สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสั่งอาหารด้วยรีโมทคอนโทรล

Food Ordering Remote Control

โดย

นายเอกคณิต จันทร์สว่าง

นายเอกลักษณ์ ชดทิศ

เลขหมู่.....
เลขทะเบียน...33928
วัน, เดือน, ปี 20 ก.ย. 2542

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2541

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การสั่งอาหารด้วยรีโมทคอนโทรล

(Food Ordering Remote Control)

จัดทำโดย

นายเอกคณิต จันทร์สว่าง 38014659

นายเอกลักษณ์ ชดทิศ 38014671

.....อาจารย์ที่ปรึกษา

(อาจารย์ วิศรุต ศรีรัตนะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสั่งอาหารด้วยรีโมทคอนโทรล

นายเอกคณิต จันทร์สว่าง 38014659
นายเอกลักษณ์ ชคทิศ 38014671
อาจารย์ที่ปรึกษา
อ. วิศรุต ศรีรัตนะ

บทคัดย่อ

เครื่องควบคุมระยะไกล หรือมักจะนิยมเรียกกันว่า รีโมทคอนโทรล ในปัจจุบันนี้ได้เข้ามามีบทบาทในชีวิตประจำวันมากยิ่งขึ้นเนื่องจากสามารถทำการควบคุมอุปกรณ์ต่างๆ ได้อย่างสะดวกและรวดเร็ว โดยมากแล้วรีโมทคอนโทรล จะใช้หลักการของการส่งคลื่นความถี่ในย่านอินฟาเรด ซึ่งเราสามารถนำเอารหัสที่ได้จากการรับ และส่งข้อมูลจากตัวรีโมทคอนโทรลมาประยุกต์ใช้กับการจัดการทางด้านฐานข้อมูลได้

กระบวนการในการส่งสัญญาณของรีโมทคอนโทรล นั้น เมื่อทำการส่งสัญญาณทางด้านตัวส่งทางด้านตัวรับก็จะทำการถอดรหัสสัญญาณที่ได้รับ และส่งต่อไปยังไมโครคอนโทรลเลอร์ 89C2051 เพื่อทำการแปลงสัญญาณออกมาโดยใช้มาตราฐานRS-232 และข้อตกลงการสื่อสารโปรโตคอลเข้าไปยังหน่วยฐานข้อมูลโดยใช้โปรแกรม Visual Basic ในการออกแบบโปรแกรมสำหรับการจัดการด้านฐานข้อมูลสำหรับการแสดงผลการตรวจสอบของสัญญาณที่ได้รับเข้านั้นตรงกับข้อมูลใดที่กำหนดไว้ นอกจากนั้นไมโครคอนโทรลเลอร์ MAX7219 จะแสดงผลเป็นรหัสออกทาง 7- Segment เพื่อยืนยันว่าได้ส่งค่าดังกล่าวแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Food Ordering Remote control

MR.Ekkanit Chansawang 38014659
MR.Ekalack Chakatis 38014671
MR.Wisarut Srirattana Advisor
1998

ABSTRACT

The long distances controlling machine , which is normally called Remote Control. At present the wireless remote control involves in our everyday life to control any components, which is quite fast and convenient. Basically Remote Control uses the principle of The Infrared Transmission which we apply the process in sending and receiving the code to access the database information.

The remote control transmission process, when transmitting, the receiver decodes the signal, then sends data to Microcontroller 89C2051 in order to interface to RS –232 standard communication and communication protocal, By used program Visual Basic design this program to access a database unit for detecting the received signal whether it will match any defined codes. In addition Microcontroller MAX7219 displays the code on 7 – segment to show the data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญภาพ	V
บทที่	
1 บทนำ	1
1.1 ความสำคัญของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 ขั้นตอนและวิธีการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
2 ทฤษฎีพื้นฐาน	3
2.1 ทฤษฎีเบื้องต้นของแสง	3
2.2 เครื่องส่งและรับอินฟราเรด (Transmitter and Receiver)	5
2.3 การสร้างสัญญาณ	7
2.4 การควบคุมแบบจัดลำดับอนุกรม	7
2.5 การเพิ่มฟังก์ชั่นการควบคุม	8
2.6 วงจรจ่ายกำลังหรือวงจรขับเอาต์พุท	8
2.7 การกำหนดช่องสัญญาณ	11
2.8 สัญญาณโทนเบิรสต์ (Tone Burst)	13
2.9 วงจรรับอย่างง่าย	15
2.10 ตัวเข้ารหัสแป้นกดข้อมูล	16
2.11 เบานส์ (Bounce)	17
2.12 โรว์โอเวอร์ (Rollover)	17
2.13 ชนิดของแป้นข้อมูล	18
2.14 ไมโครคอนโทรลเลอร์ MCS-51	19
2.15 คอนโทรลภายในของ Visual Basic5	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

บทที่		หน้า
3	วงจรส่งและรับอินฟราเรด	37
	3.1 ภาคเครื่องส่ง (Transmitter)	37
	3.1.1 การทำงานของภาคส่ง	37
	3.1.2 คุณสมบัติของ MC145026	38
	3.1.3 การคำนวณและการทำงานของภาคมอดดูเลเตอร์	42
	3.2 ภาครับ MC145027	43
4	การทดสอบการทำงานของระบบ	47
	4.1 ผลการทดลอง	47
	4.1.1 ผลการทดลองวงจรส่งและวงจรรับอินฟราเรด	47
	4.1.2 หน้าจอแสดงผลการทำงานใน Visual Basic	62
	4.2 สรุปผลการทดลอง	70
	4.3 ข้อเสนอแนะและวิจารณ์	70
บรรณานุกรม		71
ภาคผนวก		72
ภาคผนวก ก. รายการอาหาร		
ภาคผนวก ข. Source Code ของโปรแกรม Visual Basic และไมโคลคอนโทรลเลอร์		
ภาคผนวก ค. ผังการทำงานของไมโครคอนโทรลเลอร์		
ภาคผนวก ง. รายละเอียดข้อมูลของไอซีที่ใช้ในโครงงาน		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

		หน้า
2.1	แสดงการแพร่กระจายของแม่เหล็กไฟฟ้าและการตอบสนอง	4
2.2	แสดงวงจรพื้นฐานของเครื่องส่งอินฟราเรด	5
2.3	แสดงวงจรพื้นฐานของเครื่องรับอินฟราเรด	6
2.4	แสดงวงจรรับอินฟราเรดที่ปรับปรุงแล้ว	6
2.5	แสดงวงจรควบคุมการจัดลำดับอนุกรม	7
2.6	แสดงวงจรถอดรหัส	8
2.7	แสดงวงจรจ่ายกำลังอย่างง่ายที่ใช้ทรานซิสเตอร์	10
2.8	แสดงวงจรขับโหลดและทรานซิสเตอร์	11
2.9	แสดงแผนภาพเวลาการส่งรหัส	12
2.10	แสดงสัญญาณพัลส์ที่มีสภาวะสูง	13
2.11	แสดงวงจรชนิดอะสเตเบิลมัลติไวเบรเตอร์	14
2.12	แสดงการนำสัญญาณความถี่ไปขับ LED ชนิดอินฟราเรด	15
2.13	แสดงวงจรรับอย่างง่าย	16
2.14	แสดงผลของแผนภูมิการใช้งานในโหมด 1 พร้อมกับแผนภูมิเวลาสำหรับสัญญาณการส่งและรับ	22
2.15	แสดงคอนโทรล Control Button ในขณะออกแบบ	26
2.16	แสดงคอนโทรล Textbox ในการออกแบบ	26
2.17	แสดงคอนโทรล Label ที่ถูกวางลงในฟอร์มในขณะออกแบบ	27
2.18	แสดงคอนโทรล Timer ในขณะออกแบบ	27
2.19	แสดงสถาปัตยกรรมด้านฐานข้อมูลของ Visual Basic	30
2.20	แสดงคอนโทรล Data ในขณะออกแบบ	31
2.21	แสดงคอนโทรล List View ในขณะออกแบบ	32
2.22	แสดงคอนโทรล MSComm ในขณะออกแบบ	35
2.23	แสดงการใช้งาน Crystal Report	36
3.1	บล็อกไดอะแกรม	37
3.2	ตัวเข้ารหัสของ MC145026	37
3.3	บล็อกไดอะแกรมตัวเข้ารหัสของ MC 145026	38
3.4	ตัวเข้ารหัสออสซิลเลเตอร์อินฟรอแมชั่น(Oscillator Information)	39
3.5	ตัวเข้ารหัสออสซิลเลเตอร์	40
3.6	แผนภูมิเวลาของตัวเข้ารหัส	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

		หน้า
3.7	ตัวเข้ารหัสออสซิลเลเตอร์ (Pin12)	41
3.8	แสดงลักษณะการเข้ารหัสของบิทข้อมูล	41
3.9	แผนภูมิเวลา	42
3.10	รูปคลื่นของข้อมูลตัวเข้ารหัส	42
3.11	แสดงสัญญาณขาเข้าและขาออกของภาคมอดดูเลชั่น	43
3.12	แผนภูมิการทำงานของ MC 145027	44
3.13	R-C Decay on Pin 7 (C1)	46
3.14	R-C Decay on Pin 10 (R2/C2)	46
4.1	Channel 0. XXXXX0001	47
4.2	Channel 1. XXXXX0010	48
4.3	Channel 2. XXXXX0011	48
4.4	Channel 3. XXXXX0100	49
4.5	Channel 4. XXXXX0101	49
4.6	Channel 5. XXXXX0110	50
4.7	Channel 6. XXXXX0111	50
4.8	Channel 7. XXXXX1000	51
4.9	Channel 8. XXXXX1001	51
4.10	Channel 9. XXXXX1010	52
4.11	Channel Order. XXXXX1111	52
4.12	Channel Delete. XXXXX0000	53
4.13	แสดงคลื่นความถี่พาหะ	54
4.14	แสดงสัญญาณ Output เมื่อให้สัญญาณ Input กับวงจรกำเนิดคลื่นพาหะ	54
4.15	Channel 0. XXXXX0001	55
4.16	Channel 1. XXXXX0010	56
4.17	Channel 2. XXXXX0011	56
4.18	Channel 3. XXXXX0100	57
4.19	Channel 4. XXXXX0101	57
4.20	Channel 5. XXXXX0110	58
4.21	Channel 6. XXXXX0111	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

		หน้า
4.22	Channel 7. XXXXX1000	59
4.23	Channel 8. XXXXX1001	59
4.24	Channel 9. XXXXX1010	60
4.25	Channel Order. XXXXX1111	60
4.26	Channel Delete. XXXXX0000	61
4.27	แสดงการใส่ Password ก่อนเข้าสู่การทำงาน	62
4.28	แสดงรายการอาหารที่ได้สั่งออกทางหน้าจอ	62
4.29	แสดงรายการออกทางเครื่องพิมพ์	63
4.30	แสดงรายการที่เก็บเป็นฐานข้อมูล	63
4.31	รูปวงจรภาคส่ง	64
4.32	รูปวงจรภาครับ	65
4.33	แสดงวงจรภาคส่งที่ใช้ในโครงงาน	66
4.34	แสดงวงจรภาครับที่ใช้ในโครงงาน	66
4.35	ภาคควบคุมการแสดงผล	67
4.36	แสดงภาคควบคุมการแสดงผลที่ใช้ในโครงงาน	68
4.37	แสดงวงจรของไมโครคอนโทรลเลอร์	68
4.38	แสดงชุดรีโมทคอนโทรลสั่งอาหาร	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ความสำคัญของโครงงาน

ในปัจจุบัน ระบบการควบคุม ได้เข้ามามีส่วนเกี่ยวข้องกับชีวิตประจำวันของมนุษย์มากขึ้น จนกลายเป็นสิ่งจำเป็นต่อการดำเนินชีวิตในปัจจุบัน ยกตัวอย่าง เช่น ระบบการควบคุมที่ใช้กับที่พักอาศัย โดยอาจจะควบคุมการเปิดหรือปิดของประตูบ้าน โรงรถ หรือการควบคุมอุณหภูมิของเครื่องปรับอากาศ ให้อยู่ในระดับที่พอดีกับที่เราต้องการ และอื่นๆ อีกมากมาย หรือแม้กระทั่งในแวดวงของธุรกิจซึ่งจะทำให้เกิด ความสะดวก รวดเร็ว ประหยัดและปลอดภัยต่อชีวิตและทรัพย์สินต่อผู้ใช้

ระบบการควบคุมระยะไกลด้วยแสงอินฟราเรดหรืออินฟราเรดรีโมท คอนโทรลระบบควบคุมระยะไกลชนิดนี้ก็เป็นอีกชนิดหนึ่งของระบบการควบคุมที่ได้มีการนำมาใช้กันอย่างแพร่หลาย เนื่องจากมันสามารถควบคุมการทำงานของวงจรหรืออุปกรณ์เครื่องมือเครื่องใช้ไฟฟ้าได้จากระยะไกล โดยไม่มีผลกระทบในเรื่องของการแพร่กระจายของคลื่นแม่เหล็กไฟฟ้า และคลื่นวิทยุไปรบกวนวงจรหรืออุปกรณ์ข้างเคียง ซึ่งเป็นข้อดีอย่างหนึ่งของระบบการควบคุมแบบนี้

จากแนวคิดดังกล่าวนี้จึงได้พัฒนาสร้างตัว Infrared Remote Control เมื่อใช้งานสำหรับการสั่งอาหาร โดยข้อมูลต่างๆ จะส่งผ่านทางแสงอินฟราเรดไปยังโมดูลรับ ซึ่งติดต่อกับโต๊ะคิดเงิน โดยมีเครื่องไมโครคอมพิวเตอร์ และเครื่องคอมพิวเตอร์ทำการถอดรหัสที่ส่งมาแล้วทำการตรวจสอบค่าที่ได้รับว่าตรงกับข้อมูลชนิดใดเพื่อทำการระบุถึงหมายเลขของตัวรีโมต รหัสอาหาร และจำนวนที่สั่งเข้ามา สำหรับห้องครัว และจัดพิมพ์ใบเสร็จเมื่อลูกค้าต้องการ ดังนั้นหลักการของการพัฒนานี้จะแบ่งออกเป็นสองส่วนใหญ่ๆ คือ ส่วนที่หนึ่งเป็นการพัฒนาโมดูลรับข้อมูลอาหาร และส่วนที่สองการพัฒนา Software สำหรับการรับข้อมูล, ประมวลผลและพิมพ์ใบเสร็จ

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 นำหลักการของการกระจายคลื่นของอินฟราเรดมาใช้งาน

1.2.2 เพิ่มความสะดวกสบายในชีวิตประจำวัน

1.2.3 เพิ่มความเที่ยงตรงสำหรับการสั่งอาหารในแต่ละครั้ง

1.2.4 เพื่อทำสถิติและเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถบอกได้ว่าสัญญาณที่ส่งออกมาได้รับหรือไม่ โดยการแสดงผลบนจอ LED 7 ส่วน (7 segment)

1.3.2 ใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานของการรับและส่งข้อมูล

1.3.3 ใช้คอมพิวเตอร์ตรวจสอบว่าข้อมูลที่รับเข้ามาตรงกับรายการอาหารประเภทใดและแสดงผล

1.3.4 ทำฐานข้อมูลเกี่ยวกับรายการอาหารและจำนวนที่รับเข้ามา

1.4 วิธีการดำเนินการ

1.4.1 ศึกษาและค้นคว้าเกี่ยวกับการรับและส่งด้วยคลื่นอินฟราเรด

1.4.2 ศึกษาการทำงานของการส่งผ่านแบบอนุกรมของไมโครคอนโทรลเลอร์

1.4.3 ศึกษาและออกแบบการทำงาน

1.4.4 ศึกษาการทำงานของอุปกรณ์ที่ใช้ในระบบ

1.4.5 ทดสอบการทำงานของระบบ

1.4.6 ปรับปรุงแก้ไข

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำวิจัย

1.5.1 เข้าใจการทำงานของการรับและส่งด้วยคลื่นอินฟราเรด

1.5.2 เข้าใจหลักการทำงานของไมโครคอนโทรลเลอร์

1.5.3 สามารถออกแบบวงจรการทำงานทางอิเลคทรอนิกส์

1.5.4 เข้าใจหลักการทำงานของระบบฐานข้อมูล

1.5.5 เข้าใจการทำงานของโปรแกรม Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐาน

2.1 ทฤษฎีเบื้องต้นของแสง

เหตุผลที่เราเลือกใช้แสงในการสื่อสารทางโทรคมนาคมนั้นมีเหตุผลใหญ่ๆ อยู่ 2 ข้อคือ

2.1.1 สามารถป้องกันการสอดแทรกต่างๆ ทั้งยังทำให้เกิดความปลอดภัยในการสื่อสาร

2.1.2 การรบกวนจากสัญญาณรบกวน (noise) ต่างๆ เช่น จากมอเตอร์หรือ Electromagnetic Interference (EI) ไม่สามารถที่จะรบกวนได้

สำหรับปัญหาทั้งสองข้อนี้ เราสามารถป้องกันได้ โดยการใช้แสงซึ่งมีความถี่สูงมากแทนการส่งโดยใช้ความถี่วิทยุ (Radio Wave) ย่านความถี่แสงที่ใช้ส่งนี้ เราก็ใช้แสงในย่านอินฟราเรดซึ่งเป็นแสงที่เราไม่สามารถมองเห็นได้ จากภาพที่ 2.1 เป็น Spectrum ของคลื่นแม่เหล็กไฟฟ้าที่ประกอบด้วยแถบความถี่ของวิทยุ และโทรทัศน์, ไมโครเวฟ, การกระจายแสงอินฟราเรด, แสงที่มองเห็นได้, Visible Light, Ultraviolet, X-Ray Grammer Ray และความถี่อื่นๆ โดยความแตกต่างของการแพร่กระจายต่างๆ ของแต่ละแถบคลื่นนั้น จะขึ้นอยู่กับความถี่และความยาวคลื่นเท่านั้น

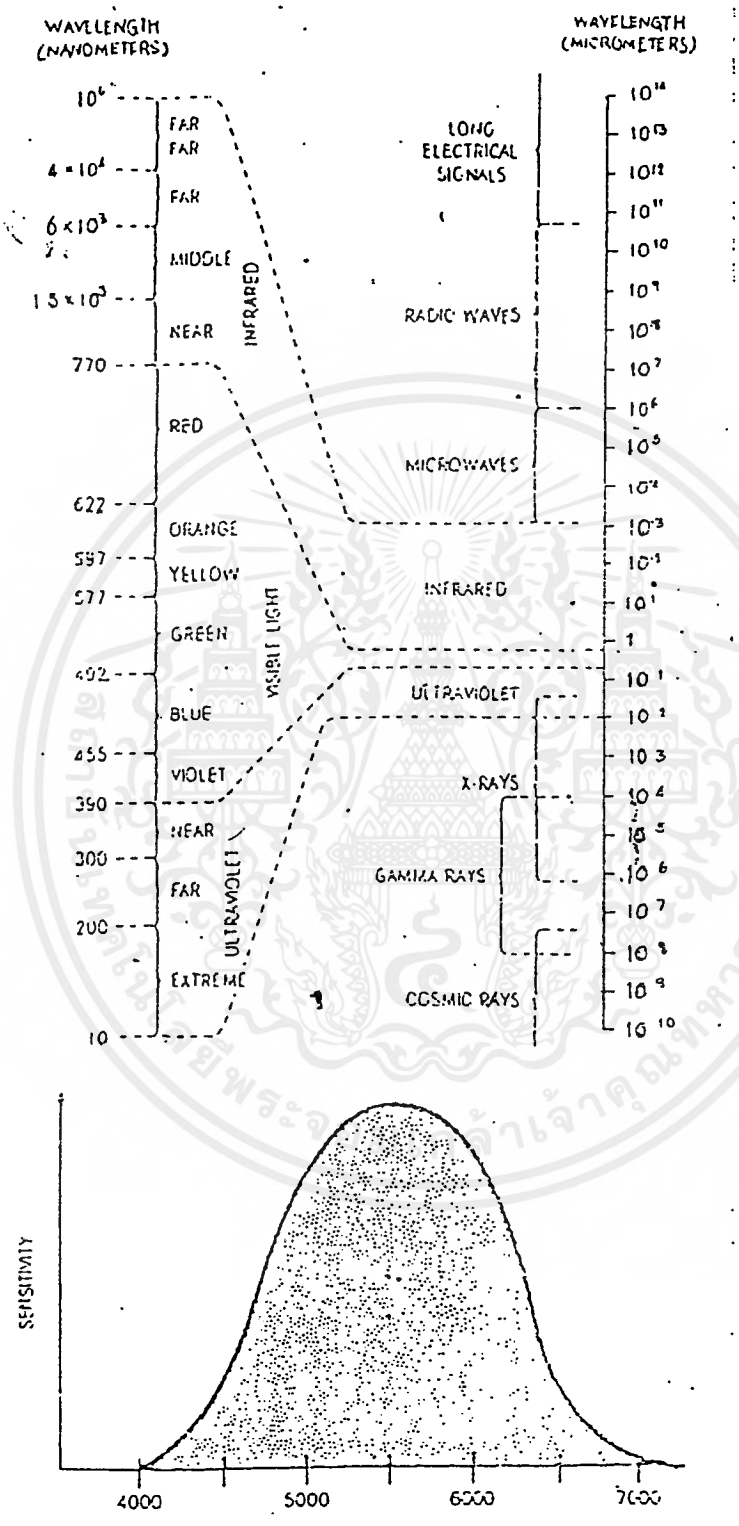
ตาของมนุษย์จะมีการตอบสนองคลื่นแม่เหล็กไฟฟ้าในแถบของแสงที่มองเห็นได้ โดยแสงแต่ละสีที่ตามองเห็นนั้น จะมีความถี่หรือความยาวคลื่นแตกต่างกัน โดยเมื่อได้พิจารณากันตามการเพิ่มของความถี่หรือการลดลงของความยาวคลื่นของสีต่างๆ จะเรียงลำดับได้ดังนี้คือ

แดง	ความยาวคลื่น	622-770	นาโนเมตร
ส้ม	ความยาวคลื่น	597-622	นาโนเมตร
เหลือง	ความยาวคลื่น	577-597	นาโนเมตร
เขียว	ความยาวคลื่น	492-577	นาโนเมตร
น้ำเงิน	ความยาวคลื่น	455-492	นาโนเมตร
ม่วง	ความยาวคลื่น	390-455	นาโนเมตร

ดังนั้น ความยาวของคลื่นที่มองเห็นได้จะอยู่ในช่วงระหว่างประมาณ 40 นาโนเมตรถึง 700 นาโนเมตร ส่งย่านของแสงที่ตาไม่สามารถตอบสนองได้มีอยู่ 2 อย่างคือ ย่านที่สูงกว่าหรือต่ำกว่า Visible Light Band ก็จะเริ่มตั้งแต่ ย่านอินฟราเรดและอุลตร้าไวโอเลท (Ultra Violet) ขึ้นไป

ดังนั้นการสื่อสารของเราซึ่งเป็นแบบระยะทางไกล เราจะใช้ในย่านอินฟราเรด ที่จะเห็นได้ว่าอยู่ในช่วงความยาวคลื่นประมาณ 700 ถึง 1000000 นาโนเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 แสดงการแพร่กระจายของแม่เหล็กไฟฟ้าและการตอบสนอง

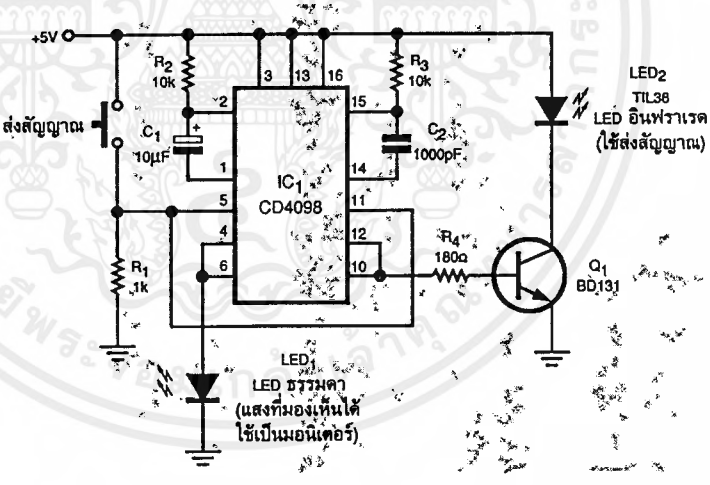
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 เครื่องส่งและรับอินฟราเรด

2.2.1 เครื่องส่งอินฟราเรด

การเพิ่มระยะทางในการทำงานของแสงอินฟราเรด นอกเหนือจากการเพิ่มจำนวน LED แล้ว อาจะทำได้ด้วยวิธีอื่นอีก ถึงแม้ว่าจะใช้ LED เพียงตัวเดียวก็ตาม นั่นคือการเพิ่มกระแสขับ LED ให้สูงขึ้น โดยที่ LED ต้องไม่เสียหาย ซึ่งสามารถกระทำได้โดยการจำกัดช่วงเวลาทำงานของ LED ให้สั้นลง ด้วยการใช้สัญญาณพัลส์เป็นตัวขับกระแสให้ LED จากวงจรในภาพที่ 2.5 (ก) หากเราป้อนสัญญาณพัลส์ที่มีความกว้างพัลส์ไม่เกิน 10 ไมโครวินาที และมีความพี่ไม่เกิน 1 กิโลเฮริตซ์ กระแสที่ใช้ขับ LED อาจมีค่าสูงได้ถึง 2 แอมป์ โดยที่ LED ไม่เสียหาย ซึ่งจะส่งผลทำให้ความเข้มแสงที่เปล่งออกมามีค่าสูงมาก และเดินทางไปได้ไกลกว่าเดิมมากทีเดียว

วงจรในภาพที่ 2.2 ใช้สำหรับสร้างพัลส์ ขนาดเวลา 10 ไมโครวินาทีให้กับ LED โดยมีทรานซิสเตอร์ Q_1 เป็นตัวขับกระแส สัญญาณควบคุมทรานซิสเตอร์มาจากวงจรโมโนสเตเบิ้ล ในรูปบางครั้งอาจก่อให้เกิดปัญหาในการส่งสัญญาณได้ เพราะตัวสวิตซ์เองอาจมีปัญหาด้านกลไกในตัวมัน เมื่อมีการกดสวิตซ์กระบบหน้าสัมผัสภายในอาจเกิดการสั่นหรือสัมผัสกันมากกว่าหนึ่งครั้ง ทำให้สัญญาณที่ได้เสมือนเปิดหรือปิดต่อๆ กันหลายครั้ง ส่งผลให้สัญญาณเอาต์พุทของวงจรโมโนสเตเบิ้ลมีการผิดพลาดได้ การแก้ไขอาจทำได้โดยการเพิ่มเกตแบบชมิตต์ทริกเกอร์เข้าไปเพื่อตัดปัญหาของสวิตซ์ดังกล่าว

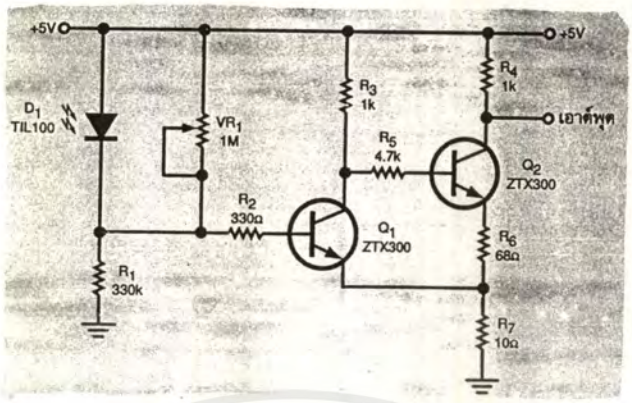


ภาพที่ 2.2 แสดงวงจรพื้นฐานของเครื่องส่งอินฟราเรด

2.2.2 เครื่องรับอินฟราเรด

การรับสัญญาณแสงอินฟราเรดด้วยวงจรพื้นฐานแสดงดังภาพที่ 2.3 ซึ่งประกอบด้วยส่วนของตัวรับแสงที่ใช้โฟโตไดโอดทำหน้าที่แปลงสัญญาณเป็นสัญญาณไฟฟ้า ในการติดตั้งโฟโตไดโอดเพื่อรับแสง ควรมีแผ่นกรองแสง หรือ ฟิลเตอร์ ชนิดที่ให้แสงอินฟราเรดผ่านได้ (เช่น แผ่นพลาสติกในสีแดงเข้ม) วางไว้ด้านหน้า เพื่อป้องกันการรบกวนของคลื่นแสงตัวอื่น ที่อาจตกกระทบเข้ามา

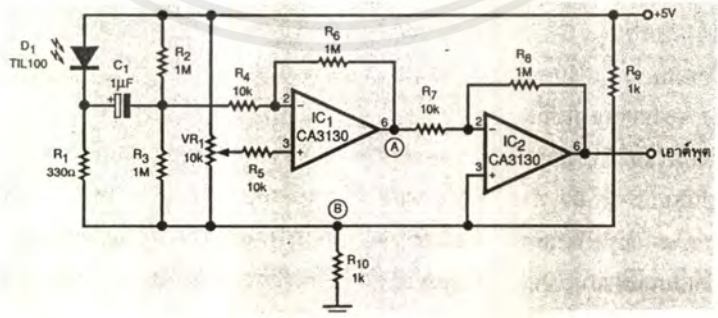
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 แสดงวงจรพื้นฐานของเครื่องรับอินฟราเรด

กระแสที่ไหลผ่านโฟโตไดโอดและตัวต้านทาน R_1 อันจะส่งผลให้เกิดการเพิ่มกระแสเบสของทรานซิสเตอร์ Q_1ทำให้ Q_1 ทำงานและเมื่อ Q_1 ทำงาน กระแสเบสทรานซิสเตอร์ Q_2 จะลดลงส่งผลให้ทรานซิสเตอร์ Q_2 ไม่ทำงาน ทำให้เอาต์พุทของวงจร อยู่ในสภาวะ "1" เมื่อมีสัญญาณอินฟราเรดเข้ามา ความไวในการรับสัญญาณของวงจร ขึ้นกับระดับที่ตกคร่อม R_1 อันเนื่องมาจากปริมาณของกระแสที่ไหลผ่านโฟโตไดโอด V_{R1} ที่ทำหน้าที่กำหนดปริมาณกระแสไบแอสที่ขาเบสของทรานซิสเตอร์ Q_1

วงจรในภาพที่ 2.4 เป็นวงจรรับแสงอินฟราเรดที่ถูกปรับปรุงให้มีความไวในการรับสัญญาณแสงได้ดีขึ้น โดยการใช้ตัวเก็บประจุ C_7 เป็นตัวผ่านของสัญญาณเริ่มต้นที่รับได้ ซึ่งช่วยให้วงจรมีการตอบสนองต่อการเปลี่ยนแปลงสัญญาณชั่วขณะได้ดีขึ้น อันเป็นผลจากคุณสมบัติในการเก็บประจุและคายประจุของ C_1 นั้นเอง



ภาพที่ 2.4 แสดงวงจรรับอินฟราเรดที่ปรับปรุงแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

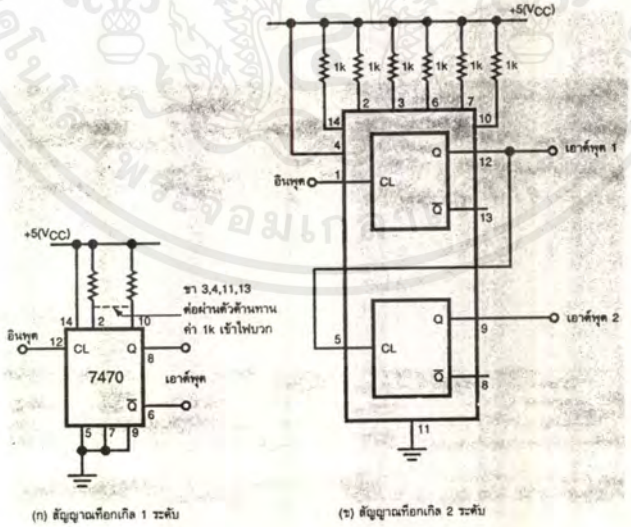
สัญญาณอินพุทที่รับได้จะถูกส่งไปเข้าวงจรขยาย โดย I_{C1} ซึ่งเป็นตัวปรับแรงดัย และแรงดันอินพุทที่ขา 2 เป็นแรงดันที่มาทำการเปรียบเทียบกับแรงดันที่ขา 3 ซึ่งแรงดันที่ขา 3 นั้นปกติจะปรับให้เป็นวงจรขยายแบบกลับสัญญาณให้เอาต์พุทออกมาที่ขา 6 เพื่อส่งไปควบคุมวงจรใช้งานอื่นๆ อีกต่อไป

2.3 การสร้างสัญญาณ

โดยทั่วไปการสร้างสัญญาณควบคุมพอที่จะแบ่งออกได้เป็น 2 .ประเภท คือการควบคุมแบบอนุกรม ที่มีรูปแบบวงจรส่งที่มีสัญญาณรูปแบบเดียวกันได้ อีกประเภทหนึ่งเป็นการควบคุมโดยใช้การเข้ารหัส วิธีนั้นทั้งวงจรส่งและวงจรรับ ต้องมีส่วนของการเข้ารหัสและถอดรหัสที่สัมพันธ์กัน ซึ่งจะกล่าวโดยละเอียดต่อไป ในที่นี้จะเริ่มกล่าวถึงการควบคุมแบบจำกัดลำดับก่อน

2.4 การควบคุมแบบจัดลำดับอนุกรม

ย้อนกลับไปพิจารณาสัญญาณเอาพุทของวงจรภาครับ ที่ขาคอลเล็กเตอร์ของทรานซิสเตอร์ Q_1 ในภาพที่ 2.2 อาจทำให้เป็นสัญญาณกระตุ้นที่มีรูปร่างเป็นฟัลล์ที่แน่นอนได้ โดยการใช้ไอซีทีทีแอลเบอร์ 7470 ทำหน้าที่เป็น J-K ฟลิปฟลอป ต่อเป็นวงจรดังแสดงในภาพที่ 2.5 (ก) โดยป้อนสัญญาณเข้าที่ขาคล็อก (C_L) ของไอซี สัญญาณเอาต์พุท Q เป็นตัวเลือกที่ถูกเลือกใช้เพื่อเป็นสัญญาณควบคุมในส่วนอื่นต่อไป เมื่อสัญญาณอินพุทที่ขา 12 เปลี่ยนสถานะจาก “1”ไปเป็น “0”



ภาพที่ 2.5 แสดงวงจรควบคุมการจัดลำดับอนุกรม

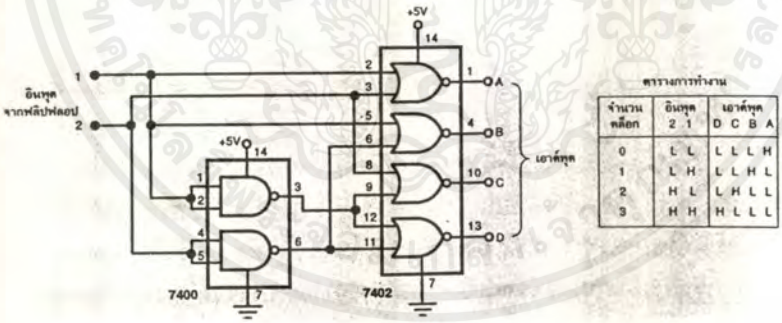
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเอาต์พุท Q จะเปลี่ยนสถานะไปจากเดิม ในขณะที่มันจะมีการรอจังหวะและคงสถานะไว้ เมื่อลักษณ์สัญญาณทางอินพุตเปลี่ยนสถานะจาก "0" ไปเป็น "1" คล้ายกับการทำงานของวงจรนับ สมมติถ้าต่อ LED เข้ากับสัญญาณเอาต์พุท Q นี้ เมื่อกดปุ่มส่งสัญญาณที่ตัวส่งอีกครั้งหนึ่ง LED จะติด(หรือดับแล้วแต่สักษณะการต่อวงจร) เสมือนกับการควบคุมอุปกรณ์ไฟฟ้านั้น ลักษณะสัญญาณควบคุมการปิดหรือเปิดอย่างง่ายนี้ เรียกว่า สัญญาณท็อกเกิล ซึ่งหมายถึงการที่สัญญาณเอาต์พุทเปลี่ยนค่าไปเป็นตรงกันข้ามในทุกครั้งที่ขาคล็อกเปลี่ยนสถาวะจาก "1"ไปเป็น"0"

2.5 การเพิ่มฟังก์ชั่นการควบคุม

จากสัญญาณควบคุมแบบท็อกเกิ้ลที่มีฟังก์ชั่นควบคุมการทำงาน 2 ฟังก์ชั่นนี้อาจดัดแปลงให้มีฟังก์ชั่นการทำงานมากขึ้นอีกได้ โดยการเพิ่มวงจรบางส่วนเข้าไป ลักษณะของวงจรที่นำมาใช้เป็นประเภทเดียวกับวงจรนับ (Counter) ดังแสดงในภาพที่ 2.5 (ข) หรือใช้ไอซีเบอร์ 7473 ที่ประกอบด้วยฟลิปฟลอป 2 ตัวอยู่ภายในตัวถังเดียวกัน ก็จะช่วยประหยัดค่าใช้จ่ายและลดจำนวนอปุกรณ์ลงไปได้

สัญญาณเอาต์พุท โดยภาพที่ 2.5 (ข) มี 2 เอาต์พุท เปรียบเสมือนเป็นค่าทางไบนารี 2 หลัก หรือวงจรนับ 4 ซึ่งทำให้เกิดสัญญาณควบคุมที่มีรูปแบบต่างกัน 4 แบบ และสามารถนำไปตัดแปลงจากรหัสไบนารีไปเป็นสัญญาณควบคุม ด้วยวงจรถอดรหัสหรือดีโคดเดอร์ สัญญาณเอาต์พุทที่ได้จากขา 12 และขา 9 ของไอซี 7473 ในภาพที่ 2.5 (ข) ดังวงจรถอดรหัสที่แสดงไว้ในภาพที่ 2.6



ตารางการทำงาน

จำนวนพัลส์	อินพุต 2 1	เอาต์พุต D C B A
0	L L	L L L H
1	L H	L L H L
2	H L	L H L L
3	H H	H L L L

ภาพที่ 2.6 แสดงวงจรถอดรหัส

วงจรในภาพที่ 2.6 เป็นโครงสร้างของวงจรถอดรหัสที่ประกอบด้วยแนนต์เกททั้งหมด 6 ตัว ซึ่งอาจมาจากไอซี 7401 และ 7402 สัญญาณควบคุมเอาต์พุท คือ ตำแหน่งเอาพุตย์ A, B, C และ D

2.6 วงจรจ่ายกำลังหรือวงจรขับเอาต์พุท

การนำสัญญาณควบคุมไปใช้ในการเปิด/ปิด ให้อุปกรณ์ไฟฟ้าทำงาน จำเป็นต้องอาศัยชุดวงจรจ่ายกำลังสำหรับโหลด หรือวงจรขับโหลดทางเอาต์พุท เพื่อจ่ายกระแสและแรงดันที่เหมาะสมกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

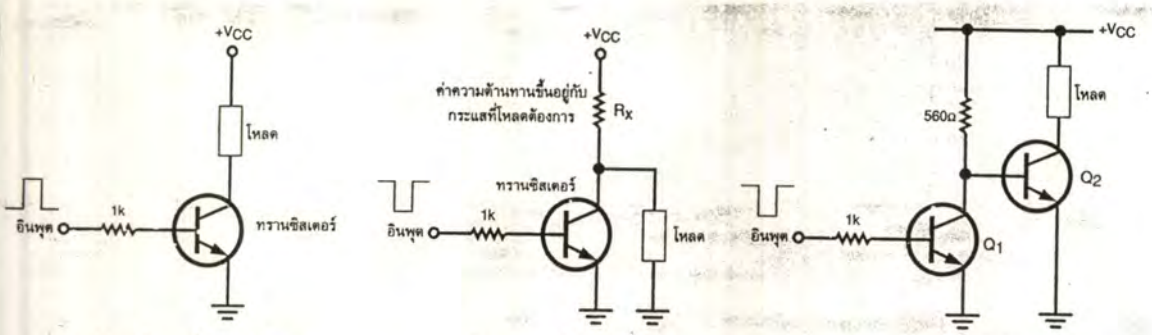
โหลดแต่ละชนิด หากสัญญาณควบคุมถูกใช้เพียงแค่สำหรับขับ LED ให้สว่างหรือดับ เพื่อแสดงสภาวะการทำงานก็อาจนำ LED ต่อเข้าโดยตรงกับสัญญาณควบคุมได้เลย (อาจเพื่อมตัวต้านทานเพื่อกำจัดกระแสอีกหนึ่งตัว) แต่ถ้าหากต้องการควบคุมอุปกรณ์ไฟฟ้าที่ต้องการกระแสและแรงดันมากกว่าที่ไอซีจะขับได้ ต้องใช้ทรานซิสเตอร์เป็นตัวสวิตซ์ หรือ ขยายกระแสให้กับโหลด โดยใช้สัญญาณควบคุมเป็นตัวกำหนดสักษณะการทำงานของทรานซิสเตอร์อีกทอดหนึ่งโดยปกติสัญญาณควบคุมที่เรากล่าวถึงมักอยู่ในรูปของสัญญาณดิจิตอลที่ขับออกมาจากไอซีประเภทต่างๆ จึงควรที่จะทราบค่าของกระแสและแรงดันทำงานของไอซีประเภทต่างๆ ดังแสดงๆในตารางที่ 1

ไอซีประเภทต่างๆ	กระแสสูงสุด (มิลลิแอมป์)	แรงดันทำงาน(โวลต์)
ทีทีแอลธรรมดา (ตระกูล 74 ทั่วไป)	16	5
ทีทีแอลที่เป็นบัฟเฟอร์ (เช่น 7440)	50	5
ทีทีแอลกำลังต่ำ (จำพวก 74L)	0.18	5
ทีทีแอลชอตต์กี้ (จำพวก 74LS)	0.36	5
ซีมอส (ที่แรงดันทำงาน 5 โวลต์)	4	5
(ที่แรงดันทำงาน 10 โวลต์)	11	10

ตารางที่ 1 แสดงค่ากระแสที่แรงดันทำงานของไอซีประเภทต่างๆ

วงจรจ่ายกำลังอย่างง่ายที่ใช้ทรานซิสเตอร์ แสดงดังภาพที่ 2.7 (ก)โหลดที่เป็นอุปกรณ์ไฟฟ้าที่ต้องการแรงดันไฟตรง จะทำงานเมื่อสัญญาณควบคุมทางขาเบสของทรานซิสเตอร์เป็น “1” ค่าของกระแสที่โหลดต้องการจะต้องไม่เกินค่าของกระแสคอลเล็กเตอร์ที่ถูกควบคุมโดยกระแสเบส ซึ่งเป็นกระแสสัญญาณควบคุมในรูปดิจิตอล วงจรนี้สามารถใช้ได้กับโหลดที่กินกระแสไฟได้หลายชนิด ขึ้นกับอัตราการขยายกระแส (Ic) ของทรานซิสเตอร์กำลังสูงเบอร์ 2N3055 จะสามารถจ่ายกระแสให้โหลดได้มากถึง 15 แอมป์เลยทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) โหลดทำงานเมื่ออินพุตเป็น "1"

(ข) โหลดทำงานเมื่ออินพุตเป็น "0"

(ค) โหลดทำงานเมื่ออินพุตเป็น "0" โดยมีการขยายกระแสด้วย Q_1 ก่อน

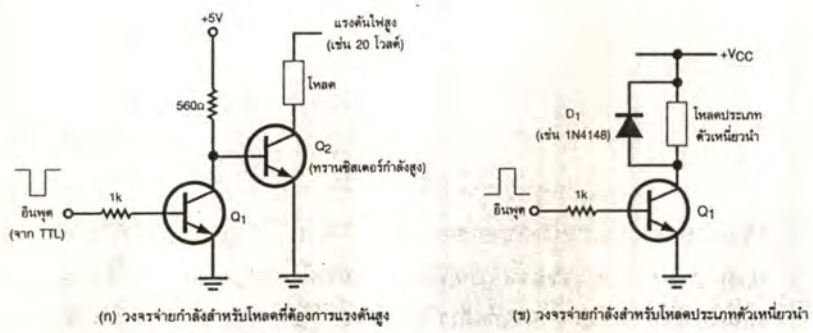
ภาพที่ 2.7 แสดงวงจรจ่ายกำลังอย่างง่ายที่ใช้ทรานซิสเตอร์

สำหรับวงจรในภาพที่ 2.7 (ข) มีหลักการทำงานตรงข้ามกับภาพที่ 2.7 (ก)คือโหลดจะทำงานเมื่อสัญญาณอินพุตเป็น "0" ทรานซิสเตอร์จะไม่ทำงาน ทำให้มีแรงดันตกคร่อมโหลดสูงเกือบเท่าแหล่งจ่าย Vcc ทำให้มีกระแสไหลผ่านตัวต้านทาน Rx ผ่านโหลดครบวงจร ถ้าอินพุตเป็น "1" ทรานซิสเตอร์จะทำงาน ทำให้แรงดันตกคร่อมโหลดเป็น ศูนย์ จึงไม่มีกระแสไหลผ่านโหลดจึงไม่ทำงาน

ส่วนวงจรในภาพ 2.7 (ค) นั้นเป็นการรวมการทำงานในภาพ (ก) และ (ข) เข้าด้วยกันคือเมื่ออินพุทเป็น "0" ทำให้ V_1 ไม่ทำงาน มีแรงดันตกคร่อม Vcc สูงเกือบเท่าแหล่งจ่าย ทำให้มีกระแสไหลผ่านตัวต้านทาน 560 โอห์ม ไบแอสให้ขาเบสของ Q_2 และ Q_2 ก็จะทำงานมีกระแสไหลผ่านโหลดครบวงจร แต่เมื่อสัญญาณอินพุทเป็น "1" จะทำให้ Q_1 ทำงานมีแรงดันตกคร่อม Vcc ของ Q_1 เป็นศูนย์โวทย์ ทำให้ Q_2 ที่ขาเบสได้รับไบแอสกลับ จึงไม่มีกระแสไหลผ่าน โหลดจึงไม่ทำงานในจังหวะที่อินพุทเป็น "1"

ในกรณีที่โหลดเป็นอุปกรณ์ที่ต้องการค่าแรงดันสูงๆ หรือค่าของแรงดัน –Vcc สูงๆอาจมีค่าตามที่โหลดต้องการได้หากมีการเลือกใช้วงจรขับโหลดและทรานซิสเตอร์ที่เหมาะสม หรืออาจใช้วงจรในภาพที่ 2.8 (ก) ซึ่งใช้ทรานซิสเตอร์กำลังต่ำและกำลังสูงร่วมกัน แต่แยกวงจรจ่ายแรงดันให้กับทรานซิสเตอร์ทั้งสอง โดยที่ยังคงใช้กราวน์ร่วมกันอยู่ ข้อสังเกตในการใช้ทรานซิสเตอร์กำลังสูงกับโหลดที่ต้องการค่าแรงดันหรือกระแสมากๆ อาจต้องมีการใส่แผ่นระบายความร้อน (hearsink) ให้กับทรานซิสเตอร์กำลังด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.8 แสดงวงจรขับโหลดและทรานซิสเตอร์

ในกรณีที่โหลดเป็นอุปกรณ์ประเภทตัวเหนี่ยวนำ เช่น ขดลวดในรีเลย์, มอเตอร์ไฟฟ้า หรือกระดิ่งไฟฟ้าเมื่อมีการป้อนไฟเข้าไปมันจะทำงาน ทำให้เกิดสนามแม่เหล็กไฟฟ้าขึ้นรอบขดลวดนั้น และขณะที่ขดลวดเริ่มทำงานหรือหยุดทำงาน จะทำให้เกิดการเปลี่ยนแปลงของสนามแม่เหล็กกระทันหันขึ้นที่ขดลวด และจะเกิดปรากฏการณ์ที่เรียกว่า electromotive force (EMF) หรือเกิดการไหลย้อนกลับของแรงดันไฟฟ้านั่นเอง ซึ่งจะมีขนาดของแรงไฟที่สูงมาก อาจเป็นอันตรายต่อวงจรควบคุมได้ โดยเฉพาะตัวเพาเวอร์ทรานซิสเตอร์ที่ขับโหลดนั่นเอง จะเป็นตัวที่เสียหายก่อน เนื่องจากกระแสแรงดันที่ไหนย้อนกลับมานั้นมีทิศทางตรงกันข้ามกับกระแสและแรงดันอันเดิม จึงทำให้อุปกรณ์ที่ต่ออยู่กับขดลวดได้รับความเสียหาย

แต่ก็มีวิธีหรือทางแก้ไขได้ โดยการดิสชาร์จกระแสที่ไม่ต้องการเหล่านี้ออกไปโดยใช้ไดโอดต่อคร่อมกับโหลดประเภทตัวเหนี่ยวนำ แสดงในภาพที่ 2.5 (ข) ในทางปฏิบัติ อุปกรณ์ไฟฟ้าที่ต้องการควบคุมมีความต้องการใช้แรงดันที่สูงกว่าแรงดันที่ใช้ในภาคควบคุม เช่นใช้ไฟบ้าน 220 โวทย์ จึงมักนิยมใช้โหลดเป็นอุปกรณ์ประเภทรีเลย์ เพื่อทำหน้าที่เป็นสวิตช์เปิด/ปิด อุปกรณ์ไฟฟ้าอีกต่อหนึ่ง

2.7 การกำหนดช่องสัญญาณ

ในการกำหนดรหัสเฉพาะเพื่อใช้ควบคุมช่องสัญญาณมักใช้รหัสที่เป็นสัญญาณดิจิตอลของเลขไบนารี โดยจำนวนช่องสัญญาณหรือจำนวนฟังก์ชั่นการทำงานที่สามารถเลือกควบคุมได้ ขึ้นอยู่กับจำนวนบิตที่ใช้ กล่าวคือ หากกำหนดให้สัญญาณควบคุมมีขนาด 8 บิต จะสามารถสร้างสัญญาณควบคุมได้ถึง 2^5 หรือ 256 ช่องเลยทีเดียว โดยที่การทำงานของวงจรในแต่ละช่องสัญญาณยังคงมีหลักการในทำนองเดียวกันทุกประการ

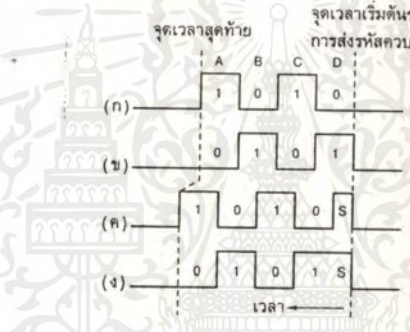
ในแต่ละบิตของสัญญาณควบคุม แสดงด้วยสัญญาณฟัลล์สำหรับค่าของบิตที่เป็น “1” และไม่มีพัลล์หรือคงสถานะต่ำไว้เมื่อค่าของบิตเป็น “0” ข้อมูลทั้งหมดตามจำนวนบิตที่ใช้มักถูกจัดเรียงเป็นลำดับอนุกรมของพัลล์ แล้วส่งไปยังเครื่องรับ ตัวอย่างเช่น หากเราออกแบบรีโมตคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของเราให้มีจำนวนบิตของสัญญาณควบคุมเป็น 4 บิต เราสามารถควบคุมอุปกรณ์ได้สูงถึง 2^4 หรือ 16 ช่องสัญญาณ โดยในแต่ละช่องสัญญาณอาจกำหนดรหัสและฟังก์ชั่นการทำงานสำหรับควบคุมโทรทัศน์ได้ ดังตารางที่ 2

รหัสการควบคุม	ฟังก์ชั่นควบคุม
0000	เปลี่ยนช่องโทรทัศน์
0001	เปลี่ยนช่องโทรทัศน์
0010	เปิดโทรทัศน์
0011	ปิดโทรทัศน์
0100	เพิ่มความดังโทรทัศน์
0101	ลดความดังโทรทัศน์
.	.
.	.
.	.
1111	เพิ่มความสว่าง

ตารางที่ 2 แสดงการกำหนดรหัสและฟังก์ชั่นการทำงาน



ภาพที่ 2.9 แสดงแผนภาพเวลาการส่งรหัส

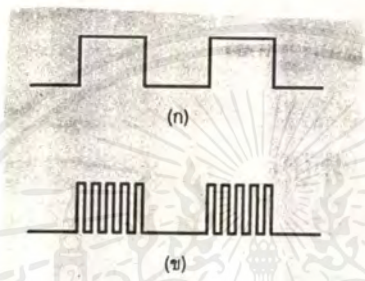
ลักษณะของรหัสสัญญาณที่กำหนด เมื่อถูกส่งออกจะจัดเรียงลำดับเป็นอนุกรมออกไป ดังเช่น รหัส 0101 และ 1010 ในภาพที่ 2.8 (ก) และ (ข) ตามลำดับ จากรูปลำดับความสำคัญเรียงจากมากที่สุดไปหาน้อยที่สุด คือ ตัดเรียงจาก บิต D, C, B และ A แต่ที่เห็นพัลส์ในรูปแสดงลำดับกลับกัน (A, B, C, D) เป็นเพราะเราคิดว่าพัลส์ด้านขวาส่งถึงเครื่องรับก่อน ข้อสังเกตของสัญญาณในภาพ 2.8 (ก) และ (ข) แม้ว่าเกิดจากรหัสที่แตกต่างกัน เครื่องรับอาจมีการผิดพลาดในการตีความได้ หากช่วงเวลาของการส่งรหัสควบคุม ซึ่งกำหนดด้วยเส้นประที่แสดงจุดเริ่มต้น และจุดสุดท้ายเวลาของการส่งข้อความเกิดความคลาดเคลื่อนไป อาจส่งผลให้เครื่องรับตีความหมายของภาพ (ก) สลับกับภาพ (ข) ได้ จึงจำเป็นต้องมีการส่งสัญญาณซิงโครนัส หรือสัญญาณที่บอกให้ทราบจุดเวลาเริ่มต้นที่แน่นอน รหัสสัญญาณควบคุมไปพร้อมกัน โดยทั่วไปมักใช้พัลส์ที่แสดงสภาวะสูงเป็นตัวบอกจุดเริ่มต้นของเวลาในการส่งสัญญาณ ดังแสดงในภาพที่ 2.8 (ค) และ (ง) อันเป็นรูปแบบของรหัสจากรูป (ก) และ (ข) ตามลำดับ เมื่อมีสัญญาณพัลส์เริ่มต้น (S) ขนาดความกว้างเท่ากับครึ่งลูกคลื่นของบิตปกติเป็นตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดเวลาซิงโครนัส ขนาดความกว้างของพัลส์ S ไม่มีข้อจำกัดที่แน่นอน แต่ขึ้นอยู่กับการเลือกและออกแบบของเราเอง

2.8 สัญญาณโทนเบิรสต์ (Toneburst)

ลักษณะสัญญาณแบบโทนเบริสต์ แสดงดังภาพที่ 2.10 (ข) ซึ่งสร้างจากสัญญาณพัลส์ที่มีสภาวะสูงของสัญญาณควบคุมแบบธรรมดาในภาพที่ 2.10 (ก) สัญญาณโทรเบริสต์ ประกอบด้วยพัลส์ความถี่สูงแบบต่อเนื่องตลอดช่วงความกว้างของบิตที่เป็น “1” ในขณะที่บิตข้อมูลอยู่ในสภาวะต่ำ สัญญาณจะคงเดิมไม่มีการเปลี่ยนแปลงแต่อย่างใด



ภาพที่ 2.10 แสดงสัญญาณพัลส์ที่มีสภาวะสูง

สำหรับวงจรรับต้องทำการออกแบบให้มีการตอบสนองต่อสัญญาณโทนเบริสต์ในช่วงความถี่ที่เหมาะสมกัน จึงสามารถออกแบบชุดรีโมทคอนโทรล ที่มีค่าความถี่ของโทนเบริสต์แตกต่างกันเพื่อนำมาควบคุมอุปกรณ์ในพื้นที่ใกล้เคียงกันได้ โดยปราศจากการรบกวนซึ่งกันและกัน

ในขณะเดียวกัน ตัววงจรส่งอาจถูกออกแบบให้มีการเปลี่ยนแปลงค่าความถี่ของสัญญาณโทนเบิรสต์เป็นค่าต่างๆ ซึ่งช่วยให้สามารถใช้เครื่องส่งเพียงตัวเดียว สำหรับการควบคุมอุปกรณ์ไฟฟ้าหลายชนิดได้พร้อมกัน

ข้อดีอีกประการของการส่งสัญญาณแบบโทนเบิรสต์ ก็คือ สามารถขจัดสัญญาณรบกวนจากภายนอกได้ดีมาก ตัวอย่างเช่นระบบควบคุมที่ใช้สัญญาณแสง หรือแสงอินฟราเรดที่ส่งด้วยพัลส์ธรรมดา อาจมีแสงจากภายนอก แสงจากหลอดไฟ หรือแม้กระทั่งแสงอาทตย์เข้าไปรบกวนที่เครื่องรับจนอาจเกิดข้อผิดพลาดในการรับสัญญาณควบคุมได้ แต่ถ้าหากมีการใช้สัญญาณโทนเบิรสต์แล้ว สิ่งรบกวนต่างๆ เหล่านี้ จะถูกขจัดออกไปโดยสิ้นเชิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

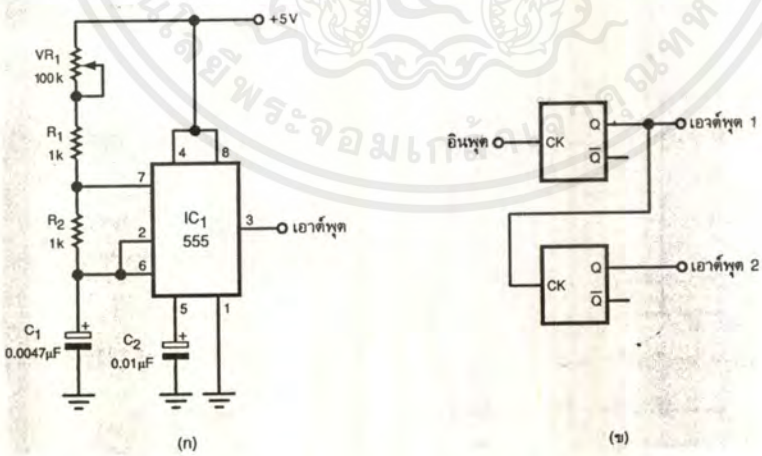
2.8.1 การควบคุมด้วยสัญญาณโทนเบิรสต์

ในการควบคุมอุปกรณ์ต่างชนิด ที่มีชุดเครื่องรับส่งแยกเฉพาะในแต่ละอุปกรณ์และอยู่ในบริเวณเดียวกัน สัญญาณควบคุมอาจเกิดการรบกวนซึ่งกันและกันได้ ถ้าหากระบบควบคุมมีลักษณะการทำงานในทำนองเดียวกัน เช่น มีรูปแบบการเข้ารหัสเหมือนกัน เป็นต้น ในกรณีเช่นนี้หากเราใช้คลื่นวิทยุ เป็นตัวส่งสัญญาณ เราสามารถกำหนดความถี่ของชุดรับส่งในแต่ละชุดให้มีค่าความถี่แตกต่างกัน เพื่อให้สามารถนำมาใช้ภายในพื้นที่เดียวกันได้ โดยไม่เกิดการรบกวนกัน

แต่ถ้าหากเป็นการส่งสัญญาณแสงอินฟราเรดแล้วต้องมีการดัดแปลงรูปแบบของสัญญาณให้มีลักษณะคล้ายกับการส่งคลื่นวิทยุ โดยใช้การส่งสัญญาณแบบ โทนเบ่ริสต์แทนการส่งสัญญาณแบบพัลส์ธรรมดา

2.8.2 วงจรส่งสัญญาณโทนเบิรสต์

เนื่องจากสิ่งที่ถูกเพิ่มเข้ามาจากสัญญาณพัลส์ธรรมดา ก็คือ พัลส์ความถี่สูง ดังนั้นวงจรส่งสัญญาณควบคุมจึงต้องเพิ่มภาคกำเนิดสัญญาณความถี่สูงเข้ามาด้วย วงจรกำเนิดสัญญาณอย่างง่ายๆ อาจใช้ไอซียอดนิยม เบอร์ 555 ต่อเป็นวงจรชนิดอะสเตเบิลมัลติไวเบรเตอร์ ดังภาพที่ 12.11 การปรับค่าความถี่สัญญาณ ทำได้โดยการปรับค่าของ V_{R1} ในวงจรตัวอย่างนี้ ช่วงของการเปลี่ยนแปลงค่าความถี่ อยู่ระหว่าง 300 เฮิรตซ์ -10 กิโลเฮิรตซ์ ซึ่งกว้างพอที่จะเลือกค่าความถี่สำหรับรีโมตคอนโทรลได้มากมายหลายชุด

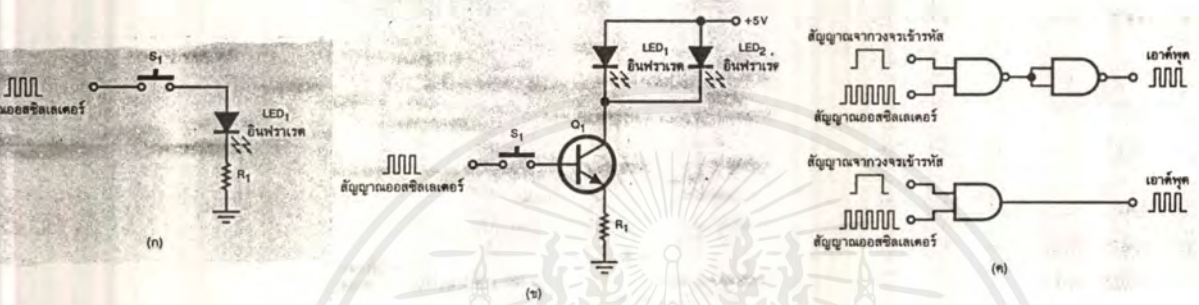


ภาพที่ 2.11 แสดงวงจรชนิดอะสเตเบิลมัลติไวเบรเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกเหนือจากการปรับค่าความถี่ด้วย V_{R1} แล้วอาจเพิ่มส่วนของวงจรหาความถี่ที่สัญญาณเอาต์พุท เพื่อลดค่าความถี่ลงมาเป็นความถี่ใช้งานอื่นอีกได้ วงจรหารความถี่อย่างง่ายๆ อาจได้แก่ วงจรฟลิปฟลอป ดังแสดงในภาพที่ 2.11 (ข)

สัญญาณเอาต์พุทความถี่สูงจากไอซี 555 สามารถนำไปขับ LED ชนิดอินฟราเรดเพื่อส่งสัญญาณควบคุมได้เลย หากต้องการส่งเพียงบิตเดียวออกไป โดยต่อกับวงจรในภาพที่ 2.12 (ก) หรือหากต้องการเพิ่มกำลังส่งให้ไปได้ไกลขึ้น อาจใช้ทรานซิสเตอร์เป็นตัวขับสัญญาณให้กับ LED 2 ตัว ดังภาพที่ 2.12 (ข)



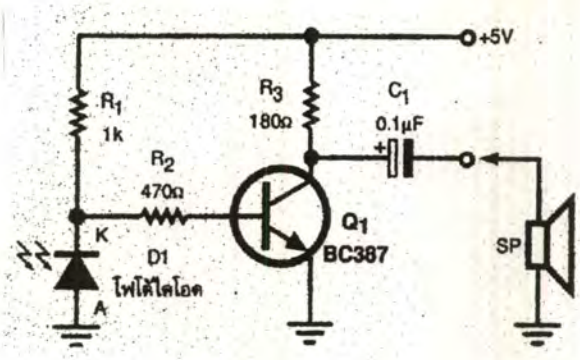
ภาพที่ 2.12 แสดงการนำสัญญาณความถี่ไปขับLED ชนิดอินฟราเรด

ในกรณีที่สัญญาณควบคุมเป็นลักษณะของอนุกรมพัลส์ที่มีการจัดรหัสควบคุม ดังที่ได้กล่าวไว้แล้วในเรื่องสัญญาณอนุกรมพัลส์ การส่งสัญญาณแบบโทนเบิรสต์ สามารถทำได้โดยการรวมสัญญาณระหว่างอนุกรมพัลส์ความถี่สูง (จากเอาต์พุท ไอซี 555) เข้าด้วยกัน ด้วยวงจรแอนด์เกตหรือวงจรแนนด์เกต ดังแสดงในภาพที่ 2.12 (ค)

2.9 วงจรรับอย่างง่าย

การรับสัญญาณโทนเบิรสต์ที่ส่งมากับสัญญาณแสง อาจทำได้ด้วยวงจรง่ายๆ ซึ่งประกอบด้วยโฟโตโอดต่ออนุกรมกับตัวต้านทาน R_1 เท่านั้น ก็จะได้สัญญาณเอาต์พุทที่มีลักษณะเช่นเดียวกับภาคส่งออกมาที่จุด A หากสัญญาณพัลส์ความถี่สูงที่ถูกส่งมาที่มีค่าความถี่ในช่วงของความถี่เสียงมนุษย์ที่ได้ยิน เช่น ระหว่าง 100 เฮิรตซ์ ถึง 20000 เฮิรตซ์ อาจใช้หูฟังขนาดจิ๋วมาต่อคร่อมจุด A กับกราวน์เพื่อรับฟังสัญญาณได้เลย แต่สัญญาณที่ได้จะมีความดังต่ำเพราะขนาดสัญญาณอ่อน จึงต้องใช้ วงจรทรานซิสเตอร์ต่อเพิ่ม เพื่อขยายสัญญาณให้สามารถรับฟังได้อย่างชัดเจนในภาพที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.13 แสดงวงจรรับอย่างง่าย

2.10 ตัวเข้ารหัสแป้นกดข้อมูล (KEY BOARD ENCODER)

บทบาทขั้นพื้นฐานของตัวเข้าหรัสแป้นกดข้อมูลก็คือจะเป็นตัวให้กำเนิดข้อมูล 8 บิทออกมา หลังจากที่มีการกดปุ่มกดและขณะเดียวกันชิพตัวนี้จะทำหน้าที่ดีเบานส์และป้องกันการเกิด โรว์โอเวอร์ อีกด้วย ตัวเข้ารหัสที่สำคัญมีอยู่ 3 ชนิด คือ

2.10.1 ตัวเข้ารหัสแบบสเตติกส์ (Static Encoder)

2.10.2 ตัวเข้ารหัสแบบสแกนนิ่งค์ (Scanning Encoder)

2.10.3 ตัวเข้ารหัสแบบคอนเวอร์ติ้งค์ (Converting Encoder)

2.10.1 ตัวเข้ารหัสแบบสเตติกส์ เป็นแป้นกดข้อมูลชนิดท่จะให้กำเนิดรหัสที่สอดคล้องกับปุ่มกดที่กดซึ่งทำให้ง่ายต่อการแก้ปัญหาตัวอย่างก็คือแปนกดข้อมูลลักษณะสายตรง(Linear Keyboard) ถ้ามี 64 ปุ่มกดก็จะต้องมีสายสำหรับทุกๆ ปุ่มกดซึ่งจะทำให้รับรู้ได้ง่าย เพราะจะเกิดพัลส์ที่เกิดขึ้นนี้จะถูกเปลี่ยนรูปให้อยู่ในรูปแมททริกส์ซึ่งสามารถจัดให้อยู่ในขนาด 8X8 ได้ จึงสามารถลดจำนวนสายให้เหลือแค่ 16 เส้นเท่านั้น ราคาของแป้นกดข้อมูลก็ถูกลงกว่าเดิม ตัวอย่างเช่น ตัวเข้ารหัสแบบสเตติกส์

2.10.2 ตัวเข้ารหัสแบบสแกนนิ่งค์

สแกนนิ่งค์ชิฟ (Scanning Chip) จะเป็นตัวช่วยแก้ปัญหาในการที่จะระบุว่าปุ่มกดตัวใดที่ถูกกดเข้ามา เมื่อใช้แป้นกดข้อมูลที่จัดในรูปแมททริกส์ แต่ละแถวของปุ่มกดจะถูกสแกนโดยตัวนับ (Counter) หลังจากทำการสแกนแล้งถ้ายังไม่มีการกดปุ่มการสแกนจะทำไปเรื่อยๆ จนครบรอบเป็นวงกลมกลับมายังตำแหน่งเดิมทำแบบนี้หมุนเวียนไปเรื่อย ๆ ถ้ามีการกดปุ่ม จังหวะนั้นวงจะสร้างสัญญาณสโตป (Strobe) ขึ้นและการสแกนจะหยุดลงตัวนับจะหยุดนับดังนั้นค่าที่นับได้จะเป็นค่าที่บอกตำแหน่งของแถวและคอลัมของตำแหน่งของปุ่มที่ถูกกดนั้น

การสแกนแบบเดินหน้านี้ไม่อาจจะป้องกันการกดปุ่มทีเดียวสองปุ่มพร้อมกันได้ เนื่องจากการสแกนจะหยุดถ้ามีการกดปุ่มตัวแรกลงดังนั้นถ้าจะกดปุ่มสองปุ่มควรเว้นระยะห่างกันพอจะทำให้ข้อมูลที่เข้าไปถูกต้อง เมื่อไรก็ตามถ้ามีการกดปุ่มพร้อม ๆ กันมากกว่าหนึ่งปุ่มในเวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ค่านั้นจะถือว่าเป็นค่าที่ไม่ถูกต้อง (Error) มันจะไม่หยุดการสแกนจนกว่าจะมีการกดปุ่มเพียงปุ่มเดียวเท่านั้นจึงหยุดสแกนสำหรับข้อดีของวิธียังช่วยในการดีเบานส์โดยอัตโนมัติอีกด้วย จะเห็นว่าตัวนับจะมี 6 บิท โดยจะแบ่งบิท 0,1,2 มาผ่านตัวถอดรหสเป็น 8 เส้นเพื่อทำการสแกนแถว 3,4,5 ของตัวนับมาถอดรหัสอกมาเป็น 8 คอลัม เท่ากัน แต่จะเปลี่ยนแปลงช้ากว่า 3 บิทแรก (ทางด้านแถว) ดังนั้น ถ้ามีการกดที่คอลัม 2 แถว 2 ทางด้านคอลัมจะถูกเลือกต่อเมื่อบิท 5,4,3 เป็น 010 ขณะเดียวกัน บิท 0,1,2 จะต้องเป็น 010 เมื่อมีการกดปุ่มและนับตัวนับได้นับมาถึงตำแหน่งของปุ่มนั้น วงจรภายในแป้นกดข้อมูลจะผลิตพัลส์สัญญาณสโตป ขึ้นเพื่อไปใช้ในการหยุดการนับของตัวนับ ขณะเดียวกันรหัสของเอาท์พุทของตัวนับนี้จะส่งไปที่รอม ซึ่งเป็นตัวถอดรหัสให้เป็นรหัสปุ่มกดที่ต้องการสัญญาณสโตปจะส่งผ่านไปแลท์ค่ารหัสของปุ่มเพื่อส่งเข้าเครื่องคอมพิวเตอร์อีกทีหนึ่งแสดงรูปของรอมและแลทช์

2.11 เบานส์ (Bounce)

เป็นตัวที่ทำให้เกิดปัญหาอย่างมาก ซึ่งจะเกิดชั่วขณะหนึ่งประมาณ 10 ms ทุกครั้งที่มีการสัมผัส (Close) หรือจากกัน (Open) ของหน้าสัมผัส (Contact) ซึ่งเป็นกลไกลของสวิทส์ที่ใช้ทำเป็นปุ่มกด เบานส์ยังเป็นตัวที่ทำให้เกิดปัญหาเป็นอย่างมากสำหรับเครื่องคอมพิวเตอร์ที่จะทำการตรวจแยกว่าสัญญาณที่ที่เข้ามานั้นเบานส์หรือเป็นระดับสัญญาณที่เกิดจากการกดปุ่ม ดังนั้นเพื่อให้คอมพิวเตอร์ได้รับสัญญาณที่ถูกต้องจึงต้องทำการแก้ปัญหาเรื่องเบานส์ก่อนที่จะส่งสัญญาณไฟฟ้าที่เป็นระหัสที่ถูกต้องให้กับเครื่องคอมพิวเตอร์เพื่อใช้งานขั้นต่อไป

การแก้ปัญหาเบานส์เรียกว่าดีเบานส์ (Debounce) โดยทั่วไปสามารถทำได้ 2 วิธี คือ

ก) ใช้วิธีการทางฮาร์ดแวร์ โดยการต่ออุปกรณ์เป็นวงจรเพื่อกำจัดเบานส์กำจัดเบานส์ให้หมดไป วิธีง่ายๆได้แก่ การต่อวงจรกรอง RC (RC filter) ที่บริเวณสัมผัสของปุ่มกดทุกปุ่มหรือก็ไม่ใช้วิธีต่อวงจรฟลิบฟล็อป อย่างเช่นในรูป ก็ได้ วิธีนี้ค่อนข้างยุ่งอยากเพราะต้องมีวงจรฮาร์ดแว์เพิ่มขึ้นเหมาะสำหรับแป้นกดข้อมูลที่มีปุ่มกดไม่กี่ปุ่ม แต่ไม่เหมาะที่จะใช้กับแป้นกดข้อมูลที่มีปุ่มกดหลายเพราะจะต้องมีวงจรลักษณะนี้เพิ่มขึ้นเท่ากับจำนวนปุ่มซึ่งทำให้เปลื่องค่าใช้จ่ายและใช้เนื้อที่เพิ่มขึ้น

ข) ใช้หลักการหน่วงเวลา (Delay) การตรวจสอบสภาพการใช้ซอฟท์แวร์ วิธีนี้จะหน่วงเวลาประมาณ 10-20ms และคอยตรวจสอบสเตตัสหลังการกดปุ่มด้วยเทคนิคทางซอฟท์แวร์วิธีนี้เหมาะสำหรับแป้นกดข้อมูลที่มีปุ่มกดจำนวนมากปุ่ม

2.12 โรว์โอเวอร์ (Rollover)

โรว์โอเวอร์ เป็นปัญหาที่เกิดขึ้นกรณีที่บังเอิญผู้ใช้กดปุ่มของแป้นกดข้อมูลสองปุ่มหรือมากกว่าในเวลาเดียวกัน ทำให้รหัสที่ส่งเข้าเครื่องคอมพิวเตอร์ผิดพลาดได้ ดังนั้นจำเป็นต้องหาวิธีป้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

33928

กันไม่ให้เกิดปัญหาเช่นนี้ขึ้นเพื่อให้คอมพิวเตอร์รับรหัสของปุ่มที่ผู้ใช้ต้องการจะกดเพียงค่าเดียวเท่านั้น วิธีการในการป้องกันมีอยู่ด้วยกัน 3 วิธีด้วยกันคือ

ก) Two-key rollover เป็นการป้องกันการกดปุ่ม 2 ปุ่ม ในขณะเดียวกันมีวิธีที่นิยมใช้กันอยู่ 2 วิธีคือ อย่างที่ง่ายที่สุดจะไม่มีการอ่านค่าจากแป้นกดข้อมูลในขณะที่มีการกดสองปุ่มพร้อมกันจนกว่าจะปล่อยตัวใดตัวหนึ่งก่อน จะอ่านค่าสองปุ่มสุดท้ายที่เหลืออยู่เท่านั้นวิธีนี้มักนิยมใช้กรณีใช้ซอฟท์แวร์สแกน (Scan) และถอดรหัสแป้นกดข้อมูลเท่านั้น อีกวิธีก็คือการใช้วงจรทางฮาร์ดแวร์ในการป้องกันไม่ให้ปุ่มที่อยู่ข้างเคียงปุ่มที่ถูกกดสามารถผลิตสัญญาณที่ส่งเข้าสู่เครื่องคอมพิวเตอร์ได้ หลังจากที่กดตัวแรกไปแล้วแล้วกดปุ่มที่สองตามมาในขณะที่ยังไม่ปล่อยมือจากตัวแรก การกดปุ่มที่สองนั้นจะไม่มีผลสัญญาณจากปุ่มที่สองจะไม่สามารถจะส่งเข้าเครื่องจนกว่าจะปล่อยมือจากปุ่มตัวแรกเสียก่อนวงจรที่ทำหน้าที่นี้จะประกอบด้วยตัวหน่วงเวลาและแลทช์

ข) N-key rollover วิธีนี้จะไม่ยอมรับข้อมูลที่เกิดจากการกดปุ่มหลายปุ่มพร้อมกันจนกว่าจะปล่อยมือเหลือปุ่มที่กดเพียงปุ่มเดียววิธีนี้คล้ายคลึงกับกรณี Two-key rollover เพียงแต่จำนวนการป้องกันปุ่มที่กดพร้อมกันมีมากกว่าสองเท่านั้น วิธีนี้มักจะใช้กับแป้นที่กดข้อมูลสองระบบที่มีราคาแพง

ค) N-key lockout วิธีนี้จะยอมรับการกดหรือปล่อยมือของปุ่มเพียงปุ่มเดียวเท่านั้น ปุ่มอื่นที่กดหรือปล่อยในเวลาใกล้เคียงกันจะไม่มีผล วิธีที่ง่ายและนิยมใช้กันมากเพียงแต่มีข้อเสียตรงที่ไม่สามารถทำการกดปุ่มได้รวดเร็วเท่านั้น เพราะสัญญาณจากปุ่มที่กดตามมาจะมีผลต่อเมื่อปล่อยมือจากปุ่มแรกแล้วเท่านั้น

2.13 ชนิดของแป้นข้อมูล

2.13.1 แป้นกดข้อมูลชนิดไม่เข้ารหัส (NON-ENCODER KEYBOARD)

แป้นกดข้อมูลแบบนี้จะใช้ซอฟท์แวร์ในการช่วยเข้ารหัสตัวเลขและตัวอักษรต่างๆบนแป้นกดแทนการใช้ตัวเข้ารหัสทางฮาร์ดแวร์แต่จะใช้ LSI พิเศษบางตัวช่วย เช่น PIA (8255) เป็นต้น โดยปกติแป้นกดข้อมูลแบบนี้จะจัดการเรียงปุ่มกดบนแป้นกดให้เป็นแถวและคอลัมซึ่งจะกำหนดให้มีขนาด N X M และจะใช้หลักการสแกนที่ละเส้น ทั้งทางด้านแถวและคอลัมคล้ายกับว่าเดินสำรวจทีละแถว

2.13.2 แป้นกดข้อมูลชนิดเข้ารหัส (ENCODER KEYBOARD)

น้อยคนนักที่ยอมลำบากในการเขียนโปรแกรมเพื่อช่วยในการเข้ารหัสแป้นกดข้อมูลโดยวิธีแป้นกดข้อมูลชนิดไม่เข้ารหัส ยังมีวิธีที่สะดวกกว่าก็คือให้แป้นกดข้อมูลแบบเข้ารหัสซึ่งแบบนี้จะใช้ฮาร์ดแวร์เป็นตัวช่วยเข้ารหัสตัวเลขหรืออักษรต่างๆ บนแป้นกดซึ่งเครื่องคอมพิวเตอร์จะสามารถรับรู้ได้ว่าปุ่มที่กดเป็นตัวเลขหรืออักษรอะไร และรหัสนั้นจะถูก Hold ไว้ชั่วขณะหนึ่งจนกว่าจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกดตัวใหม่เข้ามาแป้นกดข้อมูลแบบนี้จะถูกทำในรูปของ LSI โดยจะใช้การสแกนแบบแมททริกส์ซึ่งจมีวงจนดีเบานส์และการป้องกันโรว์โอเวอร์ รวมทั้งแลทช์สำรหับเก็บข้อมูลขณะกดไว้ใช้งานรวามอยู่ในชิพตัวเดียวกัน ชิพบางตัวจะมีรอมรวมอยู่ด้วยไว้สำหรับเป็นตารางเพื่อที่จะจ่ายรหัสที่ตรงกับปุ่มกดที่กดเข้าไปซึ่งอาจจะเป็น ASCII หรือ EBEDIC ก็ได้ขึ้นอยู่กับการโปรแกรมในรอมนั้น

2.13.3 แป้นกดข้อมูลรหัส ASCII

แป้นกดข้อมูลแบบนี้จะให้กำเนิดสัญญาณรหัส 7 บิท ASCII แป้นกดข้อมูลลักษณะนี้อาจใช้ LSI Key board Controller chip ทำหน้าที่จ่ายรหัส 7 บิท รวมทั้งสัญญาณ STROBE เพื่อใช้อินเตอร์เฟสร่วมกัน UART ก็ได้ ซึ่งวงจรอินเตอร์เฟสแสดงในรูป 2.23 จากรูปจะเห็นว่าถ้ามีการกดปุ่มกดจะมีสัญญาณสโตปมาควบคุม UART เพื่อรับข้อมูล ASCII 7 บิท แบบขนานเข้ามา UART จะเข้าจะแปลงสัญญาณเป็นแบบอนุกรม ซึ่งมีรูปแบบเป็น 10 –11 บิท ขณะที่ UART ทำการส่งข้อมูลอยู่นั้นแม้ที่แป้นกดข้อมูลจะไม่การรับข้อมูลหใหม่เข้ามา คือถ้ามีการกดก็จะไม่รับรู้ จากวงจรเราสามารถกำหนดอัตราเร็วในการส่งข้อมูลต่อบิทได้ในหนึ่งวินาที ในกรส่งแต่ละบิทจะใช้ 16 CLOCK CYCLE ดังนั้น ถ้าส่ง 110 BAUD จะใช้ต้องปรับให้ OSCILLATOR(555) มีความถี่ 1760 Hz ถ้าส่ง 300 BAUD ก็จะต้องใช้ 4800 Hz สำหรับรหัส 7 บิท

2.14 ไมโครคอนโทรลเลอร์ MCS-51

2.14.1 โครงสร้างสถาปัตยกรรม MCS-51

ลักษณะหลักทั่ว ๆ ไปของ MCS-51 จะประกอบด้วย

1 ใช้ HMOS และ CHMOS เทคโนโลยีการสร้างและทำงานด้วยแหล่งจ่ายไฟขนาด 5 V. เพียงแหล่งเดียว
2 ซีพียูมีขนาดคำ 8 บิต
3 มีวงจรออสซิลเลเตอร์ และวงจรนาฬิกาบนชิป
4 ชุดแบงค์ (Bank) รีจิสเตอร์มี 4 ชุด แต่ละชุดมีรีจิสเตอร์ 8 ตัว ทำงานเช่นเดียวกับ MCS-48
5 มีตัวจับเวลา/ตัวนับ ขนาด 16 บิต 2 ชุด และสำหรับเบอร์ 8032/8052 มี 3 ชุด
6 มีพอร์ตไอโอแบบขนานสองทิศทางจำนวน 4 พอร์ต พอร์ตละ 8 บิต รวมทั้งหมดเป็น 32 เส้น แต่จะเหลือเพียง 16 เส้น สำหรับเบอร์ 8031 อีก 16 เส้นจะใช้ในการเข้าถึงทางแอดเดรสและข้อมูล
7 พอร์ตแบบอนุกรมสามารถที่จะโปรแกรมการรับส่งแบบ Full Duplex ที่ความเร็วสูง
8 หนึ่งวัฏจักรคำสั่งจะกินเวลา 1 ไมโครวินาที ด้วยการใช้คริสทัล 12 เม็กกะเฮิรตซ์
9 แอดเดรสข้อมูลภายนอกได้ 64 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10 แอดเดรสโปรแกรมภายนอกได้ 64 กิโลไบต์

11 สามารถกำหนดเลขที่อยู่ข้อมูลขนาดไบต์หรือบิตได้โดยตรง

12 มีซอฟต์แวร์บิตแฟลกสำหรับผู้ใช้ที่จะกำหนดเองได้ถึง 128 ตำแหน่งบิต

13 โครงสร้างอินเตอร์รัพต์จะติดตั้งได้ถึง 5 แหล่ง และ 6 แหล่ง สำหรับ8032/8052 พร้อมด้วย การจัดไพรโอริตี้ (Priority) ได้ 2 ระดับ

14 ตัวโพรเซสเซอร์สามารถใช้งานแบบบูลีน (Boolean) ได้ สำหรับใช้กับกระบวนการงานควบคุม

15 มีคำสั่งคูณ และหารทางฮาร์ดแวร์ที่ทำได้ภายใน 4 ไมโครวินาที

16 ตัวเลขทางคณิตศาสตร์ ใช้ได้ทั้งระบบไบนารี และเดซิมัล

17 การใช้พื้นที่สแตกสำหรับโปรแกรมย่อยต่าง ๆ ทำได้กว้างกว่า MCS-48

18 ชุดคำสั่งของ MCS-51 จะมีความสามารถสูงกว่าคำสั่งของ MCS-48ตระกูล MCS-51 จะทั้งแบบรอม ในตัว หรือไม่มีรอม หรือมี EPROM บนชิปเดียวกันและจะมีตำแหน่งขาที่เหมือนกัน

2.14.2 การต่อเชื่อมแบบอนุกรม

พอร์ตอนุกรมเป็นแบบ Full Duplex สามารถที่จะส่งและรับข้อมูลพร้อมกันได้ โดยทำหน้าที่เป็นบัฟเฟอร์การรับ หมายถึง พอร์ตสามารถที่จะรับไบต์ที่สอง ก่อนที่ตัวแรกจะถูกรับไปจากเรจิสเตอร์ตัวรับ อย่างไรก็ตาม ไบต์ตัวแรกจะต้องถูกอ่านไปก่อนที่ช่วงเวลาการับไบต์ตัวที่สองจะสิ้นสุด มิฉะนั้นไบต์ตัวแรก จะถูกซ้อนและสูญหายไปได้ ในพอร์ตอนุกรมเรจิสเตอร์ทั้งสองจะแยกกันอยู่ก็ตาม

พอร์ตอนุกรมสามารถที่จะเลือกทำงานในโหมดต่าง ๆ ได้สี่โหมด

โหมด 0 : ข้อมูลจะเข้าและออกผ่าน RXD TXD ด้วยการเลื่อนสัญญานาฬิกาเอาต์พุต ข้อมูลจะเป็นลักษณะแปดบิต ในการับและส่งแต่ละครั้ง โดยที่ส่งค่า LSB ก่อนอัตราบ็อดจะคงที่ที่ 1/12 ของความถี่ออสเลเตอร์

โหมด 1 : จะเป็นการส่งข้อมูลขนาด 10 บิต ผ่านออก TXD หรือรับเข้ามาผ่าน RXD โดยรูปแบบบิตจะประกอบด้วย หนึ่งบิต Start เป็น '0' แปดบิตข้อมูลโดย LSB เป็นตัวแรกที่รับและส่งข้อมูลนี้ และอีกหนึ่งบิต Stop มีค่า '1' การรับบิต Stop จะนำไปเก็บที่บิต RB8 ของ SFR เรจิสเตอร์ SCON อัตราบ็อดแปรผันได้ตามการตั้งตัวจับเวลาซึ่งจะกล่าวในหัวข้อต่อไป

โหมด 2 : เป็นการส่งข้อมูลขนาด 11 บิต ผ่านออกขา TXD หรือรับเข้ามาผ่านขา RXD ประกอบด้วย หนึ่งบิต Start มีค่า '0' แปดบิตข้อมูลโดย LSB เป็นตัวแรกที่รับและส่งข้อมูล บิตที่ เก้าที่อยู่ในบิต TB8 ของเรจิสเตอร์ SCON สามารถที่จะกำหนดเลือกเป็น '1' หรือ '0' ได้ตัวอย่าง เช่น การใช้งานเป็นบิตพาริตี้ โดยการเลื่อนเอาบิต P ของ PSW มาไว้ใน TB8 เพื่อเป็นการส่งข้อมูลแบบมีการตรวจพาริตี้ของข้อมูลที่ส่ง ในการรับข้อมูลบิตที่เก้าจะเข้าไปเก็บที่ RB8 ใน SFR เรจิสเตอร์ SCON ขณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่บิต Stop จะไปรับเข้ามาเก็บ อัตราบ็อดสามารถเลือกเป็น 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ SCON เป็น SFR ที่ใช้ในการติดตั้งโหมดการทำงานของพอร์ตอนุกรม เช่น การกำหนดค่า RB8 จะเป็นการใช้ตัวรับบิตที่เก้าด้วยหรือไม่ เป็นต้น ตารางที่ 3 เป็นตารางการใช้ SCON ในการควบคุมการทำงานของพอร์ตอนุกรม

โหมด 3 : เป็นการส่งข้อมูลขนาด 11 บิต ผ่านออกขา TXD หรือรับเข้ามาผ่านขา RXD ประกอบด้วยบิตที่เก้าของข้อมูลสามารถที่จะโปรแกรมเลือกได้ และบิต Stop ค่า 1 อีกหนึ่งบิต ในความเป็นจริงโหมด 3 จะคล้ายกับโหมด 2 ทุกประการ ยกเว้นอัตราบ็อด โดยอัตราบ็อดในโหมด 3 จะแปรผันได้ไปตามการโปรแกรมการเลือกตัวจับเวลา

ทั้งสี่โหมดนี้ การส่งข้อมูลจะเริ่มติดตั้ง Initiated ด้วยคำสั่งใด ๆ ที่ใช้ตัวเรจิสเตอร์ SBUF เป็นเรจิสเตอร์ตัวรับข้อมูลจากซีพียู และในโหมด 0 การรับข้อมูลเริ่มติดตั้งด้วยการใช้สถานะ RI = 0และ REN = 1 ส่วนในโหมดอื่นการรับข้อมูลจะเริ่มติดตั้งด้วยการรับบิต Start เข้ามาตรวจสอบถ้า REN = 1

การทำงานทั้งสี่โหมดของพอร์ตอนุกรมได้สรุปไว้ในตารางที่ 3 ต่อไปนี้เป็นการทำงานโดยละเอียดในโหมดต่าง ๆ ของพอร์ตอนุกรม

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

โดย SM0 , SM1 เป็นตัวกำหนดใช้โหมดต่าง ๆ ของพอร์ตอนุกรม ดังนี้

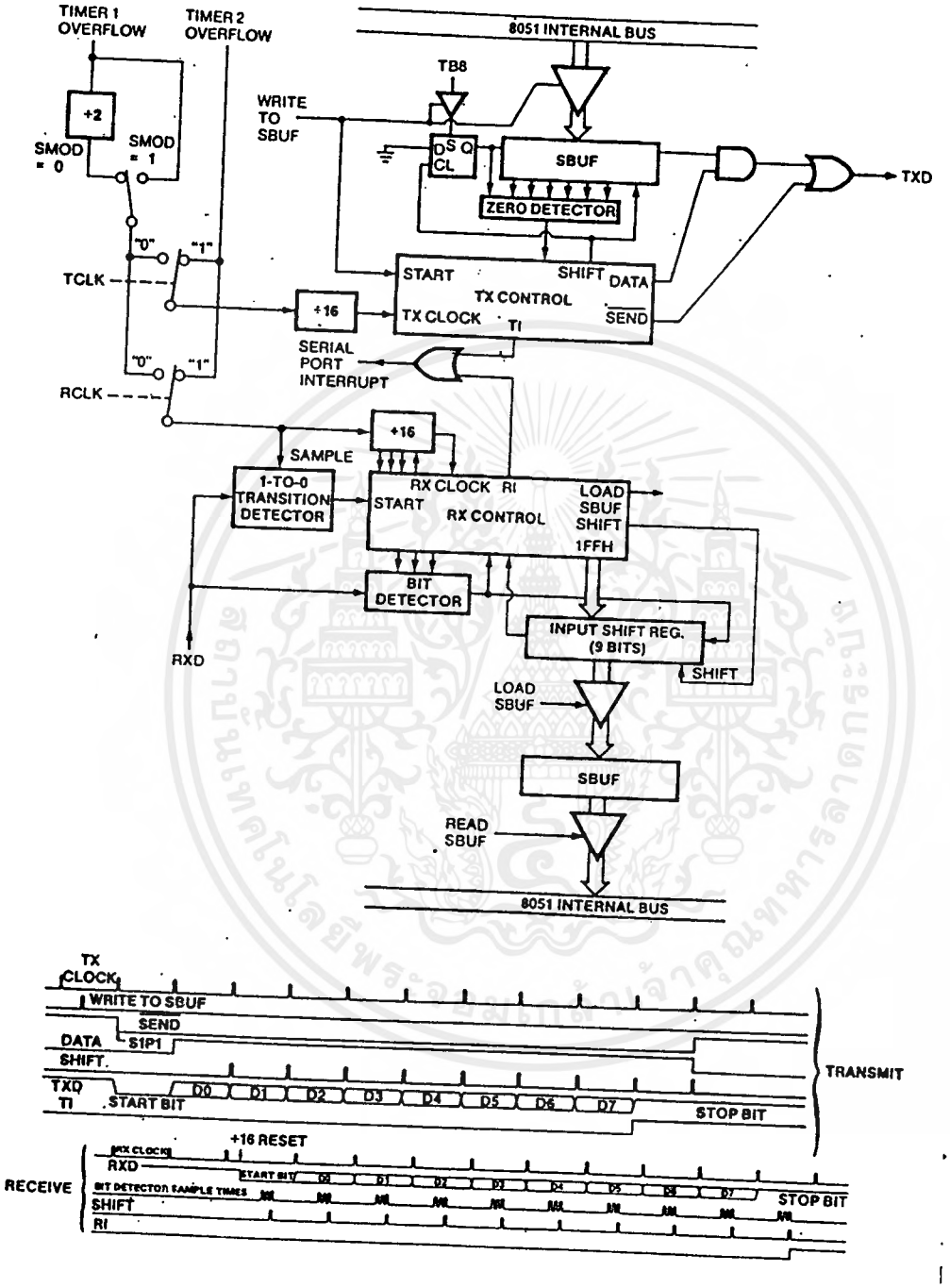
SM0	SM1	โหมด	ลักษณะการทำงาน อัตราบ็อด
0	0	0	เลื่อนเรจิสเตอร์ fosc/12
0	1	1	8-บิต UART แปรผันได้ตามการเลือกตัวจับเวลา
1	0	2	9-บิต UART fosc/64 หรือ fosc/32
1	1	3	9-บิต UART แปรผัน

* UART : Universal Asynchronous Receiver / Tranceiver

SM2	ควบคุมอีนาเบิ้ล การใช้โพรเซสเซอร์หลายตัวในการสื่อสารซึ่งกันและกันในโหมด 2 และ 3 ถ้า SM2 เซตเป็น 1 ดังนั้น RI จะต้องไม่แอกทีฟ ถ้ามีการรับบิตที่เก้า ทำให้บิต RB8 นี้ เป็น 0ในโหมด 1 ถ้า SM2 เซตเป็น 1 ดังนั้น RI จะไม่แอกทีฟถ้า STOP บิตไม่ถูกรับเข้ามาในโหมด 0 SM2 ควรมีค่าเท่ากับ 0
REN	ตัวอีนาเบิ้ลอนุกรมการรับ เซตเป็น '1' ด้วยโปรแกรมในการเลือกอีนาเบิ้ลการรับและเป็น '0' ด้วยโปรแกรม เมื่อให้เป็นดิสเอเบิ้ลการรับ
TB8	เป็นข้อมูลบิตที่เก้า ซึ่งจะถูกส่งในโหมด 2 และ 3 ซึ่งจะให้เป็น '1' หรือ '0'ได้ด้วยการโปรแกรม
RB8	ในโหมด 2 และ 3 ข้อมูลบิตที่เก้าจะถูกรับไปในโหมด 1 ถ้า SM2 = 0 RB8 จะกลายเป็น STOP บิตที่ถูกรับไปในโหมด 0 RB8 ไม่ใช้
TI	เป็นแฟลกอินเตอร์รัพต์การส่ง เซตด้วยฮาร์ดแวร์คือสัญญาณปลายช่วงเวลาของบิตที่แปดในโหมด 0 หรือที่จุดเริ่มต้นของบิต STOP ในโหมดอื่น ในการส่งแบบอนุกรมของทุกโหมดจะต้องเคลียร์บิตนี้ด้วยโปรแกรมหลงการส่งแล้ว
RI	เป็นแฟลกอินเตอร์รัพต์การรับ เซตด้วยฮาร์ดแวร์คือสัญญาณที่ปลายช่วงเวลาของบิตที่แปดในโหมด 0 หรือที่จุดครึ่งทางของช่วงบิต STOP ในโหมดอื่น ในการรับแบบอนุกรมยกเว้นกรณีใช้ SM2 จะต้องเคลียร์บิตด้วยโปรแกรมหลังการรับข้อมูลไปแล้ว

ตารางที่ 3 แสดงเรจิสเตอร์ควบคุมพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.14 แสดงผลของแผนภูมิการใช้งานในโหมด 1 พร้อมกับแผนภูมิเวลาสำหรับสัญญาณการส่งและรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 1

จำนวนสิบบิตจะถูกส่งผ่าน TXD หรือรับผ่าน RXD ที่ประกอบด้วยบิต Sart บิตข้อมูล 8 บิต และบิต Stop การรับบิต Stop จะส่งเข้า RB8 ในเรจิสเตอร์ SCON การตั้งอัตราความเร็วของบ๊อดจะแปรผันได้ ด้วยการใช้ตัวจับเวลา การตั้งหรือเคลียร์ค่าบิตใน T2CON เป็น TCLK และ RCLK ภาพที่2.14 แสดงถึงแผนภูมิการใช้งานในโหมด 1 พร้อมกับแผนภูมิเวลาสำหรับสัญญาณการส่งและรับ

การส่งจะเริ่มต้นงานด้วยคำสั่งที่ใช้ SBUF เป็นเรจิสเตอร์รับข้อมูล สัญญาณ ‘ Write to SBUF ‘ ก็จะบรรจุค่า ‘ 1 ‘ เข้าไปเป็นตำแหน่งที่เก้าในเรจิสเตอร์การเลื่อนส่ง และแฟลกในหน่วยควบคุมการส่ง (Tx Control Block) ก็จะแสดงการถูกร้องขอให้ส่งข้อมูลการส่ง ข้อมูลจะส่งที่ช่วง S_tP_1 ของวัฏแมชชีน และจะตามด้วยบิตตัวต่อมา ในช่วงเวลาของสัญญาณนาฬิกาที่หารด้วย 16 ที่ถูกตั้งที่ตัวนับ ดังนั้น แต่ละบิตจะถูกซิ้งค์ด้วยการหาร 16 ของตัวนับ ไม่ใช้ด้วยสัญญาณ ‘ Write to SBUF ‘

การส่งจะเริ่มด้วยการส่งแอกทีฟสัญญาณ SEND และใส่บิต Star เข้าที่ TXD ช่วงเวลาหลังจากนั้นหนึ่งบิต สัญญาณข้อมูลก็จะแอกทีฟ ซึ่งก็จะอีนาเบิ้ลการส่งบิตออกจากเรจิสเตอร์การเลื่อนส่งออกไปยังขา TXD พัลส์เลื่อนตัวบิตแรกจะเกิดขึ้นหลังเวลาทำงานแล้วหนึ่งบิต ขณะที่ข้อมูลเลื่อนออกทางขวา ค่า ‘0’ จะถูกใส่เข้าทางซ้าย เมื่อ MSB ของข้อมูลหนึ่งไบต์อยู่ที่ตำแหน่งเอาต์พุตของเรจิสเตอร์ตัวเลื่อน ขณะนั้น ค่า ‘1’ จะเริ่มถูกบรรจุเข้าเป็นตำแหน่งที่เก้าที่เอาต์พุต หลังจากที่ MSB ถูกส่งออกไป และทุกตำแหน่งเมื่อถูกส่งออกไปแล้ว ทีเหลือในเรจิสเตอร์การเลื่อนจะเป็น ‘0’
(Tx Control Unit) SEND ‘Write to SBUF ‘
‘1’ ‘0’ RXD RXD 01FFH

2.15 คอนโทรลภายในของ Visual Basic

Visual Basic เป็นเครื่องมือที่ช่วยพัฒนาแอปพลิเคชันสำหรับวินโดวส์ตัวแรกที่ประสบความสำเร็จเป็นอย่างมาก ทั้งนี้เนื่องมาจากแนวความคิดที่จะนำเอาความสามารถของคอนโทรลมาใช้ในการออกโปรแกรมนี้เอง เพราะคอนโทรลเป็นเครื่องมือที่ช่วยลดความซับซ้อนในการเขียนโค้ดลงไปได้มากทีเดียว และนอกจากนี้คอนโทรลยังมีส่วนที่แสดงผลเพื่อสื่อความหมายของการทำงานระหว่างคอนโทรลและผู้ใช้ได้อีกด้วย ส่วนการใช้งานก็ไม่มีความซับซ้อนเพียงแต่ผู้อ่านทำการเชื่อมต่อคอนทรลเข้ากับสภาพแวดล้อมของ Visual Basic จากนั้นก็สามารถที่จะนำมาเพิ่มลงในฟอร์มได้ทันที สำหรับ Visual Basic 5.0 ได้มีการแบ่งคอนโทรลออกเป็น 4 กลุ่มหลักๆ ดังนี้

1. คอนโทรลภายใน (Intrinsic control) เช่น ComboBox,CommandButton หรือ PictureBox เป็นต้น ซึ่งเป็นคอนโทรลที่ถูกสร้างลงในสภาพแวดล้อมของ vb.exe ดั้งนั้นที่ทุกครั้งที่ผู้อ่านโหลด Visual Basic คอนโทรลเหล่านี้ออกจอกแถบกล่องเครื่องมือได้เลย ดังนั้นจึงจัดได้ว่าเป็นคอนโทรลมาตรฐาน (Standard Control) กลุ่มหนึ่งของ Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คอนโทรลมาตรฐาน (Standard Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx ที่แยกออกมาต่างหาก เช่น DBGrid (Apex data-bound grid), MSFlexGrid หรือ CommonDialog เป็นต้น ดังนั้นก่อนที่จะสามารถใช้งานคอนโทรลในกลุ่มนี้ได้ เราต้องทำการเชื่อมต่อไฟล์ .ocx เหล่านี้เข้ากับสภาพแวดล้อมของ Visual Basic เสียก่อน โดยใช้คำสั่ง Components ในเมนู Project เช่นเดียวกัน
3. คอนโทรลร่วมวินโดวส์ (Windows Common Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx ที่ต้องใช้ร่วมกับไฟล์ .dll ของวินโดวส์ เช่น RichTextBox, Slider หรือ StatusBar เป็นต้น เช่นเดียวกับคอนโทรลมาตรฐาน เพียงแต่คอนโทรลในกลุ่มนี้ได้ถูกจัดเป็นคอนโทรลพื้นฐานของวินโดวส์ 95 ซึ่งจะติดมากับวินโดวส์ 95 โดยที่คอนโทรลร่วมกับวินโดวส์จะถูกจัดเก็บลงในไฟล์ conctl32.ocx และ conct232.ocx
4. คอนโทรล ActiveX รุ่นมืออาชีพ (Professional ActiveX Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx เช่นเดียวกับคอนโทรลมาตรฐาน เช่น MSComm (Communications), MapiMessages (MAPI message) หรือ MMControl (Multimedia MCI) เป็นต้น แต่คอนโทรลในกลุ่มนี้ได้ถูกสร้างและแจกจ่ายมากับ Visual Basic รุ่น Professional และ Enterprise เท่านั้น

คุณสมบัติแสดงค่าของคอนโทรลที่สำคัญๆ ที่ได้นำมาใช้กับโครงงานนี้

คอนโทรลทั้งหมดที่มากับ Visual Basic ไม่ว่าจะเป็นคอนโทรลภายในหรือ ActiveX จะมีคุณสมบัติตัวหนึ่งที่ถูกใช้สำหรับการกำหนดค่า (Value) หรืออ่านค่าจากคอนโทรล และคุณสมบัตินี้ได้ถูกกำหนดให้เป็นคุณสมบัติปกติ (Default) ของคอนโทรล โดยในการเขียนโค้ดเราสารถใช้เพียงชื่อของคอนโทรล (Control Name) โดยไม่ต้องกำหนดคุสมบัติปกติของทุกๆ คอนโทรลได้โดยไม่เกิดข้อผิดพลาด เช่น คอนโทรล TextBox ก็จะมีคุณสมบัติ Text เป็นคุณสมบัติปกติของคอนโทรล

สำหรับคุณสมบัติปกติของคอนโทรลที่สำคัญที่ใช้กับโครงงานนี้มีดังต่อไปนี้

คอนโทรล	คุณสมบัติ
CommandButton	**Value**
Data	**Caption**
DBGrid (Data-Bound Grid)	**Text**
Label	**Caption**
TextBox	**Text**
Timer	**Enable**

ตารางที่ 4 แสดงคุณสมบัติของคอนโทรลที่สำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแบ่งกลุ่มของคอนโทรลภายใน

เราจะแยกตามวัตถุประสงค์ของการใช้งานได้ทั้งหมด 4 กลุ่ม

1. คอนโทรลภายในทั่วไป แระกอบด้วยคอนโทรลที่แสดงผลในลักษณะของการเลือกตอบหรือเลือกรายการ เช่น CheckBox, OptionBox หรือ ListBox เป็นต้น
2. คอนโทรลภายในด้านระบบไฟล์ ประกอบด้วยคอนโทรลที่ทำหน้าที่ติดต่อหรือแสดงผลระบบไฟล์ (รวมทั้งไดร์ฟและไดเรกทอรี่ด้วย) ของวินโดวส์ เช่น FileListBox หรือ DirListBox เป็นต้น
3. คอนโทรลภายในด้านกราฟิก ประกอบด้วยคอนโทรลที่ทำหน้าที่ด้านการแสดงผลกราฟิกด้วยวิธีกราฟิกของคอนโทรล หรือ ฟังก์ชันวินโดวส์ API หรือ ไฟล์กราฟิกในรูปแบบต่างๆ เช่น PictureBox, Shape หรือ Image เป็นต้น
4. คอนโทรลภายในด้านเวลา ซึ่งจะมีอยู่คอนโทรลเดียวได้แก่ Timer ซึ่งมีหน้าที่สร้างเหตุการณ์ที่ตอบสนองเป็นครั้งๆ ตามช่วงเวลาที่ถูกกำหนด

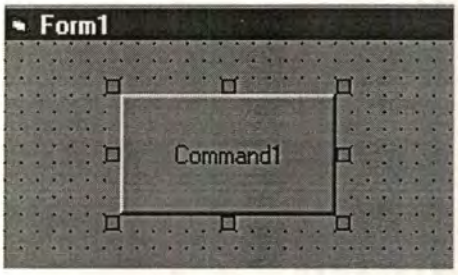
2.15.1 คอนโทรลภายใน

คอนโทรลภายใน ก็จะเป็นคอนโทรลพื้นฐานที่ถูกนำไปใช้งานมากที่สุด เพราะจะเป็นกลุ่มของคอนโทรลที่ช่วยในการสื่อสารแบบสองทางหรือรับเลือกเงื่อนไขจากผู้ใช้ ดังเช่น ทุกๆ แอปพลิเคชันจะใช้คอนโทรล CommandButton สำหรับให้ผู้ใช้เลือกที่ยอมรับ (OK) ยกเลิก (Cancel) หรืออื่นตามข้อกำหนดของแต่ละแอปพลิเคชันเป็นต้นซึ่งคอนโทรลภายในทั่วไปจะประกอบด้วยคอนโทรลต่างๆ ดังต่อไปนี้

(ก) คอนโทรล CommandButton

คอนโทรล Commandbutton (Command Button) จะเป็นคอนโทรลที่ถูกนำไปใช้งานมากที่สุด เพราะในการกำหนดให้ผู้ใช้งานเลือก OK หรือ Cancel นั้น เรามักจะใช้คอนโทรล CommandButton เป็นส่วนใหญ่ ดังนั้นจึงถือว่าเป็นคอนโทรลที่พื้นฐานที่สุดของ Visual Basic เนื่องจากคอนโทรลนี้เป็นปุ่มสำคัญที่ใช้งานในรูปแบบของการคลิกเพื่อยืนยัน ดังนั้นจึงอาจเรียกคอนโทรล CommandButton ได้อีกอย่างว่า Push Button ในขณะออกแบบคอนโทรล CommandButton ที่วางลงบนฟอร์มจะมีลักษณะดังตัวอย่างในภาพที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

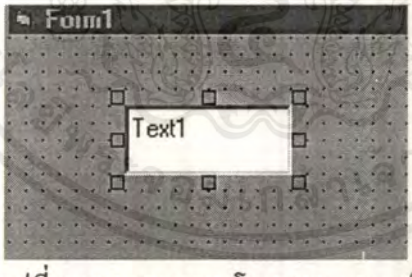


ภาพที่ 2.15 แสดงคอนโทรล ControlButton ในขณะออกแบบ

เราสามารถแก้ไขข้อความที่แสดงผลในคอนโทรลนี้ได้ โดยการแก้ไขข้อความในคุณสมบัติ Caption ของคอนโทรลในหน้าต่างคุณสมบัติหรือแก้ไขโค้ดในแอปพลิเคชันก็ได้

ข) คอนโทรล TextBox (Text Box)

คอนโทรล TextBox (Text Box) มักจะถูกนำไปใช้ทุกๆฟอร์มที่มีการรับการกรอกข้อความจากผู้ใช้เนื่องจากคอนโทรลนี้ทำหน้าที่แสดงข้อมูล (โดยผ่านทางคุณสมบัติ Text) ในคอนโทรลและยังอนุญาติให้ผู้ใช้สามารถแก้ไขตัวอักษรต่างๆ ของคุณสมบัตินี้ได้ด้วยเช่นกัน นอกจากนี้แล้วคอนโทรล TextBox ยังได้รวมเอาความสามารถหลายๆ ด้านของคอนโทรล Label มาใช้ เช่น สามรถแสดงข้อความได้มากกว่า 1 บรรทัด ความสามารถด้าน DDE (Dynamic Data Exchange) และนอกจากรี้ยังสามารถถูกนำไปใช้ในลักษณะของการกรอกรหัสผ่าน (Password) ได้อีกด้วย ดังภาพที่ 2 16



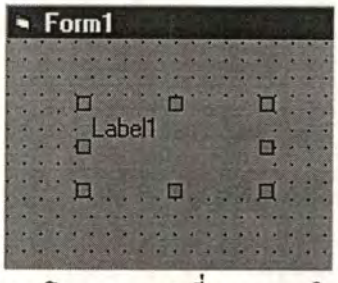
รูปที่ 2.16 แสดงคอนโทรล TextBox ในการออกแบบ

(ค) คอนโทรล Label (Lable)

คอนโทรล Label (Label) เป็นคอนโทรลในลักษณะของกราฟิกที่ถูกใช้งานด้สนการแสดงผลข้อความบนฟอร์ม เหมือนกับผู้อ่านได้นำป้ายข้อความอย่างหนึ่งไปวางไว้บนฟอร์ม เพื่อใช้ในการสื่อข้อความกับผู้ใช้ และคอนโทรลนี้ผู้ใช้ไม่สามาถแก้ไขได้โดยตรงด้วยวิธีการคีย์หรือใช้เมาส์ในขณะรันแอปพลิเคชัน นอกเสียจากภายในแอปพลิเคชันจะมีการเขียนโค้ดสำหรับแก้ไขข้อความในคอนโทรลโดยการแก้ไขค่าคุณสมบัติ Caption เท่านั้น และนอกจากนี้ Label ยังเป็นคอนโทรลที่มีความสามารถด้าน DDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Dynamic Data Exchange) อีกด้วย ในขณะออกแบบเราสารถเพิ่มคอนโทรลลงในฟอร์มหรือตัวบรรจุอื่นๆ ก็จะปรากฏหน้าตาของคอนโทรลดังในภาพที่ 2.17



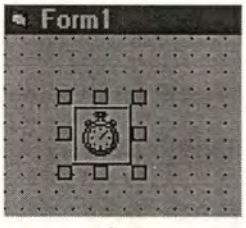
ภาพที่ 2.17 แสดงคอนโทรล Label ที่ถูกวางลงในฟอร์มในขณะออกแบบ

ในการควบคุมพฤติกรรมของคอนโทรล เราสามารถกระทำได้โดยการกำหนดค่าต่างๆ ให้กับคุณสมบัติของคอนโทรล ซึ่งการแก้ไขค่าของคุณสมบัติเราสามารถกระทำได้ทั้งในขณะออกแบบโดยการแก้ไขค่าในหน้าต่างคุณสมบัติ และรับแอปพลิเคชันโดยการเขียนโค้ดเพื่อแก้ไขค่าของคุณสมบัติ แต่ก็มีบางคุณสมบัติที่ไม่สามารถแก้ไขได้ในขณะรันแอปพลิเคชัน เช่น Name เป็นต้น
คอนโทรลภายในด้านเวลา

2.15.2 คอนโทรลภายในด้านเวลา

(ก) คอนโทรล Timer (Timer)

คอนโทรล Timer (Timer) เป็นคอนโทรลที่ใช้ในการควบคุมและจัดการเหตุการณ์ด้านเวลา ซึ่งเทียบได้กับประโยค ON TIME GOTO ของ QuickBasic โดยเราสามารถเขียนโค้ดเพื่อทำงานใดๆ เมื่อช่วงเวลาผ่านไปตามค่าที่กำหนด เช่น ทำการปรับการแสดงผลของฟอร์มทุกๆ 1 นาที เป็นต้น โดยที่คอนโทรลนี้จะตอบสนองเหตุการณ์เพียงเหตุการณ์เดียวเท่านั้น แต่เราสามารถกำหนดให้แต่ละฟอร์มมีคอนโทรล Timer มากกว่า 1 คอนโทรล เนื่องจากคอนโทรล Timer เป็นคอนโทรลที่ทำงานตามนาฬิกาของระบบ ดังนั้นมันจึงถูกควบคุมโดยตัวของระบบเอง สำหรับวินโดวส์ 95 และ NT ในทางปฏิบัติจะไม่มีการจำกัดจำนวนของคอนโทรล Timer ในแต่ละฟอร์ม ดังนั้นเราจึงสามารถใช้งานคอนโทรล Timer พร้อมๆ กันครั้งละหลายๆ คอนโทรล ได้อย่างไม่จำกัด



ภาพที่ 2.18 แสดงคอนโทรล Timer ในขณะออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขณะออกแบบคอนโทรล Timer ที่วางลงบนฟอร์มให้กับฟอร์ม ก็จะมีลักษณะดังรูปที่ 2.18 และเมื่อเรารันแอปพลิเคชัน คอนโทรลนี้จะไม่ถูกแสดงผล แต่จะมีการทำให้เกิดเหตุการณ์ Timer ทุกครั้งที่ช่วงเวลาครบตามค่าที่ได้กำหนดให้กับคุณสมบัติ Interval ของคอนโทรล Timer

คอนโทรลด้านฐานข้อมูล

แอปพลิเคชันที่ใช้กับโครงงานนี้จะต้องมีการเข้าถึงไฟล์ฐานข้อมูล ดังนั้นตัวแปลภาษาที่เหมาะสมกับการสร้างแอปพลิเคชันเหล่านี้ จึงต้องมีเครื่องมือที่สนับสนุนการจัดการฐานข้อมูลอย่างง่ายและมีประสิทะภาพ ซึ่งเราจึงเลือกใช้ Visual Basic เป็นตัวแปลภาษาที่มีการสนับสนุนระบบจัดการฐานข้อมูลในรูปแบบ Microsoft Access โดยอาศัย JET Database Engine ซึ่งมีเครื่องมือที่ให้โปรแกรมเมอร์สามารถจัดการกับฐานข้อมูลได้ 2 วิธีดังนี้

1. **คอนโทรลด้านฐานข้อมูล (Data Control)**

การเข้าถึงฐานข้อมูลด้วยคอนโทรลด้านฐานข้อมูล จะเป็นวิธีง่ายที่สุดในการเขียนโปรแกรม เฉพาะเราเพียงแต่กำหนดไฟล์ฐานข้อมูลฟิลด์สำหรับแต่ละคอนโทรลและคอนโทรล Data เท่านั้น คอนโทรลเหล่านี้จะจัดการสร้าง การแสดงผล การแก้ไขฟิลด์ต่างๆ ของฐานข้อมูลให้โดยอัตโนมัติ หรือตามที่ถูกกำหนดในคุณสมบัติต่างๆ สำหรับความสามารถโดยทั่วไปของคอนโทรลด้านฐานข้อมูลมีรายละเอียดดังต่อไปนี้

- สามารถสร้างตัวแปร Recordset โดยอ้างอิงกับตัวแปร Recordset ที่สนับสนุนโดยคอนโทรลด้านฐานข้อมูล
- แก้ไขโครงสร้างของฐานข้อมูล เช่นการแก้ไขตาราง (Table) ฟิลด์ (Field) หรือ ดัชนี (Index) เป็นต้น
- ค้นหาหรือสืบค้นข้อมูลจากฟิลด์ที่ถูกกำหนดของฐานข้อมูล

คอนโทรลที่สนับสนุนการติดต่อกับฐานข้อมูล (Bound Control) นั้น เราสามารถสังเกตได้จากคอนโทรลที่มีคุณสมบัติ Datafield, DataChange หรือ DataSource เป็นต้น ซึ่งคอนโทรลที่มีคุณสมบัติเหล่านี้ก็จะเป็นคอนโทรลด้สนฐานข้อมูลทั้งสิ้น เช่น CheckBox, PictureBox, TextBox, Data, ProgressBar, RichTextBox หรือ MaskedEdit เป็นต้น

2. **Data Access Object (DAO)**

DAOเป็นโมเดลของคลาสของออบเจ็กต์ที่สนับสนุนการจัดการฐานข้อมูลในระบบ Relational Database ซึ่งก็จะประกอบด้วยคุณสมบัติ โพรซีเยอร์เหตุการณ์และก็วิธี เพื่อช่วยในการสร้าง แก้ไข จัดเก็บ ค้นหาและลบทิ้งฐานข้อมูล โดยในการจัดการกับฐานข้อมูลในทางปฏิบัติ Visual Basic ก็จะอาศัยความสามารถของ JET Database Engine (JET) ซึ่งเป็นเอ็นจิ้นที่ช่วยในการจัดการกับฐานข้อมูลด้านกายภาพ สำหรับ Visual Basic ฐานข้อมูลในรูปแบบ Microsoft Access จะถูกกำหนดให้เป็น

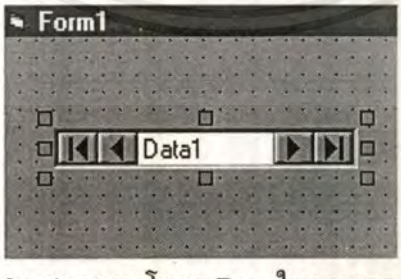
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **คอนโทรล Data** เป็นคอนทรลหลักที่ใช้ในการควบคุมการติดต่อระหว่างคอนโทรลด้านฐานข้อมูลกับฐานข้อมูล โดยที่คอนโทรล Data จะทำหน้าที่ควบคุมการเข้าถึงฐานข้อมูล เช่น การเคลื่อนที่ไปยังเรคอร์ด การเปิด-ปิด การจัดเก็บฐานข้อมูล เป็นต้น
2. **คอนโทรลภายใน Data-Aware** คอนโทรลภายใน Data-Aware เป็นคอนโทรลภายในของ Visual Basic ที่สนับสนุนคุณสมบัติการแสดงผลข้อมูลของฟิลด์หนึ่งๆ ของฐานข้อมูล เช่น CheckBox, PictureBox หรือ TextBox เป็นต้น ซึ่งคอนโทรลภายใน Data-Aware จะแตกต่างกับคอนโทรล Data-Bound ตรงที่คอนโทรลภายใน Data-Aware จะสามารถเชื่อมต่อเข้ากับฟิลด์ของฐานข้อมูลคอนโทรลละ 1 ฟิลด์เท่านั้น
3. **คอนโทรล Data-Bound** เป็นคอนโทรล ActiveX ที่ภูกออกแบบพิเศษเพื่อให้สามารถเชื่อมต่อกับ Record (ประกอบด้วยตั้งแต่ 1 ฟิลด์ขึ้นไป) ของฐานข้อมูล เช่น DBList, DBCombo, DBGrid และ MSFrexGrid ซึ่งคอนโทรล Data-Bound จะสามารถเชื่อมต่อกับหลายๆ เรคอร์ดได้ในเวลาเดียวกัน ทั้งนี้เพื่อขยายขีดความสามารถในการจัดการกับฐานข้อมูลนั่นเอง

(ก) คอนโทรล Data (Data)

คอนโทรล Data (Data) เป็นคอนโทรลที่ใช้เข้าถึงฐานข้อมูล และทำการเชื่อมต่อการแสดงผลข้อมูลของแต่ละฟิลด์ในฐานข้อมูลเข้ากับคอนโทรลด้านฐานข้อมูล (Data-Bound or Data-Aware Control) โดยเมื่อมีการเคลื่อนที่ไปยังเรคอร์ดใดๆ ในฐานข้อมูลด้วยคอนโทรล Data ข้อมูลที่ถูกแก้ไขในคอนโทรลด้านฐานข้อมูลที่เชื่อมต่อกับคอนโทรล Data ซึ่งเป็นเรคอร์ดปัจจุบันในขณะนั้น ก็จะถูกจัดเก็บลงในฐานข้อมูลโดยอัตโนมัติ แล้วจึงเคลื่อนที่ไปยังเรคอร์ดถัดไปทันที แต่เนื่องจากคอนโทรล Data สามารถที่จะจัดเก็บข้อมูลที่ถูกแก้ไขให้โดยอัตโนมัติ ดังนั้นถ้าหากเราต้องการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะถูกเก็บโดยคอนโทรล Data ก็สามารถกระทำได้โดยการเขียนโค้ดเพื่อตรวจสอบความถูกต้องของข้อมูลในโพรซีเยอร์ของเหตุการณ์ Validate

ในขณะออกแบบคอนโทรล Data ที่ถูกวางลงบนฟอร์ม จะมีลักษณะดังรูป



ภาพที่ 2.20 แสดงคอนโทรล Data ในขณะออกแบบ

(ข) คอนโทรล ListView

คอนโทรล **Listview** (ListView) เป็นคอนโทรลตัวหนึ่งในชุดของคอนโทรลร่วมวินโดวส์ ที่มีความสามารถด้านการแสดงผลได้ในหลายๆ รูปแบบ เช่น รูปแบบของไอคอนคล้ายๆกับการแสดงผลไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนต่างๆของโฟลเดอร์ หรือการแสดงผลข้อมูลจากฐานข้อมูลเป็นต้น ซึ่งเราสามารถที่จะแบ่งการแสดงผลรายการออกเป็นคอลัมน์ได้ตามที่ต้องการ โดยที่รูปแบบที่คอนโทรล ListView สามารถแสดงผลมีอยู่ด้วยกัน 4 รูปแบบดังต่อไปนี้

- ไอคอนขนาดใหญ่ (large icon)
- ไอคอนขนาดเล็ก (small icon)
- รายงาน (list)
- รายงาน (report)

คอนโทรลตัวนี้จะเป็นคอนโทรลที่มีคอลเลคชั่นอยู่ 2 อย่างรวมกัน คือ

- คอลเลคชั่น ListItems จะประกอบด้วยออบเจ็กต์ ListItems หลายออบเจ็กต์
- คอลเลคชั่น ColumnHeaders จะประกอบด้วยออบเจ็กต์ ColumnHeader หลายออบเจ็กต์ ซึ่งคอลัมน์จะมีอยู่ในเฉพาะในโหมดของ Report เท่านั้น

ภาพที่ 2.21 แสดงคอนโทรล ListView ในขณะออกแบบ

(ค) คอนโทรล MSComm (Communications)

การคอนโทรลด้านกาสื่อสารผ่านพอร์ตอนุกรม (Serial Port) โดยอาศัยคอนโทรล MSComm ซึ่งกล่าวถึงพอร์ตอนุกรมซึ่งหมายถึง พอร์ตของเครื่องคอมพิวเตอร์ที่วินโดวส์รู้จักในชื่อของ COM1: หรือ COM2: เป็นต้น แต่สำหรับแอปพลิเคชันที่เรานำคอนโทรล MSComm ไปใช้นั้นมีอยู่หลายประเภทด้วยกัน เช่น การเขียนโปรแกรมสื่อสารกับเครื่องคอมพิวเตอร์อื่นๆ หรือ ศูนย์บริการ BBSหรือแม้กระทั่งโฮสต์ที่ใช้บริการอินเตอร์เนตดยผ่านทางโมเดม หรือนอกจากนี้เรายังสามารถที่จะใช้คอนโทรล MSComm ในการติดต่อหรือควบคุมบอร์ดต่างๆ หรือแม้กระทั่งเครื่องอ่านรหัสบาร์โค้ด (Barcode Reader) ที่ต่อผ่านพอร์ตอนุกรมก็ได้เช่นกัน

ด้วยเหตุผลนี้คอนโทรล MSComm จึงมีประโยชน์อย่างมากที่ต้องการสร้างแอปพลิเคชันด้านการสื่อสารหรือการควบคุมบอร์ดต่างๆ ที่ใช้ตามโรงงานอุตสาหกรรมทั่วๆ ไปซึ่งในปัจจุบันได้มีการนำ Visual Basic มาประยุกต์ใช้งานด้านนี้กันอย่างแพร่หลาย เช่น การใช้ Visual Basic ร่วมกับคอนโทรล MSComm ในการติดต่อกับเครื่องวัดการใช้กำลังไฟฟ้าของหม้อแปลงไฟฟ้าในโรงงานอุตสาหกรรม เป็นต้น

เราสามารถที่จะนำ Visual Basic มาประยุกต์ใช้กับอุตสาหกรรมที่ต้องการควบคุมบอร์ดหรืออุปกรณ์ด้านอิเลคทรอนิกส์ผ่านทางพอร์ตอนุกรมแบบอัตโนมัติ (Realtime-Automatic Control) ได้โดยอาศัยเพียงคอนโทรล MSComm เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตอนุกรม (Serial Port)

เนื่องจากในปัจจุบันมีการใช้งานตามมาตรฐานการเชื่อมต่อแบบ RS-232-C กันอย่างแพร่หลาย ซึ่งเป็นมาตรฐานถูกกำหนดโดย EIA ซึ่งเป็นองค์กรอุตสาหกรรมอิเลคทรอนิกส์ของสหรัฐอเมริกา โดยแบ่งการเชื่อมต่อออกเป็น 2 ลักษณะคือ DTE (Data Terminal Equipment) และ DCE (Data Communication Equipment) ซึ่งโดยปกติ DTE จะต้องต่อเข้ากับ DCE เสมอ เช่น การต่อเครื่องคอมพิวเตอร์ (อุปกรณ์ DTE) เข้ากับอุปกรณ์โมเด็ม (อุปกรณ์ DCE) เป็นต้น

พอร์ตอนุกรม RS-232-C จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้งประเภท 9 และ 25 ขา และเราเรียกกันว่าพอร์ต COM1: และ COM2: นั่นเอง ในความจริงพอร์ตอนุกรมไม่ได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกเก็บโดยชิป UART (Universal Asynchronous Reciever/Transmitter) อีกทีหนึ่ง ซึ่งในปัจจุบันเบอร์ที่ใช้กันมากที่สุดก็คือเบอร์ 16550C ซึ่งเป็นเวอร์ชันที่ได้รับการแก้ไขข้อผิดพลาดแล้ว ซึ่งชิป UART นี้จะทำหน้าที่ในการรับและส่งข้อมูลดังต่อไปนี้

การส่งข้อมูล (Data Transmission)

- รับตัวอักษรจากเครื่องคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่าขบวนการ Serialization)
- สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น บิต START, STOP และ PARITY เป็นต้น
- ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมา ด้วยความเร็วของโมเด็มหรือพอร์ตอนุกรม (Baud Rate)
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์

การรับข้อมล (Data Reciever)

- รับตัวอักษรจากอินเตอร์เฟซ
- ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานเฟรมที่กำหนด โดยถ้าหากเฟรมข้อมูลมีรูปแบบทีไม่ถูกต้องก็จะมีการแจ้งข้อผิดพลาดทันที
- ตรวจสอบความถูกต้องของพาริตี้
- แปลงสายข้อมูลแบบบิตให้เป็นตัวอักษร
- ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟซ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15.4 คอนโทรล MSComm (Communications)

คอนโทรล MSComm (Communications) เป็นคอนโทรลตัวหนึ่งที่ช่วยในการติดต่อกับพอร์ตอนุกรม (Serial Port) ซึ่งผู้อ่านสามารถทำหารรับ-ส่งข้อมูลผ่านทางพอร์ตอนุกรมได้ด้วยคอนโทรลนี้ เช่น การติดต่อผ่านทางโมเด็ม (Modem) หรือ ติดต่อโดยตรงกับบอร์ดอิเลคทรอนิกส์ เป็นต้น ซึ่งคอนโทรล MSComm ที่มากับ Visual Basic จะเป็นคอนโทรลที่ทำงานโดยมีการตอบสนองต่อเหตุการณ์แบบ Event-Driven นั่นก็คือคอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือร้องขอให้เกิดเหตุการณ์ต่างๆ กับพอร์ตอนุกรมโดยอัตโนมัติ และจะมีการแจ้งเตือนให้ผู้อ่านได้รับทราบโดยผ่านโพรซีเยอร์เหตุการณ์ เช่นเดียวกับคอนโทรลทั่วไปของ Visual Basic นั่นเอง ดังนั้นในการเขียนโค้ดเราจึงไม่จำเป็นต้องสร้างโพรซีเยอร์ที่ทำหน้าที่คอยตรวจสอบเหตุการณ์ต่างๆ ของพอร์ตอนุกรมซึ่งจะทำให้ง่ายต่อการทำงานเป็นอย่างมาก

คอนโทรล MSComm จะมีหน้าที่มาตรฐานหลักๆ สำหรับการสื่อสารผ่านพอร์ตอนุกรม 3 ประการ ดังต่อไปนี้

- หมุนหมายเลขติดต่อกับโทรศัพท์ปลายทางที่กำหนด
- ตรวจสอบการเข้ามาของข้อมูลยังพอร์ตอนุกรมโดยอัตโนมัติ
- ส่งข้อมูลตามที่กำหนดจากโปรแกรมไปยังพอร์ตอนุกรม

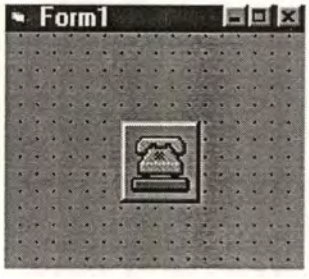
ในความจริงคอนโทรล MSComm ไม่ได้ทำหน้าที่ติดต่อกับพอร์ตอนุกรมโดยตรง แต่มันจะทำหน้าที่เรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งวินโดวส์จะทำการส่งหรือรับข้อมูลผ่านทางพอร์ตอนุกรมโดยอาศัยไดรเวอร์ Comm.drv อีกทอดหนึ่ง ดังนั้นจึงสามารถสรุปสั้นๆ ได้ว่าทุกครั้งที่เรามีการเรียกใช้คอนโทรล MSComm ก็หมายถึงเรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งจะถูกตีความอีกทอดหนึ่งโดยไดรเวอร์ Comm.drv จากนั้นก็จะส่งผ่านข้อมูลที่ถูกจัดรูปแบบตามมาตรฐานการสื่อสาร (ทั้งนี้ขึ้นกับอุปกรณ์ที่ต่อเข้ากับพอร์ตอนุกรม) ให้กับดีไวซ์ไดรเวอร์อีกทอดหนึ่งนั่นเอง

การกำหนดคุณสมบัติของคอนโทรล MSComm ในขณะออกแบบ ผูเราสามารถกระทำได้อย่างสะดวกโดยการคลิกที่ปุ่มของรายการ (Custom) ในหน้าต่างคุณสมบัติ ซึ่งก็จะปรากฏไดอะล็อกบ็อกซ์ Property Pages เพื่อให้เราได้ปรับแต่งค่าของคุณสมบัติของคอนโทรล MSComm สนับสนุน ซึ่งปุ่มต่างๆ ของไดอะล็อกบ็อกซ์ Property Pages มีความหมายดังนี้

- ปุ่มคำสั่ง OK ยอมรับการแก้ไขคุณสมบัติของคอนโทรล MSComm
- ปุ่มคำสั่ง Cancel ยกเลิกการแก้ไขคุณสมบัติของคอนโทรล MSComm
- ปุ่มคำสั่ง Apply อัปเดตคุณสมบัติที่ถูกแก้ไขของคอนโทรล MSComm
- ปุ่มคำสั่ง Help แสดงผล Help ของคอนโทรล MSComm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับฟอร์มหนึ่งๆ เราสามารถเพิ่มได้หลายๆ คอนโทรล MSComm ทั้งนี้จึ้นอยู่กับความต้องการของเราในการติดต่อกับพอร์ตอนุกรมใดบ้าง สำหรับวินโดวส์ 95 และ NT 4.0 เราสามารถติดตั้งพอร์ตอนุกรมได้มากกว่า 4 พอร์ต โดยเราสามารถเพิ่มคอนโทรล MSComm ลงในฟอร์ม ก็จะปรากฏดังรูปที่



ภาพที่ 2.22 แสดงคอนโทรล MSComm ในขณะออกแบบ

2.20.5 การสร้างรายงานด้วย Report Designer

ในการทำงานเกี่ยวกับฐานข้อมูล บางครั้งเราจำเป็นจะต้องทำรายงานข้อมูลออกมา เพื่อไปทำงานบางอย่าง เช่น ต้องการส่งจดหมายไปยังสำนักพิมพ์ตามที่อยู่ เป็นต้น ใน VB5 สามารถสร้างรายงานตามที่เราต้องการได้จากโปรแกรมที่มีมาให้ คือโปรแกรม Crystal Report ที่ช่วยให้เราสร้างรายงานต่างเช่น รูปแบบของจดหมายม, ตารางข้อมูล เป็นต้น ได้ง่ายขึ้น

คุณสมบัติที่สำคัญของคอลโทรล Crystal Report

คอนโทรล Crystal Report จะมีคุณสมบัติที่สำคัญดังต่อไปนี้

คุณสมบัติ ReportFileName

เป็นคุณสมบัติที่เก็บไดเรกทอรีของไฟล์ของรายงานที่เราต้องการพิมพ์ ค่านี้สามารถเปลี่ยนแปลงได้ทั้งตอนรันโปรแกรม และออกแบบโปรแกรม

คุณสมบัติ Destination

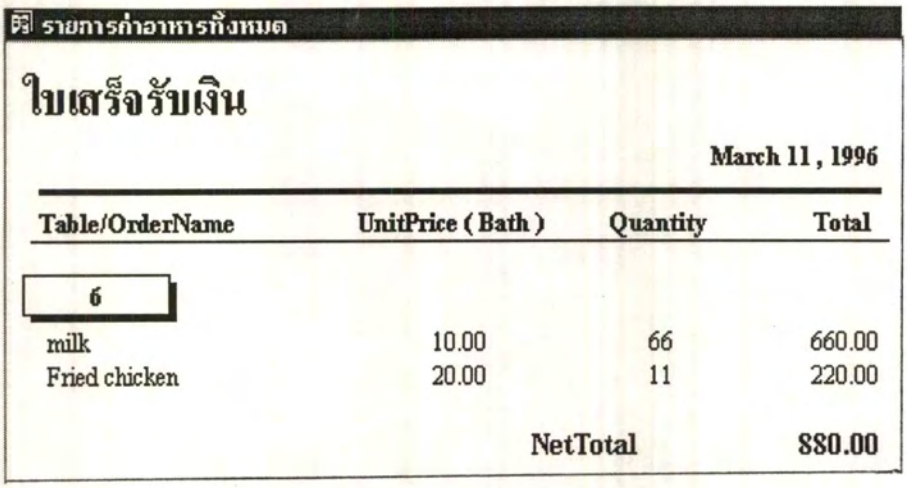
เป็นคุณสมบัติที่จะบอกว่าให้พิมพ์รายงานลงที่ใด มีค่าอยู่ 4 ค่า คือ

- 0 - To Window
- 1 - To Printer
- 2- To File
- 3- To MAPI (สำหรับ E-mail)

คุณสมบัติ SelectionFormula

ใช้สำหรับกำหนดเงื่อนไขในการพิมพ์ ซึ่งถ้าไม่ระบุจะหมายถึงทุก Record

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.23 แสดงการใช้งาน Crystal Report

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

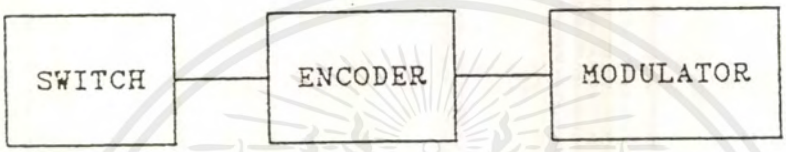
บทที่ 3
วงจรส่งและรับอินฟราเรด

3.1 ภาคเครื่องส่ง

ภาคเครื่องส่งของรีโมทคอนโทรล จะแบ่งการทำงานออกเป็น 2 ส่วนคือ

(ก) ตัวเข้ารหัส (Encoder)

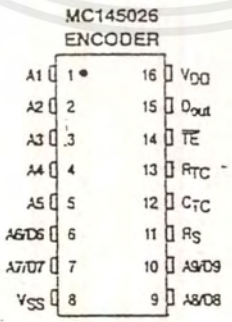
(ข) มอดดูเลเตอร์ (Modulator)



ภาพที่ 3.1 บล็อกไดอะแกรม

3.1.1 การทำงานของภาคส่ง

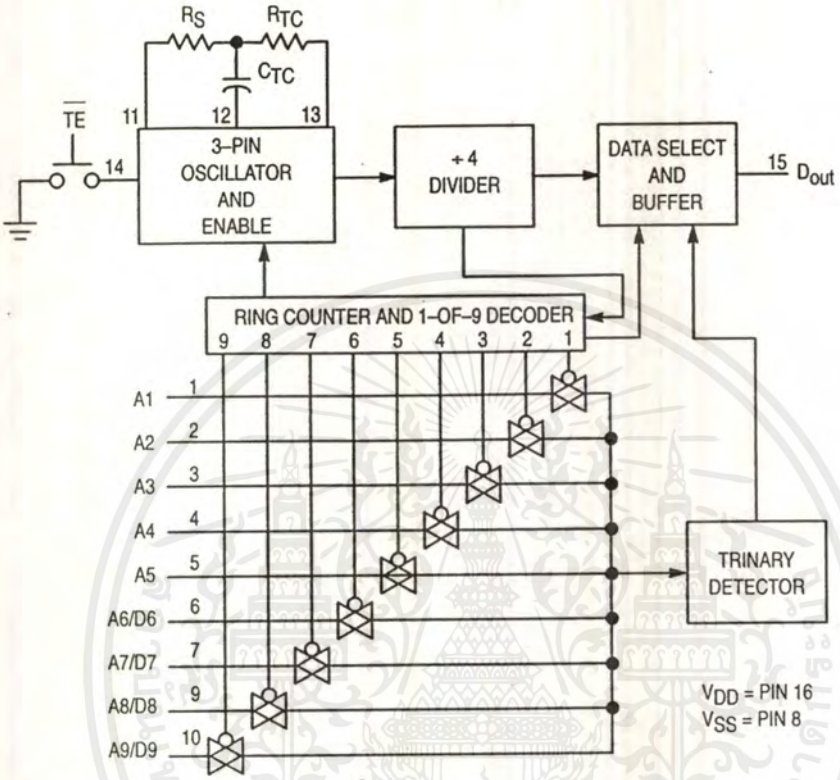
อินพุทของภาคส่งนั้นจะรับเข้ามาที่ขา A1/-A9/D9 ของไอซี MC145026 ซึ่งมีการรับข้อมูลเข้าแบบขนาน แล้วจะทำการเข้ารหัสข้อมูลแบบ 9 บิตของข้อมูล และจะส่งข้อมูลนี้ไปเป็นแบบอนุกรมโดยข้อมูลจะสามารถส่งออกไปได้ทันทีที่ขาทรานสมิทเอ็นนาเบิล (Transmit-Enable) (TE) ได้รับสภาวะเป็น “0” ซึ่งมันจะทำงานที่สภาวะ “0” เท่านั้น



ภาพที่ 3.2 ตัวเข้ารหัสของ MC145026

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ถูกส่งออกมาจากขา 15 ของ MC145026 จะถูกส่งต่อเข้าไปยังส่วนมอดดูเลเตอร์ (Modulator) เพื่อทำการมอดดูเลทข้อมูล (Modulate Data) เข้ากับสัญญาณพาหะ ซึ่งได้มาจาก วงจรแท็งค์ทูน(Tank Tune) ซึ่งกำเนิดความถี่ในย่านความถี่ UHF เมื่อข้อมูลถูกมอดดูเลทกับสัญญาณพาหะแล้ว สัญญาณจะถูกส่งออกไปยังเครื่องรับ



ภาพที่ 3.3 บล็อกไดอะแกรมตัวเข้ารหัสของ MC 145026

3.1.2 คุณสมบัติของ MC145026

(ก) แอดเดรสสามารถเป็นไปได้ทั้ง ไบนารี่ (Binary) หรือไตรนารี่ (Trinary)

(ข) โค้ดของแอดเดรสที่สูงที่สุด คือ ไตรนารี่

(ค) ใช้ในการอินเตอร์เฟส (Interfaces) กับอาร์เอฟ (RF), อัลตราโซนิก (Ultrasonic)

(ง) จะส่ง 2 ข้อมูล เพื่อสำหรับ Error checking

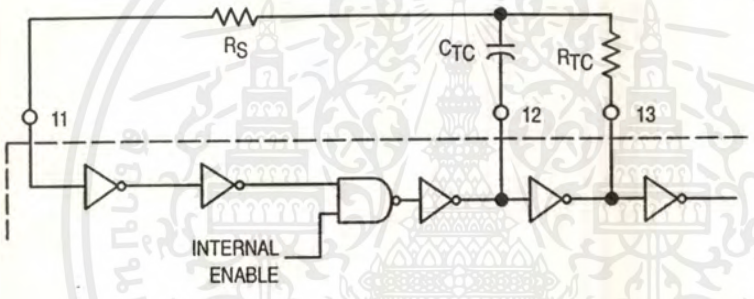
(จ) ใช้ไฟตั้งแต่ 4.5-18 โวลต์

(ฉ) ในชิฟนั้นจะมี R/C Oscillator

(ช) อินพุท และเอาท์พุทเป็นแบบอนุกรมมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนที่เกี่ยวข้องกับการเข้ารหัสหรือเรียกว่า ตัวเข้ารหัสอาศัยการทำงานของไอซีเบอร์ MC145026 ซึ่งมีการรับข้อมูลเข้าแบบขนาน และส่งข้อมูลออกแบบอนุกรม รหัสทางอินพุทของ MC145026 นี้จะสามารถเข้ารหัสได้ถึง 3 สถานะ คือ เป็นได้ทั้งระดับลอจิก "0" ระดับลอจิก "1" และสภาวะอิมพีแดนซ์สูง คือปล่อยลอยไว้ จากคุณสมบัตินี้เราจึงสามารถจะเข้ารหัสได้ถึง (3^9) = 19683 ที่ไม่ซ้ำกันซึ่งลำดับในการส่งนั้น จะเริ่มต้นที่ "Low Level" ของขาอินพุททีอี (Input TE) และ Vdd ส่วนมากจะใช้เป็น "Positive Supply" (GND) แล้วภาคตัวเข้ารหัส นี้ยังมี R_S, R_{TC}, C_{TC} ขาเหล่านี้จะเป็นส่วนของออสซิลเลเตอร์ (Oscillator) ของการเข้ารหัส สำหรับกำหนดคาบเวลาหรือความถี่ภายใน ถ้าแหล่งจ่ายสัญญาณภายนอกใช้แทนออสซิลเลเตอร์ ภายในมันจะต่อไปขา R_S,R_{TC},C_{TC} จะ "leftopen" ซึ่งจะเห็นได้ว่าออสซิลเลเตอร์นี้ทำงานที่ความถี่กำหนดโดย RC network ภายนอก



ภาพที่ 3.4 ตัวเข้ารหัสออสซิลเลเตอร์อินฟรอแมชั่น (Oscillator Information)

RC network ภายนอก สามารถคำนวณได้จากสูตร

$$f = 1/(2.3*R_{TC}*C_{TC}) \quad \text{Hz}$$

For 1kHz <=1<=400 kHz

$C_{TC'} = C_{TC} + C_{layout} + 12\ pF$

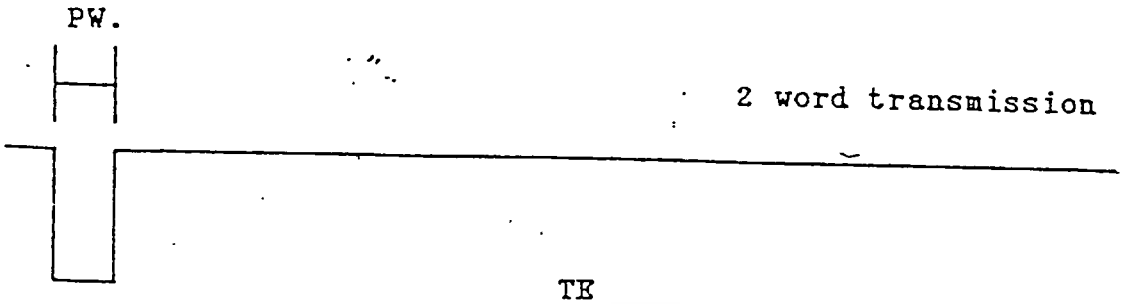
$R_S = 2R_{TC}$

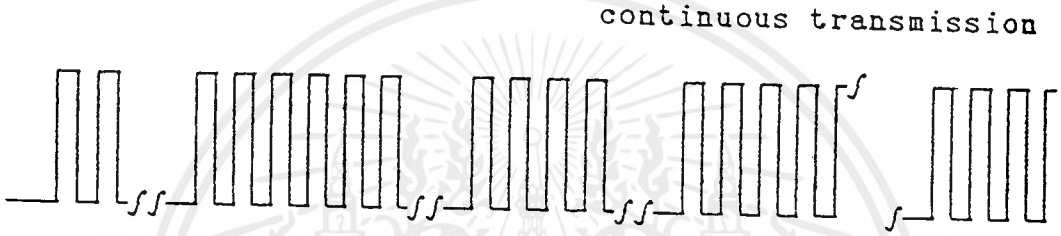
$R_S >= 20K$

$R_{TC} >= 10K$

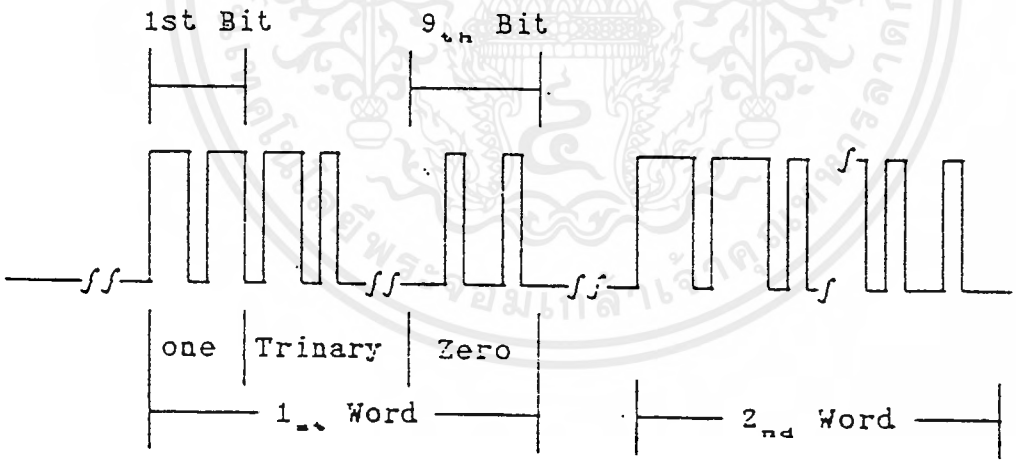
$400pF < C_{TC} < 15\ \mu F$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





ภาพที่ 3.5 ตัวเข้ารหัสออสซิลเลเตอร์ (Pin12)



ภาพที่ 3.6 แผนภูมิเวลาของตัวเข้ารหัส

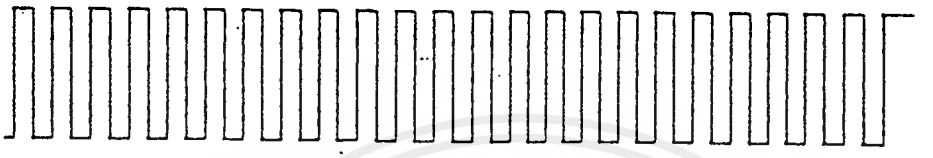
ดังนั้นข้อมูลเอาท์พุทที่ขา 15 ของ MC 145026 ที่ส่งออกไปจะส่งไปในลักษณะอนุกรมต่อกันไปขนาด 9 บิต ซึ่งในแต่ละบิตของข้อมูลที่ส่งออกไปจะมีสภาวะเป็น 0,1 Open สภาวะใดก็ได้ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

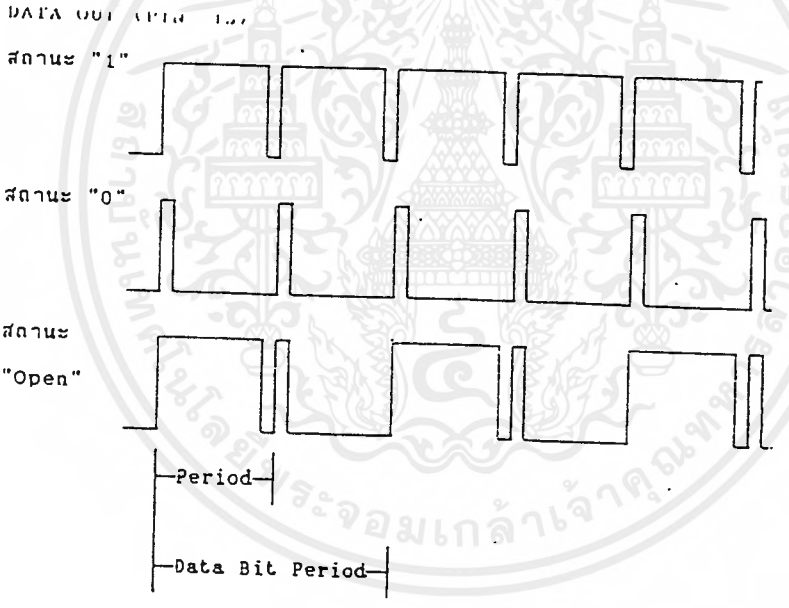
จะมีสัญญาณเป็นลักษณะพัลส์ (Pluse) ที่เข้ารหัสเป็นแบบอนุกรมแล้วดังรูปข้างล่างนี้ และส่งไปยังภาคมอดดูเลเตอร์

รูปคลื่นข้อมูลของตัวเข้ารหัส (Encoder Data Waveform) (MC 145026)

การส่งแบบต่อเนื่อง (Continous Transmission)



ภาพที่ 3.7 ตัวเข้ารหัสออสซิลเลเตอร์ (Pin12)



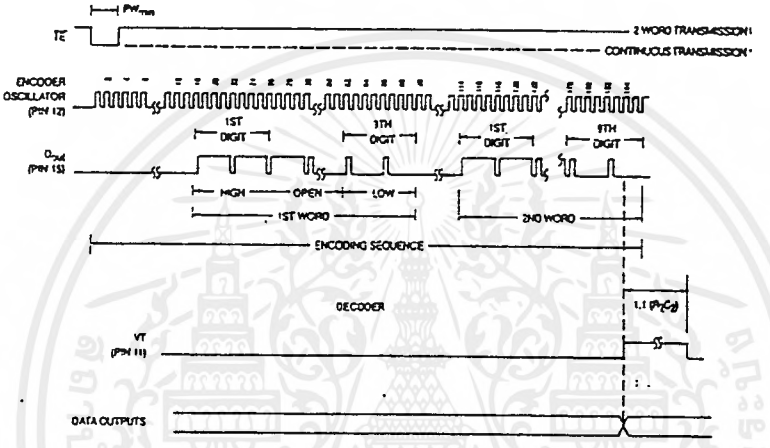
ภาพที่ 3.8 แสดงลักษณะการเข้ารหัสของบิทข้อมูล

ข้อมูลต่างๆ จะเริ่มมีการส่งอนุกรมต่อเนื่องกันไป ก็ต่อเมื่อทันทีที่สวิตช์ทีอีถูกกด (ที่ขาทีอี ได้รับ Active Low) ซึ่งข้อมูลดังกล่าวจะถูกส่งออกไปมีลักษณะเป็นคำ (Word) โดยจะถูกส่งออกไปเป็นจำนวน 2 คำ(Data Words) ทั้งสองนี้ได้เหมือนกันแสดงว่า การส่งและการรับ ข้อมูลเป็นไปอย่างถูกต้องสมบูรณ์ที่สุด ทำให้ทางภาครับผลิตสัญญาณ VT (Valid Transmission) ออกมาทุกครั้งเลย

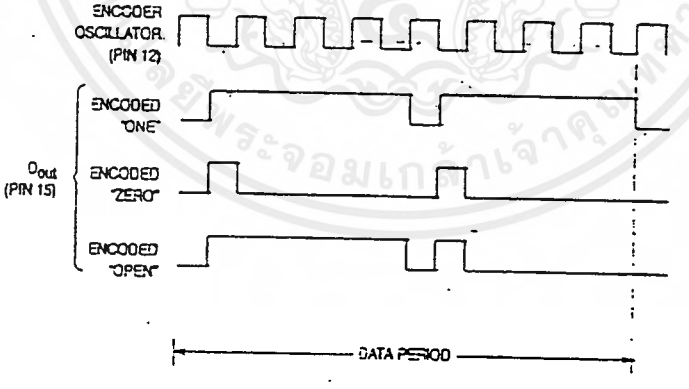
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการส่งแต่ละครั้ง บิทข้อมูล จะถูกเข้ารหัสแบบ 2 พัลส์ (Data Pluse) โดยที่ลอจิก "0" จะมีลักษณะเป็นพัลส์สั้นๆ 2 พัลส์ต่อเนื่องกันไป และลอจิก "1" จะมีลักษณะเป็นแบบพัลส์ยาวๆ 2 พัลส์ต่อเนื่องกัน ส่วนสำหรับสภาวะ "Open" จะมีลักษณะการเข้ารหัสเป็นแบบพัลส์ยาว 1 ครั้ง แล้วตามด้วยพัลส์สั้นๆ อีก 1 พัลส์ ดังแสดงการเข้ารหัสของสภาวะต่างๆ ดังกล่าว (0,1,Open) ดังรูป

ดังนั้นข้อมูลที่ออกที่ขา 15 ของ MC 145026 ขนาด 9 บิต ซึ่งในแต่ละบิตของข้อมูลที่ส่งออกไปจะมีสภาวะเป็น 0,1, Open สภาวะใดก็ได้ ซึ่งจะมีสัญญาณพัลส์ ที่ส่งไปยังภาคมอดดูเลเตอร์



ภาพที่ 3.9 แผนภูมิเวลา



ภาพที่ 3.10 รูปคลื่นของข้อมูลตัวเข้ารหัส

3.1.3 การคำนวณและการทำงานของภาคมอดดูเลเตอร์

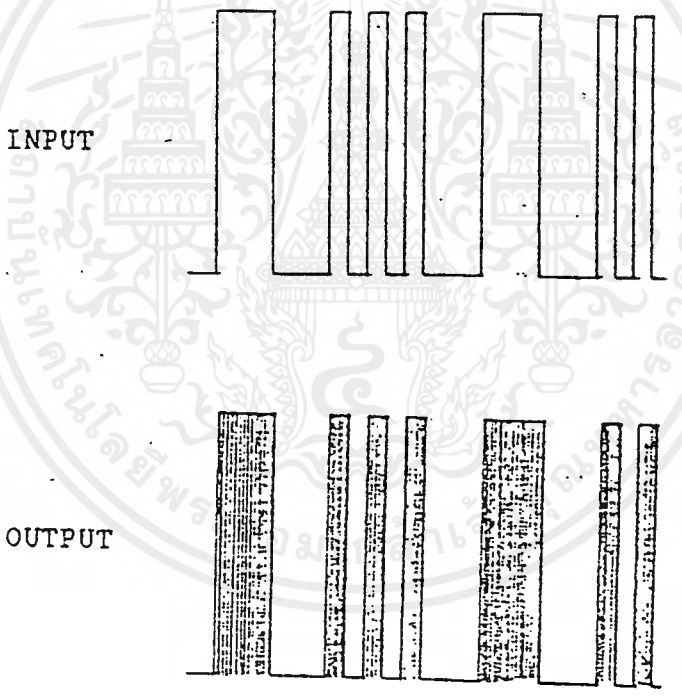
หน้าที่ของภาคมอดดูเลเตอร์ คือทำการมอดดูเลทสัญญาณที่ได้จากภาคตัวเข้ารหัสกับสัญญาณพาหะโดยใช้วงจรแท็งค์ทูน เป็นตัวกำเนิดสัญญาณพาหะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงาน

ภาคมอดดูเลเตอร์ทำการผลิตสัญญาณพาหะออกมาทันทีเมื่อขาเบสของทรานซิสเตอร์มีสภาวะเป็น "1" จึงจะทำให้มีสัญญาณด้วยความถี่ 430 MHz และในขณะเดียวกันนั้นหากระดับที่ขาเบสของทรานซิสเตอร์มีสภาวะเป็น "0" จะทำให้ภาคมอดดูเลเตอร์หยุดผลิตสัญญาณพาหะจึงทำให้ไม่มีสัญญาณใดๆ ปรากฏออกมา

ดังนั้นเมื่อนำสัญญาณข้อมูลที่ได้จากภาคตัวถอดรหัส ซึ่งจะมีลักษณะเป็นสัญญาณสี่เหลี่ยม (Square Wave) แบบพัลส์วิดมอดดูเลชั่น (Pulse Width Modulation) นำมาป้อนแก่ขาเบสของทานซิสเตอร์ จะทำให้มีสัญญาณขาออกปรากฏออกมา เป็นลักษณะเหมือนสัญญาณข้อมูลที่ป้อนเข้ามา แต่มีการมอดดูเลชั่นกับสัญญาณพาหะ เข้าไปด้วยกัน ดังรูป



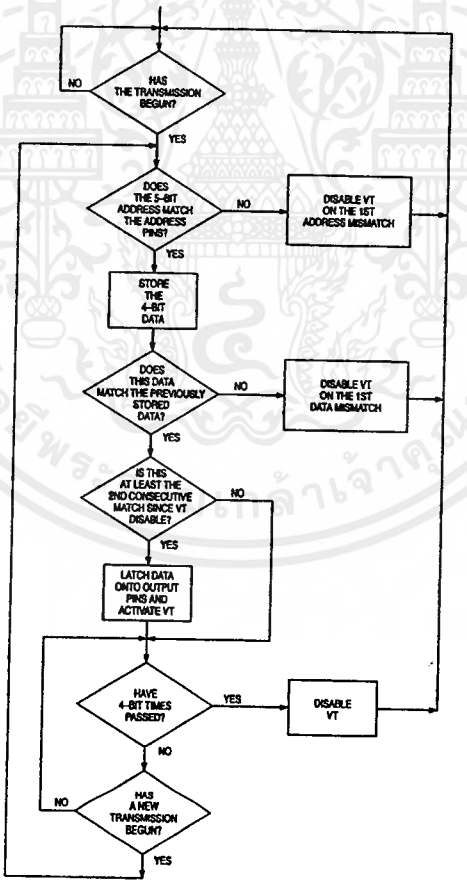
ภาพที่ 3.11 แสดงสัญญาณขาเข้าและขาออกของภาคมอดดูเลชั่น

3.2 ภาครับ MC145027

ทำหน้าที่เป็นตัวถอดรหัสสัญญาณข้อมูลอนุกรมจากตัวเข้ารหัส จากนั้นจะทำการตรวจสอบข้อมูล ถ้าสอดคล้องกันก็จะส่งข้อมูลออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณข้อมูลอนุกรมที่รับมาจากตัวเข้ารหัสประกอบด้วย 2 คำที่เหมือนกัน ซึ่งจะถูกตรวจสอบทีละบิทระหว่างการรับข้อมูล โดย 5 ไตรนารี่ดิจิต (Trinary Digit) แรก ถูกสมมติให้เป็นตำแหน่ง (Address) ถ้า ตำแหน่ง 5 ไตรนารี่ดิจิตแรกนี้สอดคล้องกัน 4 บิทข้อมูล ถัดไปจะถูกเก็บไว้ภายใน โดยจะยังไม่ถูกส่งไปยังสัญญาณขาออกแลทช์ (Output Data Latch) ต่อจากนั้นเมื่อคำที่ 2 จากตัวเข้ารหัสถูกส่งเข้ามาตำแหน่งจะต้องสอดคล้องกันอีกครั้ง ถ้าตำแหน่งที่ได้รับมานี้สอดคล้องกันเรียบร้อยแล้ว บิทข้อมูล จะถูกตรวจเทียบกับ บิทข้อมูล ที่เก็บไว้ตอนแรก ถ้าบิทข้อมูล ทั้ง 4 บิท สอดคล้องกันข้อมูลจะถูกส่งไปยังสัญญาณขาออกแลทช์ โดย VT และจะยังคงสถานะเดิมอยู่ จนกระทั่งเมื่อมีข้อมูลอื่นมาแทนที่ ในขณะเดียวกันที่ขา VT output (pin 11) จะมีสถานะเป็น high และยังคงเป็น high เว้นแต่ว่าจะมี Error เกิดขึ้น หรือไม่ได้รับสัญญาณขาเข้าเป็นเวลา 4 คาบเวลาข้อมูล (Data Period) แม้ว่าตำแหน่งจะถูกเข้ารหัสเป็นไตรนารี่แต่ข้อมูลที่แท้จริงแล้ว ต้องเป็น 1 หรือ 0 ซึ่งข้อมูลไตรนารี่ จะถูกถอดรหัส เป็นลอจิก 1



ภาพที่ 3 .12 แผนภูมิการทำงานของ MC 145027

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของขาต่าง ๆ ใน ตัวถอดรหัส MC145027

A1 – A5 address input

เป็นตำแหน่งขาเข้า (Address Input) ที่สถานะของแต่ละ pin จะต้องสอดคล้องกับตัวเข้ารหัสขาเข้า (Encoder Input) เพื่อให้ VT เปลี่ยนสถานะเป็น high ซึ่งสัญญาณขาเข้าเหล่านี้จะถูก เข้ารหัสในรูปแบบของไตรนารี่ หรือข้อมูลแบบไบนารี่ (Binary Data)

D6 –D9 data output (pin15, 14,13,12)

สัญญาณขาออกเหล่านี้เป็นรูปแบบของข้อมูลแบบไบนารี่

Din (data in pin 9)

ขานี้เป็นทางเข้า ของสัญญาณ ข้อมูลอนุกรมที่ส่งมายังตัวถอดรหัส

R1,C1 (pin6,pin7)

ค่าเวลาคงที่ (Time Constant) R1*C1 จะถูกเซ็ทให้เป็น 1.72 ของคาบเวลาของสัญญาณนาฬิกาจากตัวเข้ารหัส

$$R1*C1 = 3.95RtcCtc$$

R2/C2 (pin 10)

ค่า ความต้านทาน และค่าตัวเก็บประจุนี้ ต่อเพื่อทำหน้าที่ตรวจสอบการสิ้นสุดของคำและการสิ้นสุดการส่งสัญญาณที่รับมาจาก ตัวเข้ารหัส ซึ่งค่า Time Constant (R2*C2) นี้ มีค่าเป็น

$$R2*C2 = 77RtcCtc$$

VT Valid Transmission Output (pin11)

ที่ขานี้จะเป็น High หลังจากที่คำทั้งสองที่ส่งมาจากตัวเข้ารหัส ได้รับการตรวจสอบว่าสอดคล้องกับตำแหน่งของตัวถอดรหัส

Vss (pin 8)

เป็น Ground หรือไฟลบ

Vdd (pin16)

เป็นไฟบวก ที่ป้อนให้กับ ตัวถอดรหัส

3.2.1 ไทม์มิ่ง (Timing)ของ MC145027

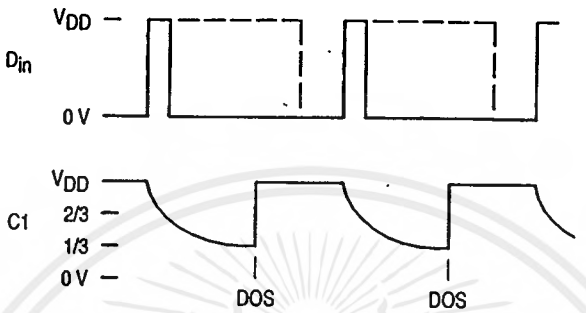
การตรวจสอบ MC145027 ไทม์มิ่งทำโดยตรวจสอบคลื่นสัญญาณที่ C1 (pin 7) และ R2/C2 (pin 10) แล้วนำมาเปรียบเทียบกับคลื่นสัญญาณของข้อมูลอนุกรมที่ถูกส่งมาจากตัวเข้ารหัสที่ Din (pin 9)

ค่าการลดลงของแรงดันที่ C1 ในขณะที่มีการคายประจุ จะลดลงจาก Vdd เป็น 1/3 Vdd ที่จุดนี้ (Dos ในภาพที่ 3.13) จะเป็นจุดรีเซ็ท ในขณะที่สัญญาณข้อมูลขาเข้าเปลี่ยนสถานะจาก 0 เป็น 1 หรือจาก 1 เป็น 0

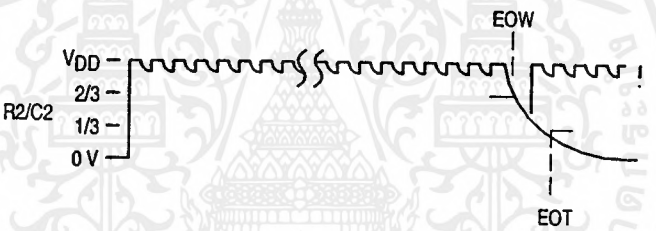
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบไทม์มิ่งสามารถตรวจสอบได้ที่ R2/C2 (pin 10) ได้เช่นเดียวกัน โดยที่แรงดันที่บิทนี้ จะมีค่า Vddในขณะที่การส่งข้อมูลสิ้นสุดลง (EOT) แรงดันที่ขานี้จะลดลงเป็น 1/3Vdd ซึ่งที่เวลานี้ทำให้เราทราบจุดสิ้นสุดของคำ (EOW) ได้ ณ จุดนี้แรงดันที่ pin 10 จะลดลงมาเป็น 2/3Vdd

จากการวัดไทม์มิ่งที่ได้กล่าวมา ทำให้เราสามารถตรวจสอบ และแน่ใจว่า การสอดคล้องกันระหว่าง ตัวเข้ารหัส และ ตัวถอดรหัส ทำงานได้อย่างสมบูรณ์



ภาพที่ 3.13 R-C Decay on Pin 7 (C1)



ภาพที่ 3.14 R-C Decay on Pin10 (R2/C2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

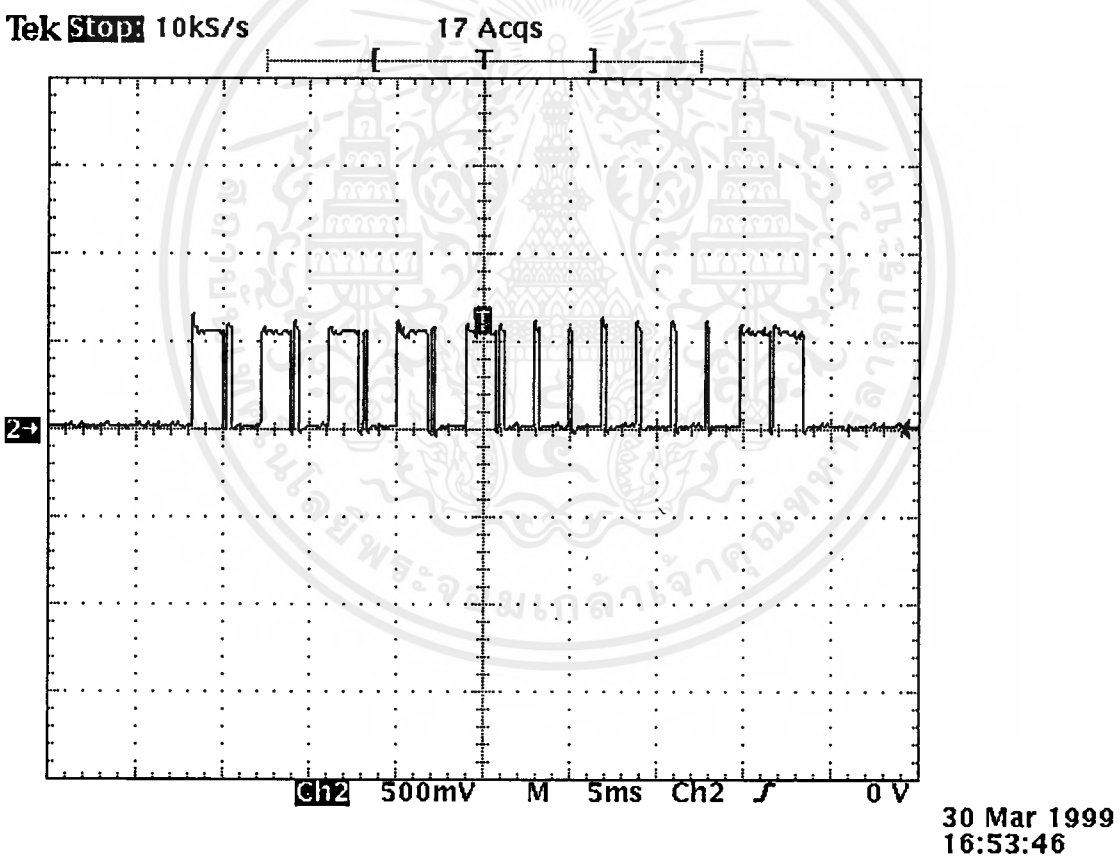
บทที่ 4

การทดสอบการทำงานของระบบ

4.1 ผลการทดลอง

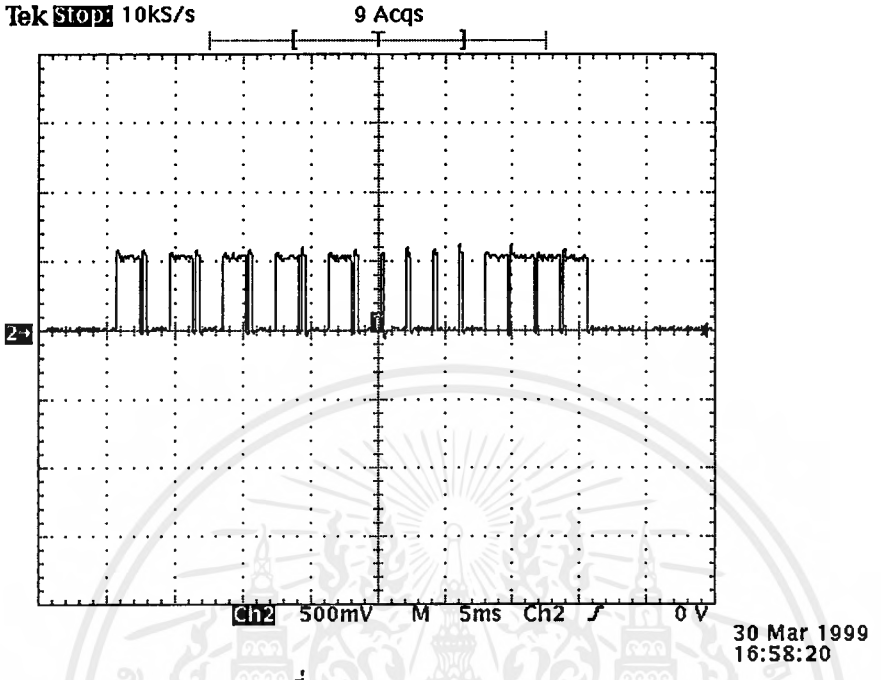
4.1.1 ผลการทดลองวงจรส่งและรับอินฟราเรด

1 เมื่อกดสวิทช์ S1-S12 (CH1-CH12) และวัดค่า Dout (pin 15)ของ MC145026 ซึ่งทำหน้าที่เป็นตัวเข้ารหัส จะได้ output เป็นรหัสต่างๆ กัน ตามรูปที่ 4.1-4.12 ตามลำดับ

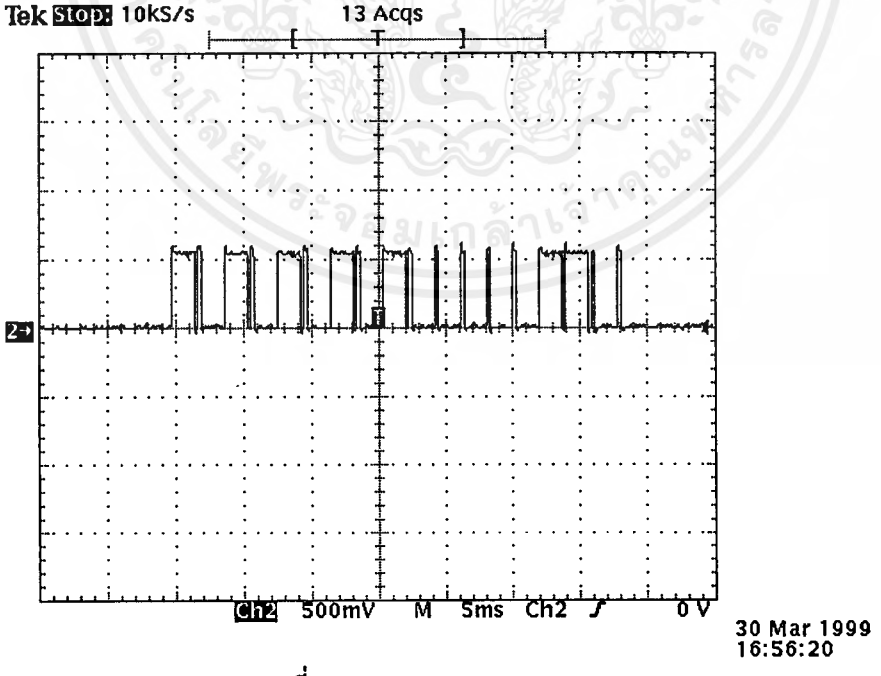


ภาพที่ 4.1 Channel 0. XXXXX0001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

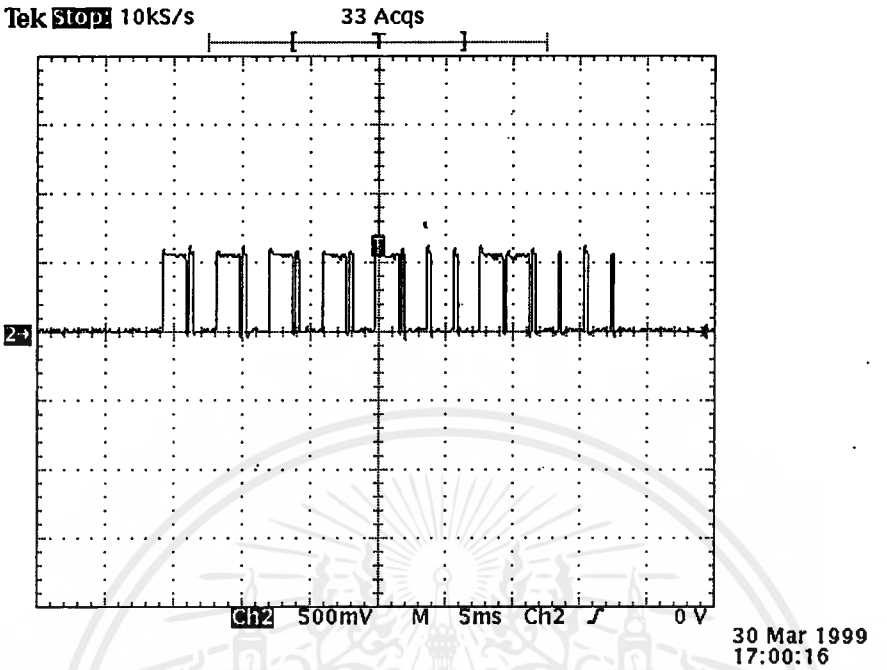


ภาพที่ 4.2 Channel 2. XXXXX0011

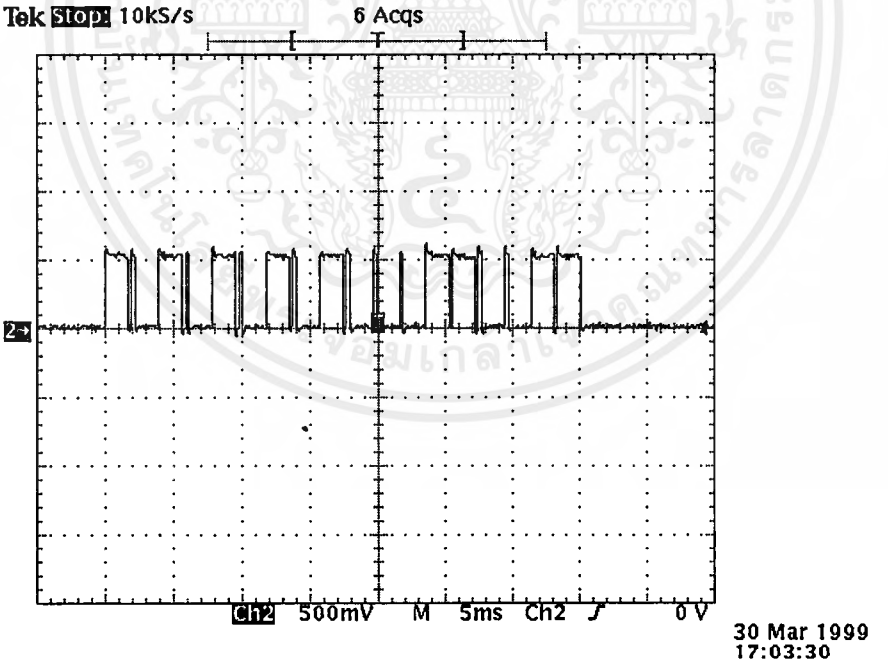


ภาพที่ 4.3 Channel 1. XXXXX0010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

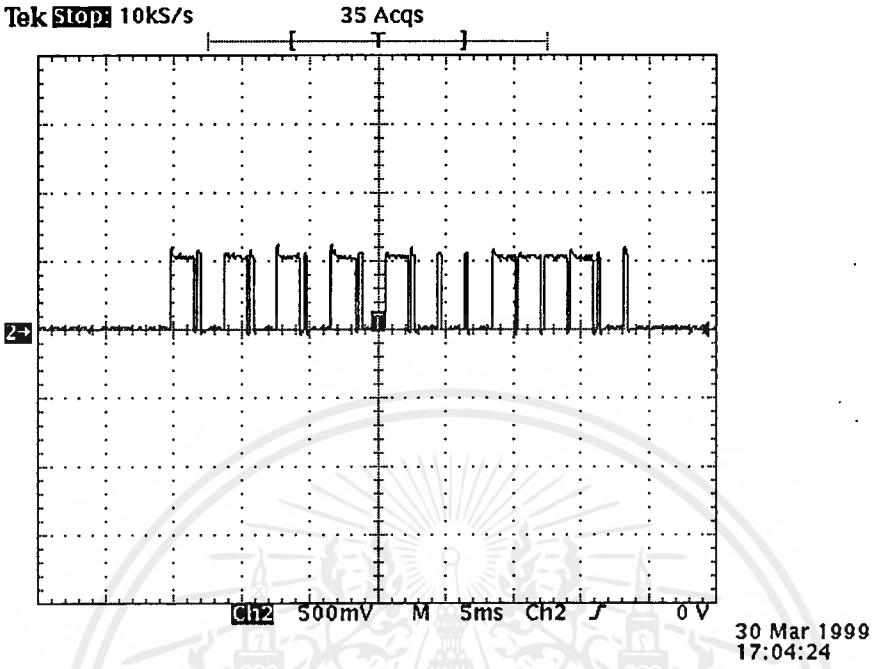


ภาพที่ 4.4 Channel 3. XXXXX0100

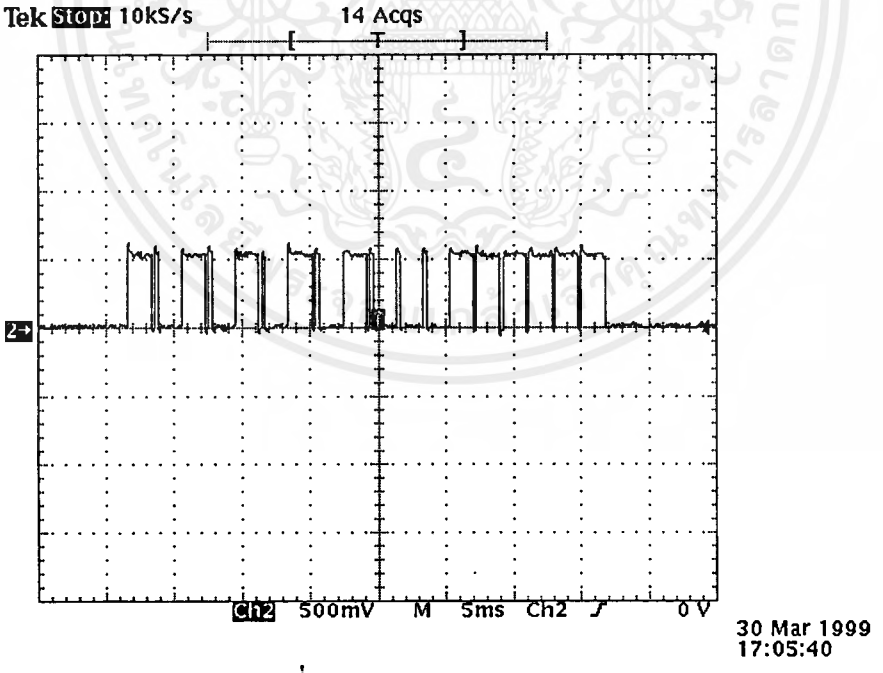


ภาพที่ 4.5 Channel 4. XXXXX0101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

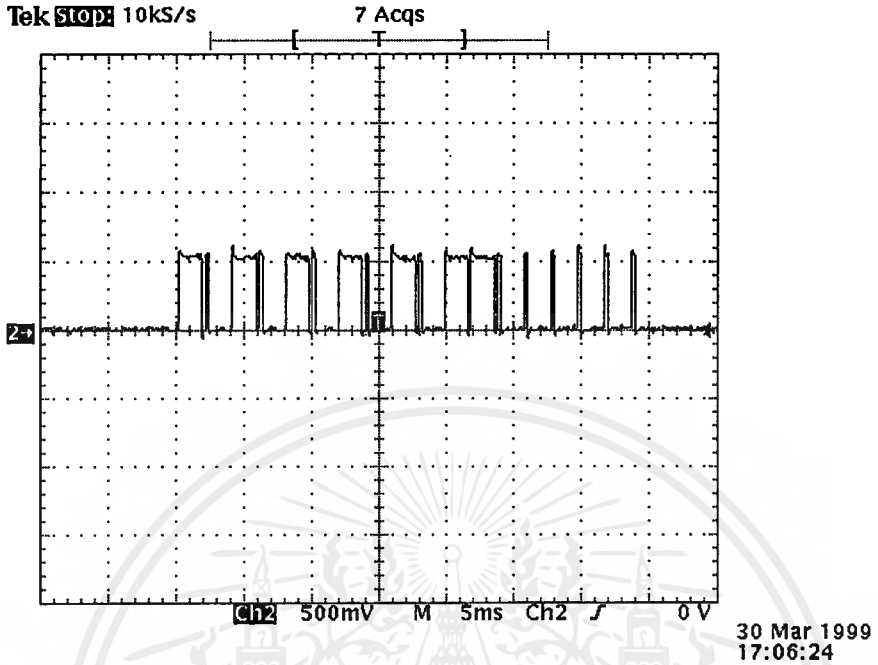


ภาพที่ 4.6 Channel 5. XXXXX0110

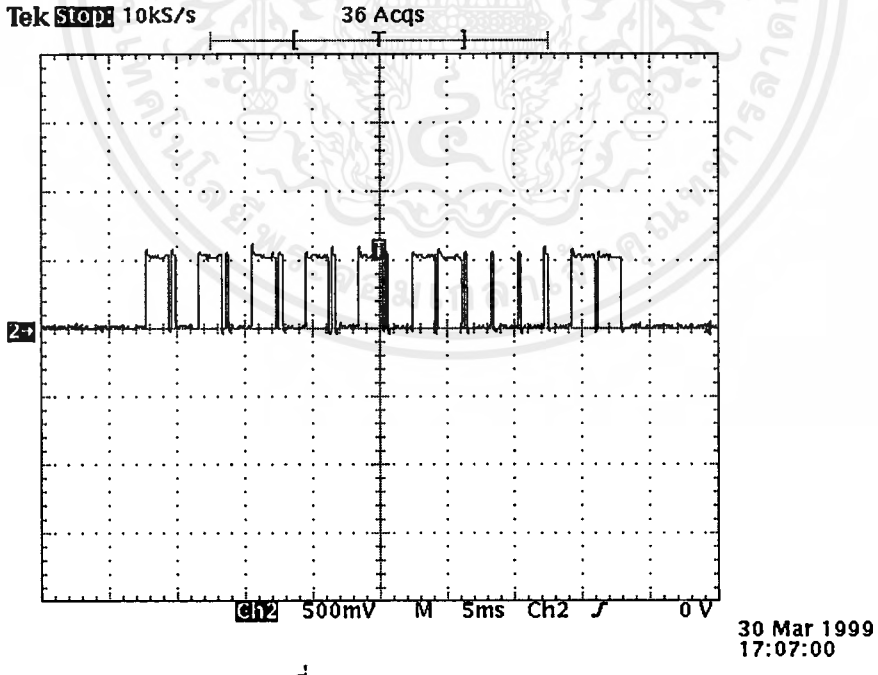


ภาพที่ 4.7 Channel 6. XXXXX0111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

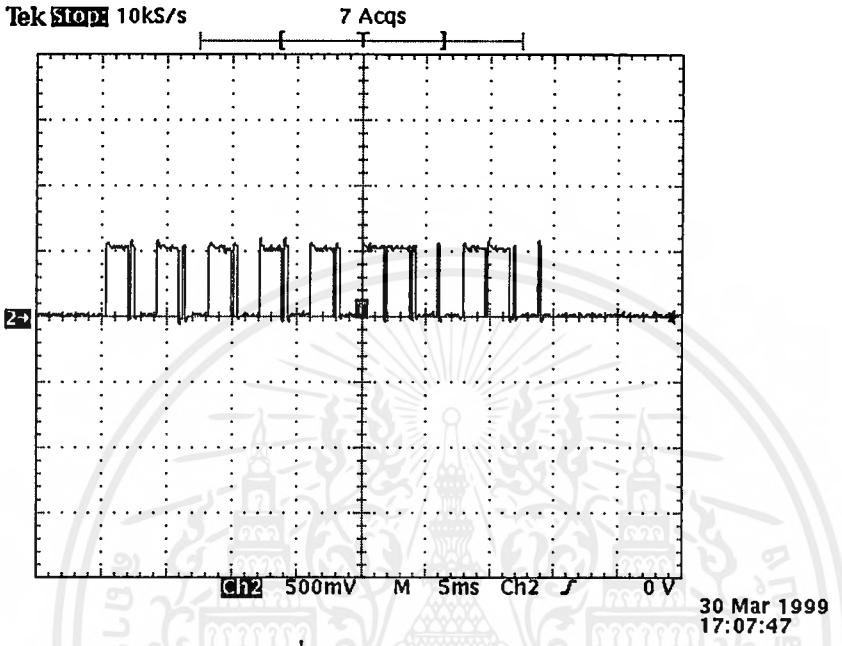


ภาพที่ 4.8 Channel 7. XXXXX1000

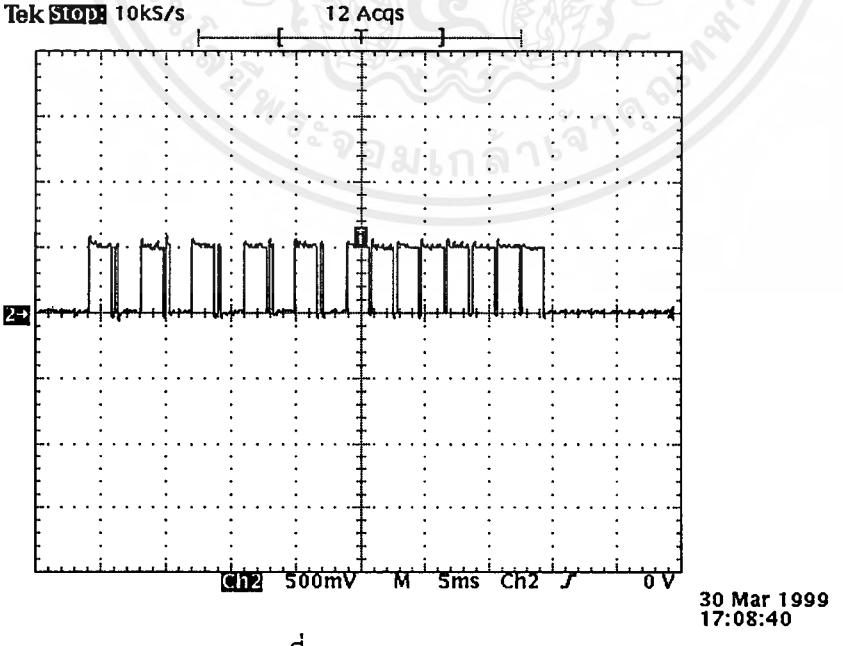


ภาพที่ 4.9 Channel 8. XXXXX1001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

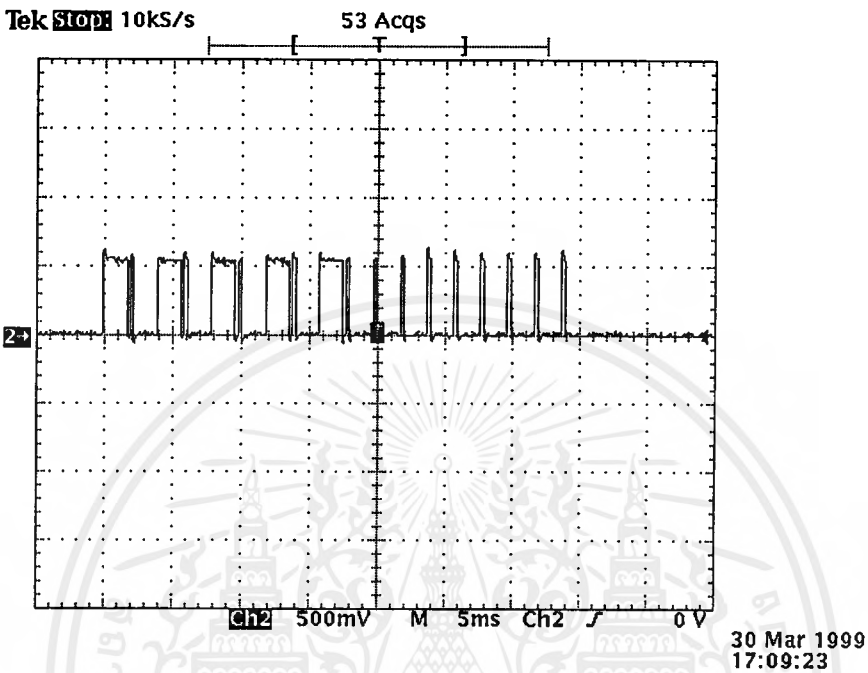


ภาพที่ 4.10 Channel 9. XXXXX1010



ภาพที่ 4.11 Channel Order XXXXX1111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.12 Channel Delete XXXXX0000

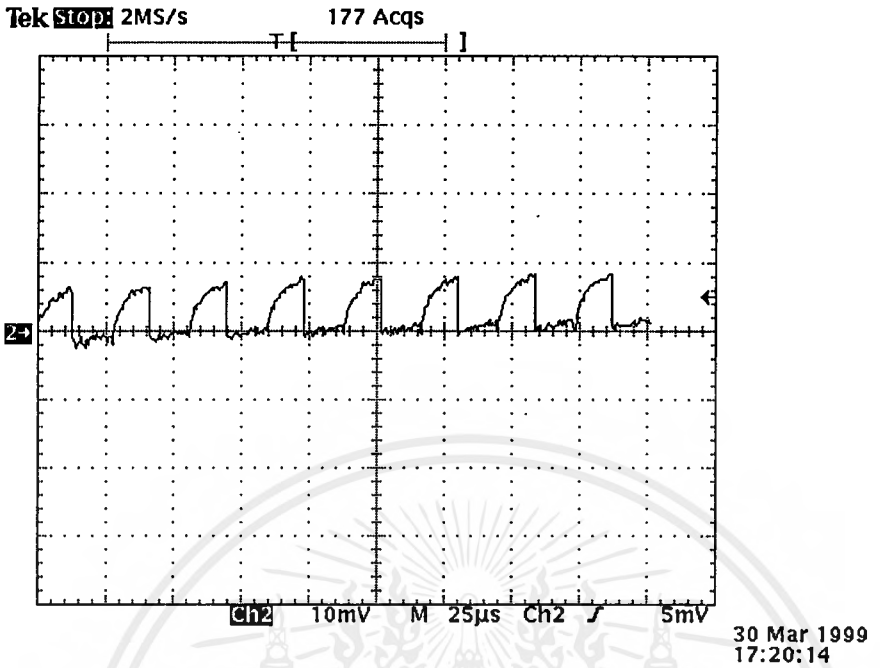
หมายเหตุ : X คือ open

1 คือ ลอจิก 1

0 คือ ลอจิก

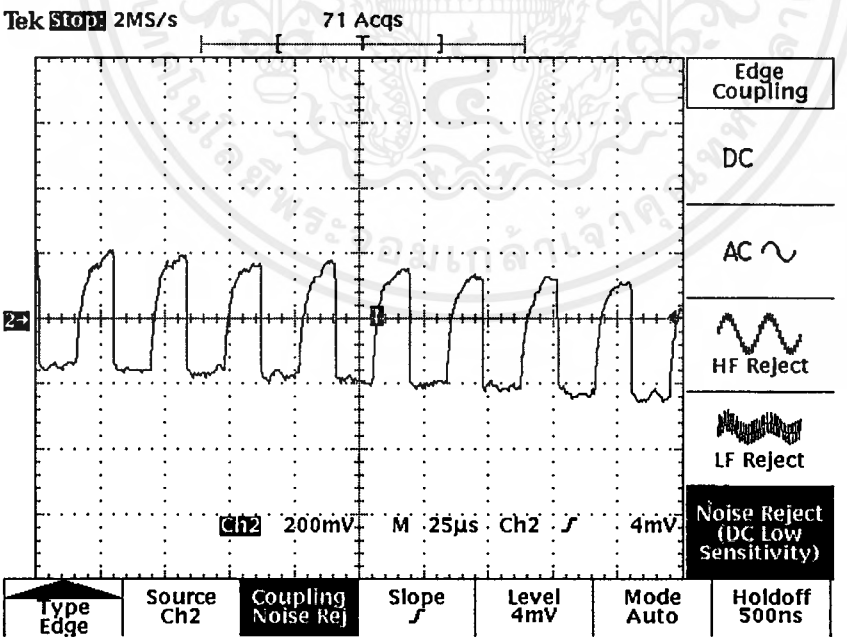
2 ทำการวัด output ของวงจรอะสเตเบิ้ลซึ่งทำหน้าที่เป็นแหล่งกำเนิดคลื่นความถี่พาหะ โดยวัดที่ขาคอลเลคเตอร์ของ TR3 จะได้ output ดังภาพที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.13 แสดงคลื่นความถี่พาหะ

3 นำ Dout ที่ได้จากข้อ 1 มาต่อเป็น input ให้กับวงจรกำเนิดคลื่นพาหะในข้อ 2 วัดสัญญาณ output ได้ดังภาพที่ 4.14

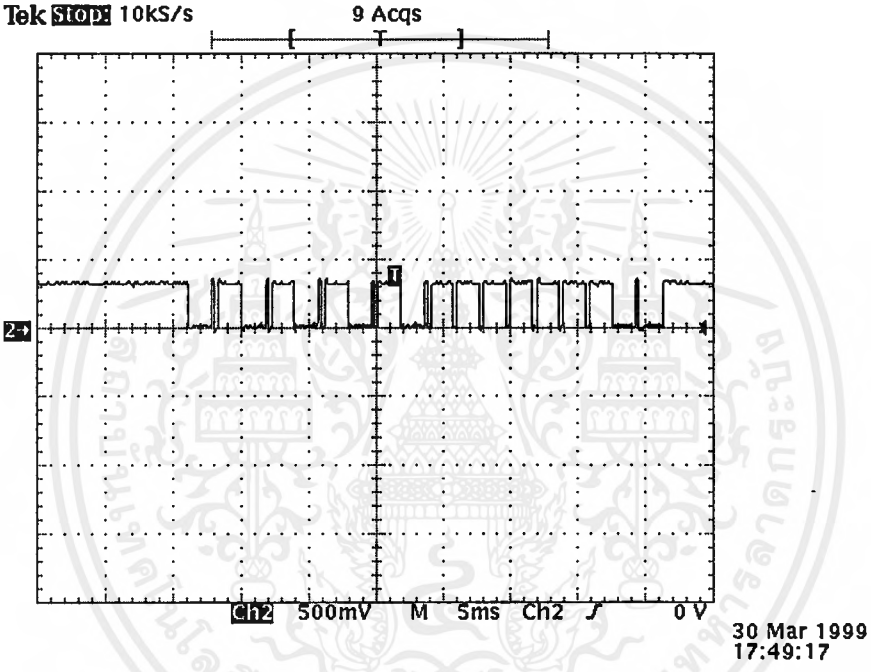


ภาพที่ 4.14 แสดงสัญญาณ output เมื่อให้สัญญาณ input กับวงจรกำเนิดคลื่นพาหะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

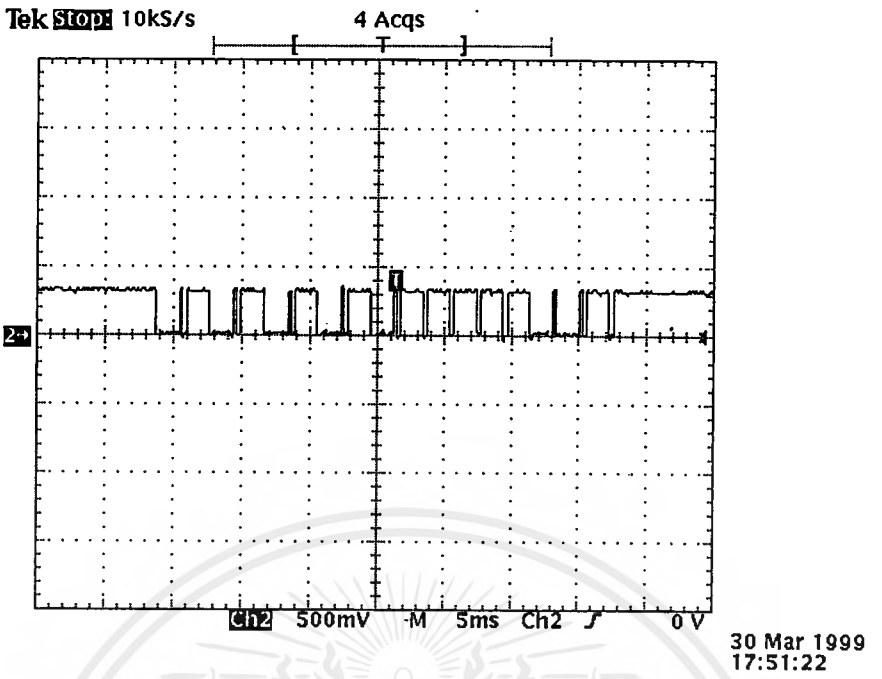
เมื่อนำ output ที่ขา 15 ของตัวเข้ารหัสมารวมกับคลื่นความถี่พาหะประมาณ 40 KHz จะเกิดการเปลี่ยนแปลงเพียงเล็กน้อยสัญญาณที่ขาคอลเลคเตอร์ของ TR3 จึงมีลักษณะไม่แตกต่างกันมากนักซึ่งรูปของสัญญาณตั้งแต่ CH1-CH12 จะมีลักษณะคล้ายภาพที่ 4.14 นั้นเอง

4 ได้ถูกดีเทคออกไป ยังคงเหลือเพียงรหัสที่ถูกส่งมาจาก MC145026 เท่านั้นที่สามารผ่านออกมาได้ แต่กราฟของสัญญาณจะมีลักษณะเป็นอินเวอร์สกับกราฟที่ได้ในข้อ 1 และมีขนาดของแรงดันลดลงประมาณครึ่งหนึ่งด้วยดังภาพที่ 4.15-4.26

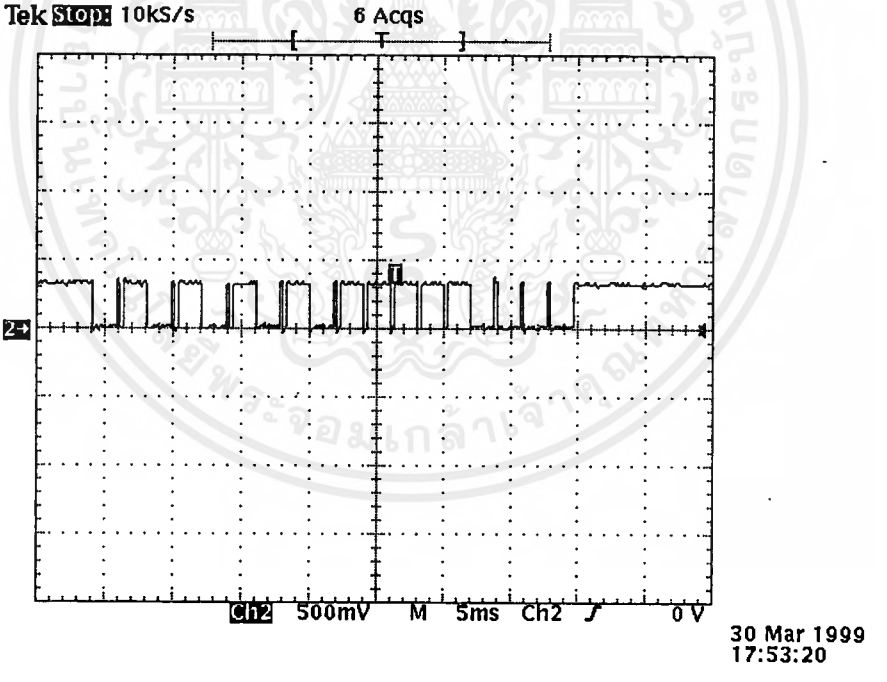


ภาพที่ 4.15 Channel 0. XXXXX0001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

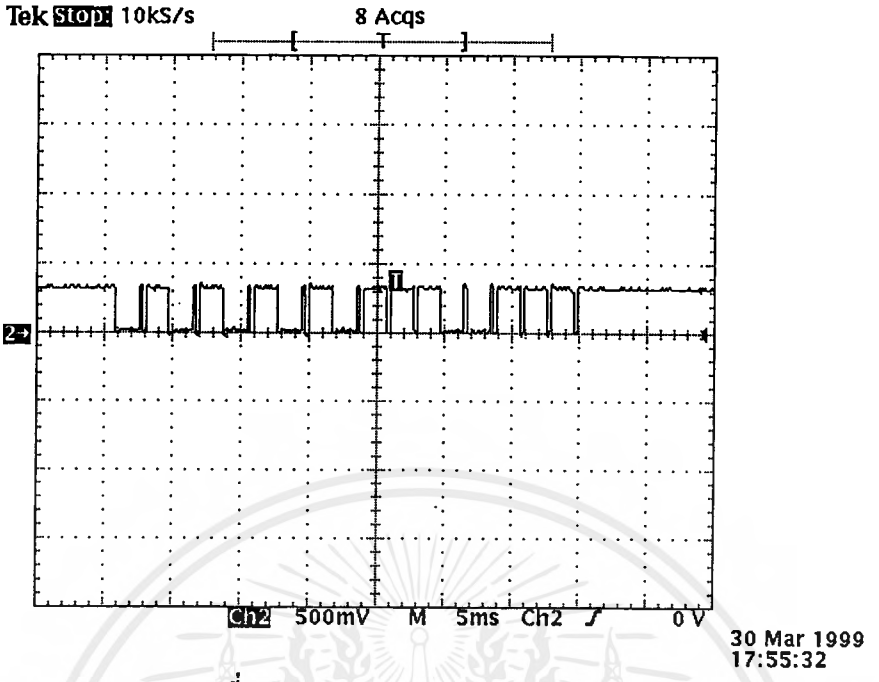


ภาพที่ 4.16 Channel 1. XXXXX0010

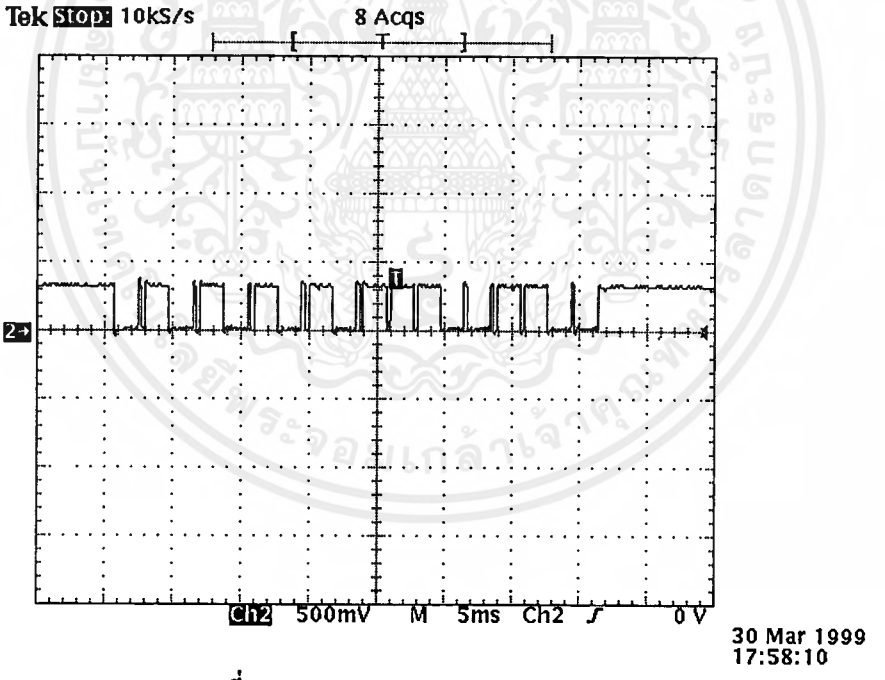


ภาพที่ 4.17 Channel 2. XXXXX0011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

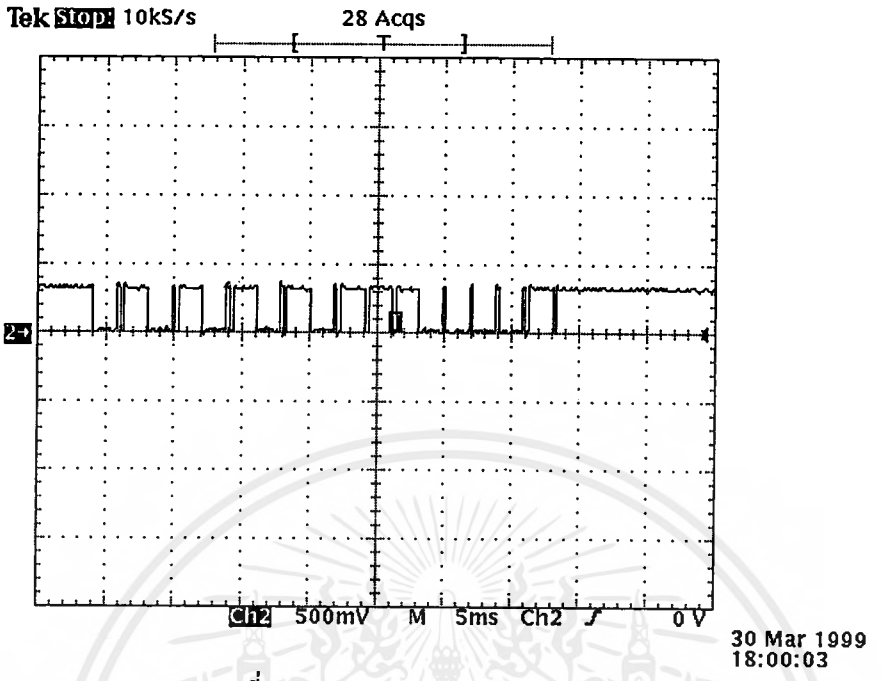


ภาพที่ 4.18 Channel 3. XXXXX0100

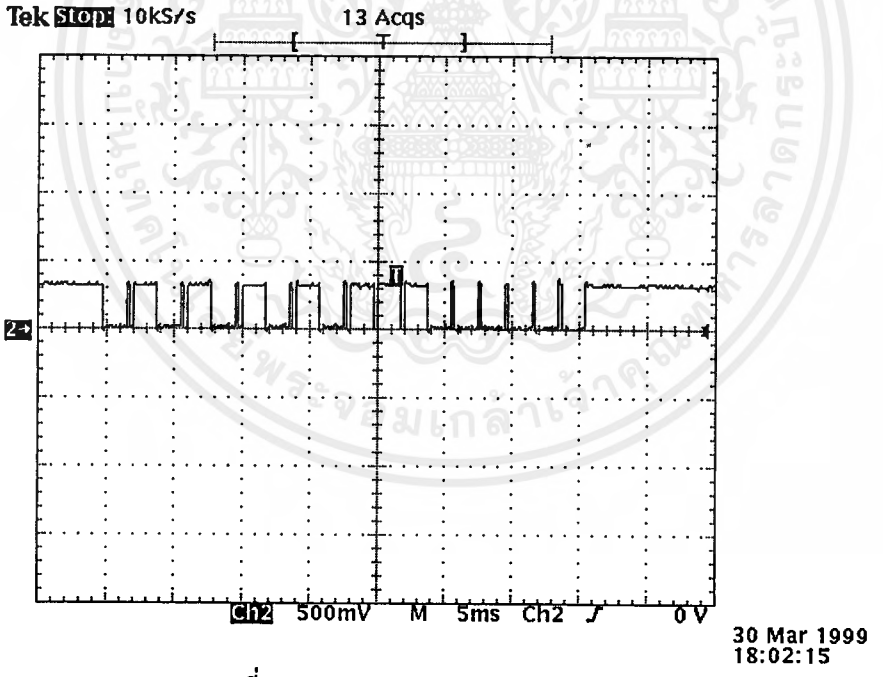


ภาพที่ 4.19 Channel 4. XXXXX0101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

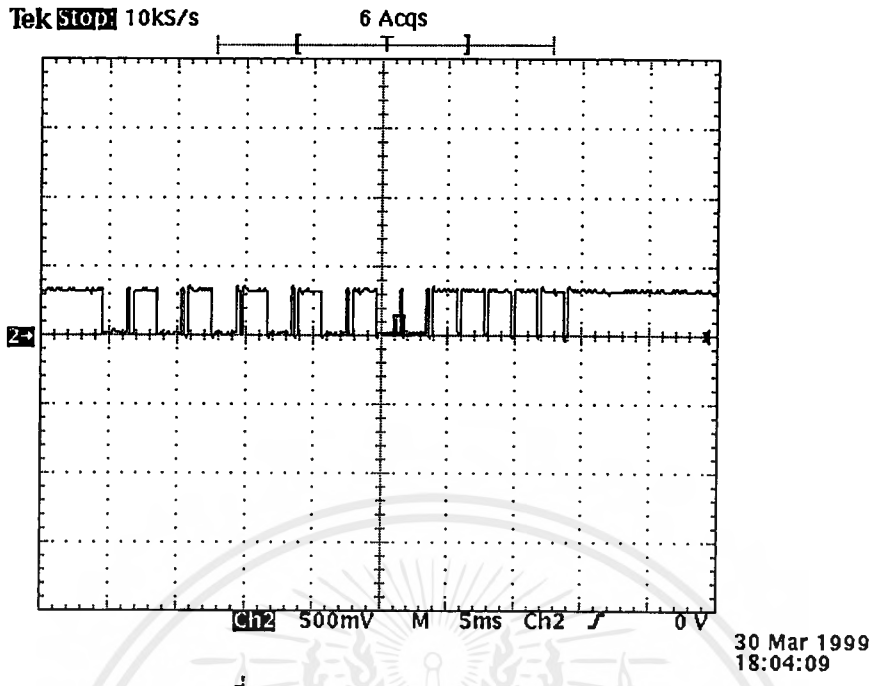


ภาพที่ 4.20 Channel 5. XXXXX0110

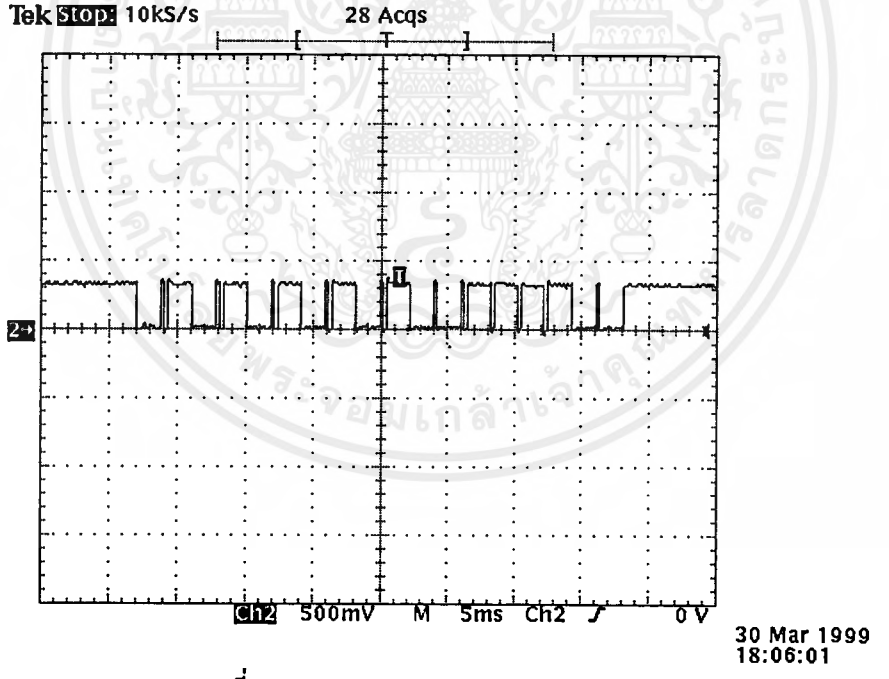


ภาพที่ 4.21 Channel 6. XXXXX0111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

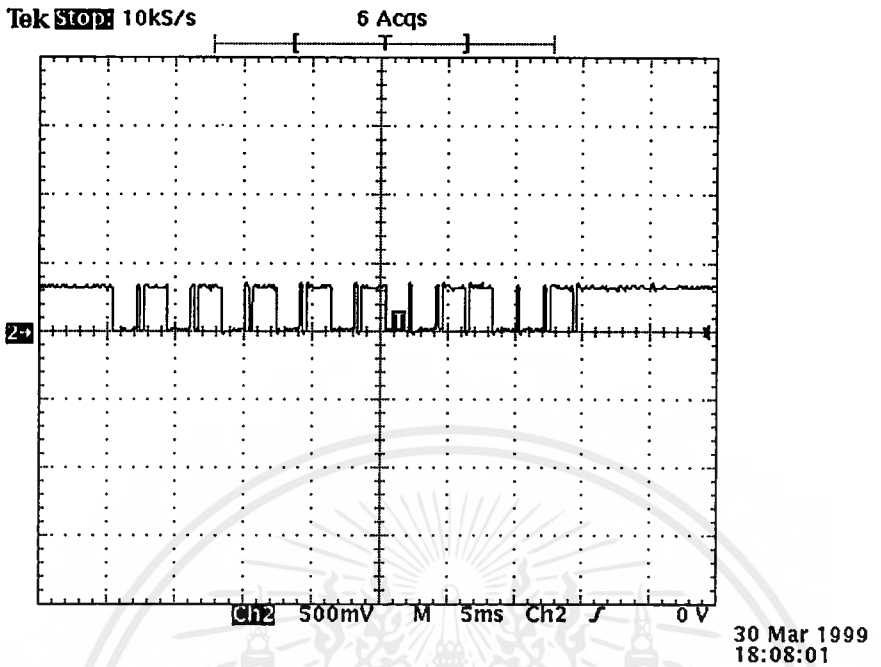


ภาพที่ 4.22 Channel 7. XXXXX1000

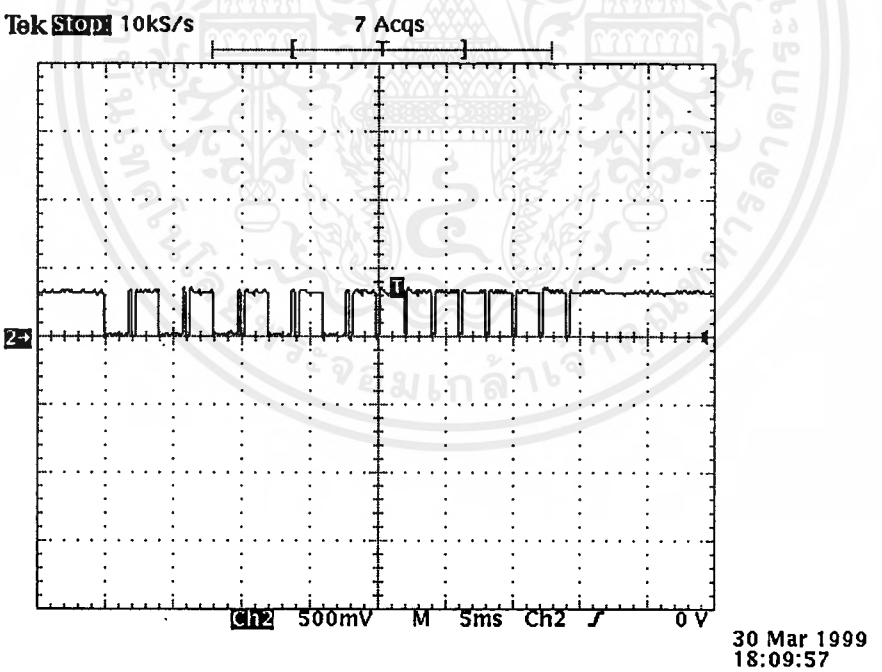


ภาพที่ 4.23 Channel 8. XXXXX1001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

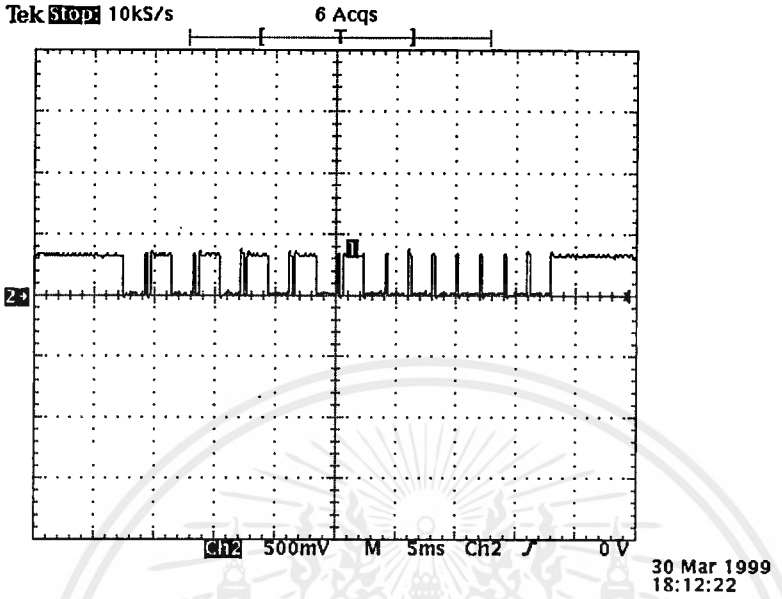


ภาพที่ 4.24 Channel 9. XXXXX1010



ภาพที่ 4.25 Channel Order. XXXXX1111

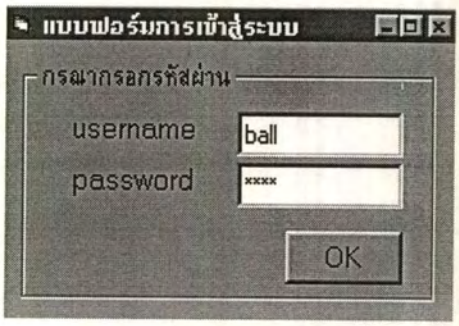
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.26 Channel delete. XXXXX0000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

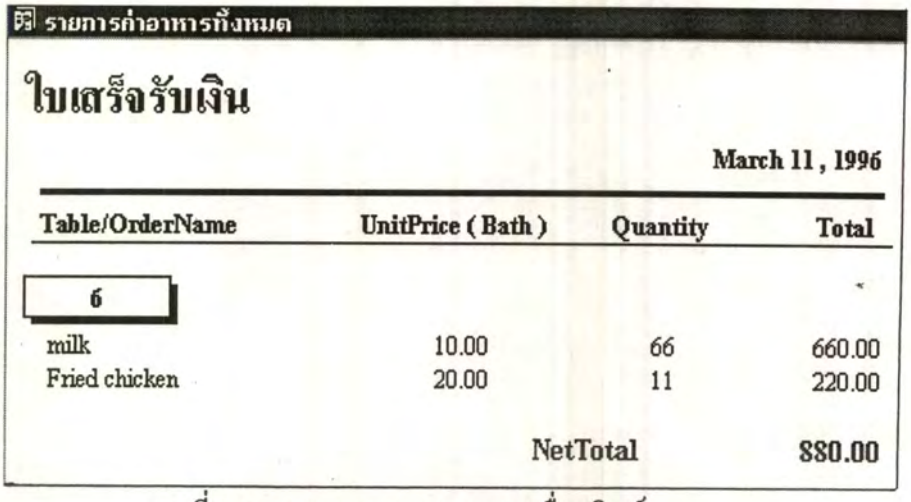
หน้าจอแสดงผลการทำงานใน Visual Basic



ภาพที่ 4.27 แสดงการใส่ Password ก่อนเข้าสู่การทำงาน

ภาพที่ 4.28 แสดงรายการอาหารที่ได้สั่งออกทางหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


รายการค่าอาหารทั้งหมด

ใบเสร็จรับเงิน

March 11 , 1996

Table/OrderName	UnitPrice (Bath)	Quantity	Total
6			
milk	10.00	66	660.00
Fried chicken	20.00	11	220.00
	NetTotal		880.00

ภาพที่ 4.29 แสดงรายการออกทางเครื่องพิมพ์

Order : Table

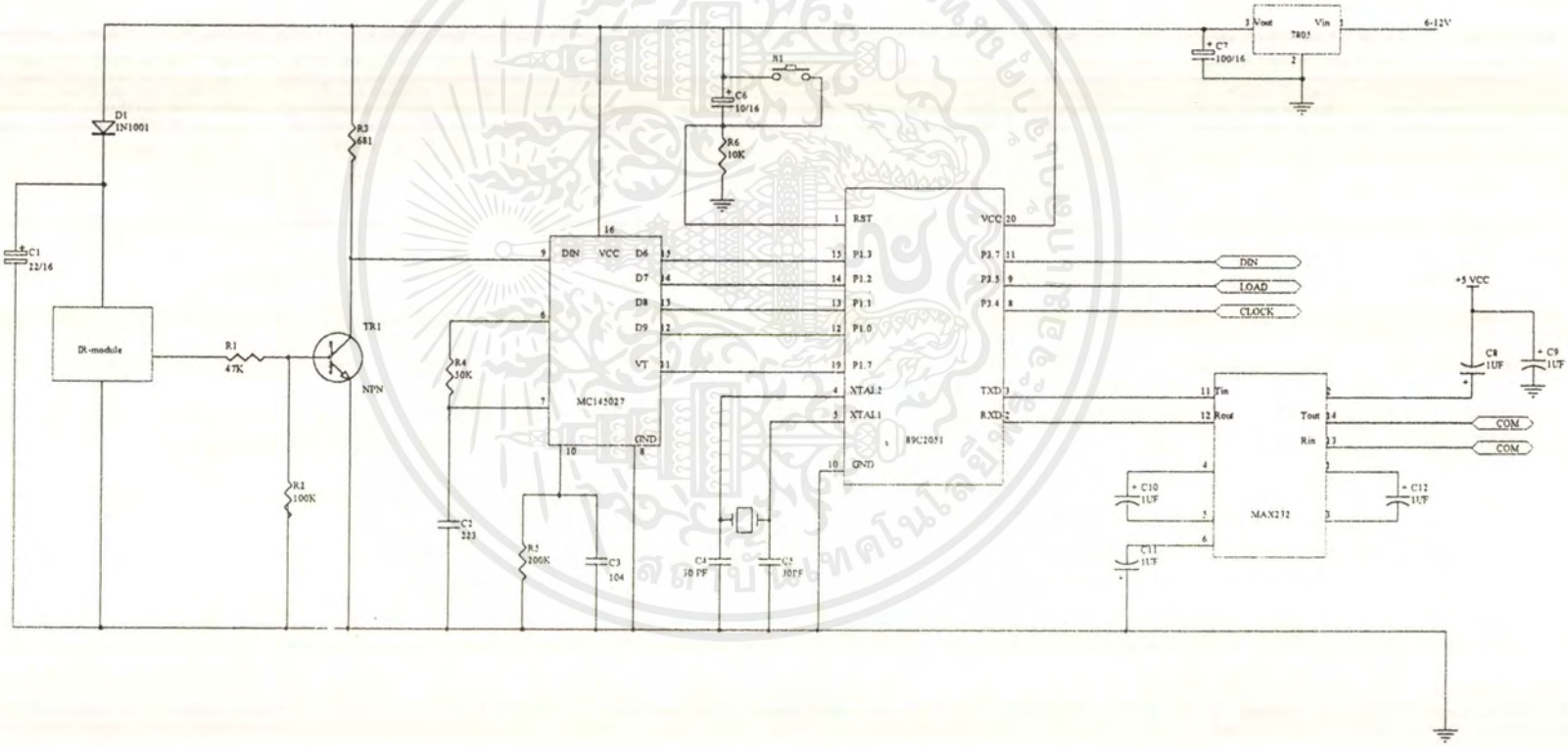
Order	TableID	OrderID	Quantity	Check
1	5	001	12	☐
2	5	002	12	☐
3	5	003	20	☐
4	4	001	11	☐
5	4	003	9	☐
6	4	002	13	☐
7	6	001	66	☑
8	6	003	11	☑
9	7	004	1	☑

ภาพที่ 4.30 แสดงรายการที่เก็บเป็นฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

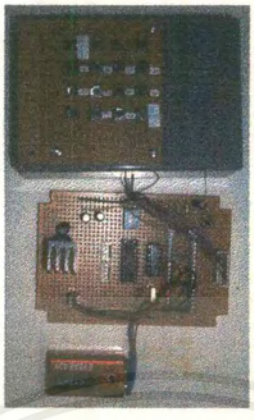
ภาพที่ 4.31 วงจรรับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

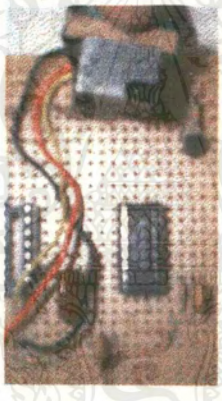


ภาพที่ 4.32วงจรภาครับ

ลิขสิทธิ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

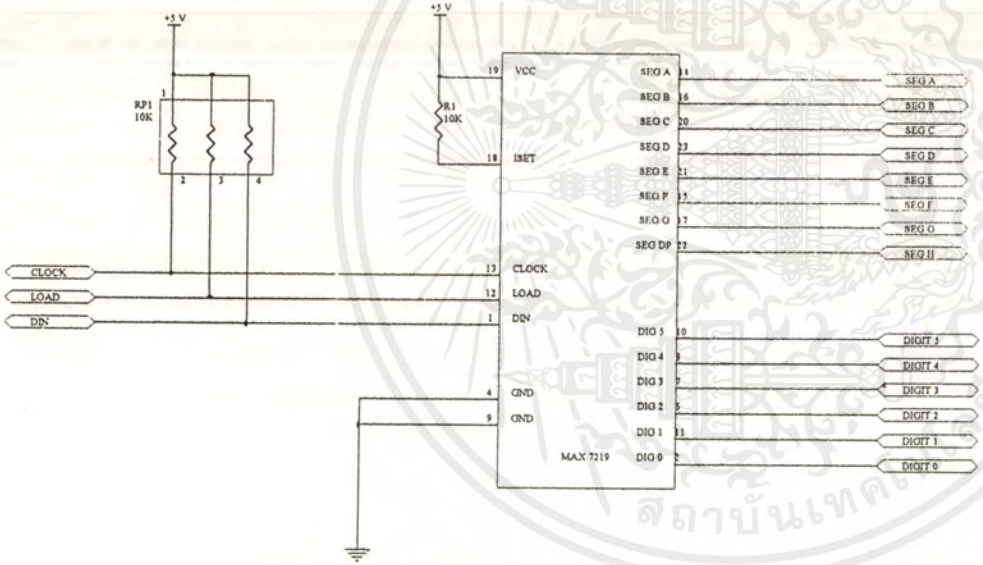


ภาพที่ 4.33 แสดงวงจรภาคส่งที่ใช้ในโครงงาน



ภาพที่ 4.34 แสดงวงจรภาครับที่ใช้ในโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.35 ภาคควบคุมการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.36 แสดงภาคควบคุมการแสดงผลที่ใช้ในโครงงาน

ภาพที่ 4.37 แสดงวงจรของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.38 แสดงชุดรีโมทคอนโทรลสั่งอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 สรุปผลการทดลอง

ปัจจุบันเทคโนโลยีต่างๆ ที่มีอยู่ยังไม่ได้ถูกนำมาใช้ประโยชน์อย่างเต็มที่ โครงงานนี้จึงถูกออกแบบและพัฒนาขึ้นมาเพื่อนำเอาเทคโนโลยีมาสร้างระบบ การสั่งอาหารด้วยรีโมตคอนโทรล ซึ่งประกอบด้วยสามส่วนคือ การส่งและรับข้อมูล ด้วยสัญญาณอินฟราเรด การจัดการข้อมูลและแสดงผลเบื้องต้นโดยไมโครคอนโทรลเลอร์ และการจัดการระบบฐานข้อมูลด้วยโปรแกรมคอมพิวเตอร์ ซึ่งโครงงานนี้ได้พัฒนาจนสำเร็จตามจุดประสงค์ คือสามารถสร้างระบบที่มีความเที่ยงตรงสูง ใช้งานง่าย มีต้นทุนต่ำ ดังมีรายละเอียดในส่วนต่างๆ ดังนี้

การส่งและรับข้อมูลด้วยสัญญาณอินฟราเรด

(ก) การส่งและรับสัญญาณเป็นแบบ PPM

(ข) ระยะทางสูงสุดที่ทำการส่งข้อมูลได้อย่างถูกต้องประมาณ 9 เมตร

(ค) ความถี่ที่ใช้ในการส่งประมาณ 40 KHz

การจัดการข้อมูลและแสดงผลเบื้องต้น

(ก) ใช้ไมโครคอนโทรลเลอร์ในการจัดการ โดยทำหน้าที่ตรวจสอบข้อมูล และจัดการในด้านการแสดงผล โดยควบคุม LED แบบ 7 ส่วนเพื่อให้ผู้ใช้ทราบสถานะของการใช้งานว่าอยู่ในขั้นตอนใดการจัดการระบบฐานข้อมูล

(ข) ใช้โปรแกรม VISUAL BASIC V5.0 ในการจัดการระบบฐานข้อมูล

4.3 ข้อเสนอแนะและวิจารณ์

โครงงานนี้ส่วนสำคัญคือการส่ง และการรับสัญญาณในย่านอินฟราเรด ซึ่งมีรายละเอียดที่จะต้องคิดและออกแบบมาก ตั้งแต่วิธีการส่งข้อมูลที่จะทำให้ข้อมูลที่ถูกส่งออกมา มีความแม่นยำสูง ซึ่งอาจต้องมีการตรวจสอบด้วยวิธีใดวิธีหนึ่งเช่น การตรวจสอบทางซอฟร์แวร์ หรือ การตรวจสอบทางฮาร์ดแวร์ เพราะว่าเกิดการรบกวนสัญญาณได้ง่ายนั่นเอง แต่หากมีการตรวจสอบที่ดีแล้ว การรับส่งข้อมูลด้วยสัญญาณอินฟราเรดนี้จะสะดวกและง่ายต่อการประยุกต์ใช้งานมาก อีกทั้งยังมีต้นทุนที่ต่ำอีกด้วย

ข้อจำกัดของโครงงานคือ ระยะทางในการส่งข้อมูล และ จำนวนข้อมูลต่อการส่งหนึ่งครั้งแนวทางพัฒนาคือ

(ก) ออกแบบวงจรที่สามารถส่งข้อมูลได้ไกลและมีความแม่นยำสูง

(ข) ออกแบบส่วนจัดการข้อมูลและแสดงผลที่สามารถตอบสนองความต้องการในการใช้งานได้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] พิพัฒน์ เลาหสงคราม "ไมโครคอนโทรลเลอร์ MCS-48 MCS-51" ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

[2] ไกรวุฒิ โรจน์ประเสริฐสุด "ไมโครโปรเซสเซอร์2" บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน) กรุงเทพฯ

[3] สุทธิศักดิ์ พงศ์ธนาพานิช "Visual Basic 5.0" บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2541 กรุงเทพฯ

[4] กิตติ ภักดีวัฒนะกุล จำลอง ครูอุตสาหะ "Visual Basic 5.0 ฉบับโปรแกรมเมอร์" บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2541 กรุงเทพฯ

[5] "รีโมท คอนโทรลไร้สาย" บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2541 กรุงเทพฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

รายการอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอาหารวันนี้

ลำดับ	รายการอาหาร	ราคา
01	Fried Rice with Pork	30.00
02	Fried Rice with Seafood	30.00
03	Noodle	30.00
04	Tuna Sandwich	20.00
05	Egg Sandwich	20.00
06	Onion Soup	20.00
07	Chicken Soup	20.00
08	Pasta	20.00
09	Pizza	20.00
10	Lemon Pork	30.00
11	Toast Pork	30.00
12	Toast Chicken	30.00
13	Fish Grill	30.00
14	Seafood Salad	20.00
15	Chicken Salad	30.00
16	Hamberger	30.00
17	Orange Juice	10.00
18	Lemon Juice	10.00
19	Coke	10.00
20	Icecream	20.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

Source Code ของโปรแกรม Visual Basic และ ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code ของโปรแกรม Visual Basic

```
Option Explicit
Private mdb As Database
Private mrs, mrs1 As Recordset
Private Sub cmdQuit_Click()
   End
End Sub
Private Sub cmdTotal_Click()
On Error GoTo ProcError
Dim str As String
Dim test As Boolean

  str = "{Order.TableID} = '" & Text4.Text & "' "
  str = str & " and {Order.Check} = " & False
  crptReceipt.SelectionFormula = str
  crptReceipt.Action = 1

  If mrs.RecordCount > 0 Then
    mrs.MoveFirst
    Do Until mrs.EOF
      If mrs![TableId] = Text4.Text Then
        mrs.Edit
        mrs![Check] = True
        mrs.Update
      End If
      mrs.MoveNext
    Loop
  End If
'---------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    List1.Clear
    Set mrs1 = mdb.OpenRecordset("view1", dbOpenDynaset, dbSeeChanges)

    If mrs1.RecordCount > 0 Then
      mrs1.MoveFirst
      Do Until mrs1.EOF
        If mrs1![Check] = False Then
          List1.AddItem "   " & mrs1![TableId] & "        :  " & _
          mrs1![OrderName] & " :  " & mrs1![Quantity]
        End If
      mrs1.MoveNext
      Loop
    End If
'-------------------------------------


ProcExit:
   Exit Sub
ProcError:
   MsgBox Err.Number & vbCrLf & Err.Description
Resume ProcExit
End Sub
Private Sub Form_Load()

Timer1.Enabled = True

MSComm1.PortOpen = True



On Error GoTo ProcError
   Dim strDBName As String
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    strDBName = "c:/My Documents/tom.mdb"
    Set mdb = DBEngine.Workspaces(0).OpenDatabase(strDBName)
    Set mrs = mdb.OpenRecordset("Order", dbOpenDynaset, dbSeeChanges)
    Set mrs1 = mdb.OpenRecordset("view1", dbOpenDynaset, dbSeeChanges)

    If mrs1.RecordCount > 0 Then
      mrs1.MoveFirst
      Do Until mrs1.EOF
        If mrs1![Check] = False Then
          List1.AddItem "   " & mrs1![TableId] & "       :  " & _
          mrs1![OrderName] & " :  " & mrs1![Quantity]
        End If
      mrs1.MoveNext
      Loop
    End If

ProcExit:
  Exit Sub
ProcError:
  MsgBox Err.Number & vbCrLf & Err.Description
  Resume ProcExit
End Sub

Private Sub Timer1_Timer()
Dim a, T, O, Q, x, y As String
Timer1.Enabled = False
a = MSComm1.Input
x = Mid$(a, 1, 1)
y = Mid$(a, 8, 1)
If x = ">" And y = "<" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
T = Mid$(a, 2, 2)
O = Mid$(a, 4, 2)
Q = Mid$(a, 6, 2)
If Data1.Enabled Then
Data1.Recordset.AddNew
Data1.Recordset!TableId = Val(T)
Data1.Recordset!OrderID = Val(O)
Data1.Recordset!Quantity = Val(Q)
Data1.Recordset.Update
Data1.Refresh
List1.Clear
  Set mrs1 = mdb.OpenRecordset("view1", dbOpenDynaset, dbSeeChanges)
    If mrs1.RecordCount > 0 Then
    mrs1.MoveFirst
    Do Until mrs1.EOF
      If mrs1![Check] = False Then
        List1.AddItem "   " & mrs1![TableId] & "       :  " & _
        mrs1![OrderName] & " :  " & mrs1![Quantity]
      End If
    mrs1.MoveNext
    Loop
  End If
End If
End If
Timer1.Enabled = True
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาษาแอสเซมบลี (Assembly) สำหรับ AT89C2051 ไมโครคอนโทรลเลอร์

```
DSP_BUFF    EQU   40H
STORE       EQU   60H
DATA        EQU   P3.7
LOAD        EQU   P3.4
CLOCK       EQU   P3.5
WAITBIT1    EQU   39H
WAITBIT2    EQU   38H
WAITBIT3    EQU   37H
LOOPBIT1    EQU   36H
LOOPBIT2    EQU   35H
LOOPBIT3    EQU   34H

;************************
;*START PROGRAM      *
;************************

            ORG     0000H
            CLR     LOAD
            CLR     CLOCK
            SETB    DATA
            MOV     P1,#0FFH
            CALL    SETMAX
            MOV     R4,#00H
            CALL    INT_DSP
START:      MOV     R2,#20
            MOV     R0,#DSP_BUFF
            MOV     A,#0FH
LINE:       MOV     @R0,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                INC         R0
                DJNZ        R2,LINE

;**********************
;*MAIN PROGRAM      *
;**********************

STANDBY:        JNB         P1.7,$
                CALL        XALL
MAIN:           MOV         A,P1
                ANL         A,#0FH
                MOV         R1,A

                MOV         A,R1
CK_ORDER:       CJNE        A,#00001111B,CK_DELETE
                CALL        CHECK6
                MOV         A,R7
                JNZ         GO_STDB
                JMP         ORDER
GO_STDB:        CALL        DSP_ORDER
                JMP         STANDBY
CK_DELETE:      MOV         A,R1
                CJNE        A,#00000000B,MAIN3
                CALL        DEL_ERROR
                JMP         DELETE

MAIN3:          NOP
CK_DATA:        MOV         A,R4
                CJNE        A,#06H,CK_IT
                CALL        DSP_ORDER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JMP     STANDBY
CK_IT:  JC      KEEP
KEEP:   MOV     R0,#STORE
        MOV     A,R0
        ADD     A,R4
        MOV     R0,A
        MOV     A,R1
        DEC     A
        MOV     @R0,A

        MOV     R0,#DSP_BUFF
        MOV     A,R0
        ADD     A,R4
        MOV     R0,A
        MOV     A,R1
        DEC     A
        MOV     @R0,A
        MOV     P1,#0FFH
        INC     R4
SHOW:   CALL    DSP_ORDER
        JMP     STANDBY

;************************
;*SUB PROGRAM          *
;************************

ORDER:  MOV     SCON,#50H
        MOV     TMOD,#20H
        MOV     TH1,#0FDH
        MOV     TR1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     R0,#5
        DJNZ    R0,$
        MOV     R5,#00H
        MOV     A,R4
        MOV     R5,A
        MOV     R0,#STORE
        MOV     A,#3EH
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
TRANSF: MOV     A,@R0           ;SEND DATA TO COMPUTER
        ORL     A,#00110000B
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        INC     R0
        DJNZ    R5,TRANSF
        MOV     A,#3CH
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        CALL    CLEAR
        MOV     P1,#0FFH
        CALL    SAY_ORDER
        MOV     R7,#2
SAYORDER1: CALL WAIT
        DJNZ    R7,SAYORDER1
        MOV     DPTR,#09FFH
        CALL    SHIFT
        CALL    XALL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JMP     START
CHECK6: MOV     A,R4
        CJNE    A,#00000110B,NOT6
        MOV     R7,#00H
        RET
NOT6:   MOV     R7,#00H
        RET
CLEAR:  MOV     A,R4
        MOV     R5,A
        MOV     R0,#STORE
        MOV     A,#00H
CLEAR1: MOV     @R0,A
        INC     R0
        DJNZ    R4,CLEAR1
        MOV     R0,#DSP_BUFF
        MOV     A,#0FH
CLEAR2: MOV     @R0,A
        INC     R0
        DJNZ    R5,CLEAR2

        MOV     A,#00H
        MOV     R0,A
        MOV     R1,A
        MOV     R2,A
        MOV     R3,A
        MOV     R4,A
        MOV     R5,A
        MOV     R6,A
        MOV     R7,A
        CLR     SBUF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR     RI
                CLR     TR1
                RET
DSP_ORDER:      MOV     R2,#6
                MOV     R6,#01H
                MOV     R0,#DSP_BUFF
DSP2:           MOV     A,@R0
                MOV     DPH,R6
                MOV     DPL,A
                CALL    SHIFT
                INC     R0
                INC     R6
                DJNZ    R5,DSP2
                RET
SETMAX:         MOV     DPTR,#0A04H
                CALL    SHIFT
                MOV     DPTR,#0B05H
                CALL    SHIFT
                MOV     DPTR,#09FFH
                CALL    SHIFT
                MOV     DPTR,#0C01H
                CALL    SHIFT
                MOV     DPTR,#0F00H
                CALL    SHIFT
                RET
SHIFT:          MOV     R5,#8
                MOV     A,DPH
                CLR     CY
                CLR     CLOCK
                SETB    LOAD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SHIFT1:   RLC     A
          MOV     DATA,C
          SETB    CLOCK
          NOP
          CLR     CLOCK
          DJNZ    R5,SHIFT1
          MOV     R5,#8
          MOV     A,DPL
          CLR     CY
SHIFT2:   RLC     A
          MOV     DATA,C
          SETB    CLOCK
          NOP
          CLR     CLOCK
          DJNZ    R5,SHIFT2
          CLR     LOAD
          NOP
          SETB    LOAD
          RET
DELETE:   MOV     A,R4
          MOV     R5,A
          MOV     R0,#STORE
          MOV     A,R0
          ADD     A,R4
          MOV     R0,A
          MOV     @R0,#00H
          MOV     R0,#DSP_BUFF
          MOV     A,R0
          ADD     A,R4
          DEC     A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     R0,A
        MOV     @R0,#00001111B
        DEC     R4
        CALL    WAIT
        CALL    DSP_ORDER
        JMP     STANDBY
WAIT:   MOV     WAITBIT1,#5
        MOV     WAITBIT2,#5
        MOV     WAITBIT3,#5
WAIT1:  DJNZ    WAITBIT2,$
        DJNZ    WAITBIT3,WAIT1
        DJNZ    WAITBIT1,WAIT1
        RET
LONGWAIT: MOV   WAITBIT1,#100
        MOV     WAITBIT2,#100
LW1:    DJNZ    WAITBIT1,$
        DJNZ    WAITTBIT2,LW1
        RET
XALL:   MOV     LOOPBIT1,#6
        MOV     LOOPBIT2,#00H
XXX:    INC     LOOPBIT2
        MOV     DPH,LOOPBIT2
        MOV     DPL,#0FH
        CALL    SHIFT
        DJNZ    LOOPBIT1,XXX
        CALL    WAIT
        RET
SAY_ORDER: NOP
        MOV     DPTR,#0900H
        CALL    SHIFT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     DPTR,#017EH
                CALL    SHIFT
                MOV     DPTR,#0246H
                CALL    SHIFT
                MOV     DPTR,#033DH
                CALL    SHIFT
                MOV     DPTR,#044FH
                CALL    SHIFT
                MOV     DPTR,#0546H
                CALL    SHIFT
                MOV     DPTR,#0600H
                CALL    SHIFT
                RET
DEL_ERROR:      NOP
                MOV     A,R4
                JNZ     HOME
                CALL    SAY_ERROR
                MOV     R7,#2
SAYERROR1:      CALL    WAIT
                DJNZ    R7,SAYERROR1
                MOV     DPTR,#09FFH
                CALL    SHIFT
                CALL    DSP_ORDER
                JMP     STANDBY
SAY_ERROR:      MOV     DPTR,#0900H
                CALL    SHIFT
                MOV     DPTR,#014FH
                CALL    SHIFT
                MOV     DPTR,#0246H
                CALL    SHIFT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

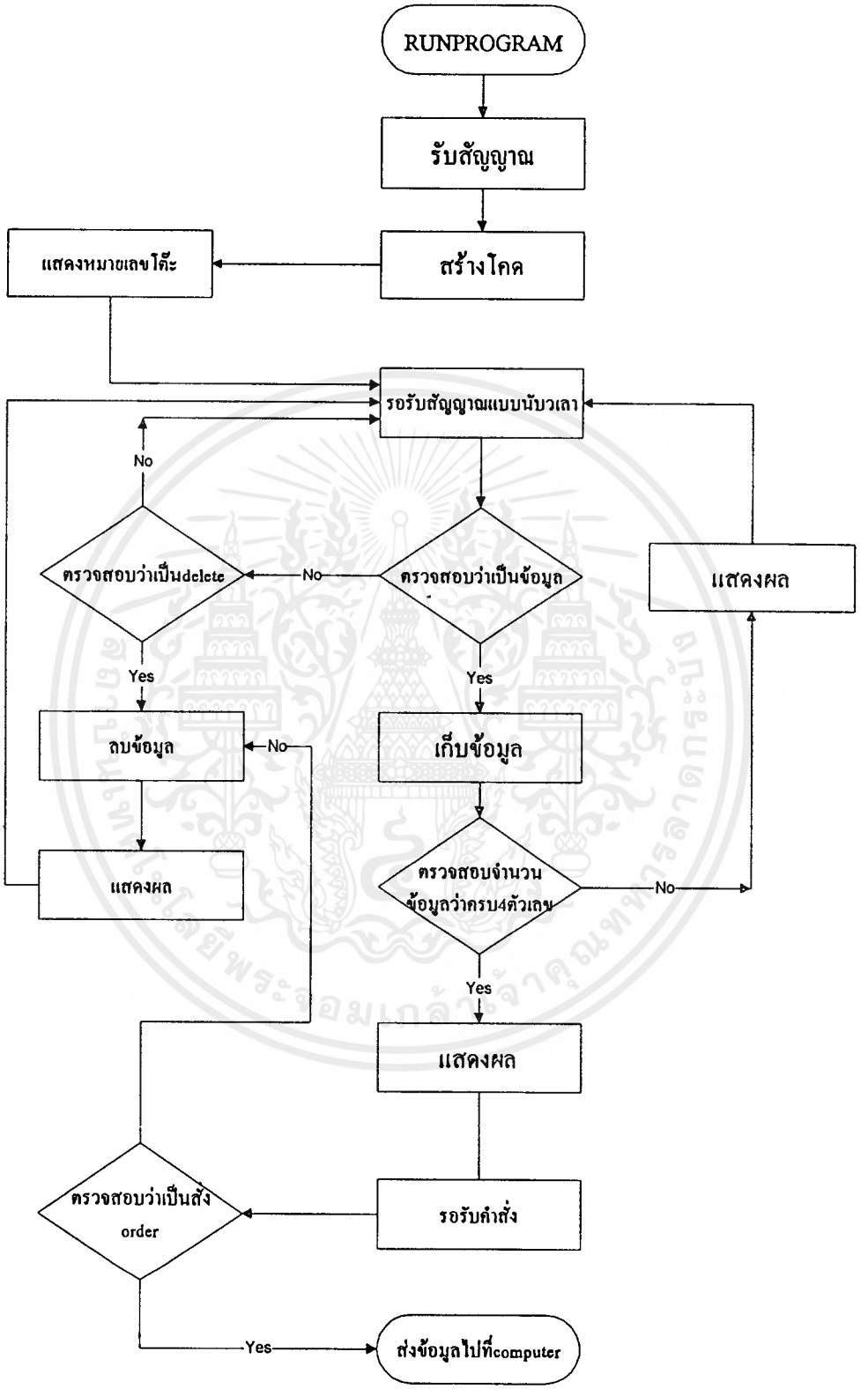
```
        MOV     DPTR,#0346H
        CALL    SHIFT
        MOV     DPTR,#047EH
        CALL    SHIFT
        MOV     DPTR,#0546H
        CALL    SHIFT
        MOV     DPTR,#0600H
        CALL    SHIFT
HOME:   RET
INT_DSP: MOV    DPH,#01H
        MOV     DPL,#0CH
        CALL    SHIFT
        MOV     DPH,#02H
        MOV     DPL,#0BH
        CALL    SHIFT
        MOV     DPH,#03H
        MOV     DPL,#0DH
        CALL    SHIFT
        MOV     DPH,#04H
        MOV     DPL,#0DH
        CALL    SHIFT
        MOV     DPH,#05H
        MOV     DPL,#00H
        CALL    SHIFT
        MOV     DPH,#06H
        MOV     DPL,#0FH
        CALL    SHIFT
        CALL    LONGWAIT
        RET
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

ผังการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



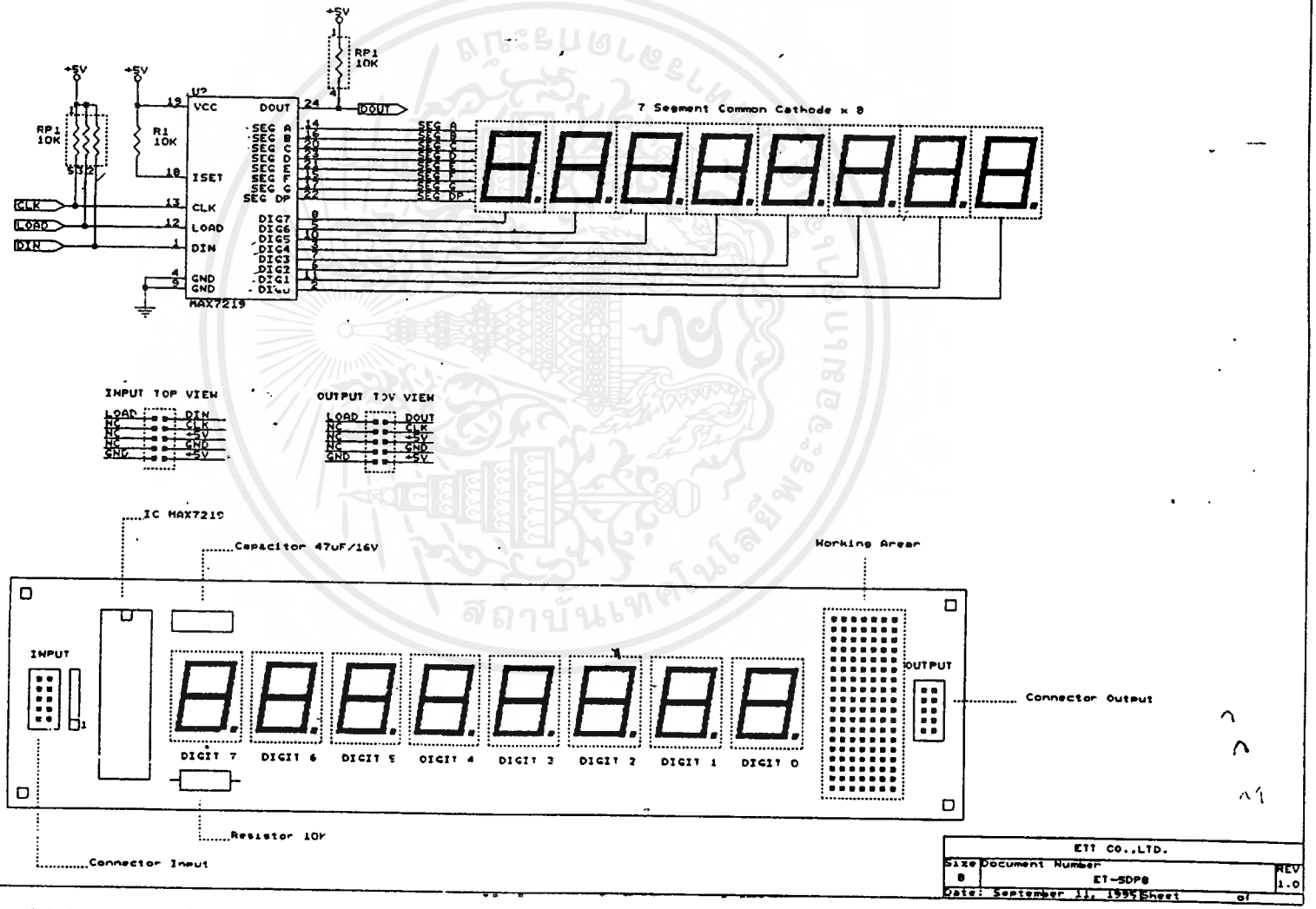
แผนภาพการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

รายละเอียดข้อมูลของไอซีที่ใช้ในโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


+5V
RP1
10K
U2
VCC
DOUT
ISET
R1
10K
CLK
LOAD
DIN
GND
GND
MAX7219
SEG A
SEG B
SEG C
SEG D
SEG E
SEG F
SEG G
SEG DP
DIG7
DIG6
DIG5
DIG4
DIG3
DIG2
DIG1
DIG0
7 Segment Common Cathode x 8
INPUT TOP VIEW
OUTPUT TOP VIEW
IC MAX7219
Capacitor 47uF/16V
Working Area
INPUT
OUTPUT
DIGIT 7
DIGIT 6
DIGIT 5
DIGIT 4
DIGIT 3
DIGIT 2
DIGIT 1
DIGIT 0
Connector Output
Resistor 10K
Connector Input
ETT CO.,LTD.
Size
B
Document Number
ET-SDP8
REV
1.0
Date: September 11, 1995
Sheet
of

19-4452; Rev 1; 5/92

MAXIM

Serially Interfaced, 8-Digit LED Display Driver

General Description

The MAX7219 is a compact, serial input/output common-cathode display driver that interfaces microprocessors (µPs) to 7-segment numeric LED displays of up to 8 digits, bar-graph displays, or 64 individual LEDs. Included on-chip are a BCD code-B decoder, multiplex scan circuitry, segment and digit drivers, and an 8x8 static RAM that stores each digit. Only one external resistor is required to set the segment current for all LEDs.

A convenient 3-wire serial interface connects to all common µPs. Individual digits may be addressed and updated without rewriting the entire display. The MAX7219 also allows the user to select code-B decoding or no-decode for each digit.

The MAX7219 includes a 150µA low-power shutdown mode, analog and digital brightness control, a scan-limit register which allows the user to display from 1 to 8 digits, and a test mode which forces all LEDs on.

Applications

Bar-Graph Displays
7-Segment Displays
Industrial Controllers
Panel Meters
LED Matrix Displays

Features

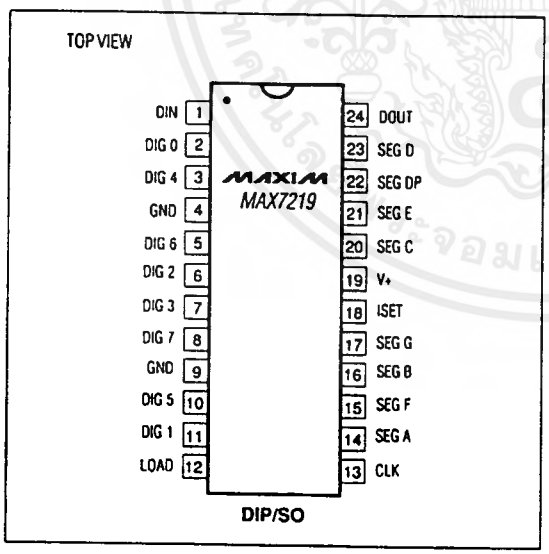
- ♦ 10MHz Serial Interface
- ♦ Individual LED Segment Control
- ♦ Decode/No-Decode Digit Selection
- ♦ 150µA Low-Power Shutdown (Data Retained)
- ♦ Digital and Analog Brightness Control
- ♦ Display Blanked on Power-Up
- ♦ 24-Pin DIP and SO Packages
- ♦ Drives Common-Cathode LED Display

Ordering Information

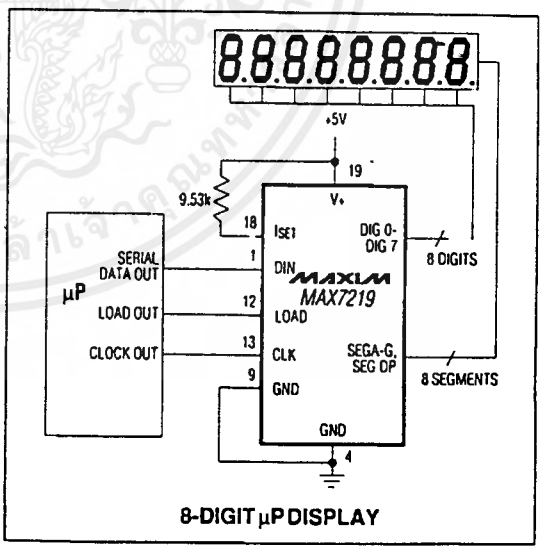
PART	TEMP. RANGE	PIN-PACKAGE
MAX7219CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX7219CWG	0°C to +70°C	24 Wide SO
MAX7219C/D	0°C to +70°C	Dice*
MAX7219ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX7219EWG	-40°C to +85°C	24 Wide SO
MAX7219ERG	-40°C to +85°C	24 Narrow CERDIP

* *Contact factory for dice specifications.*

Pin Configuration



Typical Application Circuit



Call toll free 1-800-998-8800 for free samples or literature.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serially Interfaced, 8-Digit LED Display Driver

ABSOLUTE MAXIMUM RATINGS

V+ Voltage 7V
DIG0-DIG7 Sink Current 500mA
SEG A-G, DP Source Current 100mA
Input Voltage (any pin) ..... V+ + 0.3V to -0.3V
Continuous Power Dissipation (T_A = +85°C)
Narrow Plastic DIP 0.87W
Wide SO 0.76W
CERDIP 1.1W

Operating Temperature Ranges:
MAX7219C_G ..... 0°C to +70°C
MAX7219E_G ..... -40°C to +85°C
Storage Temperature Range ..... -65°C to +160°C
Lead Temperature (soldering, 10 sec) ..... +300°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 in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V+ = 5V ±10%, R_{SET} = 9.53kΩ ±1%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

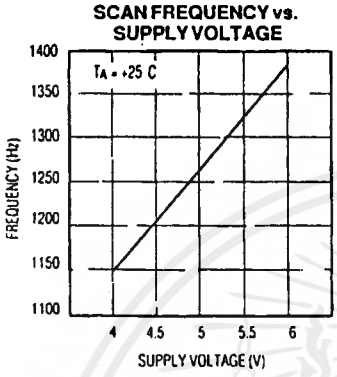
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Operating Supply Voltage	V+		4.0		6.0	V
Shutdown Supply Current	I_Q	DIN,CLK and LOAD = GND or V+, shutdown register set to 0, T_A = +25°C			150	µA
Operating Supply Current	I_{OP}	R_{SET} = infinity			8	mA
		All segments and decimal points on, I_{SEGO} = -40mA		330		mA
Display Scan Rate	f_{OSC}	V+ = 5V, 8 digits scanned	500	1300	2000	Hz
Digit Drive Sink Current	I_{DIGI}	T_A = +25°C, V+ = 5V, V_{OUT} = 0.65V	320			mA
Segment Drive Current Source	I_{SEGO}	T_A = +25°C, V+ = 5V, V_{OUT} = V+ - 1V	-30	-37	-40	mA
Segment Drive Current Matching				3.0		%
Digit Drive Source Current	I_{DIGO}	Digit off, V_{OUT} = V+ - 0.3V	-2			mA
Segment Drive Current Sink	I_{SEGI}	Segment off, V_{OUT} = 0.3V	5			mA
LOGIC INPUTS						
Input Current	I	DIN, CLK and LOAD V_{IN} = 0V V_{IN} = V+			-1 1	µA
Logic 1 Input Voltage	V_{IH}		3.5			V
Logic 0 Input Voltage	V_{IL}				0.8	V
Hysteresis Voltage		DIN, CLK, and LOAD		1.0		V
Output High Voltage	V_{OH}	DOUT I_{OUT} = -1mA, I_{OUT} -1µA	V+-1.0	V+		V
Output Low Voltage	V_{OL}	DOUT, I_{OUT} = 1.6mA			0.4	V
Data-Hold Time DATAIN to Clock	t_{IDH}		0	-5		ns
Data-Setup Time DATAIN To Clock	t_{IDS}		25			ns
Clock-to-Serial Output Prop Delay	t_{OPD}	C_{LOAD} = 50pF			25	ns
Clock Low Time	t_{CKL}		50			ns
Clock High Time	t_{CKH}		50			ns
Data-to-Segment Prop Delay (Note 1)	t_{DSPD}	C_{LOAD} = 50pF	0		2.25	ms
Load-Rising Edge to Next Clock Rising Edge	t_{LDCK}		50			ns
Clock-to-Load Rising Edge Setup Time	t_{CKLD}		0			ns
Load Low Time	t_{LDL}		50			ns
Load High Time	t_{LDH}		50			ns

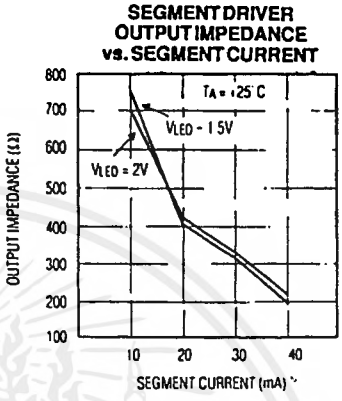
Note 1: Guaranteed by design.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serially Interfaced, 8-Digit LED Display Driver

Typical Operating Characteristics





Pin Description

PIN	NAME	FUNCTION
1	DIN	Serial Data Input. Data is loaded into an internal 16-bit shift register on the rising edge of CLK.
2, 3, 5-8, 10, 11	DIG0-7	8 digit drive lines that sink current from the display.
4, 9	GND	Ground (both GND pins must be connected)
12	LOAD	Load Data Input. On LOAD's rising edge, the last 16 bits of serial input data are latched
13	CLK	Clock Input. 10MHz maximum rate. On CLK's rising edge, data is shifted into the internal shift register. On CLK's falling edge, data is clocked out of DOUT
14-17, 20-23	SEG A-G, DP	7-segment drive and decimal point lines that source current to the display.
18	ISET	Connect to V+ through a resistor (R_{SET}) to set the peak segment current (Refer to "*Selecting R_{SET} Resistor*" section).
19	V+	Supply Voltage
24	DOUT	Serial Data Output. The data into DIN is valid at DOUT 16.5 clock cycles later.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serially Interfaced, 8-Digit LED Display Driver

Block Diagram

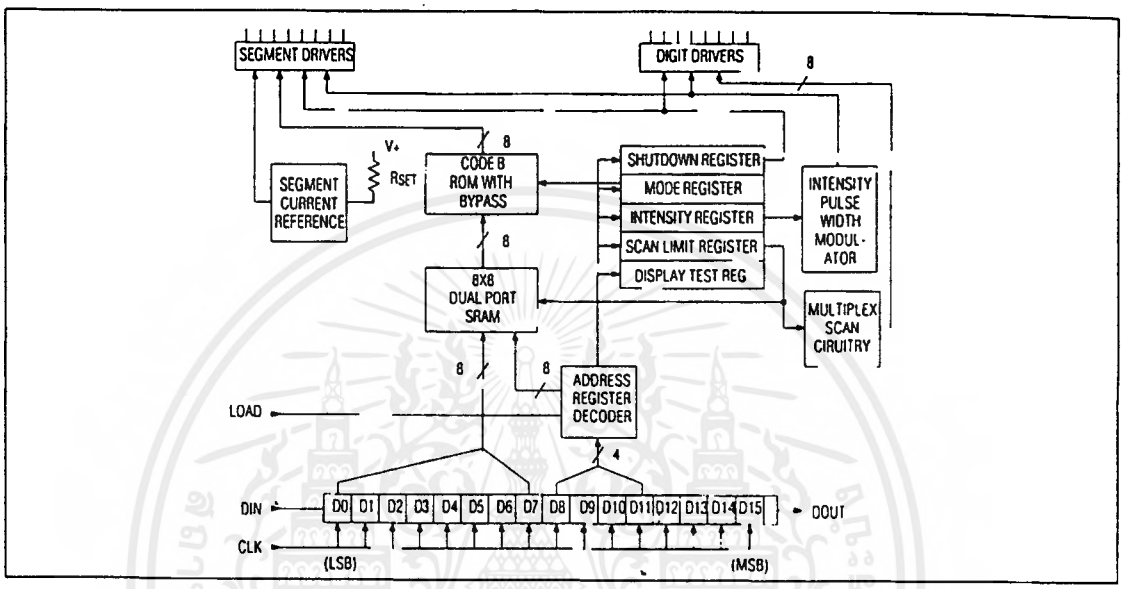


TABLE 1. SERIAL DATA FORMAT (16 BITS)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB			DATA				LSB

X = "don't care" bit

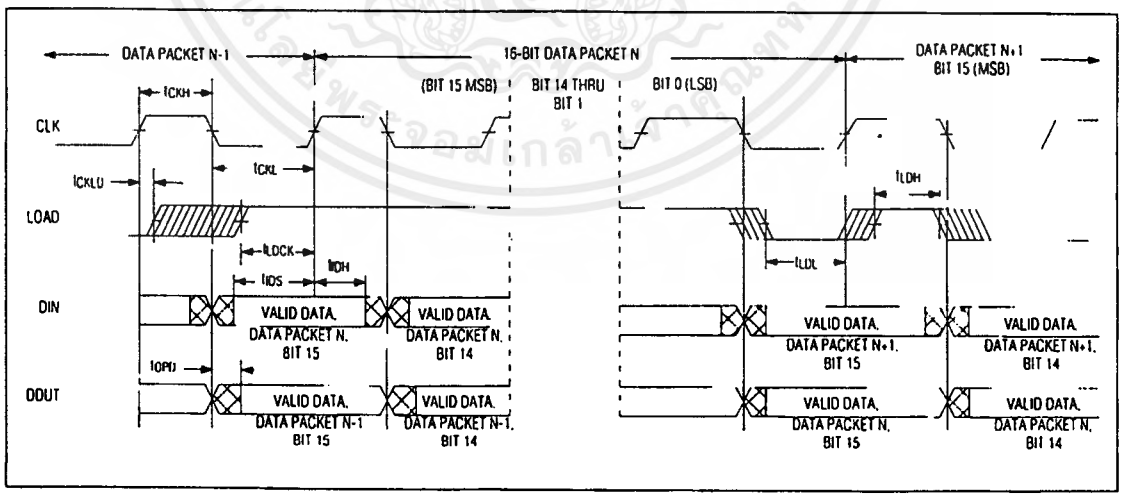


Figure 1. Timing Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Detailed Description

Serial Addressing Modes

Serial data at DIN, sent in 16-bit packets, is shifted into the internal 16-bit shift register with each rising edge of CLK. The data is then latched into either the digit or control registers on the rising edge of LOAD. LOAD must go high concurrently with or after the 16th rising clock edge, but before the next rising clock edge or data will be lost. Data at DIN is propagated through the shift register and appears at DOUT 16.5 clock cycles later. Data is clocked out on the falling edge of CLK. Data bits are labeled D0-D15 (Table 1). D8-D11 contain the register address, D0-D7 contain the data, and D12-D15 are "don't care" bits. The first bit received is D15, the most significant bit (MSB).

Digit and Control Registers

Table 2 lists the 14 addressable digit and control registers. The digit registers are realized with an on-chip, 8x8 dual-port SRAM. They are addressed directly so that individual digits can be updated and retain data as long as V+ typically exceeds 2V. The control registers consist of: decode mode, display intensity, scan limit (number of scanned digits), shutdown, and display test (all LEDs on). A no-operation (no-op) register is also included, which allows data to be passed from DIN to DOUT when devices are cascaded without changing the display or affecting any control registers.

Shutdown Mode

When the MAX7219 is in shutdown mode, the scan oscillator is halted, all segment current sources are pulled to ground, and all digit drivers are pulled to V+, thereby blanking the display. Data in the digit and control registers remains unaltered. Shutdown can be used to save power or as an alarm to flash the display by successively entering and leaving the shutdown mode. For minimum supply current in shutdown mode, logic inputs should be at ground or V+ (CMOS logic levels).

Typically, it takes less than 250µs for the MAX7219 to leave shutdown mode. Note that the display driver can still be programmed while in shutdown mode, and that shutdown mode can be overridden by the display-test function.

Table 2. Register Address Map

REGISTER	ADDRESS D15-D12	D11	D10	D9	D8	HEX CODE
NO-OP	X	0	0	0	0	X0
DIGIT 0	X	0	0	0	1	X1
DIGIT 1	X	0	0	1	0	X2
DIGIT 2	X	0	0	1	1	X3
DIGIT 3	X	0	1	0	0	X4
DIGIT 4	X	0	1	0	1	X5
DIGIT 5	X	0	1	1	0	X6
DIGIT 6	X	0	1	1	1	X7
DIGIT 7	X	1	0	0	0	X8
DECODE MODE	X	1	0	0	1	X9
INTENSITY	X	1	0	1	0	XA
SCAN LIMIT	X	1	0	1	1	XB
SHUTDOWN	X	1	1	0	0	XC
DISPLAY TEST	X	1	1	1	1	XF

Table 3. Shutdown Register Format (Address (Hex) = XC)

	ADDR CODE (HEX)	REGISTER DATA D7	D6	D5	D4	D3	D2	D1	D0
SHUTDOWN MODE	XC	X	X	X	X	X	X	X	0
NORMAL OPERATION	XC	X	X	X	X	X	X	X	1

Initial Power-Up

On initial power-up, all control registers are reset, the display is blanked, and the MAX7219 enters shutdown mode. Therefore the user must program the display driver prior to display use since it will initially be set to scan one digit, it will not decode data in the data registers, and the intensity register will be set to its minimum value.

Decode-Mode Register

The decode-mode register sets BCD code B (0-9, E, H, L, P, and -) or no-decode operation for each digit. Each bit in the register corresponds to one digit. A logic high selects code B decoding while a logic low bypasses the decoder. Examples of the decode mode control-register format are shown in Table 4.

Table 4. Decode-Mode Register Examples (Address (Hex) = X9)

	REGISTER DATA D7	D6	D5	D4	D3	D2	D1	D0	(HEX CODE)
NO DECODE FOR DIGITS 7-0	0	0	0	0	0	0	0	0	00
CODE B DECODE FOR DIGIT 0 NO DECODE FOR DIGITS 7-1	0	0	0	0	0	0	0	1	01
CODE B DECODE FOR DIGITS 3-0 NO DECODE FOR DIGITS 7-4	0	0	0	0	1	1	1	1	0F
CODE B DECODE FOR DIGITS 7-0	1	1	1	1	1	1	1	1	FF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serially Interfaced, 8-Digit LED Display Driver

Table 5. Code B Font

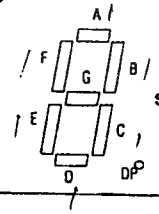
7-SEGMENT CHARACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7*	D6-D4	D3	D2	D1	D0	DP*	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	1	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1
9		X	1	0	0	1		1	1	1	1	0	1	1
-		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	1	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

* The decimal point is set by bit D7 = 1

When the code B decode mode is used, the decoder looks only at the lower nibble of the data in the digit registers (D3-D0), disregarding bits D4-D6. D7, which sets the decimal point (SEG DP), is independent of the decoder and is positive logic (D7=1 turns the decimal point on). The code-B font is listed in Table 5.

When no-decode is selected, data bits D7-D0 correspond to the segment lines of the MAX7219. Table 6 shows the one-to-one pairing of each data bit to the appropriate segment line.

Table 6. No-decode Mode Data Bits and Corresponding Segment Lines



STANDARD 7-SEGMENT LED

	REGISTER DATA							
	D7	D6	D5	D4	D3	D2	D1	D0
CORRESPONDING SEGMENT LINE	DP	A	B	C	D	E	F	G

Intensity Control and Interdigit Blanking

The MAX7219 allows the display brightness to be controlled with an external resistor (R_{SET}) connected between V+ and ISET, and digitally using the intensity register. The peak current sourced from the segment drivers will nominally be 100 times the current entering ISET. This resistor can either be fixed, or variable to allow brightness adjustment from the front panel. Its minimum value should be 9.53kΩ, which typically sets the segment current at 37mA.

Digital control of segment current is provided by an internal pulse-width modulated DAC, which is loaded from the lower nibble of the intensity register. The DAC scales the average segment current in 16 steps from a maximum of 31/32, down to 1/32 of the peak current set by R_{SET}. The intensity register format is listed in Table 7. Maximum brightness occurs with a duty cycle of 31/32 because the interdigit blanking time is set to 1/32 of a cycle. Interdigit blanking time can be increased by decreasing the duty cycle.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 7. Intensity Register Format (Address (Hex) = XA)

DUTY CYCLE	REGISTER DATA D7	D6	D5	D4	D3	D2	D1	D0	(HEX CODE)
1/32 (min on)	X	X	X	X	0	0	0	0	X0
3/32	X	X	X	X	0	0	0	1	X1
5/32	X	X	X	X	0	0	1	0	X2
7/32	X	X	X	X	0	0	1	1	X3
9/32	X	X	X	X	0	1	0	0	X4
11/32	X	X	X	X	0	1	0	1	X5
13/32	X	X	X	X	0	1	1	0	X6
15/32	X	X	X	X	0	1	1	1	X7
17/32	X	X	X	X	1	0	0	0	X8
19/32	X	X	X	X	1	0	0	1	X9
21/32	X	X	X	X	1	0	1	0	XA
23/32	X	X	X	X	1	0	1	1	XB
25/32	X	X	X	X	1	1	0	0	XC
27/32	X	X	X	X	1	1	0	1	XD
29/32	X	X	X	X	1	1	1	0	XE
31/32 (max on)	X	X	X	X	1	1	1	1	XF

Scan-Limit Register

The scan-limit register sets how many digits are displayed, from 1 to 8. They are displayed in a multiplexed manner with a typical display scan rate of 1300Hz with 8 digits displayed. If fewer digits are displayed, the scan rate is $8f_{OSC}/N$, where N is the number of digits scanned. Since the number of scanned digits affects the display brightness, the scan-limit register should not be used to blank portions of the display (such as leading zero suppression). The scan-limit register format is listed in Table 8.

If the scan-limit register is set for three digits or less, individual digit drivers will dissipate excessive amounts of power. Consequently, the value of the R_{SET} resistor must be adjusted according to the number of digits displayed, to limit individual digit driver power dissipation. Table 9 lists the number of digits displayed and the corresponding maximum recommended segment current when the internal digit drivers are used.

Display-Test Register

The display-test register operates in two modes: normal and display test. Display-test mode turns all LEDs on by overriding - but not altering - all controls and digit registers (including the shutdown register). In display-test mode, 8 digits are scanned and the duty cycle is 31/32. Table 9 lists the display-test register format.

Table 8. Scan-Limit Register Format (Address (Hex) = XB)

	REGISTER DATA D7	D6	D5	D4	D3	D2	D1	D0	(HEX CODE)
*DISPLAY DIGIT 0 ONLY	X	X	X	X	X	0	0	0	X0
*DISPLAY DIGITS 0 & 1	X	X	X	X	X	0	0	1	X1
*DISPLAY DIGITS 0 1 2	X	X	X	X	X	0	1	0	X2
DISPLAY DIGITS 0 1 2 3	X	X	X	X	X	0	1	1	X3
DISPLAY DIGITS 0 1 2 3 4	X	X	X	X	X	1	0	0	X4
DISPLAY DIGITS 0 1 2 3 4 5	X	X	X	X	X	1	0	1	X5
DISPLAY DIGITS 0 1 2 3 4 5 6	X	X	X	X	X	1	1	0	X6
DISPLAY DIGITS 0 1 2 3 4 5 6 7	X	X	X	X	X	1	1	1	X7

* See *"Scan-Limit Register"* text for application.

Table 9. Maximum Segment Current for 1, 2 or 3 Digit Displays

NUMBER OF DIGITS DISPLAYED	MAXIMUM SEGMENT CURRENT
1	10mA
2	20mA
3	30mA

Table 10. Display-Test Register Format (Address (Hex) = XF)

	REGISTER DATA D7	D6	D5	D4	D3	D2	D1	D0
NORMAL OPERATION	X	X	X	X	X	X	X	0
DISPLAY TEST MODE	X	X	X	X	X	X	X	1

Note: The MAX7219 remains in display-test mode (all LEDs on) until the display-test register is reconfigured for normal operation.

No-Op Register

The no-op register is used when cascading MAX7219s. Connect all devices' LOAD inputs together and connect DOUT to DIN on adjacent MAX7219s. DOUT is a CMOS logic level output that easily drives DIN of a successively cascaded MAX7219. Refer to the *"Serial Addressing Modes"* section for detailed information on serial input/output timing. For example, if four MAX7219s are cascaded, then to write to the fourth chip, send the desired 16-bit word, followed by three no-op codes (hex X0XX, see Table 2). When load goes high, data is latched in all devices. The first three chips receive no-op commands, and the fourth receives the intended data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serially Interfaced, 8-Digit LED Display Driver

Applications Information

Supply Bypassing and Wiring

To minimize power-supply ripple due to the peak digit driver currents, connect a 10µF electrolytic and a 0.1µF ceramic capacitor between V+ and GND as close to the device as possible. The MAX7219 should be placed in close proximity to the LED display, and connections should be kept as short as possible to minimize the effects of wiring inductance and electromagnetic interference. Also, both GND pins must be connected to ground.

Selecting R_{SET} Resistor and Using External Drivers

The current per segment is approximately 100 times the current in ISET. To select RSET, see Table 11. The MAX7219's maximum allowable segment current is 40mA. For an LED forward voltage drop of 2.5V, RSET must be greater than 9.53k. For segment current levels above the MAX7219 limits, external drivers will be needed. In this application, the MAX7219 serves as only a controller for other high-current drivers or transistors. Therefore, to conserve power in the MAX7219, use RSET = 47k when using external current sources as segment drivers.

The example in Figure 2 uses the MAX7219's segment drivers, a MAX333 single-pole double-throw analog switch, and external transistors to drive 4.0" AND4107SCL common-cathode displays. The 5.6V zener diode has been added in series with the decimal point LED because the decimal point LED forward voltage is typically 4.2V, while for all other segments the LED forward voltage is typically 8V. Note that since external transistors are used to sink current (DIG 0 and DIG 1 are used as logic switches), peak segment currents of 40mA are allowed even though only two digits are displayed. In applications where the MAX7219's digit drivers are used to sink current and fewer than four digits are displayed, see Table 9 which specifies the maximum allowable segment current. RSET must be selected accordingly (see Table 11).

Refer to the "Power Dissipation" section to calculate acceptable limits for ambient temperature, segment current, and the LED forward-voltage drop.

Table 11. RSET vs. Segment Current and LED Forward Voltage

	V_{LED} (V)				
I_{SEG} (mA)	1.5	2	2.5	3	3.5
40	11.3	10.4	9.8	8.9	7.8
30	16.3	15	14	12.9	11.4
20	26.2	24.6	22.8	20.9	18.6
10	60.1	56	51.7	47	41.9

Table 12. Package Thermal Resistance Data

PACKAGE	THERMAL RESISTANCE (θ_{JA})
24 Narrow DIP	+75°C/W
24 Wide SO	+85°C/W
24 CERDIP	+60°C/W
Maximum Junction Temperature (T_J) = +150°C	
Maximum Ambient Temperature (T_A) = +85°C	

Computing Power Dissipation

The upper limit for power dissipation (PD) for the MAX7219 is determined from the following equation:

$$PD = (V+ \times 8mA) + (V+ - V_{LED})(DUTY \times I_{SEG} \times N)$$

where:

V+ = Supply Voltage

DUTY = Duty Cycle set by intensity register

N = number of segments driven (worst case is 8)

V_{LED} = LED forward voltage

I_{SEG} = Segment Current set by RSET

Dissipation Example:

I_{SEG} = 40mA, N = 8, DUTY = 31/32, V_{LED} = 1.8V at 40mA, V+ = 5.25V

$$PD = 5.25V(8mA) + (5.25V - 1.8V)(31/32 \times 40mA \times 8)$$
$$= 1.11W$$

Thus, for a CERDIP package (θ_{JA} = +60C°/W from table 12), the maximum allowed ambient temperature T_A is given by:

$$T_Jmax = T_A + PD \times \theta_{JA}$$
$$+150°C = TA + 1.11W \times +60°C/W$$
$$T_A = +83.4°C$$

Cascading Drivers

The example in Figure 3 drives 16 digits using a 3-wire µP interface. If the number of digits is not a multiple of 8, set both drivers' scan-limit registers to the same number so one display will not appear brighter than the other. For example, if 12 digits are needed, use 6 digits per display

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

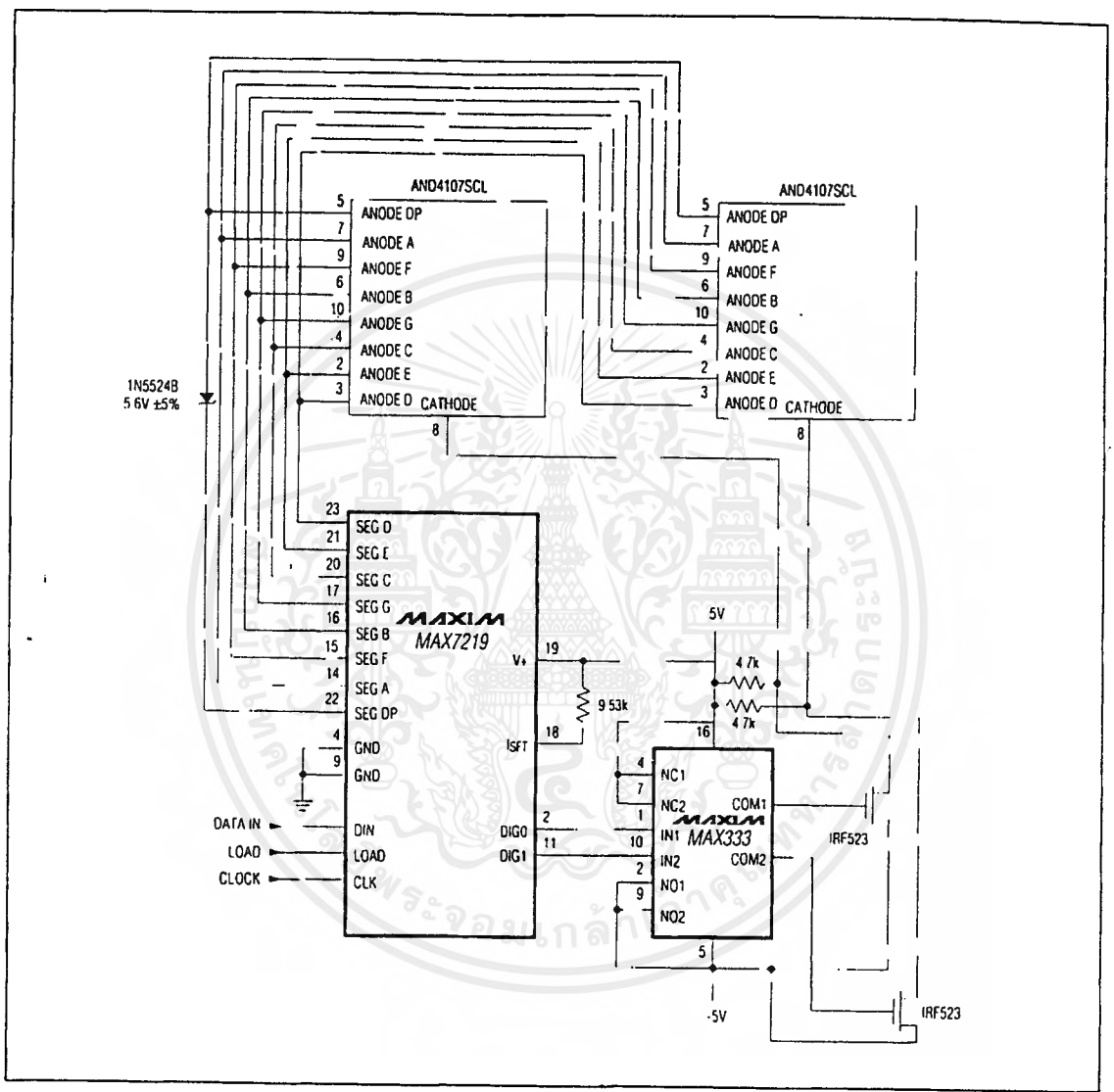


Figure 2. MAX7219 Driving 4 Inch Displays

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serially Interfaced, 8-Digit LED Display Driver

with both scan-limit registers set for 6 digits so that both displays have a 1/6 duty cycle per digit. If 11 digits are needed, set both scan-limit registers for 6 digits and leave one digit driver unconnected. If one display is set for 6 digits and the other for 5 digits, the second display will appear brighter because its duty cycle per digit will be 1/5 while the first display's will be 1/6. Refer to the "No Op Register" section for additional information.

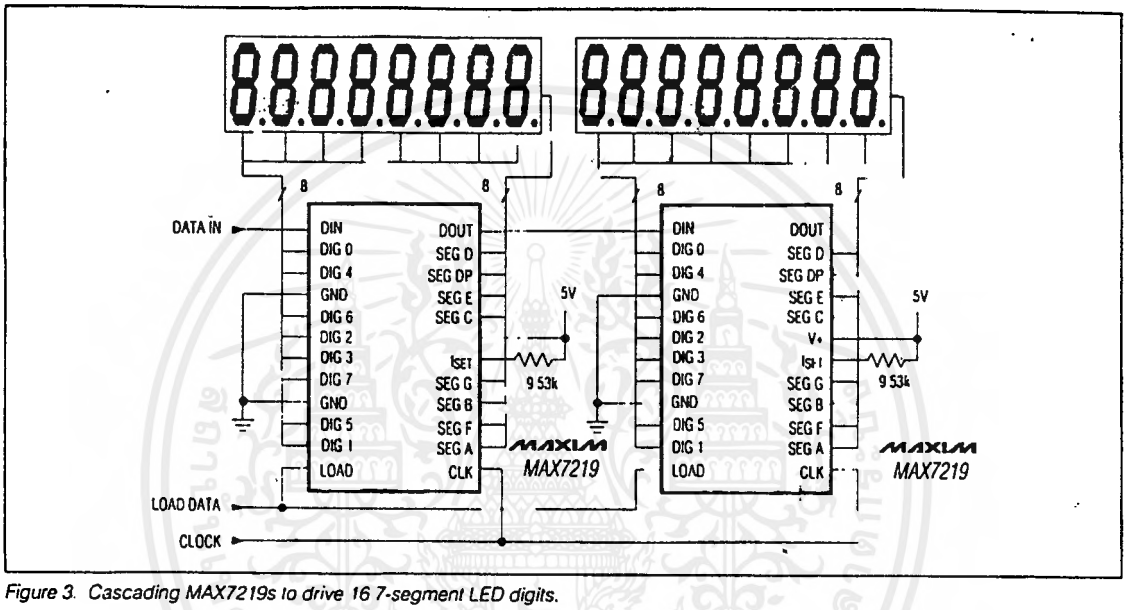
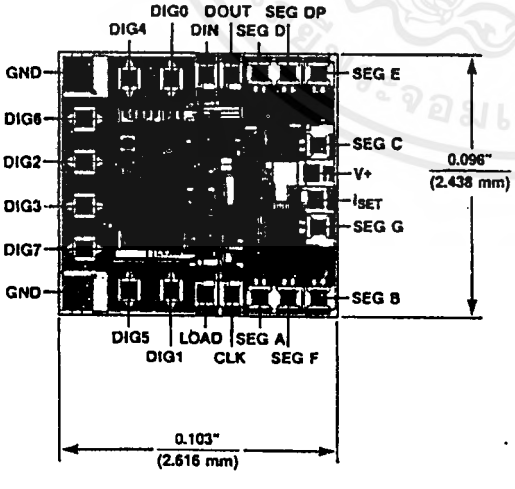


Figure 3. Cascading MAX7219s to drive 16 7-segment LED digits.

Chip Topography



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19-4323; Rev 3; 5/94

MAXIM

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

General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, and in particular, for those applications where ±12V is not available.

These parts are particulary useful in battery-powered systems since their low-power shutdown mode reduces power dissipation to less than 5µW. The MAX225, MAX233, MAX235, and MAX245-MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multi-Drop RS-232 Networks

Features

Superior to Bipolar

- ♦ Operate from Single +5V Power Supply (+5V and +12V—MAX231 and MAX239)
- ♦ Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- ♦ Meet All EIA/TIA-232E and V.28 Specifications
- ♦ Multiple Drivers and Receivers
- ♦ 3-State Driver and Receiver Outputs
- ♦ Open-Line Detection (MAX243)

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

**Contact factory for dice specifications.*

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (µF)	SHDN. & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	4.7/10	No		120	Ultra low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes		200	+5V IBM PC serial port with receivers active in shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 + receivers active in shutdown
MAX225	+5	5/5	0	–	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes		120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No		120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No		120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No		200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	–	No		120	No external caps
MAX233A	+5	2/2	0	–	No		200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No		120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	–	Yes		120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes		120	Shutdown, three-state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No		120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No		120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No		120	Standard +5/+12V or battery supplies; single package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes		120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes		120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No		200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No		120	High slew rate
MAX245	+5	8/10	0	–	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	–	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	–	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

Call toll free 1-800-998-8800 for free samples or literature.

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

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC}) ..... -0.3V to +6V
Input Voltages
T_{IN} ..... -0.3V to (V_{CC} + 0.3V)
R_{IN} ..... ±30V
T_{OUT} (Note 1) ..... ±15V
Output Voltages
T_{OUT} ..... ±15V
R_{OUT} ..... -0.3V to (V_{CC} + 0.3V)
Driver/Receiver Output Short Circuited to GND ..... Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) 842mW
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C) 889mW
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) 440mW
16-Pin Narrow SO (derate 8.70mW/°C above +70°C) 696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
18-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
20-Pin Wide SO (derate 10.00mW/°C above +70°C) 800mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C) 800mW
18-Pin CERDIP (derate 10.53mW/°C above +70°C) 842mW
Operating Temperature Ranges
MAX2__AC___, MAX2__C__ ..... 0°C to +70°C
MAX2__AE___, MAX2__E__ ..... -40°C to +85°C
MAX2__AM___, MAX2__M__ ..... -55°C to +125°C
Storage Temperature Range ..... -65°C to +160°C
Lead Temperature (soldering, 10 sec) ..... +300°

Note 1: Input voltage measured with T_{OUT} in high-impedance state, $\overline{SHDN}$ or V_{CC} = 0V.

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 in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1-C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High			2	1.4		V
Logic Pull-Up/Input Current	Normal operation			5	40	µA
	$\overline{SHDN}$ = 0V, MAX222/242, shutdown			±0.01	±1	
Output Leakage Current	V_{CC} = 5.5V, $\overline{SHDN}$ = 0V, V_{OUT} = ±15V, MAX222/242			±0.01	±10	µA
	V_{CC} = $\overline{SHDN}$ = 0V, V_{OUT} = ±15V			±0.01	±10	
Data Rate	Except MAX220, normal operation			200	116	kbits/sec
	MAX220			22	20	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V			±7	±22	mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	V_{CC} = 5V	Except MAX243 $R2_{IN}$	0.8	1.3		V
		MAX243 $R2_{IN}$ (Note 2)	-3			
RS-232 Input Threshold High	V_{CC} = 5V	Except MAX243 $R2_{IN}$		1.8	2.4	V
		MAX243 $R2_{IN}$ (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	Except MAX243, V_{CC} = 5V, no hyst. in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND			-2	-10	mA
	Shrinking V_{OUT} = V_{CC}			10	30	
TTL/CMOS Output Leakage Current	$\overline{SHDN}$ = V_{CC} or $\overline{EN}$ = V_{CC}, 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	µA

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1-C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
EN Input Threshold Low	MAX242			1.4	0.8	V
EN Input Threshold High	MAX242		2.0	1.4		V
POWER SUPPLY						
Operating Supply Voltage			4.5		5.5	V
V_{CC} Supply Current (SHDN = V_{CC}), Figures 5, 6, 9, 19	No load	MAX220		0.5	2	mA
		MAX222/232A/233A/242/243		4	10	
	3kΩ load both outputs	MAX220		12		
		MAX222/232A/233A/242/243		15		
Shutdown Supply Current	MAX222/242	T_A = +25°C		0.1	10	µA
		T_A = 0° to +70°C		2	50	
		T_A = -40° to +85°C		2	50	
		T_A = -55° to +125°C		35	100	
SHDN Input Leakage Current	MAX222/242				±1	µA
SHDN Threshold Low	MAX222/242			1.4	0.8	V
SHDN Threshold High	MAX222/242		2.0	1.4		V
AC CHARACTERISTICS						
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/µs
		MAX220	1.5	3	30	
Transmitter Propagation Delay TTL to RS-232 (Normal Operation), Figure 1	t_{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	µs
		MAX220		4	10	
	t_{PLHT}	MAX222/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Receiver Propagation Delay RS-232 to TTL (Normal Operation), Figure 2	t_{PHLR}	MAX222/232A/233A/242/243		0.5	1	µs
		MAX220		0.6	3	
	t_{PLHR}	MAX222/232A/233A/242/243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TTL (Shutdown), Figure 2	t_{PHLS}	MAX242		0.5	10	µs
	t_{PLHS}	MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	t_{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t_{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time (SHDN goes High), Figure 4	t_{ET}	MAX222/242, 0.1µF caps (includes charge-pump start-up)		250		µs
Transmitter-Output Disable Time (SHDN goes Low), Figure 4	t_{DT}	MAX222/242, 0.1µF caps		600		ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	t_{PHLT} - t_{PLHT}	MAX222/232A/233A/242/243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (Normal Operation)	t_{PHLR} - t_{PLHR}	MAX222/232A/233A/242/243		100		ns
		MAX220		225		

Note 2: MAX243 $R2_{OUT}$ is guaranteed to be low when $R2_{IN}$ ≥ 0V or is floating.

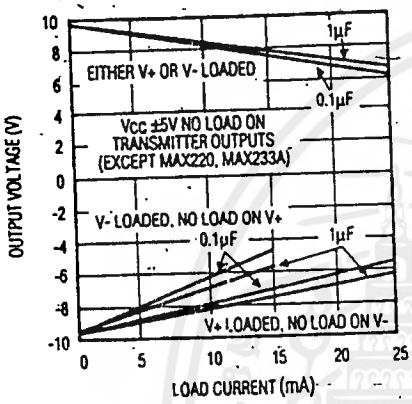
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกา

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

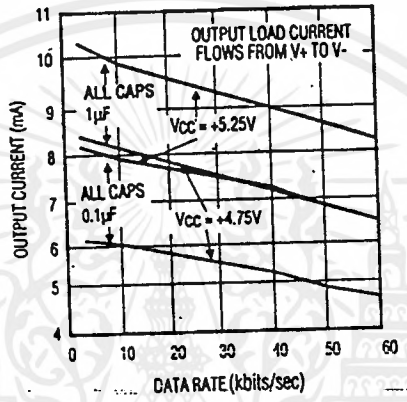
Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243

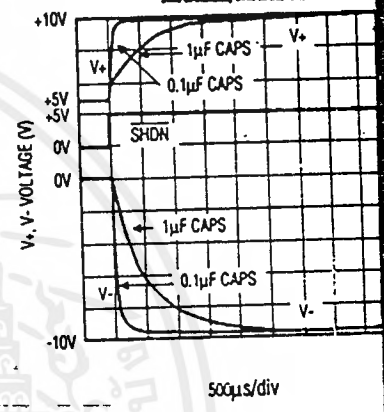
OUTPUT VOLTAGE vs. LOAD CURRENT



AVAILABLE OUTPUT CURRENT vs. DATA RATE

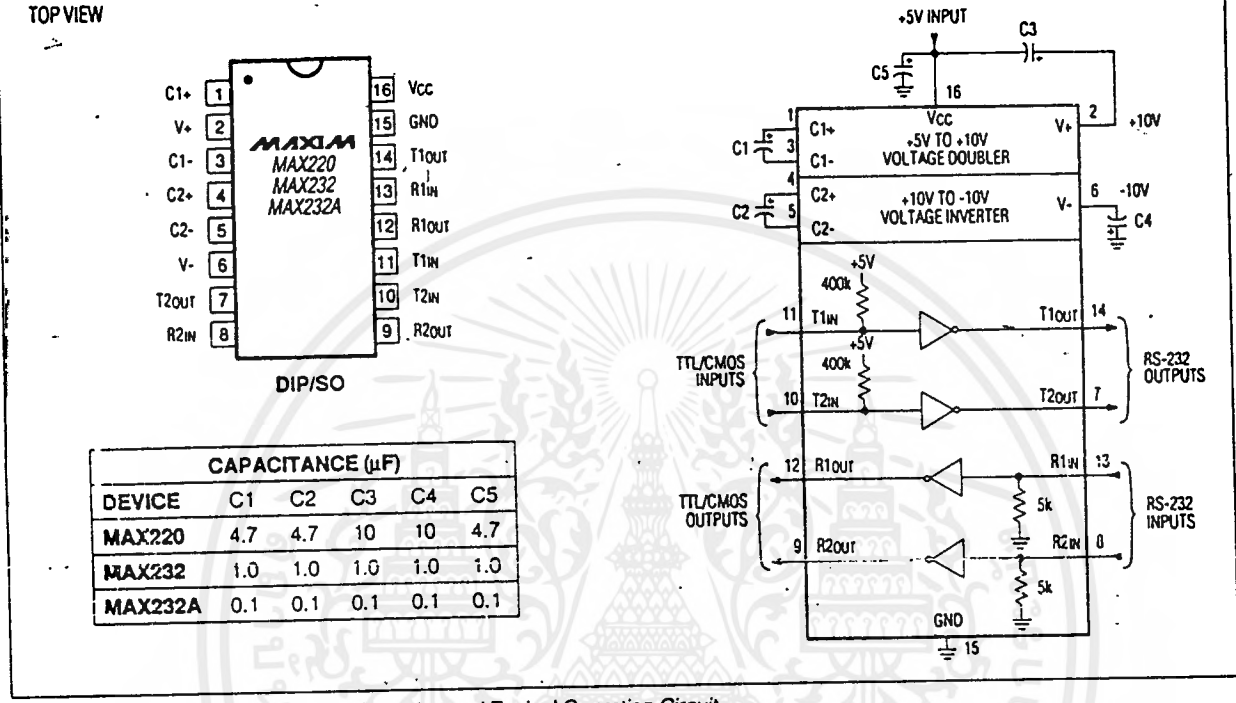


ON-TIME EXITING SHUTDOWN – MAX222/MAX242



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา

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



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 5. MAX220/232/232A Pin Configuration and Typical Operating Circuit

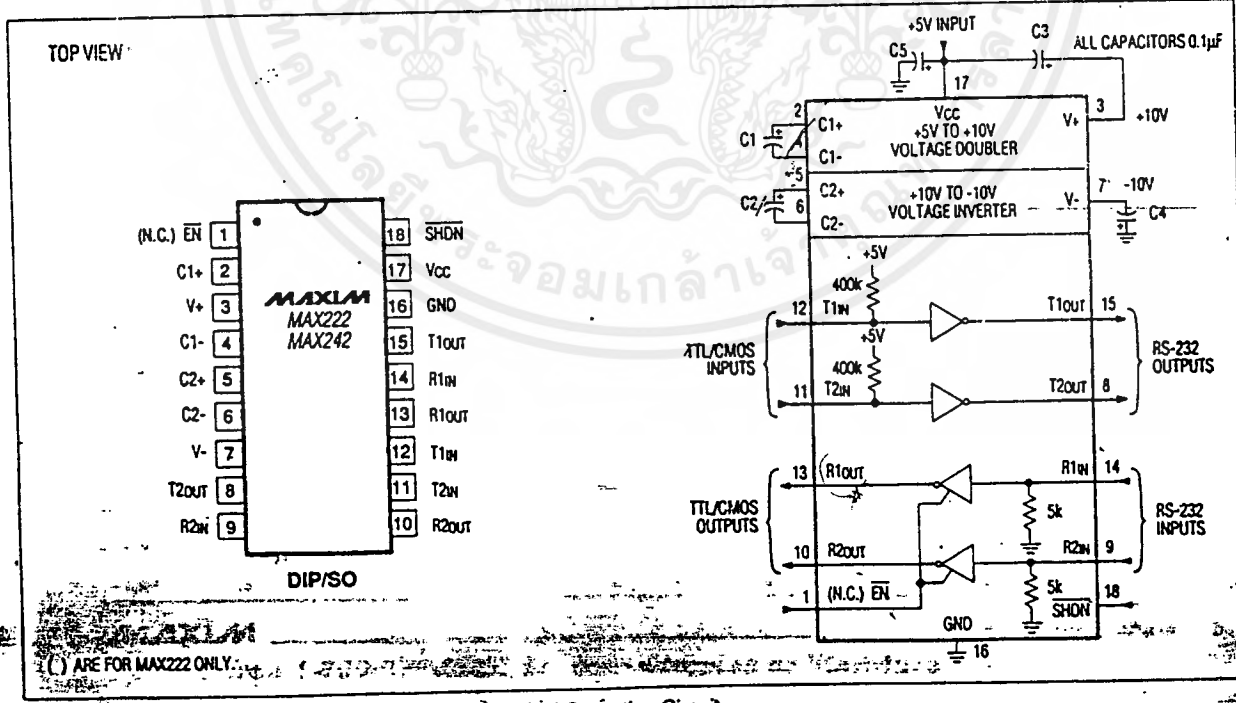


Figure 6. MAX222/MAX242 Pin Configuration and Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า