทเมอร์ไบต์สูง (TH) โดยรีจิสเตอร์ TL0 มีแอดเดรสอยู่ที่ 8AH จิสเตอร์ TH0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ TL1 และ TH1 มีแอดเดรสอยู่ที่ 8CH และ 8DH

10. รีจิสเตอร์ควบคุม (Control Register)

รีจิสเตอร์ PCON เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตราการรับส่งข้อมูลของวงจรสื่อสารอนุกรมและกำหนดการทำงานในโหมดประหยัดพลังงานของไมโครคอนโทรลเลอร์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SMOD	-	-	-	GF1	GF0	PD	IDL

SMOD (Double Baud rate bit) : ใช้ในการเพิ่มอัตราบอดของการรับส่งข้อมูลของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ เป็น 2 เท่า เมื่อบิตนี้เซตเป็น "1" และใช้ไทเมอร์ 1 ในการสร้างอัตราบอด โดยพอร์ตอนุกรมต้องถูกกำหนดให้ทำงานในโหมด 1, 2 หรือ 3

GF1 และ GF2 (General-purpose flag bit) : เป็นแฟลกสำหรับใช้งานทั่วไป เซตและเคลียร์ด้วยซอฟแวร์

PD (Power Down bit) : ใช้เอ็นเอเบิลการทำงานในโหมดประหยัดพลังงานแบบลดพลังงานหรือเพาเวอร์ดาวน์ (power down mode) สามารถเซตด้วยกระบวนการทางซอฟต์แวร์และเคลียร์ด้วยกระบวนการทางฮาร์ดแวร์ ถ้าต้องการให้ทำงานในโหมดประหยัดพลังงานแบบลดพลังงานต้องเซตบิตนี้ให้เป็น "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IDL (Idle mode bit) : ใช้เอ็นเอเบิลการทำงานในโหมดประหยัดพลังงานแบบไอเดิล (idle mode) สามารถเซตด้วยกระบวนการทางซอฟต์แวร์และเคลียร์ด้วยกระบวนการทางฮาร์ดแวร์ ถ้าต้องการให้ทำงานในโหมดประหยัดพลังงานแบบไอเดิลต้องแซตบิตนี้ให้เป็น "1"

รีจิสเตอร์ SCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของวงจรสื่อสารอนุกรมภายในไมโครคอนโทรลเลอร์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0-SM1 (Serial port mode bit 0-1) : ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์

SM2 : ใช้ในการเอ็นเอเบิลการสื่อสารในแบบมัลติโปรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกตีฟ ถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิต RI จะไม่แอกตีฟถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable Serial Reception) : ใช้ในการเอ็นเอเบิลการรับข้อมูลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้มีการรับข้อมูลต้องเซตบิตนี้เป็น "1"

TB8 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB9 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ แต่ถ้าหากพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น "0" ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด (Stop bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ใช้งาน บิต RB8 นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt flag) : ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ สามารถเซตด้วนกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (Receive Interrupt flag) : ใช้แสดงการเกิดอินเตอร์รัปต์เมื่อมีการรับข้อมูลเข้าสู่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ สามารถเซตด้วนกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อสามารถรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

รีจิสเตอร์ TCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1(Timer 1 overflow flag) : เซตด้วยกระบวนการทางฮาร์ดแวร์ เมื่อค่าของรีจิสเตอร์ Timer 1 เกิดการนับเกินหรือเกิดโอเวอร์โฟลว การเคลียร์บิตนี้ทำได้ด้วยกระบวนการทางฮาร์ดแวร์เช่นกัน โดยบิตนี้จะเคลียร์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

TR1 (Timer 1 run control bit) : ใช้ในการเปิดปิดการทำงานของไทเมอร์ 1 ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้ไทเมอร์ 1 ทำงานต้องเซตบิตนี้ให้เป็น "1"

TF0 (Timer 0 overflow flag) : เซตด้วยกระบวนการทางฮาร์ดแวร์ เมื่อค่าของรีจิสเตอร์ Timer 0 เกิดการนับเกินหรือเกิดโอเวอร์โฟลว การเคลียร์บิตนี้ทำได้ด้วยกระบวนการทางฮาร์ดแวร์เช่นกัน โดยบิตนี้จะเคลียร์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

TR0 (Timer 0 run control bit) : ใช้ในการเปิดปิดการทำงานของไทเมอร์ 0 ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้ไทเมอร์ 0 ทำงานต้องเซตบิตนี้ให้เป็น "1"

IE1 (External Interrupt 1 edge flag) : ใช้ในกระบวนการอินเตอร์รัปต์ สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อสามารถตรวจจับขอบขาของสัญญาณอินเตอร์รัปต์จากภายนอกที่ขาอินพุตอินเตอร์รัปต์ 1 (INT1) ได้ และจะทำการเคลียร์เมื่อมีการบริการอินเตอร์รัปต์เกิดขึ้น

IT1 (Interrupt 1 type control bit) : ใช้ในกระบวนการอินเตอร์รัปต์ โดยใช้ในการเลือกลักษณะของสัญญาณอินเตอร์รัปต์จากภายนอกที่ขาอินพุตอินเตอร์รัปต์ 1 (INT1) การเซตและเคลียร์ทำให้ได้ด้วยกระบวนการซอฟต์แวร์

"0" เลือกขอบขาลงของสัญญาณ (falling edge)

"1" เลือกระดับลอจิกต่ำ (low level triggered)

IE0 (External Interrupt 0 edge flag) : ใช้ในกระบวนการอินเตอร์รัปต์ สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อสามารถตรวจจับขอบขาของสัญญาณอินเตอร์รัปต์จากภายนอกที่ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุตอินเตอร์รัปต์ 0 (INT0) ได้ และจะทำการเคลียร์เมื่อมีการบริการอินเตอร์รัปต์เกิดขึ้น

IT0 (Interrupt 0 type control bit) : ใช้ในกระบวนการอินเตอร์รัปต์ โดยใช้ในการเลือกลักษณะของสัญญาณอินเตอร์รัปต์จากภายนอกที่ขาอินพุตอินเตอร์รัปต์ 0 (INT0) การเซตและเคลียร์ทำให้ได้ด้วยกระบวนการซอฟต์แวร์

"0" เลือกขอบขาลงของสัญญาณ (falling edge)

"1" เลือกระดับลอจิกต่ำ (low level triggered)

รีจิสเตอร์ TMOD เป็นรีจิสเตอร์ที่ใช้กำหนดโหมดหรือลักษณะในการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Timer 1				Timer 0			

GATE : ใช้เลือกลักษณะการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์

"0" ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TRx ในรีจิสเตอร์ TCON เป็น "1" เรียกการควบคุมแบบนี้ว่า การควบคุมทางซอฟต์แวร์

"1" ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TRx ในรีจิสเตอร์ TCON เป็น "1" และสถานะลอจิกที่ขาอินพุตอินเตอร์รัปต์ INT0 และ INT1 เป็น "1" เรียกการควบคุมแบบนี้ว่า การควบคุมทางฮาร์ดแวร์

C/$\overline{T}$ (Timer or Counter selector) : ใช้เลือกลักษณะการทำงานของไทเมอร์/เคาน์เตอร์

"0" เลือกให้ทำงานเป็นไทเมอร์ โดยใช้สัญญาณอินพุตจากสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์

"1" เลือกให้ทำงานเป็นเคาน์เตอร์ โดยรับสัญญาณอินพุตทางขา T0 หรือ T1

M1, M0 (Mode selsect bit) : ใช้เลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์

"00" เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ 13 บิต

"01" เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ 16 บิต

"10" เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ 8 บิต แบบตั้งค่าอัตโนมัติ

"11" สำหรับไทเมอร์ 0 เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์แยกส่วน โดยแยกออกเป็นไทเมอร์/เคาน์เตอร์ 8 บิต 2 ตัว รีจิสเตอร์ TL0 จะได้รับการควบคุมการเปิดปิดจากบิต TR0 ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ TCON และรีจิสเตอร์ TH0 ซึ่งเป็นไทเมอร์/เคาน์เตอร์ 8 บิตอีกตัวหนึ่ง จะได้รับการควบคุมจากบิต TR1 ในรีจิสเตอร์ TCON ในกรณีของไทเมอร์ 1 เป็นการสั่งให้ไทเมอร์/เคาน์เตอร์ 1 หยุดทำงาน

รีจิสเตอร์ IE และ IP เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการตอบสนองการอินเตอร์รัปต์ (interrupt : การขัดจังหวะการทำงานปกติของซีพียู) โดย IE เป็นรีจิสเตอร์สำหรับเอ็นเอเบิลหรือใช้ในการกำหนดลักษณะของการตอบสนองการอินเตอร์รัปต์ ในขณะที่ IP เป็นรีจิสเตอร์สำหรับกำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่า จะให้ซีพียูตอบสนองการเกิดอินเตอร์รัปต์ในลักษณะใดก่อนหรือหลัง

รีจิสเตอร์ IE (Interrupt Enable register)

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

EA (Globle enable/disable interrupt) : ใช้เอ็นเอเบิลและดิสเอเบิลการตอบสนองการอินเตอร์รัปต์ทั้งหมด

"0" ดิสเอเบิลการอินเตอร์รัปต์ นั่นคือ กำหนดให้ไมโครคอนโทรลเลอร์ไม่ตอบสนองการอินเตอร์รัปต์

"1" เอ็นเอเบิลการอินเตอร์รัปต์ นั่นคือ กำหนดให้ไมโครคอนโทรลเลอร์สามารถตอบสนองการอินเตอร์รัปต์จาแหล่งกำเนิดต่างๆ

ET2 (Timer 2 interrupt enable) : ใช้ในการเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากโอเวอร์โฟลวหรือการแคปเจอร์ในไทเมอร์/เคาน์เตอร์ 2 จะมีเฉพาะในเบอร์ AT89C52 และในอนุกรม AT89Sxx เท่านั้น บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

ES (Serial port interrupt enable bit) : ใช้ในการเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากการรับหรือส่งข้อมูลทางพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

ET1 (Timer 1 interrupt enable) : ใช้ในการเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากโอเวอร์โฟลวในไทเมอร์/เคาน์เตอร์ 1 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

EX1 (External interrupt 1 enable bit) : ใช้ในการเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากสัญญาณภายนอกที่ป้อนเข้ามายังขา $\overline{INT1}$ บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ET0 (Timer 0 interrupt enable) : ใช้ในการเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากโอเวอร์โฟลวในไทเมอร์/เคาน์เตอร์ 0 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

EX0 (External interrupt 0 enable bit) : ใช้ในการเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากสัญญาณภายนอกที่ป้อนเข้ามายังขา $\overline{INT0}$ บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

รีจิสเตอร์ IP (Interrupt Priority register)

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

PT0 (Timer 2 interrupt priority bit) : ใช้ในการกำหนดความสำคัญของการอินเตอร์รัปต์อันเนื่องมาจากการโอเวอร์โฟลวหรือการแคปเจอร์ในไทเมอร์/เคาน์เตอร์ 2 จะมีเฉพาะในเบอร์ AT89C52 และในอนุกรม AT89Sxx เท่านั้น บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

PS (Serial port interrupt priority bit) : ใช้ในการกำหนดความสำคัญของเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากการรับหรือส่งข้อมูลทางพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

PT1 (Timer 1 interrupt priority) : ใช้ในการกำหนดความสำคัญของเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากโอเวอร์โฟลวในไทเมอร์/เคาน์เตอร์ 1 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

PX1 (External interrupt 1 priority bit) : ใช้ในการกำหนดความสำคัญของเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากสัญญาณภายนอกที่ป้อนเข้ามายังขา $\overline{INT1}$ บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

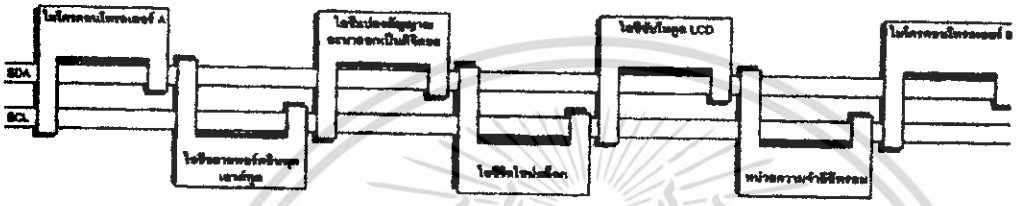
PT0 (Timer 0 interrupt priority) : ใช้ในการกำหนดความสำคัญของเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากโอเวอร์โฟลวในไทเมอร์/เคาน์เตอร์ 0 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

PX0 (External interrupt 0 priority bit) : ใช้ในการกำหนดความสำคัญของเอ็นเอเบิลการอินเตอร์รัปต์อันเนื่องมาจากสัญญาณภายนอกที่ป้อนเข้ามายังขา $\overline{INT0}$ บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การติดต่อระหว่างไอซีโดยบัส I^2C

I^2C ย่อมาจาก Inter-IC Communication พัฒนาโดยฟิลิปส์ด้วยจุดมุ่งหมายคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือสายข้อมูล อีกสายหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

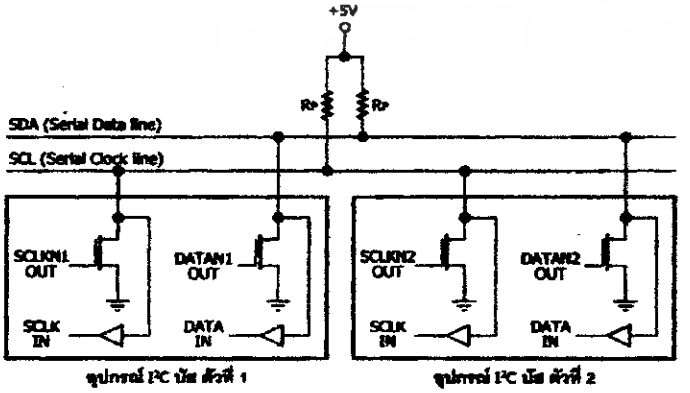


รูปที่ 2.21 ผังแสดงการเชื่อมต่อของอุปกรณ์ต่างๆบนระบบบัส I^2C

สายข้อมูลบนบัส I^2Cมีชื่อเรียกว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock line)

คุณสมบัติทั่วไปของบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง ต้องมีการต่อตัวต้านทานพูลอัปกับแรงดัน +5 V ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่ต่ออยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด(Open-drain) หรือคอลเล็กเตอร์เปิด(Open-collector) ดังรูปที่ 2.22

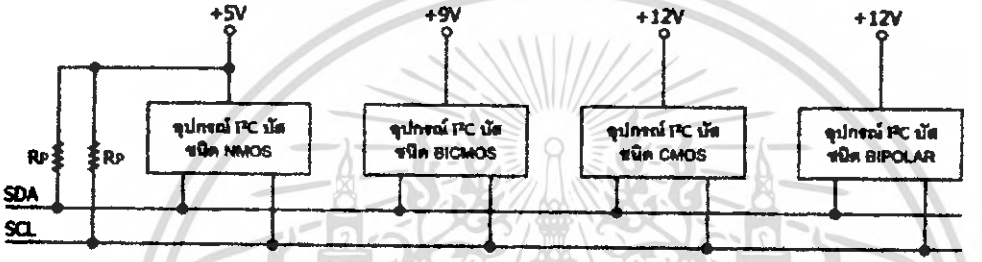


รูปที่ 2.22 แสดงวงจรเอาต์พุตของอุปกรณ์ในระบบบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

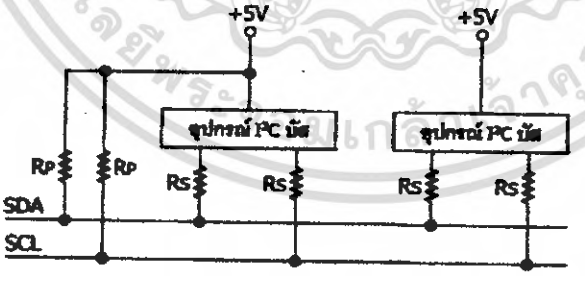
อัตราถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่าคือ 7 บิต และ 10 บิต

ข้อเด่นอีกประการหนึ่งของบัส I^2C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไปเลี้ยงไม่เท่ากันให้สามารถต้อต่อสื่อสารกันได้ โดยต่อสาย SDA และ SCL ของแต่ละอุปกรณ์เข้าด้วยกัน และต่อตัวต้านทานพูลอัปเข้ากับแรงดัน +5V ไว้ด้วยดังรูปที่ 2.23



รูปที่ 2.23 การต่อตัวต้านทาน เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้ามาในบัส I^2C

ในกรณีที่อาจมีแรงดันไฟกระชากปะปนเข้ามาในบัส I^2C ที่ขา SDA และ SCL ของอุปกรณ์แต่ละตัวต้องต่อต้านทานอนุกรมกับขา SDA และ SCL ก่อนต่อเข้าสู่บัส I^2C ดังรูปที่ 2.24



รูปที่ 2.24 การต่อตัวต้านทานเพื่อลดสัญญาณรบกวนที่อาจเข้ามาในบัส I^2C

สภาวะที่เกิดขึ้นบนบัส I^2C

1. บัสว่าง (Bus not busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

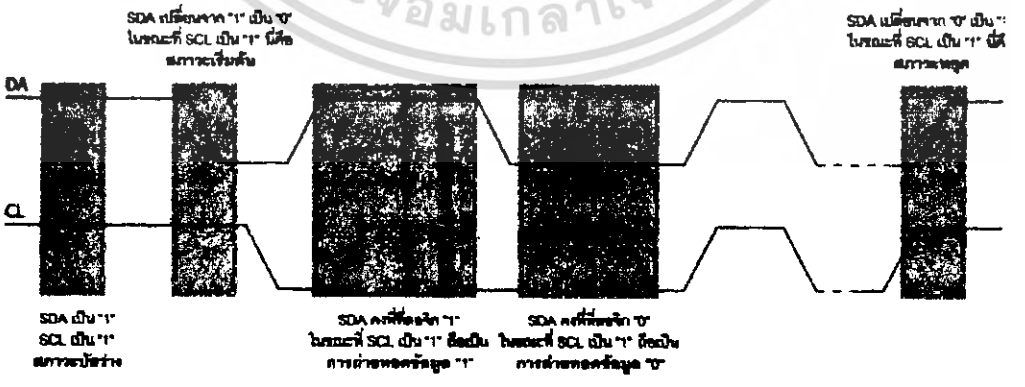
ลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นได้

2. เริ่มต้นการถ่ายข้อมูล (Start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น (START)

3. หยุดการถ่ายข้อมูล (Stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูงในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (STOP)

4. ข้อมูลคำรงอยู่บนบัส I^2C (data valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น "0" หรือ "1" ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้นอุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

5. รับรู้ข้อมูล (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิต เรียกว่า บิตรับรู้ (acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกาเพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว



รูปที่ 2.25 ไดอะแกรมเวลาแสดงสถานะต่างๆในบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

โครงงานนี้สามารถแบ่งระบบการสื่อสารออกเป็น 2 ส่วน คือ

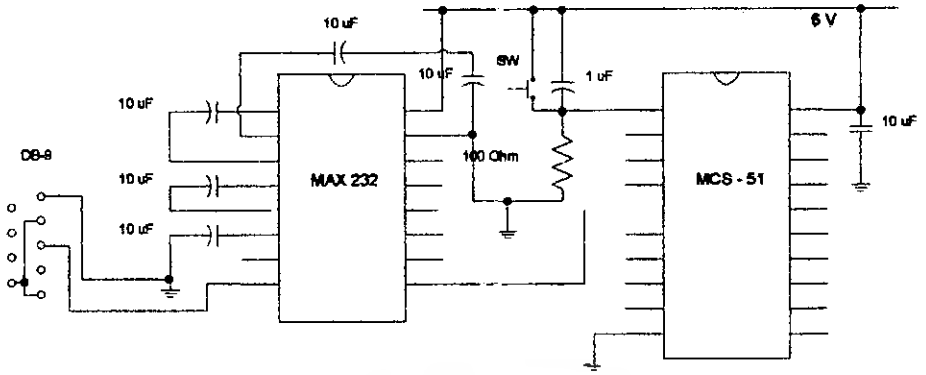
1. การสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์
2. การสื่อสารแบบไร้สายระหว่างไมโครคอนโทรลเลอร์ 2 ตัว เพื่อส่งค่าไปประมวลผลและแสดงผลออกทาง LED 7-Segment

3.1 การสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์

รูปที่ 3.1 บล็อกไดอะแกรมของการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์

คอมพิวเตอร์จะรับค่าที่ต้องการแสดงจากหน้าจอและส่งข้อมูลออกมาทางพอร์ตอนุกรม ส่งมายัง MAX-232 เพื่อเปลี่ยนระดับแรงดัน ต่อจากนั้นจะส่งข้อมูลไปยังที่ขา RxD ของไมโครคอนโทรลเลอร์ซึ่งเป็นขาที่มีไว้สำหรับรับข้อมูลที่เป็นอนุกรม รับไปประมวลผลและแปลงรหัสเพื่อส่งแบบไร้สายไปให้ไมโครคอนโทรลเลอร์อีกตัวหนึ่งเพื่อทำการประมวลผลและแสดงผลออกทาง LED 7-Segment ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

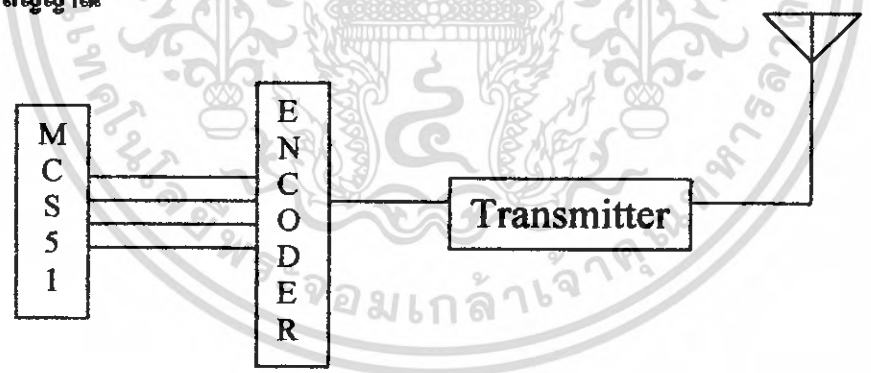


รูปที่ 3.2 วงจรที่ใช้ในการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์

3.2 การสื่อสารระบบไร้สายระหว่างไมโครคอนโทรลเลอร์ 2 ตัว เพื่อส่งค่าไปประมวลผลและแสดงผลออกทาง LED 7-Segment ซึ่งจะประกอบด้วย

1. ภาคส่งสัญญาณ
2. ภาครับสัญญาณ

ภาคส่งสัญญาณ



รูปที่ 3.3 บล็อกไดอะแกรมของภาคส่งสัญญาณ

จากการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ เราจะได้รหัสออกมา 5 ตัวด้วยกัน คือ รหัสชนิดข้อมูลที่แสดงว่าเป็นการส่งค่าราคา เวลา หรือล้างค่าบนบอร์ดแสดงผล รหัสตัวเลขในหลักสิบ หน่วย และสตางค์อีก 2 หลัก ตัวเลขที่ได้รับจากคอมพิวเตอร์นี้ไมโครคอนโทรลเลอร์จะทำการแปลงรหัสเพื่อส่งผ่านไปยังอุปกรณ์เข้ารหัสแปลงเป็นข้อมูลอนุกรมส่งให้ตัวส่งสัญญาณวิทยุส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเป็นอนาลอคผ่านอากาศต่อไป โดยการส่งจะส่งออกไปทีละหลักโดยเริ่มจากรหัสของชนิดข้อมูล ตามด้วยหลักสิบ หลักหน่วยและสตางค์ ตามลำดับ รหัสแต่ละตัวจะถูกกำหนดให้มีรหัสดังนี้

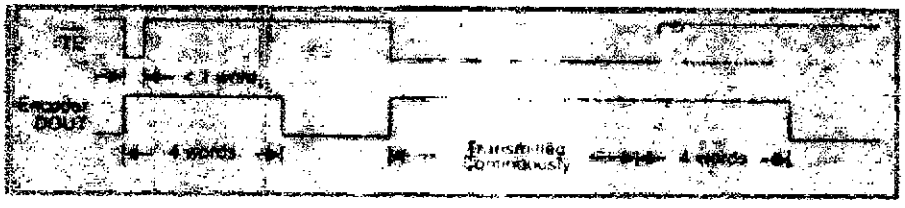
รหัสการส่งค่าราคา	P1 จะมีรหัสเป็น	00010100
รหัสการส่งเวลา	P1 จะมีรหัสเป็น	00011100
รหัส reset	P1 จะมีรหัสเป็น	00011000
เลข 0	P1 จะมีรหัสเป็น	00000000
เลข 1	P1 จะมีรหัสเป็น	00000010
เลข 2	P1 จะมีรหัสเป็น	00000100
เลข 3	P1 จะมีรหัสเป็น	00000110
เลข 4	P1 จะมีรหัสเป็น	00001000
เลข 5	P1 จะมีรหัสเป็น	00001010
เลข 6	P1 จะมีรหัสเป็น	00001100
เลข 7	P1 จะมีรหัสเป็น	00001110
เลข 8	P1 จะมีรหัสเป็น	00010000
เลข 9	P1 จะมีรหัสเป็น	00010010

อุปกรณ์เข้ารหัสสัญญาณ (Encoder : HT12E)

ทำหน้าที่เข้ารหัส 12 bit โดยที่ bit ที่ 1-8 (A0-A7) เป็น bit address ซึ่งจะต้องกำหนดให้ตรงกับตัวรับ โดย bit ที่ 9-12 (AD8-AD11) เป็นทั้ง bit data และ bit address ซึ่งจะเป็นตัวรับคำสั่งมาจากไมโครคอนโทรลเลอร์

เมื่อไมโครคอนโทรลเลอร์ส่งข้อมูล ก็จะใช้บิตสุดท้ายในการกำหนดสถานะที่ ขา $\overline{TE}$ (Transmission Enable) ซึ่งจะเป็น Low ต่อจากนั้นก็จะรับข้อมูลที่ขา AD8-AD11 ตามที่ไมโครคอนโทรลเลอร์ได้ส่งมาให้ ต่อจากนั้นก็จะเข้ารหัสแปลงเป็นสัญญาณแบบอนุกรมแล้วทำการส่งไป

ตัวส่งสัญญาณ ซึ่งมี Timing Diagram ในการส่งสัญญาณดังนี้

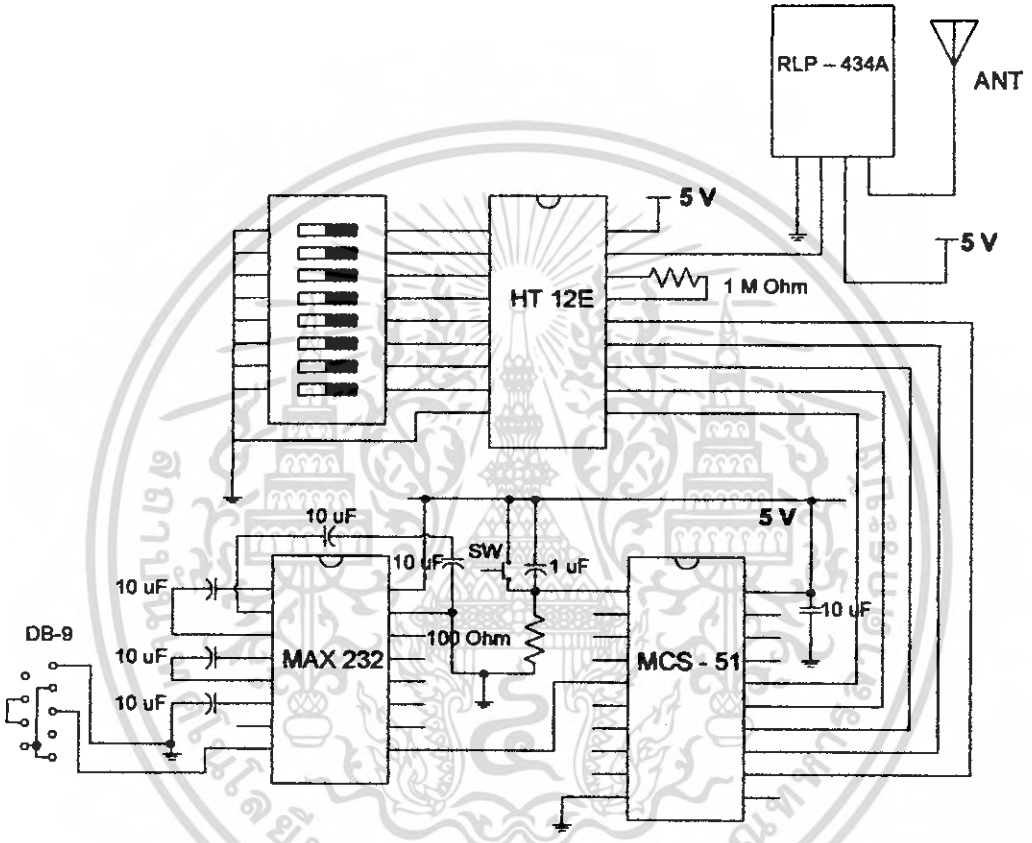


รูปที่ 3.4 Timing Diagram ของ Encoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวส่งสัญญาณ (Transmitter: TPL-434A)

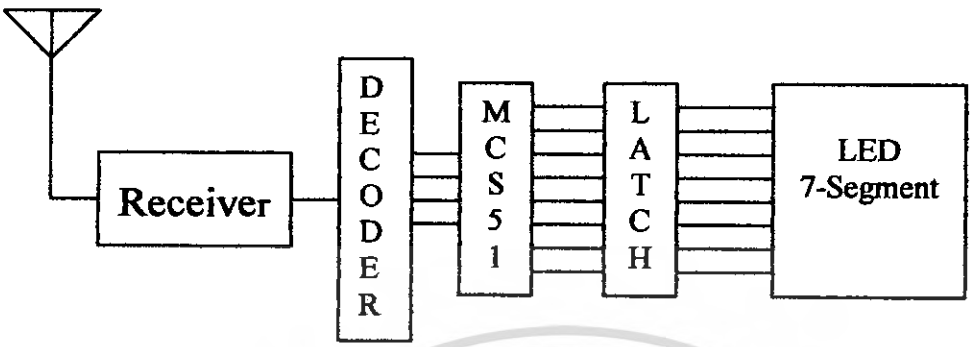
เป็นตัว modulate สัญญาณและส่งสัญญาณที่ได้รับจากอุปกรณ์เข้ารหัสออกไปด้วยคลื่นวิทยุความถี่ 315 MHz โดยเป็นการมอดดูเลตสัญญาณแบบ ASK (Amplitude Shift Keying) ไปยังตัวรับสัญญาณ



รูปที่ 3.5 วงจรภาคส่งสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาครับสัญญาณ



รูปที่ 3.6 บล็อกไดอะแกรมภาครับสัญญาณ

การทำงานเริ่มจากตัวรับสัญญาณวิทยุมาจากอากาศส่งเข้าอุปกรณ์ถอดรหัส (Decoder) แล้วส่งเข้าไมโครคอนโทรลเลอร์เพื่อแปลงรหัสที่ถูกส่งมาจากไมโครคอนโทรลเลอร์ที่ภาคส่งสัญญาณ เป็นรหัสที่ใช้สำหรับควบคุมการแสดงค่าตัวเลขออกทาง LED 7 Segment ซึ่งจะมี ตัวช่วยในการจ่ายกระแส (Latch) เพื่อให้ LED ติดสว่างตามที่เราต้องการ

ภาครับสัญญาณประกอบด้วยส่วนประกอบต่างๆดังต่อไปนี้

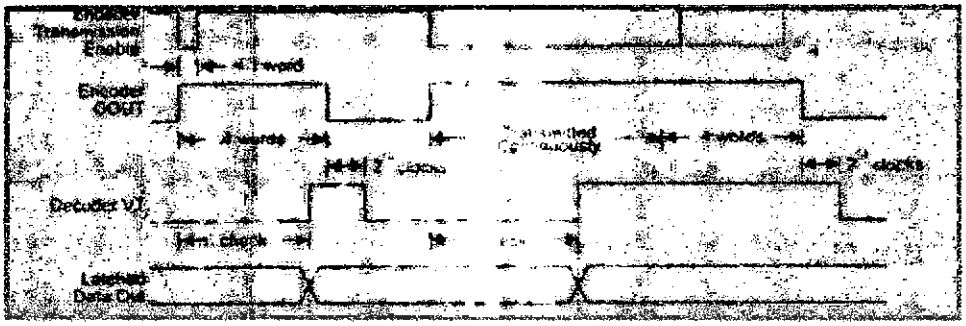
ตัวรับสัญญาณ (Receiver: RLP-434)

ทำการรับคลื่นวิทยุความถี่ 315 MHz ที่ส่งมาจาก TLP-434A แล้วทำการ Demodulation สัญญาณที่ได้แล้วส่งสัญญาณไปยังอุปกรณ์ถอดรหัสสัญญาณเพื่อนำข้อมูลที่รับได้ไปประมวลผลต่อไป

อุปกรณ์ถอดรหัสสัญญาณ (Decoder : HT12D)

ทำการรับสัญญาณแบบอนุกรมจากตัวรับสัญญาณ และทำถอดรหัสข้อมูลออกมาโดยขั้นแรกจะ check address ว่าตรงกันหรือไม่ ถ้าตรงกันข้อมูลจะถูกส่งไปยัง data output และยังคงค้างอยู่จนกว่าข้อมูลชุดใหม่จะส่งเข้ามาแทนที่ ขณะเดียวกันขา V_T (Valid Transmission Output) จะเป็น high และยังคงเป็น high จนกระทั่งเกิดความผิดพลาดขึ้นในการรับ หรือจนกระทั่งไม่ได้รับสัญญาณซึ่งมี Timing Diagram ในการรับสัญญาณและแปลงสัญญาณดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



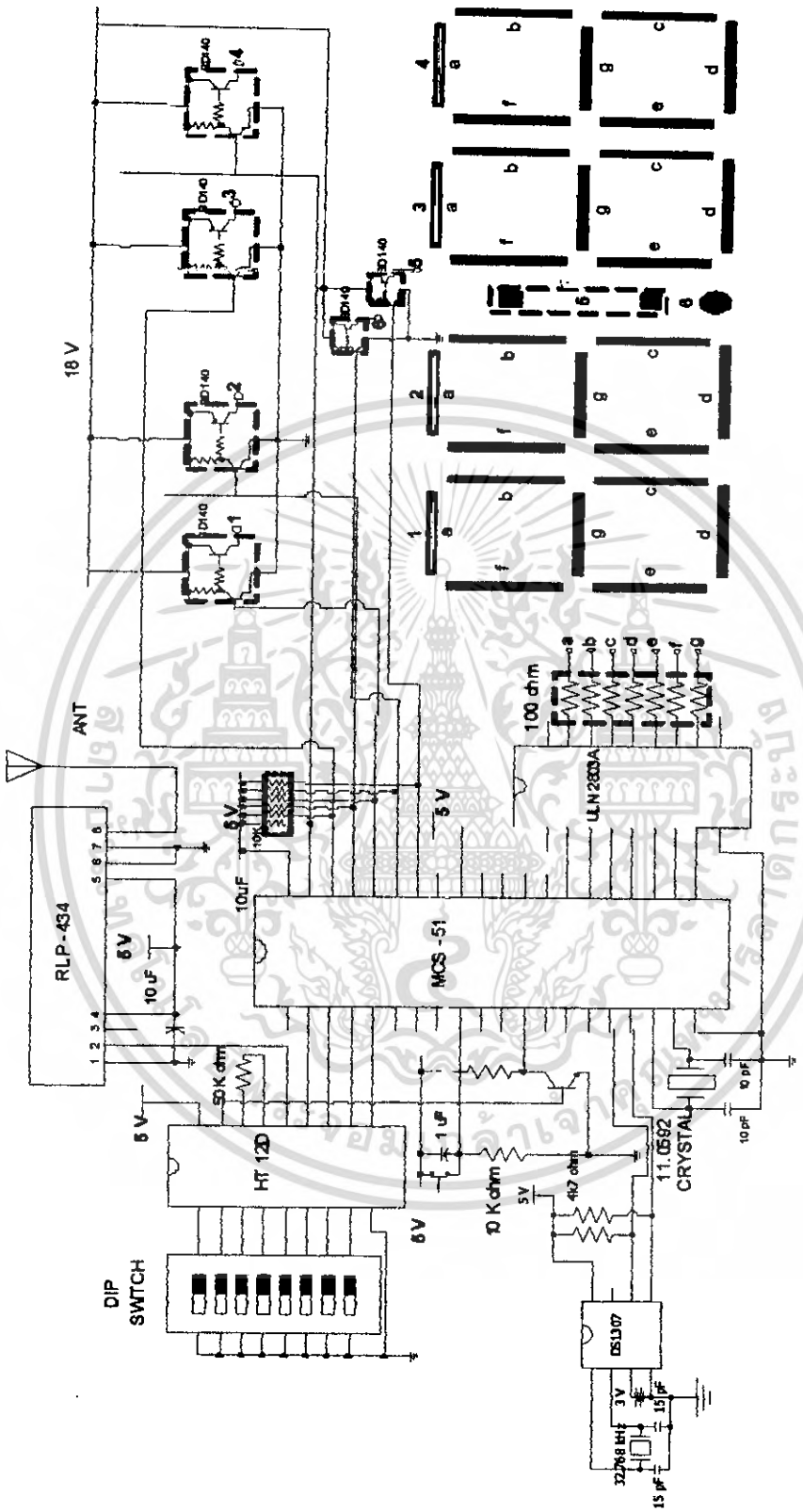
รูปที่ 3.7 Timing Diagram ของ Decoder

ส่วนไมโครคอนโทรลเลอร์

ในขณะที่มีการส่งสัญญาณขา V_T ของอุปกรณ์ถอดรหัสสัญญาณจะมีสถานะเป็น "1" เราจะต้องเปลี่ยนสถานะให้เป็น "0" จึงจะทำให้เกิดอินเตอร์รัปต์ขึ้น ไมโครคอนโทรลเลอร์จะรับค่ารหัสจากอุปกรณ์ถอดรหัสสัญญาณ เมื่อรับรหัสตัวแรกไมโครคอนโทรลเลอร์ก็จะวิเคราห์ว่าเป็นการส่งค่าราคาหรือส่งค่าเวลามาต่อมาก็จะรับรหัสตัวเลขจนครบ 4 หลัก จากนั้นจะทำการแปลงรหัสเพื่อไปควบคุม LED 7-Segment ต่อไป ซึ่งรหัสที่นำไปควบคุม LED 7-Segment เป็นดังนี้

เลข 0	P2 จะมีรหัสเป็น	01000000
เลข 1	P2 จะมีรหัสเป็น	01111001
เลข 2	P2 จะมีรหัสเป็น	00100100
เลข 3	P2 จะมีรหัสเป็น	00110000
เลข 4	P2 จะมีรหัสเป็น	00011001
เลข 5	P2 จะมีรหัสเป็น	00010010
เลข 6	P2 จะมีรหัสเป็น	00000010
เลข 7	P2 จะมีรหัสเป็น	01111000
เลข 8	P2 จะมีรหัสเป็น	00000000
เลข 9	P2 จะมีรหัสเป็น	00010000
-	P2 จะมีรหัสเป็น	00111111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

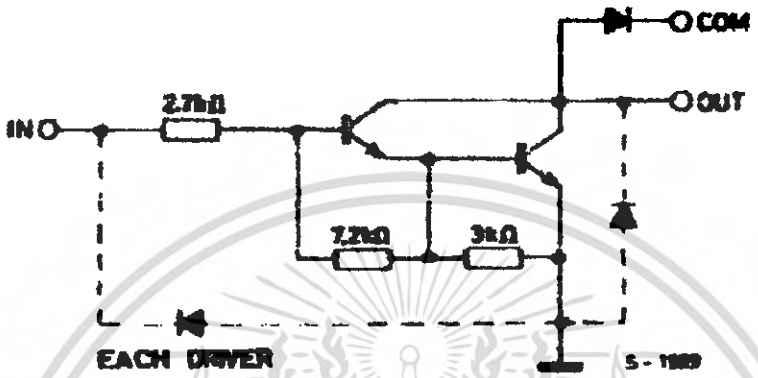


รูปที่ 3.8 วงจรภาครับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

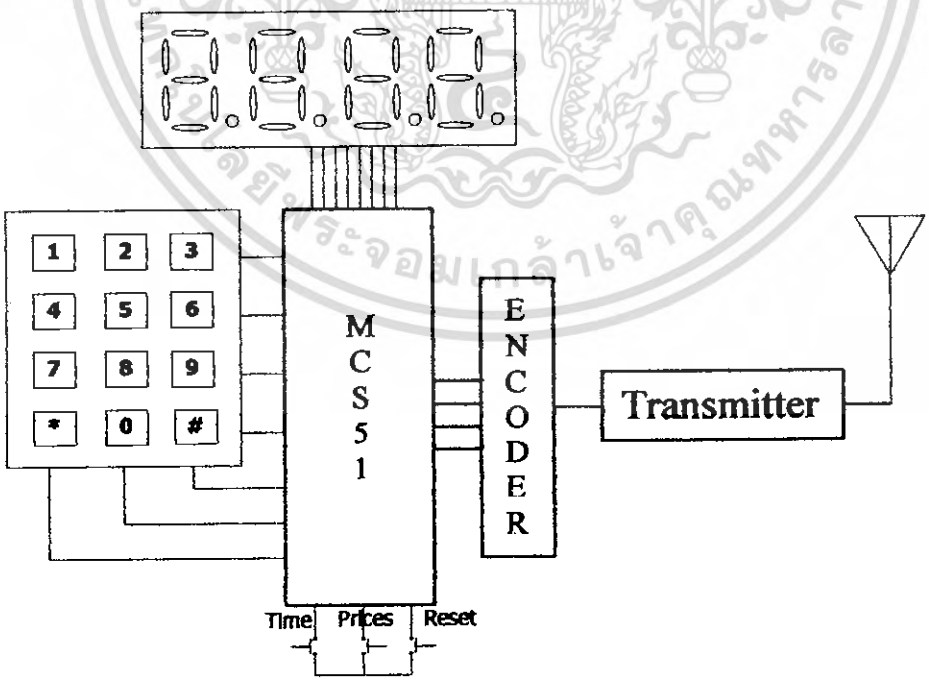
วงจรช่วยจ่ายกระแส (ULN2803)

เป็นวงจรที่ช่วยจ่ายกระแสให้กับ LED 7-Segment เพื่อให้ได้ความสว่างตามที่เราต้องการ ภายในจะประกอบด้วยทรานซิสเตอร์ต่อกันแบบคาริงตันเพื่อเพิ่มความสามารถในการจ่ายกระแสนั่นเอง



รูปที่ 3.9 วงจรภายในของ ULN2803A

ภาครีโมทควบคุม



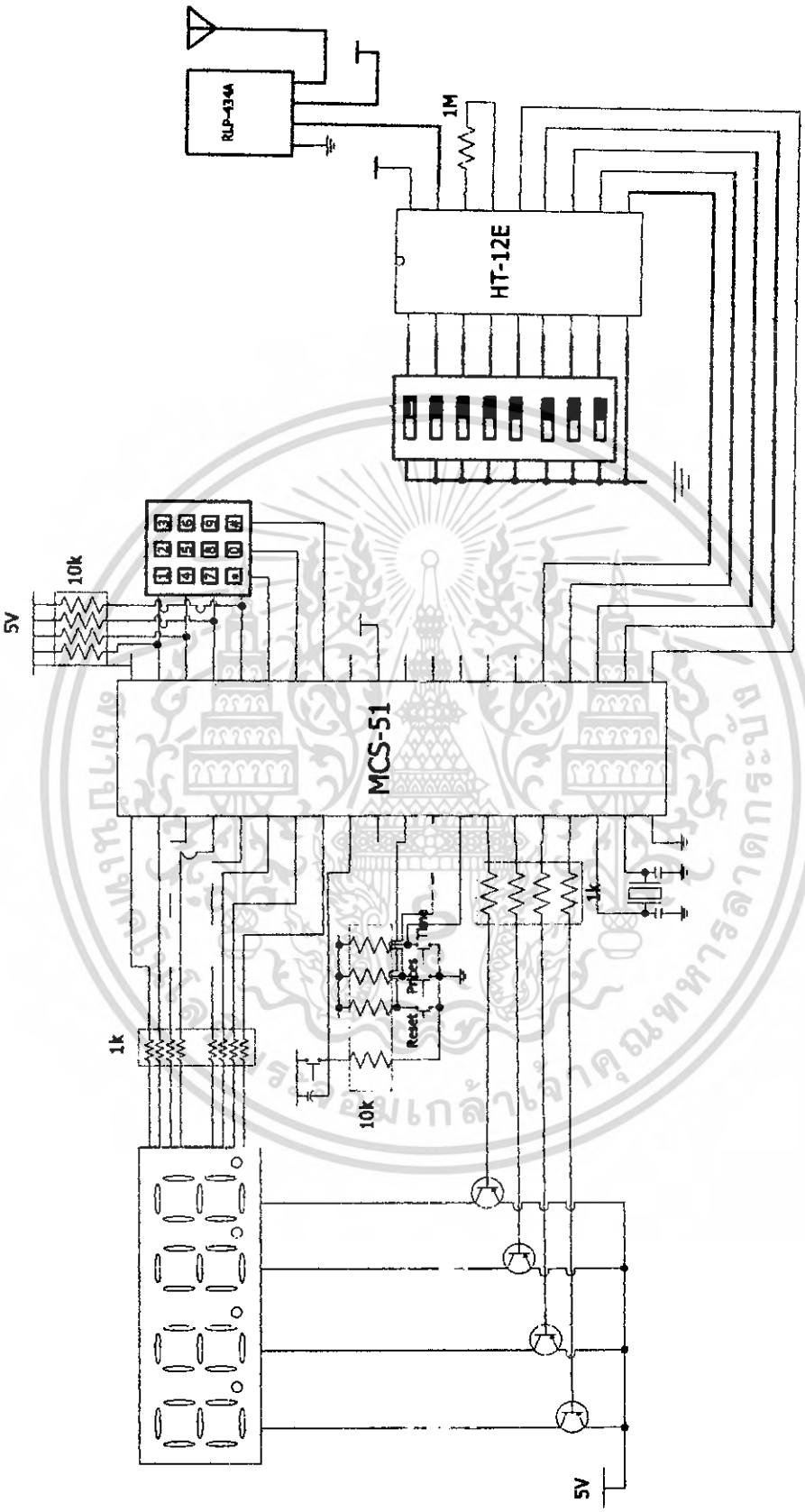
รูปที่ 3.10 บล็อกไดอะแกรมภาครีโมทควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานเริ่มจากกดสวิตช์เลือกว่าจะส่งค่าราคา เวลา หรือจะล้างข้อมูลบนบอร์ดแสดงผล ต่อจากนั้นเราก็สามารถกำหนดค่าที่เราต้องการแสดงผลโดยการกดตัวเลขบนปุ่มกด และกดปุ่ม # ไมโครคอนโทรลเลอร์จะทำการแปลงรหัสเพื่อส่งผ่านไปยังอุปกรณ์เข้ารหัสแปลงเป็นข้อมูลอนุกรมส่งให้ตัวส่งสัญญาณวิทยุส่งสัญญาณเป็นอนาลอคผ่านอากาศต่อไป โดยการส่งจะส่งออกไปทีละหลักโดยเริ่มจากรหัสของชนิดข้อมูล ตามด้วยหลักสิบ หลักหน่วยและสตางค์ ตามลำดับ รหัสแต่ละตัวจะถูกกำหนดเหมือนกับภาคส่งสัญญาณ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 วงจรภาครีโมทควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

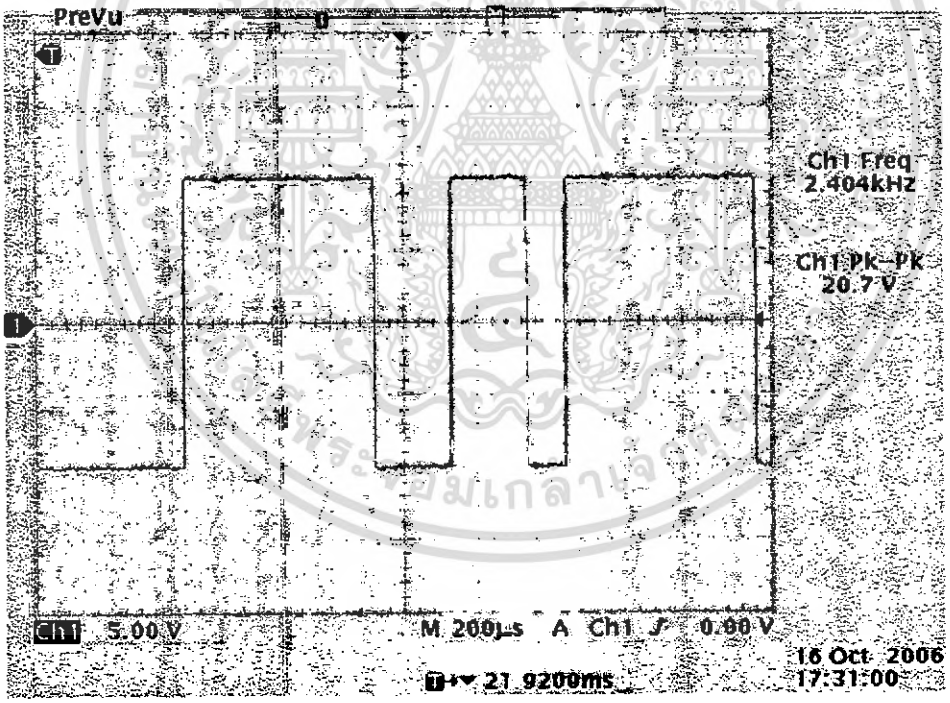
บทที่ 4
การทดลอง

4.1 การทดสอบสัญญาณที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

สัญญาณที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์นั้นจะประกอบด้วย

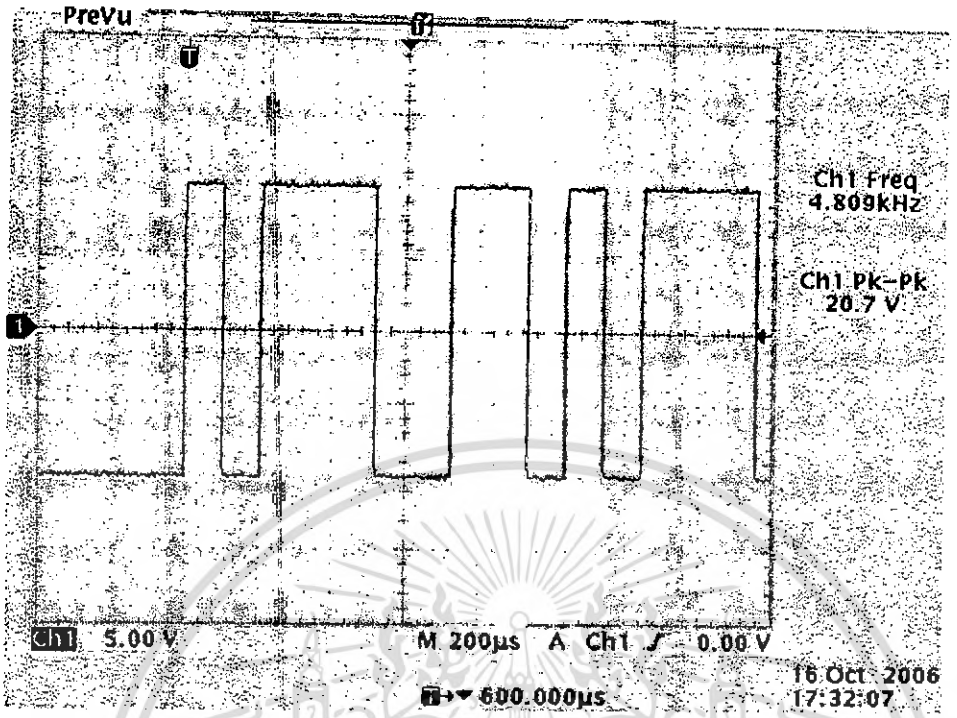
1. บิตเริ่มต้น 1 บิต สถานะ Low
2. บิตข้อมูล 8 บิต ซึ่งเป็นรหัส ASCII ของแต่ละตัวเลข เช่น เลข 0 จะมีรหัส ASCII เป็น 0x30 หรือ 00110000 นั่นเอง
3. บิตจบ 1 บิต สถานะ High

สัญญาณที่ออกมาจะอยู่ในรูปของแรงดัน 2 ระดับ คือ ± 12 V. โดยที่บิต " 1 " หรือสภาวะ High จะถูกแทนด้วยระดับแรงดัน –12 V. และบิต " 0 " หรือสภาวะ Low จะถูกแทนด้วยระดับแรงดัน +12 V.

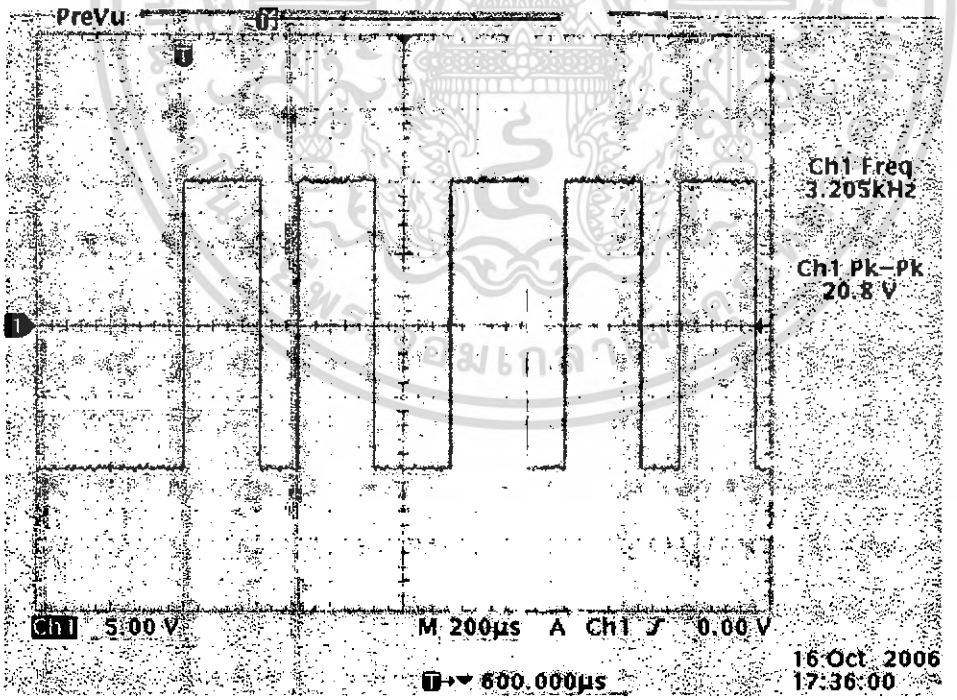


รูปที่ 4.1 สัญญาณเลข 0 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

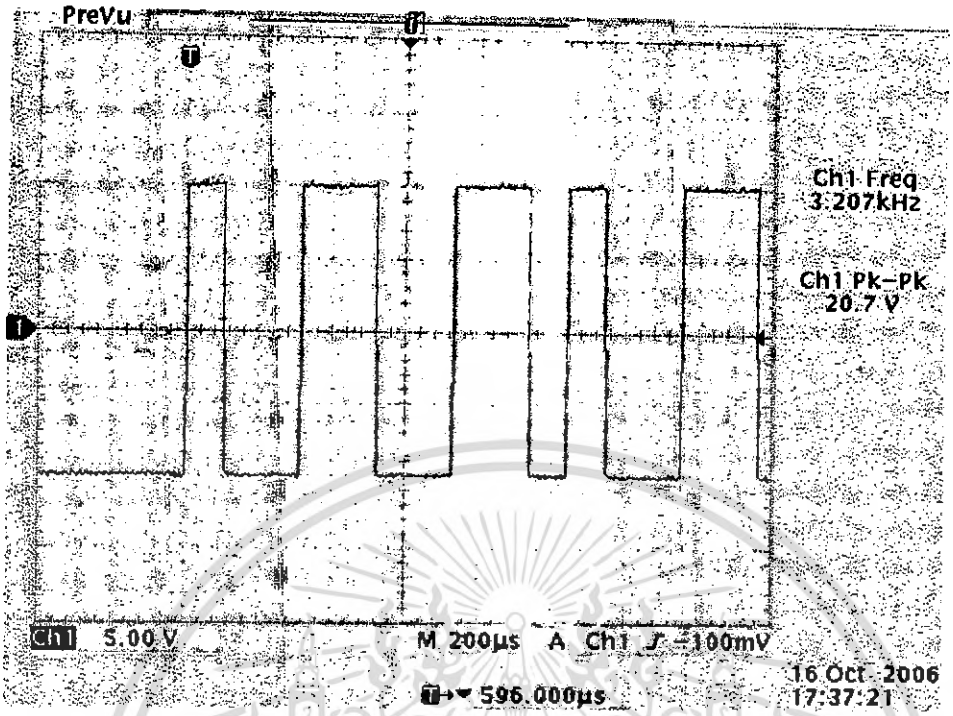


รูปที่ 4.2 สัญญาณเลข 1 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

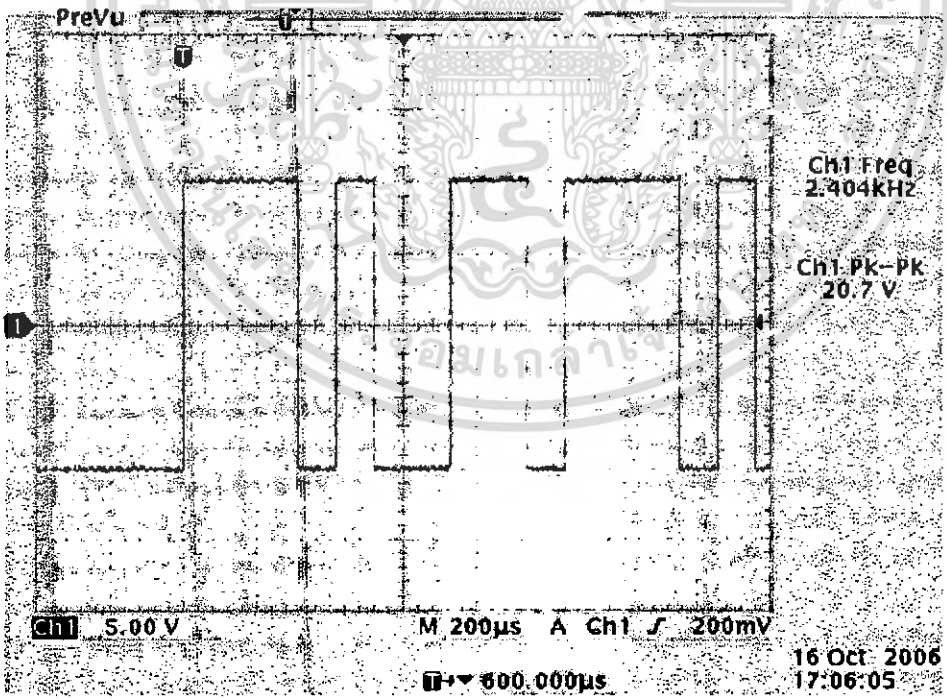


รูปที่ 4.3 สัญญาณเลข 2 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

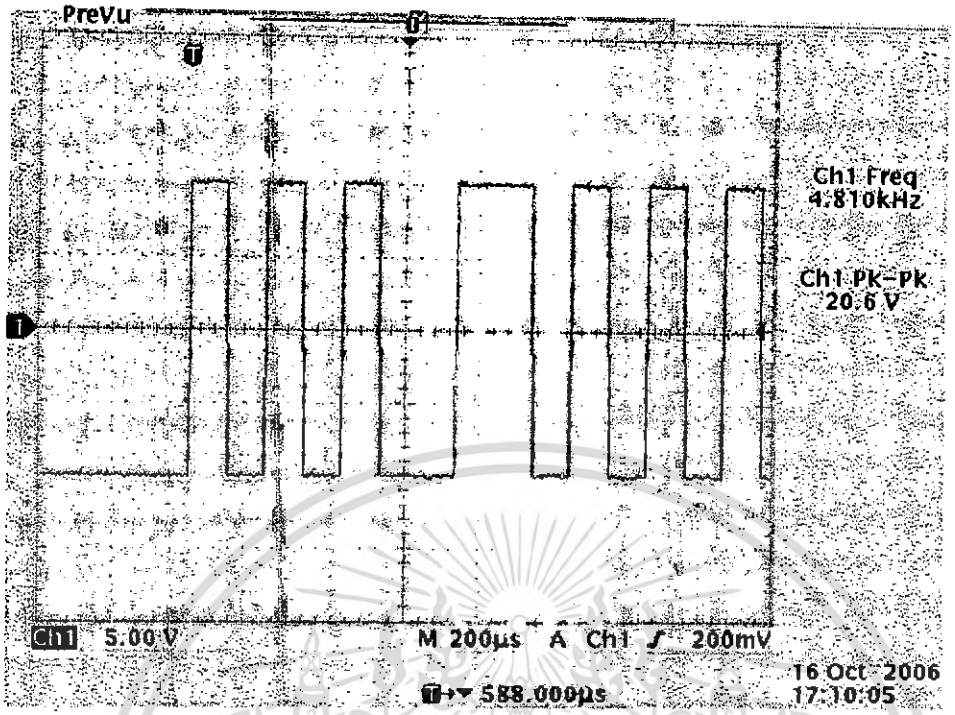


รูปที่ 4.4 สัญญาณเลข 3 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

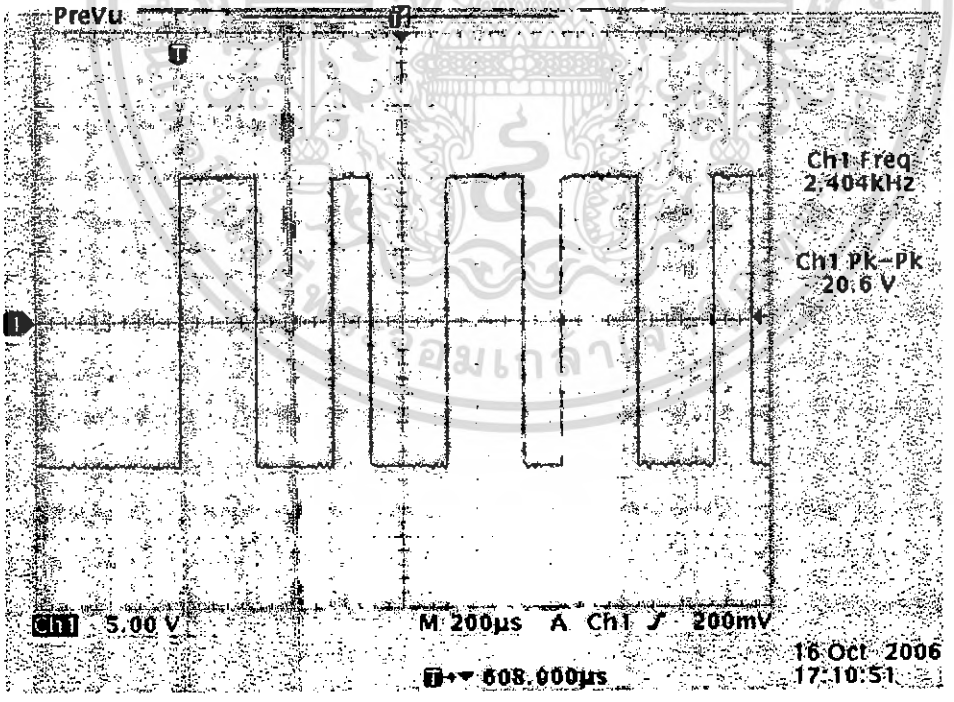


รูปที่ 4.5 สัญญาณเลข 4 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

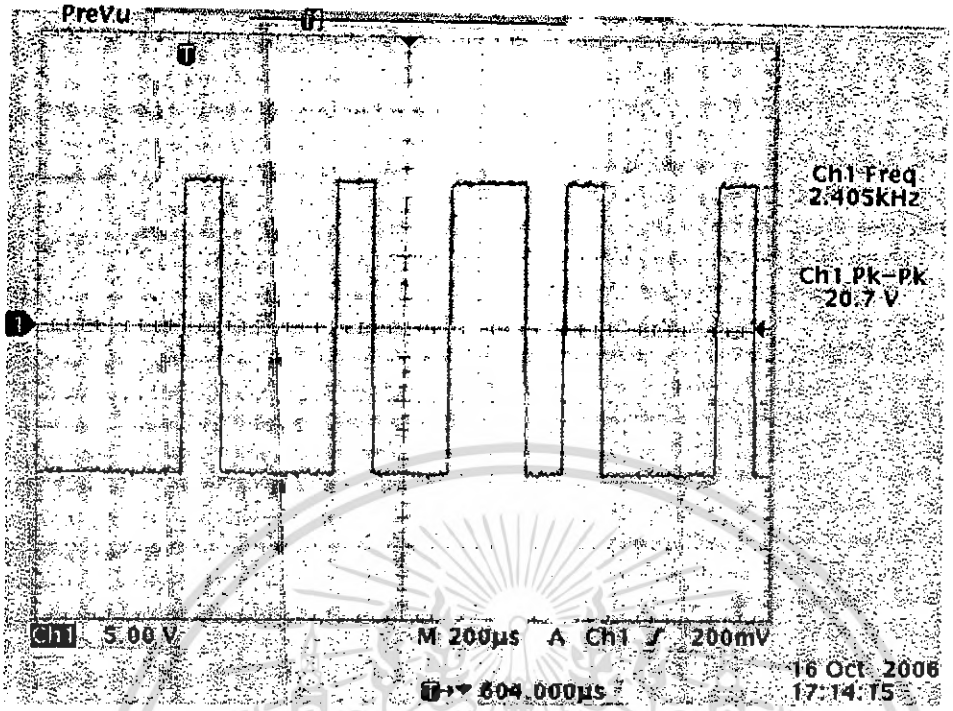


รูปที่ 4.6 สัญญาณเลข 5 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

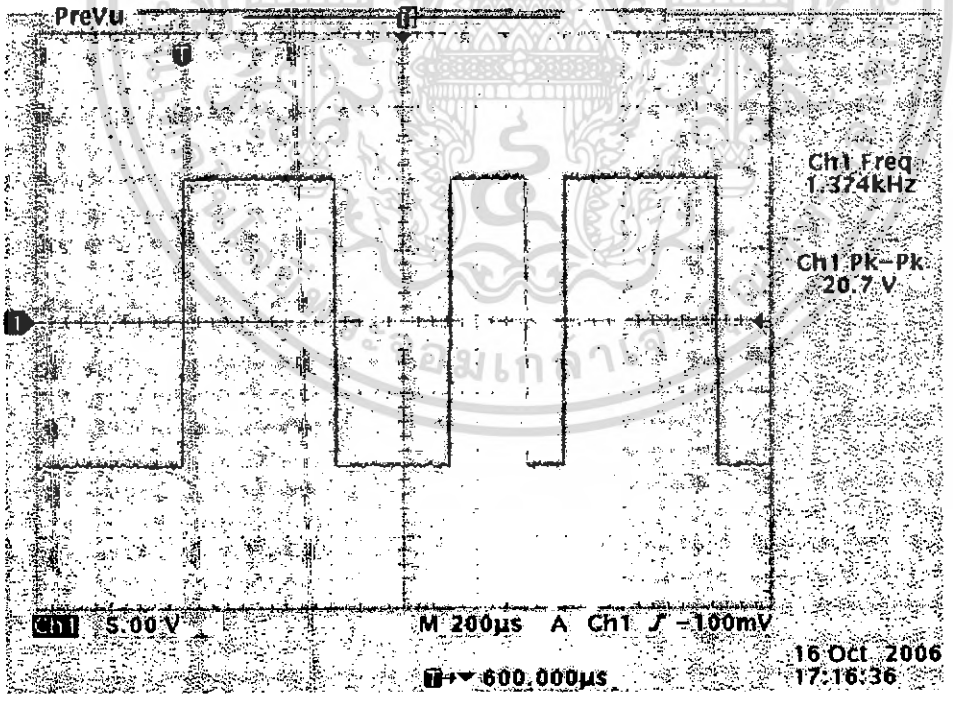


รูปที่ 4.7 สัญญาณเลข 6 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

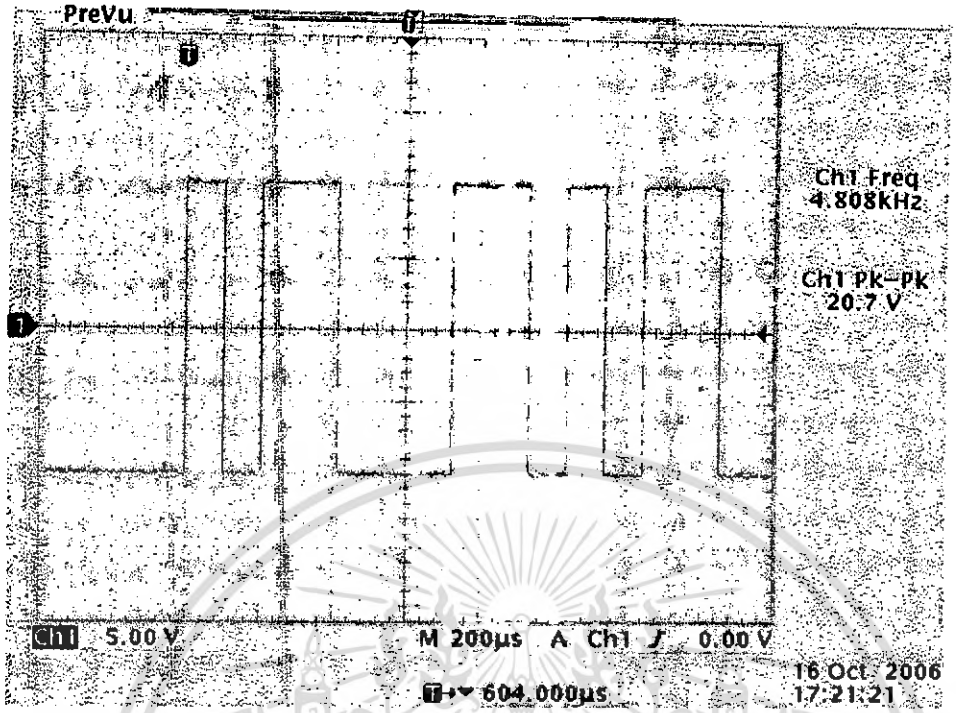


รูปที่ 4.8 สัญญาณเลข 7 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

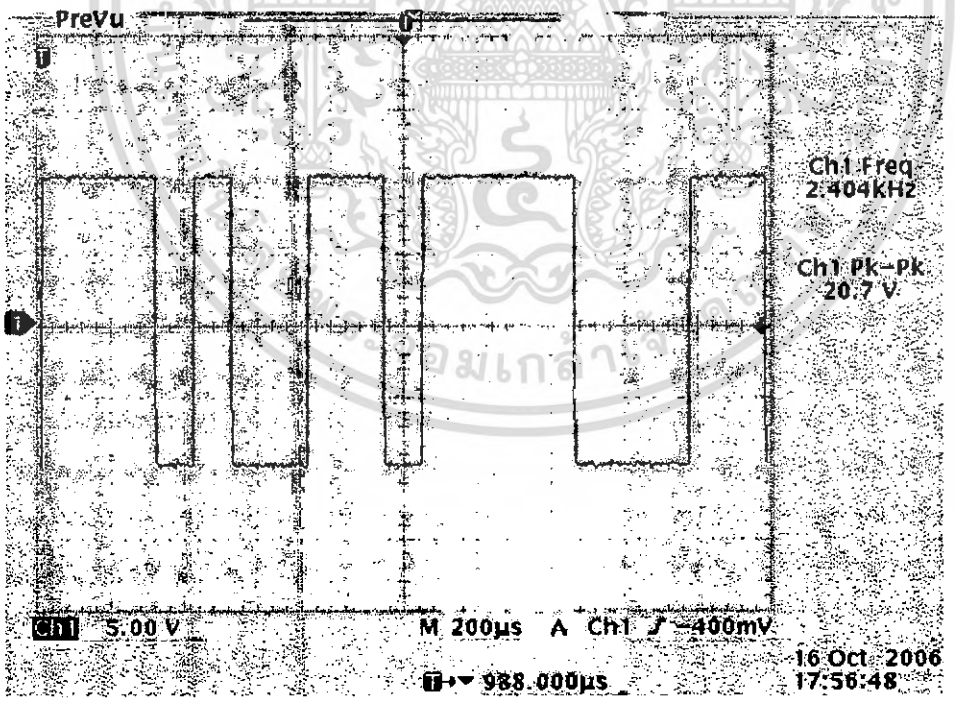


รูปที่ 4.9 สัญญาณเลข 8 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 สัญญาณเลข 9 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

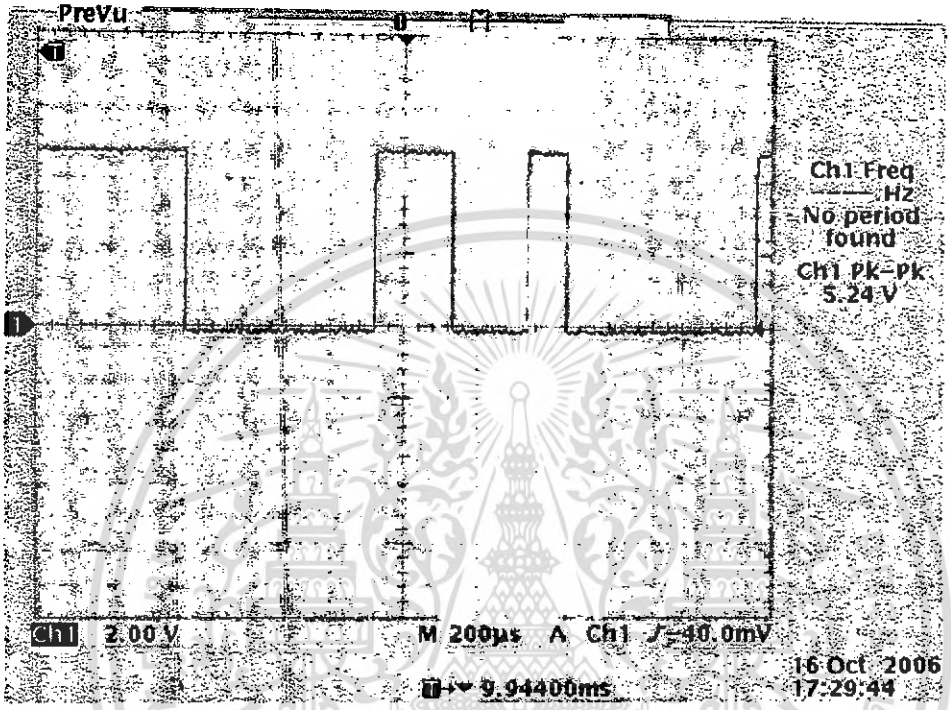


รูปที่ 4.11 สัญญาณเลข 48 ที่ออกจากพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

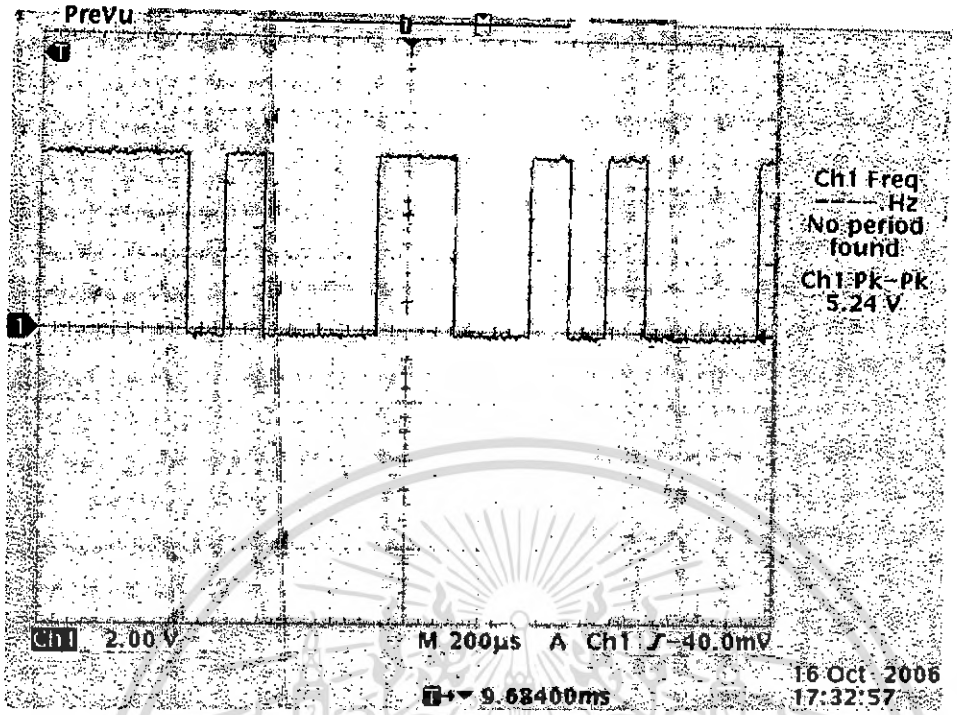
4.2 สัญญาณหลังจากผ่าน RS-232

สัญญาณที่ได้จะถูกกลับแรงดัน จาก –12 V เป็น 5 V หรือสถานะ " 1 " และ จาก 12 V เป็น 0 V หรือสถานะ " 0 "

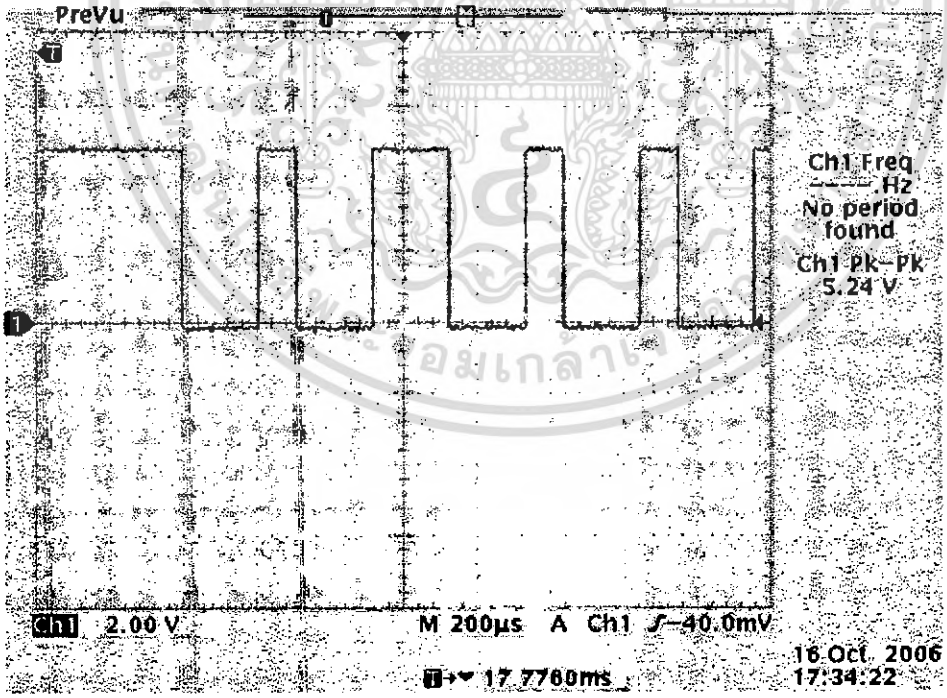


รูปที่ 4.12 สัญญาณเลข 0 หลังจากผ่าน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

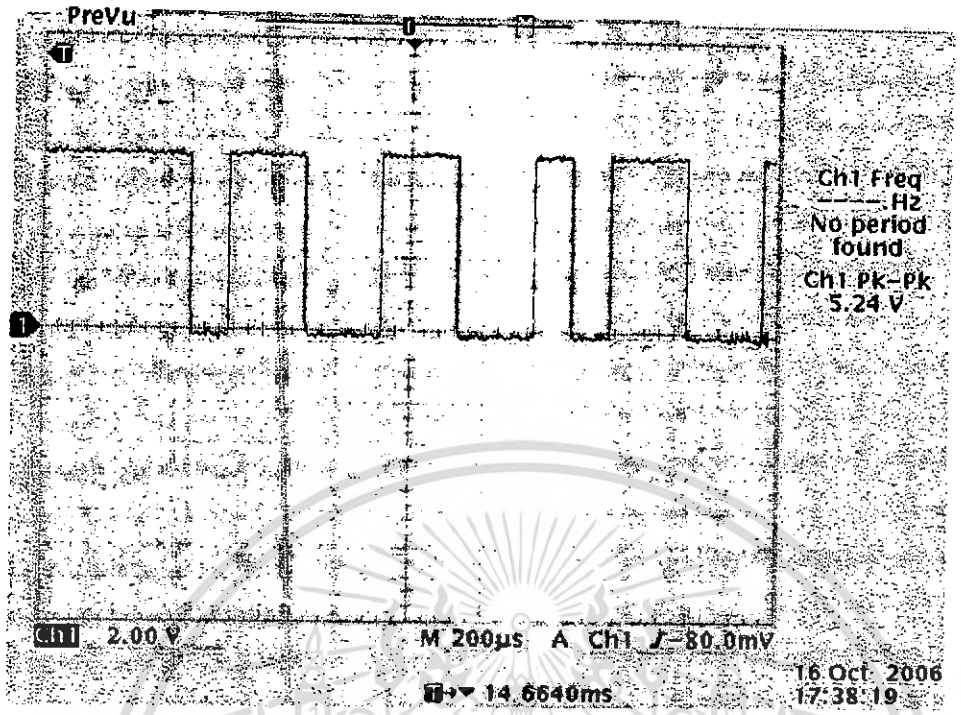


รูปที่ 4.13 สัญญาณเลข 1 หลังจากผ่าน RS-232

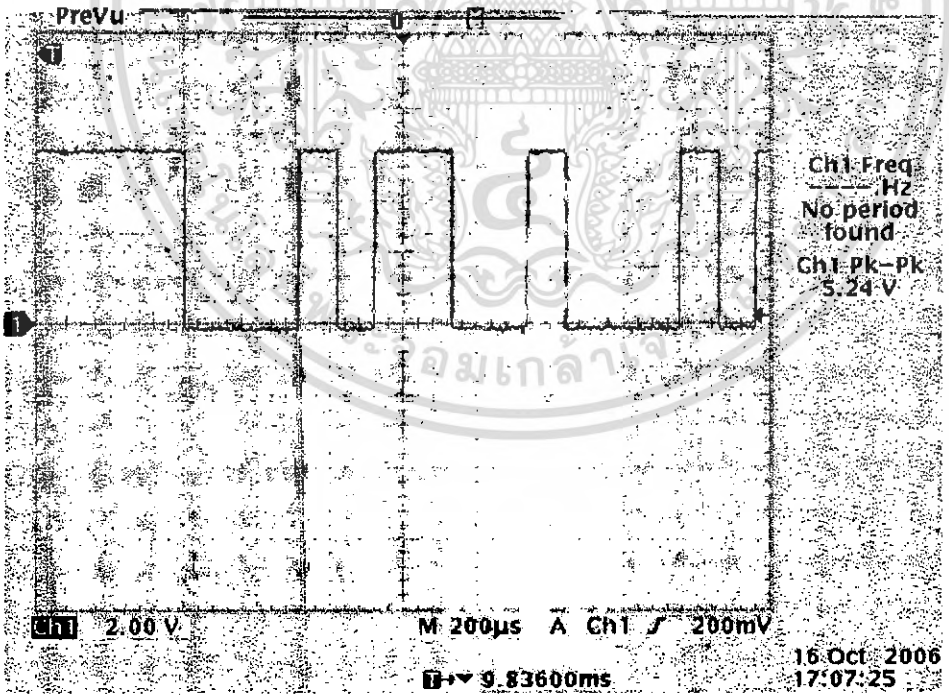


รูปที่ 4.14 สัญญาณเลข 2 หลังจากผ่าน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

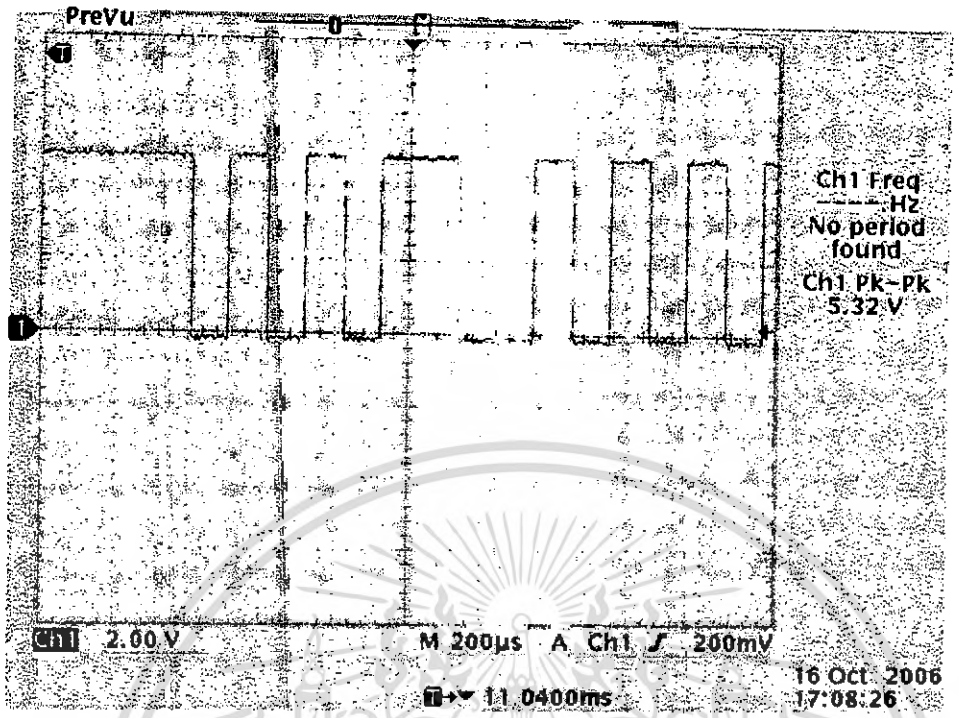


รูปที่ 4.15 สัญญาณเลข 3 หลังจากผ่าน RS-232

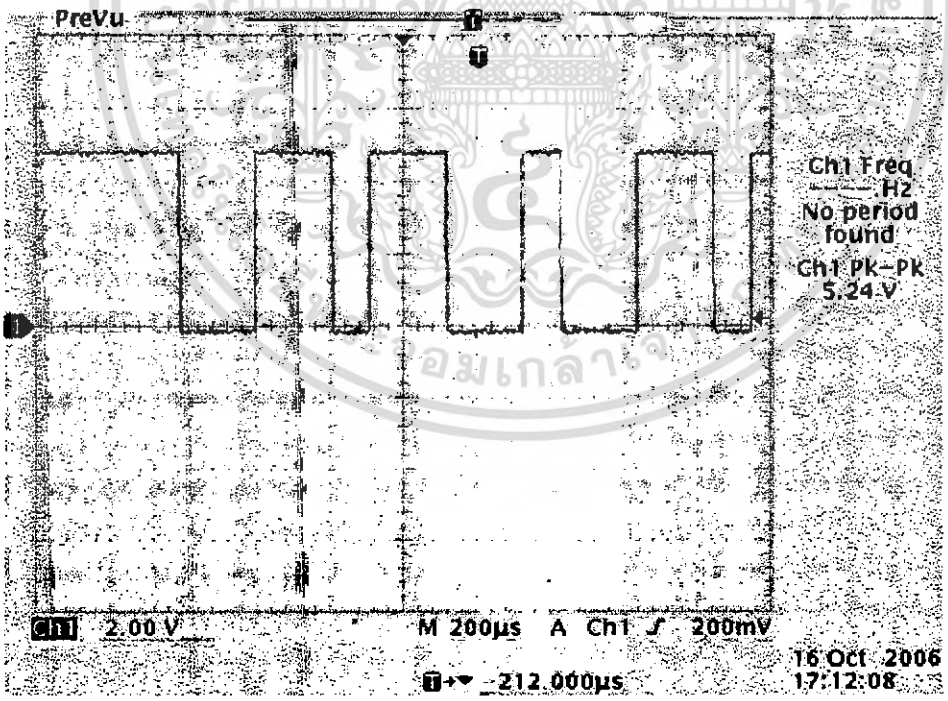


รูปที่ 4.16 สัญญาณเลข 4 หลังจากผ่าน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

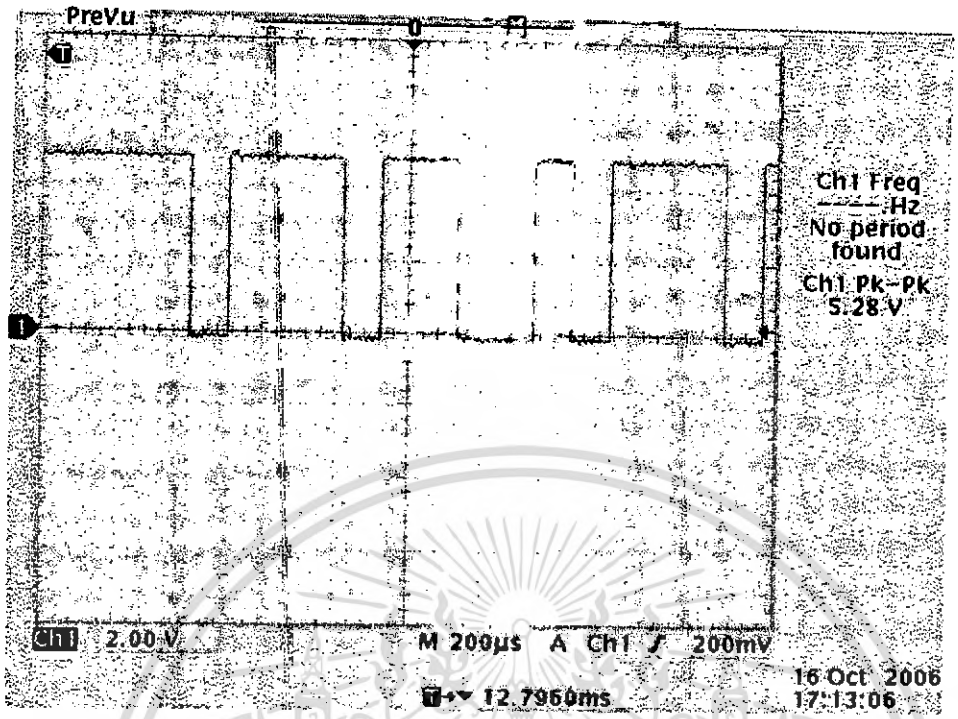


รูปที่ 4.17 สัญญาณเลข 5 หลังจากผ่าน RS-232

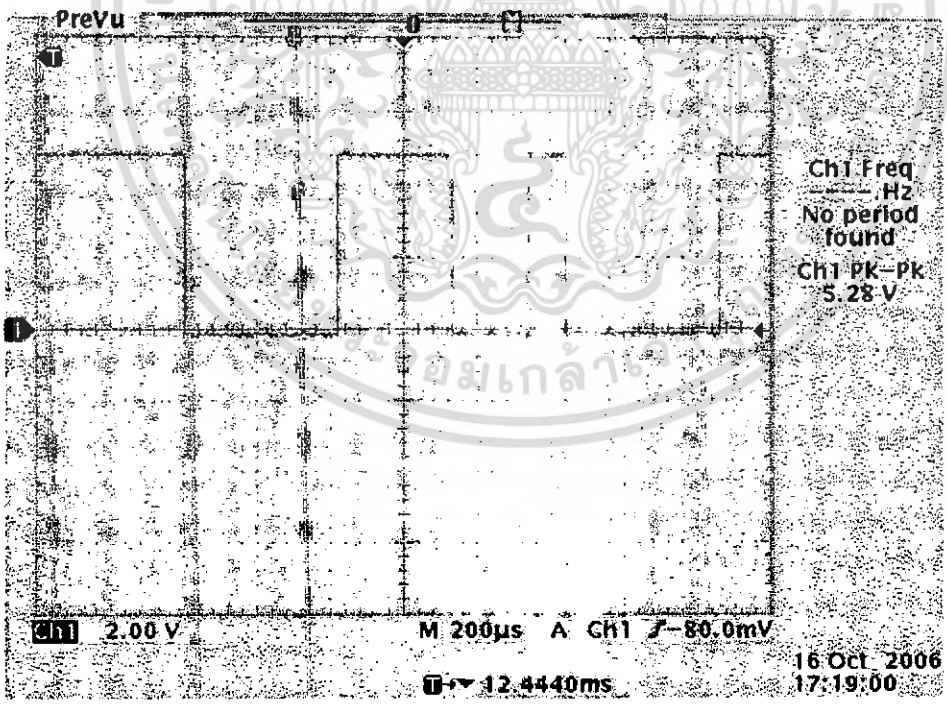


รูปที่ 4.18 สัญญาณเลข 6 หลังจากผ่าน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

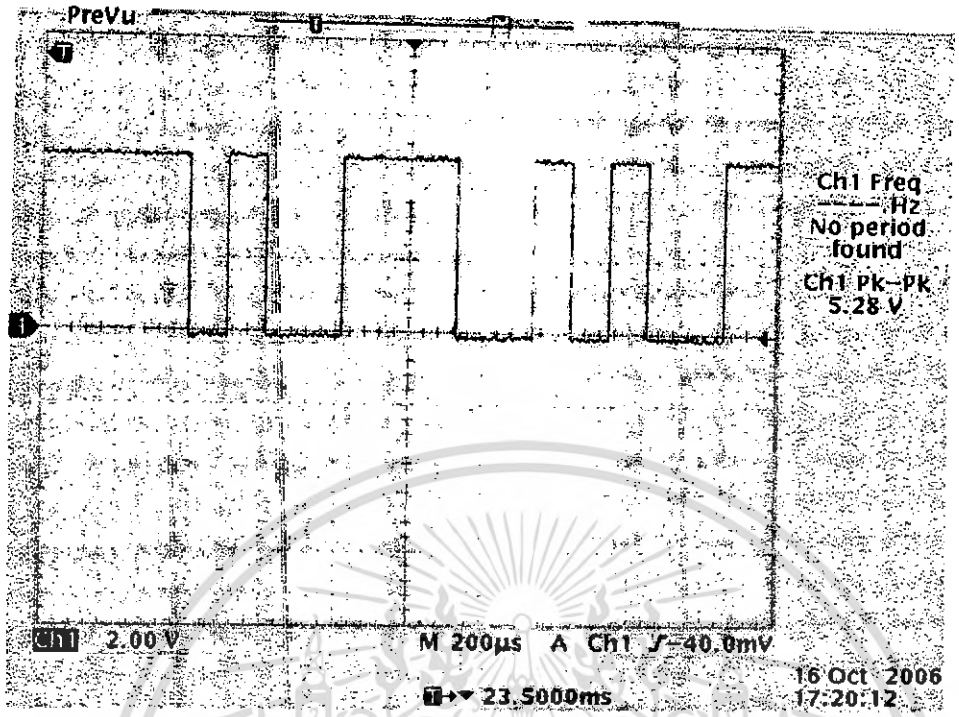


รูปที่ 4.19 สัญญาณเลข 7 หลังจากผ่าน RS-232

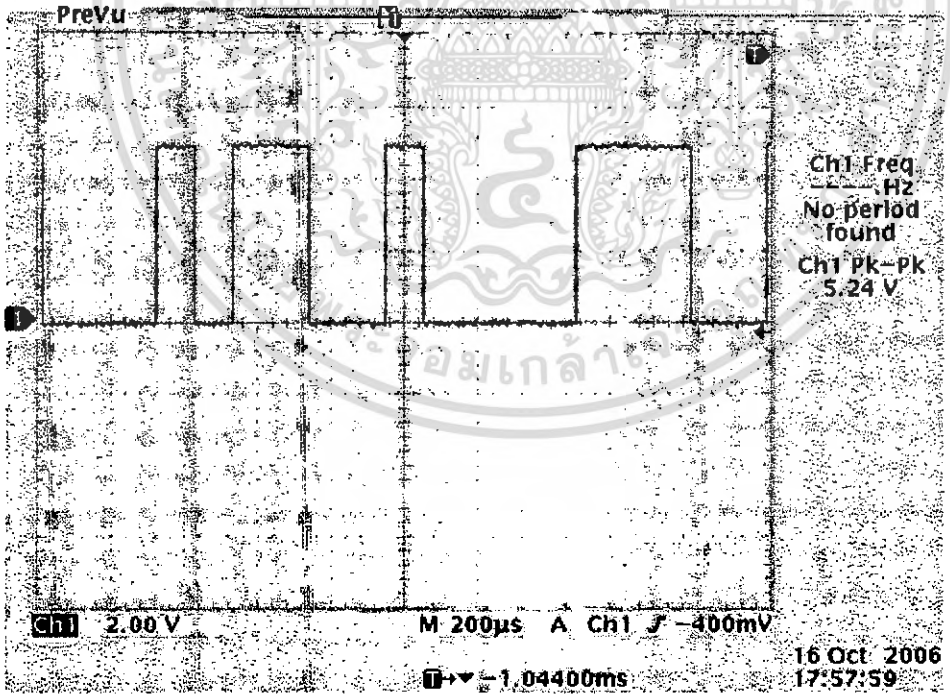


รูปที่ 4.20 สัญญาณเลข 8 หลังจากผ่าน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 สัญญาณเลข 9 หลังจากผ่าน RS-232



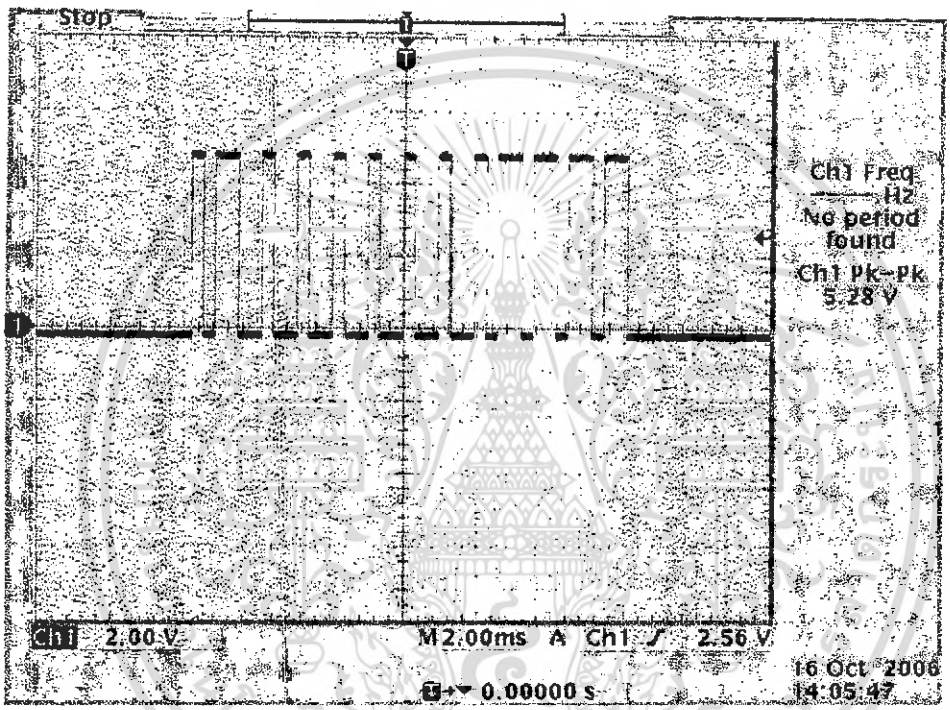
รูปที่ 4.22 สัญญาณเลข 48 หลังจากผ่าน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 สัญญาณหลังจากการเข้ารหัส

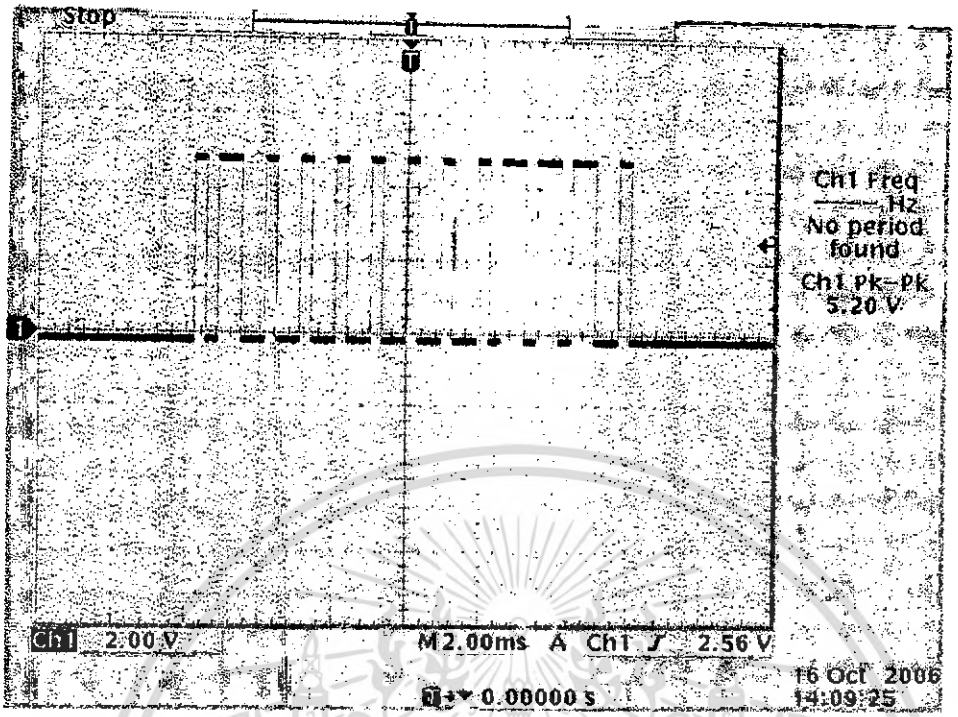
เป็นการนำบิตต่างๆที่เรากำหนดเป็น Address และข้อมูลให้กับอุปกรณ์เข้ารหัสสัญญาณมาทำเป็นข้อมูลที่อนุกรมกันก่อนที่จะส่งผ่านตัวส่งสัญญาณวิทยุต่อไปโดยประกอบด้วย

1. บิตเริ่มต้น	1 บิต	สถานะ Low
2. บิต Address	8 บิต	ในที่นี้เรากำหนด Address ไว้เป็น 01111111
3. บิตข้อมูล	4 บิต	สถานะตามรหัสตัวเลขนั้นๆ

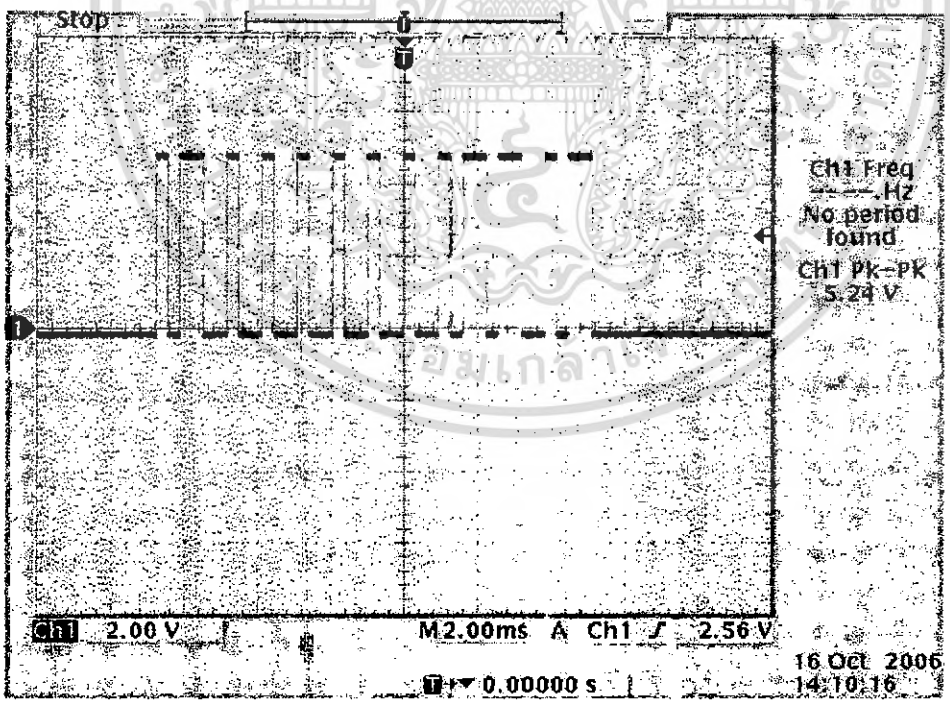


รูปที่ 4.23 สัญญาณเลข 0 หลังจากการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

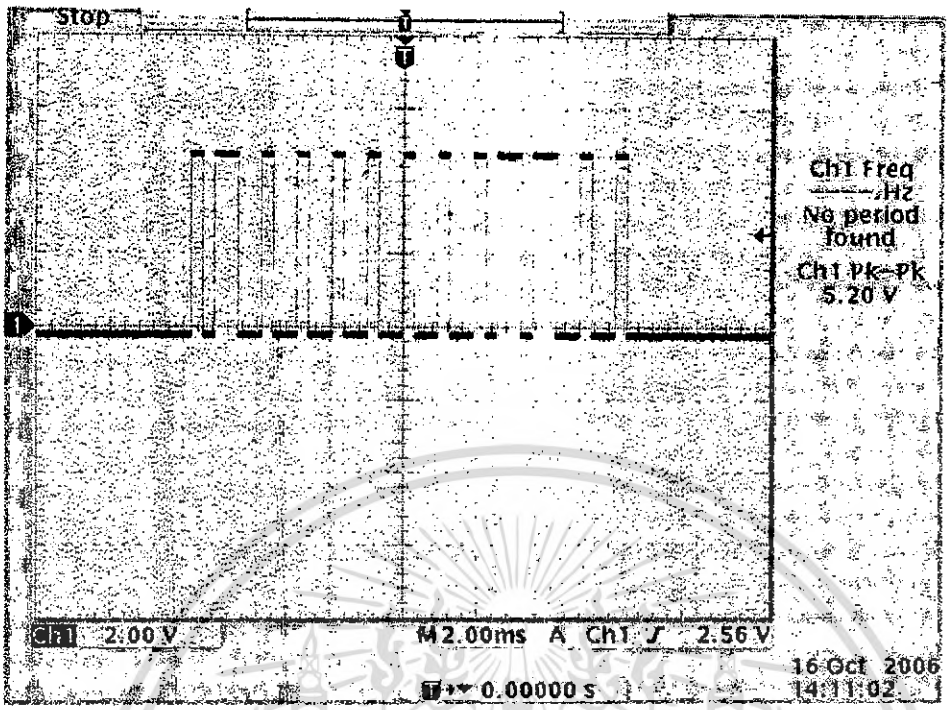


รูปที่ 4.24 สัญญาณเลข 1 หลังจากการเข้ารหัส

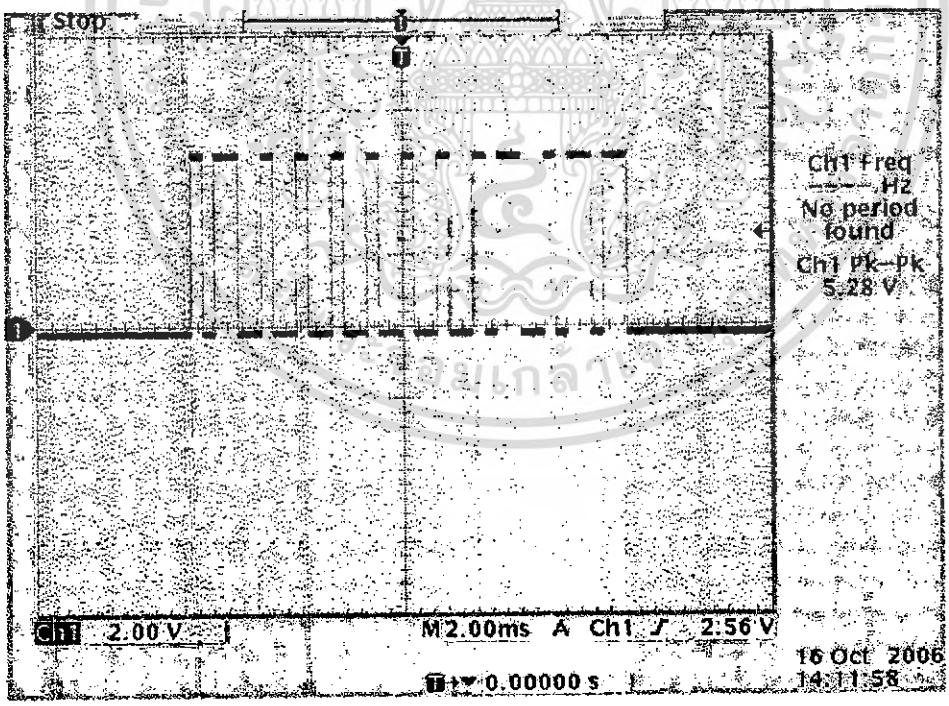


รูปที่ 4.25 สัญญาณเลข 2 หลังจากการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

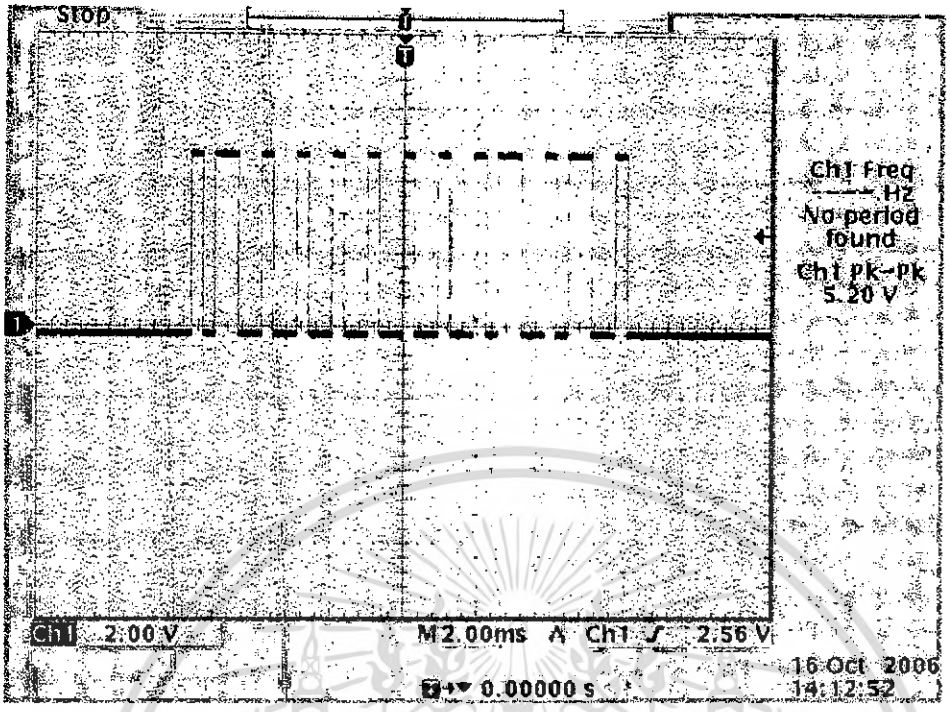


รูปที่ 4.26 สัญญาณเลข 3 หลังจากการเข้ารหัส

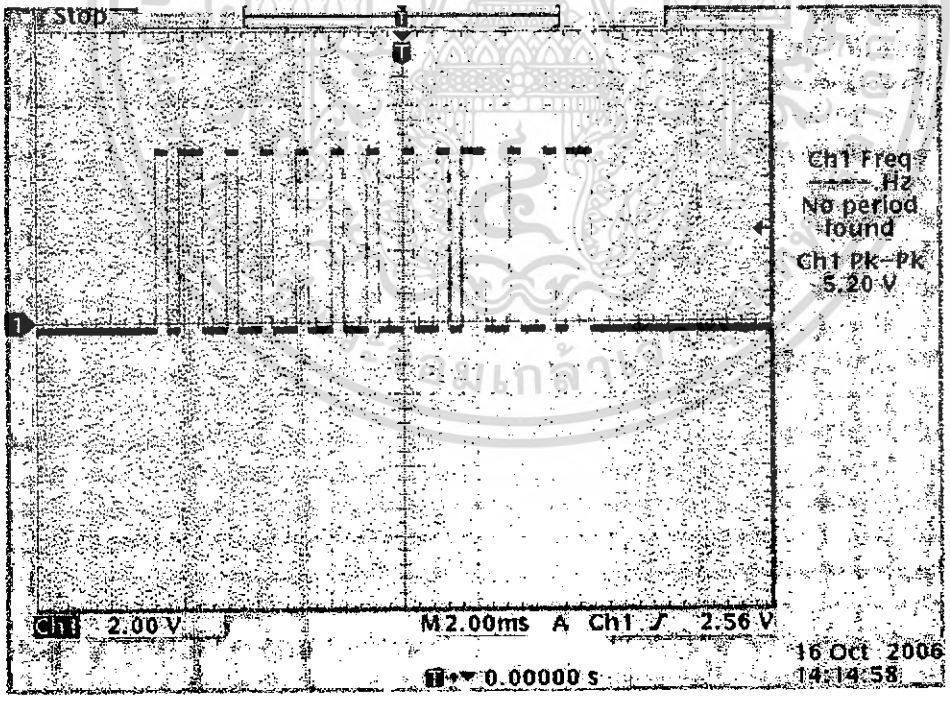


รูปที่ 4.27 สัญญาณเลข 4 หลังจากการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

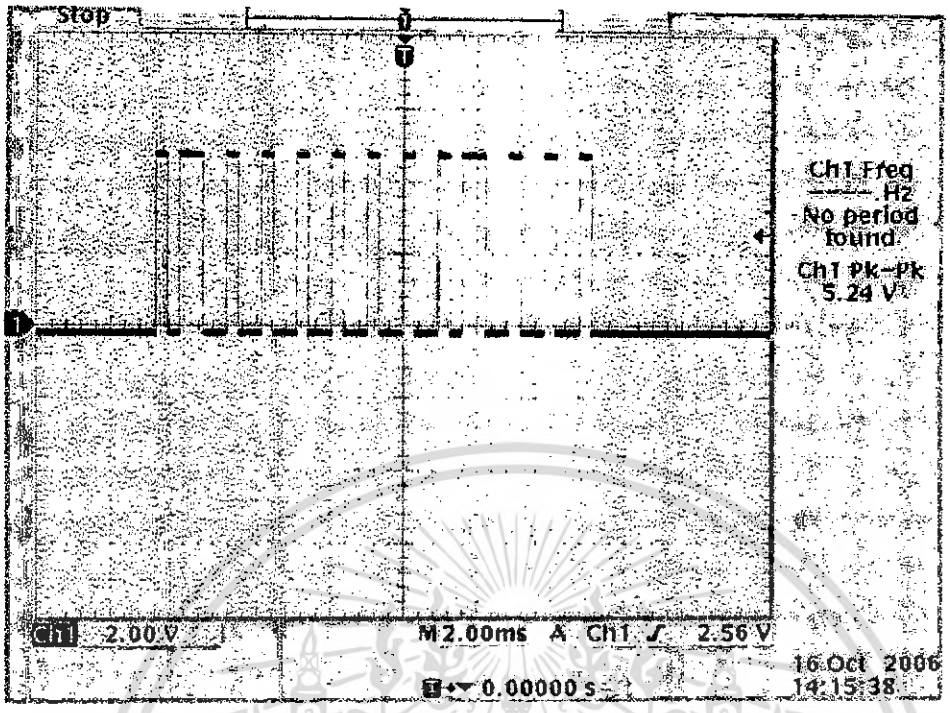


รูปที่ 4.28 สัญญาณเลข 5 หลังจากการเข้ารหัส

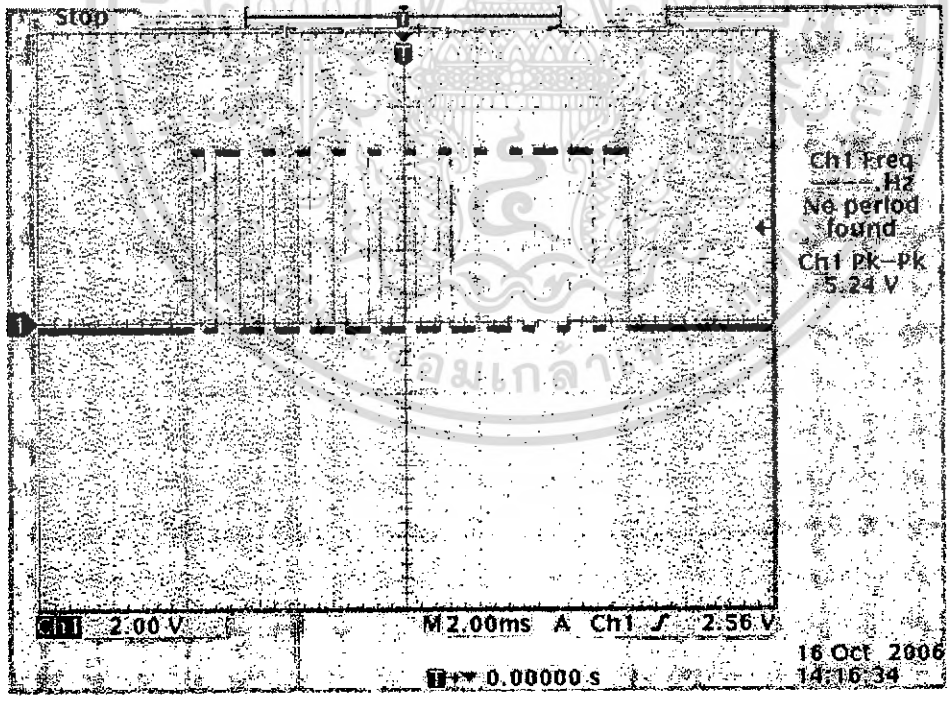


รูปที่ 4.29 สัญญาณเลข 6 หลังจากการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

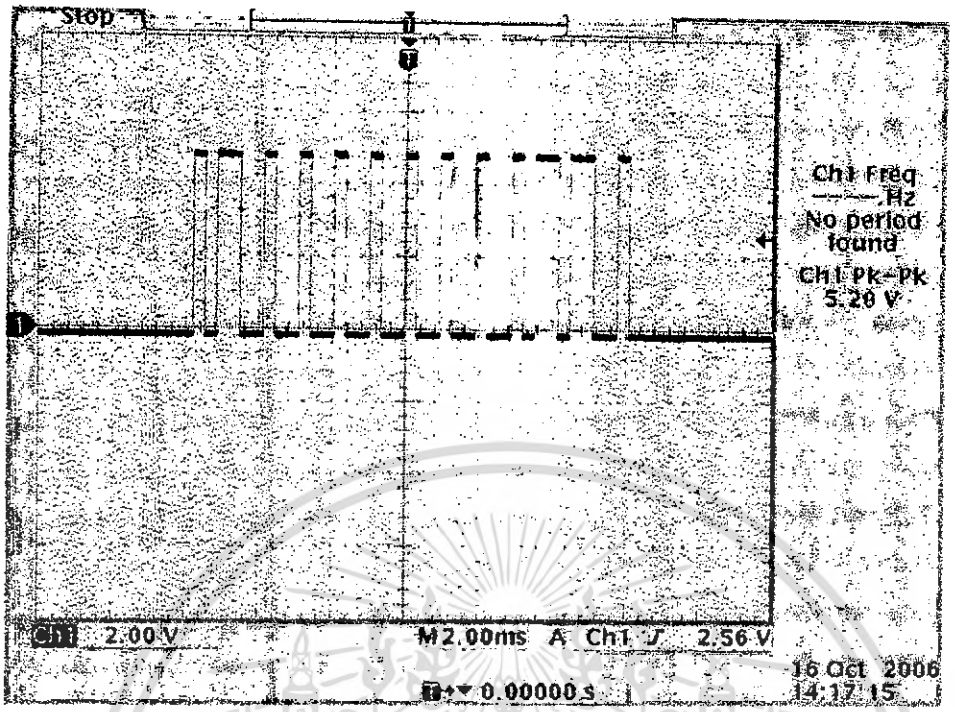


รูปที่ 4.30 สัญญาณเลข 7 หลังจากการเข้ารหัส



รูปที่ 4.31 สัญญาณเลข 8 หลังจากการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 สัญญาณเลข 9 หลังจากการเข้ารหัส

สรุปผลการทดลอง

จากการทดลองวัดสัญญาณมาทั้งหมดนี้แสดงให้เห็นว่าข้อมูลที่เราต้องการส่งถูกส่งออกมาได้ตามที่เรากำหนดไว้อย่างถูกต้องแม่นยำ จากกราฟสัญญาณต่างๆเราจะสังเกตได้ว่าเส้นกราฟจะไม่เรียบ ซึ่งอาจมีผลมาจากสัญญาณรบกวนของสภาพแวดล้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์โครงการ

5.1 สรุปผลการปฏิบัติงาน

การปฏิบัติงานเริ่มต้นจากการค้นคว้าข้อมูลที่เกี่ยวข้องกับป้ายแสดงราคาน้ำมัน รวมทั้งทำความเข้าใจ Data Sheet ของอุปกรณ์ต่างๆที่ต้องใช้ในการทำป้ายราคาน้ำมัน เพื่อที่จะนำข้อมูลและความรู้ที่ได้ไปออกแบบและทำการทดลองต่อไป

ผลการทดลองเป็นไปตามทฤษฎีแต่อาจมีสัญญาณรบกวนเข้ามาบ้างเมื่อมีการส่งสัญญาณระยะไกลๆ และความถี่ Oscillator มีความคลาดเคลื่อนบ้างเนื่องจากไม่สามารถหาค่าความต้านทานที่คำนวณได้ในท้องตลาด

5.2 ปัญหาและอุปสรรค

1. ใช้เวลาในการศึกษาโปรแกรม Visual Basic 6.0 และการติดต่อสื่อสารผ่านพอร์ตอนุกรมนานทำให้งานล่าช้าเพราะเป็นการศึกษาครั้งแรก
2. ใช้เวลาในการศึกษาโปรแกรมภาษา C และโปรแกรมที่เกี่ยวข้องกับการเขียนโปรแกรมภาษา C เป็นเวลานานทำให้งานล่าช้าเพราะเป็นการศึกษาเป็นครั้งแรก
3. ใช้โปรแกรมยังไม่คล่องงานจึงล่าช้า
3. มีสัญญาณรบกวนจากสิ่งแวดล้อมทำให้เกิดปัญหาในการทดลองส่งสัญญาณ
4. อุปกรณ์บางชนิดที่หาได้จากการคำนวณไม่มีขายตามท้องตลาด ทำให้ต้องใช้ค่าใกล้เคียง

5.3 แนวทางในการแก้ไข

1. เริ่มทำงานให้เร็วขึ้น
2. พยายามทำความเข้าใจในเนื้อหาให้เร็วขึ้น
3. ออกแบบแผงวงจรเพื่อต่ออุปกรณ์ลงบนแผ่นปริ้นเพื่อลดปัญหาเรื่องสัญญาณรบกวน
4. ใช้งานโปรแกรมให้บ่อยขึ้นเพื่อให้เกิดความชำนาญ
5. การออกแบบต้องคำนึงถึงอุปกรณ์ที่มีขายจริงในท้องตลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ประโยชน์และการประยุกต์ใช้งาน

เนื่องจากราคาน้ำมันในปัจจุบันมีการเปลี่ยนแปลงไปอย่างรวดเร็วทำให้ผู้ให้บริการน้ำมันไม่สะดวกต่อการเปลี่ยนแผ่นป้ายราคา เนื่องจากเดิมการแสดงราคาเป็นแบบแผ่นป้ายทำให้ต้องใช้คนขึ้นไปเปลี่ยนแผ่นป้ายราคาบ่อยครั้ง รวมทั้งในปัจจุบันนี้ได้มีการพัฒนาเทคโนโลยีทางด้านอิเล็กทรอนิกส์ให้เจริญก้าวหน้าไปอย่างมาก ทำให้เครื่องมืออุปกรณ์ต่างๆเปลี่ยนแปลงไปอย่างรวดเร็วรวมทั้งเทคโนโลยีทางด้านการสื่อสารด้วย ในการทำป้ายแสดงราคาน้ำมันนี้ทำให้เราทราบวิธีการส่งข้อมูลข่าวสารแบบต่างๆ การแปลงข้อมูล ซึ่งสามารถนำไปประยุกต์ใช้งานในด้านการสื่อสาร และทราบวิธีการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ทางอิเล็คทรอนิคส์ ซึ่งสามารถพัฒนาเป็นแผ่นป้ายโฆษณาเพื่อดึงดูดความสนใจของลูกค้าได้

5.5 สิ่งที่ได้จากการทำโครงงานนี้

1. การศึกษาการสื่อสารแบบต่างๆซึ่งจะเป็นพื้นฐานสำคัญในการพัฒนาระบบสื่อสารในอนาคตต่อไป

2. ได้ศึกษาโปรแกรม Visual Basic 6.0 และการติดต่อสื่อสารผ่านพอร์ตอนุกรม ทำให้ทราบการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ทางอิเล็คทรอนิกส์

3. ได้ศึกษาโปรแกรมภาษา C และโปรแกรมที่เกี่ยวข้องกับการเขียนโปรแกรมภาษา C ทำให้สามารถที่จะเขียนโปรแกรมสั่งงาน Microcontroller เพื่อใช้ในการควบคุมอุปกรณ์ทางอิเล็กทรอนิกส์ได้

นอกจากนี้แล้วยังได้รับคำแนะนำจากอาจารย์ที่ปรึกษามากมาย ทำให้รู้ถึงการนำไปประยุกต์โครงงานเพื่อทำประโยชน์อื่นได้ เช่น การทำ Display Board เพื่อใช้ในการโฆษณา หรือแสดงข้อมูลเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล , " เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 " . กรุงเทพฯ ; บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด , 2521

ธีรบูลย์ หล่อวิเชียรรุ่ง และคณะ , " ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C ". กรุงเทพฯ ; บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด , 2521

สัจจะ จรัสรุ่งรวีวร , " คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6 " . พิมพ์ครั้งที่ 1 . กรุงเทพฯ ; อินโฟเพรส , 2544

อภิชาต ภู่พลับ , " เริ่มต้นเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์ด้วย Visual Basic " . พิมพ์ครั้งที่ 1 . นนทบุรี ; อินโฟเพรส , 2546

อภิชาต ภู่พลับ , " เขียนโปรแกรม Hardware Interface ด้วย Visual Basic 6 " . พิมพ์ครั้งที่ 1 . นนทบุรี ; ไอดีซีฯ , 2548

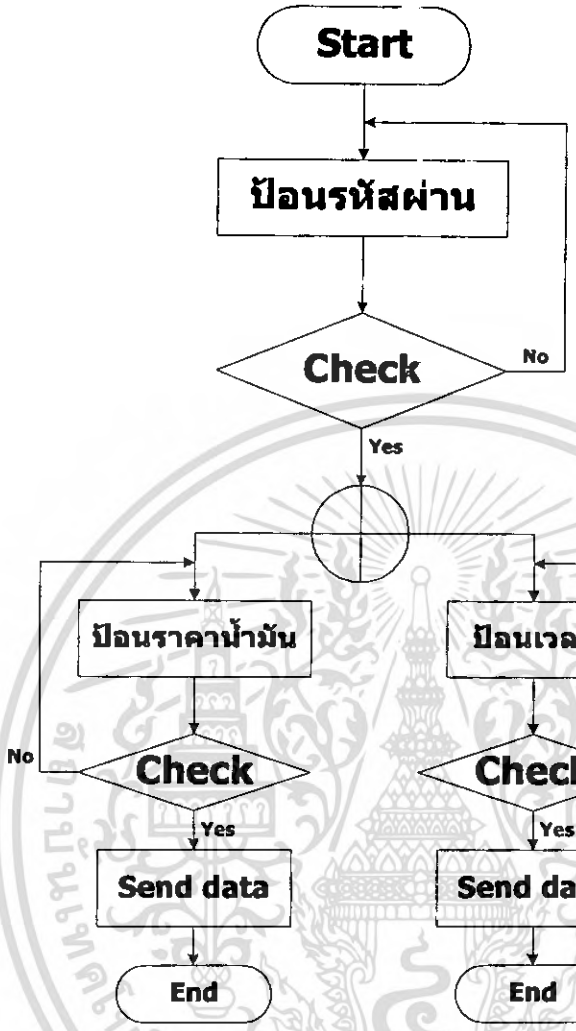
สมเกียรติ ศุภเดช , " เซมีคอนดัคเตอร์ ดีไวซ์ " . พิมพ์ครั้งที่ 2 . คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2547

สัญฉกร วุฒิสิทธิกุลกิจ , " หลักการไฟฟ้าสื่อสาร " . พิมพ์ครั้งที่ 1 . คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย , 2546

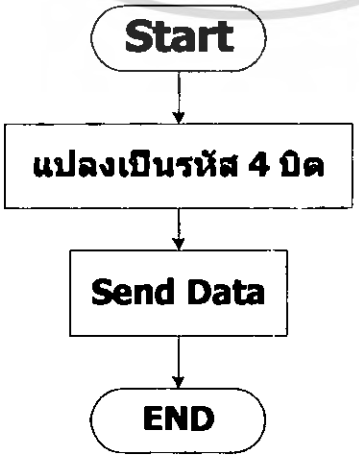
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

Flow Chart การทำงานของ Page Visual Basic

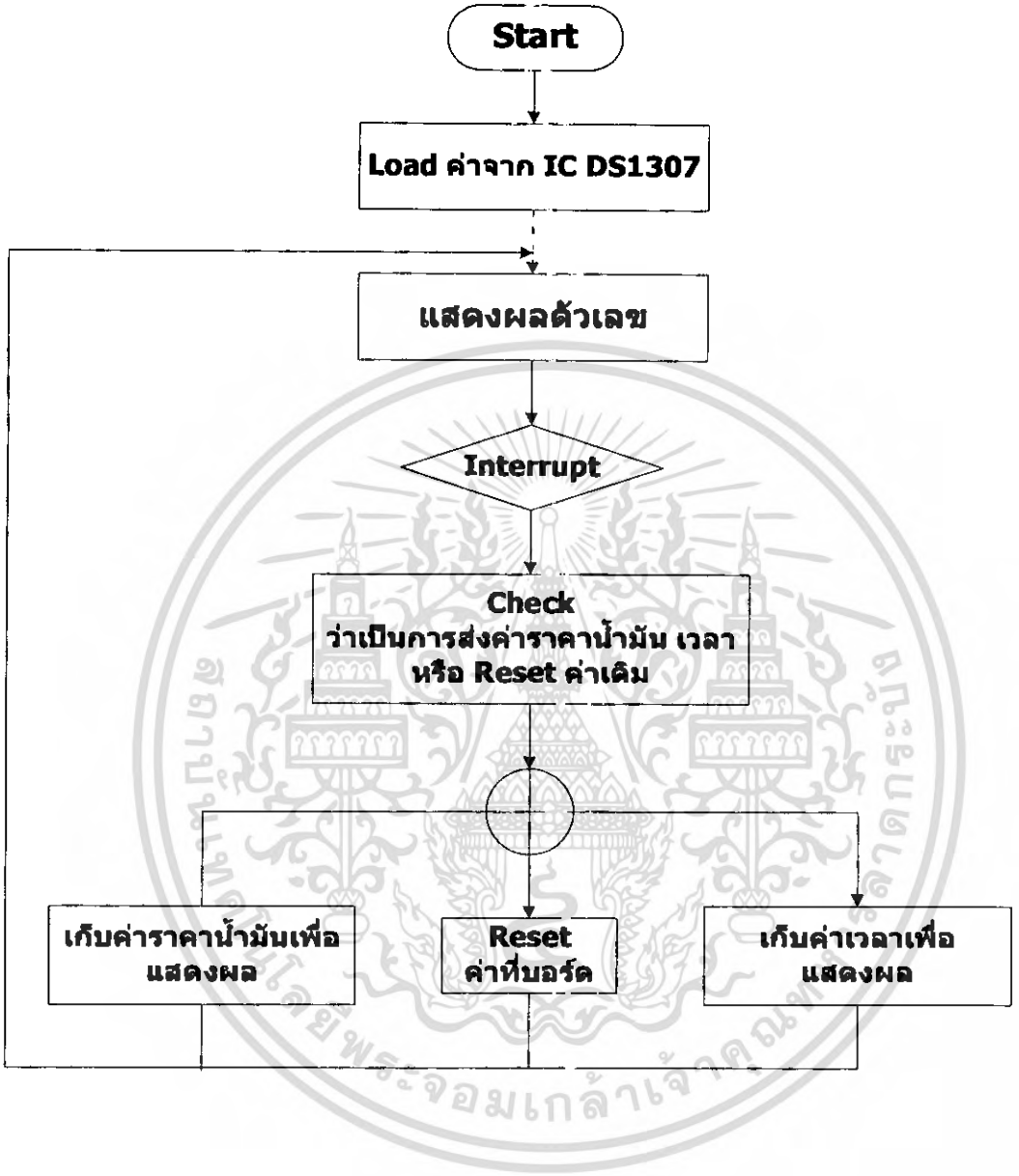


Flow Chart การทำงานของภาคส่งสัญญาณ



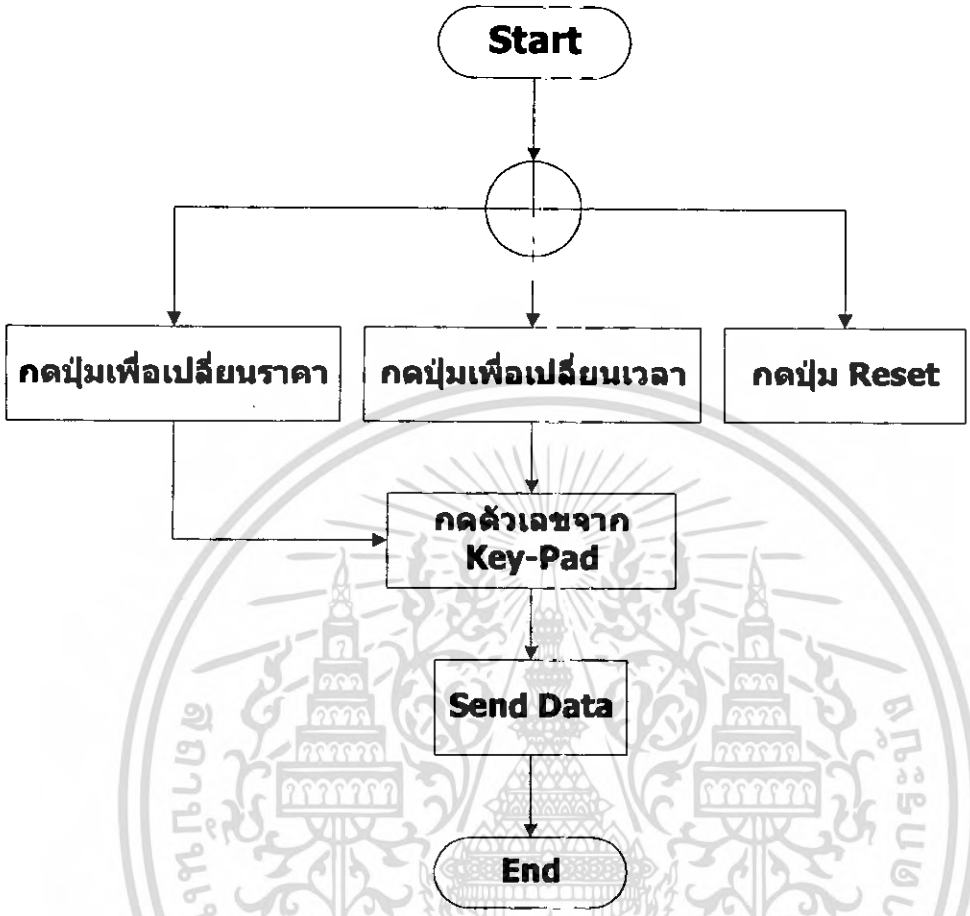
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chart การทำงานของภาครับสัญญาณ

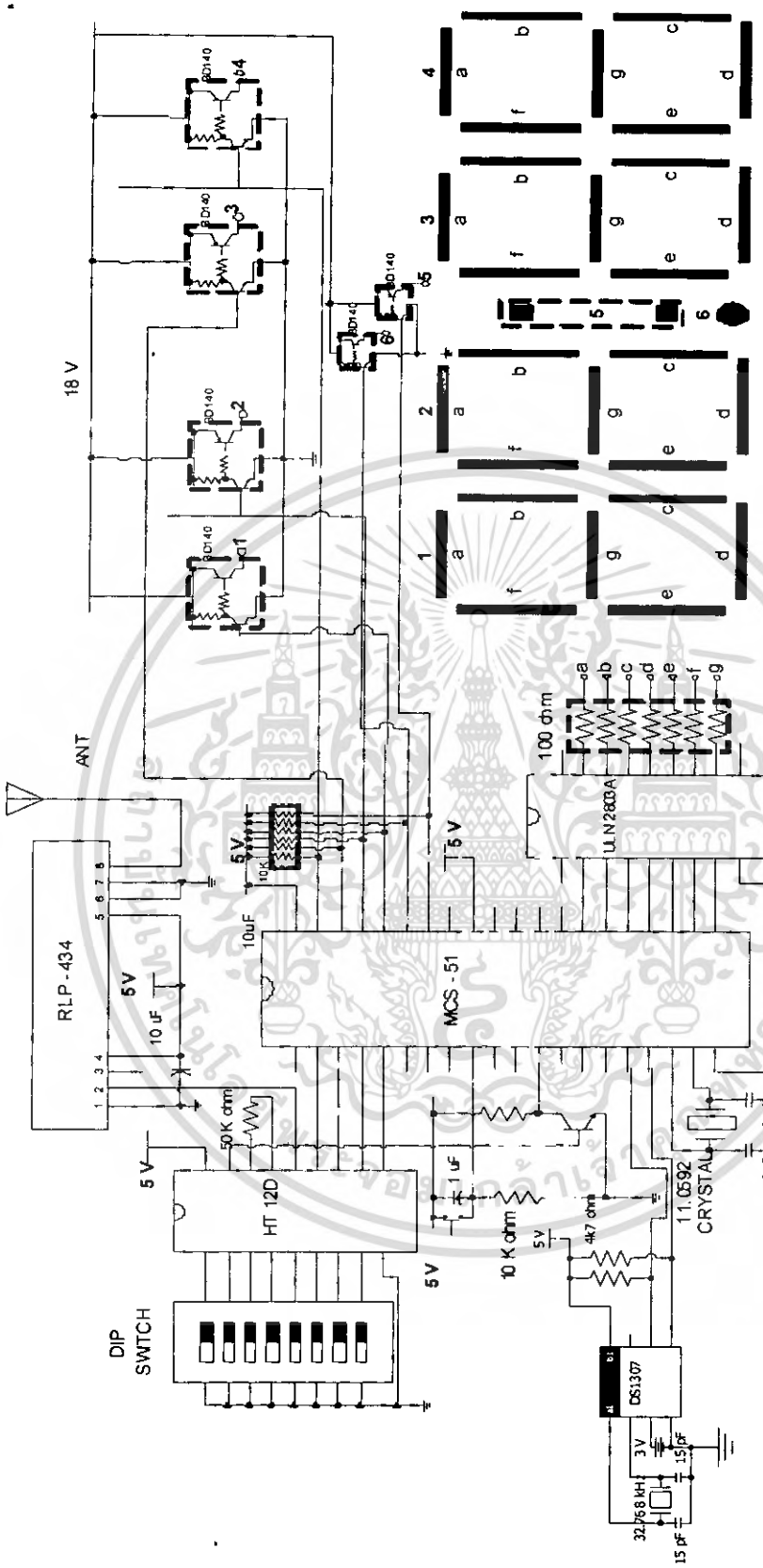


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chart การทำงานของภาครีโมทควบคุม

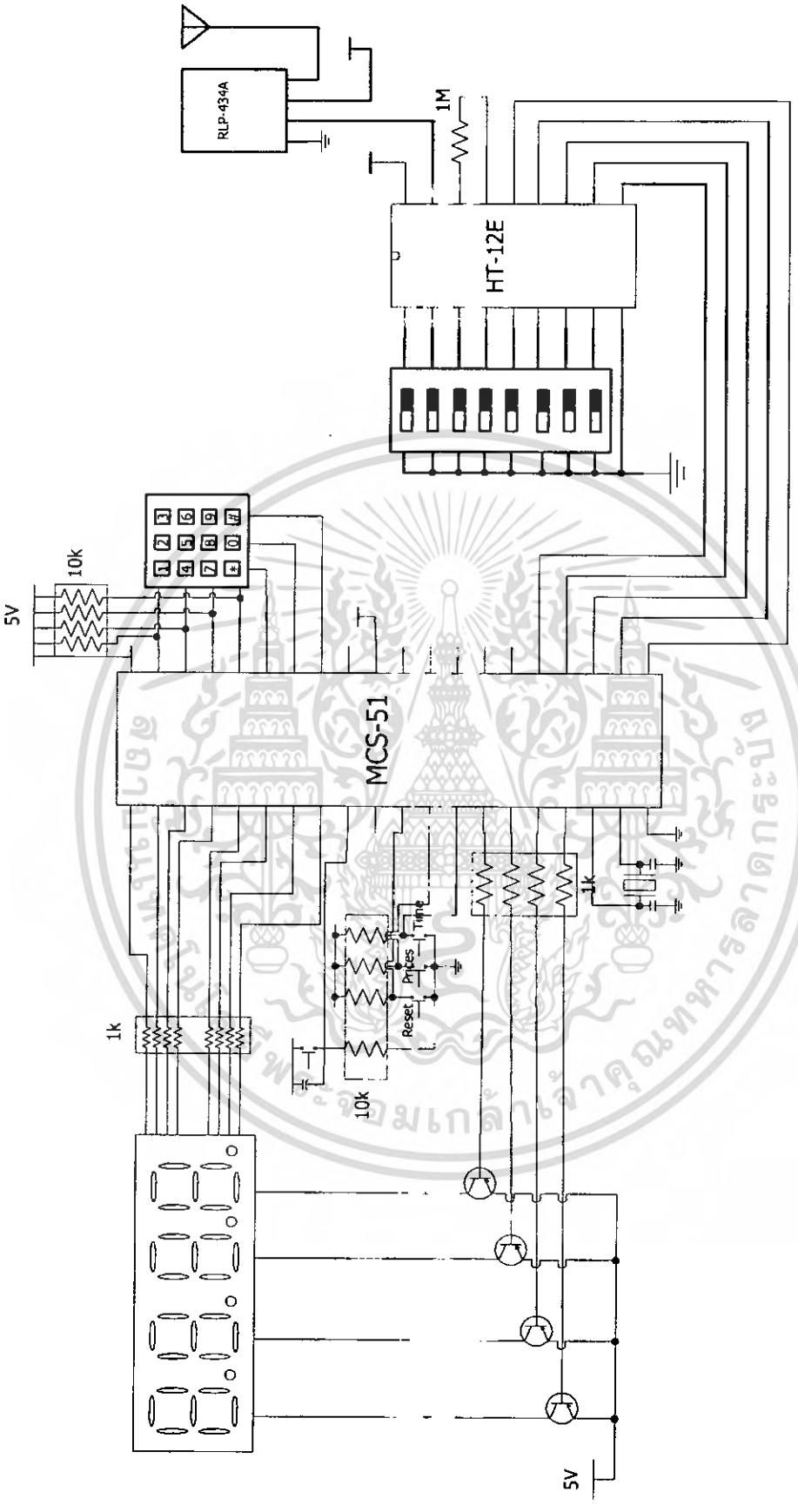


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปวงจรภาครับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปวงจรภาครีโมทควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Password

```
Private Sub CmdPassword_Click()
Dim Pwd As String
Dim Con As Integer
   Pwd = "S6010404"
   If TxtPassword.Text = Pwd Then
      Con = MsgBox("ยินดีต้อนรับเข้าสู่โปรแกรมแสดงราคาน้ำมัน", vbOKOnly, "You are welcome")
      If Con = vbOK Then
      Load FrmSelect
      FrmSelect.Show
      Unload FrmPassword
      End If
   Else
      MsgBox "กรุณาใส่ Password ให้ถูกต้องด้วยครับ", vbOKOnly, "Please type new password"
      TxtPassword.Text = ""
   End If
End Sub
```

ส่วนเลือกโหมดการทำงาน

```
Private Sub CmdSelectPrice_Click()
   Load FrmPrice
   FrmPrice.Show
   Unload FrmSelect
End Sub
Private Sub CmdSelectTime_Click()
   Load FrmTime
   FrmTime.Show
   Unload FrmSelect
End Sub
```

ส่วนเปลี่ยนราคาน้ำมัน

```
Private Sub CmdOK_Click()
   Dim resp As Boolean, i As Integer
   resp = CheckParameter(TxtBaht.Text, TxtStrang.Text)
   If resp = False Then
      MsgBox "กรุณาใส่ราคาน้ำมันให้ครบและถูกต้องด้วยครับ", vbOKOnly + vbExclamation, "ผิดพลาด"
      TxtBaht.Text = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TxtStrang.Text = ""
    Exit Sub
  End If
  If Len(TxtStrang.Text) = 1 Then
      TxtStrang.Text = TxtStrang.Text * 10
  End If
  aa = MsgBox("คุณต้องการเปลี่ยนราคาน้ำมันเป็น " & TxtBaht.Text & LblDot &
TxtStrang.Text & LblBaht, vbYesNoCancel, "New Price")
  Select Case aa
    Case vbYes
      If TxtBaht.Text < 10 Then
          If Len(TxtBaht.Text) = 1 Then
          MSComm1.Output = Chr(&H0)
          MSComm1.Output = TxtBaht.Text
          Else
          MSComm1.Output = Chr(&H0)
          MSComm1.Output = Right(TxtBaht.Text, 1)
          End If
      Else
        MSComm1.Output = TxtBaht.Text
      End If
      If TxtStrang.Text < 10 Then
        MSComm1.Output = Chr(&H0)
        MSComm1.Output = Right(TxtStrang.Text, 1)
      Else
        MSComm1.Output = TxtStrang.Text
      End If
    bb = MsgBox("คุณต้องการทำอย่างอื่นต่อไปหรือไม่?", vbYesNo + vbInformation,
            "Do you want to do other thing")
      If bb = vbYes Then
        Load FrmSelect
        FrmSelect.Show
        Unload FrmPrice
      Else
      End
      End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Case vbNo
          Load FrmPrice
        Case vbCancel
          End
        End Select
End Sub

Function CheckParameter(baht As String, strang As String) As Boolean
  If IsNumeric(baht) And IsNumeric(strang) Then
        CheckParameter = True
        Exit Function
  End If
        CheckParameter = False
End Function
Private Sub CmdReset_Click()
  MSComm1.Output = "<"
End Sub
Private Sub Form_Load()
  MSComm1.CommPort = 1
  MSComm1.Settings = "9600, n, 8, 1"
  MSComm1.PortOpen = True
End Sub
```

ส่วนเปลี่ยนเวลา

```
Private Sub CmdOK_Click()
  Dim Parameter As Boolean
  Dim Number As Boolean
  Parameter = CheckParameter(TxtHr.Text, TxtMin.Text)
  Number = CheckNumber(TxtHr.Text, TxtMin.Text, Parameter)
  If Parameter = False Or Number = False Then
    MsgBox "กรุณาเวลาให้ครบและถูกต้องด้วยครับ", vbOKOnly + vbExclamation, "ผิดพลาด"
    TxtHr.Text = ""
    TxtMin.Text = ""
    Exit Sub
  End If
    If TxtHr.Text = 24 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        TxtHr.Text = 0
      End If
      If TxtMin.Text = 60 Then
        TxtMin.Text = 0
      End If
    If Len(TxtMin.Text) = 1 Then
        TxtMin.Text = TxtMin.Text * 10
    End If
aa = MsgBox("คุณต้องการเปลี่ยนเวลาเป็น " & TxtHr.Text & LblColon &                    TxtMin.Text
& " น.", vbYesNoCancel, "New Price") Select Case aa
      Case vbYes
        MSComm1.Output = ">"
        If TxtHr.Text < 10 Then
            If Len(TxtHr.Text) = 1 Then
            MSComm1.Output = Chr(&H0)
            MSComm1.Output = TxtHr.Text
            Else
            MSComm1.Output = Chr(&H0)
            MSComm1.Output = Right(TxtHr.Text, 1)
            End If
        Else
          MSComm1.Output = TxtHr.Text
        End If
        If TxtMin.Text < 10 Then
          MSComm1.Output = Chr(&H0)
          MSComm1.Output = Right(TxtMin.Text, 1)
        Else
          MSComm1.Output = TxtMin.Text
        End If
         bb = MsgBox("คุณต้องการทำอย่างอื่นต่อไปหรือไม่?", vbYesNo + vbInformation,
                "Do you want to do other thing")
        If bb = vbYes Then
          Load FrmSelect
          FrmSelect.Show
          Unload FrmTime
        Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาคส่งสัญญาณ

```
#include<AT892051.h>
unsigned char save[5];
unsigned char change[5];
unsigned char transmit [] = {0x00,0x02,0x04,0x06,0x08,
0x0A,0x0C,0x0E,0x10,0x12,0x18,0x1C,0x14};
//************** Function Delay ****************
void delay (int tick)
{
        int i, j;
        for(i=0;i<tick;i++)
                for(j=0;j<250;j++);
}
//************** Function send data *************//
void send (unsigned char change)
{
        switch (change)
        {
                case 0x00       : P1 = transmit[0];
                                  break;
                case 0x01       : P1 = transmit[1];
                                  break;
                case 0x02       : P1 = transmit[2];
                                  break;
                case 0x03       : P1 = transmit[3];
                                   break;
                case 0x04       : P1 = transmit[4];
                                   break;
                case 0x05       : P1 = transmit[5];
                                   break;
                case 0x06       : P1 = transmit[6];
                                   break;
                case 0x07       : P1 = transmit[7];
                                   break;
                case 0x08       : P1 = transmit[8];
                                  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 0x09       :P1 = transmit[9];
                                break;
                case 0x10       :P1 = transmit[10];
                                break;
                case 0x11       : P1 = transmit[11];
                                break;
                case 0x12       : P1 = transmit[12];
                                break;
        }
        delay(500);
        P1 = 0x01;
        delay(500);
}//********** Function main *****************//
void main (void)
{
        int k;
        TMOD = 0x21;
        SCON = 0x50;
        TH1 = 0xFD;
        TL1 = 0xFD;
        RI = 0;
        TR1 = 1;
        P1 = 0x01;
        k = 0;
        while (1)
        {
        for (k=0;k<5;k++)
                {
                while (~RI);
                RI = 0;
                save[k] = SBUF;
                }
        change[1] = save[1] & 0x0F;change[2] = save[2] & 0x0F;
        change[3] = save[3] & 0x0F;
        change[4] = save[4] & 0x0F;
        switch(save[0])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                case 0x3C       :.send(0x10);       . Reset
                        break;
                case 0x3E       :send(0x11);                ./ Time
                        send(change[1]);
                        send(change[2]);
                        send(change[3]);
                        send(change[4]);
                        break;
                case 0x3A       : send(0x12);       :. Price
                        send(change[1]);
                        send(change[2]);
                        send(change[3]);
                        send(change[4]);
                        break;
                }
        }
}
```

โปรแกรมภาครับสัญญาณ

```
#include<reg51.h>
#include<i2c.h>
#define DS1307_ID 0xD0
unsigned char display [] = {0x3F,0x06,0x5B,0x4F,0x66,
                0x6D,0x7D,0x07,0x7F,0x6F,0x40};
sbit rs = P3^0;
sbit dot1 = P0^5;
sbit dot2 = P0^4;
sbit digital1 = P0^7;
sbit digital2 = P0^6;
sbit digital3 = P0^3;
sbit digital4 = P0^2;
unsigned char save[4];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char Time[4];
unsigned char sTime[2];
unsigned char price[4];
unsigned char min, hour, minnew;
unsigned int p = 0;
unsigned int T = 0;
unsigned int i,j,k,l,m,n;
/***************** Function Delay ******************/
void delay (int tick)
{
        int x,y;
        for (x=0;x<tick;x++)
                for (y=0;y<250;y++);
}

/*** Function write hour:min on chip DS1307 ***/
void DS1307_wrtime(unsigned char hh,
                        unsigned char mm)
{
        i2c_start();
        i2c_wrdata(DS1307_ID);
        i2c_wrdata(0x01);
        i2c_wrdata(mm);
        i2c_wrdata(hh);
        i2c_stop();
}
/**** Function write price on chip DS1307 ****/
void DS1307_wrprice(unsigned char b1,
          unsigned char b2, unsigned char s1,
                        unsigned char s2)
{
        i2c_start();
        i2c_wrdata(DS1307_ID);
        i2c_wrdata(0x0A);
        i2c_wrdata(b1);
        i2c_wrdata(b2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        i2c_wrdata(s1);
        i2c_wrdata(s2);
        i2c_stop();
}

/******** Function read data from DS1307 ********/
unsigned char DS1307_rd(unsigned char addr)
{
        unsigned char ret;
        i2c_start();
        i2c_wrdata(DS1307_ID);
        i2c_wrdata(addr);
        i2c_start();
        i2c_wrdata(DS1307_ID+1);
        ret = i2c_rddata();
        i2c_stop();
        return (ret);
}
/******** Function change number from P1 ********/
void change_p1 ()
{
        if (k >= 4)
                k = 0;
        switch(P1)
        {
                case 0xE1 :     save[k] = 0x00;
                                k++;
                                break;
                case 0xE3 :     save[k] = 0x01;
                                k++;
                                break;
                case 0xE5 :     save[k] = 0x02;
                                k++;
                                break;
                case 0xE7 :     save[k] = 0x03;
                                k++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
            case 0xE9 :     save[k] = 0x04;
                            k++;
                            break;
            case 0xEB :     save[k] = 0x05;
                            k++;
                            break;
            case 0xED :     save[k] = 0x06;
                            k++;
                            break;
            case 0xEF :     save[k] = 0x07;
                            k++;
                            break;
            case 0xF1 :     save[k] = 0x08;
                            k++;
                            break;
            case 0xF3 :     save[k] = 0x09;
                            k++;
                            break;
        }
}
/*************** Function save time ******************/
void save_time()
{
        T++;
        change_p1();
        if (T >= 5)
        {
                T = 0;
                Time[0] = save[0];
                sTime[0] = save[0] << 4;
                Time[1] = save[1];
                Time[2] = save[2];
                sTime[1] = save[2] << 4;
                Time[3] = save[3];
                DS1307_wrtime(sTime[0] | Time[1],
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    sTime[1] : Time[3]);
        }
}

/************ Function save price **********.
void save_price()
{
        p++;
        change_p1();
        if (p >= 5)
        {
        p = 0;
        DS1307 wrprice(save[0], save[1],
                        save[2], save[3]);
        price[0] = save[0];
        price[1] = save[1];
        price[2] = save[2];
        price[3] = save[3];
        }
}
/************ Function Reset ************/
void reset()
{
        rs = 0;
}

/************** Function interrupt ********************/
void start() interrupt 0
{
        digital1 = 0;
        digital2 = 0;
        digital3 = 0;
        digital4 = 0;
        if (T > 0)
                save_time();
        if (p > 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                save_price();
        switch(P1)
        {
                case 0xFD :     T++;
                        break;
                case 0xF5 :     p++;
                        break;
                case 0xF9 :     reset();
                        break;
        }
        delay(800);
}

/********** Function show time ************/
void show_time ()
{
        for (i=0;i<5;i++)
        {
        dot1 = 1;
        dot2 = 0;
        for (j=0;j<100;j++)
        {
                digital1 = 1;
                digital2 = 0;
                digital3 = 0;
                digital4 = 0;
                P2 = display[Time[0]];
                delay(1);

                digital1 = 0;
                digital2 = 1;
                digital3 = 0;
                digital4 = 0;
                P2 = display[Time[1]];
                delay(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                digital1 = 0;
                digital2 = 0;
                digital3 = 1;
                digital4 = 0;
                P2 = display[Time[2]];
                delay(1);

                digital1 = 0;
                digital2 = 0;
                digital3 = 0;
                digital4 = 1;
                P2 = display[Time[3]];
                delay(1);
                }

                digital1 = 0;
                digital2 = 0;
                digital3 = 0;
                digital4 = 0;
                dot1 = 0;
                dot2 = 0;
                delay(200);
        }
}
/************ Function show price ***************/
void show_price ()
{
        for (n=0;n<1000;n++)
        {
                digital1 = 1;
                digital2 = 0;
                digital3 = 0;
                digital4 = 0;
                P2 = display[price[0]];
                delay(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            digital1 = 0;
            digital2 = 1;
            digital3 = 0;
            digital4 = 0;
            P2 = display[price[1]];
            delay(1);

            dot1 = 0;
            dot2 = 1;

            digital1 = 0;
            digital2 = 0;
            digital3 = 1;
            digital4 = 0;
            P2 = display[price[2]];
            delay(1);

            digital1 = 0;
            digital2 = 0;
            digital3 = 0;
            digital4 = 1;
            P2 = display[price[3]];
            delay(1);
        }
}
/************* Function Main *****************/
void main (void)
{
        rs = 1;
        i=j=k=l=m=n = 0;
        P1 = 0xFF;                  // set input port
        EA = 1;
        EX0 = 1;
        min = DS1307_rd(0x01);
        hour = DS1307_rd(0x02);
        price[0] = DS1307_rd(0x0A);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        price[1] = DS1307_rd(0x0B);
        price[2] = DS1307_rd(0x0C);
        price[3] = DS1307_rd(0x0D);
        i2c_start();
        // set sec = 00 if no old data time or when wake up
        i2c_wrdata(DS1307_ID);
        i2c_wrdata(0x00);          // address
        i2c_wrdata(0x00);          // data sec
        i2c_stop();

        while (1)
        {
                show_price();
                minnew = DS1307_rd(0x01);
                if (minnew - min >= 1)
                {
                hour = DS1307_rd(0x02);
                Time[0] = (hour & 0xF0) >> 4;
                Time[1] = hour & 0x0F;
                Time[2] = (minnew & 0xF0) >> 4;
                Time[3] = minnew & 0x0F;
                min = minnew;
                show_time();
                }
        }
}
```

โปรแกรมภาครีโมทควบคุม

```
#include<reg51.h>
#include<scankey4x3.h>
sbit digital1 = P3^7;
sbit digital2 = P3^6;
sbit digital3 = P3^5;
sbit digital4 = P3^4;
sbit reset = P3^1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char display[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92
,0x82,0xF8,0x80,0x90,0x8C,0x87,0xBF,0x86};
unsigned char displaydot[] = {0x40,0x79,0x24,0x30,0x19,
0x12,0x02,0x78,0x00,0x10,0x0C,0x07,0x3F,0x06};
unsigned char transmit[] = {0x00,0x02,0x04,0x06,0x08,0x0A
,0x0C,0x0E,0x10,0x12,0x14,0x1C,0x18};
unsigned char save[4];
/************** Function Delay ******************
void delay (int tick)
{
        int i,j;
        for (i=0;i<tick;i++)
                for (j=0;j<250;j++);
}/***************** Function Send ****************;
void send (unsigned char tran)
{
        switch (tran)
        {
                case 0x00       : P2 = transmit[0];
                                  break;
                case 0x01       : P2 = transmit[1];
                                  break;
                case 0x02       : P2 = transmit[2];
                                  break;
                case 0x03       : P2 = transmit[3];
                                  break;
                case 0x04       : P2 = transmit[4];
                                  break;
                case 0x05       : P2 = transmit[5];
                                  break;
                case 0x06       : P2 = transmit[6];
                                  break;
                case 0x07       :P2 = transmit[7];
                                  break;
                case 0x08       : P2 = transmit[8];
                                  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 0x09       : P2 = transmit[9];
                                  break;
                case 0x10 :     P2 = transmit[10];
                                  break;
                case 0x11 :     P2 = transmit[11];
                                 break;
                case 0x12 :     P2 = transmit[12];
                                  break;
                default :  P2 = 0xFF;
                                 break;
        }
                delay(500);
        P2 = 0x01;
        delay(500);
}
/**********Function External Interrupt 0 **********/
void intprice (void) interrupt 0
{
        int k = 0;
        int a = 1;
        unsigned char key = 0xFF;
        save[0]=save[1]=save[2]=save[3]= 10;
        while(a == 1)
        {
                digital1 = 0;
                digital2 = 1;
                digital3 = 1;
                digital4 = 1;
                P1 = display[save[0]];
                delay(1);

                digital1 = 1;
                digital2 = 0;
                digital3 = 1;
                digital4 = 1;
                P1 = displaydot[save[1]];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(1);

digital1 = 1;
digital2 = 1;
digital3 = 0;
digital4 = 1;
P1 = display[save[2]];
delay(1);

digital1 = 1;
digital2 = 1;
digital3 = 1;
digital4 = 0;
P1 = display[save[3]];
delay(1);
key = scankey();
if (key != 0xFF)
{
        switch(key)
        {
        case 0x10       :k = 0;
                        save[0]=save[1]=save[2]=save[3]= 10;
                        break;
        case 0x11       :if (save[0]==10 || save[1]==10 || save[2]==10 ||save[3]==10)
                        {
                                save[0]=save[1]=save[2]=save[3] = 13;
                                k = 0;
                        }
                        else
                        {
                                send(0x10);
                                digital1 = 0;
                                digital2 = 1;
                                digital3 = 1;
                                digital4 = 1;
                                P1 = display[save[0]];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        send(save[0]);
                                        digital1 = 1;
                                        digital2 = 0;
                                        digital3 = 1;
                                        digital4 = 1;
                                        P1   display[save[1]];
                                        send(save[1]);
                                        digital1   1;
                                        digital2   1;
                                        digital3   0;
                                        digital4   1;
                                        P1   display[save[2]];
                                        send(save[2]);
                                        digital1   1;
                                        digital2   1;
                                        digital3   1;
                                        digital4   0;
                                        P1   display[save[3]];
                                        send(save[3]);
                                        a = 0;
                                        }
                                        break;
                default   :save[k] = key;
                          k++;
                          if (k >= 4)
                                    k = 0;
                          break;
                }

        }
    }
}
/*******Function External Interrupt 1 ********/
void inttime (void) interrupt 2
{
        int m = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int b = 1;
unsigned char key = 0xFF;
save[0]=save[1]=save[2]=save[3]= 11;
while(b == 1)
{
        digital1 = 0;
        digital2 = 1;
        digital3 = 1;
        digital4 = 1;
        P1 = display[save[0]];
        delay(1);
        digital1 = 1;
        digital2 = 0;
        digital3 = 1;
        digital4 = 1;
        P1 = displaydot[save[1]];
        delay(1);

        digital1 = 1;
        digital2 = 1;
        digital3 = 0;
        digital4 = 1;
        P1 = display[save[2]];
        delay(1);

        digital1 = 1;
        digital2 = 1;
        digital3 = 1;
        digital4 = 0;
        P1 = display[save[3]];
        delay(1); key = scankey();
        if (key != 0xFF)
        {
                switch(key)
        {
                case 0x10       :       m = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                save[0] save[1]=save[2]=save[3]= 11;
                break;
case 0x11       :if (save[0]   11 | save[1]==11 || save[2]==11 || save[3]==11)
                {
                save[0] save[1]=save[2]=save[3] = 13;
                m   0;
                }
                else
                {
                send(0x11);
                digital1   0;
                digital2   1;
                digital3   1;
                digital4   1;
                P1 = display[save[0]];
                send(save[0]);
                digital1   1;
                digital2   0;
                digital3   1;
                digital4   1;
                P1   display[save[1]];
                send(save[1]);
                digital1   1;
                digital2   1;
                digital3   0;
                digital4   1;
                P1   display[save[2]];
                send(save[2]);
                digital1   1;
                digital2   1;
                digital3   1;
                digital4   0;
                P1   display[save[3]];
                send(save[3]);
                b   0;
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

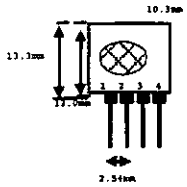
```
                                        break;
                        default         :save[m] = key;
                                        m++;
                                        if (m >= 4)
                                        m = 0;
                                        break;
                }

        }
}
}/************** Function Main **************/
void main (void)
{
        P0 = 0xFF;
        IT0 = 1;
        IT1 = 1;
        EA = 1;
        EX0 = 1;
        EX1 = 1;
        while(1)
        {
                digital1 = 0;
                digital2 = 1;
                digital3 = 1;
                digital4 = 1;
                P1 = display[12];
                delay(1);
                digital1 = 1;
                digital2 = 0;
                digital3 = 1;
                digital4 = 1;
                P1 = display[12];
                delay(1);
                digital1 = 1;
                digital2 = 1;
                digital3 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            digital4 = 1;
            P1 = display[12];
            delay(1);
            digital1 = 1;
            digital2 = 1;
            digital3 = 1;
            digital4 = 0;
            P1 = display[12];
            delay(1);
            if (reset == 0)
            send(0x12);
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Easy-Link Wireless



pin 1 : GND
pin 2 : Data In
pin 3 : Vcc
pin 4 : Antenna (RF output)

Frequency 315, 418 and 433.92 Mhz

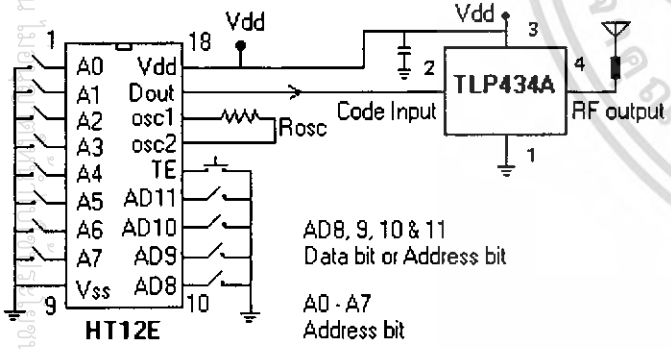
Modulation : ASK
Operation Voltage : 2 - 12 VDC

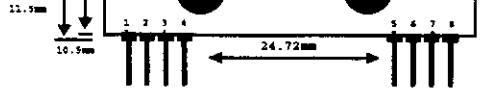
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		2.0	-	12.0	V
Icc 1	Peak Current (2V)		-	-	1.64	mA
Icc 2	Peak Current (12V)		-	-	19.4	mA
Vh	Input High Voltage	Idata= 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
Vl	Input Low Voltage	Idata= 0 uA (Low)	-	-	0.3	V
FO	Absolute Frequency	315Mhz module	314.8	315	315.2	MHz
PO	RF Output Power- 50ohm	Vcc = 9V-12V	-	16	-	dBm
		Vcc = 5V-6V	-	14	-	dBm
DR	Data Rate	External Encoding	512	4.8K	200K	bps

Notes : (Case Temperature = 25°C +- 2°C . Test Load Impedance = 50 ohm)

Application Circuit :

Typical Key-chain Transmitter using HT12E-18DIP, a Binary 12 bit Encoder from Holtek Semiconductor Inc.





pin 1 : Gnd
pin 2 : Digital Data Output
pin 3 : Linear Output /Test
pin 4 : Vcc
pin 5 : Vcc
pin 6 : Gnd
pin 7 : Gnd
pin 8 : Antenna

Frequency 315, 418 and 433.92 Mhz

Modulation : ASK
Supply Voltage : 3.3 - 6.0 VDC
Output : Digital & Linear

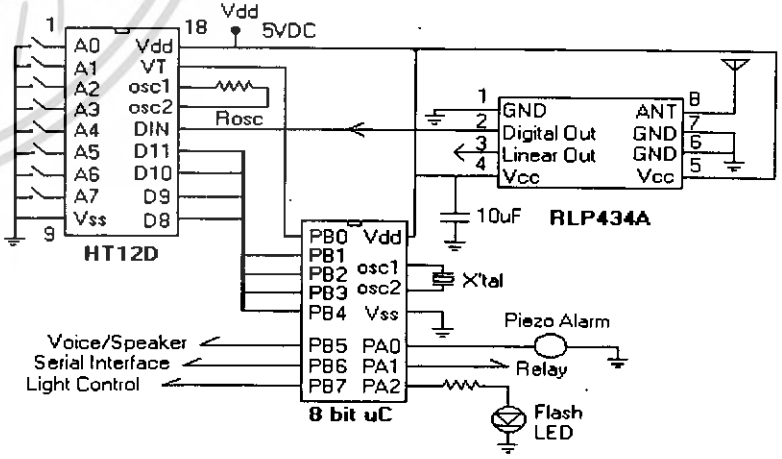
Symbol	Parameter	Conditions	Min	Typ	Max	
Vcc	Operating supply voltage		3.3	5.0V	6.0	V
Itot	Operating Current		-	4.5		mA
Vdata	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

Electrical Characteristics

Characteristics	SYM	Min	Typ	Max	Unit
Operation Radio Frequency	FC		315, 418 and 433.92		MHz
Sensitivity	Pref		-110		dBm
Channel Width			+-500		Khz
Noise Equivalent BW			4		Khz
Receiver Turn On Time			5		ms
Operation Temperature	Top	-20	-	80	C
Baseboard Data Rate			4.8		KHz

Application Circuit :

Typical RF Receiver using HT12D-18DIP, a Binary 12 bit Decoder with 8 bit uC HT48RXX from Holtek Semiconductor Inc.



Laipac Technology, Inc.
105 West Beaver Creek Rd. Unit 207 Richmond Hill Ontario L4B 1C6 Canada
Tel: (905)762-1228 Fax: (905)763-1737 e-mail: info@laipac.com

LAIPAC TECH

ıtures

- mpatible with MCS-51™ Products
- Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- lly Static Operation: 0 Hz to 24 MHz
- ree-Level Program Memory Lock
- 3 x 8-Bit Internal RAM
- Programmable I/O Lines
- o 16-Bit Timer/Counters
- Interrupt Sources
- ›grammable Serial Channel
- w Power Idle and Power Down Modes

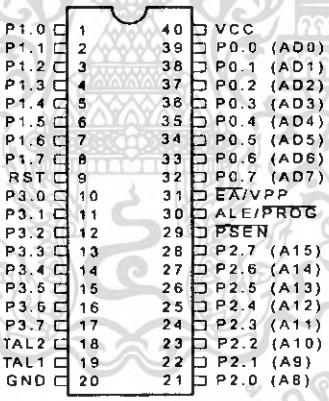
ɔcription

AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K s of Flash Programmable and Erasable Read Only Memory (PEROM). The ce is manufactured using Atmel's high density nonvolatile memory technology s compatible with the industry standard MCS-51™ instruction set and pinout. The hip Flash allows the program memory to be reprogrammed in-system or by a con- onal nonvolatile memory programmer. By combining a versatile 8-bit CPU with n on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which des a highly flexible and cost effective solution to many embedded control appli- ns.

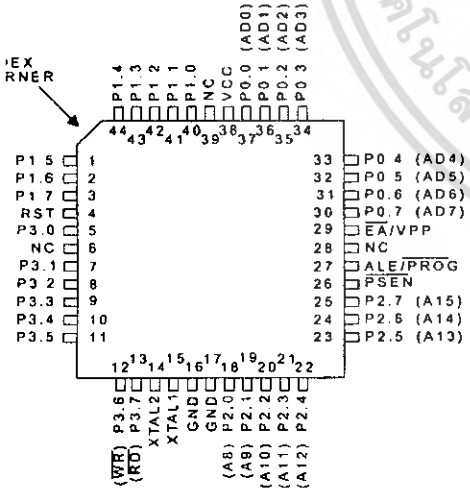
(continued)

Configurations

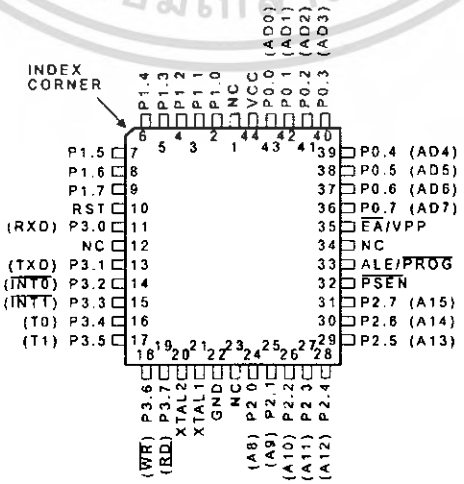
PDIP



PQFP/TQFP



PLCC



8-Bit Microcontroller with 4K Bytes Flash

AT89C51

0265F-A–12/97

ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

solute Maximum Ratings*

erating Temperature	-55°C to +125°C
rage Temperature	-65°C to +150°C
age on Any Pin Respect to Ground	-1.0V to +7.0V
timum Operating Voltage	6.6V
Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Characteristics

-40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

bol	Parameter	Condition	Min	Max	Units
	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	µA
	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
r	Reset Pulldown Resistor		50	300	KΩ
	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode[2]	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port: Port 0: 26 mA
   Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2V.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HOLTEK

HT12A/HT12E
2^{12} Series of Encoders

Features

- Operating voltage
  - 2.4V~5V for the HT12A
  - 2.4V~12V for the HT12E
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1μA (typ.) at V_{DD}=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission word
  - Four words for the HT12E
  - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- HT12A/E: 18-pin DIP/20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12-N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a $\overline{TE}$ trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 2^{12} series of encoders. The HT12A additionally provides a 38kHz carrier for infrared systems.

Selection Table

Function / Part No.	Address No.	Address/ Data No.	Data No.	Oscillator	Trigger	Package	Carrier Output	Negative Polarity
HT12A	8	0	4	455kHz resonator	D8~D11	18 DIP 20 SOP	38kHz	No
HT12E	8	4	0	RC oscillator	$\overline{TE}$	18 DIP 20 SOP	No	No

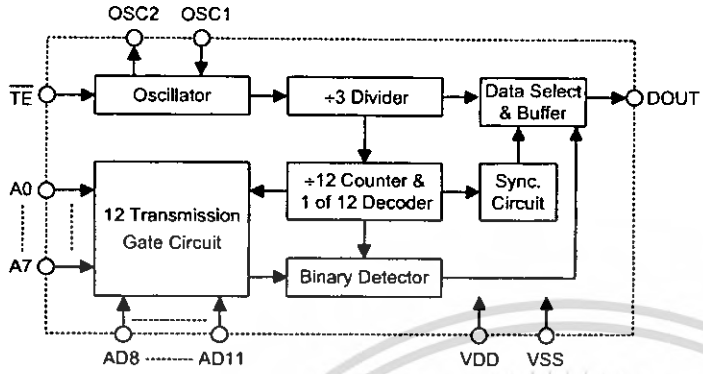
Note: Address/Data represents pins that can be address or data according to the decoder requirement.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram

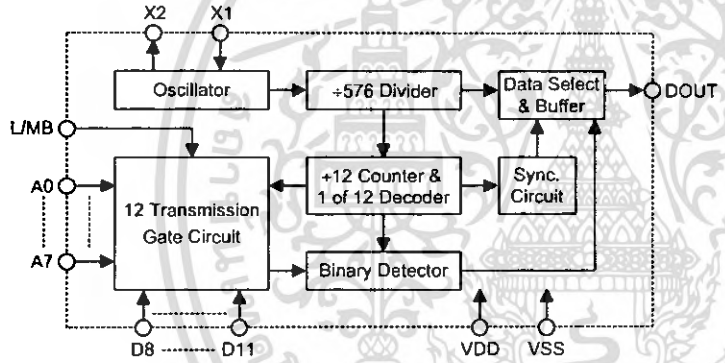
$\overline{TE}$ trigger

HT12E



DATA trigger

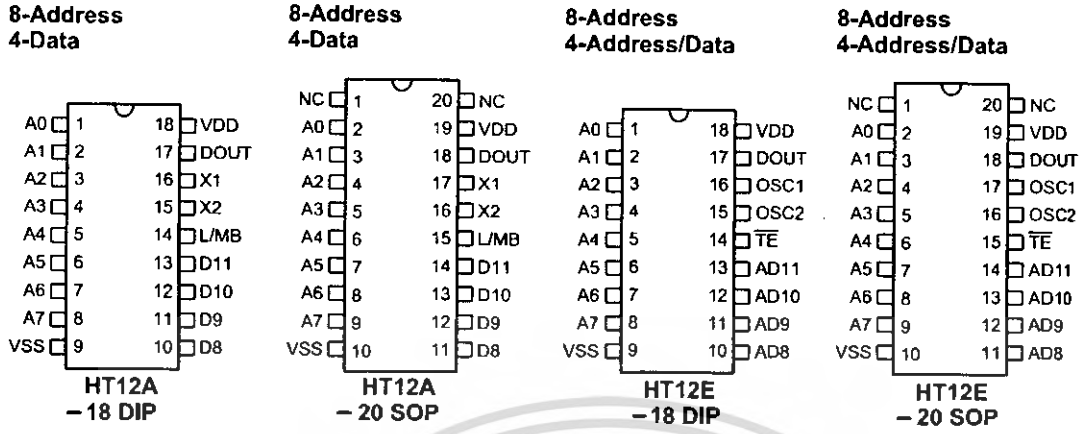
HT12A



Note: The address data pins are available in various combinations (refer to the address/data table).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Assignment



Pin Description

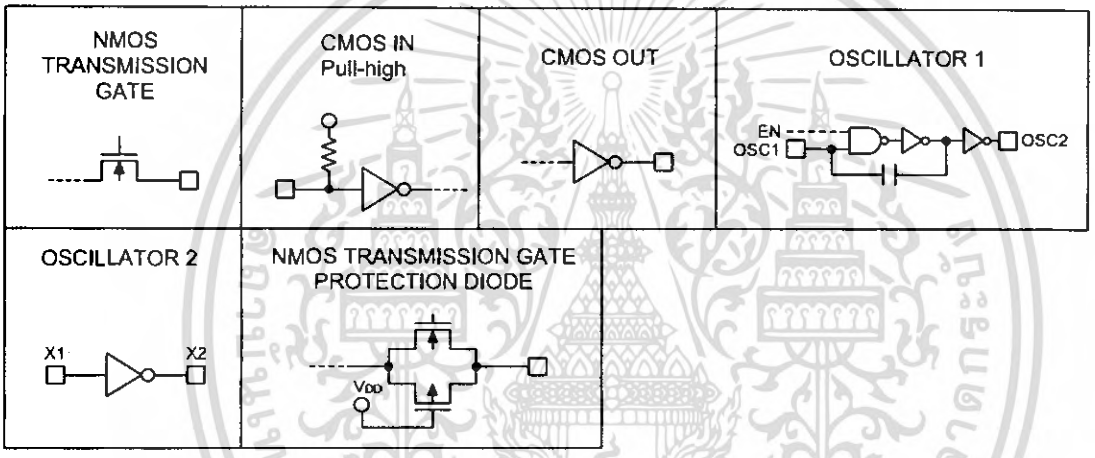
Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A) NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/MB	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Name	I/O	Internal Connection	Description
$\overline{TE}$	I	CMOS IN Pull-high	Transmission enable, active low (see Note)
OSC1	I	OSCILLATOR 1	Oscillator input pin
OSC2	O	OSCILLATOR 1	Oscillator output pin
X1	I	OSCILLATOR 2	455kHz resonator oscillator input
X2	O	OSCILLATOR 2	455kHz resonator oscillator output
VSS	I	—	Negative power supply, grounds
VDD	I	—	Positive power supply

Note: D8~D11 are all data input and transmission enable pins of the HT12A.
$\overline{TE}$ is a transmission enable pin of the HT12E.

Approximate internal connections



Absolute Maximum Ratings

Supply Voltage (HT12A) .....–0.3V to 5.5V
Supply Voltage (HT12E) .....–0.3V to 13V
Input Voltage.....V_{SS}–0.3 to V_{DD}+0.3V
Storage Temperature.....–50°C to 125°C
Operating Temperature.....–20°C to 75°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics

HT12A

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	3	5	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		5V		—	0.1	1	μA
I_{DD}	Operating Current	3V	No load f_{OSC}=455kHz	—	200	400	μA
		5V		—	400	800	μA
I_{DOUT}	Output Drive Current	5V	V_{OH}=0.9V_{DD} (Source)	−1	−1.6	—	mA
			V_{OL}=0.1V_{DD} (Sink)	2	3.2	—	mA
V_{IH}	"H" Input Voltage	—	—	0.8V_{DD}	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	0.2V_{DD}	V
R_{DATA}	D8~D11 Pull-high Resistance	5V	V_{DATA}=0V	—	150	300	kΩ

HT12E

Ta=25°C

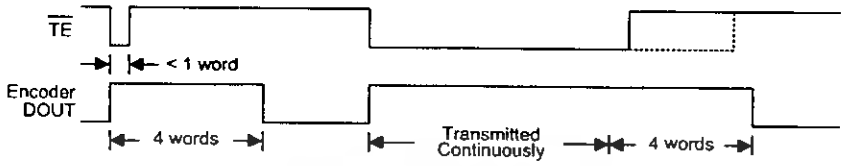
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	5	12	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I_{DD}	Operating Current	3V	No load f_{OSC}=3kHz	—	40	80	μA
		12V		—	150	300	μA
I_{DOUT}	Output Drive Current	5V	V_{OH}=0.9V_{DD} (Source)	−1	−1.6	—	mA
			V_{OL}=0.1V_{DD} (Sink)	1	1.6	—	mA
V_{IH}	"H" Input Voltage	—	—	0.8V_{DD}	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	0.2V_{DD}	V
f_{OSC}	Oscillator Frequency	5V	R_{OSC}=1.1MΩ	—	3	—	kHz
$R_{\overline{TE}}$	$\overline{TE}$ Pull-high Resistance	5V	$V_{\overline{TE}}$=0V	—	1.5	3	MΩ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

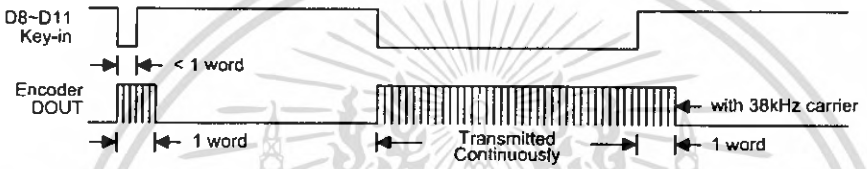
Functional Description

Operation

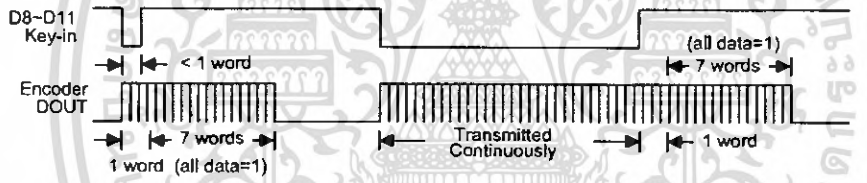
The 2^{12} series of encoders begin a 4-word transmission cycle upon receipt of a transmission enable ($\overline{TE}$ for the HT12E or D8~D11 for the HT12A, active low). This cycle will repeat itself as long as the transmission enable ($\overline{TE}$ or D8~D11) is held low. Once the transmission enable returns high the encoder output completes its final cycle and then stops as shown below.



Transmission timing for the HT12E



Transmission timing for the HT12A (L/MB=Floating or VDD)



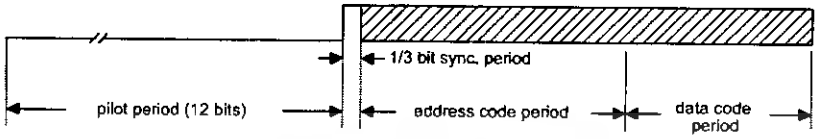
Transmission timing for the HT12A (L/MB=VSS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Information word

If L/MB=1 the device is in the latch mode (for use with the latch type of data decoders). When the transmission enable is removed during a transmission, the DOUT pin outputs a complete word and then stops. On the other hand, if L/MB=0 the device is in the momentary mode (for use with the momentary type of data decoders). When the transmission enable is removed during a transmission, the DOUT outputs a complete word and then adds 7 words all with the "1" data code.

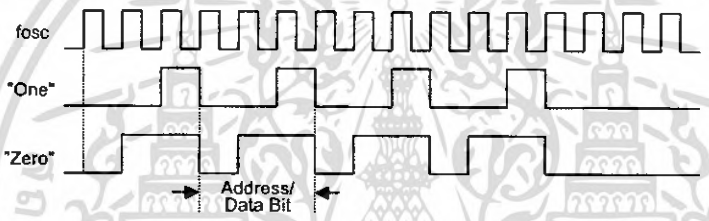
An information word consists of 4 periods as illustrated below.



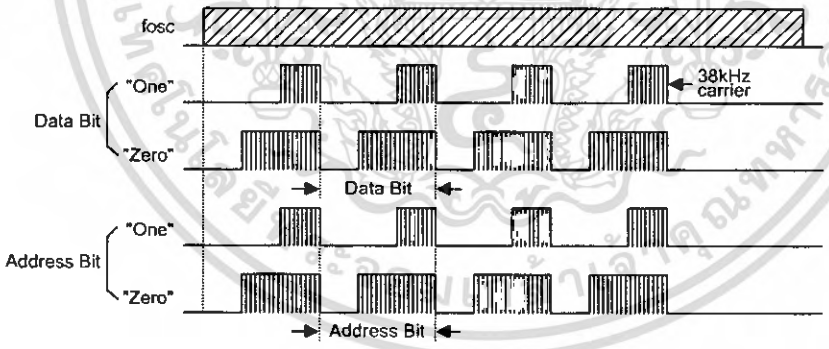
Composition of information

Address/data waveform

Each programmable address/data pin can be externally set to one of the following two logic states as shown below.



Address/Data bit waveform for the HT12E



Address/Data bit waveform for the HT12A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The address/data bits of the HT12A are transmitted with a 38kHz carrier for infrared remote controller flexibility.

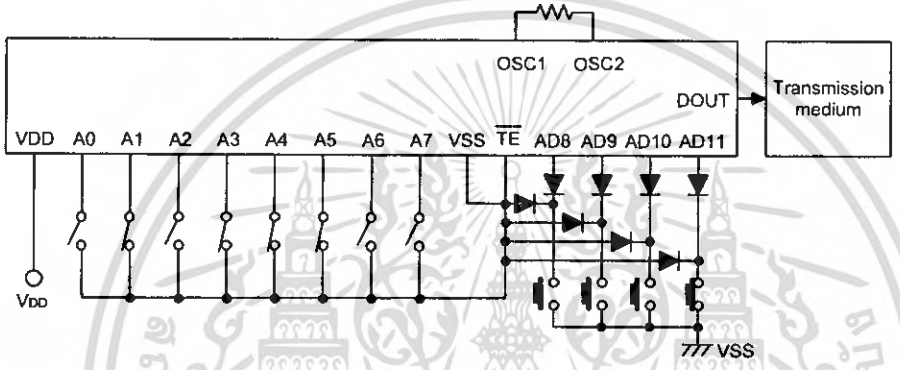
Address/data programming (preset)

The status of each address/data pin can be individually pre-set to logic "high" or "low". If a transmission-enable signal is applied, the encoder scans and transmits the status of the 12 bits of address/data serially in the order A0 to AD11 for the HT12E encoder and A0 to D11 for the HT12A encoder.

During information transmission these bits are transmitted with a preceding synchronization bit. If the trigger signal is not applied, the chip enters the standby mode and consumes a reduced current of less than 1μA for a supply voltage of 5V.

Usual applications preset the address pins with individual security codes using DIP switches or PCB wiring, while the data is selected by push buttons or electronic switches.

The following figure shows an application using the HT12E:



The transmitted information is as shown:

Pilot & Sync.	A0	A1	A2	A3	A4	A5	A6	A7	AD8	AD9	AD10	AD11
	1	0	1	0	0	0	1	1	1	1	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address/Data sequence

The following provides the address/data sequence table for various models of the 2^{12} series of encoders. The correct device should be selected according to the individual address and data requirements.

Part No.	Address/Data Bits											
	0	1	2	3	4	5	6	7	8	9	10	11
HT12A	A0	A1	A2	A3	A4	A5	A6	A7	D8	D9	D10	D11
HT12E	A0	A1	A2	A3	A4	A5	A6	A7	AD8	AD9	AD10	AD11

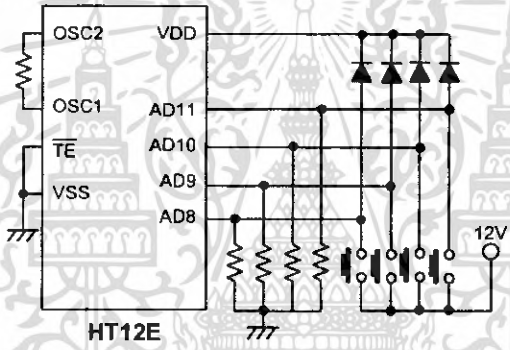
Transmission enable

For the HT12E encoders, transmission is enabled by applying a low signal to the $\overline{TE}$ pin. For the HT12A encoders, transmission is enabled by applying a low signal to one of the data pins D8~D11.

Two erroneous HT12E application circuits

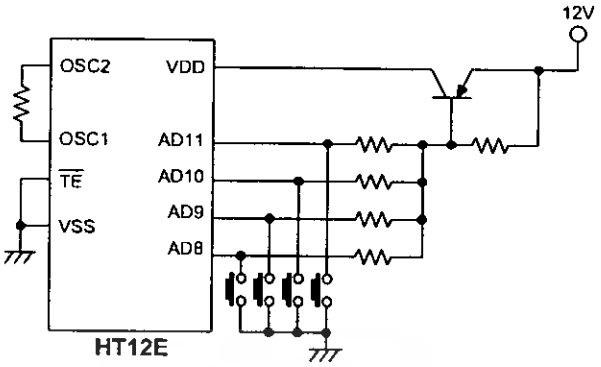
The HT12E must follow closely the application circuits provided by Holtek (see the "Application circuits").

- Error: AD8~AD11 pins input voltage > V_{DD}+0.3V



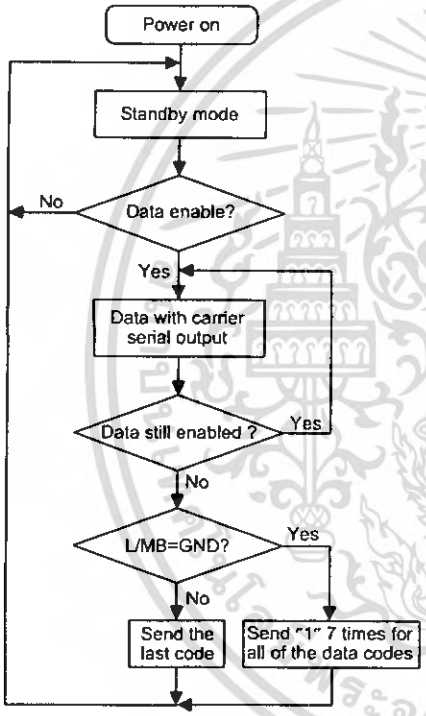
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• Error: The IC's power source is activated by pins AD8~AD11

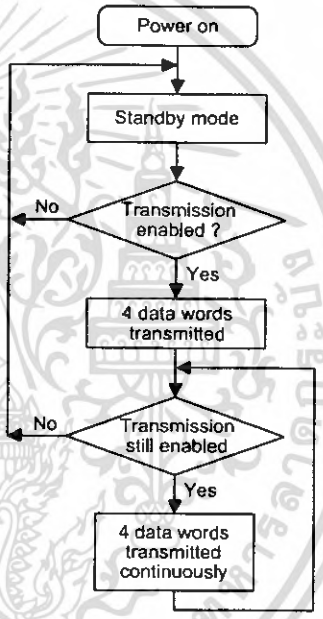


Flowchart

• HT12A



• HT12E



Note: D8~D11 are transmission enables of the HT12A.
$\overline{TE}$ is the transmission enable of the HT12E.

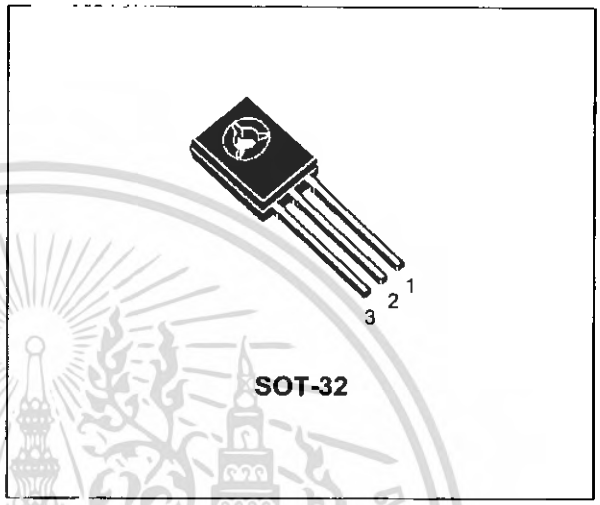
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGS-THOMSON MICROELECTRONICS

BD136
BD138/BD140

PNP SILICON TRANSISTORS

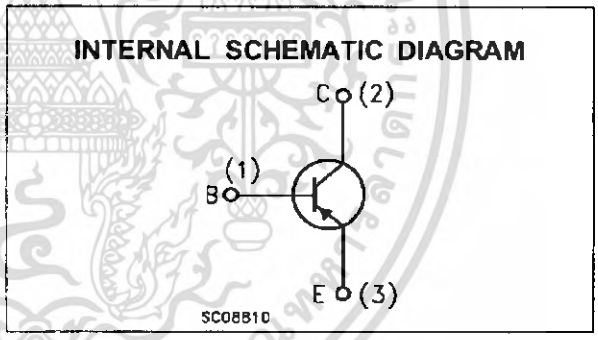
- SGS-THOMSON PREFERRED SALESTYPES
- PNP TRANSISTOR



DESCRIPTION

The BD136, BD138 and BD140 are silicon epitaxial planar PNP transistors in Jedec SOT-32 plastic package, designed for audio amplifiers and drivers utilizing complementary or quasi compementary circuits.

The complementary NPN types are the BD135 BD137 and BD139.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value			Unit
		BD136	BD138	BD140	
V_{CBO}	Collector-Base Voltage (I_E = 0)	-45	-60	-80	V
V_{CEO}	Collector-Emitter Voltage (I_B = 0)	-45	-60	-80	V
V_{EBO}	Emitter-Base Voltage (I_C = 0)	-5			V
I_C	Collector Current	-1.5			A
I_{CM}	Collector Peak Current	-3			A
I_B	Base Current	-0.5			A
P_{tot}	Total Dissipation at $T_c \leq 25$ °C	12.5			W
P_{tot}	Total Dissipation at $T_{amb} \leq 25$ °C	1.25			W
T_{stg}	Storage Temperature	-65 to 150			°C
T_j	Max. Operating Junction Temperature	150			°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HOLTEK

2^{12} Series of Decoders

Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Pair with Holtek's 2^{12} series of encoders
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination
  - HT12D: 8 address bits and 4 data bits
  - HT12F: 12 address bits only
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2^{12} series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen.

The decoders receive serial addresses and data from a programmed 2^{12} series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

The 2^{12} series of decoders are capable of decoding informations that consist of N bits of address and 12–N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

Selection Table

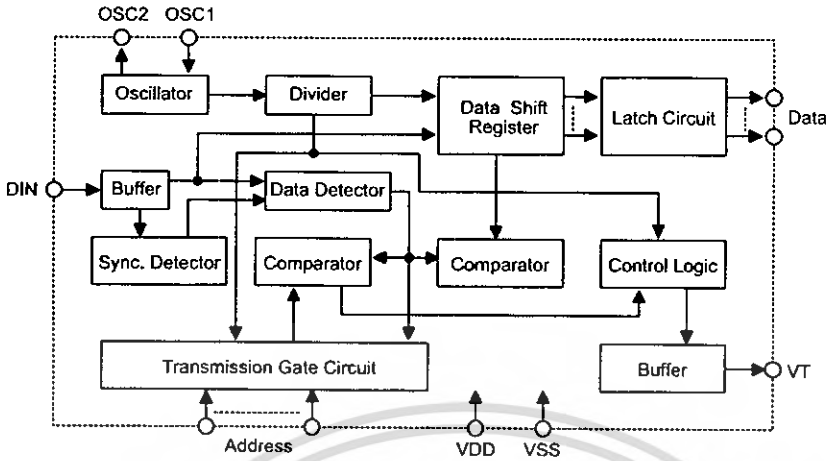
Function / Part No.	Address No.	Data No.	Data Type	VT	Oscillator	Trigger	Package
HT12D	8	4	L	√	RC oscillator	DIN active "Hi"	18 DIP/20 SOP
HT12F	12	0	—	√	RC oscillator	DIN active "Hi"	18 DIP/20 SOP

Notes: Data type: L stands for latch type data output.

VT can be used as a momentary data output.

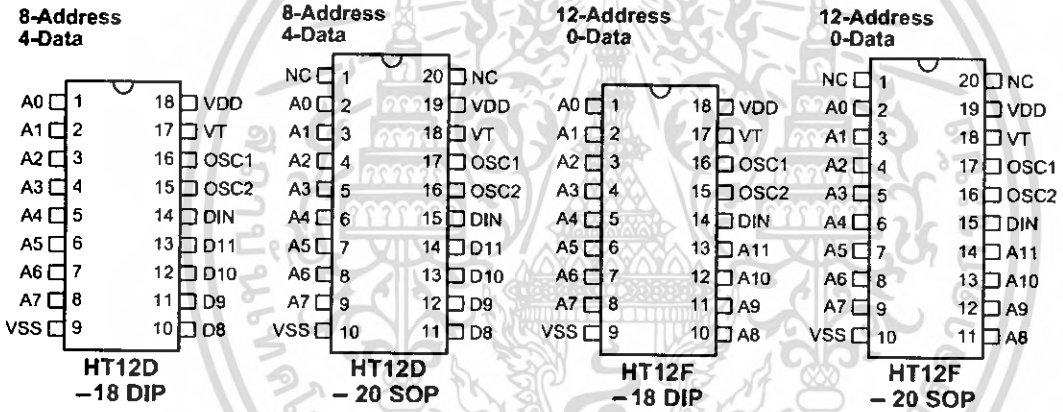
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



Note: The address/data pins are available in various combinations (see the address/data table).

Pin Assignment

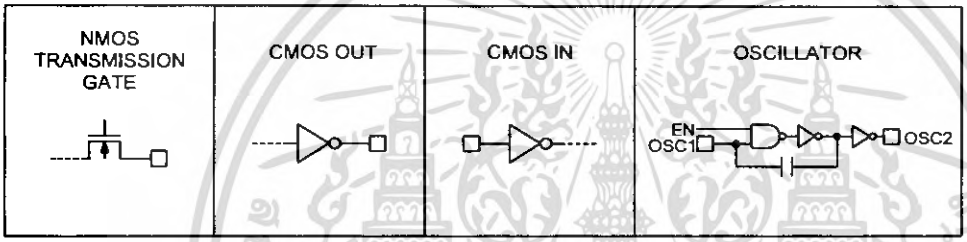


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A11	I	NMOS TRANSMISSION GATE	Input pins for address A0~A11 setting They can be externally set to VDD or VSS.
D8~D11	O	CMOS OUT	Output data pins
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	OSCILLATOR	Oscillator input pin
OSC2	O	OSCILLATOR	Oscillator output pin
VSS	I	—	Negative power supply (GND)
VDD	I	—	Positive power supply

Approximate internal connection circuits



Absolute Maximum Ratings

Supply Voltage.....–0.3V to 13V Storage Temperature.....–50°C to 125°C

Input Voltage.....V_{SS}–0.3 to V_{DD}+0.3V Operating Temperature .....–20°C to 75°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	5	12	V
I_{STB}	Standby Current	5V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I_{DD}	Operating Current	5V	No load f_{OSC}=150kHz	—	200	400	μA
I_O	Data Output Source Current (D8~D11)	5V	V_{OH}=4.5V	-1	-1.6	—	mA
	Data Output Sink Current (D8~D11)	5V	V_{OL}=0.5V	1	1.6	—	mA
I_{VT}	VT Output Source Current	5V	V_{OH}=4.5V	-1	-1.6	—	mA
	VT Output Sink Current		V_{OL}=0.5V	1	1.6	—	mA
V_{IH}	"H" Input Voltage	5V	—	3.5	—	5	V
V_{IL}	"L" Input Voltage	5V	—	0	—	1	V
f_{OSC}	Oscillator Frequency	5V	R_{OSC}=51kΩ	—	150	—	kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

Operation

The 2^{12} series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 2^{12} series of encoders.

The decoders receive data that are transmitted by an encoder and interpret the first N bits of code period as addresses and the last 12–N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoder's local address, the 12–N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received.

The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

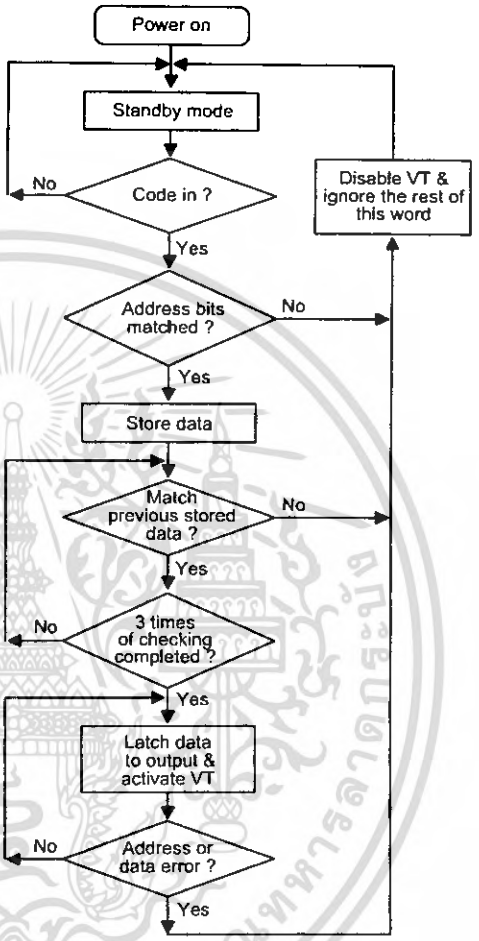
Output type

Of the 2^{12} series of decoders, the HT12F has no data output pin but its VT pin can be used as a momentary data output. The HT12D, on the other hand, provides 4 latch type data pins whose data remain unchanged until new data are received.

Part No.	Data Pins	Address Pins	Output Type	Operating Voltage
HT12D	4	8	Latch	2.4V~12V
HT12F	0	12	—	2.4V~12V

Flowchart

The oscillator is disabled in the standby state and activated when a logic "high" signal applies to the DIN pin. That is to say, the DIN should be kept low if there is no signal input.

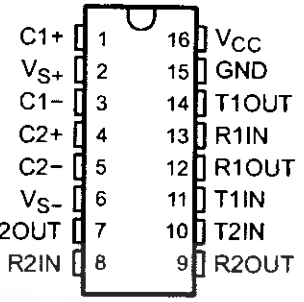


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0-μF Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1-μF Charge-Pump Capacitors is Available With the MAX202
- Applications
  - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
	SOP (NS)	Reel of 2000	MAX232NSR	MAX232
−40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2004, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

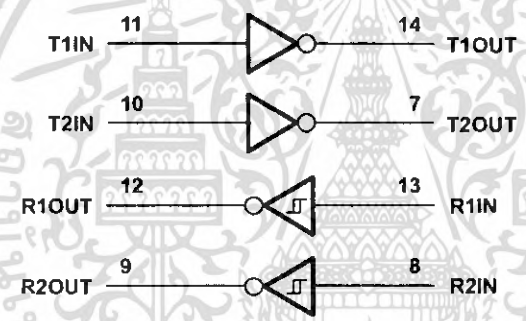
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

ɟic diagram (positive logic)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, V_{CC} (see Note 1) . . . . . −0.3 V to 6 V
Positive output supply voltage range, V_{S+} . . . . . V_{CC} − 0.3 V to 15 V
Negative output supply voltage range, V_{S-} . . . . . −0.3 V to −15 V
Input voltage range, V_I: Driver . . . . . −0.3 V to V_{CC} + 0.3 V
Receiver . . . . . ±30 V
Output voltage range, V_O: T1OUT, T2OUT . . . . . V_{S-} − 0.3 V to V_{S+} + 0.3 V
R1OUT, R2OUT . . . . . −0.3 V to V_{CC} + 0.3 V
Short-circuit duration: T1OUT, T2OUT . . . . . Unlimited
Package thermal impedance, θ_{JA} (see Notes 2 and 3): D package . . . . . 73°C/W
DW package . . . . . 57°C/W
N package . . . . . 67°C/W
NS package . . . . . 64°C/W
Operating virtual junction temperature, T_J . . . . . 150°C
Storage temperature range, T_{stg} . . . . . −65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to network GND.
2. Maximum power dissipation is a function of $T_J(max)$, θ_{JA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
3. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN,T2IN)		2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)				0.8	V
R1IN, R2IN	Receiver input voltage				±30	V
T_A	Operating free-air temperature	MAX232	0		70	°C
		MAX232I	−40		85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER		TEST CONDITIONS	MIN	TYP‡	MAX	UNIT
I_{CC}	Supply current	V_{CC} = 5.5 V, All outputs open, T_A = 25°C		8	10	mA

‡ All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DRIVER SECTION

ctrical characteristics over recommended ranges of supply voltage and operating free-air nperature range (see Note 4)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
OH	High-level output voltage	T1OUT, T2OUT	R_L = 3 kΩ to GND	5	7		V
OL	Low-level output voltage‡	T1OUT, T2OUT	R_L = 3 kΩ to GND		−7	−5	V
	Output resistance	T1OUT, T2OUT	V_{S+} = V_{S-} = 0, V_O = ±2 V	300			Ω
OS§	Short-circuit output current	T1OUT, T2OUT	V_{CC} = 5.5 V, V_O = 0		±10		mA
	Short-circuit input current	T1IN, T2IN	V_I = 0			200	µA

typical values are at V_{CC} = 5 V, T_A = 25°C.
e algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage vels only.
t more than one output should be shorted at a time.
E 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

itching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
	Driver slew rate	R_L = 3 kΩ to 7 kΩ, See Figure 2			30	V/µs
(t)	Driver transition region slew rate	See Figure 3		3		V/µs
	Data rate	One TOUT switching		120		kbit/s

E 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

ctrical characteristics over recommended ranges of supply voltage and operating free-air perature range (see Note 4)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
H	High-level output voltage	R1OUT, R2OUT	I_{OH} = −1 mA	3.5			V
L	Low-level output voltage‡	R1OUT, R2OUT	I_{OL} = 3.2 mA			0.4	V
+	Receiver positive-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V, T_A = 25°C		1.7	2.4	V
−	Receiver negative-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V, T_A = 25°C	0.8	1.2		V
s	Input hysteresis voltage	R1IN, R2IN	V_{CC} = 5 V	0.2	0.5	1	V
	Receiver input resistance	R1IN, R2IN	V_{CC} = 5, T_A = 25°C	3	5	7	kΩ

ypical values are at V_{CC} = 5 V, T_A = 25°C.
algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage ls only.
E 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

tching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4 and Figure 1)

PARAMETER		TYP	UNIT
H(R)	Receiver propagation delay time, low- to high-level output	500	ns
L(R)	Receiver propagation delay time, high- to low-level output	500	ns

E 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

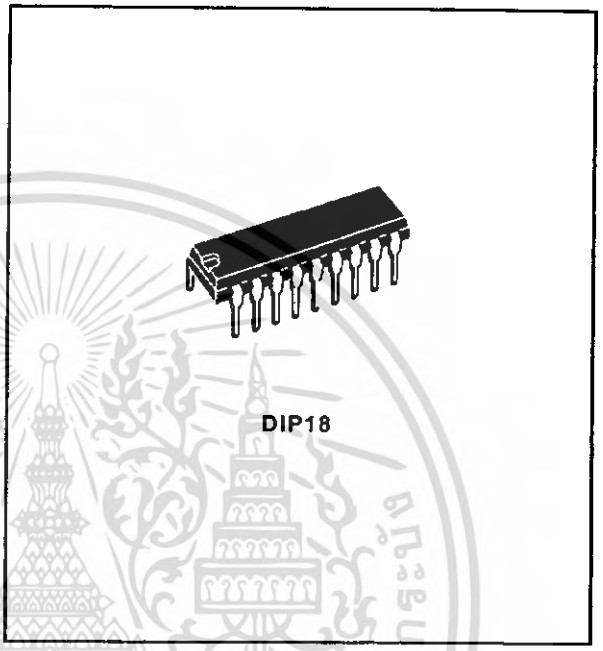
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGS-THOMSON MICROELECTRONICS

ULN2801A
ULN2802A - ULN2803A
ULN2804A - ULN2805A

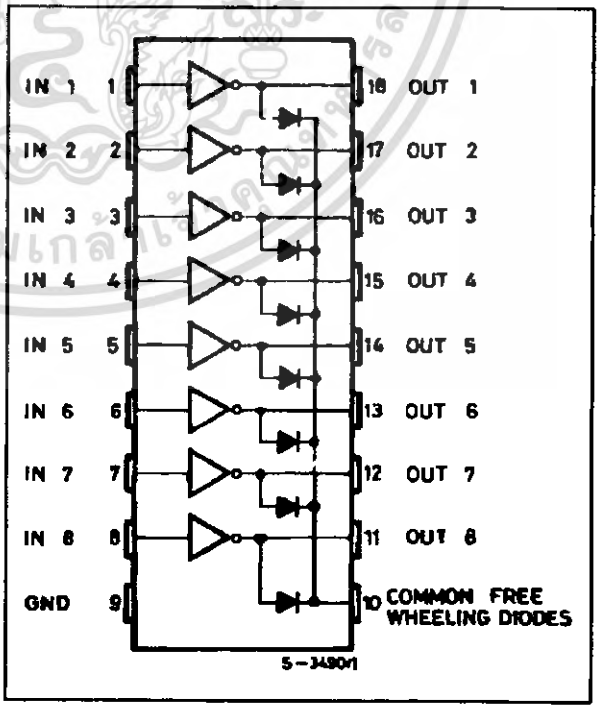
EIGHT DARLINGTON ARRAYS

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT



DIP18

PIN CONNECTION (top view)



DESCRIPTION

The ULN2801A-ULN2805A each contain eight darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability.

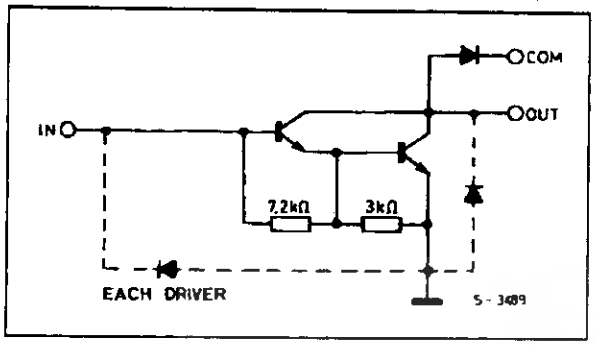
Five versions are available to simplify interfacing to standard logic families : the ULN2801A is designed for general purpose applications with a current limit resistor; the ULN2802A has a 10.5kΩ input resistor and zener for 14-25V PMOS; the ULN2803A has a 2.7kΩ input resistor for 5V TTL and CMOS ; the ULN2804A has a 10.5kΩ input resistor for 6-15V CMOS and the ULN2805A is designed to sink a minimum of 350mA for standard and Schottky TTL where higher output current is required.

All types are supplied in a 18-lead plastic DIP with a copper lead from and feature the convenient input-opposite-output pinout to simplify board layout.

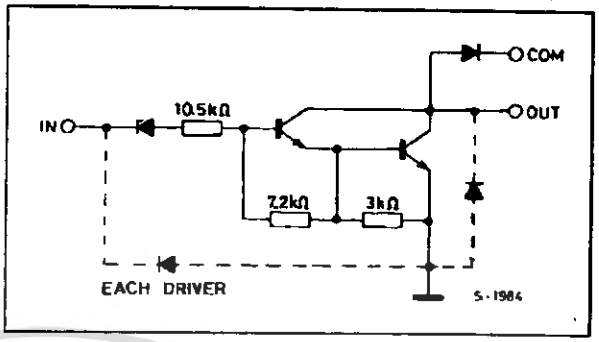
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCHEMATIC DIAGRAM AND ORDER CODES

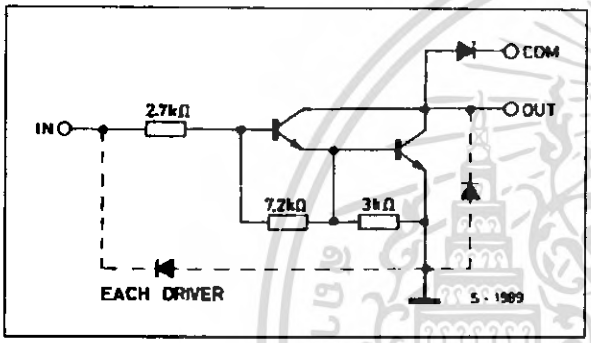
For ULN2801A (each driver for PMOS-CMOS)



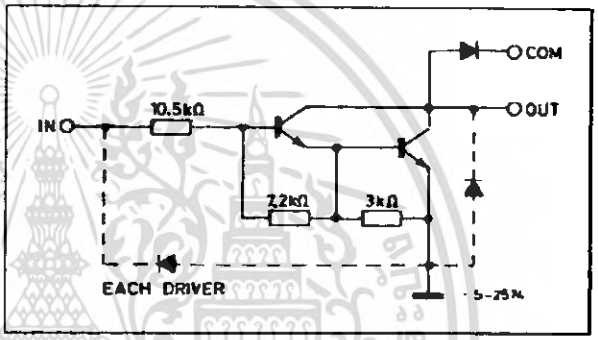
For ULN2802A (each driver for 14-15 V PMOS)



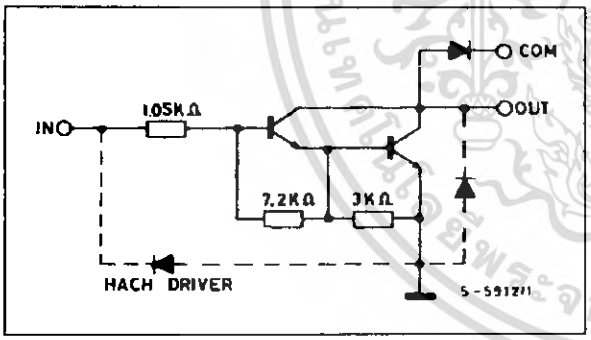
For ULN2803A (each driver for 5 V, TTL/CMOS)



For ULN2804A (each driver for 6-15 V CMOS/PMOS



For ULN2805A (each driver for high out TTL)



SGS-THOMSON
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_i	Input Voltage for ULN2802A, UL2803A, ULN2804A for ULN2805A	 30 15	V
I_C	Continuous Collector Current	500	mA
I_B	Continuous Base Current	25	mA
P_{tot}	Power Dissipation (one Darlington pair) (total package)	 1.0 2.25	W
T_{amb}	Operating Ambient Temperature Range	– 20 to 85	°C
T_{stg}	Storage Temperature Range	– 55 to 150	°C
Tj	Junction Temperature Range	– 20 to 150	°C

THERMAL DATA

Symbol	Parameter		Value	Unit
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	55	°C/W

ELECTRICAL CHARACTERISTICS (T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	V_{CE} = 50V			50	µA	1a
		T_{amb} = 70°C, V_{CE} = 50V			100	µA	1a
		T_{amb} = 70°C for ULN2802A V_{CE} = 50V, V_i = 6V			500	µA	1b
		for ULN2804A V_{CE} = 50V, V_i = 1V			500	µA	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	I_C = 100mA, I_B = 250µA		0.9	1.1	V	
		I_C = 200mA, I_B = 350µA		1.1	1.3	V	2
		I_C = 350mA, I_B = 500µA		1.3	1.6	V	
$I_{i(on)}$	Input Current	for ULN2802A V_i = 17V		0.82	1.25	mA	
		for ULN2803A V_i = 3.85V		0.93	1.35	mA	
		for ULN2804A V_i = 5V		0.35	0.5	mA	3
		V_i = 12V		1	1.45	mA	
		for ULN2805A V_i = 3V		1.5	2.4	mA	
$I_{i(off)}$	Input Current	T_{amb} = 70°C, I_C = 500µA	50	65		µA	4
$V_{i(on)}$	Input Voltage	V_{CE} = 2 V for ULN2802A					
		I_C = 300mA			13	V	
		for ULN2803A					
		I_C = 200mA			2.4	V	
		I_C = 250mA			2.7	V	
		I_C = 300mA			3	V	5
		for ULN2804A					
		I_C = 125mA			5	V	
		I_C = 200mA			6	V	
		I_C = 275mA			7	V	
		I_C = 350mA			8	V	
		for ULN2805A I_C = 350mA			2.4	V	
h_{FE}	DC Forward Current Gain	for ULN2801A V_{CE} = 2V, I_C = 350mA	1000			–	2
C_i	Input Capacitance			15	25	pF	–
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	–
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	–
I_R	Clamp Diode Leakage Current	V_R = 50V			50	µA	6
		T_{amb} = 70°C, V_R = 50V			100	µA	6
V_F	Clamp Diode Forward Voltage	I_F = 350mA		1.7	2	V	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

itures

- mpatible with MCS-51™ Products
- Bytes of Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- V to 6V Operating Range
- ly Static Operation: 0 Hz to 24 MHz
- o-Level Program Memory Lock
- x 8-Bit Internal RAM
- Programmable I/O Lines
- 16-Bit Timer/Counters
- Interrupt Sources
- grammable Serial UART Channel
- ect LED Drive Outputs
- Chip Analog Comparator
- Power Idle and Power Down Modes

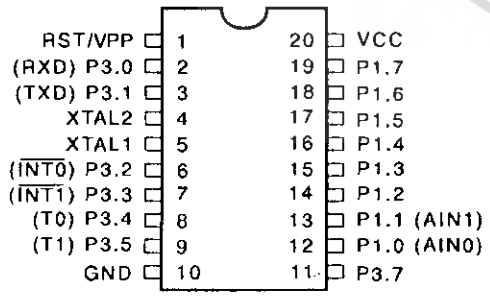
cription

T89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with ytes of Flash programmable and erasable read only memory (PEROM). The e is manufactured using Atmel's high density nonvolatile memory technology s compatible with the industry standard MCS-51™ instruction set. By combining satile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a pow- microcomputer which provides a highly flexible and cost effective solution to embedded control applications.

T89C2051 provides the following standard features: 2K Bytes of Flash, 128 of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt ecture, a full duplex serial port, a precision analog comparator, on-chip oscillator lock circuitry. In addition, the AT89C2051 is designed with static logic for opera- lown to zero frequency and supports two software selectable power saving s. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial nd interrupt system to continue functioning. The Power Down Mode saves the contents but freezes the oscillator disabling all other chip functions until the next are reset.

Configuration

PDIP/SOIC



8-Bit Microcontroller with 2K Bytes Flash

AT89C2051

0368D-B–12/97

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้