

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เช็คข้อมูลสินค้าด้วยบาร์โค้ดแบบไร้สาย

WIRELESS BARCODE FOR STOCK CHECKING



โดย

นายชัยมงคล วงษ์พานิช

นางสาวบุศรินทร์ ปัญญางาม

นางสาวอรอมล ถ้วยงาม

รพ.
๕๔๑๑๘
๒๕๔๙

เลขหมู่.....

เลขทะเบียน..... 72881

วัน,เดือน,ปี. 2.๖ ส.ย. 2550

b.	11๙๖๓๙๓๕
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ผ่านการตรวจรูปเล่มแล้ว

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

เช็คข้อมูลสินค้าด้วยบาร์โค้ดแบบไร้สาย

WIRELESS BARCODE FOR STOCK CHECKING



ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เช็คข้อมูลสินค้าด้วยบาร์โค้ดแบบไร้สาย

WIRELESS BARCODE FOR STOCK CHECKING

ผู้จัดทำ

1. นายชัยมงคล วงษ์พานิช 46010160
2. นางสาวบุตรีนทร์ ปัญญางาม 46010391
3. นางสาวอรอมล ถ้ายงาม 46010941



(ผศ.ดร. พิพัฒน์ พรหมมี)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช็กรหัสสินค้าด้วยบาร์โค้ดแบบไร้สาย

WIRELESS BARCODE FOR STOCK CHECKING

โดย นายชัยมงคล วงษ์พานิช 46010160
นางสาวบุศรินทร์ ปัญญาภาม 46010391
นางสาวอรอมล ถ้วยงาม 46010941

อาจารย์ที่ปรึกษา ศศ.ดร.พิพัฒน์ พรหมมี

บทคัดย่อ

โครงการนี้นำเสนอระบบตรวจสอบราคาและจำนวนสินค้าแบบไร้สาย ซึ่งประกอบด้วย เครื่องอ่านบาร์โค้ด, หน่วยรับคำสั่ง, หน่วยแสดงผล, วงจรรับ - ส่งสัญญาณซึ่งทำงานด้วยไมโครคอนโทรลเลอร์ และระบบฐานข้อมูลกลาง ในส่วนของการทำงาน ชุดอ่านบาร์โค้ดจะส่งรหัสสินค้าที่อ่านได้พร้อมกับคำสั่งงานระบบฐานข้อมูล ด้วยความถี่วิทยุไปยังฐานข้อมูลเพื่อทำการร้องขอข้อมูลของสินค้า เมื่อฐานข้อมูลทำการประมวลผลก็จะส่งข้อมูลกลับมาซึ่งหน่วยแสดงผลของหน่วยรับคำสั่งต่อไป

Abstract

This project describes a wireless barcode for stock checking system. First part consists of a barcode reader, LCD and a transceiver which are controlled by a microcontroller. Second part consists of a transceiver that connected with a database server computer. The radio frequency is used for query and response between a first and second parts. The LCD display can be shown for inventory, price and etc.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้ด้วยดี โดยได้รับความช่วยเหลือจากหลายท่าน คณะผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ.ดร. พิพัฒน์ พรหมมี ที่ให้คำปรึกษา คำแนะนำทั้งทางทฤษฎีและปฏิบัติ ความอนุเคราะห์เครื่องมือและอุปกรณ์ที่ใช้ในการทดสอบทดลอง ตลอดจนชี้แนะแนวทางในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้น ขอขอบคุณนายอัครวิระ ศิวินารถ, นายมรกต กรพิพัฒน์, นายศรัณย์วุฒิ ชื่นถนอมบุญ และนายมนัส วงษ์ลา ที่สละเวลามาให้คำปรึกษาในการทำปริญญาบัตรเป็นอย่างดีมาโดยตลอด คณะผู้จัดทำระลึกอยู่เสมอว่าหากไม่ได้รับความช่วยเหลือจากบุคคลที่กล่าวมานั้น รายงานฉบับนี้ก็ไม่สามารที่จะสำเร็จลุล่วงไปด้วยดี จึงขอขอบพระคุณมา ณ ที่นี้

ท้ายที่สุดนี้คณะผู้จัดทำขอขอบพระคุณบิดา มารดา ที่ให้การอบรมสั่งสอน เลี้ยงดู และให้โอกาสในการศึกษาอย่างเต็มที่ และขอบคุณเพื่อนๆ ที่ให้กำลังใจและช่วยเหลือกันอย่างเต็มที่เสมอมา

นายชัยมงคล วงษ์พานิช
นางสาวบุศรินทร์ ปัญญากรม
นางสาวอรอมล ถ้วยงาม

12 มีนาคม 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	
1.1 ความเป็นมาของปริญาานิพนธ์	1
1.2 วัตถุประสงค์ของปริญาานิพนธ์	1
บทที่ 2 ทฤษฎีและหลักการ	
2.1 บาร์โค้ด	2
2.1.1 ความหมาย	2
2.1.2 ส่วนประกอบของบาร์โค้ด	2
2.1.3 หลักการทำงานของบาร์โค้ด	3
2.1.4 โครงสร้างของบาร์โค้ด	3
2.1.5 เซนเซอร์อ่านบาร์โค้ด	4
2.1.6 ประเภทของเครื่องอ่านบาร์โค้ด	4
2.1.7 มาตรฐานบาร์โค้ด	5
2.2 ไมโครคอนโทรลเลอร์	9
2.2.1 คุณลักษณะพื้นฐานของ MCS-51	9
2.2.2 โครงสร้างหน่วยความจำ	9
2.2.2.1 หน่วยความจำโปรแกรมของ MCS-51	9
2.2.2.2 หน่วยความจำข้อมูลของ MCS-51	10
2.2.3 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register)	11
2.2.3.1 แอ็กคิวมูเลเตอร์ (Accumulator)	11
2.2.3.2 รีจิสเตอร์ B	11
2.2.3.3 โปรแกรมเคาน์เตอร์(Program Counter)	11
2.2.3.4 สแตคพอยน์เตอร์(Stack Pointer)	11
2.2.3.5 ตัวชี้ข้อมูลหรือค้ำพอยน์เตอร์(Data Pointer)	11
2.2.3.6 โปรแกรมสเตตัสเวิร์ด(PSW)	11
2.2.3.7 รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ท(Port Register)	11
2.2.3.8 รีจิสเตอร์ SBUF	11
2.2.3.9 รีจิสเตอร์ PCON	11
2.2.3.10 รีจิสเตอร์ IP,IE,TMOD,SCON	12
2.2.4 การโปรแกรมพอร์ทอนุกรม	12
2.2.4.1 รีจิสเตอร์ต่างๆที่เกี่ยวข้องในการใช้งานพอร์ทอนุกรม	12
2.2.4.2 การส่งข้อมูลแบบอนุกรมในโหมด 3	14
2.3 เครื่องรับส่งวิทยุ	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
2.3.1 การมอดูเลต (Modulation)	15
2.3.1.1 การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM)	16
2.3.1.2 การมอดูเลตความถี่ (Frequency Modulation หรือ FM)	16
2.3.1.3 การมอดูเลตเฟส	16
2.3.2 การมอดูเลตแอมพลิจูดแบบแถบข้างคู่จำกัดคลื่นพาห์	16
2.3.3 การมอดูเลตแอมพลิจูดสำหรับวิทยุกระจายเสียง	21
2.3.4 รูปคลื่นของสัญญาณเอเอ็ม	22
2.3.5 วงจรตรวจจับกรอบสัญญาณ	23
2.3.6 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (D/A)	24
2.3.6.1 การมอดูเลตเชิงเลขทางความถี่ (FSK)	25
2.3.6.2 การมอดูเลตเชิงเลขทางเฟส (PSK)	25
2.3.6.3 การมอดูเลตเชิงเลขทางแอมพลิจูด (ASK)	25
2.4 Delphi 7	27
2.4.1 ความสามารถของ Delphi 7	27
2.4.1.1 สร้างแอปพลิเคชันสำหรับ Windows	27
2.4.1.2 สร้างระบบงานด้านฐานข้อมูล	28
2.4.1.3 สร้างแอปพลิเคชันรองรับ .NET Web Service	28
2.5 พอร์ตอนุกรม	28
2.5.1 การสื่อสารข้อมูล	28
2.5.1.1 การสื่อสารข้อมูลแบบขนาน (Parallel Communication)	28
2.5.1.2 การสื่อสารข้อมูลแบบอนุกรม (Serial Communication)	29
2.6 แอลซีดี	34
บทที่ 3 การคำนวณและการสร้าง	36
3.1 บล็อกไดอะแกรม	36
3.2 ไมโครคอนโทรลเลอร์	37
3.3 การทำงานของ RF module	38
3.3.1 การทำงานเมื่อเป็นตัวส่ง	38
3.3.2 การทำงานเมื่อเป็นตัวรับ	39
3.4 ฐานข้อมูล	40
3.5 การออกแบบผังงานของไมโครคอนโทรลเลอร์ฝังบาร์โค้ด	42
3.6 การออกแบบผังงานของโปรแกรมอินเตอร์รัปต์ฝังบาร์โค้ด	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
3.7 การออกแบบพลังงานของไมโครคอนโทรลเลอร์ฝั่งเซิร์ฟเวอร์	44
3.8 การออกแบบพลังงานของโปรแกรมอินเทอร์รัปต์ฝั่งเซิร์ฟเวอร์	45
3.9 การออกแบบพลังงานของโปรแกรมแสดงผลทางแอลซีดี	46
3.10 การออกแบบพลังงานของโปรแกรมแสดงผลทางด้านเซิร์ฟเวอร์	47
บทที่ 4 การทดลองและผลการทดลอง	49
4.1 การทดลองวัดค่าสัญญาณที่ส่งจาก AT89C4051 ไปยัง TRW (ด้านบาร์โค้ด)	49
4.2 การทดลองวัดค่าสัญญาณที่ส่งจาก TRW ไปยัง AT89C4051 (ด้านเซิร์ฟเวอร์)	53
4.3 การทดลองเปรียบเทียบสัญญาณด้านบาร์โค้ดกับด้านเซิร์ฟเวอร์	53
4.4 การทดลองแสดงผลเป็นหมายเลขบาร์โค้ดบนไฮเปอร์เทอร์มินอล	60
4.5 การทดลองโปรแกรมเคลฟ	61
4.6 การทดลอง นำค่าที่ได้จากเรียกค้นข้อมูลในฐานข้อมูลกลับ	61
4.6.1 ทดลองรับข้อมูลที่ได้จากการเรียกค้นฐานข้อมูลมาแสดงบนไฮเปอร์เทอร์มินอลของคอมพิวเตอร์อีกเครื่องหนึ่ง	61
4.7 การทดลองแสดงผลที่เรียกค้นได้จากฐานข้อมูล มาแสดงบนหน้าจอแอลซีดี	62
4.8 รูปวงจรที่ใช้ในการทดลอง	64
บทที่ 5 บทวิจารณ์และสรุปผล	66
5.1 บทวิจารณ์	66
5.1.1 ส่วนของเครื่องอ่านบาร์โค้ด	66
5.1.2 ส่วนของไมโครคอนโทรลเลอร์และพอร์ตอนุกรม	66
5.1.3 การรับส่งผ่านคลื่นวิทยุ	66
5.1.4 ส่วนของฐานข้อมูล	66
5.2 บทสรุป	67
5.3 แนวทางที่จะทำการแก้ไขเพิ่มเติม	67

สารบัญรูป

	หน้า
รูปที่ 2.1 โครงสร้างของบาร์โค้ด	3
รูปที่ 2.2 เครื่องอ่านบาร์โค้ดประเภทต่าง ๆ	5
รูปที่ 2.3 ตัวอย่างมาตรฐานบาร์โค้ดระบบ UPC	6
รูปที่ 2.4 ตัวอย่างมาตรฐานบาร์โค้ดระบบ EAN	6
รูปที่ 2.5 ตัวอย่างมาตรฐานบาร์โค้ดระบบ CODE 39	7
รูปที่ 2.6 ตัวอย่างมาตรฐานบาร์โค้ดระบบ IFT 2 of 5	7
รูปที่ 2.7 ตัวอย่างมาตรฐานบาร์โค้ดระบบ CodaBar	7
รูปที่ 2.8 ตัวอย่างมาตรฐานบาร์โค้ดระบบ ISBN/ISSN	8
รูปที่ 2.9 การจัดพื้นที่หน่วยความจำของไมโครคอนโทรลเลอร์	10
รูปที่ 2.10 แสดงลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล	10
รูปที่ 2.11 รูปแบบการรับส่งข้อมูลในโหมด 3	14
รูปที่ 2.12 รูปแบบของการสื่อสารในการรับส่งสัญญาณ	15
รูปที่ 2.13 กระบวนการมอดูเลตสัญญาณดิจิตอล เอสซี	18
รูปที่ 2.14 การคิ่อมอดูเลตสัญญาณดิจิตอล เอสซี และสัญญาณต่าง ๆ พร้อมทั้ง ค่าความหนาแน่นสเปกตรัมของสัญญาณที่อินพุตของวงจรกรองความถี่ต่ำผ่าน	19
รูปที่ 2.15 รูปสัญญาณเอเอ็ม ที่เปลี่ยนแปลงตามสภาวะของสัญญาณข้อมูล	22
รูปที่ 2.16 ประกอบการอธิบายการทำงานของวงจรตรวจจับรอบสัญญาณ	23
รูปที่ 2.17 แผนผังระบบการสร้างสัญญาณเอเอ็มแบบง่าย	24
รูปที่ 2.18 การแปลงสัญญาณ D/A แบบ FSK	25
รูปที่ 2.19 การแปลงสัญญาณ D/A แบบ PSK	25
รูปที่ 2.20 รูปแบบของการมอดูเลตทางแอมพลิจูด	26
รูปที่ 2.21 ON-OFF ASK	26
รูปที่ 2.22 การแปลงสัญญาณ D/A แบบ ASK	27
รูปที่ 2.23 แสดงบล็อกไดอะแกรมรูปแบบการสื่อสารแบบขนาน	29
รูปที่ 2.24 แสดงบล็อกไดอะแกรมรูปแบบการสื่อสารแบบอนุกรม	30
รูปที่ 2.25 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม	30
รูปที่ 2.26 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส	31
รูปที่ 2.27 ลักษณะของคอนเน็คเตอร์แบบ DB-9	32
รูปที่ 2.28 อธิบายการทำงานของแอลซีดี	34
รูปที่ 3.1 บล็อกไดอะแกรมของปริณูณานิพนธ์	36
รูปที่ 3.2 แสดงวงจรภาคจ่ายไฟ (Power Supply)	37
รูปที่ 3.3 แสดงวงจรฝั่งบาร์โค้ด	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 3.4 แสดงวงจรฝั่งเซิร์ฟเวอร์	38
รูปที่ 3.5 แผนผังเวลาของการส่งแบบ ShockBurst	39
รูปที่ 3.6 แผนผังเวลาของการรับแบบ ShockBurst	40
รูปที่ 3.7 แสดงฐานข้อมูล	41
รูปที่ 3.8 ฟังก์ชันของโปรแกรมควบคุมไมโครคอนโทรลเลอร์ฝั่งบาร์โค้ด	42
รูปที่ 3.9 ฟังก์ชันของโปรแกรมอินเตอร์รัปต์ฝั่งบาร์โค้ด	43
รูปที่ 3.10 ฟังก์ชันของโปรแกรมควบคุมไมโครคอนโทรลเลอร์ฝั่งบาร์โค้ด	44
รูปที่ 3.11 ฟังก์ชันของโปรแกรมอินเตอร์รัปต์ฝั่งเซิร์ฟเวอร์	45
รูปที่ 3.12 ฟังก์ชันของโปรแกรมแสดงผลทางแอลซีดี	46
รูปที่ 3.13 ฟังก์ชันของโปรแกรมแสดงผลด้านเซิร์ฟเวอร์ขณะรับข้อมูล	47
รูปที่ 3.14 ฟังก์ชันของโปรแกรมแสดงผลทางด้านเซิร์ฟเวอร์ขณะส่งข้อมูลกลับ	48
รูปที่ 4.1 แสดงสัญญาณที่ไม่โครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 1-2 รหัส 8-8)	49
รูปที่ 4.2 แสดงสัญญาณที่ไม่โครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 3-4 รหัส 5-1)	50
รูปที่ 4.3 แสดงสัญญาณที่ไม่โครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 5-6 รหัส 9-5)	50
รูปที่ 4.4 แสดงสัญญาณที่ไม่โครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 7-8 รหัส 2-3)	51
รูปที่ 4.5 แสดงสัญญาณที่ไม่โครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 9-10 รหัส 5-0)	51
รูปที่ 4.6 แสดงสัญญาณที่ไม่โครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 11-12 รหัส 1-6)	52
รูปที่ 4.7 แสดงสัญญาณที่ไม่โครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 13 รหัส 1)	52
รูปที่ 4.8 แสดงตัวอย่างสัญญาณที่ส่งจาก TRW ไปยัง AT89C4051	53
รูปที่ 4.9 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 1-2-3 รหัส 8-8-5)	54
รูปที่ 4.10 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 4 -5-6 รหัส 1-9-5)	54
รูปที่ 4.11 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 7-8-9 รหัส 2-3-5)	55
รูปที่ 4.12 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 10-11-12 รหัส 0-1-6)	55
รูปที่ 4.13 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 12-13 รหัส 6-และเซต โหมด)	56
รูปที่ 4.14 แสดงสัญญาณที่วัดได้ (เซต โหมด และไบต์ที่ 1 รหัส 8)	56
รูปที่ 4.15 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 2-3-4 รหัส 8-5-1)	57
รูปที่ 4.16 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 5-6-7 รหัส 9-5-2)	57
รูปที่ 4.17 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 8-9-10 รหัส 3-5-0)	58
รูปที่ 4.18 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 11-12-13 รหัส 1-6-1)	58
รูปที่ 4.19 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 1-2-3 ของราคาสินค้า 8-7-.)	59
รูปที่ 4.20 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 1-2-3 ของราคาสินค้า 0-0-*)	59
รูปที่ 4.21 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 1-2-3 ของราคาสินค้า 3-6-*)	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 4.22 แสดงหน้าจอโปรแกรมไฮเปอร์เทอร์มินอลเมื่อได้รับหมายเลขบาร์โค้ด	60
รูปที่ 4.23 แสดงการเรียกค้นข้อมูลจากฐานข้อมูล ออกมาแสดงบนหน้าจอ	61
รูปที่ 4.24 แสดงการเรียกค้นข้อมูลจากฐานข้อมูล ออกมาแสดงบนหน้าจอ	62
รูปที่ 4.25 แสดงผลที่เรียกค้นได้จากฐานข้อมูล มาแสดงบนหน้าจอแอลซีดี โดยแสดงเฉพาะราคา	62
รูปที่ 4.26 แสดงผลที่เรียกค้นได้จากฐานข้อมูล มาแสดงบนหน้าจอแอลซีดี โดยแสดงเฉพาะจำนวน	63
รูปที่ 4.27 แสดงผลที่เรียกค้นได้จากฐานข้อมูล มาแสดงบนหน้าจอแอลซีดี โดยแสดงทั้งราคาและจำนวน	63
รูปที่ 4.28 วงจรที่ใช้รับ-ส่งข้อมูลฝั่งเซิร์ฟเวอร์	64
รูปที่ 4.29 วงจรที่ใช้รับ-ส่งข้อมูลฝั่งบาร์โค้ด	64
รูปที่ 4.30 เครื่องสแกนบาร์โค้ด	65



สารบัญตาราง

	หน้า
ตารางที่ 2.1 ตารางเปรียบเทียบเซนเซอร์อ่านบาร์โค้ด แบบ Laser และแบบ CCD	4
ตารางที่ 2.2 แสดงบิตภายใน PCON	12
ตารางที่ 2.3 แสดงบิตภายใน SCON	13
ตารางที่ 2.4 แสดงการเลือกโหมด	13
ตารางที่ 2.5 การเชื่อมต่อของพอร์ตสื่อสารสำหรับคอนเน็คเตอร์แบบ DB-9	33
ตารางที่ 3.1 ชนิดข้อมูลทั้งหมดในฐานข้อมูลโดยมีโค้ด (CODE) เป็นคีย์ของตาราง	39



บทที่ 1 บทนำ

1.1 ความเป็นมาของปริญญานิพนธ์

เทคโนโลยีบาร์โค้ดถูกนำมาใช้ทดแทนในส่วนการบันทึกข้อมูล (Data Entry) จากเดิมที่มนุษย์ใช้คีย์บอร์ดในการบันทึกข้อมูล การบันทึกด้วยคีย์บอร์ดมีอัตราความผิดพลาดอยู่ประมาณ 1 ใน 10^2 หรือ บันทึกข้อมูลผิดพลาด 1 ตัวอักษรในทุกๆ 100 ตัวอักษร และเมื่อเปลี่ยนมาใช้ระบบบาร์โค้ดแทนในขั้นตอนการบันทึกข้อมูล อัตราการเกิดความผิดพลาดจะลดลงเหลือเพียง 1 ใน 10^7 หรือ 10,000,000 ตัวอักษรเท่านั้น โดยทั่วไปอุปกรณ์ที่เชื่อมต่อคอมพิวเตอร์มีการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์โดยใช้สายนั้น สามารถแบ่งลักษณะการส่งข้อมูลได้เป็น 2 วิธีคือ

1. การส่งข้อมูลแบบขนาน ข้อมูลจะส่งออกไปทั้งไบต์ในครั้งเดียว โดยที่ทุกบิตจะถูกส่งออกไปพร้อมกัน ซึ่งต้องใช้บิตในการส่งข้อมูลอย่างน้อยเท่ากับจำนวนบิตใน 1 ไบต์
2. การส่งข้อมูลแบบอนุกรม ข้อมูลจะถูกส่งออกไปทีละบิต ดังนั้นจึงใช้บิตในการส่งข้อมูลเพียงเส้นเดียวเท่านั้น

ด้วยเหตุที่ว่า การส่งผ่านข้อมูลระหว่างคอมพิวเตอร์กับอุปกรณ์เชื่อมต่อคอมพิวเตอร์โดยใช้สายเชื่อมต่อนั้น ไม่สามารถที่จะทำได้ในระยะทางที่ไกล เนื่องจากสภาพความเป็นตัวเก็บประจุที่เกิดขึ้นในสาย และประกอบกับความไม่สะดวกในการใช้งาน บางครั้งก็ไม่อาจเดินสายสัญญาณไปในบริเวณที่ต้องการใช้งานอุปกรณ์ดังกล่าวได้ จึงได้มีการพัฒนาให้อุปกรณ์นั้นสามารถที่จะเชื่อมต่อเข้ากับคอมพิวเตอร์ได้โดยไม่ต้องใช้สายสัญญาณซึ่งคือระบบไร้สาย (Wireless system) มีหลายรูปแบบ เช่น อินฟราเรด (Infrared), คลื่นวิทยุ (Radio Frequency), และบลูทูธ (Bluetooth)

โดยการใช้อินฟราเรดที่มีการพัฒนาอย่างต่อเนื่อง และได้มีการนำมาผลิตขายแล้ว แต่ยังคงมีราคาสูงอยู่จึงยังไม่เป็นที่แพร่หลายมากเท่าไรนัก และการส่งสัญญาณแบบใช้อินฟราเรดนี้ยังมีข้อจำกัดอยู่หลายอย่าง เช่น สามารถรับ-ส่ง สัญญาณได้ในระยะทางสั้นๆ เท่านั้น อีกทั้งยังต้องจัดให้อุปกรณ์หันส่วนที่รับ-ส่งอินฟราเรดอยู่ในแนวเดียวกันด้วยจึงจะสามารถรับ-ส่งข้อมูลได้

แต่ในปริญญานิพนธ์นี้เราจะใช้หลักการส่งที่อาศัยคลื่นวิทยุ เนื่องจากประโยชน์คือทำให้สามารถที่จะนำอุปกรณ์เชื่อมต่อคอมพิวเตอร์ไปวางไว้ตรงไหนก็ได้ภายในระยะรัศมีที่กำหนด อีกทั้งไม่จำเป็นต้องวางเครื่องรับและเครื่องส่งให้ตรงกันเหมือนกับระบบที่ใช้อินฟราเรด จึงทำให้เกิดความสะดวกสบายเป็นอย่างมากกับผู้ใช้งานและไม่ต้องเสียค่าใช้จ่ายในการเดินสายและไม่ยุ่งยาก

1.2 วัตถุประสงค์ของปริญญานิพนธ์

- เพื่อศึกษาเกี่ยวกับการรับ-ส่ง สัญญาณในรูปผ่านคลื่นวิทยุ
- เพื่อศึกษาการรับ-ส่งข้อมูลระหว่างคอมพิวเตอร์กับเครื่องอ่านบาร์โค้ด
- เพื่อออกแบบวงจรเชื่อมต่อ ระหว่างเครื่องคอมพิวเตอร์กับเครื่องอ่านบาร์โค้ด
- สร้างวงจรเพื่อเชื่อมต่อคอมพิวเตอร์กับเครื่องอ่านบาร์โค้ดผ่านคลื่นวิทยุได้
- เพื่อนำผลงานที่ได้ไปใช้ประโยชน์ กรณีที่ไม่สะดวกในการเดินสาย ในระยะที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีและหลักการ

สำหรับปริญญาโทฉบับนี้มีทฤษฎีและหลักการแบ่งออกเป็น 5 ส่วน ได้แก่ บาร์โค้ด, ส่วนไมโครคอนโทรลเลอร์, ส่วนเครื่องรับส่งผ่านคลื่นวิทยุ, ส่วนของฐานข้อมูลและส่วนของ การสื่อสารผ่านพอร์ตอนุกรม

2.1 บาร์โค้ด

2.1.1 ความหมาย

คำว่าบาร์โค้ดในภาษาไทยนั้น ราชบัณฑิตยสถาน (2546) กำหนดให้ใช้คำว่า รหัสแท่ง แต่คนทั่วไปเรียก บาร์โค้ด ทับศัพท์จากคำภาษาอังกฤษโดยตรง คำว่า รหัสแท่ง หรือบาร์โค้ดนั้นหมายถึง สัญลักษณ์ที่มีรูปแบบเป็นแท่ง (Bar) ที่มีความหนาและบางแตกต่างกัน เพื่อใช้แทนข้อมูล (Code) ตัวเลขฐานสอง สามารถแยกแยะหรือระบุให้สิ่งของแต่ละชั้นมีความแตกต่างกันได้

ความหมายของบาร์โค้ดโดยลักษณะทางกายภาพว่า เป็นแถบรหัสรูปลายทางสีดำ (Bar) และแถบขาว (Space) เรียงขนานกันคล้ายทางม้าลาย แต่ขนาดความหนาและแถบระยะห่างมีลักษณะเป็นสัญลักษณ์เฉพาะ (Symbology) ที่กำหนดขึ้นตามเลขที่กำกับอยู่ ขนาดของแถบบาร์โค้ดมีความยาวแตกต่างกันขึ้นกับการใช้งาน และรูปแบบบาร์โค้ดที่พบเป็นกันมากในสินค้าผลิตภัณฑ์ต่างๆ อาจอยู่ในรูปแถบกระดาษ ภาพพิมพ์กระดาษห่อผลิตภัณฑ์ หรือเป็นส่วนหนึ่งของผลิตภัณฑ์เลขก็ได้

บาร์โค้ดที่ปรากฏบนสินค้าต่างๆ นั้นไม่ได้แสดงข้อมูลการขาย แต่เป็นตัวเลขอ้างอิง (Reference number) ที่กำหนดขึ้นเพื่อแยกชนิดของสินค้านั้นๆ ส่วนรายละเอียดต่างๆ เช่น บริษัทผู้ผลิต ประเภทของสินค้า ปริมาณ เลขที่ของผลิตภัณฑ์ และอื่นๆ จะเก็บไว้ในบาร์โค้ด ซึ่งจะถูกนำข้อมูลออกมาเมื่อแถบบาร์โค้ดถูกอ่านโดยเครื่องอ่านบาร์โค้ด

2.1.2 ส่วนประกอบของบาร์โค้ด

หมายถึง สัญลักษณ์ของบาร์โค้ดที่ใช้กันมีการกำหนดขึ้นมาหลายรูปแบบ ตามมาตรฐานของแต่ละองค์กร และตามจุดประสงค์ของการใช้งานแต่โดยทั่วไปแล้วบาร์โค้ดจะมีส่วนประกอบต่างๆ ดังต่อไปนี้

- Quiet Zone เป็นบริเวณที่ว่างเปล่าไม่มีการพิมพ์ข้อความใดๆ โดยจะอยู่ก่อนและหลังบาร์โค้ด
- Start/ Stop Character เป็นบริเวณแถบแท่งหรือช่องว่าง เพื่อเตรียมสั่งให้เซนเซอร์เริ่มต้นหรือหยุดบาร์โค้ด
- Data เป็นบริเวณแถบแท่งหรือช่องว่างที่แทนข้อมูลต่างๆ ที่เราต้องการ
- Check Digit เป็นบริเวณแถบแท่งที่ไว้สำหรับเก็บค่าตัวเลข เพื่อตรวจสอบในข้อมูลส่วน Data เพื่อให้มั่นใจว่าถูกต้องแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 หลักการทำงานของบาร์โค้ด

รหัสบาร์โค้ดประกอบด้วย 3 ส่วนดังนี้

1. ส่วนลายเส้น ซึ่งเป็นลายเส้นสีขาว (โปร่งแสง) และสีดำ มีขนาดความกว้างของลายเส้นตามมาตรฐานแต่ละ ชนิดของบาร์โค้ด

2. ส่วนข้อมูลตัวอักษร เป็นส่วนที่แสดงความหมายของชุดข้อมูลลายเส้นสำหรับให้อ่านเข้าใจ

3. ส่วนแถบว่าง เป็นส่วนที่เครื่องอ่านบาร์โค้ดใช้กำหนดขอบเขตของบาร์โค้ดและกำหนดค่าให้กับสีขาว (ความเข้มของการสะท้อนแสงในสีของพื้นผิวแต่ละชนิดที่ใช้แทนสีขาว) โดยทุกเส้นจะมีความยาวเท่ากันเรียงตามลำดับในแนวนอนจากซ้ายไปขวา

แถบสีทั้งสีขาวและสีดำที่มีความกว้างจะแทนค่าเป็น 1 และแถบสีที่มีความแคบ (หรือมองด้วยตาเหมือนเป็นเส้นตรงเล็กๆ) ทั้งขาวและดำจะมีค่าเป็น 0 แถบขาวและดำที่มีลักษณะและชื่อที่ใช้คือ

- แถบสีดำที่มีความกว้างมากกว่าเรียกว่า Wide Bar ถ้ามีความกว้างน้อยกว่าเรียกว่า Narrow Bar
- ช่องว่างหรือแถบสีขาวที่มีความกว้างมากกว่าเรียกว่า Wide Space ถ้ามีความกว้างน้อยกว่าเรียกว่า Narrow Space

2.1.4 โครงสร้างของบาร์โค้ด

Structure of a Width - Based Bar Code



รูปที่ 2.1 โครงสร้างของบาร์โค้ด

บาร์โค้ดประกอบด้วยแถบสีดำและสีขาว โดยความกว้างของแถบสีดำสลับขาวเป็นรหัสแทนข้อมูล เรียงจากซ้ายไปขวา การถอดรหัสจำเป็นต้องทราบความกว้างของแถบดำและแถบขาวนำไปเทียบกับตารางมาตรฐานเครื่องอ่านบาร์โค้ด ประกอบด้วยหัวอ่านอินฟราเรด แบบปากกาและแบบวงจรถอดรหัส การใช้งานเริ่มต้นด้วยการกวาดหัวอ่านผ่านบาร์โค้ด ซึ่งหัวอ่านจะมีตัวตรวจจับแสงสะท้อน ไปจุดชนวนวงจรถอดรหัส ทำให้เกิดคลื่นสัญญาณไฟฟ้าแบบพัลส์โดยความกว้างของรูปคลื่นจะเป็นสัดส่วนกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความกว้างของแถบโค้ดต่อนั้นวงจรถอดรหัสจะตรวจสอบความกว้างของรูปคลื่นแล้ว นำไปเปรียบเทียบกับแถบขาวดำทั้งหมด ที่แทนข้อมูลตัวเลขหรือตัวอักษร โดยปกติเครื่องอ่านจะต่อเข้ากับคอมพิวเตอร์ ดังนั้นวงจรภายในเครื่องอ่านจะส่งข้อมูลตัวเลขที่ถอดรหัสได้ไปยังคอมพิวเตอร์เพื่อประมวลผลต่อไป

2.1.5 เซนเซอร์อ่านบาร์โค้ด

เซนเซอร์อ่านบาร์โค้ดส่วนใหญ่แล้วแบ่งออกเป็น 2 ชนิดคือ

1. แบบ Laser จะใช้อ่านบาร์โค้ดที่คิดในสายการผลิต ซุปเปอร์มาร์เก็ต และคลังสินค้า หลักการทำงานคือ ลำแสงเลเซอร์ถูกปล่อยออกจากเลเซอร์ไดโอดมากระทบกับกระจกแบบหลายเหลี่ยมเพื่อที่จะสแกนบาร์โค้ด เมื่อลำแสงเลเซอร์กระทบบาร์โค้ดจะกระจายออก และถูกส่งมาที่โฟโต้ไดโอด ลักษณะของลำแสงที่กระจายตามบาร์โค้ดจะถูกแปลงไปเป็นสัญญาณอนาล็อก จากนั้นทำการแปลงสัญญาณเป็นดิจิทัล ลักษณะของสัญญาณดิจิทัลจะขึ้นอยู่กับขนาดของแท่ง และที่ว่างในแถบบาร์โค้ด จากนั้นก็จะแปลงรหัสเป็นข้อมูลผ่านพอร์ตคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์ไปประมวลผลหรือเก็บข้อมูลไว้ใช้

2. แบบ CCD จะใช้อ่านบาร์โค้ดที่คิดขึ้นงานที่มีขนาดเล็ก เช่น หลอดทดลอง แผงวงจร ที่ขึ้นงานกับตัวอ่านใกล้เคียงกัน หลักการทำงานคือ หลอด LED จะเปล่งแสงมากระทบบาร์โค้ดแล้วสะท้อนมาที่เซนเซอร์ CCD Image เพื่อจับภาพของบาร์โค้ดขึ้นมาเป็นข้อมูลเก็บไว้ใช้งานต่อไป การสแกนของเซนเซอร์อ่านบาร์โค้ดจะมี 2 แบบคือ แบบ Singer Scan จะปล่อยลำแสงขวางในการสแกน 1 แถว ซึ่งเหมาะแก่การเคลื่อนที่ของบาร์โค้ดแบบ Picket Fence Direction และแบบ Raster Scan จะปล่อยลำแสงขวางในการสแกนหลายแถว แม้บาร์โค้ดที่พิมพ์จะคุณภาพไม่ดีก็สามารถอ่านค่าได้ถูกต้อง การสแกนแบบนี้เหมาะสำหรับการเคลื่อนที่ของบาร์โค้ดแบบ Ladder Direction

ตารางที่ 2.1 ตารางเปรียบเทียบเซนเซอร์อ่านบาร์โค้ด แบบ Laser และแบบ CCD

เปรียบเทียบ	แบบ Laser	แบบ CCD
ข้อดี	อ่านได้ในระยะไกล มุมในการอ่านกว้าง สามารถอ่านวัตถุเคลื่อนที่ได้	ขนาดเล็ก ราคาไม่แพง อายุการใช้งานยาวนานกว่า
ข้อเสีย	ราคาแพง	ไม่เหมาะกับชิ้นงานเคลื่อนที่

2.1.6 ประเภทของเครื่องอ่านบาร์โค้ด

เครื่องอ่านบาร์โค้ดสามารถแบ่งตามลักษณะการใช้งานได้ดังนี้

1. Moving Bean Scanner เครื่องอ่านอยู่กับที่ แต่แสงฉายกวาดไปที่สินค้า เพื่อหาบาร์โค้ดที่กำกับบนสินค้านั้น

2. Fixed Bean Scanner เครื่องอ่านอยู่กับที่ลำแสงไม่เคลื่อนที่สินค้าเคลื่อนที่ผ่านจุดที่แสงฉาย

3. Hand Held Scanner เครื่องอ่านที่ต้องใช้คนควบคุมและถือได้ เหมาะสำหรับกรอ่านบาร์โค้ดของสินค้าที่มีขนาดใหญ่เคลื่อนที่ยาก เช่น ม้วนกระดาษใหญ่ที่ผลิตจากโรงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Wand Scanner เครื่องอ่านที่ให้แสงสีแดงอินฟราเรดในการอ่านต้องใช้เครื่องอ่านสัมผัสกับแถบบาร์โค้ด

5. Hand Held Laser Scanner เครื่องอ่านที่มีหลักการทำงานแบบ Moving Beam Scanner ที่ให้แสงเลเซอร์

ตัวอย่างเครื่องอ่านบาร์โค้ดประเภทต่างๆ



รูปที่ 2.2 เครื่องอ่านบาร์โค้ดประเภทต่างๆ

2.1.7 มาตรฐานบาร์โค้ด

บาร์โค้ดสามารถแบ่งออกตามชนิดของข้อมูลได้ดังนี้

- ตัวเลขเพียงอย่างเดียว (Numeric-only barcodes)
- ตัวอักษรและตัวเลข (Alphanumeric barcodes)
- 2 มิติ (2-Dimensional barcodes)

บาร์โค้ดมาตรฐานที่ใช้กันอยู่ปัจจุบันมีประมาณ 11 ระบบดังนี้

1. UPC – Uniform Product Code

บาร์โค้ดระบบแรกของโลก พัฒนาและทดลองใช้ครั้งแรกในปี ค.ศ. 1949 โดยชาวอเมริกันชื่อ Mr. Norm Woodland และ Mr. Barnard Silvers และสามารถใช้ได้อย่างสมบูรณ์เมื่อปี ค.ศ. 1973 โดย Uniform Code Council ตั้งอยู่ที่เมือง Dayton รัฐ โอไฮโอ สหรัฐอเมริกา ระบบนี้นิยมใช้กันมากในประเทศแคนาดาและสหรัฐอเมริกา



UPC (A) General
Bar Height: 21.0100 mm
Magnification: 80.13%
Printer dpi: 2400
Company: 3S INDUSTRIES
Client: 3S Industries

รูปที่ 2.3 ตัวอย่างมาตรฐานบาร์โค้ดระบบ UPC

2. EAN – European Article Number

เป็นระบบบาร์โค้ดที่กลุ่มประเทศในแถบยุโรปพัฒนาขึ้นใช้ เพราะเริ่มสนใจการใช้บาร์โค้ดระบบ UPC ของประเทศสหรัฐอเมริกา แล้วเสร็จเมื่อปี ค.ศ. 1976 จนปัจจุบัน EAN เป็นระบบที่ใช้กันอย่างแพร่หลาย ทั้งยุโรป ออสเตรเลีย และเอเชีย รวมทั้งประเทศไทยด้วย

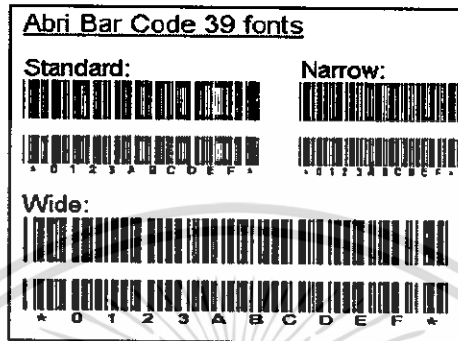


EAN-13
Bar Height: 26.2600 mm
Magnification: 99.42%
Printer dpi: 2400
Company: 3S INDUSTRIES
Client: 3S Industries

รูปที่ 2.4 ตัวอย่างมาตรฐานบาร์โค้ดระบบ EAN

3. CODE 39

ระบบนี้ได้รับการพัฒนาขึ้นในปี ค.ศ. 1974 โดย Dr. David Allais และ Ray Steven ชาวอเมริกัน วัตถุประสงค์เพื่อนำไปใช้ในธุรกิจอุตสาหกรรม ระบบนี้สามารถใช้ร่วมกับตัว อักษรได้เป็นระบบแรกและเก็บข้อมูลได้ปริมาณมาก



รูปที่ 2.5 ตัวอย่างมาตรฐานบาร์โค้ดระบบ CODE 39

4. ITF – INTERLEAVE 2 of 5

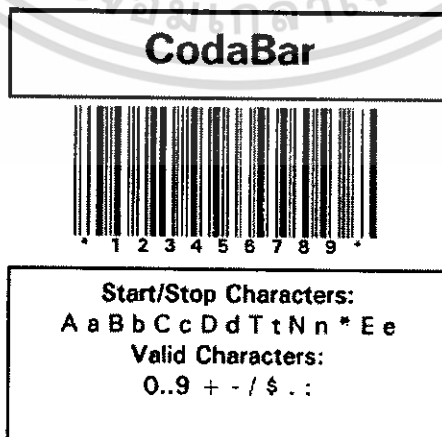
เรียกย่อๆ ว่า ITF เป็นบาร์โค้ดตัวใหญ่ใช้สำหรับหีบบรรจุสินค้าหรือเรียกว่า Case code



รูปที่ 2.6 ตัวอย่างมาตรฐานบาร์โค้ดระบบ IFT 2 of 5

5. CODABAR

ถูกพัฒนาขึ้นมาใช้กับธุรกิจเวชกรรมในปี ค.ศ. 1972 และนิยมใช้ในระบบงานห้องสมุด



รูปที่ 2.7 ตัวอย่างมาตรฐานบาร์โค้ดระบบ CodaBar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. CODE 128

ได้ถูกพัฒนาขึ้นและยอมรับว่าได้ใช้เป็นทางการในสหรัฐอเมริกา เมื่อปี ค.ศ. 1981 นิยมใช้ในวงการดีวีซีดีและแฟชัน ปัจจุบันกำลังเริ่มนิยมใช้ในสหรัฐอเมริกา

7. CODE 93

ได้เริ่มพัฒนาขึ้นในปี ค.ศ. 1982 ปัจจุบันเริ่มนิยมใช้ในสหรัฐอเมริกา

8. CODE 49

ได้เริ่มพัฒนาขึ้นในปี ค.ศ. 1987 โดย Dr. Davis Allais ผู้คิดค้น CODE 39 ได้ปรับปรุงพัฒนาให้บรรจุข้อมูลได้มากขึ้นด้วยพื้นที่เท่าเดิม

9. CODE 16k

เหมาะสำหรับอุตสาหกรรมผลิตสินค้าที่มีขนาดเล็กมาก พื้นที่ในการใส่บาร์โค้ดน้อย เช่น อุปกรณ์อะไหล่เครื่องบิน

10. ISBN/ ISSN – International Standard Book Number/ International Standard Serial Number

ใช้สำหรับหนังสือและนิตยสาร

ตัวอย่าง ISBN และ ISSN



รูปที่ 2.8 ตัวอย่างมาตรฐานบาร์โค้ดระบบ ISBN/ISSN

11. EAN/ UCC 128 (shipping container code)

เป็นระบบใหม่ ซึ่งเป็นการร่วมมือกันระหว่าง EAN ของยุโรป และ UCC ของสหรัฐอเมริกา โดยนำเอาระบบ EAN มาใช้ร่วมกับ CODE 128 เพื่อบอกรายละเอียดของสินค้ามากขึ้น เช่น วันเดือนปีที่ผลิตครั้งที่ผลิต วันที่สั่งซื้อ มีกิโล กี่ขนาด เป็นต้น

2.2 ไมโครคอนโทรลเลอร์

2.2.1 คุณสมบัติพื้นฐานของ MCS-51

คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
2. ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช สามารถลบและเขียนใหม่ได้พันครั้ง
3. หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำ

แบบอีพรอมเพิ่มเติม

4. ขาพอร์ดเป็นแบบสองทิศทาง สามารถใช้งานเป็น ได้ทั้งอินพุตและเอาต์พุต
5. มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
6. ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
7. สามารถรองรับแหล่งกำเนิดอินเตอร์รัพต์ได้ 6 ประเภท
8. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
9. มีวงจรกำเนิดสัญญาณพิกายูภายในชิป

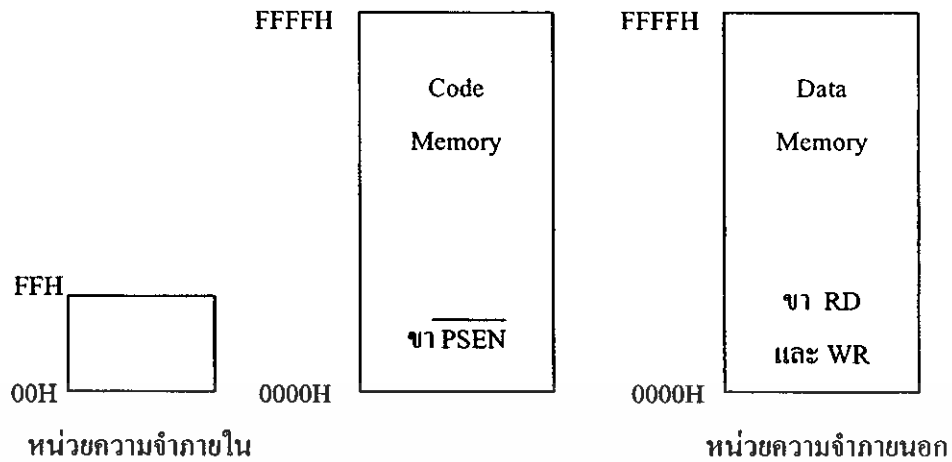
2.2.2 โครงสร้างหน่วยความจำ

2.2.2.1 หน่วยความจำโปรแกรมของ MCS-51

หน่วยความจำโปรแกรม(Program Memory) มีไว้เพื่อบรรจุคำสั่งหรือโปรแกรมที่ผู้ใช้พัฒนาขึ้น จัดเก็บไว้ในหน่วยความจำ โดยอาจจะประกอบอยู่ในตัวของไอซี 89C51 เอง หรือเป็นไอซีหน่วยความจำ EPROM หรือ ROM แยกออกต่างหากได้ ในกรณีหลังจำเป็นต้องมีการใช้พอร์ทอินพุตเอาต์พุต ทำหน้าที่เป็นบัสแอดเดรส และบัสข้อมูลเพื่อให้สามารถติดต่อกับหน่วยความจำมาตรฐานทั่วไปได้

หน่วยความจำโปรแกรมของ 89C51 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้ระบบข้อมูลเหล่านี้ก็ยังไม่สูญหายโครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซีหน่วยความจำประเภทต่างๆ เช่น หน่วยความจำแบบ ROM (Read Only Memory) หรือ EPROM (Erasable Programmable Read Only Memory)

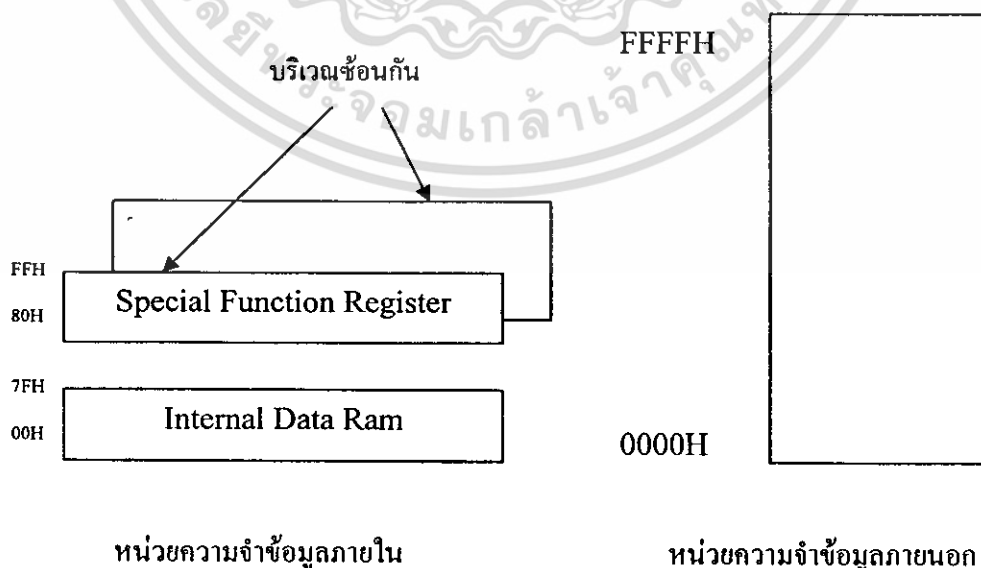
การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ 89C51 จะมีการจัดพื้นที่ดังรูปที่ 2.9 โดยที่ไมโครคอนโทรลเลอร์ 89C51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมสูงสุดได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะตามตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ในตัวไอซี ไมโครคอนโทรลเลอร์เองและหน่วยความจำภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ



รูปที่ 2.9 การจัดพื้นที่หน่วยความจำของไมโครคอนโทรลเลอร์

2.2.2.2 หน่วยความจำข้อมูลของ MCS-51

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยที่หน่วยความจำข้อมูลจะมีลักษณะเป็นหน่วยความจำ RAM แบบสแตติก (Static) ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับระบบ ก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ในหน่วยความจำนี้สูญหายไป พื้นที่ของหน่วยความจำข้อมูลของ 89C51 สามารถมีได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตำแหน่งที่ตั้งของหน่วยความจำนั้นดังลักษณะในรูปที่ 2.10 คือ หน่วยความจำโปรแกรมภายใน (Internal Data Memory) หรือ RAM ที่อยู่ภายในตัวไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 2.10 แสดงลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register)

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์หรือพอร์ทของ 89C51 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็

2.2.3.1 แอควิวมูลเตอร์ (Accumulator)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บข้อมูลที่ส่งให้หน่วยทำงานในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ A

2.2.3.2 รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณหารตัวเลข ในกรณีที่ไมใช้การคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

2.2.3.3 โปรแกรมเคาน์เตอร์(Program Counter)

เป็นรีจิสเตอร์ที่ใช้สำหรับการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรมซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ PC

2.2.3.4 สแตกพอยน์เตอร์(Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ของบริเวณสแตกสำหรับเก็บข้อมูลแอควิวมูลเตอร์รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม ค่าเริ่มต้นของสแตกจะอยู่ที่ตำแหน่ง 07H การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ SP

2.2.3.5 ตัวชี้ข้อมูลหรือดาต้าพอยน์เตอร์(Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจิสเตอร์ DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิต สองตัว คือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าแอดเดรสของหน่วยความจำที่ต้องใช้งานภายในโปรแกรม หรืออาจเป็นแอดเดรสของอุปกรณ์ภายนอก

2.2.3.6 โปรแกรมสแตตัสเวิร์ด(PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กส์สภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์(Bank) ของรีจิสเตอร์ที่ใช้งานด้วย

2.2.3.7 รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ท (Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุทเอาต์พุทโดยตรงซึ่งจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะการอินพุทหรือการเอาต์พุทข้อมูลได้

2.2.3.8 รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล

2.2.3.9 รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ควบคุมการทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรแกรมเซ็นเซอร์ การกำหนดอัตราทวิคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรมและแฟล็กส์ สภาวะการทำงานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.10 รีจิสเตอร์ IP,IE,TMOD,SCON

เป็นกลุ่มรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอินเทอร์รัพต์ต่างๆ

2.2.4 การโปรแกรมพอร์ตอนุกรม

โดยปกติแล้ว MCS-51 จะเป็นไมโครคอนโทรลเลอร์ที่มีความสามารถในการรับข้อมูลจากภายนอกและนำมาประมวลผล พร้อมทั้งสามารถส่งสัญญาณเพื่อทำการควบคุมการทำงานของอุปกรณ์ต่างๆ ได้อย่างดี และในส่วนของ การติดต่อสื่อสารข้อมูล (Data Communication) กับระบบภายนอกอื่นๆก็สามารถกระทำโดยผ่านทางพอร์ตอนุกรม (Serial Port) ซึ่งพอร์ตอนุกรมนี้นี้ จะเป็นส่วนที่เหมาะสมในการรับ หรือส่งข้อมูลในระยะทางไกลได้ดีกว่าพอร์ตนาน

2.2.4.1 รีจิสเตอร์ต่างๆที่เกี่ยวข้องในการใช้งานพอร์ตอนุกรม

1. รีจิสเตอร์ควบคุมไทม์เมอร์เนื่องจากว่าการใช้งานพอร์ตอนุกรมนั้นมีสิ่งที่จะต้องคำนึงถึงคือ อัตราการรับ-ส่งข้อมูล หรือ เรียกว่า อัตราบอด (Baud Rate) จริงๆ แล้ว ก็คือจังหวะการเคลื่อนข้อมูลเข้าหรือออกจาก MCS-51 นั่นเอง โดยอัตราบอดนี้สามารถสร้างขึ้นภายในชิพของ MCS-51 ได้จากไทม์เมอร์แซนแนล 1 โดยทำงานในโหมด 2 คือ โหมดค่ากลับอัตราโนมิติ ดังนั้นรีจิสเตอร์ที่ต้องทำการโปรแกรม มีดังนี้

- TMOD ตำแหน่ง 89H ทำหน้าที่เลือกโหมดของไทม์เมอร์
- TCON ตำแหน่ง 88H ทำหน้าที่เริ่มต้นการสร้างอัตราบอด
- TH1 ตำแหน่ง 8CH ทำหน้าที่ใส่ข้อมูลการนับของไทม์เมอร์ 1 เพื่อสร้างอัตราบอด

2. รีจิสเตอร์ควบคุมการลดกำลัง เนื่องจากว่า การสร้างอัตราบอดนั้นจะต้องนำบิตในรีจิสเตอร์ PCON มาใช้ในการคำนวณข้อมูลของ TH1 ดังนั้นรีจิสเตอร์ที่ใช้ก็คือ

- PCON ตำแหน่ง 87H ทำหน้าที่ในการคำนวณข้อมูลที่จะใส่ในรีจิสเตอร์ TH1

ดังนี้

ตารางที่ 2.2 แสดงบิตภายใน PCON

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SMOD	-	-	-	GF1	GF0	PD	IDL

SMOD บิตกำหนดอัตราความเร็วการรับส่งข้อมูลอนุกรม UART

0 = อัตราความเร็ว 1 เท่า

1 = อัตราความเร็ว 2 เท่า

GF1 แฟล็กใช้งานทั่วไป ไม่เกี่ยวข้องกับ การควบคุมใด ๆ

GF2 แฟล็กใช้งานทั่วไป ไม่เกี่ยวข้องกับ การควบคุมใด ๆ

PD Power down Bit 1 = หยุดออกสวิตช์เลเตอร์ของซีพียูสัญญาณรีเซต หรือ อินเทอร์รัพต์ (83C154, 83C154D เท่านั้น) ที่จะยกเลิกโหมดนี้

IDL Idle Mode Bit 1 = หยุดการทำงานของซีพียูสัญญาณรีเซต หรือ อินเทอร์รัพต์เท่านั้นที่จะยกเลิกโหมดนี้

3. รีจิสเตอร์ควบคุมการอินเทอร์รัพต์ เนื่องจากว่า MCS-51 สามารถใช้งานพอร์ตอนุกรม ในลักษณะของการอินเทอร์รัพต์ จึงมีรีจิสเตอร์ที่เกี่ยวข้องดังนี้

- IE ตำแหน่ง A8H ทำหน้าที่ยอมให้เกิดการอินเทอร์รัพต์จากพอร์ตอนุกรมได้หรือไม่

- IP ตำแหน่ง B8H ทำหน้าที่จัดลำดับความสำคัญของการอินเทอร์รัพต์

4. รีจิสเตอร์ควบคุมพอร์ตอนุกรม การใช้งานพอร์ตอนุกรมจะขึ้นอยู่กับรีจิสเตอร์โดยตรงคือ

- SBUF ตำแหน่ง 99H ทำหน้าที่เป็นบัฟเฟอร์การรับหรือส่งข้อมูลของพอร์ตอนุกรม

- SCON ตำแหน่ง 98H ทำหน้าที่ควบคุมและกำหนดโหมดการใช้งานพอร์ตอนุกรมทั้งหมดซึ่งมีรายละเอียดของบิตต่างๆ ดังนี้

ตารางที่ 2.3 แสดงบิตภายใน SCON

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

ตารางที่ 2.4 แสดงการเลือกโหมด

SM0, SM1 บิตเลือกโหมดการทำงาน

SM0	SM1	โหมด	การทำงาน
0	0	0	Shift Register ความเร็วในการรับส่งข้อมูลเท่ากับ (1/12) ของ OSC
0	1	1	8 บิต UART ความเร็วในการรับส่งข้อมูลกำหนดจาก Timer 1,2
1	0	2	9 บิต UART ความเร็วในการรับส่งข้อมูล = (1/32) หรือ (1/64) เท่าของ CPU OSC โดยขึ้นกับบิต SMOD ใน PCON
1	1	3	9 บิต UART ความเร็วในการรับส่งข้อมูลกำหนดจาก Timer 1,2

SM2 บิตเลือกการทำงานแบบ Single Processor Environment หรือ Multiprocessor Environment

1: เลือก Multiprocessor Environment ใช้ได้กับโหมด 2, 3

0: เลือก Single Processor Environment ใช้ได้กับทุกโหมด

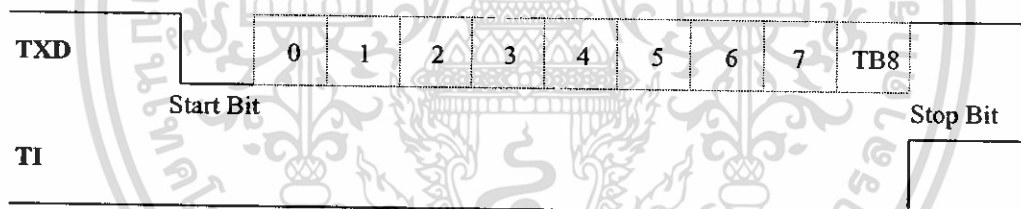
- REN บิตควบคุมให้รับหรือไม่รับข้อมูล
 1: ให้รับข้อมูลได้
 0: ห้ามรับข้อมูล
- TB8 ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2, 3 ให้ใส่ในบิตนี้ได้เลย
- RB8 ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมากับในบิตนี้
- TI แฟล็ก TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์
- RI แฟล็ก RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมจะต้องเคลียร์เอง)

2.2.4.2 การส่งข้อมูลแบบอนุกรมในโหมด 3

พอร์ตสื่อสารอนุกรม 11 บิต UART โดย ข้อมูล (DATA) 8 บิต, 1 สตาร์ทบิต (Start Bit) และ 1 สต็อปปบิต (Stop Bit) เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer 1 สำหรับ 8051

สมการแสดงการคำนวณหาอัตราบอดในโหมด 3

$$\text{BaudRateMode3} = \frac{2^{\text{SMOD}} \times \text{CPUOSC}}{32 \times 12 \times [256 - (\text{TH1})]}$$



รูปที่ 2.11 รูปแบบการรับส่งข้อมูลในโหมด 3

2.3 เครื่องรับส่งวิทยุ

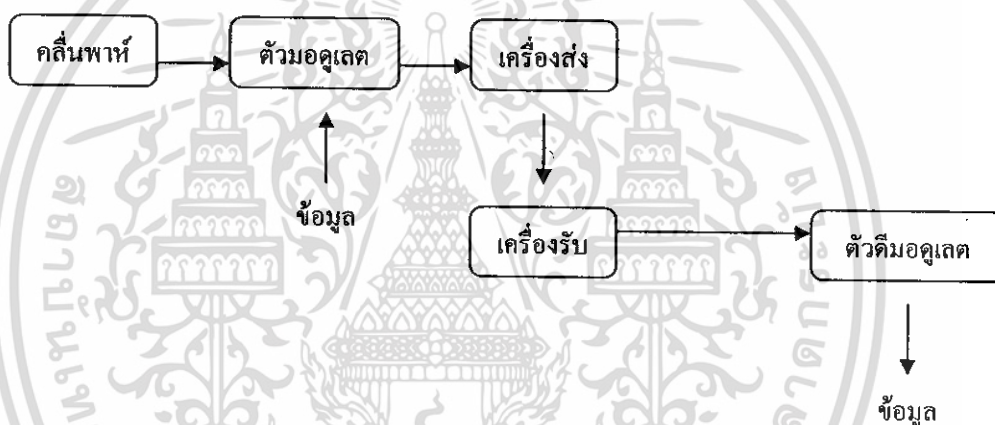
สัญญาณอิเล็กทรอนิกส์ที่ใช้ในการสื่อสารสามารถแบ่งได้เป็น 2 ประเภทคือ สัญญาณอนาลอก และสัญญาณดิจิทัล สัญญาณอนาลอกได้แก่สัญญาณเสียง และสัญญาณในธรรมชาติทั้งหมด ปัญหาที่สำคัญของสัญญาณอนาลอกก็คือเรื่องสัญญาณรบกวน ซึ่งในบางครั้งอาจทำให้ระบบไม่สามารถใช้งานได้ ดังนั้นจึงมีการนำสัญญาณดิจิทัลเข้ามาแทนที่

- สัญญาณแบบอนาลอก (Analog Signal) จะเป็นสัญญาณแบบต่อเนื่องที่ทุกๆ ค่า ที่เปลี่ยนแปลงไป ของระดับสัญญาณจะมีความหมาย การส่งสัญญาณแบบนี้จะถูกรบกวนให้มีการแปลความหมายผิดพลาดได้ง่าย เนื่องจากค่าทุกค่าถูกนำมาใช้งาน ซึ่งสัญญาณ

แบบอนาลอกนี้จะเป็นสัญญาณที่สื่อกลางในการสื่อสารส่วนมากใซ้อยู่เช่นสัญญาณเสียงในสายโทรศัพท์ เป็นต้น

- สัญญาณแบบดิจิทัล (Digital Signal) จะประกอบขึ้นจากระดับสัญญาณเพียง 2 ค่าคือสัญญาณ ระดับสูงสุด และสัญญาณระดับต่ำสุด ดังนั้นจะมีประสิทธิภาพ และความน่าเชื่อถือสูงกว่าแบบอนาลอกเนื่องจากการใช้งานค่าสองค่า เพื่อนำมาตีความหมายเป็น on/off หรือ 0/1 เท่านั้น ซึ่งเป็นสัญญาณที่ คอมพิวเตอร์ใช้ในการติดต่อสื่อสารกัน

2.3.1 การมอดูเลต (Modulation) เป็นการผสมสัญญาณของข้อมูลเข้าไปกับสัญญาณอีกสัญญาณหนึ่งเรียกว่า คลื่นพาห์ (carrier) ซึ่งสัญญาณนี้มีความถี่ที่เหมาะสมกับช่องสัญญาณนั้นๆ เพื่อให้ข้อมูลที่ส่งเข้าไปในช่องสัญญาณเดินทางได้ไกลมากขึ้น การเลือกวิธีมอดูเลตขึ้นอยู่กับปัจจัยหลายประการ เช่น ชนิดของสัญญาณ แบนด์วิดท์ ประสิทธิภาพของระบบที่ต้องการ และความต้านทานต่อสัญญาณรบกวน เป็นต้น



รูปที่ 2.12 รูปแบบของการสื่อสารในการรับส่งสัญญาณ

จากรูปแสดงรูปแบบของการสื่อสารในการรับส่งสัญญาณอย่างง่าย โดยคลื่นพาห์ผสมสัญญาณข้อมูลที่ตัวมอดูเลต (Modulator) แล้วส่งไปที่เครื่องส่ง จากเครื่องส่งไปยังเครื่องรับจะเป็นช่องสัญญาณสำหรับลำเลียงสัญญาณผสมนี้ สัญญาณผสมจากเครื่องรับจะไปเข้าตัวดีมอดูเลต (Demodulate) เพื่อแยกสัญญาณข้อมูลออกมา เพื่อให้เราสามารถสื่อสารไปได้ไกลยิ่งขึ้น จำเป็นต้องมอดูเลตสัญญาณข่าวสารลงพาหะ วิธีมอดูเลตแบ่งออกเป็น 2 แบบที่สำคัญคือ มอดูเลตทางแอมพลิจูด และมอดูเลตเชิงมุมซึ่งการมอดูเลตทั้งสองแบบนี้เป็นการมอดูเลตเชิงอนาลอก (Analog)

การสื่อสารอนาลอกเป็นระบบที่ออกแบบให้ส่งข้อมูลสัญญาณอนาลอกเช่น สัญญาณเสียง แต่ได้มีการพัฒนาจนประยุกต์ให้สามารถส่งข่าวสารได้ด้วยในปัจจุบัน ปัญหาสำคัญสำหรับการสื่อสารอนาลอกก็คือเรื่องสัญญาณรบกวน แต่เนื่องจากสัญญาณในธรรมชาติทั้งหมดเป็นสัญญาณอนาลอกจึงยังคงเห็นการพัฒนาของการสื่อสารแบบอนาลอกในปัจจุบัน เช่น การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM) การมอดูเลตความถี่ (Frequency Modulation หรือ FM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การมอดูเลตที่ใช้อยู่ในปัจจุบัน 3 วิธีได้แก่

2.3.1.1 การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM) วิธีนี้แอมพลิจูดของคลื่นพาห้จะเปลี่ยนแปลงตามสัญญาณของข้อมูลที่เข้ามา การมอดูเลตแบบ AM เป็นวิธีที่ง่ายที่สุดในการมอดูเลต แต่คุณภาพของสัญญาณไม่ดี มีความต้านทานสัญญาณรบกวนต่ำ เหมาะกับข้อมูลที่ไม่ต้องการคุณภาพมากนัก เช่น สัญญาณเสียง เป็นต้น

2.3.1.2 การมอดูเลตความถี่ (Frequency Modulation หรือ FM) วิธีการนี้เป็นการเปลี่ยนแปลงความถี่ของคลื่นพาห้ตามสัญญาณของข้อมูลที่เข้ามา การมอดูเลตแบบความถี่ ให้คุณภาพที่ดีกว่าการมอดูเลตแบบแอมพลิจูด แต่ระบบจะซับซ้อนกว่า

2.3.1.3 การมอดูเลตเฟส (Phase Modulation หรือ PM) เป็นการมอดูเลตที่ใช้ในการเปลี่ยนแปลงเฟสของคลื่นพาห้ตามสัญญาณข้อมูลข้อมูลที่เข้ามา ทั้งคุณภาพของสัญญาณและความซับซ้อนไม่ค้ยแตกต่างจากการมอดูเลตแบบความถี่เท่าใดนัก ข้อแตกต่างระหว่างการมอดูเลตแบบความถี่ กับการมอดูเลตแบบเฟส คือการมอดูเลตแบบเฟสใช้คลื่นพาห้เพียงความถี่เดียว การมอดูเลตและดีมอดูเลตสามารถทำได้ประหยัดกว่า แต่ไม่ได้หมายความว่าซับซ้อนน้อยกว่า

2.3.2 การมอดูเลตแอมพลิจูดแบบแถบข้างคู่จัดคลื่นพาห้

ในกระบวนการมอดูเลตแอมพลิจูดนั้น ขนาดของสัญญาณข่าวสารจะถูกใช้ไปบังคับค่าแอมพลิจูด A_c ของคลื่นพาห้ $A_c \cos(\omega_c t + \theta_c)$ ให้เปลี่ยนแปลงในขณะที่ความถี่ ω_c และเฟส θ_c จะยังมีค่าคงเดิม สมการของคลื่นที่มอดูเลตแล้ว $\phi_{DSB-SC}(t)$ จะเป็นดังนี้คือ

$$\phi_{DSB-SC}(t) = km(t) \cos(\omega_c t + \theta_c) \quad (2-1)$$

โดย k คือค่าคงที่ ที่กระบวนการมอดูเลตจัดการควบคุมสัดส่วนของ A_c ให้แปรผันตามสัญญาณข่าวสาร $m(t)$

เพื่อความสะดวกในการวิเคราะห์สัญญาณ เราสามารถสมมุติให้ $k=1$ และ $\theta_c=0$ ได้โดยไม่เสียความหมายของการวิเคราะห์สัญญาณ โดยทั่วไปแต่อย่างไร เพียงแต่จะทำให้รูปของสมการที่กระชับขึ้น คือ จะได้สมการเป็น

$$\phi_{DSB-SC}(t) = m(t) \cos(\omega_c t) \quad (2-2)$$

เพื่อที่จะทำความเข้าใจถึงการเปลี่ยนแปลงที่เกิดขึ้นในโดเมนความถี่ จะสมมุติให้ $M(\omega)$ คือค่าฟังก์ชันสเปกตรัมของ $m(t)$ กล่าวคือ

$$m(t) \leftrightarrow M(\omega) \quad (2-3)$$

โดยอาศัยคุณสมบัติของการแปลงฟูเรียร์ จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m(t)\cos(\omega_c t) \leftrightarrow \frac{1}{2}[M(\omega + \omega_c) + M(\omega - \omega_c)] \quad (2-4)$$

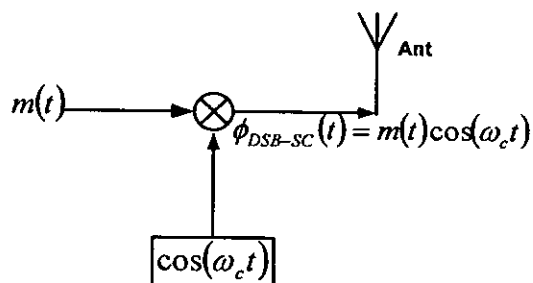
นั่นคือสัญญาณเอเอ็มในโดเมนความถี่ $\Phi_{DSB-SC}(\omega)$ จะมีค่าเป็น

$$\Phi_{DSB-SC}(\omega) = \frac{1}{2}[M(\omega + \omega_c) + M(\omega - \omega_c)] \quad (2-5)$$

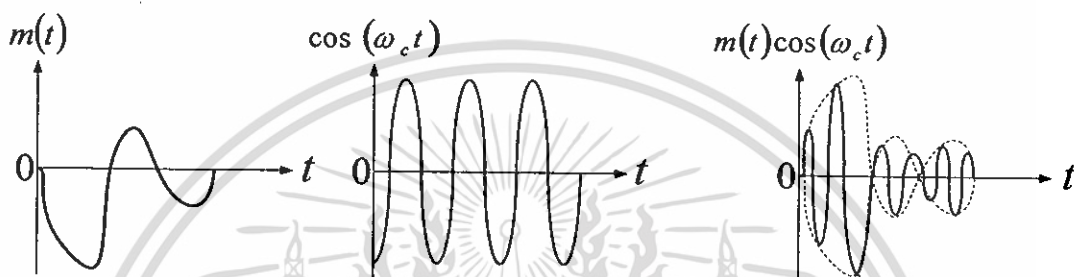
ถ้าแบนด์วิดท์ของ $M(\omega)$ ตามสมการ (2-3) มีค่าเป็น B เฮิรตซ์หรือ W เรเดียนต่อวินาที ($W = 2\pi B$) เราจะได้จากสมการ (2-5) ได้ว่า $\Phi_{DSB-SC}(\omega)$ จะมีแบนด์วิดท์เป็น $2W$ เรเดียนต่อวินาที ดังรูปที่ 2.13 และเป็นที่น่าสนใจว่า ค่าความถี่กึ่งกลางแบนด์ของสัญญาณเอเอ็มนั้นอยู่ที่ความถี่ของคลื่นพาห้คือ ω_c ซึ่งแกนตั้งที่ ω_c ในโดเมนของความถี่นี้จะแบ่งสเปกตรัมที่เกิดขึ้นออกเป็นสองส่วนที่สมมาตรกัน ส่วนของสัญญาณที่มีสเปกตรัมอยู่ที่ความถี่สูงกว่าและต่ำกว่าความถี่ ω_c นั้นมีชื่อว่าแถบข้างส่วนบนหรือไซด์แบนด์สูง (upper side band) นิยมเขียนย่อแทนด้วย ยูเอสบี (USB) และแถบข้างส่วนล่างหรือไซด์แบนด์ต่ำ (lower side band) นิยมเขียนย่อด้วย แอลเอสบี (LSB) ตามลำดับในทำนองเดียวกันบนแกนความถี่ด้านลบค่าความถี่ $-\omega_c$ ก็จะทำให้เกิดไซด์แบนด์สูงและไซด์แบนด์ต่ำเช่นกันแต่ในกรณีของความถี่ด้านลบไซด์แบนด์สูงจะหมายถึงส่วนสเปกตรัมที่มีความถี่เป็นลบมากกว่า $-\omega_c$ และไซด์แบนด์ต่ำจะหมายถึงส่วนสเปกตรัมที่มีความถี่เป็นลบน้อยกว่า $-\omega_c$

ควรสังเกตอีกด้วยว่า ถ้าสัญญาณ $m(t)$ นั้นไม่มีส่วนประกอบเชิงความถี่ที่เป็นไฟตรง ไซด์แบนด์ทั้งสองที่เกิดจากการมอดูเลตก็จะอยู่ห่างกันอย่างเด่นชัด และสเปกตรัมของสัญญาณเอเอ็มก็จะไม่เกิดมีองค์ประกอบความถี่ที่ $\omega = \pm\omega_c$ ปรากฏ กล่าวคือ ส่วนประกอบสัญญาณที่มีความถี่ตรงกับความถี่ของคลื่นพาห้จะถูกขจัดออกไปในกรณีดังกล่าว ด้วยเหตุนี้เองทำกระบวนการมอดูเลตสัญญาณแบบนี้ได้ชื่อว่า การมอดูเลตแอมพลิจูดแบบไซด์แบนด์คู่ขจัดคลื่นพาห้ (AM double-sideband-suppressed carrier) หรือการมอดูเลตแอมพลิจูดแบบแถบข้างคู่ขจัดคลื่นพาห้ซึ่งเรียกย่อว่าดีเอสบี เอสซี (DSB-SC) ตามสมการ (2-5) จะบอกให้รู้ว่าการมอดูเลตแบบนี้จะทำให้เกิดการย้ายสเปกตรัมของสัญญาณข่าวสาร $M(\omega)$ ออกไปจากเดิม $\pm\omega_c$ (หมายถึงย้ายไปจากเดิม $+\omega_c$ และ $-\omega_c$ ตามลำดับ) ในกรณีที่แยกสัญญาณข่าวสาร $m(t)$ ออกมาจาก $\phi_{DSB-SC}(t)$ นั้นลักษณะการที่ทำหน้าที่นี้เรียกว่า การกู้สัญญาณ หรือ การดีมอดูเลต (Demodulate) สัญญาณหรือการตรวจจับอีกครั้งหนึ่ง ซึ่งเมื่อทำเช่นว่านี้แล้ว จะได้ผลลัพธ์ $v_d(t)$ เป็น

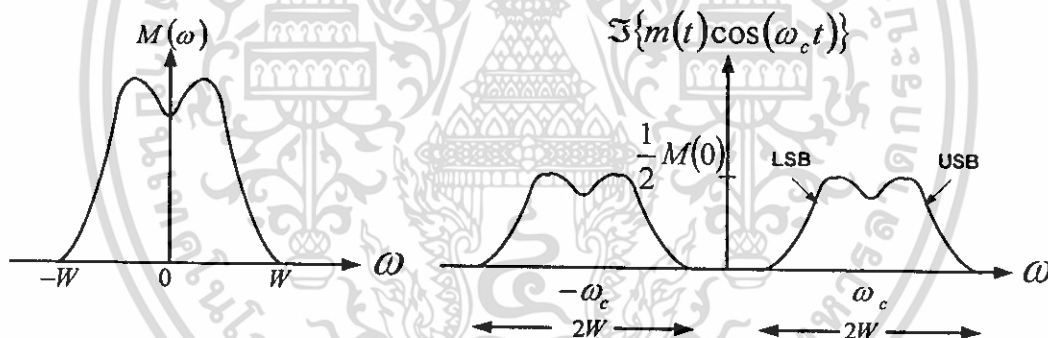
$$\begin{aligned} v_d(t) &= \phi_{DSB-SC}(t)\cos(\omega_c t) \\ &= \{m(t)\cos(\omega_c t)\}\cos(\omega_c t) \\ &= m(t)\cos^2(\omega_c t) \\ &= \frac{1}{2}m(t) + \frac{1}{2}m(t)\cos(2\omega_c t) \end{aligned} \quad (2-6)$$



(ก) แผนภาพแสดงกระบวนการมอดูเลตสัญญาณดีเอสบี เอสซี



(ข) สัญญาณข่าวสาร $m(t)$ (ค) คลื่นพาห์ $\cos(\omega_c t)$ (ง) สัญญาณดีเอสบี เอสซี



(จ) ความหนาแน่นของสเปกตรัมของ $m(t)$ (ฉ) ความหนาแน่นของสัญญาณดีเอสบี เอสซี

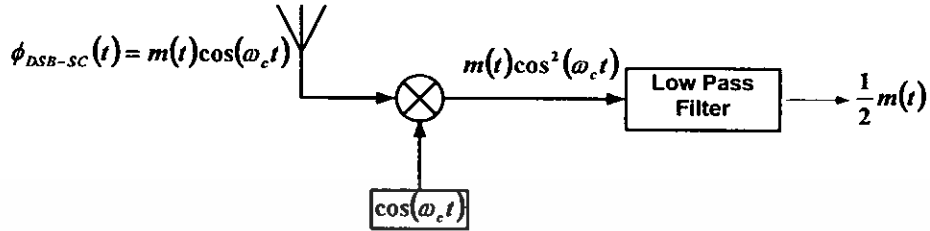
ในที่นี้ $W = 2\pi B$

รูปที่ 2.13 กระบวนการมอดูเลตสัญญาณดีเอสบี เอสซี

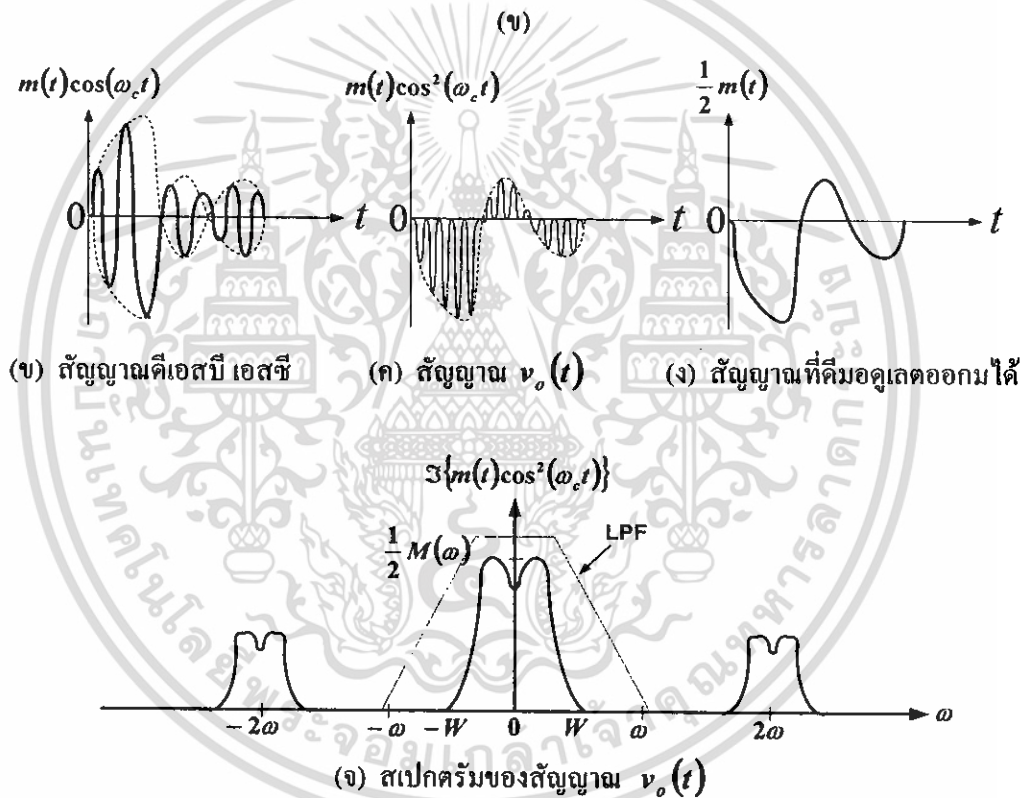
และ โดยคุณสมบัติของการแปลงฟูเรียร์จะได้

$$\phi_{DSB-SC}(t)\cos(\omega_c t) \leftrightarrow \frac{1}{2}M(\omega) + \frac{1}{4}[M(\omega + 2\omega_c) + M(\omega - 2\omega_c)] \quad (2-7)$$

เมื่อนำสัญญาณ $\phi_{DSB-SC}(t) \cos(\omega_c t)$ นี้ไปผ่านวงจรกรองความถี่ต่ำผ่านเพื่อขจัดสัญญาณที่มีความถี่สูงคือพจน์ที่อยู่ในวงเล็บใหญ่ของ (2-4) ออก ก็จะเหลือเพียงแต่ส่วนของสเปกตรัม $\frac{1}{2}M(\omega)$ ออกมาซึ่งก็คือสเปกตรัมของสัญญาณเดิม $\frac{1}{2}m(t)$ นั่นเอง



(ก) แผนภาพแสดงกระบวนการคืนมอดูเลตสัญญาณคือเอสบี เอสซี



รูปที่ 2.14 การคืนมอดูเลตสัญญาณคือเอสบี เอสซี และสัญญาณต่าง ๆ พร้อมทั้งค่าความหนาแน่นสเปกตรัมของสัญญาณที่อินพุตของวงจรกรองความถี่ต่ำผ่าน

อย่างไรก็ดีในการตรวจจับสัญญาณด้วยวิธีการดังกล่าวนี้ มักจะพบปัญหาอันเนื่องมาจากคลื่นพาห้ที่เครื่องรับสร้างขึ้นมานั้น มักจะมีความถี่และเฟสไม่เท่ากับความถี่และเฟสของคลื่นพาห้จากเครื่องส่ง การที่จะเข้าใจถึงผลของความคลาดเคลื่อนของคลื่นพาห้เหล่านั้น ทำได้โดยสมมุติว่าเมื่อมีสัญญาณจากเครื่องส่งคือ $\phi_{DSB-SC}(t) = m(t)\cos(\omega_c t)$ นั้นคลื่นพาห้จากออสซิลเลเตอร์ทางเครื่องรับมีความ

คลาดเคลื่อนไปเล็กน้อยเป็น $\cos\{(\omega_c + \Delta\omega)t + \delta\}$ โดย $\Delta\omega$ และ δ คือค่าความคลาดเคลื่อนทางความถี่และเฟสตามลำดับ

ตามกระบวนการตีมอดูเลต เครื่องรับจะทำการตีมอดูเลตสัญญาณโดยใช้คลื่นพาห้ $\cos\{(\omega_c + \Delta\omega)t + \delta\}$ คูณกับ $\phi_{DSB-SC}(t)$ ทำให้ได้สัญญาณลัพธ์เป็น

$$\begin{aligned} v_d(t) &= \phi_{DSB-SC}(t)\cos\{(\omega_c + \Delta\omega)t + \delta\} \\ &= m(t)\cos(\omega_c t)\cos\{(\omega_c + \Delta\omega)t + \delta\} \\ &= \frac{1}{2}m(t)\cos(\Delta\omega t + \delta) + \frac{1}{2}m(t)\cos\{(2\omega_c + \Delta\omega)t + \delta\} \end{aligned} \quad (2-8)$$

วงจรกรองความถี่ต่ำผ่านในเครื่องรับจะขจัดส่วนความถี่สูง คือ พจน์หลังสุดของสมการ (2-8) ออก คงเหลือสัญญาณเอาต์พุต $v_o(t)$ ของวงจรตรวจจับสัญญาณ คือ

$$v_o(t) = \frac{1}{2}m(t)\cos(\Delta\omega t + \delta) \quad (2-9)$$

ถ้าเครื่องผลิตคลื่นพาห้ขึ้นมาโดยไม่มีความคลาดเคลื่อนกล่าวคือ $\Delta\omega = 0$ และ $\delta = 0$ จากสมการ(2-9) จะพบว่า เพราะ $\cos(0) = 1$ ดังนั้น $v_o(t)$ จึงมีค่าเท่ากับ $\frac{1}{2}m(t)$ ซึ่งเป็นสัญญาณข่าวสาร $m(t)$ ที่ถูกต้อง แต่ถ้าเครื่องรับสร้างคลื่นพาห้ได้มีความถี่ถูกต้อง (คือ $\Delta\omega = 0$) แต่ทว่ายังมีความคลาดเคลื่อนทางเฟสอยู่บ้าง (คือ $\delta \neq 0$) ในกรณีนี้เอาต์พุต ตามสมการ (2-8) จะกลายเป็น

$$v_o(t) = \frac{1}{2}m(t)\cos(\delta) \quad (2-10)$$

เนื่องจาก $|\cos(\delta)| \leq 1$ ดังนั้นผลที่เกิดขึ้นก็คือ สัญญาณเอาต์พุตจะมีระดับลดลง และ ถ้า $\delta = \pm\frac{\pi}{2}$ ระดับของสัญญาณเอาต์พุตก็จะเป็นศูนย์ (เพราะ $\cos\left(\pm\frac{\pi}{2}\right) = 0$)

ถ้าสมมุติว่าเครื่องรับสร้างคลื่นพาห้ที่มีเฟสเริ่มต้นถูกต้อง ($\delta = 0$) แต่มีความถี่คลาดเคลื่อน ($\Delta\omega \neq 0$) จะพบว่าเอาต์พุตตามสมการ (2-9) จะได้เป็น

$$v_o(t) = \frac{1}{2}m(t)\cos(\Delta\omega t) \quad (2-11)$$

ซึ่งแสดงให้เห็นว่า $v_o(t)$ มีลักษณะผิดเพี้ยนไปจาก $m(t)$ ด้วยอิทธิพลของการบึงคับขนาด (การถุ่น) ด้วยคลื่นรูปไซน์ที่มีความถี่ $\Delta\omega$ ซึ่งเป็นความถี่ต่ำ ผลที่เกิดขึ้นนี้เรียกว่า มีการบิต (Beating) ด้วยความถี่ต่ำนั้น ทำให้เอาต์พุตไม่ใช่สัญญาณข่าวสาร $m(t)$ เดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราได้ว่า การตรวจจับสัญญาณดีเอสบี เอสซี นั้นทำได้ด้วยการมอดูเลตคลื่นพาห์ที่สร้างขึ้นทางเครื่องรับซ้ำเข้ากับสัญญาณ $(\phi_{DSB-SC}(t))$ นั้นอีกครั้งหนึ่ง แล้วใช้วงจรกรองความถี่ต่ำผ่านมาจัดส่วนประกอบที่เป็นความถี่สูงออก ก็จะได้สัญญาณข่าวสารเดิมกลับคืนมาประเด็นสำคัญในกระบวนการนี้คือ คลื่นพาห์ทางเครื่องรับจะต้องมีความถี่และเฟสตรงกับคลื่นพาห์จากเครื่องส่งพอดี การตรวจจับสัญญาณจึงจะมีประสิทธิภาพเต็มที่ จึงทำให้กระบวนการตรวจจับสัญญาณแบบนี้มีชื่อว่า การดีมอดูเลตแบบสัมพันธ์ (Synchronous demodulation) หรือการดีมอดูเลตแบบร่วมนัย (Coherent demodulation)

การสร้างวงจรมอดูเลตสัญญาณแบบแถบข้างคู่จับคลื่นพาห์ อาจทำได้โดยใช้วงจรมอดูเลตแบบตัด (Chopper modulator) แบบไม่เชิงเส้น ฯลฯ และที่เคยได้รับความนิยมมากคือ ริงมอดูเลเตอร์ (Ring modulator) สำหรับวงจรรับนั้นวงจรเครื่องรับที่น่าสนใจคือเครื่องรับแบบคอสแทต ซึ่งเป็นเครื่องรับที่นิยมใช้กันอยู่ สามารถที่จะปรับค่าความถี่ของสัญญาณจากออสซิลเลเตอร์ภายในเครื่องรับให้ตรงกับความถี่ของคลื่นพาห์จากเครื่องส่งได้โดยอัตโนมัติ

2.3.3 การมอดูเลตแอมพลิจูดสำหรับวิทยุกระจายเสียง

จากที่ได้รู้ว่าการดีมอดูเลตสัญญาณดีเอสบี เอสซี และสัญญาณเอสเอสบี สามารถทำได้โดยวิธีการดีมอดูเลตแบบสัมพันธ์ ซึ่งเป็นวิธีการที่จะต้องใช้ในการผลิตคลื่นพาห์ทางด้านเครื่องรับให้มีความถี่และเฟสให้เท่ากับค่าของคลื่นพาห์จากเครื่องส่ง และการที่จะควบคุมให้เครื่องรับสามารถสร้างคลื่นพาห์ที่ต้องการได้ จะต้องอาศัยวงจรที่มีความซับซ้อนพอควร ซึ่งย่อมเป็นการแน่นอนว่า เมื่อเป็นเช่นนี้ต้นทุนในการสร้างเครื่องรับดังกล่าวก็ย่อมจะต้องมีราคาสูงด้วย เพื่อลดปัญหาในเรื่องนี้จึงได้เกิดการคิดหาวิธีการที่จะดีมอดูเลตสัญญาณ ดีเอสบี เอสซี ชนิดที่ไม่จำเป็นต้องมีการสร้างคลื่นพาห์ทางด้านเครื่องรับขึ้น วิธีการนี้ทำได้โดยการใช้คลื่นพาห์ $\cos(\omega_c t)$ ซึ่งมีขนาดที่เหมาะสมส่งรวมไปกลับสัญญาณดีเอสบี เอสซี จากเครื่องส่ง การกระทำเช่นนี้ทำให้คลื่นสัญญาณที่ส่งมีรูปฟังก์ชันเป็น

$$\phi_{AM}(t) = A_c \cos(\omega_c t) + m(t)\cos(\omega_c t) \quad (2-12)$$

โดยในที่นี้ A_c คือ ค่าแอมพลิจูดของคลื่นพาห์ที่ใช้เพิ่มเข้ามาและเมื่อจัดรูป (2-12) ใหม่จะได้

$$\phi_{AM}(t) = A_c \left(1 + \frac{m(t)}{A_c} \right) \cos(\omega_c t) \quad (2-13)$$

ปกติค่าความถี่ของคลื่นพาห์ ω_c จะมีค่าสูงกว่าค่าความถี่ที่เป็นองค์ประกอบสูงสุดของสัญญาณ $m(t)$ มาก ดังนั้นถ้าขนาดของคลื่นพาห์ A_c มีค่ามากพอที่จะทำให้พจน์ในวงเล็บแรกของ (2-13) มีค่าไม่เป็นลบตลอดเวลาทุกค่า t ใดๆ แล้ว สัญญาณ $\phi_{AM}(t)$ ตามสมการ (2-13) ก็จะมียอดคลื่นที่ค่อยๆ เปลี่ยนไปตามเอนเวโลป (Envelope) หรือกรอบที่มีลักษณะเหมือนกับสัญญาณ $m(t)$ ทุกประการ เพราะฉะนั้นถ้านำสัญญาณ $\phi_{AM}(t)$ นี้ไปป้อนให้กับ วงจรตรวจจับกรอบสัญญาณ (Envelope detector) แล้วก็จะทำให้เกิดการตรวจจับแยกสัญญาณ $m(t)$ กลับออกมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

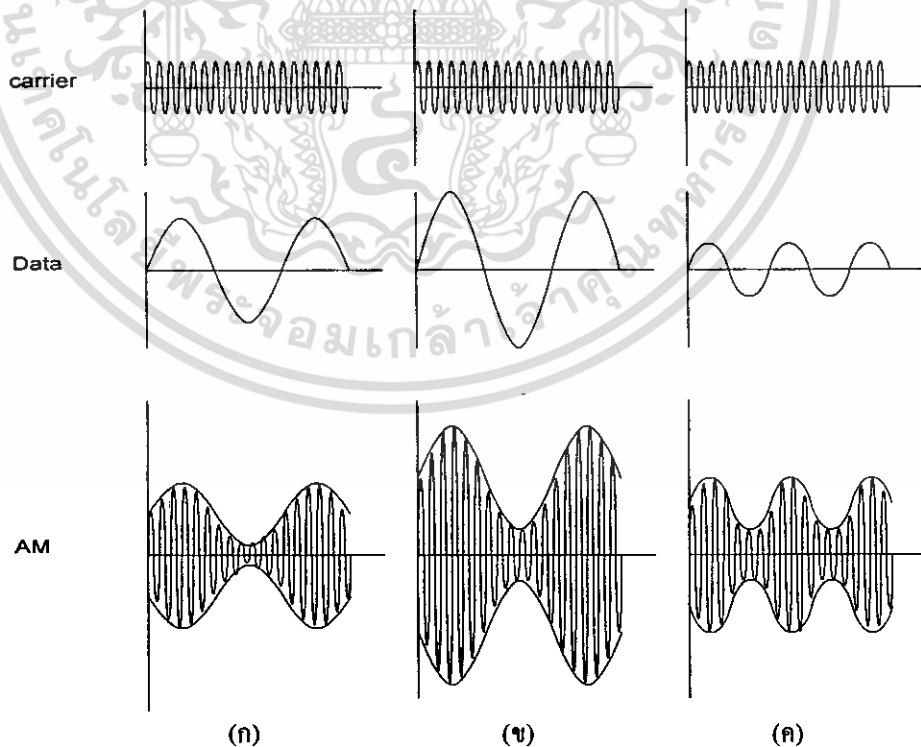
อย่างไรก็ดีสมการของสัญญาณเอเอ็ม (Standard AM) โดยทั่วไปนั้นนิยมเขียนดัดแปลงไปจาก (2-13) เล็กน้อยคือ

$$\phi_{AM}(t) = A_c(1 + k_a m(t))\cos(\omega_c t) \quad (2-14)$$

โดยที่ k_a นั้นเทียบได้กับ $\frac{1}{A_c}$ และมีชื่อเรียกว่าความไวของการมอดูเลตแอมพลิจูด (AM modulation sensitivity) แต่อย่างไรก็ดี k_a นี้ไม่จำเป็นต้องเท่ากับ $\frac{1}{A_c}$ แต่จะต้องมีค่าที่ไม่ทำให้ $1 + k_a m(t)$ มีค่าเป็นลบ

2.3.4 รูปคลื่นของสัญญาณเอเอ็ม

จากรูปที่ 2.15 (ก) จะเห็นว่ารูปของสัญญาณเอเอ็ม นั้นจะมีความถี่ตามคลื่นพาห์แต่ความสูงจะเปลี่ยนแปลงตามสัญญาณข้อมูล ซึ่งเมื่อสัญญาณข้อมูลมีความสูงมากที่สุดทางด้านบวก รูปสัญญาณเอเอ็มจะมีความถี่มากที่สุดและคลื่นสัญญาณเอเอ็มมีความสูงต่ำสุด เมื่อสัญญาณข้อมูลมีค่าลบสูงสุดจากรูปที่ 2.15 (ข) สัญญาณข้อมูลมีความสูงเพิ่มขึ้นจะทำให้รูปเอเอ็มมีค่าสูงสุดมากกว่าเดิมและค่าต่ำสุดต่ำกว่าเดิม ส่วนในรูปที่ 2.15 (ค) สัญญาณข้อมูลมีขนาดเล็กลงแต่ความถี่มากขึ้น จะทำให้รูปคลื่นเอเอ็ม มีค่าสูงสุดและต่ำสุดลดลง และอัตราการแกว่งขึ้นลงเพิ่มขึ้นเท่ากับความถี่ของสัญญาณข้อมูล

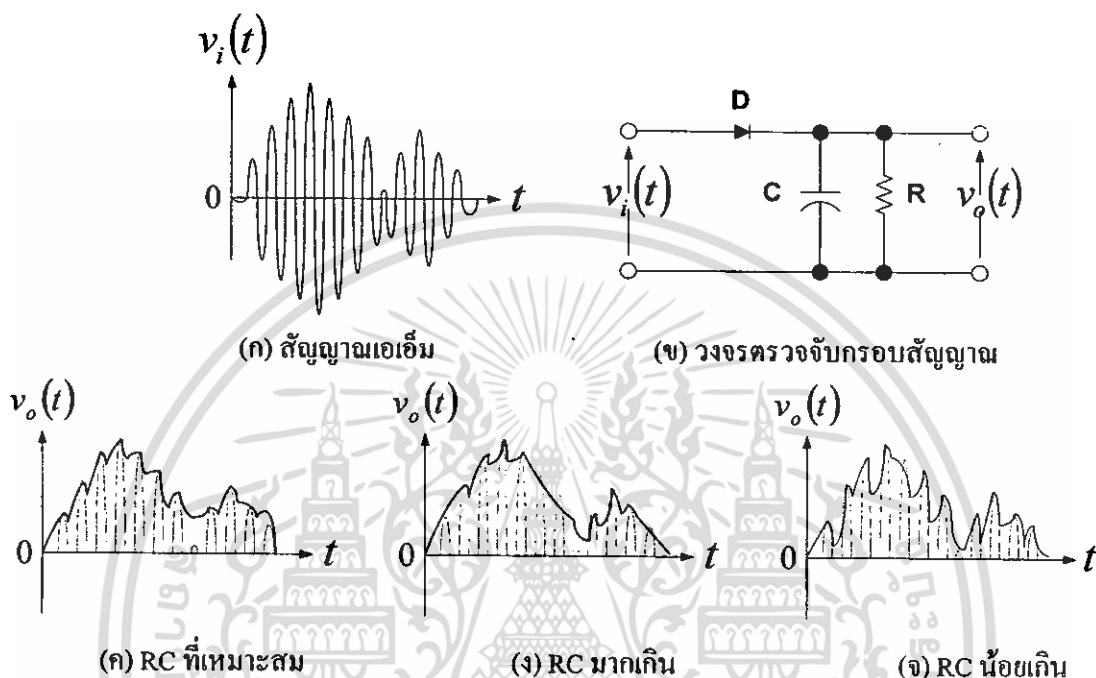


รูปที่ 2.15 รูปสัญญาณเอเอ็ม ที่เปลี่ยนแปลงตามสภาวะของสัญญาณข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 วงจรตรวจจับกรอบสัญญาณ

วงจรตรวจจับกรอบสัญญาณ คือวงจรที่ให้เอาต์พุตมีลักษณะเป็นไปตามกรอบหรือแนวทางเดินของยอดคลื่นของสัญญาณอินพุต วงจรตรวจจับกรอบสัญญาณแบบง่ายจะเป็นวงจรชนิดไม่เชิงเส้นดังแสดงในรูป



รูปที่ 2.16 ประกอบการอธิบายการทำงานของวงจรตรวจจับกรอบสัญญาณ

การทำงานของวงจรมี เมื่อมีสัญญาณ $\phi_{AM}(t)$ เข้ามาสู่วงจรในตอนแรกนั้น ขณะที่สัญญาณมีค่าเพิ่มขึ้นในทางบวกไดโอด D จะนำกระแส ทำให้เกิดการประจุตัวเก็บประจุ C ให้มีแรงดันเอาต์พุตเกิดขึ้นคร่อม C ตามค่าการเพิ่มของสัญญาณอินพุต จนกระทั่งถึงค่าสูงสุด (Peak) หรือยอดของสัญญาณ แต่ต่อจากนี้เมื่ออินพุต $\phi_{AM}(t)$ ลดลงก็จะทำให้ค่าแรงดันไฟฟ้าทางเอาโนดของ D ต่ำกว่าคาโทด ในสถานการณ์เช่นนี้ไดโอด D จะหยุดนำกระแสและกั้นไม่ให้ C คายประจุผ่านตัวมันได้ ดังนั้นการคายประจุของ C จึงมีทางเดียวคือ คายประจุผ่าน R ไปอย่างช้า ๆ จนกระทั่งถึงเวลาที่ $\phi_{AM}(t)$ จะมีค่ากลับเป็นบวก และมีค่าเพิ่มขึ้นมากกว่าแรงดันไฟฟ้าคร่อม C ไดโอด D จึงจะนำกระแสอีก และจากการประจุค่าของ C ทำให้เอาต์พุตเป็นไปตามค่าของสัญญาณ $\phi_{AM}(t)$ อีกไปจนถึงค่ายอดของคลื่นพาดิคมมา แล้วพฤติกรรมการทำงานจะเกิดขึ้นในลักษณะเดิมอีกกล่าวคือการคายประจุของ C ผ่าน R อีกครั้งหนึ่งพฤติกรรมเช่นนี้จะเกิดขึ้นแล้วซ้ำอีกเรื่อยไป ดังนั้นถ้าเราเลือกค่าคงตัวเชิงเวลา (Time constant) RC ที่ใช้ให้เหมาะสม ก็จะได้เอาต์พุตของวงจรเป็นไปตามเส้นที่บดรูปที่ 2.16 (ค) แต่ถ้าค่าคงตัวเชิงเวลาของวงจรตรวจจับกรอบสัญญาณมีค่ามากเมื่อเทียบกับคาบเวลาที่สั้นที่สุดของสัญญาณที่ประกอบอยู่ใน $m(t)$ แล้วก็จะทำให้เอาต์พุตของวงจรเกิดลดตามยอดคลื่นของ $\phi_{AM}(t)$ ไม่ทันจะทำให้เกิดการผิดเพี้ยนของสัญญาณเอาต์พุตดังรูปที่ 2.16 (ง) และในกรณีที่เกิดการใช้ RC มีค่าน้อยเมื่อเทียบกับคาบเวลาของ

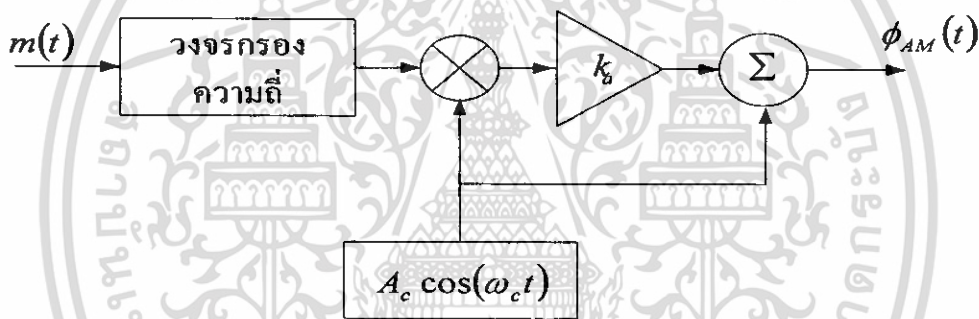
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลื่นพหุก็จะทำให้เกิดการกระเพื่อม (Ripple) บนสัญญาณเอาต์พุตมากดังรูปที่ 2.16 (จ) อย่างไรก็ตาม ค่าความถี่ของคลื่นพหุ ω_c ที่ใช้ในทางปฏิบัตินั้นจะมีค่าสูงกว่าความถี่ของสัญญาณที่เป็นส่วนประกอบของ $m(t)$ มาก หรือกล่าวอีกนัยหนึ่งได้ว่า $m(t)$ จะเปลี่ยนแปลงช้ามากเมื่อเทียบกับคลื่นพหุที่ใช้จึงทำให้การเลือกค่าเวลาคงตัวของวงจร RC ทำได้อย่างเหมาะสมง่าย กล่าวคือจะทำให้การกระเพื่อมบนเอาต์พุตของวงจรมีค่าน้อยมากได้จนไม่ต้องคำนึงถึงในทางปฏิบัติ เงื่อนไขสำหรับการเลือก ค่า RC ที่เหมาะสมสามารถสรุปได้ในรูป ของอสมการดังนี้

$$\frac{1}{f_c} \ll RC \ll \frac{1}{f_m} \quad (2-15)$$

โดยในที่นี้ f_c และ f_m คือค่าความถี่ของคลื่นพหุและค่าความถี่สูงสุดที่ประกอบอยู่ในสัญญาณข่าวสาร $m(t)$ ตามลำดับ

โดยอาศัยสมการที่ (2-13) เราสามารถสร้างสัญญาณเอเอ็มได้โดยระบบที่ง่ายดังรูป



รูปที่ 2.17 แผนผังระบบการสร้างสัญญาณเอเอ็มแบบง่าย

อย่างไรก็ตามเราสามารถที่จะสร้างสัญญาณเอเอ็ม โดยวิธีอื่น เช่น ใช้วงจรสวิตช์ หรือวงจรไม่เชิงเส้นได้เช่นเดียวกับการสร้างสัญญาณดีเอสบี เอสซี

2.3.6 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก (D/A)

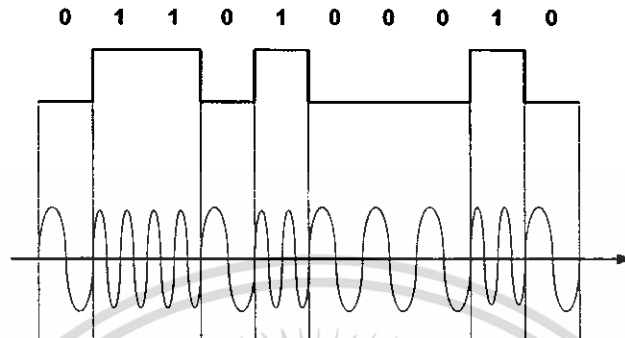
อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่แปลงสัญญาณข้อมูลดิจิทัลให้เป็นสัญญาณอนาลอกย่านความถี่เสียง เราเรียกว่า โมเด็ม (MODEM หรือ Modulator Demodulator)

เทคนิคการแปลงสัญญาณข้อมูลดิจิทัลให้เป็นสัญญาณอนาลอกนั้นมีอยู่ด้วยกัน 3 วิธีคือ

1. การมอดูเลตเชิงเลขทางแอมพลิจูด (Amplitude-Shift keying หรือ ASK)
2. การมอดูเลตเชิงเลขทางความถี่ (Frequency-Shift Keying หรือ FSK)
3. การมอดูเลตเชิงเลขทางเฟส (Phase-Shift Keying หรือ PSK)

2.3.6.1 การมอดูเลตเชิงเลขทางความถี่ (FSK)

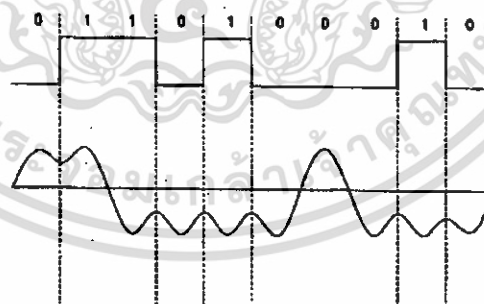
ในการมอดูเลตแบบ FSK ขนาดของคลื่นพาห้จะไม่เปลี่ยนแปลง ที่เปลี่ยนแปลงคือความถี่ของคลื่นพาห้ นั่นคือเมื่อบิตมีค่าเป็น “1” ความถี่ของคลื่นพาห้จะสูงกว่าปกติ และเมื่อบิตมีค่าเป็น “0” ความถี่ของคลื่นพาห้ก็จะต่ำกว่าปกติ



รูปที่ 2.18 การแปลงสัญญาณ D/A แบบ FSK

2.3.6.2 การมอดูเลตเชิงเลขทางเฟส (PSK)

หลักการของ PSK คือค่าของขนาดและความถี่ของคลื่นพาห้จะไม่มีการเปลี่ยนแปลง แต่ที่จะเปลี่ยนคือเฟสของสัญญาณ กล่าวคือเมื่อมีการเปลี่ยนแปลงสถานะของบิตจาก “1” ไปเป็น “0” หรือเปลี่ยนจาก “0” ไปเป็น “1” เฟสของคลื่นจะเปลี่ยน (Shift) ไป 180 องศาด้วย วิธี การ แบบ PSK จะมีสัญญาณรบกวนเกิดขึ้นน้อยที่สุด ได้สัญญาณที่มีคุณภาพดีที่สุด แต่วงจรการทำงานจะยุ่งยากกว่าและราคาสูงกว่า



รูปที่ 2.19 การแปลงสัญญาณ D/A แบบ PSK

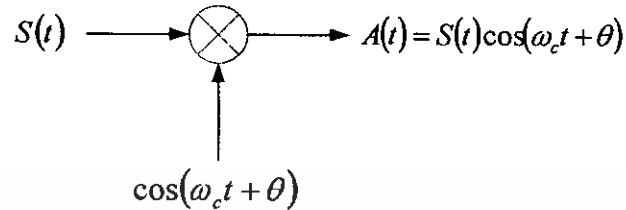
2.3.6.3 การมอดูเลตเชิงเลขทางแอมพลิจูด (ASK)

คลื่นที่ถูกมอดูเลตทางแอมพลิจูด ถูกทำให้แอมพลิจูดของคลื่นพาห้เปลี่ยนแปลงเป็นสัดส่วนกับสัญญาณข้อมูล ซึ่งทำได้โดยการคูณคลื่นพาห้ไซน์เข้ากับสัญญาณข้อมูล คลื่นที่ถูกมอดูเลตทางแอมพลิจูด A (t) สามารถกำหนดโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

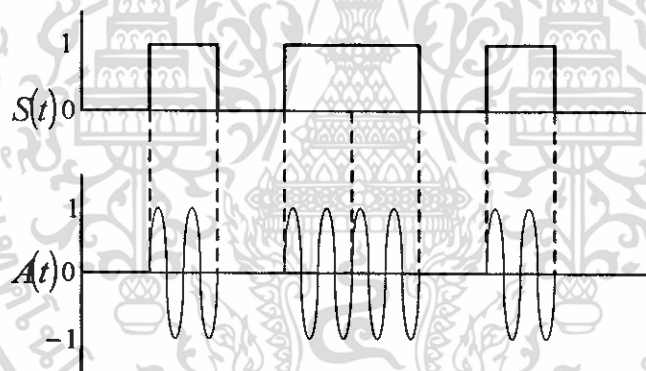
$$A(t) = S(t) \times \cos(\omega_c t + \theta)$$

$S(t)$ = สัญญาณข้อมูล
 $\cos(\omega_c t + \theta)$ = คลื่นพาห้



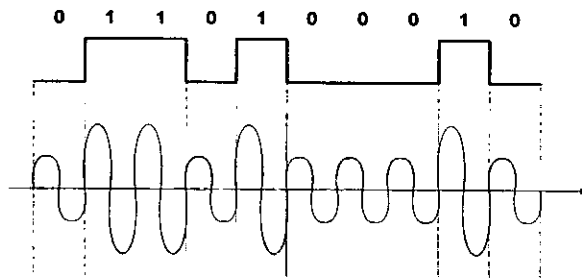
รูปที่ 2.20 รูปแบบของการมอดูเลตทางแอมพลิจูด

พิจารณาสัญญาณข้อมูล ซึ่งเป็นขุ่นีโพล่าพัลส์ เมื่อพัลส์เป็น 1 จะมีคลื่นพาห้ออกมา และเมื่อพัลส์เป็น 0 จะไม่มีคลื่นพาห้ออกมา การมอดูเลตทางแอมพลิจูดแบบนี้เรียกว่า ON-OFF ASK หรือ ON-OFF Keying (OOK)



รูปที่ 2.21 ON-OFF ASK

ความถี่ของคลื่นพาห้ (Carrier Wave) ซึ่งทำหน้าที่นำสัญญาณอนาลอกผ่านตัวกลางสื่อสารนั้นจะคงที่ ลักษณะของสัญญาณมอดูเลตเมื่อค่าของบิตของสัญญาณข้อมูลดิจิทัลมีค่าเป็น "1" ขนาดของคลื่นพาห้จะสูงขึ้นกว่าปกติ และเมื่อบิตมีค่าเป็น "0" ขนาดของคลื่นพาห้จะตกลงกว่าปกติ การมอดูเลต ASK มักไม่ค่อยได้รับความนิยม เพราะจะถูกรบกวนจากสัญญาณอื่นได้ง่าย



รูปที่ 2.22 การแปลงสัญญาณ D/A แบบ ASK

2.4 Delphi 7

Delphi 7 ก็คือซอฟต์แวร์ที่เรานำมาใช้ในการเขียนโปรแกรมเพื่อสร้างแอปพลิเคชัน หรือซอฟต์แวร์อีกที โดยมันจะประกอบไปด้วยเครื่องมือชนิดต่างๆที่ใช้ให้การเขียนโปรแกรมทำได้ง่าย สะดวก

Delphi 7 จัดเป็นเครื่องมือเขียนโปรแกรมชนิด Visual Programming เช่นเดียวกับ Visual Basic หรือ Visual C++ โดยมีข้อดีคือ สามารถเขียนโปรแกรมได้ง่าย และให้ผลงานออกมาอย่างรวดเร็ว ซึ่งจะแตกต่างจากเครื่องมือเขียนโปรแกรมรุ่นเดิมนั้นๆ เช่น Turbo Pascal หรือ Borland C ที่มีความยุ่งยากในการทำงานและการเรียนรู้ในการเขียนโปรแกรม ดังนั้นจึงจัดให้ Delphi 7 เป็นซอฟต์แวร์ประเภท RAD หรือ Rapid Application Development ซึ่งแปลว่าสามารถสร้างแอปพลิเคชันได้อย่างรวดเร็ว

Delphi 7 นั้นผ่านการพัฒนามาเกือบ 10 ปี ตั้งแต่เวอร์ชัน 1.0 ที่ทำงานบน Windows 3.1X โดยมีจุดเด่นหลายๆ ตั้งแต่สมัยนั้นคือ โปรแกรมที่ได้จากการเขียนโปรแกรมมีขนาดเล็ก ทำงานได้รวดเร็ว ซึ่งมักจะถูกนำไปเปรียบเทียบกับ Visual Basic 3.0 ในสมัยนั้น อีกประการหนึ่ง Delphi ใช้ภาษาออบเจกต์ ปาสคาล จึงเคยถูกเปรียบว่าเป็น Visual Pascal มาแล้ว

เวอร์ชันปัจจุบันของ Delphi นั้นได้รับการพัฒนาให้สามารถสร้างแอปพลิเคชันที่ทำงานบน Windows ได้ดีเหมือนเดิม โดยมีการปรับปรุงให้สามารถพัฒนาแอปพลิเคชันตามแนวความคิดของ .NET ซึ่งจะช่วยให้สามารถเขียนโปรแกรมครั้งเดียว แล้วนำไปใช้งานบนอุปกรณ์ต่างๆ ไม่ว่าจะเป็น PDA , โทรศัพท์มือถือ และ บนเว็บได้

2.4.1 ความสามารถของ Delphi 7

Delphi 7 นั้นมีความสามารถมากมาย ได้รับการต้อนรับเป็นอย่างดีจากนักพัฒนาแอปพลิเคชันทั่วโลก รวมทั้งเมืองไทยด้วย ซึ่งจะเห็นได้จากการนำ Delphi ไปประกอบการเรียนการสอน การฝึกอบรม ตลอดจนการนำไปสร้างเป็นซอฟต์แวร์เชิงพาณิชย์จำนวนมาก

2.4.1.1 สร้างแอปพลิเคชันสำหรับ Windows

Delphi ได้ชื่อว่าเป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันที่ทำงานบน Windows ที่ใช้งานกันแพร่หลายมาก สามารถสร้างงานได้อย่างรวดเร็ว นิยมนำไปสร้างแอปพลิเคชันทั้งเพื่อการศึกษา และแอปพลิเคชันที่ใช้ในระบบงานจริงๆ ในโลกธุรกิจ

Delphi สามารถสร้างแอปพลิเคชันบน Windows ได้หลายเวอร์ชัน ซึ่งแอปพลิเคชันที่สร้างจาก Delphi ได้รับความนิยมเป็นอย่างมากในด้านขนาดโปรแกรมที่เล็กกะทัดรัด และทำงานได้อย่างรวดเร็ว เมื่อเทียบกับแอปพลิเคชันที่สร้างจากเครื่องมือตัวอื่นๆทำให้ Delphi แต่ละเวอร์ชันได้รับรางวัลเกียรติยศจากสถาบันต่างๆทั่วโลกมาโดยตลอด

2.4.1.2 สร้างระบบงานด้านฐานข้อมูล

ถ้าจะถามว่า Delphi ถูกนำไปสร้างผลงานอะไรมากที่สุดสำหรับเมืองไทยแล้ว หนีไม่พ้นเรื่องของการพัฒนาระบบงานด้านฐานข้อมูล ซึ่งถูกนำไปใช้งานอย่างแพร่หลายตั้งแต่องค์กรขนาดเล็กไปจนถึงองค์กรระดับประเทศ โดย Delphi มีจุดเด่นในการสร้างแอปพลิเคชันที่ติดต่อกับฐานข้อมูลได้หลายรูปแบบ มีวิธีการและเครื่องมือในการสร้างระบบงาน ระบบบริหารงานบุคคล ระบบคลังสินค้า หรือแม้แต่ระบบของตัวในโรงพยาบาลชั้นนำ

แอปพลิเคชันที่เกี่ยวกับฐานข้อมูลที่สร้างจาก Delphi สามารถนำไปใช้งานกับระบบฐานข้อมูลชั้นนำแทบทุกชนิดทั่วโลก นับตั้งแต่ระบบฐานข้อมูลส่วนบุคคลทั้ง Access , Paradox , Foxpro ไปจนถึงระบบฐานข้อมูลขนาดใหญ่ทั้ง Oracle , Sybase , SQL Server รวมถึงระบบไฟล์ชนิดต่างๆได้ด้วย

2.4.1.3 สร้างแอปพลิเคชันรองรับ .NET Web Service

แนวโน้มของการพัฒนาแอปพลิเคชันในยุคหน้าที่เราต้องเตรียมตัวให้พร้อมก็คือ การพัฒนาแอปพลิเคชันด้วยเทคโนโลยี .NET ซึ่งจะช่วยให้แอปพลิเคชันชนิดต่างๆ ไม่ว่าจะทำงานบนพีซี โน้ตบุ๊ก PDA หรือแม้แต่โทรศัพท์มือถือ สามารถเชื่อมโยงข้อมูล และคุยกันได้

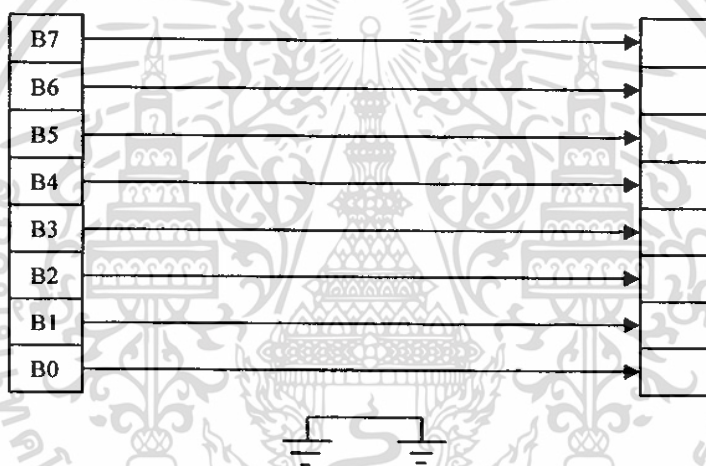
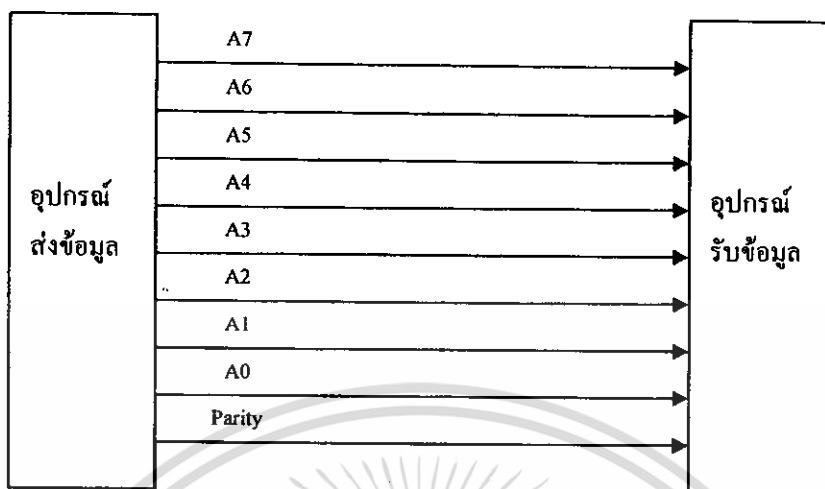
ใน Delphi 7 นั้นรองรับการพัฒนาแอปพลิเคชันที่เรียกว่า .NET Web Service แล้ว โดยสามารถใช้เครื่องมือชนิดต่างๆพัฒนาแอปพลิเคชันตามแนวความคิดของ .NET เป็นอย่างดี

2.5 การสื่อสารผ่านพอร์ตอนุกรม RS-232

2.5.1 การสื่อสารข้อมูล

2.5.1.1 การสื่อสารข้อมูลแบบขนาน (Parallel Communication)

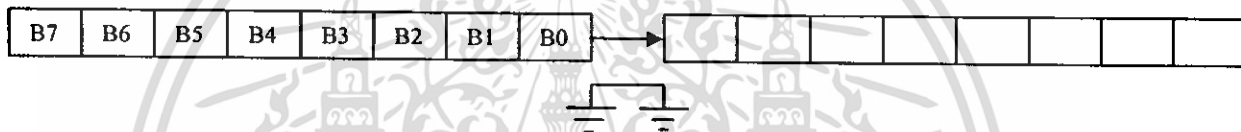
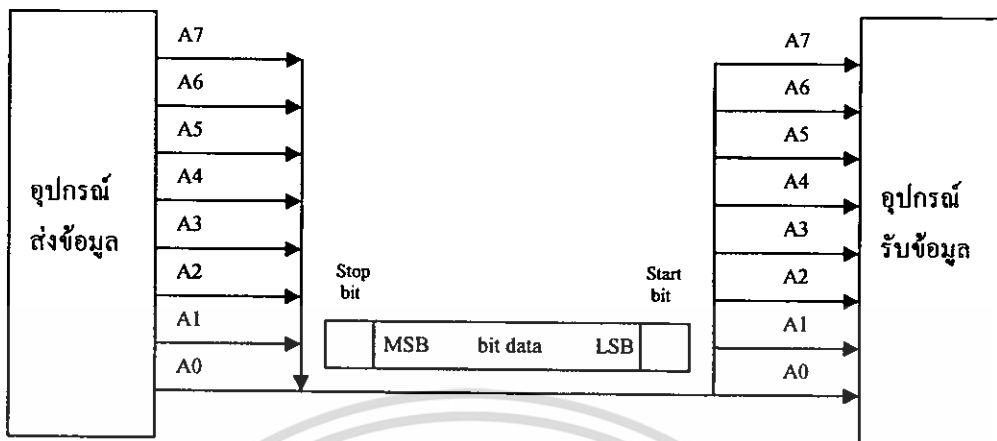
การสื่อสารข้อมูลแบบขนานคือการสื่อสารแบบที่ส่งข้อมูลพร้อมๆกัน n บิตผ่านสายสัญญาณ n เส้น สามารถแสดงรูปแบบการสื่อสารข้อมูลแบบขนานได้ดังรูป 2.23



รูปที่ 2.23 แสดงบล็อกโคอะแกรมรูปแบบการสื่อสารแบบขนาน

2.5.1.2 การสื่อสารข้อมูลแบบอนุกรม (Serial Communication)

การสื่อสารข้อมูลแบบอนุกรมคือการสื่อสารแบบที่ส่งข้อมูลที่ละบิต ผ่านสายสัญญาณเส้นเดียวกันจนครบจำนวนข้อมูลที่ต้องการ โดยเฟรมของการสื่อสารข้อมูลแบบอนุกรมประกอบด้วย สตาร์ทบิต(start bit), สตอปบิต (stop bit), ข้อมูล (data bit) สามารถแสดงรูปแบบการสื่อสารข้อมูลแบบอนุกรมได้ดังรูปที่ 2.24

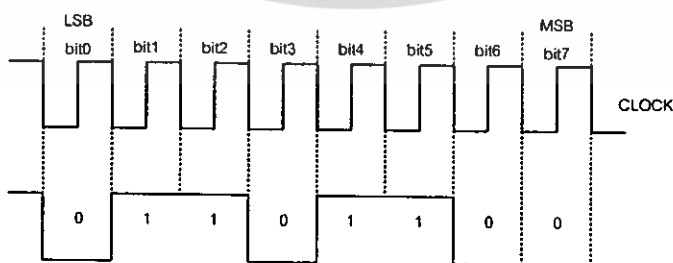


รูปที่ 2.24 แสดงบล็อกโคแอดแมมรูปแบบการสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนี้จะแบ่งออกได้เป็น 2 แบบ คือ การสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส

- การสื่อสารข้อมูลแบบซิงโครนัส

การสื่อสารข้อมูลแบบซิงโครนัสจะมีสัญญาณนาฬิกา ร่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสคือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา, ข้อมูล และกราวด์



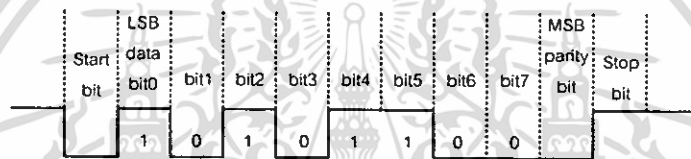
รูปที่ 2.25 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

- การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับส่งข้อมูลไปในสายสัญญาณ โดยไม่จำเป็นต้องส่งสัญญาณนาฬิกาพร้อมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายเทข้อมูล หรือ อัตราบอด (baudrate) ที่มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5 หรือ 2 บิต



รูปที่ 2.26 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

รูปที่ 2.26 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่ส่ง ขาดค่าจะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะว่าง (Idle stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขาดค่ามีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไปโดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในบิตที่จะส่งอาจจะมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูลบิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้ขาดค่ามีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและการส่งข้อมูลแบบอะซิงโครนัส เรียกว่า Universal Asynchronous Receiver / Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส คือ ค่าอัตราบอด ซึ่งก็คือค่าอัตราการเข้ารหัสที่ใช้ในการรับและส่งข้อมูล อัตราบอดมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าอัตราบอดได้สูงถึง 115200 บิตต่อวินาที เนื่องจากอัตราบอดคือจำนวนบิตของข้อมูลที่สามารถถ่ายเทได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิตและบิตปิดท้าย 1 บิตความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้อัตราบอดในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พริตตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

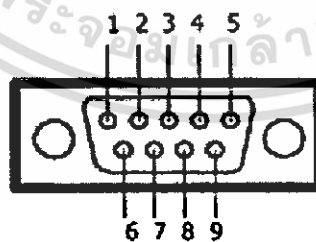
เป็น IC เป็นวงจรที่ทำหน้าที่เปลี่ยนแรงดันที่เข้ามาจาก Serial Port ไปเป็นแรงดันตามมาตรฐานของ RS-232 โดยเปลี่ยนระดับแรงดัน TTL เพื่อให้ใช้ร่วมกับไมโครคอนโทรลเลอร์

ลักษณะของการส่งข้อมูลแบบอนุกรมนี้ ข้อมูลจะส่งออกมาทีละบิตจากตัวส่งไปตัวรับข้อมูล ช่องสัญญาณในการส่งข้อมูลอาจใช้เพียง 1 หรือ 2 ช่องสัญญาณเท่านั้น ทำให้ค่าใช้จ่ายในการสื่อสารจะถูกกว่าแบบขนาน แต่อัตราการรับ-ส่งข้อมูลจะช้ากว่าแบบขนาน ในการส่งข้อมูลแบบอนุกรมข้อมูลที่ต้องการส่งจะอยู่ในลักษณะเป็นไบต์จะทยอยส่งทีละบิต และทางตัวรับจะต้องรับข้อมูลเข้ามาทีละบิตแล้วมารวมกันเป็นไบต์ซึ่งทางตัวรับต้องคอยตรวจสอบว่าบิตใดเป็นบิตเริ่มต้นหรือบิตสุดท้ายของข้อมูล การตรวจสอบนั้นจะขึ้นอยู่กับรูปแบบของรหัสของบิตข้อมูลที่ใช้ ซึ่งในการรับส่งข้อมูลแบบอนุกรมระหว่างไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้นจำเป็นจะต้องมีมาตรฐานในการรับส่งข้อมูล ซึ่งมาตรฐานที่นิยมมากที่สุดก็คือมาตรฐาน RS-232

มาตรฐาน RS-232

เพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันอย่างกว้างขวางที่สุดคือ RS-232 ซึ่งโดยปกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นแบบอนุกรมอยู่ในตัวแล้ว และจะทำหน้าที่รับส่งข้อมูลในแบบอนุกรม

ตามจุดประสงค์ของมาตรฐาน RS-232 นั้นเพื่อจะสามารถเชื่อมต่อกันระหว่างอุปกรณ์รับส่งปลายทาง (Data Terminal Equipment: DTE) เช่น พอร์ตของคอมพิวเตอร์หลักหรืออุปกรณ์ปลายทางกับอุปกรณ์สื่อสาร RS-232 เป็นข้อกำหนดของการอินเตอร์เฟซมาตรฐาน และสามารถและสามารถใช้เพื่อจุดประสงค์อื่นต่างกันไป เช่นการสื่อสารแบบซิงโครนัส (synchronous communication) และรูปแบบการสื่อสารที่ต้องการสัญญาณนาฬิกา และสัญญาณกำหนดจังหวะเพิ่มเติมขึ้นมา ในความเป็นจริงแล้วเราสามารถทำให้มีการสนทนากันระหว่าง DTE และ DCE โดยการใช้สายสัญญาณเพียง 3 เส้นเท่านั้น คือ ใช้สาย TD สาย RD และสายกราวด์เท่านั้น



รูปที่ 2.27 ลักษณะของคอนเน็คเตอร์แบบ DB-9

การเชื่อมต่อของพอร์ตสื่อสารสำหรับคอนเน็คเตอร์แบบ DB-9 สามารถแสดงได้ดังตารางที่ 2.5

ตารางที่ 2.5 การเชื่อมต่อของพอร์ตสื่อสารสำหรับคอนเน็คเตอร์แบบ DB-9

ตำแหน่งขาของ DB-9	สัญญาณ
1	Data Carrier Detect : DCD
2	Received Data : RxD
3	Transmitted Data : TxD
4	Data Terminal Ready : DTR
5	Signal Ground : GND
6	Data Set Ready : DSR
7	Request To Send : RTS
8	Clear To Send :CTS
9	Ring Indicator : RI

ขั้นตอนการติดต่อระหว่างอุปกรณ์ DTE และ DCE

1. เมื่อจ่ายกำลังงานให้กับ DTE และอุปกรณ์ก็จะส่งสัญญาณ DTR ออกมา
2. อุปกรณ์ DCE ถูกเปิดขึ้นและรับรู้สัญญาณ DTR ที่ส่งมาจากอุปกรณ์ DTE
3. อุปกรณ์ DCE ส่งสัญญาณ DSR ออกมา และโมเด็มก็กระทำกระบวนการ OFF HOOK
4. ถ้าสายสัญญาณอยู่ในสภาพดีและปลายทางอีกด้านหนึ่งก็พร้อมจะรับข้อมูลแล้ว จะมีการตรวจจับสัญญาณพาหะแล้วอุปกรณ์ DCE ส่งสัญญาณ DCD ออกมา
5. อุปกรณ์ DCE จะตอบสนองด้วยการ ส่งสัญญาณ CTS ออกมา
6. การติดต่อสื่อสารก็เริ่มขึ้น โปรแกรมควบคุมจะทำการส่งหรือรับข้อมูล

ส่วนลำดับขั้นในการตอบรับ

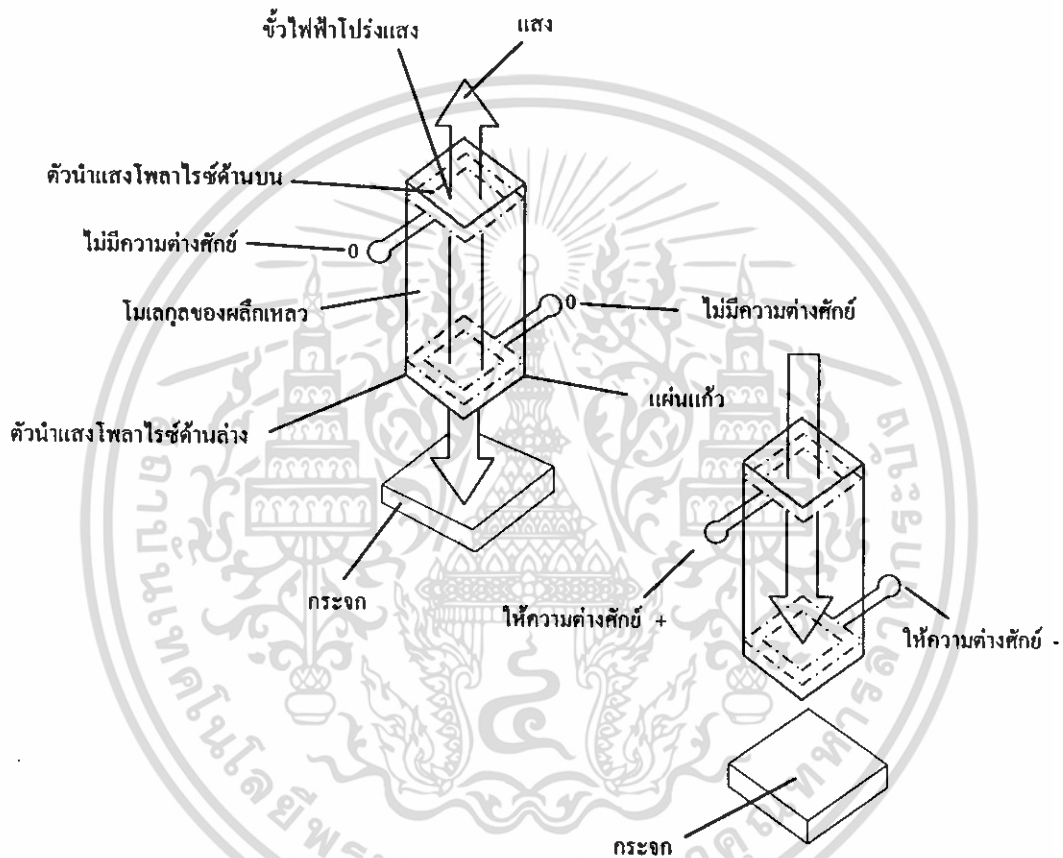
1. อุปกรณ์ DTE จะส่งสัญญาณ DTR ออกมา
2. อุปกรณ์ DCE จะอยู่ในโหมดตอบรับอัตโนมัติ (auto answer mode) โดยมีสัญญาณออกมา
3. สถานีปลายทางส่งสัญญาณเรียกอุปกรณ์ DCE และอุปกรณ์ DCE ส่งสัญญาณ RI ออกมา
4. อุปกรณ์ DTE รับรู้ถึงอุปกรณ์ RI ที่ส่งมาจากเครื่องปลายทาง และอุปกรณ์ DCE ก็เข้าสู่สถานะ OFF -HOOK
5. อุปกรณ์ DCE ทำการแลกเปลี่ยนข้อมูลกับอุปกรณ์ DCE ที่อีกปลายทางหนึ่ง และมีการส่งสัญญาณ DCD ออกมา
6. อุปกรณ์ DTE จะส่งสัญญาณ RTS ออกมาหรืออาจจะรอข้อมูลก็ได้ ขึ้นอยู่กับโปรแกรมควบคุม
7. อุปกรณ์ DCE จะตอบสนองด้วยการส่งสัญญาณ DTS กลับออกมา
8. การติดต่อสื่อสารก็จะเริ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบข้อดีและข้อเสียของการสื่อสารข้อมูลแบบอนุกรมและขนาน

- การสื่อสารข้อมูลแบบอนุกรมสามารถสื่อสารได้ระยะทางที่ไกลกว่า
- การสื่อสารข้อมูลแบบอนุกรมใช้สายสัญญาณที่ประหยัดกว่า
- การสื่อสารข้อมูลแบบขนานสามารถสื่อสารข้อมูลได้ทีละหลายๆ และมีความรวดเร็วมากกว่า

2.6 แอลซีดี



รูปที่ 2.28 อธิบายการทำงานของแอลซีดี

จอแอลซีดี (Liquid Crystal Display, LCD) หรือจอผลึกเหลวนั้น เป็นเทคโนโลยีที่เราคุ้นเคยกันมากที่สุด ตัวจอมีหลักการทำงานซึ่งอาศัยการเบี่ยงเบนของแสง ตามการควบคุมทิศทางของโพลาไรเซชันของวัตถุที่กั้นระหว่างแหล่งกำเนิดแสงและแผ่นเคลือบสารเรืองแสง เมื่อต้องการให้มีแสงเกิดขึ้น ก็ป้อนแรงดันเข้าไปยังแผ่นเพลตที่มีคุณสมบัติบิดงอได้เมื่อได้รับแรงดันไฟฟ้า มีผลให้แสงจากแหล่งกำเนิดสามารถผ่านทะลุกระทบกับสารเรืองแสง จนสามารถแสดงแสงสีตามต้องการได้ การทำงานดังกล่าวก็จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมือนกับมูลีที่เราใช้กันแสงหน้าต่างบ้านนั่นเอง เมื่อปรับมูลีให้เปิดกว้างความเข้มแสงในห้องก็จะมากขึ้นตามการ หมุนแผ่นกันให้เปิดมากขึ้นตามความต้องการ

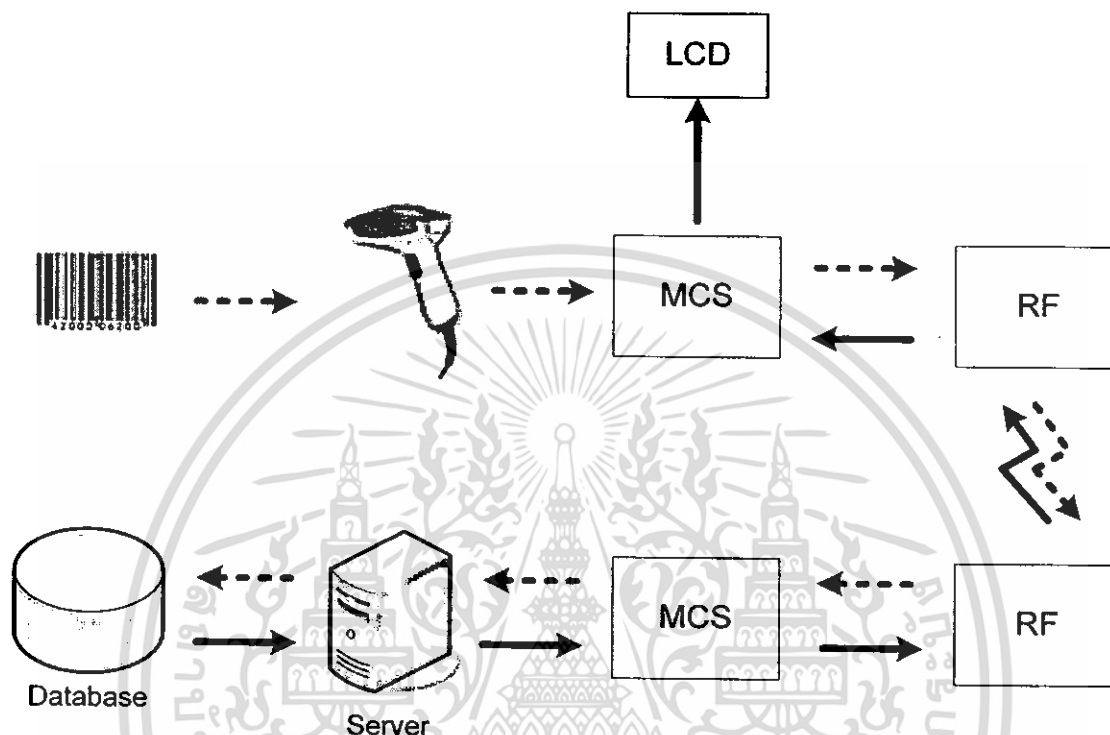
จอแอลซีดียังแบ่งออกเป็นสองประเภท คือ แอลซีดีแบบพาสซีฟ และ แอลซีดีแบบแอคทีฟ ในแบบพาสซีฟนั้น ภาพหนึ่งจุด (1 พิกเซล) เกิดจากการตัดกันของเส้นกริดแนวตั้งและแนวนอน เซลล์ในแอลซีดี จะทำตัวเสมือนกับตัวเก็บประจุ หลังจากประจุไฟแล้วจะค่อยๆ ปล่อยแรงดันออกมา และจะกลับไปสู่สภาวะเดิมอย่างช้าๆ ด้วยเหตุนี้เองจอแบบนี้จึงไม่สามารถเปลี่ยนสถานะอย่างรวดเร็วได้ ดังนั้นจึงไม่เหมาะกับการใช้งานกราฟิกที่มีการเคลื่อนไหวความเร็วสูง นอกจากนี้จอแบบพาสซีฟยังมีข้อเสียตรงที่มุมมอง ของภาพที่กว้างประมาณ 70-80 องศาเท่านั้น จึงอาจมองภาพไม่เห็นถ้ายืนอยู่ริมมากๆ เราจะพบจอแบบนี้ได้ในอุปกรณ์ เช่น นาฬิกาข้อมือแบบดิจิทัล หน้าปัดเครื่องคิดเลข หน้าจอเครื่องแฟกซ์ เป็นต้น ส่วนจอแอลซีดีแบบแอคทีฟ หรือที่มักมี วงเล็บ (TFT) ตามหลัง TFT คือ Thin Film Transistor เป็นการใช้ทรานซิสเตอร์ในการขับเซลล์ของ แอลซีดี ทำให้สามารถตอบสนองต่อการใช้งานที่ต้องการความเร็วสูง และมุมมองที่กว้างถึง 180 องศา แอลซีดีแบบนี้มีใช้ในจอคอมพิวเตอร์ จอทีวีที่ติดตั้งภายในรถยนต์ และจอทีวีขนาดใหญ่ รวมถึงจอบนหน้าโทรศัพท์มือถือบางรุ่น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การคำนวณและการสร้าง

3.1 บล็อกไดอะแกรม



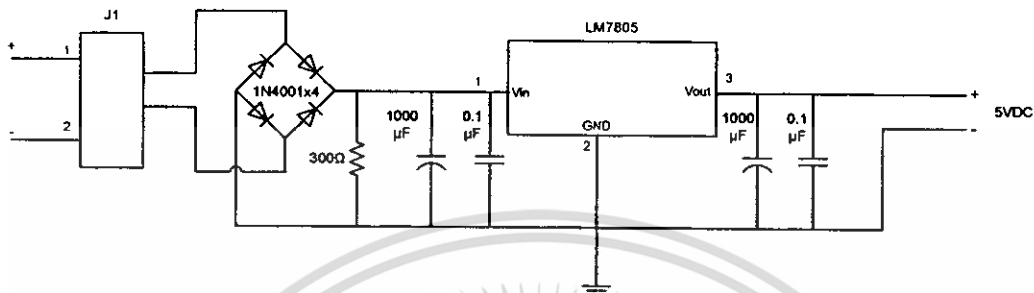
รูปที่ 3.1 บล็อกไดอะแกรมของปริญญาบัตร

ในปริญญาบัตรนี้จะใช้เครื่องอ่านบาร์โค้ดแบบซีซีดีความละเอียดสูง (High Resolution CCD) ในการอ่านค่าบาร์โค้ดของสินค้าที่ต้องการจะตรวจสอบข้อมูล แล้วทำการส่งข้อมูลค่าบาร์โค้ดนั้นผ่านพอร์ตอนุกรมไปยังวงจรซึ่งใช้ไมโครคอนโทรลเลอร์ตระกูล เอ็มซีเอส-51 เบอร์ AT89C2051 ซึ่งเป็นไอซี 20 ขา ทำหน้าที่ควบคุมการทำงานของวงจร เพื่อส่งค่าบาร์โค้ดไปที่วงจรส่วน RF module แบบ transceiver ที่ทำหน้าที่เป็นตัวส่งสัญญาณค่าบาร์โค้ดออกไปในรูปคลื่นวิทยุ ไปยัง RF module แบบ transceiver อีกตัวหนึ่งที่อยู่ทางด้านเครื่องเซิร์ฟเวอร์ ที่จะทำหน้าที่เป็นตัวรับสัญญาณเข้ามาเพื่อทำการส่งต่อค่าบาร์โค้ดที่ได้รับเข้ามาไปยังวงจรที่จะทำการส่งค่าบาร์โค้ดเข้าพอร์ตอนุกรมของเครื่องเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ได้รับค่าบาร์โค้ดแล้วจะทำการเชื่อมต่อไปยังฐานข้อมูลที่มีรายละเอียดต่างๆของสินค้า แล้วทำการเรียกค้น (Query) ข้อมูลคือ ชื่อสินค้า, ราคาสินค้า และจำนวนสินค้าคงเหลือจากในฐานข้อมูลมาแสดงที่เครื่องเซิร์ฟเวอร์ ในขณะเดียวกันก็จะทำการส่งค่าที่ได้จากฐานข้อมูลกลับไปยังฝั่งผู้ใช้เพื่อแสดงผลออกทางหน้าจอแอลซีดี (LCD) โดยข้อมูลทั้งหมดจะถูกส่งออกไปผ่านทางพอร์ตอนุกรมของเซิร์ฟเวอร์ ผ่านวงจรเพื่อให้ไมโครคอนโทรลเลอร์ส่งค่าเหล่านี้ไปยังวงจรส่วน RF module แบบ transceiver ที่ทำหน้าที่เป็นตัวส่งสัญญาณออกไปในรูปคลื่นวิทยุ ไปยัง RF module แบบ transceiver อีกตัวที่อยู่ทางด้านผู้ใช้ ซึ่ง

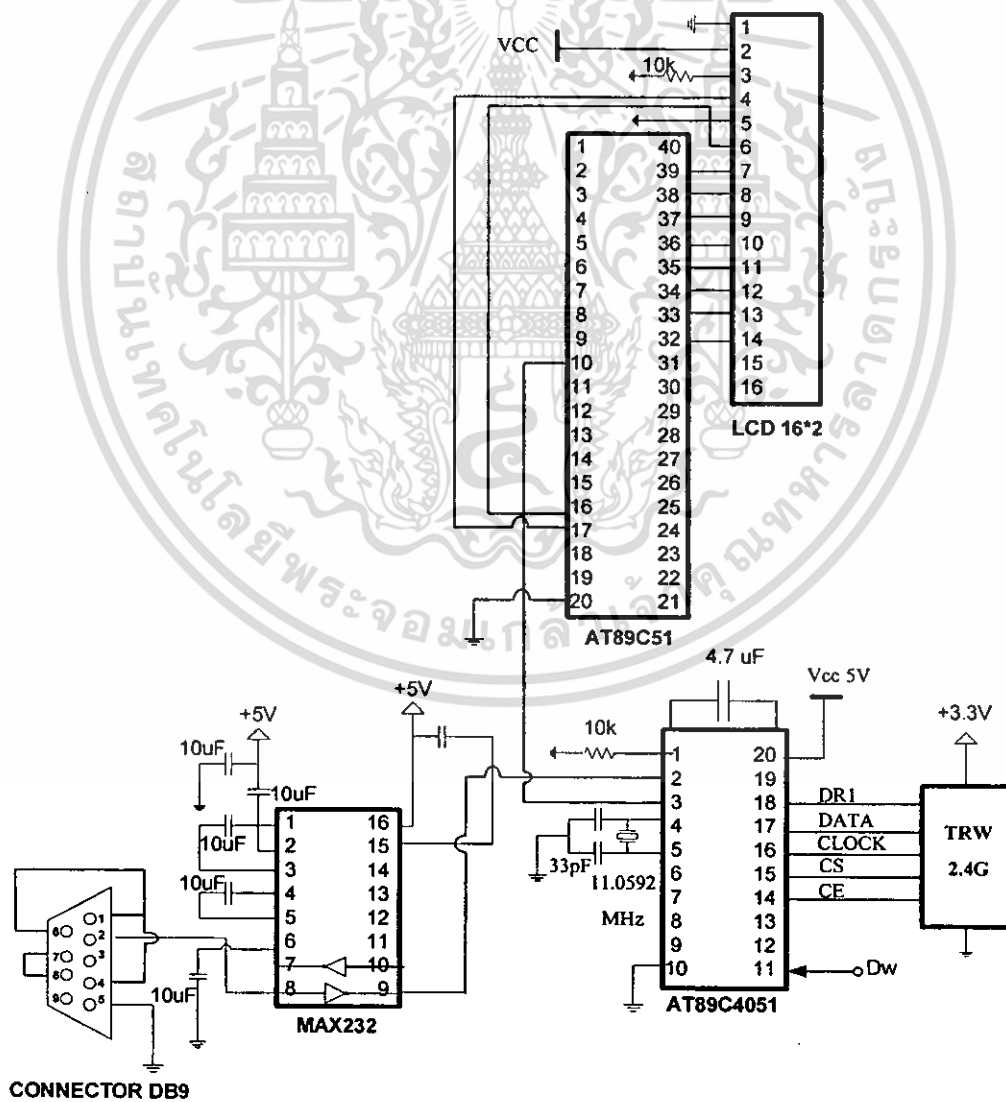
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำหน้าที่ในการรับสัญญาณแล้วส่งผ่านไปยังไมโครคอนโทรลเลอร์ที่ใช้การประมวลผลเพื่อแสดงรายละเอียดข้อมูลที่ได้รับมาจากฐานข้อมูลออกทางหน้าจอแอลซีดีแบบ 16 x 2

3.2 ไมโครคอนโทรลเลอร์

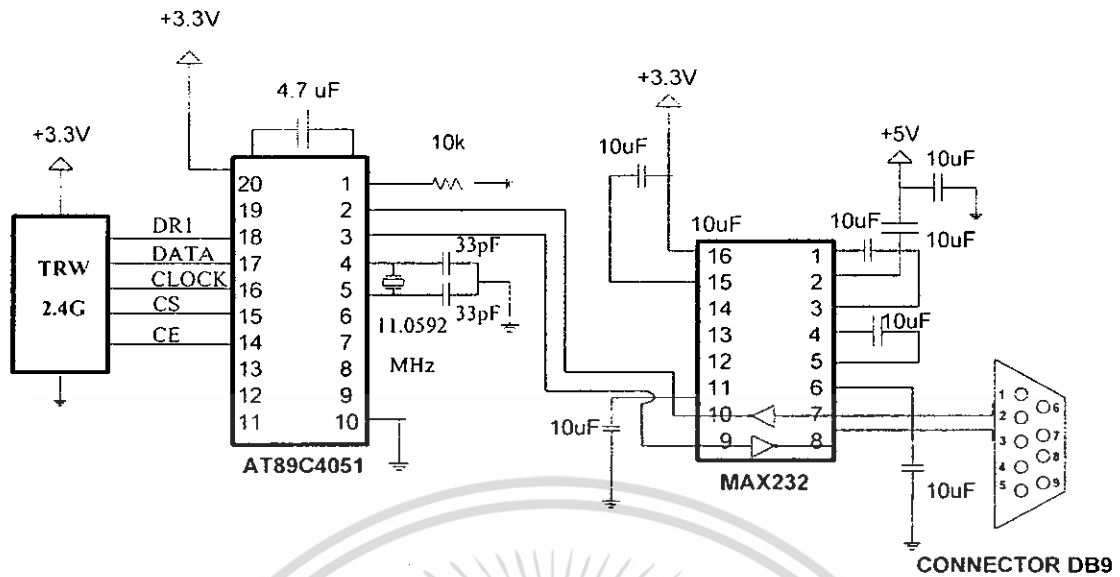


รูปที่ 3.2 แสดงวงจรภาคจ่ายไฟ (Power Supply)



รูปที่ 3.3 แสดงวงจรฝั่งบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงวงจรฝั่งเซิร์ฟเวอร์

3.3 การทำงานของ RF module

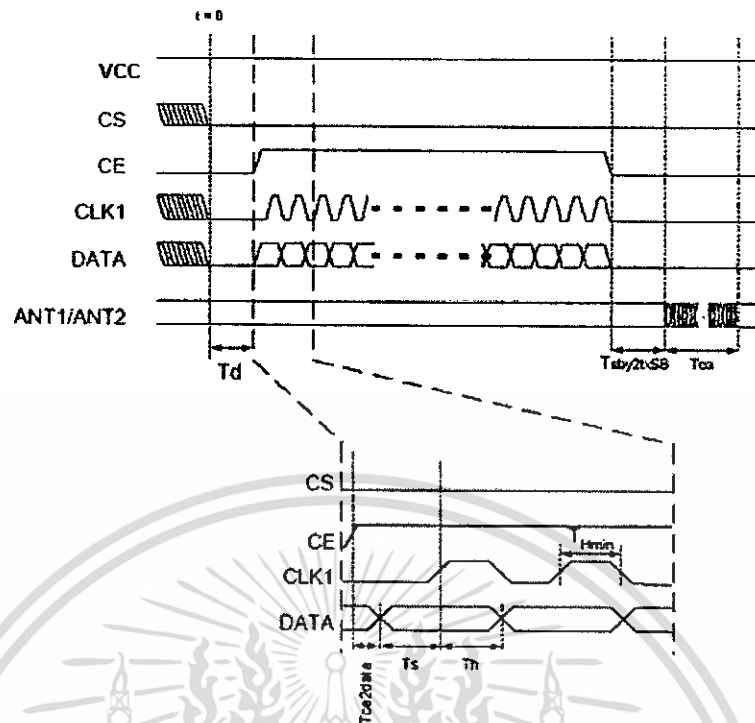
ในที่นี้เราเลือกใช้ TRW-2.4G เป็น FSK module ซึ่งมีข้อดีดังนี้คือ สามารถกำหนดให้เป็นตัวรับหรือส่งได้โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน และยังสามารถทำการเข้ารหัสและตรวจสอบความผิดพลาดได้ภายในตัวของมันเองเมื่อทำงานภายใต้การทำงานแบบ ช็อกเบิร์สต์ (ShockBurst mode)

3.3.1 การทำงานเมื่อเป็นตัวส่ง

ขาที่ต่อกับไมโครคอนโทรลเลอร์ : CE, CLK1, DATA โดยมีขั้นตอนการทำงานดังนี้

- เมื่อไมโครคอนโทรลเลอร์ มีข้อมูลที่จะส่ง จะกำหนดให้ CE เป็น high เป็นการกระตุ้นให้ TRW-2.4G ทำการประมวลผลข้อมูล
- แอดเดรสของตัวรับ (RX address) และข้อมูลใน เพย์โหลด (payload) จะจับเวลาเข้าสู่ช่วงเวลาระบบย่อยของ TRW-2.4G
- ไมโครคอนโทรลเลอร์กำหนดให้ CE เป็น low จะเป็นการกระตุ้นให้ทำการส่งแบบ ShockBurst
- TRW-2.4G ShockBurst:
 - RF front end มีกำลังงานสูงขึ้น
 - RF package ถูกทำให้สมบูรณ์ (เพิ่ม preamble, จำนวน CRC)
 - ข้อมูลถูกส่งด้วยความเร็วสูง (250 kbps หรือ 1 Mbps แล้วแต่การกำหนด)
 - TRW-2.4G กลับสู่สถานะเตรียมพร้อม เมื่อสิ้นสุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

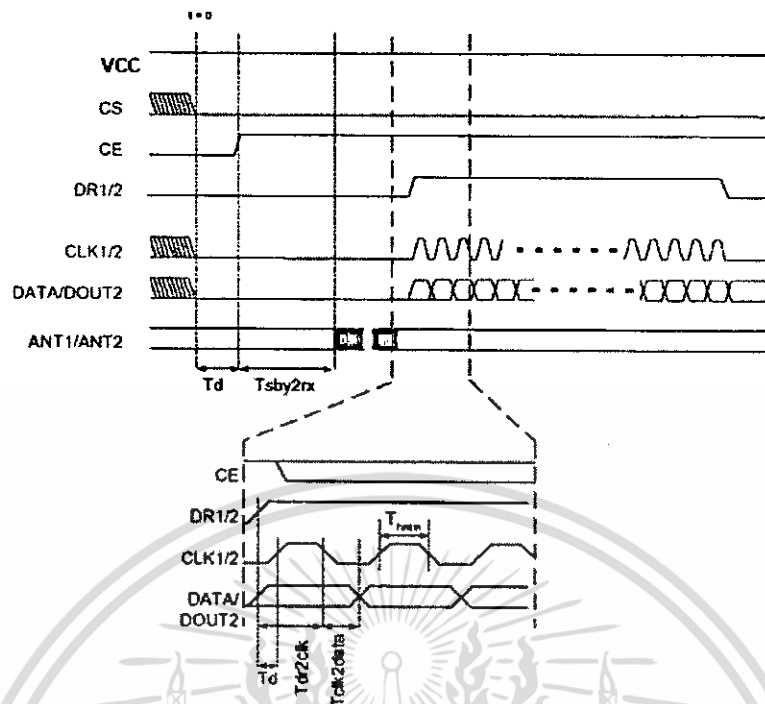


รูปที่ 3.5 แผนผังเวลาของการส่งแบบ ShockBurst

3.3.2 การทำงานเมื่อเป็นตัวรับ

ขาที่ต่อกับไมโครคอนโทรลเลอร์ : CE,DRI,CLK1,DATA โดยมีขั้นตอนการทำงานดังนี้

- แอคเคอเรสที่ถูกต้องและขนาดของแพย์โพลด ของสัญญาณ RF packages ที่เข้ามาจะถูกกำหนดให้กับ TRW-2.4G
- กระตุ้นให้เป็นการทำงานของตัวรับโดยกำหนดให้ CE เป็น low
- หลังจาก 200 ไมโครวินาที TRW-2.4G จะคอยตรวจสอบการสื่อสารที่เข้ามา
- เมื่อรับ package ที่ถูกต้องแล้ว TRW-2.4G จะทำการลบ preamble,address,CRC ออกไป
- จากนั้น TRW-2.4G จะแจ้งให้ไมโครคอนโทรลเลอร์รู้โดยการกำหนดให้ DRI มีสถานะเป็น high
- ไมโครคอนโทรลเลอร์จะ กำหนดให้ CE เป็น low เพื่อป้องกัน RF front end (โหมดที่กระแสดำ)
- ไมโครคอนโทรลเลอร์จะส่งสัญญาณนาฬิกาที่เหมาะสมกับข้อมูลในแพย์โพลด
- เมื่อข้อมูลทั้งหมดถูกรับเสร็จสิ้นแล้ว TRW-2.4G จะกำหนดให้ค่า DRI เป็น low อีกครั้ง และพร้อมที่จะรับข้อมูลชุดต่อไปที่กำลังจะเข้ามา โดยที่ CE จะรักษาสถานะเป็น high อยู่ระหว่างทำการรับข้อมูล ถ้า CE ถูกกำหนดให้มีสถานะเป็น low จะเป็นการเริ่มรับข้อมูลใหม่



รูปที่ 3.6 แผนผังเวลาของการรับแบบ ShockBurst

3.4 ฐานข้อมูล

ฐานข้อมูลจะรวบรวมข้อมูลที่สัมพันธ์กัน โดยข้อมูลเหล่านี้จะถูกเก็บในรูปแบบตารางซึ่งจะประกอบด้วยฟิลด์และ เร็คคอร์ด โดยฟิลด์ก็คือหัวข้อของข้อมูลที่เป็นตัวจัดกลุ่มในตาราง ในที่นี่มีอยู่ 4 ชนิด ถ้าเราใส่ข้อมูลจนครบทุกฟิลด์ในตารางแล้ว เราก็จะได้เร็คคอร์ดขึ้นมา อาจมองได้ว่าฟิลด์ และ เร็คคอร์ดก็คือ คอลัมน์ และ แถวของตาราง สำหรับปริมาณขั้นนี้จะทำการเก็บตัวอย่างชนิดสินค้าทั้งสิ้น 50 ชนิด โดยมีชนิดข้อมูลดังตารางที่ 3.1

ตารางที่ 3.1 ชนิดข้อมูลทั้งหมดในฐานข้อมูลโดยมีโค้ด (CODE) เป็นคีย์ของตาราง

CODE	PRODUCT	QUANTITY	PRICE
------	---------	----------	-------

Database Desktop - [Table : :PBDLMOS:barcode.D0]

File Edit View Table Record Tools Window Help

barcode	CODE	PRODUCT	QUANTITY	PRICE
1	885195290161	CRYSTAL	36	\$7.00
2	8850002651698	SHOKUBUTSU	6	\$55.00
3	8850007620174	JOHNSON POWDER	13	\$22.00
4	8850039600069	TISSUE POPUP	4	\$40.00
5	8850051006207	ODOLY FRESH	67	\$5.00
6	8850124003829	NESTLE BIS	16	\$14.00
7	8850124003850	NESTLE PURE LIFE	51	\$7.00
8	8850124003881	NESTLE SL	50	\$50.00
9	8850157500371	STICK BISCUIT	9	\$13.00
10	8850188250109	FOREMOST UHT	65	\$10.00
11	8850262520045	ENTREES	2	\$20.00
12	8850277002618	BENSONS STRAWBERRY	16	\$10.00
13	8850277002663	BENSONS KMM	23	\$10.00
14	8850309200136	FUN'O	33	\$5.00
15	8850309201293	DEWBERRY BLUBERRY	26	\$5.00
16	8850309201736	DEWBERRY STRAWBERRY	18	\$5.00
17	8850346114600	OVALTINE	12	\$17.00
18	8850360025838	DETTOL SHOWER GEL	19	\$107.00
19	8850718501519	CITOS CHEESE	47	\$5.00
20	8850718802517	FRITO LAYS	77	\$5.00
21	8850718802578	TAWAN SPICY	66	\$10.00
22	8850765014987	PUTHAI	41	\$10.00
23	8850943518382	BIGGA TWOTONE	33	\$5.00
24	8850987101021	MAMA TOMYUMKUNG	150	\$5.00
25	8850999322001	SINGHA WATER	21	\$9.00
26	8851004101116	PARTY	6	\$10.00
27	8851016000162	CALBEE	23	\$20.00
28	8851019030524	COLLON CHOCOLATE	17	\$15.00
29	8851234310029	GREENNUT	17	\$10.00
30	885172001021	POTAE	68	\$5.00
31	885172002028	PAPRIKA	35	\$5.00
32	885172003025	CORNAE	50	\$10.00
33	8851907018569	NOTEBOOK	4	\$25.00
34	8851959132012	COCA COLA CAN	68	\$15.00

Record 1 of 50

Database Desktop - [Table : :PBDLMOS:barcode.D0]

File Edit View Table Record Tools Window Help

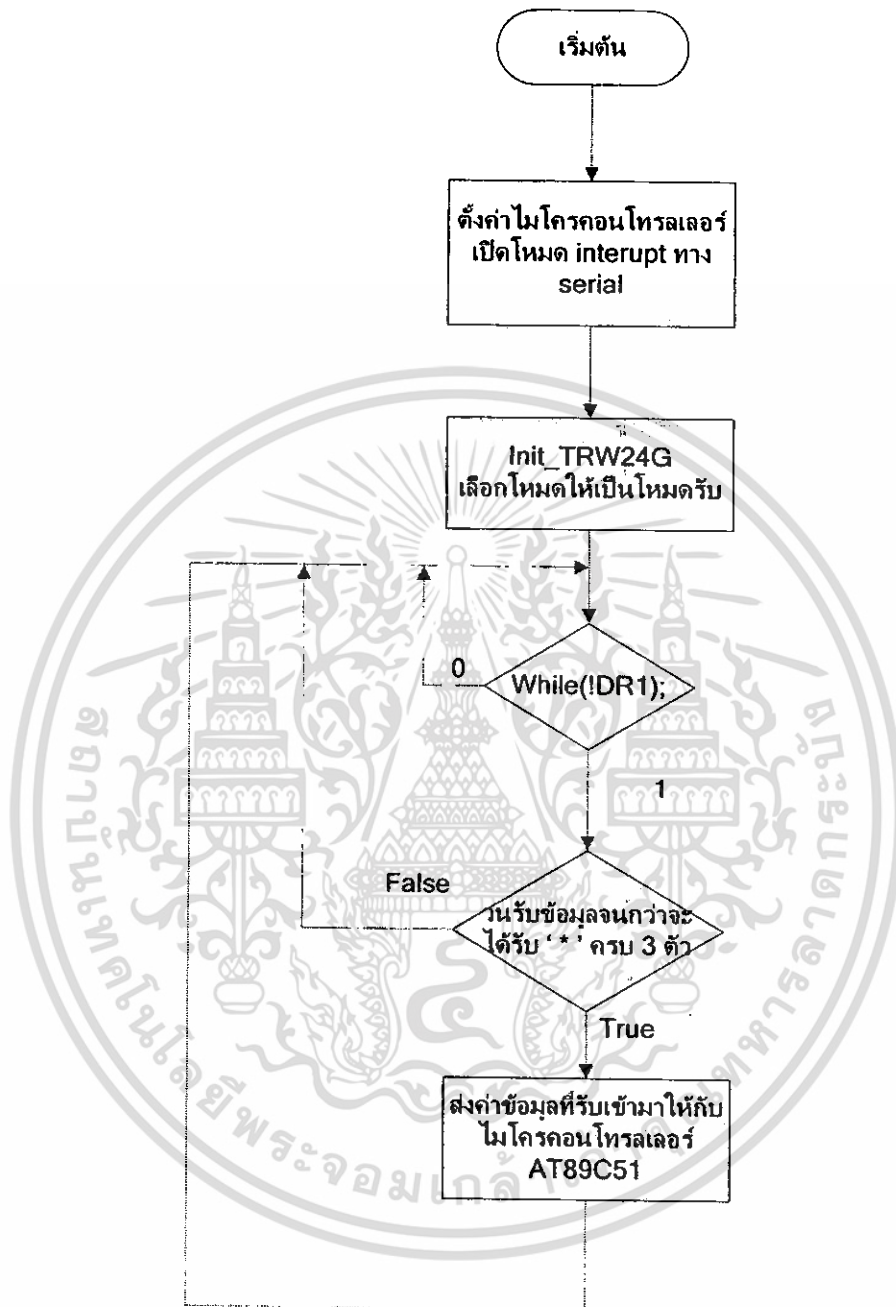
barcode	CODE	PRODUCT	QUANTITY	PRICE
17	8850346114600	OVALTINE	12	\$17.00
18	8850360025838	DETTOL SHOWER GEL	19	\$107.00
19	8850718501519	CITOS CHEESE	47	\$5.00
20	8850718802517	FRITO LAYS	77	\$5.00
21	8850718802578	TAWAN SPICY	66	\$10.00
22	8850765014987	PUTHAI	41	\$10.00
23	8850943518382	BIGGA TWOTONE	33	\$5.00
24	8850987101021	MAMA TOMYUMKUNG	150	\$5.00
25	8850999322001	SINGHA WATER	21	\$9.00
26	8851004101116	PARTY	6	\$10.00
27	8851016000162	CALBEE	23	\$20.00
28	8851019030524	COLLON CHOCOLATE	17	\$15.00
29	8851234310029	GREENNUT	17	\$10.00
30	885172001021	POTAE	68	\$5.00
31	885172002028	PAPRIKA	35	\$5.00
32	885172003025	CORNAE	50	\$10.00
33	8851907018569	NOTEBOOK	4	\$25.00
34	8851959132012	COCA COLA CAN	68	\$15.00
35	8852098703750	SHINMAI	1	\$18.00
36	8852854000758	SIAM MINERAL WATER	102	\$13.00
37	8854641001429	THIP SIAM WATER	72	\$7.00
38	8854690002479	OISHI WHITEMALT	49	\$20.00
39	8854690005043	OISHI LEMONHONEY	57	\$20.00
40	8854690005050	OISHI ORIGINAL	32	\$20.00
41	8854690005289	OISHI JAP RICE	48	\$20.00
42	8854690005426	AMINO OK MUSCAT	33	\$20.00
43	8854690005517	AMINO OK GRAPEFRUIT	52	\$20.00
44	8854690005968	OISHI NINJA	65	\$20.00
45	8856976000016	DOUBLE A	9	\$110.00
46	88666760114935	ORAL-B CONTURA	64	\$10.00
47	8927760211036	RITZ CRACKER	40	\$53.00
48	8850124041517	BEAR BRAND	78	\$13.00
49	8556001032990	KITKAT PACKS	11	\$50.00
50	8850718801436	RAY ORIGINAL	25	\$20.00

Record 50 of 50

รูปที่ 3.7 แสดงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

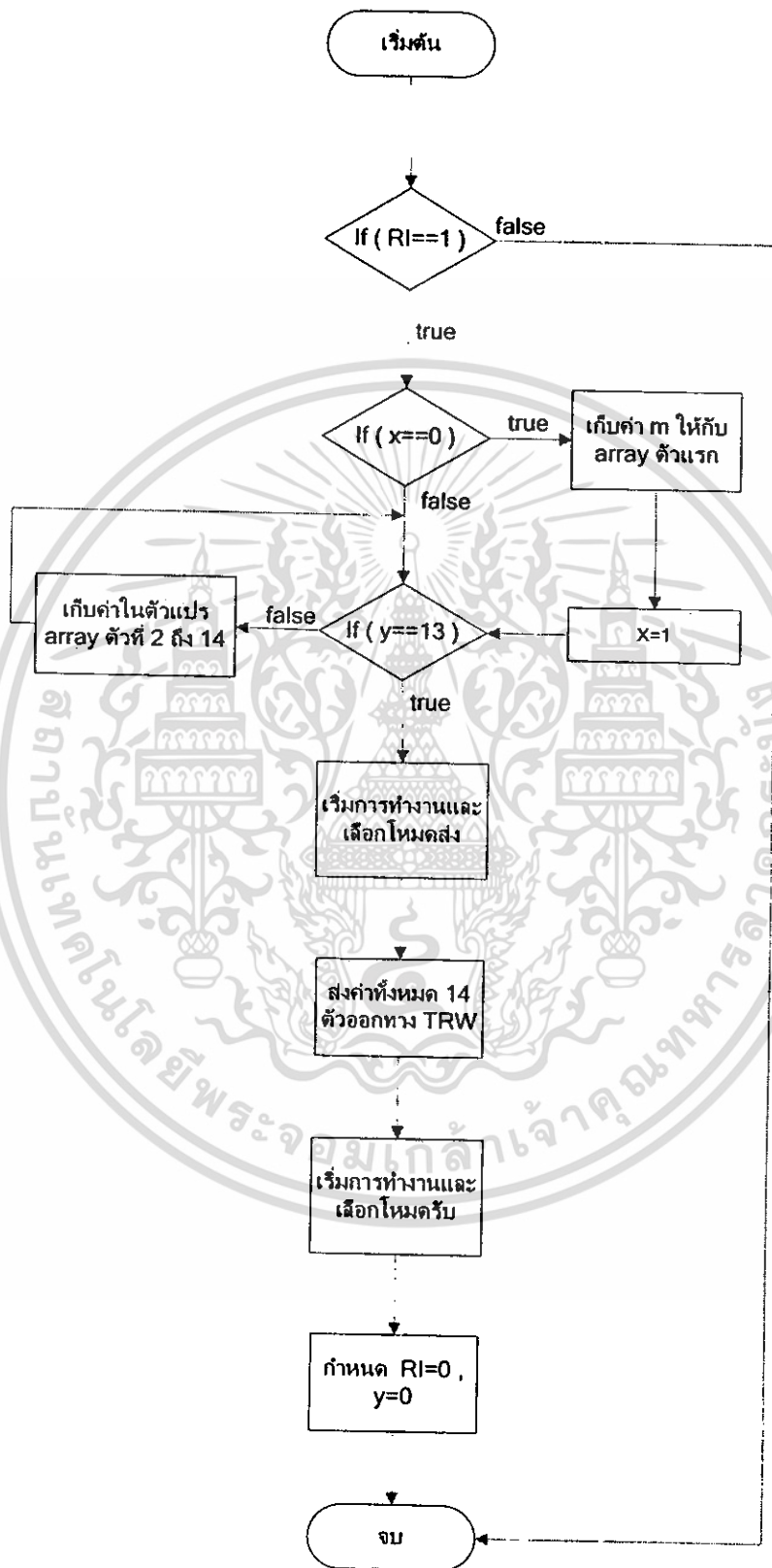
3.5 การออกแบบผังงานของไมโครคอนโทรลเลอร์ฟังบาร์โค้ด



รูปที่ 3.8 ผังงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์ฟังบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

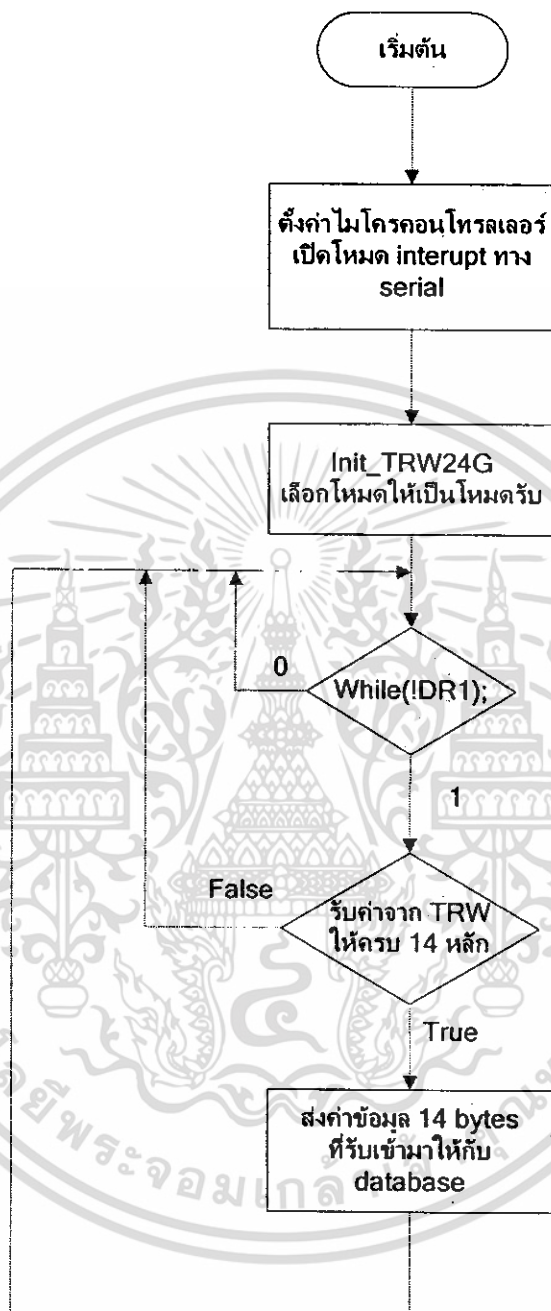
3.6 การออกแบบผังงานของโปรแกรมอินเทอร์เน็ตฟิ่งบาร์โค้ด



รูปที่ 3.9 ผังงานของโปรแกรมอินเทอร์เน็ตฟิ่งบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ยูสเซอร์นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

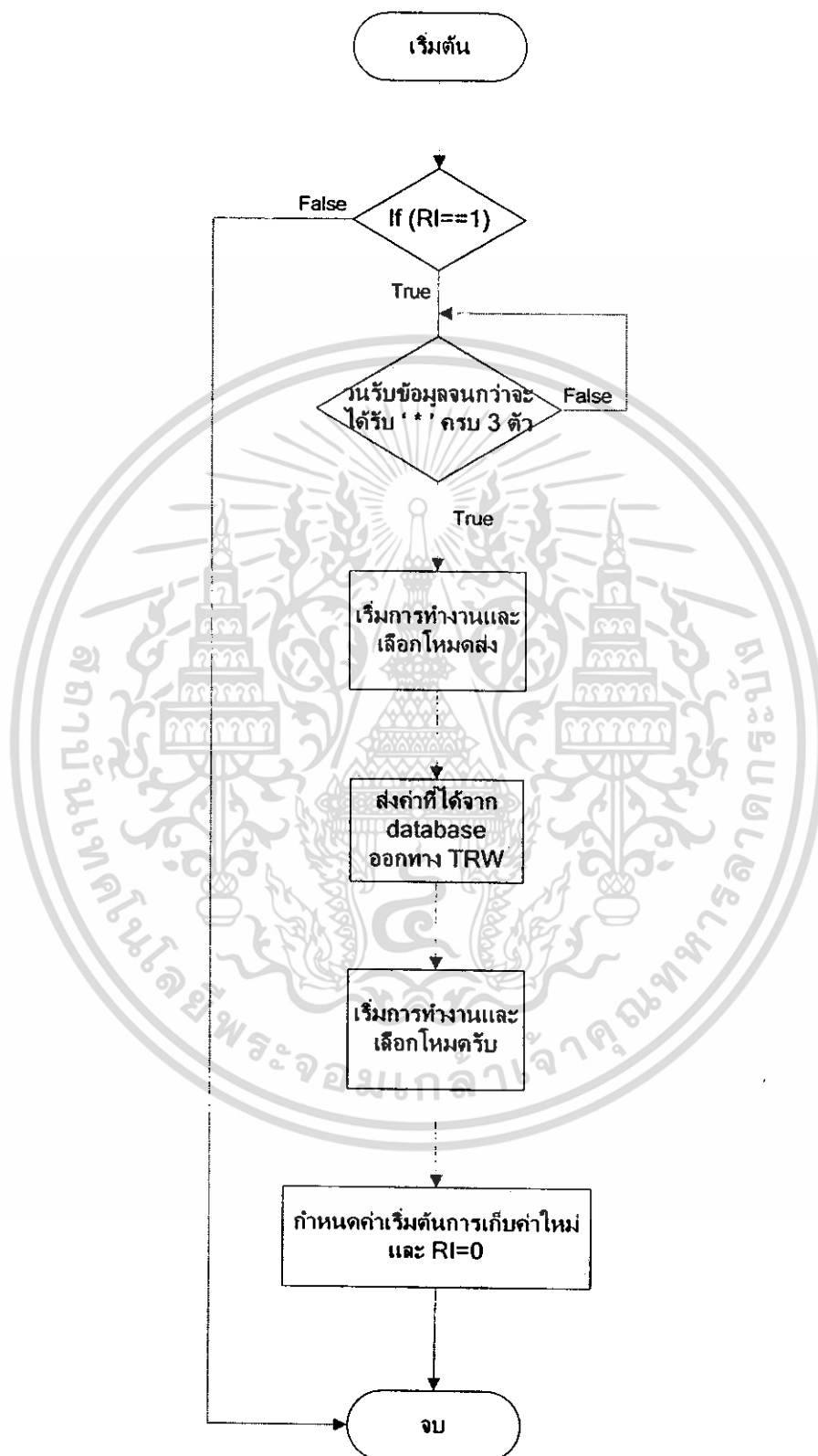
3.7 การออกแบบผังงานของไมโครคอนโทรลเลอร์ฝังเซิร์ฟเวอร์



รูปที่ 3.10 ผังงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์ฝังบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

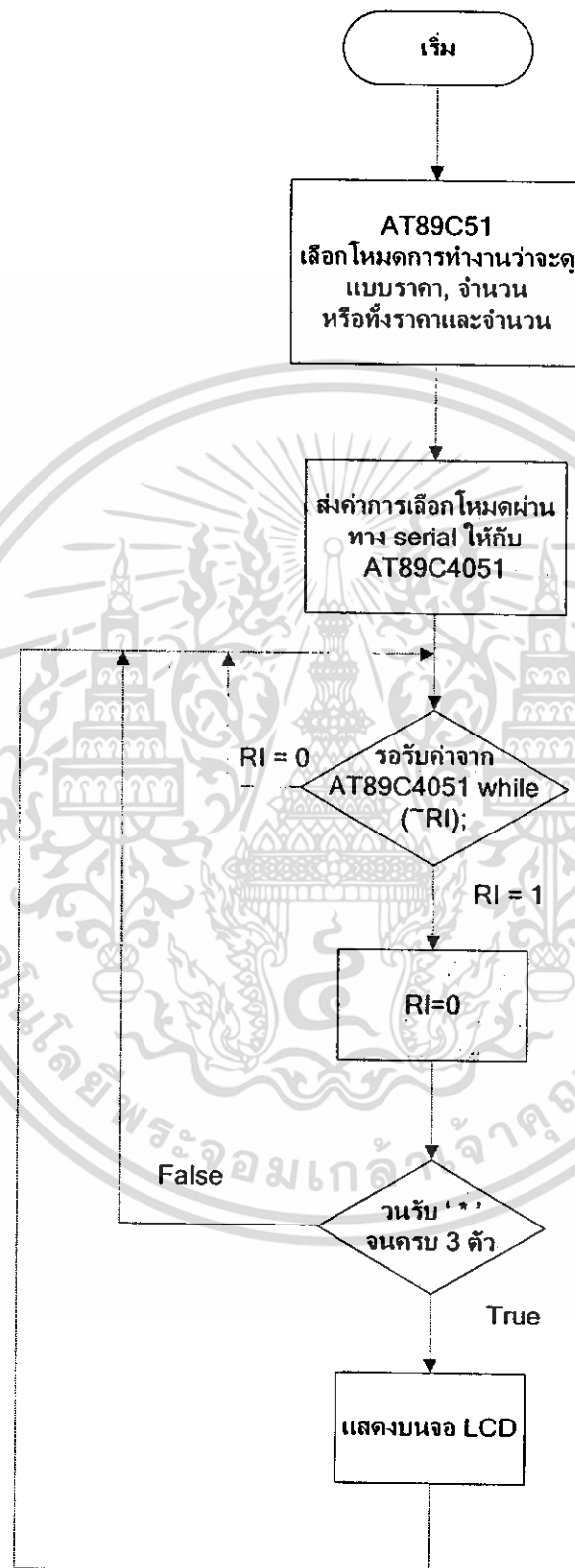
3.8 การออกแบบผังงานของโปรแกรมอินเทอร์เน็ตर्फิงเจอร์ฟเวอร์



รูปที่ 3.11 ผังงานของโปรแกรมอินเทอร์เน็ตर्फิงเจอร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

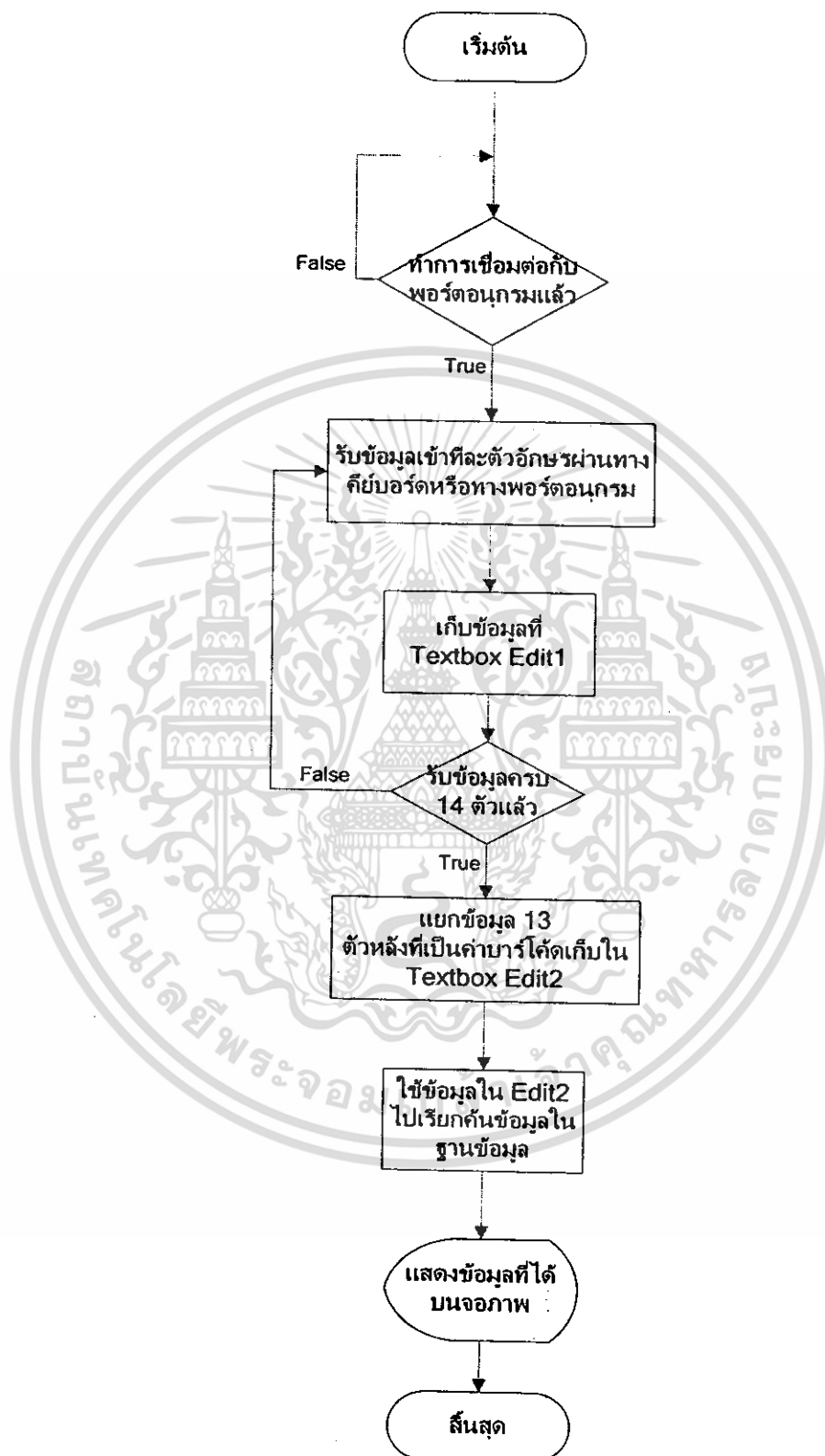
3.9 การออกแบบผังงานของโปรแกรมแสดงผลทางแอลซีดี



รูปที่ 3.12 ผังงานของโปรแกรมแสดงผลทางแอลซีดี

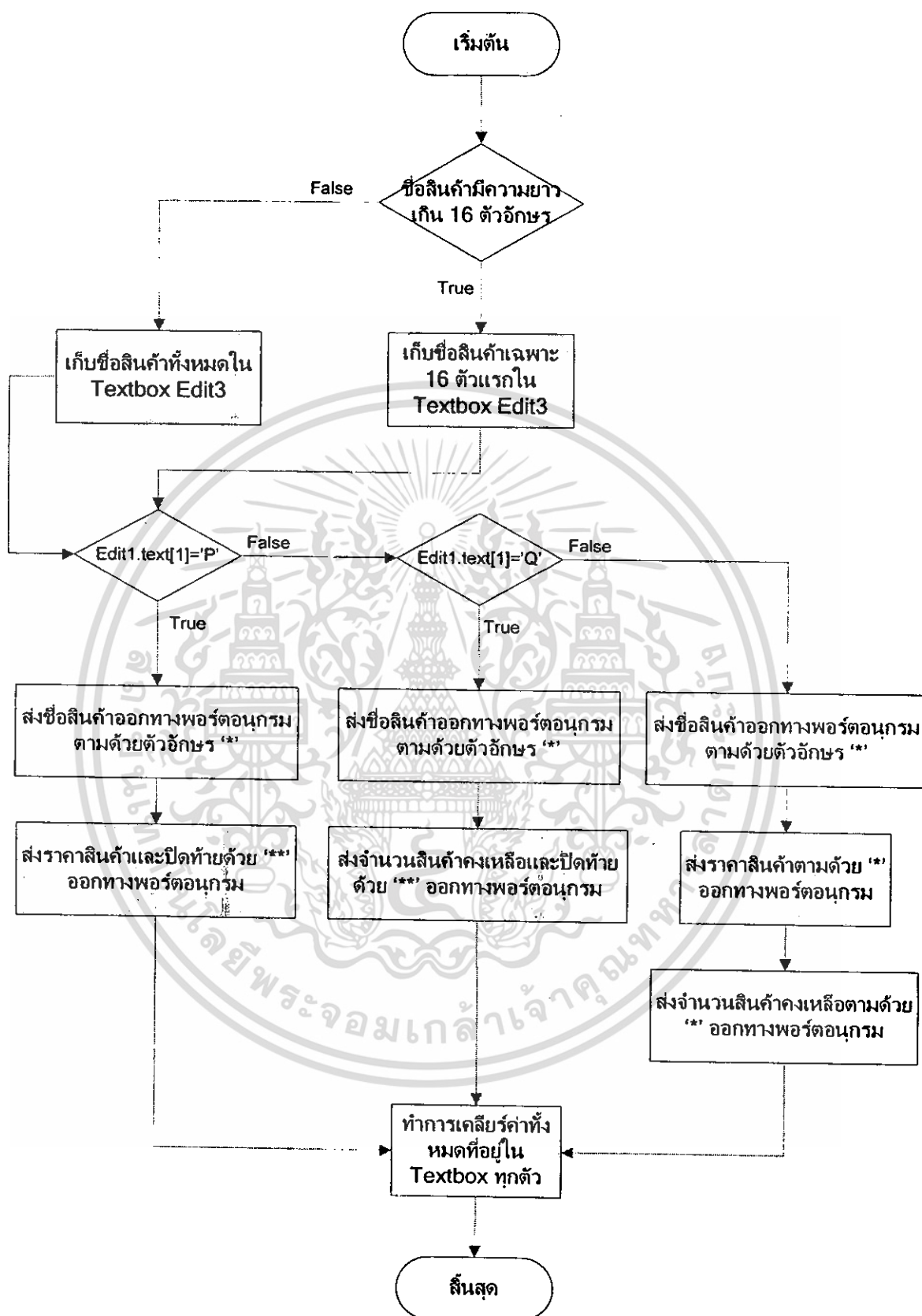
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 การออกแบบผังงานของโปรแกรมแสดงผลทางด้านเซิร์ฟเวอร์



รูปที่ 3.13 ผังงานของโปรแกรมแสดงผลด้านเซิร์ฟเวอร์ขณะรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 ผลงานของ โปรแกรมแสดงผลทางด้านเซิร์ฟเวอร์ขณะส่งข้อมูลกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองวัดค่าสัญญาณที่ส่งจาก AT89C4051 ไปยัง TRW (ด้านบาร์โค้ด)

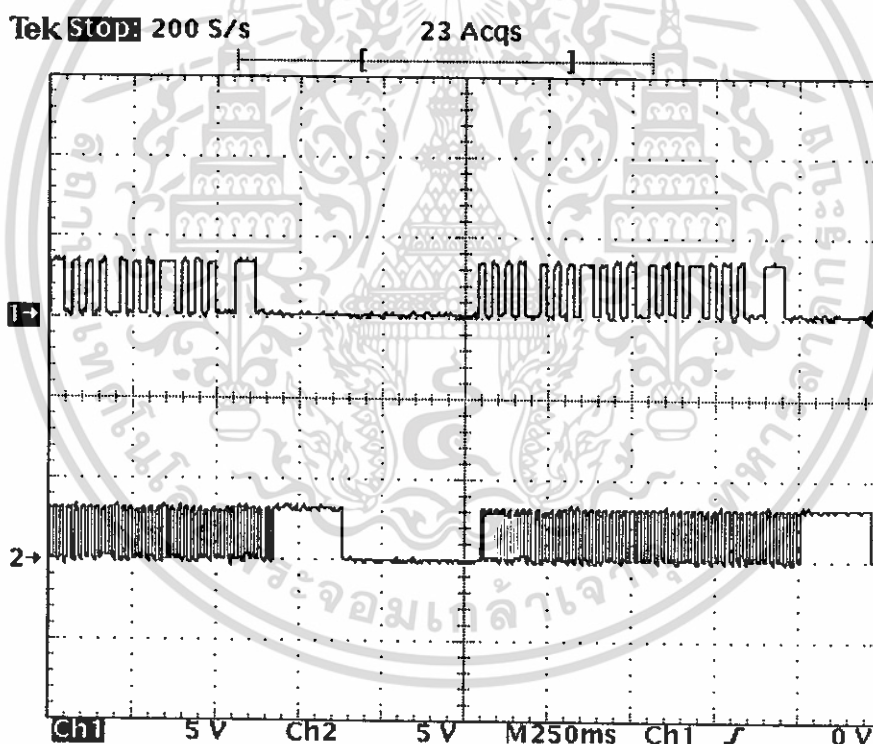
ขั้นตอนการทดลอง

1. โปรแกรมไมโครคอนโทรลเลอร์ เพื่อให้รับค่าผ่านพอร์ตอนุกรม

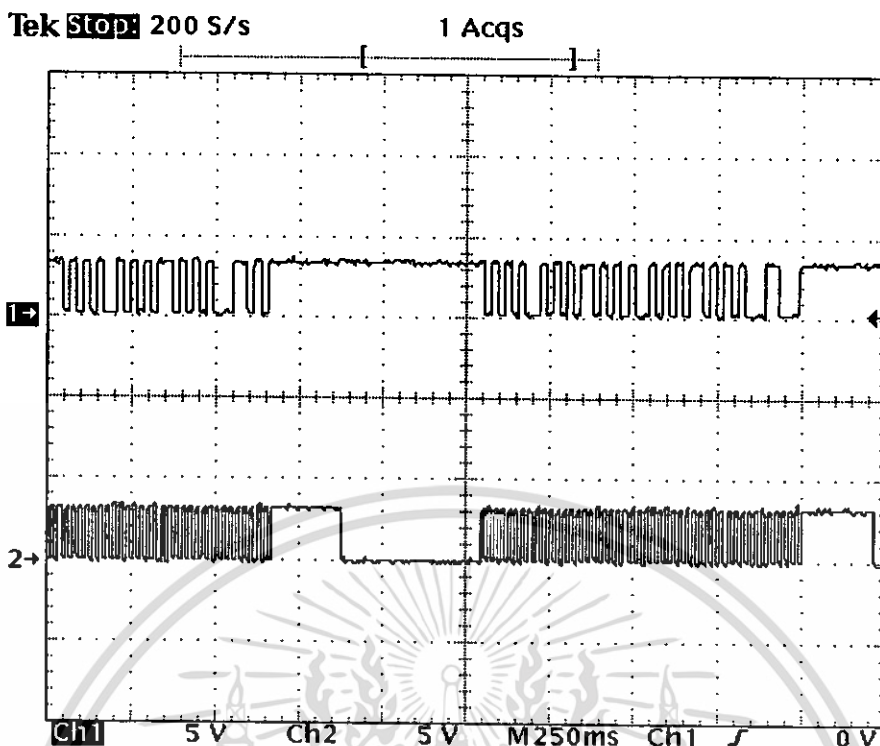
ต้องจรมตามรูปที่ 3.3

2. จ่ายไฟเลี้ยงให้กับวงจร และทำการสแกนบาร์โค้ด
3. ทำการวัดค่าสัญญาณที่ขา 17 ของ ไอซี AT89C4051 ด้านบาร์โค้ด ให้แสดงในช่องสัญญาณที่ 1
4. วัดค่าสัญญาณนาฬิกา จากขา 16 ของ AT89C4051 ให้แสดงในช่องสัญญาณที่ 2

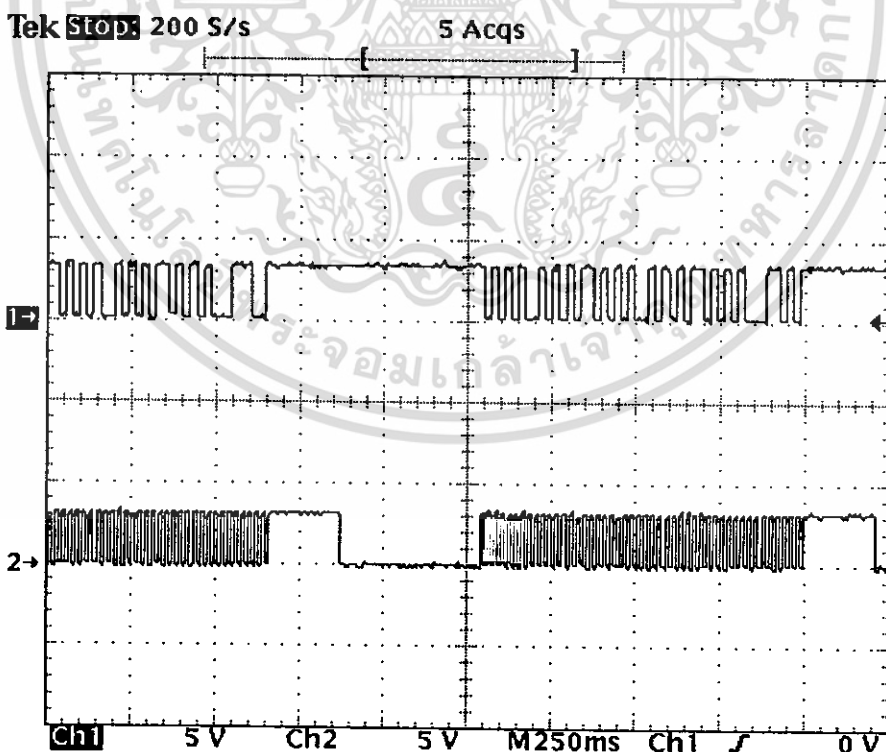
ผลการทดลอง



รูปที่ 4.1 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 1-2 รหัส 8-8)

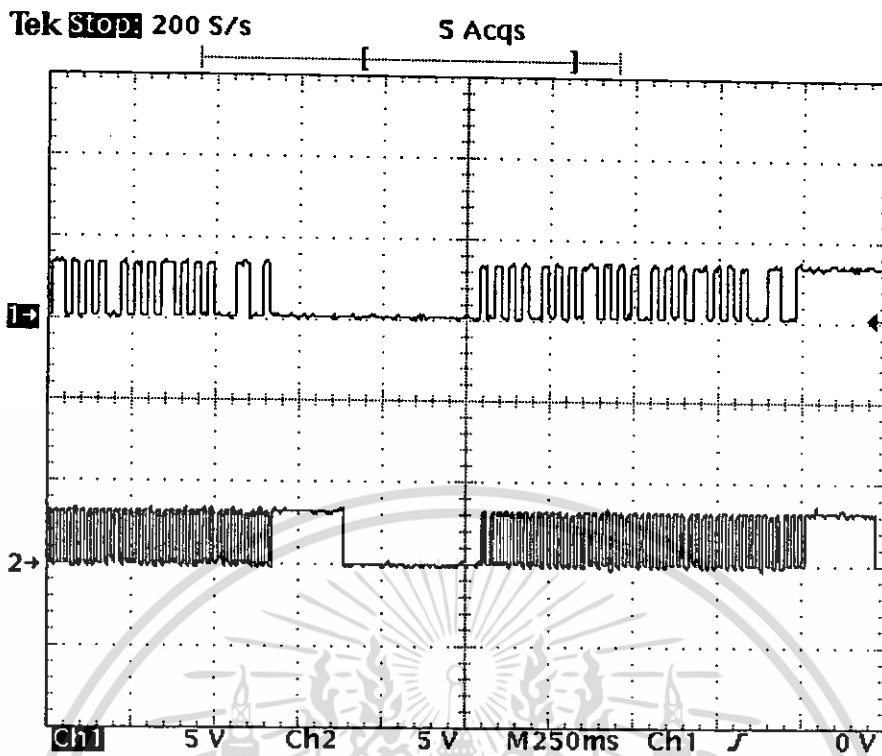


รูปที่ 4.2 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 3-4 รหัส 5-1)

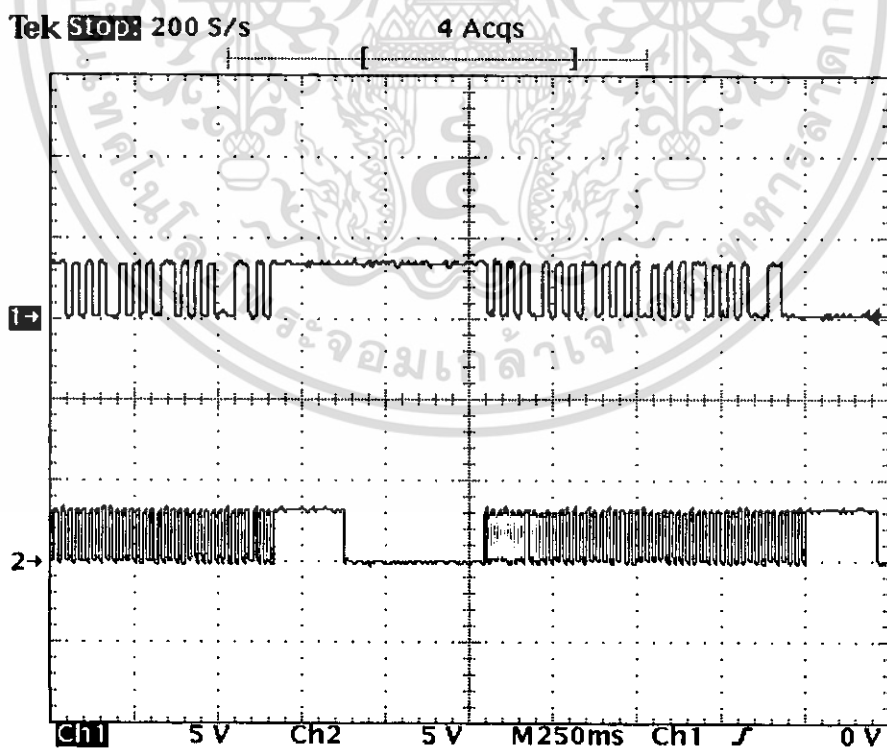


รูปที่ 4.3 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 5-6 รหัส 9-5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

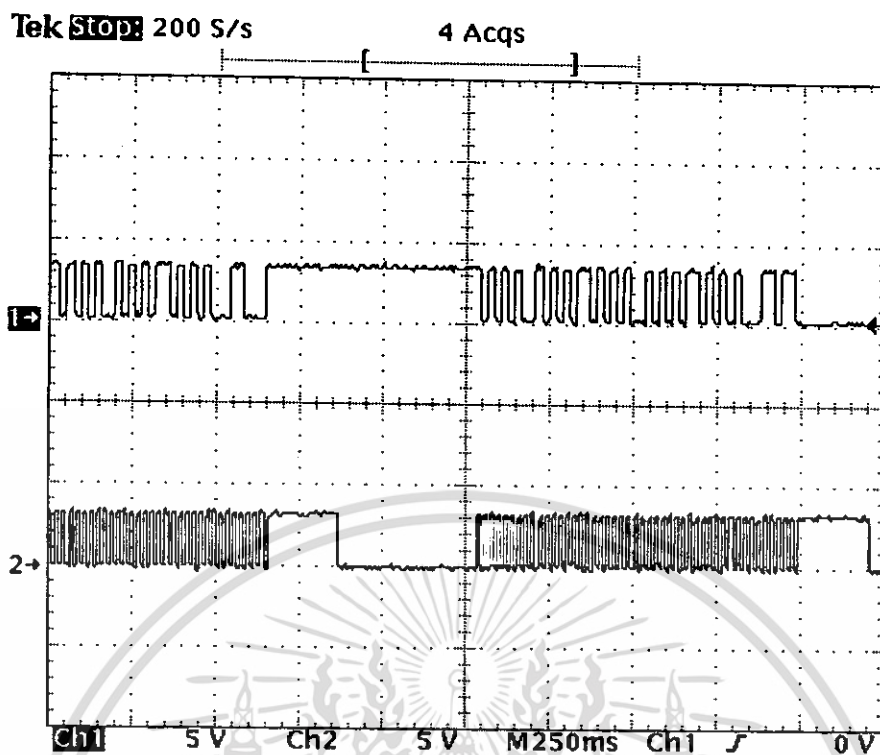


รูปที่ 4.4 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 7-8 รหัส 2-3)

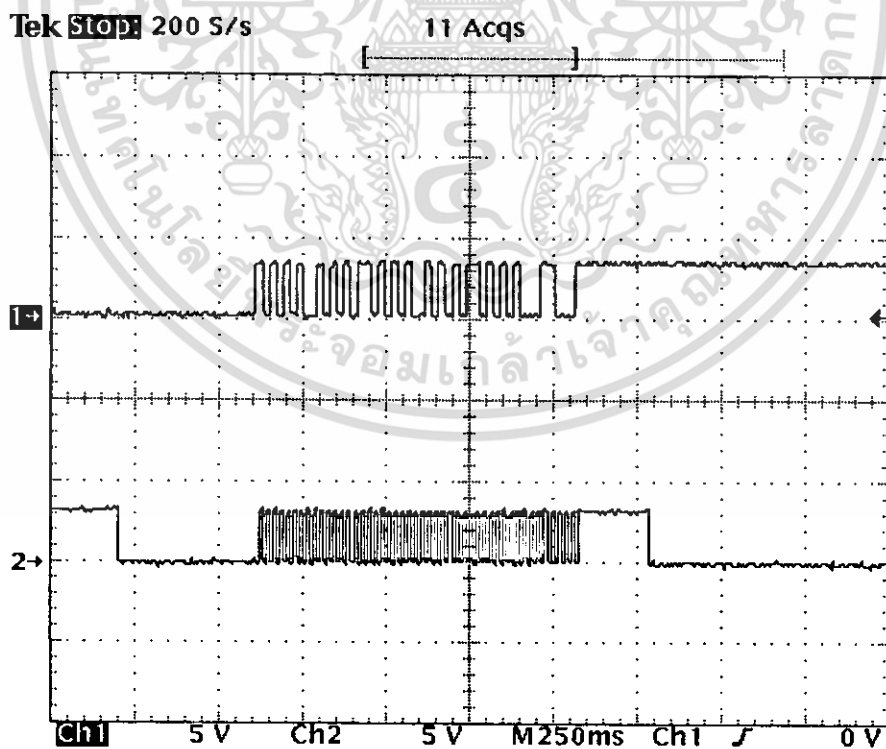


รูปที่ 4.5 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 9-10 รหัส 5-0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 11-12 รหัส 1-6)



รูปที่ 4.7 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 13 รหัส 1)

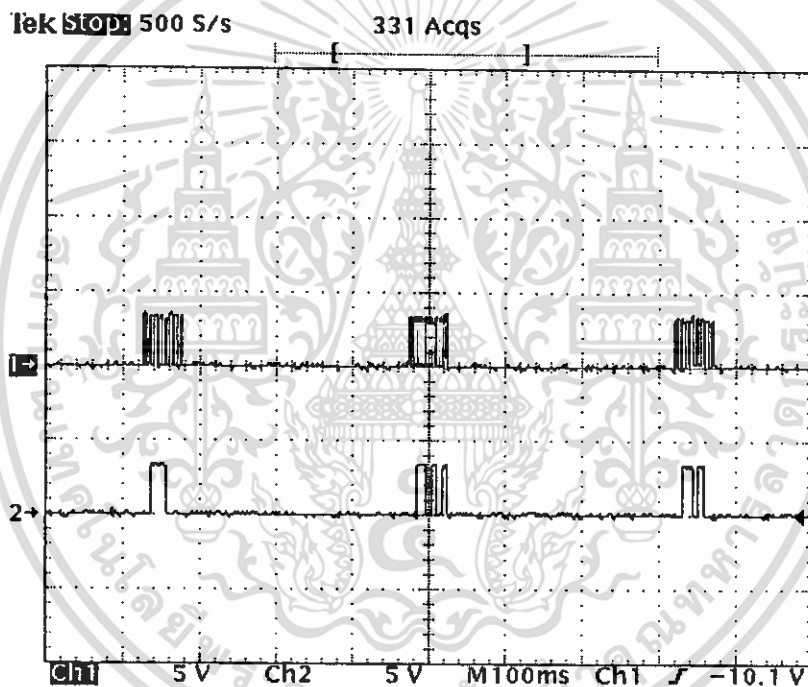
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองวัดค่าสัญญาณที่ส่งจาก TRW ไปยัง AT89C4051 (ด้านเซิร์ฟเวอร์)

ขั้นตอนการทดลอง

1. เขียนโปรแกรมรับค่าจาก TRW ลงไมโครคอนโทรลเลอร์
2. ต่อวงจรตามรูปที่ 3.4
3. จ่ายไฟให้กับวงจร และทำการสแกนบาร์โค้ด
4. วัดค่าสัญญาณนาฬิกา จากขา ขา 16 ของ AT89C4051 ให้แสดงในช่องสัญญาณที่ 1
5. ทำการวัดค่าสัญญาณที่ขา 17 ของ ไอซี AT89C4051 ด้านเซิร์ฟเวอร์ ให้แสดงในช่องสัญญาณที่ 2

ผลการทดลอง



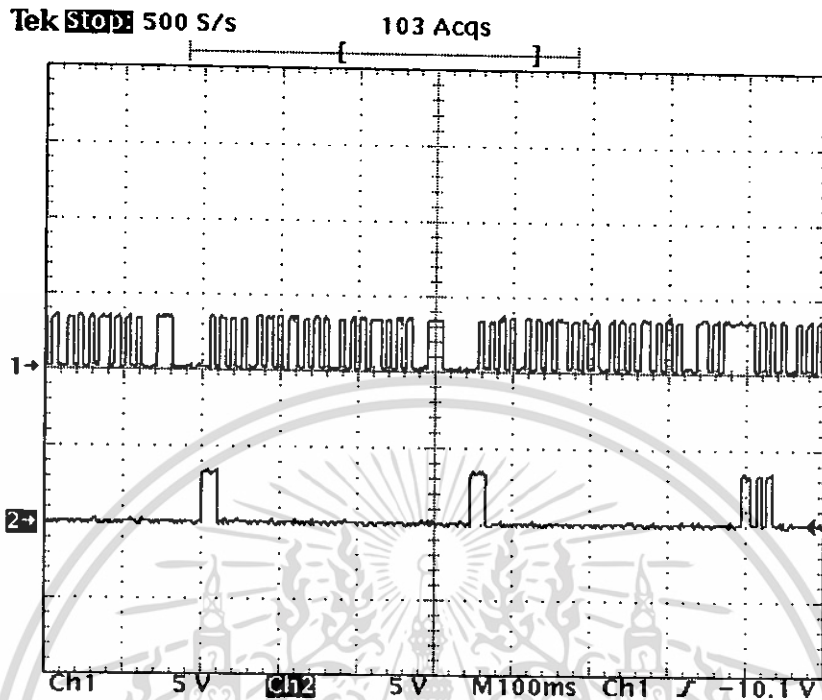
รูปที่ 4.8 แสดงตัวอย่างสัญญาณที่ส่งจาก TRW ไปยัง AT89C4051

4.3 การทดลองเปรียบเทียบสัญญาณด้านบาร์โค้ดกับด้านเซิร์ฟเวอร์

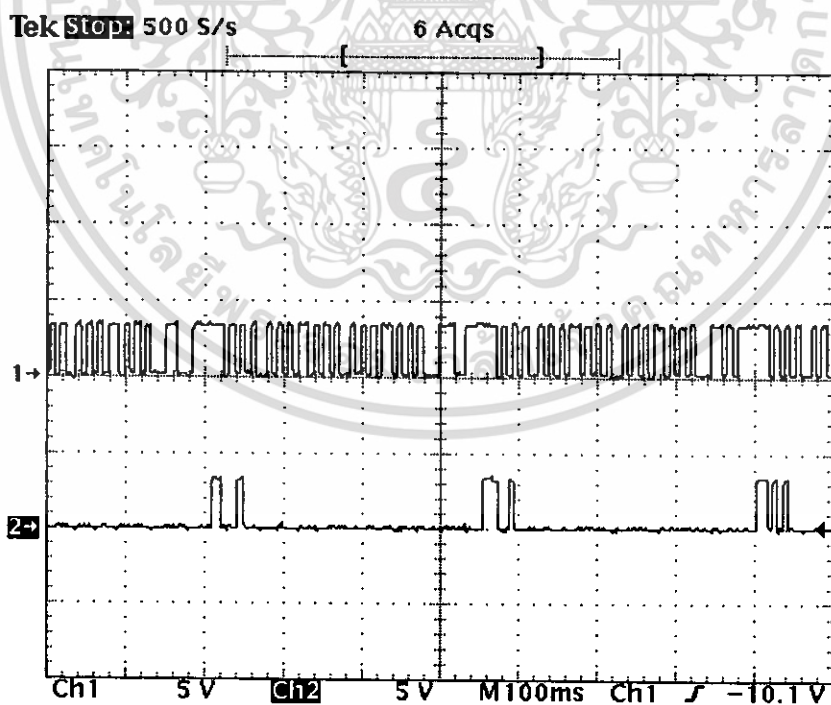
ขั้นตอนการทดลอง

1. วัดค่าสัญญาณจากขาที่ 17 ของไอซี AT89C4051 ด้านบาร์โค้ด ให้แสดงในช่องสัญญาณที่ 1
2. วัดค่าสัญญาณจากขาที่ 17 ของไอซี AT89C4051 ด้านเซิร์ฟเวอร์ ให้แสดงในช่องสัญญาณที่ 2

ผลการทดลอง

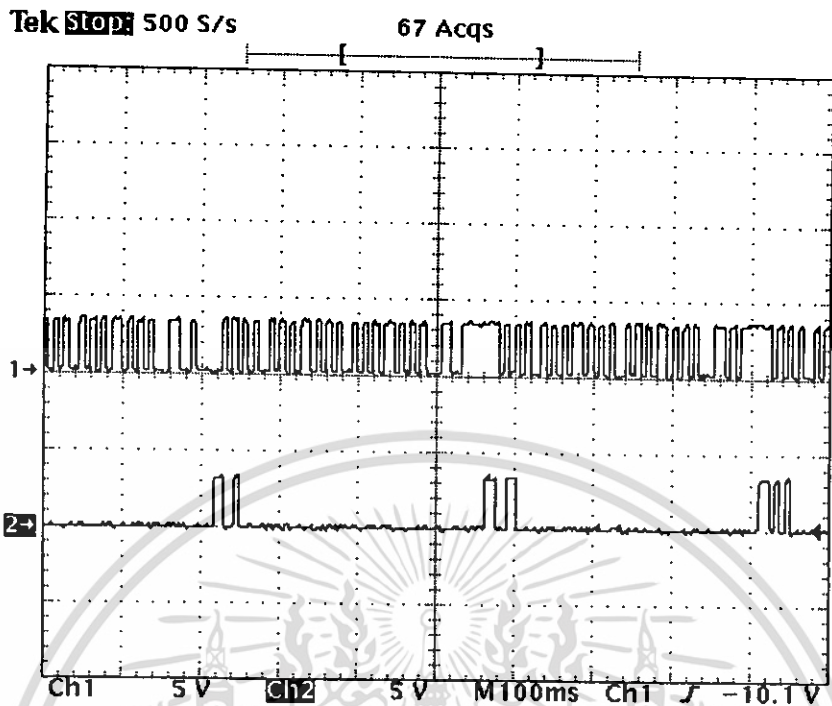


รูปที่ 4.9 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 1-2-3 รหัส 8-8-5)

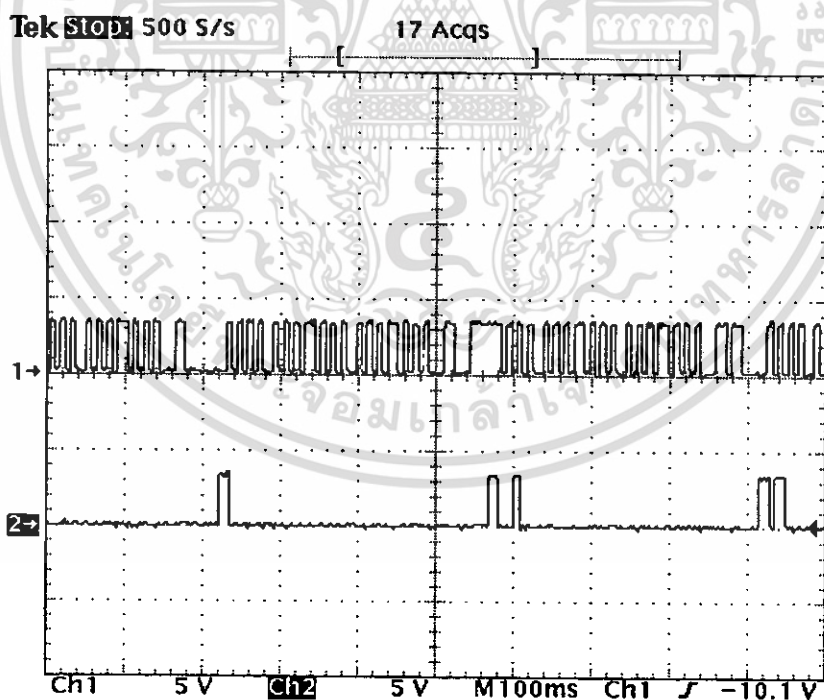


รูปที่ 4.10 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 4-5-6 รหัส 1-9-5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

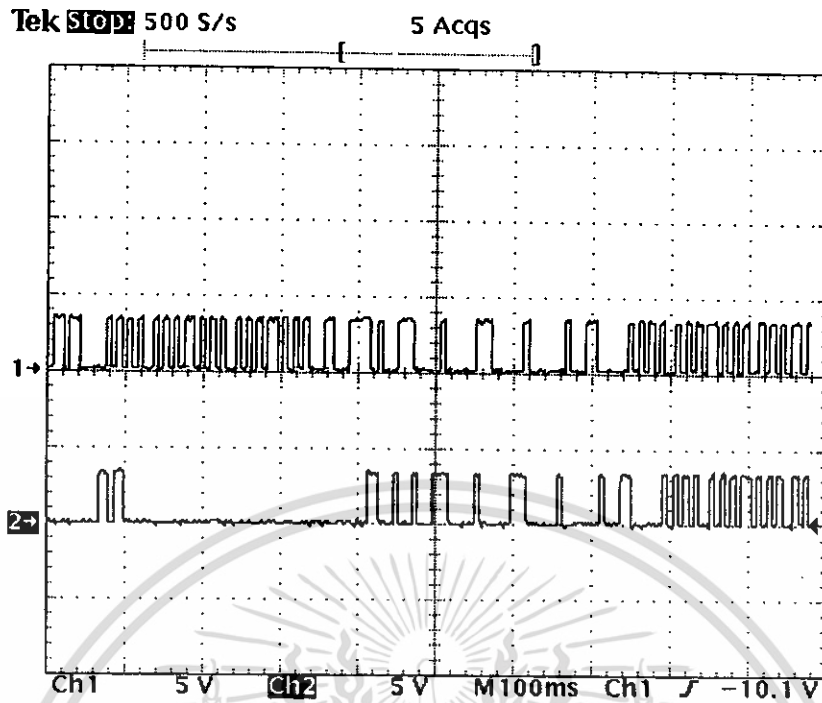


รูปที่ 4.11 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 7-8-9 รหัส 2-3-5)

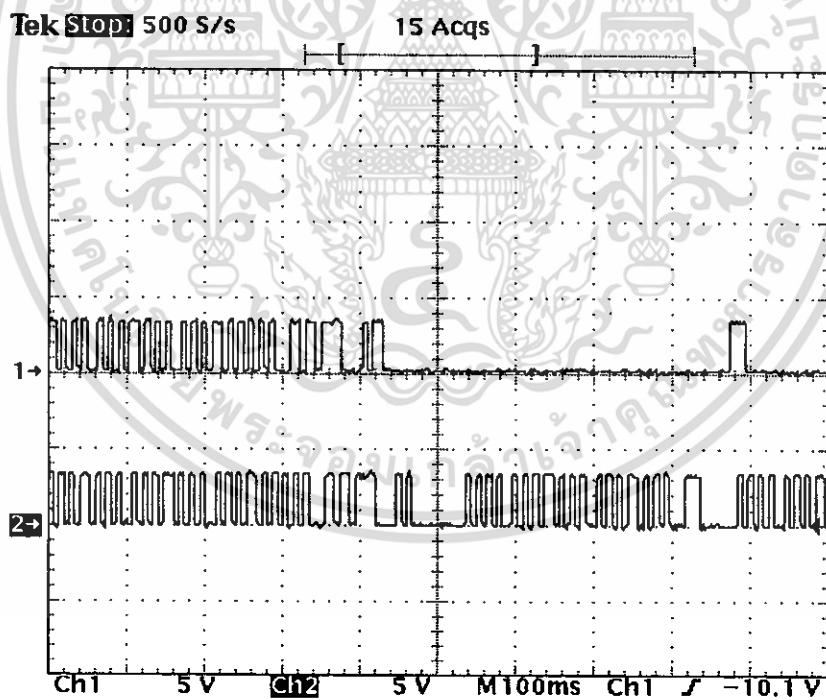


รูปที่ 4.12 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 10-11-12 รหัส 0-1-6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

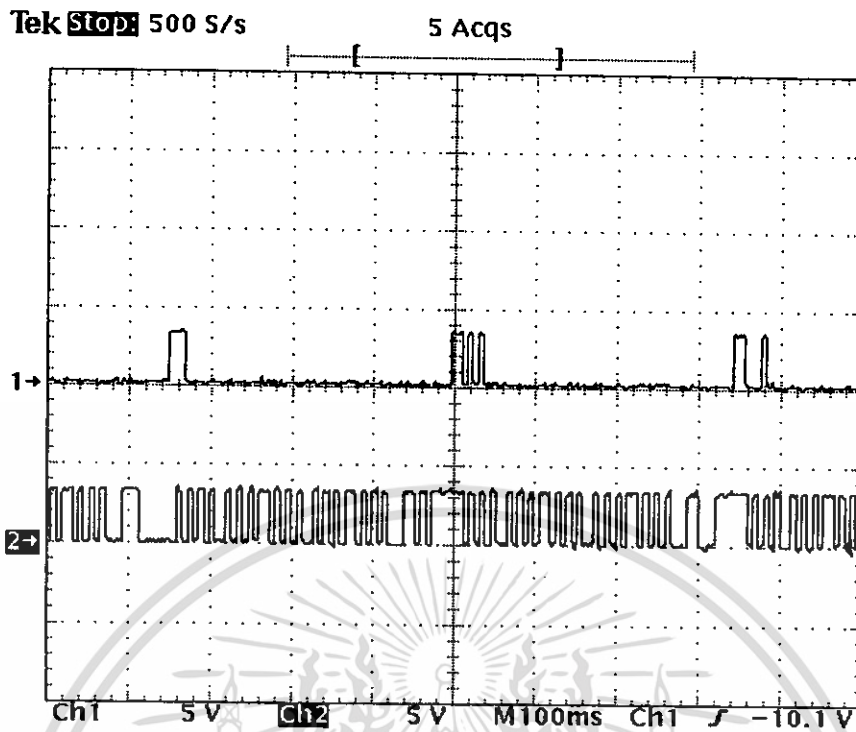


รูปที่ 4.13 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 12-13 รหัส 6-1 และเซตโหมด)

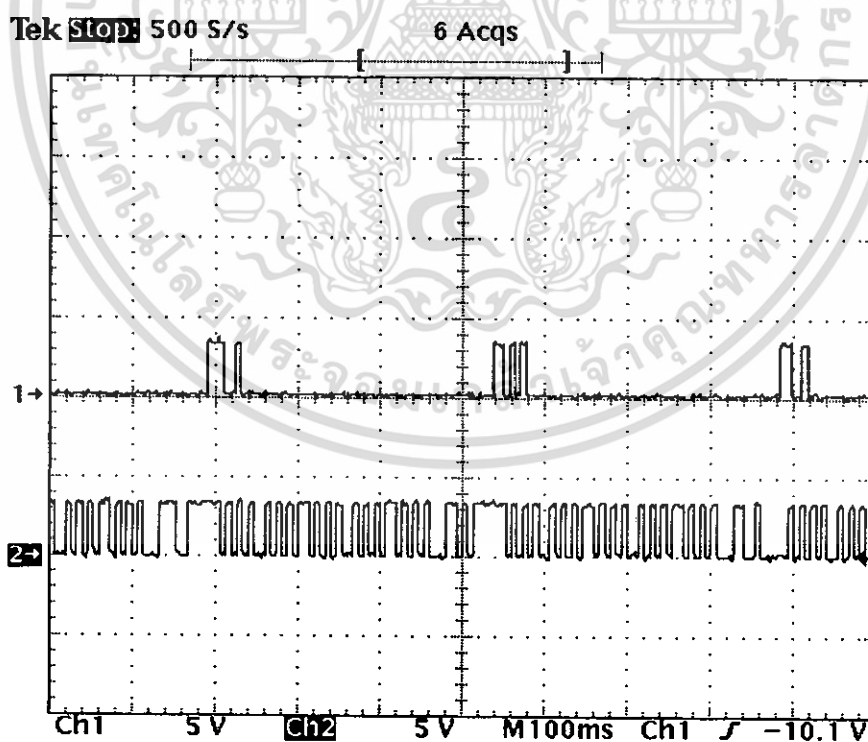


รูปที่ 4.14 แสดงสัญญาณที่วัดได้ (เซตโหมด และไบต์ที่ 1 รหัส 8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

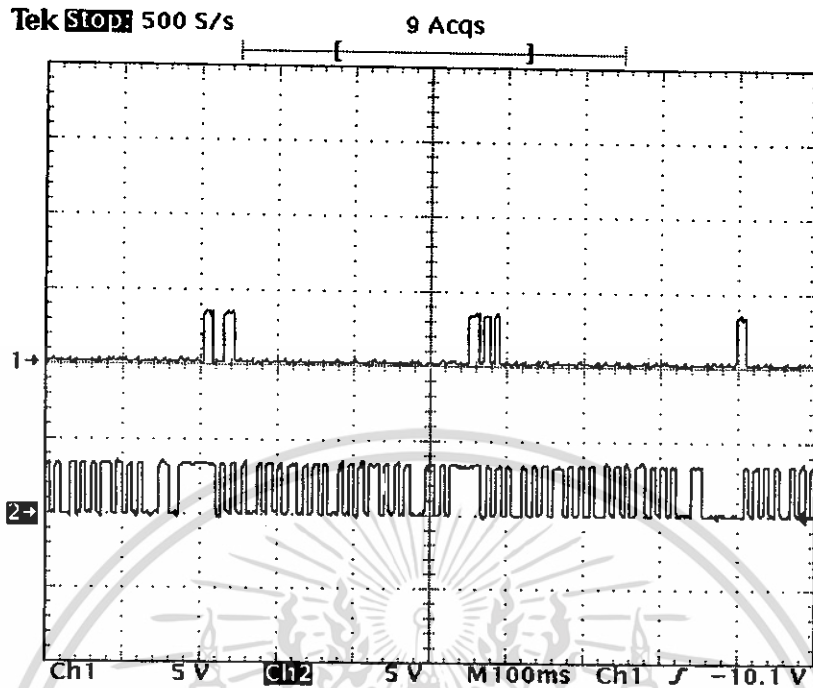


รูปที่ 4.15 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 2-3-4 รหัส 8-5-1)

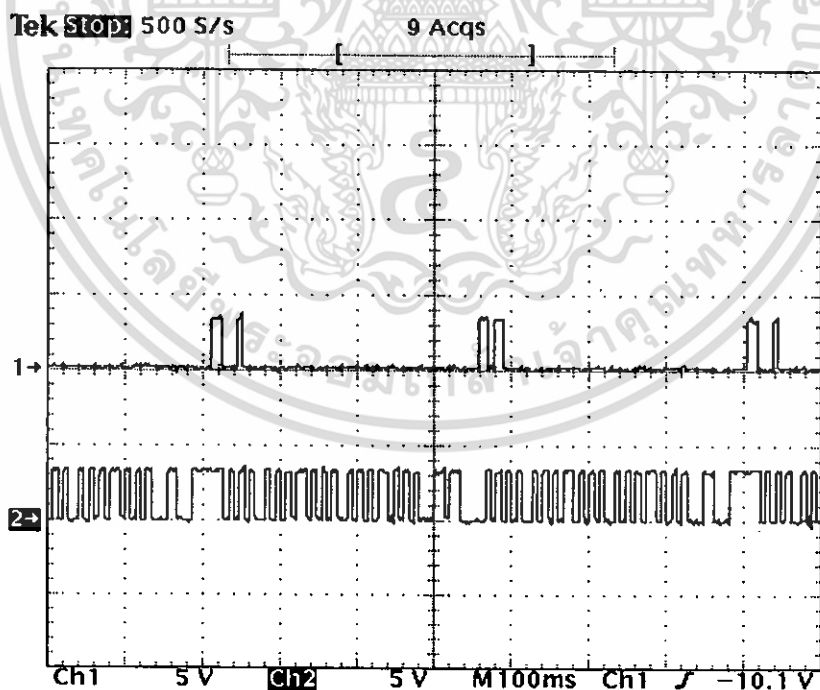


รูปที่ 4.16 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 5-6-7 รหัส 9-5-2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

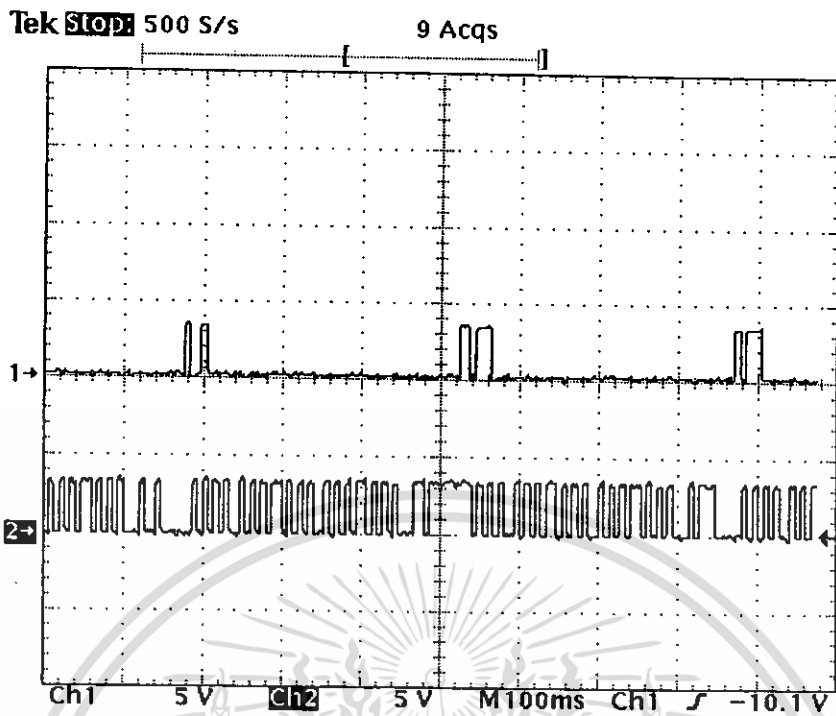


รูปที่ 4.17 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 8-9-10 รหัส 3-5-0)

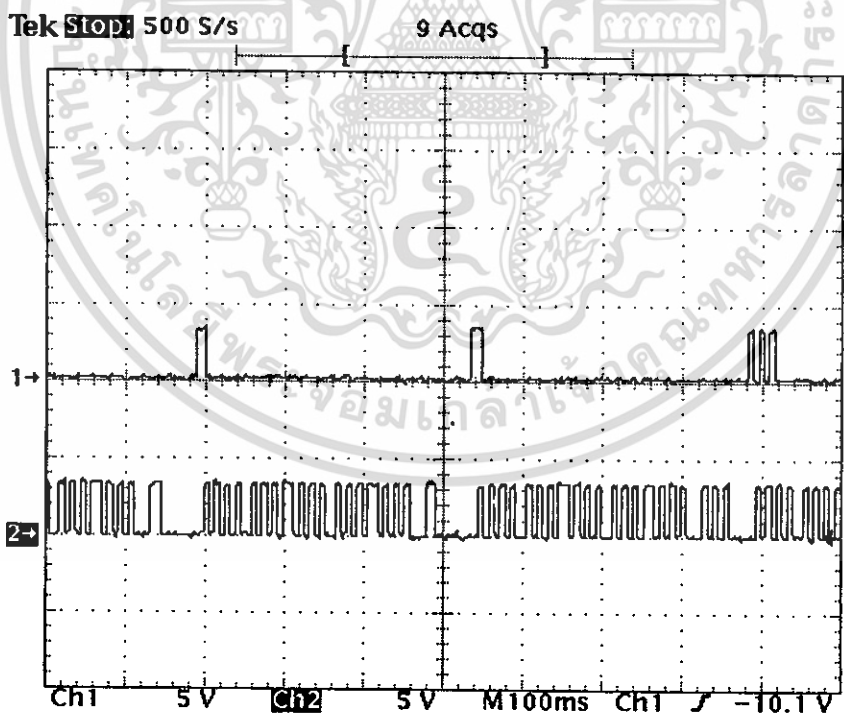


รูปที่ 4.18 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 11-12-13 รหัส 1-6-1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

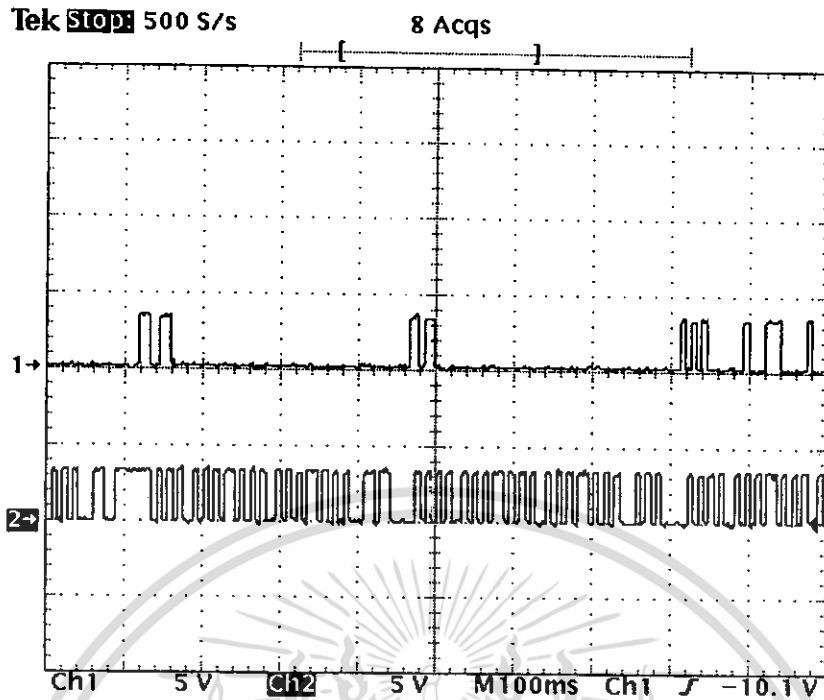


รูปที่ 4.19 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 1-2-3 ของราคาสินค้า S-7-)



รูปที่ 4.20 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 1-2-3 ของราคาสินค้า 0-0-*)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



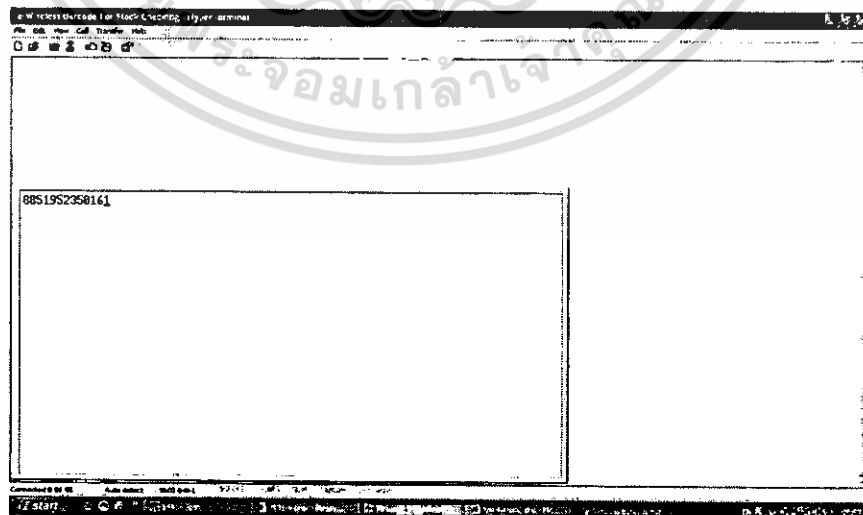
รูปที่ 4.21 แสดงสัญญาณที่วัดได้ (ไบต์ที่ 1-2-3 ของราคาสินค้า 3-6-*)

4.4 การทดลองแสดงผลเป็นหมายเลขบาร์โค้ดบนไฮเปอร์เทอร์มินอล

ขั้นตอนการทดลอง

1. จากวงจรที่ 3.4 ทำการเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม
2. จ่ายไฟเลี้ยงให้กับวงจร และสแกนบาร์โค้ด
3. เปิดโปรแกรมไฮเปอร์เทอร์มินอล เพื่อรับค่าจากพอร์ต COM1

ผลการทดลอง



รูปที่ 4.22 แสดงหน้าจอโปรแกรมไฮเปอร์เทอร์มินอลเมื่อได้รับหมายเลขบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองโปรแกรมเคลฟ

ขั้นตอนการทดลอง

1. เขียนโปรแกรมเคลฟ รongรับหมายเลขบาร์โค้ดที่เข้ามาทางพอร์ตอนุกรม เพื่อทำการเรียกค้นข้อมูลจากฐานข้อมูล ออกมาแสดงบนหน้าจอ

2. ต่อวงจรตามรูปที่ 3.4 จ่ายไฟเลี้ยง และสแกนบาร์โค้ด

ผลการทดลอง

The screenshot shows a window titled 'Form1' with a 'Connect' button and a 'Disconnect' button. Below the buttons, there is a text box containing the barcode number '8851952350161'. Below this, there are two columns of data: 'Barcode Number' with the value '8851952350161', 'Product Name' with the value 'CRYSTAL', 'Price' with the value '\$7.00', and 'Quantity' with the value '36'.

Barcode Number	Product Name
8851952350161	CRYSTAL
Price	Quantity
\$7.00	36

รูปที่ 4.23 แสดงการเรียกค้นข้อมูลจากฐานข้อมูล ออกมาแสดงบนหน้าจอ

4.6 การทดลอง นำค่าที่ได้จากเรียกค้นข้อมูลในฐานข้อมูลกลับ

4.6.1 ทดลองรับข้อมูลที่ได้จากการเรียกค้นฐานข้อมูลมาแสดงบนไฮเปอร์เทอร์มินอลของคอมพิวเตอร์อีกเครื่องหนึ่ง

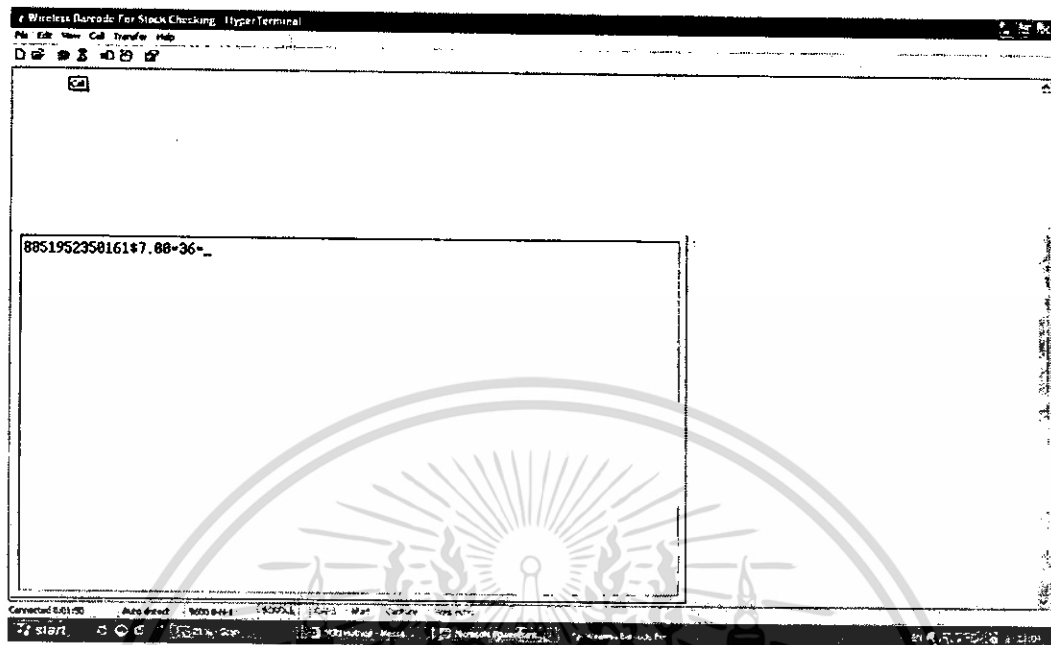
ขั้นตอนการทดลอง

1. จากโปรแกรมเคลฟ ทำการป้อนตัวเลขบาร์โค้ดลงในช่องใส่ข้อความ

2. โปรแกรมเคลฟทำการเรียกค้นข้อมูล ได้แก่ ชื่อสินค้า ราคาสินค้า และจำนวนสินค้าคงเหลือมาแสดงที่ หน้าจอ โปรแกรม พร้อมกับส่งหมายเลขบาร์โค้ด ราคาสินค้า และจำนวนสินค้าคงเหลือออกทางพอร์ตอนุกรม

3. คอมพิวเตอร์ทางฝั่งบาร์โค้ดรับข้อมูลทั้งหมด โดยใช้โปรแกรมไฮเปอร์เทอร์มินอล ได้ผลดังรูปที่ 4.24

ผลการทดลอง



รูปที่ 4.24 แสดงการเรียกค้นข้อมูลจากฐานข้อมูล ออกมาแสดงบนหน้าจอ

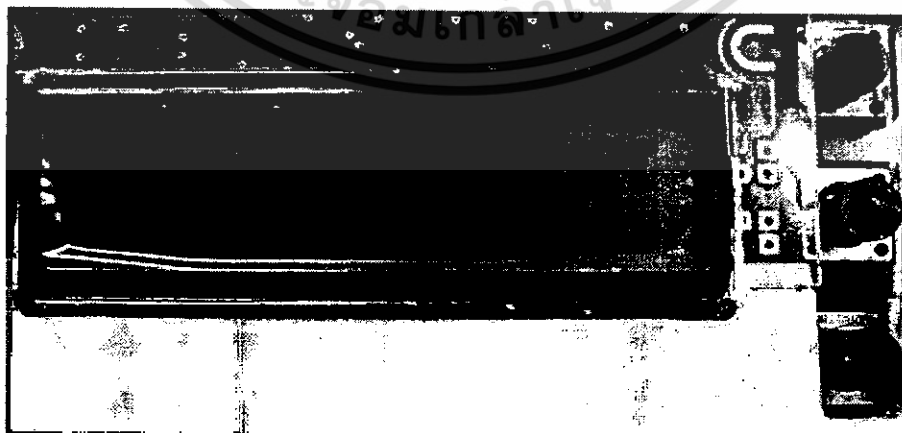
4.7 การทดลองแสดงผลที่เรียกค้นได้จากฐานข้อมูล มาแสดงบนหน้าจอแอลซีดี

กรณีที่ 1 เลือกแสดงเฉพาะราคา

ขั้นตอนการทดลอง

1. ทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ AT89C51 รับค่าจากขา Rx แล้วนำข้อมูลที่ได้มาแสดงที่หน้าจอแอลซีดี ได้ผลดังรูปที่ 4.25

ผลการทดลอง



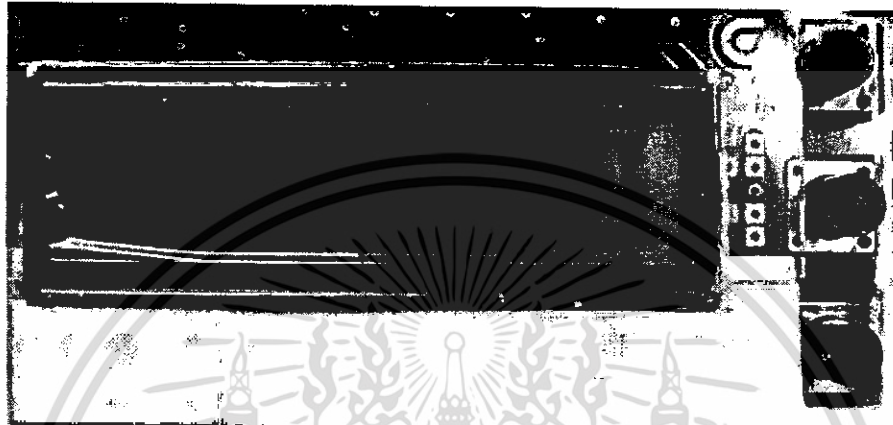
รูปที่ 4.25 แสดงผลที่เรียกค้นได้จากฐานข้อมูล มาแสดงบนหน้าจอแอลซีดี โดยแสดงเฉพาะราคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ 2 เลือกแสดงเฉพาะจำนวน
ขั้นตอนการทดลอง

1. ทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ AT89C51 รับค่าจากขา Rx แล้วนำข้อมูลที่
 ได้มาแสดงที่หน้าจอแอลซีดี ได้ผลดังรูปที่ 4.26

ผลการทดลอง



รูปที่ 4.26 แสดงผลที่เรียกค้นได้จากฐานข้อมูล มาแสดงบนหน้าจอแอลซีดี โดยแสดงเฉพาะจำนวน

กรณีที่ 3 เลือกแสดงทั้งราคาและจำนวน
ขั้นตอนการทดลอง

1. ทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ AT89C51 รับค่าจากขา Rx แล้วนำข้อมูลที่
 ได้มาแสดงที่หน้าจอแอลซีดี ได้ผลดังรูปที่ 4.27

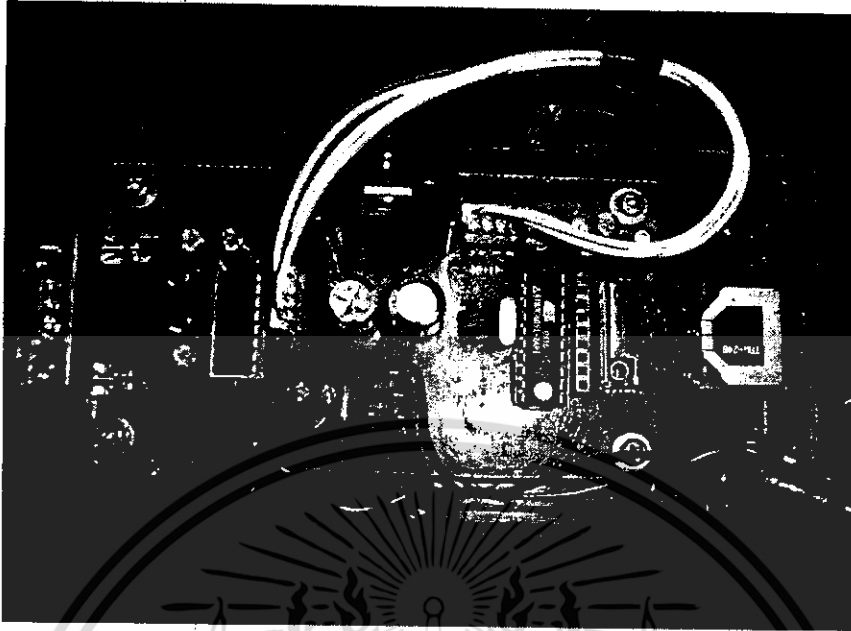
ผลการทดลอง



รูปที่ 4.27 แสดงผลที่เรียกค้นได้จากฐานข้อมูล มาแสดงบนหน้าจอแอลซีดี โดยแสดงทั้งราคาและจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 รูปวงจรที่ใช้ในการทดลอง

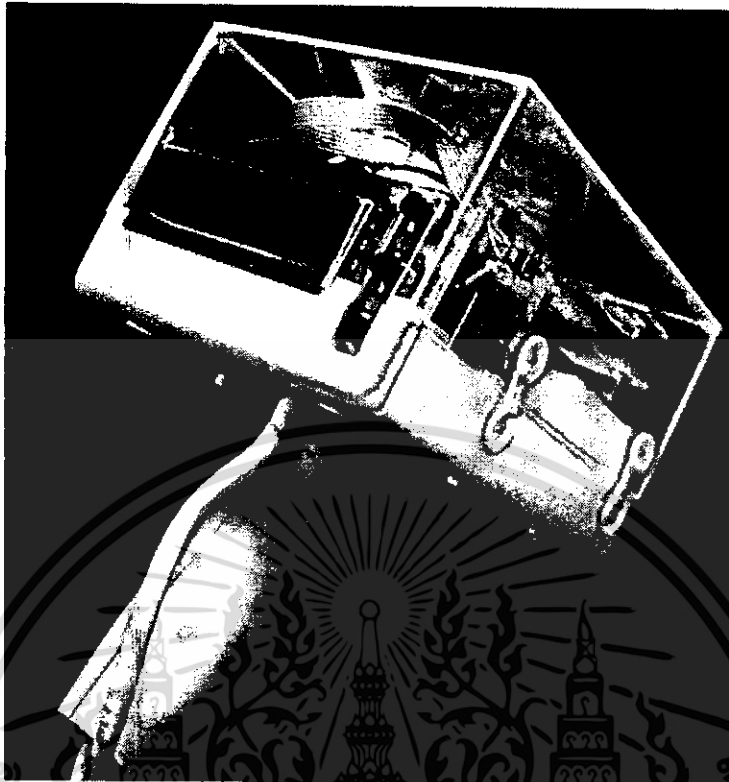


รูปที่ 4.28 วงจรที่ใช้รับ-ส่งข้อมูลฝั่งเซิร์ฟเวอร์



รูปที่ 4.29 วงจรที่ใช้รับ-ส่งข้อมูลฝั่งบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 เครื่องสแกนบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุปผล

5.1 บทวิจารณ์

จากการศึกษาเกี่ยวกับการทำงานของเครื่องอ่านบาร์โค้ด, ไมโครคอนโทรลเลอร์, เครื่องรับส่งผ่านคลื่นวิทยุ, ฐานข้อมูล และส่วนของการสื่อสารผ่านพอร์ตอนุกรม พบปัญหาดังต่อไปนี้

5.1.1 ส่วนของเครื่องอ่านบาร์โค้ด

สำหรับเครื่องอ่านบาร์โค้ดที่ใช้ในปฏิยานิพนธ์นี้ สามารถใช้งานได้ปกติ จากเดิมเครื่องอ่านมีไฟเลี้ยงอยู่ในช่วง 4 ถึง 6 โวลต์ โดยต้องใช้หม้อแปลง แต่เนื่องจากปฏิยานิพนธ์นี้ต้องการทำให้เครื่องอ่านบาร์โค้ดเป็นแบบไร้สาย จึงต้องทำการสร้างแหล่งจ่ายไฟจากแบตเตอรี่ให้ได้แรงดันไฟฟ้า 6 โวลต์

5.1.2 ส่วนของไมโครคอนโทรลเลอร์และพอร์ตอนุกรม

จากการออกแบบระบบสื่อสารไร้สาย ได้ทำการออกแบบให้ทางฝั่งส่งใช้ไมโครคอนโทรลเลอร์ 2 ตัว คือ AT89C51 และ AT89C4051 โดย AT89C4051 จะทำหน้าที่ในการรับค่าจากเครื่องอ่านบาร์โค้ดผ่านทางพอร์ตอนุกรม ซึ่งต้องใช้พอร์ต 3.0 และ 3.1 ของไมโครคอนโทรลเลอร์ นอกจากนี้ในขั้นตอนการออกแบบ ได้มีการคำนึงถึงการแสดงผลออกทางหน้าจอแอลซีดี แบบ 16 x 2 แต่เนื่องจากการที่ต้องเปลี่ยนไมโครคอนโทรลเลอร์มาใช้ AT89C4051 ซึ่งมีเพียง 20 ขา ส่งผลทำให้ไม่สามารถเขียนโปรแกรมเพื่อควบคุมการทำงานดังกล่าวได้

ปัญหาในส่วนของโปรแกรมคือ การคำนวณเวลาที่ใช้ในการทำงานของโปรแกรมเพื่อให้สามารถรับและส่งข้อมูลกันได้ทัน จึงต้องมีการลองผิดลองถูกหลายครั้ง

และเนื่องจากในการทำปฏิยานิพนธ์นี้ได้ใช้คอมพิวเตอร์โน้ตบุ๊กซึ่งไม่มีพอร์ตอนุกรมในการทำการทดลอง จึงต้องใช้อุปกรณ์แปลงพอร์ตอนุกรมให้เป็นพอร์ตยูเอสบี เพื่อให้สามารถใช้งานกับคอมพิวเตอร์โน้ตบุ๊กได้ แต่จะเกิดปัญหาเนื่องจากอุปกรณ์แปลงพอร์ตไม่สามารถส่งข้อมูลกลับไปได้ จึงไม่สามารถทดลองในส่วนของการส่งข้อมูลขากลับไปแสดงที่หน้าจอแอลซีดีได้

5.1.3 การรับส่งผ่านคลื่นวิทยุ

ในส่วนของการรับส่งผ่านคลื่นวิทยุ นั้น สามารถส่งและรับรหัสข้อมูลสินค้าได้จริง เนื่องจาก TRW 2.4G มีประสิทธิภาพค่อนข้างสูง แต่มีข้อเสียคือราคาสูง

5.1.4 ส่วนของฐานข้อมูล

การเขียนโปรแกรมเคลฟและบันทึกค่าต่างๆลงในฐานข้อมูลสำหรับโครงงานนี้สามารถทำงานได้อย่างปกติ โดยสามารถทำการเรียกค้นข้อมูลชื่อสินค้า, ราคาสินค้าและจำนวนสินค้าคงเหลือได้จากฐานข้อมูลที่มีอยู่ จากการที่มีรหัสสินค้าผ่านเข้ามาทางพอร์ตอนุกรม

5.2 บทสรุป

จากการที่ได้ศึกษาและทดลองเพื่อสร้างเครื่องอ่านบาร์โค้ดแบบไร้สายนั้น สามารถส่งรหัสสินค้า จากด้านส่งไปยังด้านรับได้และสามารถเรียกค้นข้อมูลของรหัสสินค้านั้นจากฐานข้อมูล เพื่อแสดงผลบน หน้าจอฝั่งเซิร์ฟเวอร์ได้ พร้อมกับสามารถส่งข้อมูลต่างๆกลับมาแสดงที่หน้าจอแอลซีดีทางฝั่งบาร์โค้ดได้

5.3 แนวทางที่จะทำการพัฒนาต่อ

เนื่องจากการส่งข้อมูลแบบไร้สายนั้นมีการเกิดสัญญาณรบกวนทำให้ในบางครั้งการส่งข้อมูล อาจเกิดผิดพลาดได้ดังนั้นจึงต้องทำการปรับปรุงความสามารถในการส่งสัญญาณของวงจรให้ดียิ่งขึ้น เพื่อให้ได้ข้อมูลที่ถูกต้องและได้ระยะเวลาในการใช้งานที่ไกลขึ้นกว่าเดิมด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ชีรบุญย์ หล่อวิเชียรรุ่ง , นคร ภักดีชาติ , ชัยวัฒน์ ลิมพิจิตรวิไล “ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษาซี” , บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด , กรุงเทพฯ , 2521 .
2. รศ.สมยศ จุณณปิยะ , “การประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล MCS-51” คณะวิศวกรรมศาสตร์ , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2546.
3. อุดม รานอก , “ภาษาซีสำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51” , นนทบุรี , ไอดีซีฯ , 2548.
4. สัจจะ จรัสรุ่งวิীর , จักรพงษ์ สุขประเสริฐ “เริ่มต้นอย่างมืออาชีพด้วย Delphi 7 ฉบับสมบูรณ์” , บริษัท เอช เอ็น กรุ๊ป จำกัด , กรุงเทพฯ , 2546.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sourcecode โปรแกรม Delphi

```
unit bar;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, CPortCtl, CPort, DB, DBTables, StdCtrls, Mask, DBCtrls,
  ExtCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    DataSource1: TDataSource;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    Edit1: TEdit;
    Query1: TQuery;
    ComPort1: TComPort;
    ComLed1: TComLed;
    Edit2: TEdit;
    Edit3: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}
```

```
//Event ที่จะทำงานเมื่อมีข้อมูลเข้ามาทางพอร์ค Serial
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var data:string; //ประกาศตัวแปร data เป็นตัวแปรชนิด String
begin
  //คำสั่งที่ทำการเก็บข้อมูลที่เข้ามาทางพอร์ค Serial เข้าไปในตัวแปร data
  comport1.ReadStr(data, count);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//นำข้อมูล data มาเก็บและแสดงทาง textbox ชื่อ Edit1
Edit1.Text:=Edit1.Text+data;
end;

//Event ที่ทำงานเมื่อมีการคลิกที่ Button1 เพื่อทำการเชื่อมต่อกับพอร์ต Serial
procedure TForm1.Button1Click(Sender: TObject);
begin
    comport1.Open; //เปิดพอร์ต Serial
    comport1.Connected:=true; //คำสั่งเพื่อทำการเชื่อมต่อ
end;

//Event ที่ทำงานเมื่อมีการคลิกที่ Button2 เพื่อยกเลิกการเชื่อมต่อกับพอร์ต Serial
procedure TForm1.Button2Click(Sender: TObject);
begin
    comport1.Connected:=false; //คำสั่งเพื่อยกเลิกการเชื่อมต่อ
    comport1.Close; //ปิดพอร์ต Serial
end;

//Event ที่ทำงานเมื่อข้อความใน Textbox ชื่อ Edit1 มีการเปลี่ยนแปลง
procedure TForm1.Edit1Change(Sender: TObject);
var x:integer; //ประกาศตัวแปร x เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม
begin
    //ตรวจสอบเงื่อนไขว่าเมื่อข้อมูลที่ได้รับมารอบ 14 ตัวแล้วจึงเริ่มการทำงาน
    if Edit1.GetTextLen>=14 then
    begin
        //วนรอบคัดค่าเฉพาะค่าบาร์โค้ด 13 ตัวหลังมาแสดงที่ Textbox ชื่อ Edit2 และนำค่าที่ได้ไปทำการเรียกค้นข้อมูลในฐานข้อมูลต่อไป
        for x:=2 to 14 do
            Edit2.Text:=Edit2.Text + Edit1.Text[x];
        //เริ่มทำการเรียกค้นข้อมูลโดยผ่านทาง Component Query1 โดยใช้ภาษา SQL ดังนี้
        //select code,product,quantity,price
        //from barcode
        //where code =:param
        with Query1 do
            begin
                close; //ทำการ clear ค่าที่เรียกค้นไว้ก่อนหน้า
                //กำหนดค่าให้ตัวแปร param ที่ใช้ใน SQL เป็นข้อมูลบาร์โค้ด 13 หลักที่อยู่ใน Edit2
                parambyname('param').AsString:= Edit2.Text;
                open; //เริ่มทำการเรียกค้น
            end;
        //ตรวจสอบว่าข้อสินค้าที่ได้มีความยาวเกิน 16 หลักหรือไม่
        //กรณีไม่เกิน 16 ตัวให้นำค่าทั้งหมดเก็บและแสดงใน Textbox Edit3
        if DBEdit2.GetTextLen<=16 then
            begin
                for x:=1 to DBEdit2.GetTextLen do
                    Edit3.Text:=Edit3.Text+DBEdit2.Text[x];
                end;
            //กรณียาวเกิน 16 ตัวให้เก็บค่าเฉพาะ 16 ตัวแรกเท่านั้น
            if DBEdit2.GetTextLen>16 then
                begin
                    for x:=1 to 16 do
                        Edit3.Text:=Edit3.Text+DBEdit2.Text[x];
                    end;
                //ตรวจสอบค่าเงื่อนไขการส่งข้อมูลกลับว่าเป็น 'P', 'Q' หรือ 'B'
                //กรณีเป็น 'P' แสดงว่าร้องขอค่าราคาสินค้า
                if Edit1.Text[1]='P' then
                    begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//คำสั่งการส่งข้อมูลออกทางพอร์ต Serial โดยส่งชื่อสินค้า 16 หลักและราคาสินค้า โดยมี '*' เป็นตัวขึ้นระหว่างข้อมูลและมี '**'
เป็นตัวปิดท้ายข้อมูล
comport1.WriteString(Edit3.Text);
comport1.WriteString('*');
comport1.WriteString(DBEdit3.Text);
comport1.WriteString '**');
end;

//กรณีเป็น 'Q' แสดงว่าร้องขอค่าจำนวนสินค้าคงเหลือ
if Edit1.Text[1]='Q' then
begin
//คำสั่งการส่งข้อมูลออกทางพอร์ต Serial โดยส่งชื่อสินค้า 16 หลักและจำนวนสินค้าคงเหลือ โดยมี '*' เป็นตัวขึ้นระหว่างข้อมูลและมี
 '**' เป็นตัวปิดท้ายข้อมูล
comport1.WriteString(Edit3.Text);
comport1.WriteString('*');
comport1.WriteString(DBEdit4.Text);
comport1.WriteString '**');
end;

//กรณีเป็น 'B' แสดงว่าร้องขอทั้งการราคาสินค้าและจำนวนสินค้าคงเหลือ
if Edit1.Text[1]='B' then
begin
//คำสั่งการส่งข้อมูลออกทางพอร์ต Serial โดยส่งชื่อสินค้า 16 หลักและจำนวนสินค้าคงเหลือ โดยมี '*' เป็นตัวขึ้นระหว่างข้อมูลและมี
 '**' เป็นตัวปิดท้ายข้อมูล
comport1.WriteString(Edit3.Text);
comport1.WriteString('*');
comport1.WriteString(DBEdit3.Text);
comport1.WriteString('*');
comport1.WriteString(DBEdit4.Text);
comport1.WriteString('*');
end;

//ทำการ clear ค่าที่อยู่ใน Textbox ทุกตัวเพื่อรองรับค่าใหม่
Edit1.Clear;
Edit2.Clear;
Edit3.Clear;
end;

end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุม AT89C4051 ผังส่ง

```
//Transmitter.c
```

```
#include<reg2051.h>
#define CE P1_2 // pin 5      กำหนดพอร์ต 1.2ให้กับขา CE ของ TRW
#define CS P1_3 // pin 2      กำหนดพอร์ต 1.3ให้กับขา CS ของ TRW
#define DR1 P1_6 // pin 3     กำหนดพอร์ต 1.6 ให้กับขา DR1 ของ TRW
#define CLK P1_4 // pin 4     กำหนดพอร์ต 1.4 ให้กับขา CLK ของ TRW
#define DAT P1_5 // pin 1     กำหนดพอร์ต 1.5 ให้กับขา DATA ของ TRW
unsigned char x=0,y=0; //กำหนดตัวแปรเพื่อใช้ในอินเทอร์พรีด

void Wait(unsigned char n) //ฟังก์ชันดีเลย์
{
    int i;
    while(n)
    {
        i = 0x01;
        while(i--);
        n--;
    }
}

void Init_TRW24G(void) //ฟังก์ชันกำหนดค่าเริ่มต้นของ TRW
{
    //wait(10);
    CE = 0;
    CS = 0;
    CLK = 0;
    DAT = 0;
    DR1 = 0;
    //wait(10);
}

void CLK_TRW24(void) //ฟังก์ชันกำหนดนาฬิกาของ TRW
{
    CLK = 0;
    Wait(1);
    CLK = 1;
    Wait(1);
}

void Write_TRW24(unsigned char Data) //ฟังก์ชันเขียนค่าDataให้กับTRW
{
    unsigned char i;
    bit Out; // กำหนดตัวแปรOutมีค่าเป็นbit 0 หรือ 1
    for (i=0;i<8;i++) // วนเขียนค่าจนครบ 1 ไบต์
    {
        Out = Data & 0x80; //Data bit ชั่วสุด แอนกับ 1 เก็บค่าไปOut
        DAT = Out; // ส่งค่าOut ให้กับขา DAT ของ TRW
        CLK_TRW24 ();
        Data = Data << 1; // เลื่อน Data ไปทางซ้าย 1 บิต
    }
}

void SetMode_TRW24( unsigned char Mode) //ฟังก์ชันกำหนดการส่ง หรือรับ
{
    //wait(20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CE = 0;
CS = 1;
Write_TRW24(0x8E); /* Reserved for testing */
Write_TRW24(0x08); /* Reserved for testing */
Write_TRW24(0x1C); /* Reserved for testing */

Write_TRW24(0x08); /* Length of Bit Ch 2 */
Write_TRW24(0x08); /* Length of Bit Ch 1 */

Write_TRW24(0xC0); /* Address 5 Byte Ch 2 */
Write_TRW24(0xAA);
Write_TRW24(0x55);
Write_TRW24(0xAA);
Write_TRW24(0x55);

Write_TRW24(0xAA); /* Address 5 Byte Ch 1 */
Write_TRW24(0x55);
Write_TRW24(0xAA);
Write_TRW24(0x55);
Write_TRW24(0xAA);

Write_TRW24(0xA3); /* Number of Address bit + CRC */
Write_TRW24(0x6F); /* RF Programming */
Write_TRW24(0x0A+Mode);

DAT = Mode;
DR1 = Mode;
CE = Mode;

CS = 0; // ปิดการ Set Mode
//Wait(20);
}

void Send_TRW24(unsigned char Data) // ฟังก์ชันส่งค่าออกจาก TRW
{
    Wait(100);
    CS = 0;
    CE = 1;
    Write_TRW24(0xAA); /* Address 5 Byte Ch 1 */ //เขียนaddress5byteก่อน
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
    Write_TRW24(0x55);
    Write_TRW24(0xAA);

    Write_TRW24(Data); /* Data 8 bit */ // แฉ้าตามด้วย data 8 bit
//    Wait(20);
    CLK = 0;
    CE = 0;
//    Wait(20);
}

unsigned char Read_TRW24(void) // ฟังก์ชันรับค่าเข้า TRW
{
    unsigned char i,Temp;
    bit Out; // กำหนดตัวแปรOutมีค่าเป็นbit 0 หรือ 1
    DAT = 1; // กำหนดให้ขา DAT เริ่มต้น ของ TRW มีค่าเป็น 1
    for (i=0;i<8;i++) // กำหนดLoop for ให้วนรับค่า 8 ครั้ง
    {
        Temp = Temp << 1; //เลื่อน Temp ไปทางซ้าย 1 บิต
        CLK = 1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Wait(1);
Out = DAT; // เก็บค่าจาก DAT ของ TRW ให้แก่ Out
if (Out) { Temp = Temp + 0x01; } // ถ้า Out เป็น 1 ให้ Temp บวก 1
CLK = 0;
Wait(1);
}
return(Temp); // ส่งค่า Temp คืนให้กับฟังก์ชัน
}

```

```

void sevice_serial() interrupt 4 // ฟังก์ชันอินเทอร์รัปต์รองรับค่าจากบารโค้ด
{

```

```

unsigned char i,arr[14];
if(RI)
{
    if(x==0)
    {
        arr[0] = SBUF; // เก็บค่าตัวแรกในการร้องขอคือ 'P', 'Q' หรือ 'B'
        x=1;
        RI=0;
        RI=1;
        RI=0;
    }
    else
    {
        y++;
        arr[y] = SBUF; // เก็บค่าจากเครื่องอ่านบาร์โค้ด
        RI=0;
        RI=1;
        RI=0;
        if(y==13) // เก็บค่าจนครบ 13 หลัก
        {
            Init_TRW24G();
            SetMode_TRW24(0); // เลือกโหมดส่ง
            for(i=0;i<14;i++)
            {
                Wait(1);
                Send_TRW24(arr[i]); // ส่งค่าอาร์เรย์ออก TRW
                Wait(1);
            }
            Init_TRW24G();
            SetMode_TRW24(1); // เลือกโหมดการรับ
            y=0; // กำหนดตัวเริ่มต้นให้เท่ากับ y ใหม่
            RI=1;
            RI=0;
        }
    }
}
}

```

```

void main(void)
{
    unsigned char i,j,arr1[32];
    P1 = 0X00; // ตั้งค่าแอดเดรสเริ่มต้น LCD ที่ตำแหน่ง 0x00
    TMOD=0x21; //Timer1 Mode2 for serial port
    SCON=0x50; //กำหนดการใช้serial port Tx และ Rx
    TH1=0xFD; //เซต 9600bps timer1
    TL1=0xFD;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RI=0; //เปิดการรับทางserial
TI=0; //เปิดการส่งทางserial

EA=1; // เปิดการอินเตอร์รัปต์
ES=1; // กำหนดให้พอร์ต serial มีการอินเตอร์รัปต์
TR1=1;//Start timer1

Init_TRW24G();
SetMode_TRW24(1); // เลือกโหมดรับ
while(1)
{
    i=0;
    j=0;
    while(j!=3)
    {
        while(!DR1);
        arr1[i] = Read_TRW24(); // เก็บค่าในตัวแปรอาร์เรย์
        if(arr1[i]=='*') // ถ้าข้อมูลเป็น '*' ให้เพิ่มค่า j
            {j++;}
        i++;
    }
    for(j=0;j<i;j++) // ส่งค่าออกพอร์ต serial
    {
        SBUF=arr1[j];
        while(-TI);
        TI=0;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุม AT89C4051 ผังรับ

```
//receiver.c
```

```
#include<reg2051.h>
#define CE P1_2 // pin 5      กำหนดพอร์ต 1.2ให้กับ ขา CE ของ TRW
#define CS P1_3 // pin 2      กำหนดพอร์ต 1.3ให้กับขา CS ของ TRW
#define DR1 P1_6 // pin 3     กำหนดพอร์ต 1.6 ให้กับขา DR1 ของ TRW
#define CLK P1_4 // pin 4     กำหนดพอร์ต 1.4 ให้กับขา CLK ของ TRW
#define DAT P1_5 // pin 1     กำหนดพอร์ต 1.5 ให้กับขา DATA ของ TRW
unsigned char x=0,y=0; //กำหนดตัวแปรเพื่อใช้ในอินเทอร์รัปต์
```

```
void Wait(unsigned char n) //ฟังก์ชันคิเลย์
```

```
{
    int i;
    while(n)
    {
        i = 0x01;
        while(i--);
        n--;
    }
}
```

```
void Init_TRW24G(void) //ฟังก์ชันกำหนดค่าเริ่มต้นของ TRW
```

```
{
//    Wait(10);
    CE = 0;
    CS = 0;
    CLK = 0;
    DAT = 0;
    DR1 = 0;
//    Wait(10);
}
```

```
void CLK_TRW24(void) //ฟังก์ชันกำหนดคิเล็กของ TRW
```

```
{
    CLK = 0;
    Wait(1);
    CLK = 1;
    Wait(1);
}
```

```
void Write_TRW24(unsigned char Data) //ฟังก์ชันเขียนค่าDataให้กับTRW
```

```
{
    unsigned char i;
    bit Out; // กำหนดตัวแปรOutมีค่าเป็นbit 0 หรือ 1
    for (i=0;i<8;i++) // วนเขียนค่าจนครบ 1 ไบต์
    {
        Out = Data & 0x80; //Data bit ชั่วชุด แขนกับ 1 เก็บค่าให้Out
        DAT = Out; // ส่งค่าOut ให้กับขา DAT ของ TRW
        CLK_TRW24();
        Data = Data << 1; // เอ็น Data ไปทางซ้าย 1 บิต
    }
}
```

```
void Send_TRW24(unsigned char Data) // ฟังก์ชันส่งค่าออกจาก TRW
```

```
{
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```

Wait(100);
CS = 0;
CE = 1;
Write_TRW24(0xAA); /*Address 5 Byte Ch1*/ //เขียนaddress5byteก่อน
Write_TRW24(0x55);
Write_TRW24(0xAA);
Write_TRW24(0x55);
Write_TRW24(0xAA);

Write_TRW24(Data); /* Data 8 bit */ // แล้วตามด้วย data 8 bit
// Wait(20);
CLK = 0;
CE = 0;
// Wait(20);
}

unsigned char Read_TRW24(void) // ฟังก์ชันรับค่าเข้า TRW
{
    unsigned char i,Temp;
    bit Out; // กำหนดตัวแปรOutมีค่าเป็นbit 0 หรือ 1
    DAT = 1; // กำหนดให้ขา DAT เริ่มต้น ของ TRW มีค่าเป็น 1
    for (i=0;i<8;i++) // กำหนดLoop for ให้วนรับค่า 8 ครั้ง
    {
        Temp = Temp << 1; // เลื่อน Temp ไปทางซ้าย 1 บิต
        CLK = 1;
        Wait(1);
        Out = DAT; // เก็บค่าจากขา DAT ของ TRW ให้แก่ Out
        if (Out) { Temp = Temp + 0x01; } // ถ้าOut เป็น 1 ให้Temp บวก 1
        CLK = 0;
        Wait(1);
    }
    return(Temp); // ส่งค่า Temp คืนให้กับฟังก์ชัน
}

void SetMode_TRW24( unsigned char Mode) // ฟังก์ชันกำหนดการส่งหรือรับ
{
    // Wait(20);
    CE = 0;
    CS = 1; // เปิดการ Set Mode ของTRW
    Write_TRW24(0x8E); /* Reserved for testing */
    Write_TRW24(0x08); /* Reserved for testing */
    Write_TRW24(0x1C); /* Reserved for testing */

    Write_TRW24(0x08); /* Length of Bit Ch 2 */
    Write_TRW24(0x08); /* Length of Bit Ch 1 */

    Write_TRW24(0xC0); /* Address 5 Byte Ch 2 */
    Write_TRW24(0xAA);
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
    Write_TRW24(0x55);

    Write_TRW24(0xAA); /* Address 5 Byte Ch 1 */
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Write_TRW24(0xA3); /* Number of Address bit + CRC */
Write_TRW24(0x6F); /* RF Programming */
Write_TRW24(0x0A+Mode);

DAT = Mode;
DR1 = Mode;
CE = Mode;

CS = 0; //ปิดการ Set Mode
//Wait(20);
}

void sevice_serial() interrupt 4 // ฟังก์ชันอินเทอร์รัปต์
{
    unsigned char i,arr[32]; // กำหนดตัวแปรอาร์เรย์ขนาด 32 ไบต์
    if(RI) // ถ้า RI set ให้ทำในรูป
    {
        arr[x]=SBUF; // ส่งค่า SBUF ให้แก่อาร์เรย์
        RI=0; // กำหนดเพื่อรอรับค่าผ่านพอร์ต Serial ใหม่
        RI=1;
        RI=0; // ปิดการรับค่าแล้วเปิดการรับค่าใหม่
        if (arr[x]=='*') // ถ้าค่าอาร์เรย์มีค่าเท่ากับ '*' ให้เพิ่มค่า y ขึ้น 1
            {y++;}
        x++;
        if (y==3) // ถ้า y มีค่าเท่ากับ 3
        {
            Init_TRW24G();
            SetMode_TRW24(0); // เลือกโหมดการส่ง
            for(i=0;i<x;i++) // ส่งค่าข้อมูลในอาร์เรย์ทั้งหมด
            {
                Wait(1);
                Send_TRW24(arr[i]); // ส่งค่าอาร์เรย์ออก TRW
                Wait(1);
            }
            Init_TRW24G();
            SetMode_TRW24(1); // เลือกโหมดรับข้อมูล
            x=0; // กำหนดค่าเริ่มต้นให้ x และ y
            y=0;
            RI=0;
            RI=1;
            RI=0;
        }
    }
}

void main(void)
{
    unsigned char i,arr[14]; // กำหนดอาร์เรย์ขนาด 14 ไบต์
    P1 = 0X00; // ตั้งค่าแอดเดรสเริ่มต้น LCD ที่ตำแหน่ง 0x00
    TMOD=0x21; //Timer1 Mode2 for serial port
    SCON=0x50; //กำหนดการใช้serial port Tx และ Rx
    TH1=0xFD; //เซต 9600 bps timer1
    TL1=0xFD;
    RI=0; //เปิดการรับทางserial
    TI=0; //เปิดการส่งทางserial

    EA=1; // เปิดการอินเทอร์รัปต์
    ES=1; // กำหนดให้พอร์ต serial มีการอินเทอร์รัปต์
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TR1=1;//Start timer1
Init_TRW24G();
SetMode_TRW24(1); // เลือกโหมดรับ
while(1)
{
    for(i=0;i<14;i++) // ดูปรับค่าจากฟังก์ชันบาร์โค้ด
    {
        while(!DR1); // วนรับค่าจน DR1 เป็น 1
        arr[i] = Read_TRW24(); // เก็บค่าในตัวแปรอาร์เรย์
    }
    for(i=0;i<14;i++)
    {
        SBUF=arr[i]; // ส่งค่าที่เก็บได้ผ่านทางพอร์ต serial ให้ฝั่งเซิร์ฟเวอร์
        while(~TI);
        TI=0;
    }
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุม AT89C51 (LCD)

```
#include<reg51.h>
#include<stdio.h>
sbit e = P3^6;
sbit rs = P3^7;
sbit right = P3^2;
sbit left = P3^3;
sbit enter = P3^4;
unsigned char tx1[] = "B:";
unsigned char tx2[] = "P:";
unsigned char tx3[] = "Q:";
unsigned char textall[] = "Select_menu:>Price Quantity Both";//
unsigned char textp[] = " !!!Please!!! Scan Barcode ";//
unsigned char i,n,m;
void delay(int tick) // ฟังก์ชัน delay
{
    unsigned int i;
    unsigned char j;
    for(i=0;i<tick;i++)
        for(j=0;j<100;j++); //220
}
void lcd_command(unsigned char com) // ฟังก์ชันส่งคำสั่งควบคุม LCD
{
    rs = 0;
    e = 1;
    P0 = com;
    delay(3);
    e = 0;
    delay(3);
}
void lcd_text(unsigned char text) // ฟังก์ชันส่งข้อความแสดงที่ LCD
{
    rs = 1;
    e = 1;
    P0 = text;
    delay(3);
    e = 0;
    delay(3);
}
void lcd_init() // ฟังก์ชันกำหนดค่าเริ่มต้นให้ LCD
{
    delay(50);
    lcd_command(0x38);
    lcd_command(0x38);
    lcd_command(0x38);
    lcd_command(0x0C);
    lcd_command(0x01);
}
unsigned char getdata(void) // ฟังก์ชันรับค่าจากพอร์ต serial
{
    unsigned char c;
    RI=0;
    RI=1;
    RI=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(~RI);
c=SBUF;
RI=0;
RI=1;
RI=0;
return c;
}

void showlcd(void) // ฟังก์ชันเริ่มการทำงาน LCD ให้showการเลือกโหมด
{
    unsigned char j;
    lcd_command(0x80); // ให้เริ่มคั่นเขียนLCD บรรทัดแรกซ้ายสุด
    for(j=0;j<16;j++)
        lcd_text(textall[j]); //เขียน Select_menu:>Pri
    lcd_command(0xC0); //ให้เริ่มคั่นเขียนLCD บรรทัดที่2ซ้ายสุด
    for(j=16;j<32;j++)
        lcd_text(textall[j]); //เขียน ce Quantity Both
}

void pushswitch(void) //ฟังก์ชันการกดสวิทช์ซ้าย ขวา เ็นเตอร์
{
    i=12; // กำหนดค่าให้มีค่า=12 เพื่อให้อยู่ในช่วง 12-27
    m='P'; //กำหนดค่าเริ่มต้นให้กับ m
    right=1;
    left=1;
    enter=1;
    while(enter!=0)
    {
        if (enter!=0) //ถ้ามีการกด สวิทช์ enter ก็จะส่งค่า m = 'P'
        {
            if(right==0) //ถ้ามีการกด ขวา ก็จะแสดง > เลื่อนไปทางขวา
            {
                delay(50); //คิลด์การกด ขวา 1ที
                if(right==0)
                {
                    do
                    {
                        n=0;
                        if(textall[i]=='>') //ดัชนีเครื่องหมาย '>'
                            textall[i]=' '; //ให้เปลี่ยนเป็น ' '
                        if(i<27)
                            i++;
                        if(textall[i]==' ') //ถ้าอยู่ทางขวาสุด ' '
                        {
                            n=1;
                            textall[i]='>'; //ให้เปลี่ยนเป็น '>'
                            m=textall[i+1]; //ให้ค่าถัดไปเก็บ
                        }
                    } while(n!=1); //อุปเปลี่ยน '>' ไปทางขวาเมื่อเจอช่องว่าง ' '
                    showlcd(); //แสดงการเลื่อน '>'
                    delay(800); //คิลด์เพื่อไม่ให้มีการกดค้างเร็วเกินไป
                    right=1; //เซตค่าเริ่มต้นการกดขวาใหม่
                }
            }
            if(left==0) //ถ้ามีการกดซ้าย ก็จะแสดง > เลื่อนไปทางซ้าย

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        delay(50); //ดีเลย์การกด ซ้าย 1ที
        if (left==0)
        {
            do
            {
                n=0;
                if(textall[i]=='>') //ถ้าเจอเครื่องหมาย '>'
                textall[i]=' '; //ให้เปลี่ยนเป็น ' '
                if(i>12)
                i--;
                if(textall[i]==' ') //ถ้าอยู่ทางซ้ายสุด ' '
                {
                    n=1;
                    textall[i]='>'; //ให้เปลี่ยนเป็น '>'
                    m=textall[i+1]; //ให้ค่าถัดไปแก่ m
                }
            }
            while(n!=1); //ดูเปลี่ยน '>' ไปทางซ้ายเมื่อเจอช่องว่าง ' '
            showlcd(); //แสดงLCDว่ากรรมเดือน '>' ไปทางซ้าย
            delay(800); //ดีเลย์เพื่อไม่ให้มีการกดค้างเร็วเกินไป
            left=1; //เซตค่าเริ่มต้นการกดซ้ายใหม่
        }
    }
    delay(150); //ดีเลย์การกด enter
}
}
void main(void)
{
    unsigned char co[16],pri[16],qua[8];
    P0 = 0X00; //ตั้งค่าแอดเดรสเริ่มต้น LCD ที่ตำแหน่ง 0x00
    TMOD=0x21; //Timer1 Mode2 for serial port
    SCON=0x50; //กำหนดการใช้serial port Tx และ Rx
    TH1=0xFD; //เซต 9600bps timer1
    TL1=0xFD;
    TR1=1; //Start timer1
    lcd_init(); //กำหนดค่าเริ่มต้นให้ LCD
    showlcd(); //แสดงการเลือก โหนด ราคา จำนวน หรือทั้งสอง
    pushswitch(); //กดเลือกซ้ายขวา เ็นเตอร์
    lcd_command(0x01); //เคลียร์หน้าจอ
    for(i=0;i<16;i++) //แสดง !!!Please!!! Scan Barcode
        lcd_text(textp[i]);
    lcd_command(0xC0);
    for(i=16;i<32;i++)
        lcd_text(textp[i]);

    TI=0; //เปิดการส่งทาง serial
    SBUF=m; //ตั้งค่าการเลือกโหนดให้กับไมโคร 20ขา
    while(~TI);
    TI=1; //ปิดการส่งทาง TX
while(1)
{
    if(m=='P') //ถ้ามีการเลือกโหนด Price ก็จะมีการรอรับค่า จากไมโคร 20ขาแล้วนำไปแสดงทางLCD
    {
        i=-1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do
{
    i++;
    co[i] = getdata();
}
while(co[i]!='*');

i=-1;
do
{
    i++;
    pri[i] = getdata();
}
while(pri[i]!='*');
lcd_command(0x01);
lcd_command(0x80);
i=0;
while(co[i]!='*')
{
    lcd_text(co[i]);
    i++;
}
lcd_command(0xC0);
for(i=0;i<2;i++)
    lcd_text(tx2[i]);
i=0;
while(pri[i]!='*')
{
    lcd_text(pri[i]);
    i++;
}
}

if(m=='Q') //ถ้ามีการเลือกโหมดQuantity ก็จะมีการรับค่าจากไมโคร20ชนวนนำไปแสดงทางLCD
{
    i=-1;
    do
    {
        i++;
        co[i] = getdata();
    }
    while(co[i]!='*');
    i=-1;
    do
    {
        i++;
        qua[i] = getdata();
    }
    while(qua[i]!='*');
    lcd_command(0x01);
    lcd_command(0x80);
    i=0;
    while(co[i]!='*')
    {
        lcd_text(co[i]);
        i++;
    }
    lcd_command(0xC0);
    for(i=0;i<2;i++)
        lcd_text(tx3[i]);
    i=0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        while(qua[i]!='*')
        {
            lcd_text(qua[i]);
            i++;
        }
    }

    if(m=='B') //ถ้ามีการเลือกโหมดBoth ก็จะมีการรอรับค่า จากไมโคร20จนเสี้นำไปแสดงทางLCD
    {
        i=-1;
        do
        {
            i++;
            co[i] = getdata();
        }
        while(co[i]!='*');
        i=-1;
        do
        {
            i++;
            pri[i] = getdata();
        }
        while(pri[i]!='*');
        i=-1;
        do
        {
            i++;
            qua[i] = getdata();
        }
        while(qua[i]!='*');
        lcd_command(0x01);
        lcd_command(0x80);
        i=0;
        while(co[i]!='*')
        {
            lcd_text(co[i]);
            i++;
        }
        lcd_command(0xC0);
        for(i=0;i<2;i++)
            lcd_text(tx2[i]);
        i=0;
        while(pri[i]!='*')
        {
            lcd_text(pri[i]);
            i++;
        }
        lcd_text(' ');
        for(i=0;i<2;i++)
            lcd_text(tx3[i]);
        i=0;
        while(qua[i]!='*')
        {
            lcd_text(qua[i]);
            i++;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้