สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบศูนย์อาหารโดยใช้ RFID

FOOD COURT SYSTEM USING RFID

โดย

นายกมลาสน์ วรรณคง

นายจุมพล สุขวัฒนานุกิจ

วพ
ก1388
2550

เลขหมู่.....
เลขทะเบียน 83302
วัน,เดือน,ปี 11 ส.ค. 2551

b 11967507
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบศูนย์อาหารโดยใช้ RFID

FOOD COURT SYSTEM USING RFID

โดย

นายกมลาสน์ วรรณคง 47010012

นายจุมพล สุขวัฒนานุกิจ 47010113

อาจารย์ที่ปรึกษา

รศ.ดร. ปราโมทย์ วาดเขียน

ผศ.ดร. จีรสุดา โกษียาภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบศูนย์อาหารโดยใช้ RFID**

FOOD COURT SYSTEM USING RFID

ผู้จัดทำ

1.นายกมลาสน์ วรรณคง 47010012

2. นายจุมพล สุขวัฒนานุกิจ 47010113

..... อาจารย์ที่ปรึกษา

(รศ.ดร. ปราโมทย์ วาดเขียน)

..... อาจารย์ที่ปรึกษา

(ผศ.ดร. จีรสุดา โกษียาภรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบศูนย์อาหารโดยใช้RFID

FOOD COURT SYSTEM USING RFID

โดย นายกมลาสน์ วรรณคง 47010012
นายจุมพล สุขวัฒนานุกิจ 47010113

อาจารย์ที่ปรึกษา รศ.ดร.ปราโมทย์ วาดเขียน
ผศ.ดร. จีรสุดา โกษียาภรณ์

บทคัดย่อ

โครงงานนี้ได้เสนอระบบศูนย์อาหารโดยใช้ RFID และโมดูล TRW 2.4 GHz จะใช้บัตร RFID ในการซื้ออาหาร โดยจะมีเครื่องอ่านสัญญาณติดตั้งไว้ที่ร้านแต่ละร้าน เครื่องอ่านสัญญาณจะอ่านรหัสของบัตร RFID และทำการส่งรหัส ยังคอมพิวเตอร์ศูนย์กลาง จากนั้นคอมพิวเตอร์ศูนย์กลางก็จะส่งข้อมูลจำนวนเงินที่ลูกค้าซื้อไว้ในบัตร RFID นั้นกลับไป ยังเครื่องอ่านสัญญาณ เมื่อทางร้านค้าระบุ จำนวนเงินในการซื้ออาหาร ก็จะส่งข้อมูลกลับมายังคอมพิวเตอร์ศูนย์กลาง อีกครั้ง เพื่อเก็บบันทึกข้อมูลลงในฐานข้อมูล โดยการรับส่งข้อมูลระหว่างกันนี้จะสื่อสารแบบไร้สายโดยใช้โมดูล TRW 2.4 GHz ซึ่งควบคุมด้วยไมโครคอนเทรลเลอร์

Abstract

This project presents about food court system using RFID and TRW 2.4 GHz. In this system, each restaurant will have its own RFID's card reader. When the customers buy the food using RFID card, the card reader will read the code of the card and then send to the central computer. After that, the central computer will send information of the customer's money to the card reader. When the card reader receiver the money's information, seller will input the price of the food then the card reader will send the information to the central computer to calculate the total money. Finally, central computer will send the information back to card reader. The card reader will write the remaining money into the card. The wireless communication, in this project uses TRW 2.4 GHz module controlling by microcontroller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

สารบัญ II

สารบัญรูป V

สารบัญตาราง VII

บทที่

1 บทนำ 1

1.1 ความสำคัญและที่มา 1

1.2 วัตถุประสงค์ของโครงงาน 1

2 ทฤษฎีและหลักการ 2

2.1 หลักการทำงานของ RFID 2

2.1.1 ฉลากอิเล็กทรอนิกส์ (Transponder) 2

2.1.1.1 ชนิดของแทกส์ 2

2.1.1.2 เทคโนโลยีของหน่วยความจำ 2

2.1.2 เครื่องอ่านสัญญาณ (Reader) 3

2.1.3 ความถี่ ของคลื่นพาห์ 3

2.1.4 หลักการทำงานของระบบ RFID 3

2.1.5 ประเภทของ ระบบ RFID 5

2.1.5.1 RFID ที่จำแนกโดยขนาดของหน่วยความจำ 5

2.1.5.2 RFID ที่จำแนกโดยลักษณะการคล้องของสัญญาณ 6

2.1.5.3 RFID ที่จำแนกตามความสามารถของระบบ 8

2.1.6 กระบวนการรับข้อมูลของ RFID 8

2.1.6.1 กระบวนการ Half Duplex (HDX) 8

2.1.6.2 กระบวนการ Full Duplex (FDX) 8

2.1.7 พลังงานที่จ่ายให้แทกส์ 9

2.1.8 เทคโนโลยีการเข้ารหัสและถอดรหัส 11

2.2 คู่มือการใช้งานเครื่องรับ-ส่ง TRW-2.4GHz 14

2.2.1 โหมดการทำงานของ TRW-2.4GHz 14

2.2.2 ShockBurst 14

2.3 ไมโครคอนโทรลเลอร์ MCS-51 24

2.3.1 คุณสมบัติพื้นฐานของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.3.2 พอร์ทของ MCS-51 24
2.3.3 ประเภทของหน่วยความจำ 27
2.3.4 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register: SFR) 28
2.3.5 การรับส่งข้อมูลอนุกรม (UART) 29
2.4 แอลซีดี 32
2.4.1 รายละเอียดเกี่ยวกับโมดูล LCD 32
2.4.2 โครงสร้างภายในของตัวควบคุมโมดูล LCD 32
2.4.3 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD 36
2.5 Matrix – Switch 37
2.6 Visual Studio .NET 38
2.6.1 ความสามารถของ Visual C# 39
2.6.1.1 สร้างแอพพลิเคชันสำหรับ Windows 39
2.6.1.2 สร้างระบบงานด้านฐานข้อมูล 39
2.6.1.3 สร้างแอพพลิเคชันรองรับ .NET Web Service 39
3 การคำนวณและการสร้าง 40
3.1 บล็อกไดอะแกรม 40
3.2 ไมโครคอนโทรลเลอร์ 41
3.3 ส่วนของเครื่อง Server 46
3.4 ฐานข้อมูล 48
4 การทดลองและผลการทดลอง 49
4.1 ทดลองอ่านค่าบัตร RFID 49
4.2 การทดลองอ่านค่ารหัสบัตร RFID แล้วส่งผ่านโมดูล TRW-2.4GHz 49
4.3 การทดลองรับค่าสัญญาณที่ส่งมาจากด้านส่ง 51
4.4 การทดลองอ่านค่ารหัสบัตร RFID พร้อมทำการระบุจำนวนเงินของบัตร ไว้ในส่วนของ Database 52
4.5 การทดลองอ่านค่ารหัสบัตร RFID ส่งผ่านโมดูล TRW-2.4GHz แล้วส่งรหัสที่ได้ เข้าสู่ในส่วนของ Database 52
4.6 การทดลองทำการคืนเงินลูกค้าทางด้านฝั่งเซิร์ฟเวอร์ 54
5 บทวิจารณ์และสรุปผล 55
5.1 บทวิจารณ์ 55
5.1.1 ส่วนของเครื่องอ่าน RFID 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

5.1.2 ส่วนของไมโครคอนโทรลเลอร์ 55
5.1.3 โมดูล TRW-2.4 GHz 55
5.1.4 ส่วนของฐานข้อมูล 55
5.2 บทสรุป 55
บรรณานุกรม 56
ภาคผนวก 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 วงจรเทียบเคียงของระบบ RFID 4
รูปที่ 2.2 การสื่อสารระหว่างเครื่องอ่านกับแทกส์ 5
รูปที่ 2.3 แทกส์ของระบบ EAS จะมีเฉพาะเสาอากาศเพียงอย่างเดียว 5
รูปที่ 2.4 เสาอากาศของระบบ EAS 6
รูปที่ 2.5 วงจรเทียบเคียงของ Close-Coupling 6
รูปที่ 2.6 วงจรเทียบเคียงของ Remote Coupling หรือ Inductive Coupling 7
รูปที่ 2.7 วงจรเทียบเคียงของ Long Range 7
รูปที่ 2.8 แสดง Full Duplex, Half Duplex และ Sequential System บนเวลา 9
รูปที่ 2.9 แสดงพลังงานการเหนี่ยวนำที่เกิดจากการคล้องสัญญาณของแทกส์ 9
รูปที่ 2.10 การไหลของสัญญาณและข้อมูลในระบบการสื่อสารแบบดิจิตอล 11
รูปที่ 2.11 รหัสสัญญาณ NRZ 11
รูปที่ 2.12 รหัสสัญญาณ Biphase 12
รูปที่ 2.13 แสดงการผสมข้อมูลแบบ BinaryASK 12
รูปที่ 2.14 แสดงการผสมข้อมูลแบบ Binary FSK 13
รูปที่ 2.15 แสดงการผสมข้อมูลแบบ Binary PSK 13
รูปที่ 2.16 เครื่องส่งสัญญาณ TRW 2.4 GHz 14
รูปที่ 2.17 แสดงการทำงานของ ShockBurst 15
รูปที่ 2.18 กระแสที่ใช้ในการทำงานขณะใช้ ShockBurst และขณะที่ไม่ได้ใช้ 15
รูปที่ 2.19 การรับข้อมูล 2 ช่องสัญญาณของ TRW-2.4GHz 17
รูปที่ 2.20 การรับข้อมูล 2 ช่องสัญญาณพร้อมกัน 17
รูปที่ 2.21 ส่วนประกอบของ Data package 18
รูปที่ 2.22 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 25
รูปที่ 2.23 แสดงการต่อสัญญาณนาฬิกาที่ขา XTAL1 และ XTAL2 26
รูปที่ 2.24 การจัดพื้นที่หน่วยความจำของไมโครคอนโทรลเลอร์ 27
รูปที่ 2.25 แสดงลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล 27
รูปที่ 2.26 รูปแบบการรับส่งข้อมูลในโหมด 0 31
รูปที่ 2.27 รูปแบบการรับส่งข้อมูลในโหมด 1 31
รูปที่ 2.28 รูปแบบการรับส่งข้อมูลในโหมด 2 31
รูปที่ 2.29 รูปแบบการรับส่งข้อมูลในโหมด 3 32
รูปที่ 2.30 รูปร่างและการจัดขาโมดูลแอลซีดี 33
รูปที่ 2.31 วงจรสวิตช์แบบต่อเข้ากับไฟเลี้ยงและกราวด์ 37

เอกสารนี้มีไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 2.32 วงจรสวิตช์แบบเมตริกซ์หรือคีย์แพด 37
รูปที่ 2.33 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51 38
รูปที่ 3.1 บล็อกไดอะแกรมของปริญญานิพนธ์ 40
รูปที่ 3.2 แสดงวงจรฝั่งร้านค้า 41
รูปที่ 3.3 แสดงวงจรฝั่งเซิร์ฟเวอร์ 42
รูปที่ 3.4 แสดงวงจรภาคจ่ายไฟ (Power Supply) 42
รูปที่ 3.5 แสดงโฟลวชาร์ทการทำงานในไมโครคอนโทรลเลอร์ของฝั่งร้านค้าและเซิร์ฟเวอร์ 43
รูปที่ 3.6 แสดง โฟลวชาร์ทการทำงานของ TRW 2.4 GHz ทางฝั่งร้านค้า 44
รูปที่ 3.7 แสดง โฟลวชาร์ทการทำงานของ TRW 2.4 GHz ทางฝั่งเครื่องเซิร์ฟเวอร์ 45
รูปที่ 3.8 แสดง โฟลวชาร์ทการทำงานในส่วนของเครื่องเซิร์ฟเวอร์ 48
รูปที่ 3.9 แสดงฐานข้อมูล 47
รูปที่ 4.1 สัญญาณของรหัสบัตร RFID ที่เครื่องอ่านอ่านได้ 49
รูปที่ 4.2 สัญญาณ Clock และสัญญาณของข้อมูลในไบต์ที่ 1 (STX) 50
รูปที่ 4.3 สัญญาณ Clock และสัญญาณข้อมูลในไบต์ที่สอง 50
รูปที่ 4.4 สัญญาณ Clock และสัญญาณของข้อมูลในไบต์ที่ 13 (EXT) 51
รูปที่ 4.5 สัญญาณ Clock และสัญญาณของข้อมูลทางด้านรับ 51
รูปที่ 4.6 รหัสบัตร RFID ที่ทางด้านรับอ่านค่าได้ 52
รูปที่ 4.7 รหัสบัตรที่รับเข้ามาและจำนวนเงินที่ลูกค้าต้องการใช้ 52
รูปที่ 4.8 รหัสบัตร ราคาบัตร จำนวนเงินที่กดเข้ามาและเงินคงเหลือของบัตร แสดงที่จอแสดงผล 53
รูปที่ 4.9 จำนวนเงินคงเหลือที่ส่งกลับมาแสดงที่จอแอลซีดีทางด้านส่ง 54
รูปที่ 4.10 แสดงจำนวนเงินคงเหลือของบัตรคืนแก่ลูกค้า 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 ย่านความถี่ต่าง ๆ ของระบบ RFID และการใช้งาน 4
ตารางที่ 2.2 แสดงการใช้พลังงานของหลายๆ RFID-ASIC 10
ตารางที่ 2.3 แสดง Configuration Word 17
ตารางที่ 2.4 แสดง Configuration data word 19
ตารางที่ 2.5 แสดงการตั้งค่า PLL 20
ตารางที่ 2.6 แสดงจำนวนบิตของ payload 20
ตารางที่ 2.7 Address ของตัวรับตัวที่ 2 และตัวรับตัวที่ 1 21
ตารางที่ 2.8 แสดงจำนวนบิตสำรองของ RX Address + CRC 21
ตารางที่ 2.9 แสดงการตั้งค่าการทำงานของ RF 21
ตารางที่ 2.10 แสดงการตั้งค่าของ Crystal 22
ตารางที่ 2.11 แสดงการตั้งค่าของ RF output power 22
ตารางที่ 2.12 แสดงการตั้งค่าช่องสัญญาณความถี่ และ RX/TX 22
ตารางที่ 2.13 อธิบาย Data package 23
ตารางที่ 2.14 แสดงบิตภายใน PCON 29
ตารางที่ 2.15 แสดงบิตภายใน SCON 30
ตารางที่ 2.16 แสดงการเลือกโหมดการใช้งานพอร์ทอนุกรม 30
ตารางที่ 2.17 รายละเอียดรูปแบบโหมดการป้อนข้อมูลของแอลซีดี 34
ตารางที่ 2.18 รายละเอียดรูปแบบคำสั่งควบคุมการแสดงผล 34
ตารางที่ 2.19 รายละเอียดรูปแบบคำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร 35
ตารางที่ 2.20 รายละเอียดรูปแบบคำสั่งกำหนดฟังก์ชันการทำงาน 35
ตารางที่ 2.21 รายละเอียดรูปแบบคำสั่งอ่านค่าแฟลก BUSY และแอดเดรส 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันนี้ระบบบ่งชี้อัตโนมัติ(Automatic Identification หรือ Auto ID) ถูกนำมาใช้งานและพัฒนาไปอย่างมากทั้งในภาคอุตสาหกรรม ลอจิสติกส์ กระบวนการผลิตการขนถ่ายวัตถุดิบ ฯลฯ ระบบ Auto ID จะนำมาใช้ในการจัดเก็บข้อมูล ระบุสถานะของคน สัตว์ สิ่งของ เป็นต้น

ระบบ Auto ID ที่เป็นที่รู้จักและใช้งานกันแพร่หลายที่สุด คือ ระบบบาร์โค้ด (Barcode System) ซึ่งสามารถพบเห็นได้ทั่วไปในชีวิตประจำวัน เช่น บันทึกห่อสินค้า หนังสือหรือบนตัวสินค้า เนื่องจากมีต้นทุนต่อหน่วยที่ต่ำ ง่ายต่อการใช้งาน จึงเป็นเหตุผลให้บาร์โค้ดถูกนำมาใช้งานมากที่สุด แต่บาร์โค้ดก็มีข้อกำจัดหลายประการ ได้แก่ จัดเก็บข้อมูลได้จำกัดเกิดความเสียหายหรือมีปัญหาระหว่างการอ่านได้ง่าย ระบบ Auto ID ที่เรารู้จักรองลงมาจากบาร์โค้ดก็คือ ระบบสมาร์ทการ์ด (Smart Card System) เป็นระบบที่กำลังมีบทบาทอย่างรวดเร็ว ในปัจจุบันเราจะพบระบบสมาร์ทการ์ดในรูปแบบของบัตรต่างๆ เช่น บัตรชมภาพยนตร์ ซิมการ์ดของโทรศัพท์เคลื่อนที่ และบัตรสมาชิกตามคลับต่างๆซึ่งบัตรประเภทนี้มีส่วนของแถบแม่เหล็กหรือไมโครชิพในการอ่านและเขียนข้อมูล มีข้อดีคือ สามารถเก็บข้อมูลได้มากและปลอดภัย แต่เนื่องจากส่วนประกอบของบัตรเป็นแถบแม่เหล็ก ซึ่งวิธีการอ่านข้อมูลของระบบสมาร์ทการ์ดจะต้องใช้วิธีสัมผัส อาจส่งผลทำให้เกิดการสึกหรอของเครื่องอ่านเมื่อใช้ไปนานๆ ระบบ Auto ID อีกชนิดหนึ่งที่เราจะกล่าวถึงนี้เป็นระบบที่ขจัดข้อเสียของทั้งระบบบาร์โค้ดและระบบสมาร์ทการ์ด เราเรียกระบบนี้ว่า RFID (Radio Frequency Identification) ซึ่งเป็นระบบ Auto ID ที่ใช้คลื่นแม่เหล็กไฟฟ้าในช่วงคลื่นความถี่วิทยุ เป็นพาหะในการสื่อสารข้อมูล

RFID คือ ระบบบ่งชี้เฉพาะอัตโนมัติ (Automatic Identification) แบบไร้สัมผัส ซึ่งประกอบด้วยส่วนประกอบหลัก 2 ส่วนคือ 1.เครื่องอ่านสัญญาณ (reader) 2.ตัวบัตร (tag) ซึ่งระบบบ่งชี้เฉพาะอัตโนมัติจะใช้ความถี่คลื่นเป็นคลื่นพาห์ในการติดต่อสื่อสารระหว่างกัน เทคโนโลยีนี้จะช่วยอำนวยความสะดวกในการที่จะบ่งบอกความแตกต่างของข้อมูล เทคโนโลยี RFID มีข้อดีเช่น สามารถอ่านและเขียนข้อมูลได้โดยไม่ต้องสัมผัส สื่อสารได้ทุกทิศทาง และ ทนต่อการใช้งาน เป็นต้น มีการประยุกต์ใช้งานต่างๆเช่น บัตรเข้าออกสำนักงาน บัตรจอดรถ และระบบห้องสมุดอัตโนมัติ ฯลฯ

1.2 วัตถุประสงค์ของโครงงาน

1. ศึกษาการส่งผ่านข้อมูลระหว่างแทกส์กับเครื่องอ่านสัญญาณ
2. ศึกษาการสร้างระบบฐานข้อมูลเพื่อเชื่อมต่อฐานข้อมูลเข้าสู่ระบบศูนย์อาหาร
3. เพื่อนำไปประยุกต์ในการพัฒนาระบบศูนย์อาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 หลักการทำงานของ RFID

โดยทั่วไประบบ RFID จะประกอบไปด้วย 2 ส่วนสำคัญ คือ

2.1.1 ฉลากอิเล็กทรอนิกส์ (Transponder) หรือเรียกง่าย ๆ ว่า แทกส์

จะเป็นส่วนของการเก็บข้อมูลและจะถูกติดตั้งอยู่กับวัตถุที่เราต้องการบ่งชี้ (Identify) แทกส์จะประกอบไปด้วยเสาอากาศทำหน้าที่คล้องสัญญาณที่ส่งมาจาก เครื่องอ่านและส่วนของไมโครชิพ

2.1.1.1 ชนิดของแทกส์

แทกส์ที่มีการใช้งานอยู่นั้นจะมีอยู่ 2 ชนิดใหญ่ๆ โดยแต่ละชนิดก็จะมีความแตกต่างกันในแง่ของการใช้งาน ราคาโครงสร้างและหลักการทำงานอยู่ ซึ่งจะขอกล่าวและอธิบายแยกเป็นหัวข้อดังนี้

1. แทกส์ของ RFID แบบพาสซีฟ (Passive RFID Tags)

แทกส์ชนิดนี้ไม่จำเป็นต้องรับแหล่งจ่ายไฟใดๆ เพราะมีวงจรกำเนิดไฟฟ้าเหนี่ยวนำขนาดเล็กเป็นแหล่งจ่ายไฟในตัวอยู่แล้ว ระยะการสื่อสารข้อมูลที่ทำได้สูงสุด 1.5 เมตร มีหน่วยความจำขนาดเล็ก (ทั่วไปประมาณ 32-128 บิต) มีขนาดเล็กและน้ำหนักเบา ราคาต่อหน่วยต่ำ ไอซีของแทกส์ชนิดพาสซีฟที่มีการผลิตออกมาจะมีทั้งขนาดและรูปร่างเป็นได้ตั้งแต่แท่งหรือแผ่นขนาดเล็กจนแทบไม่สามารถมองเห็นไปจนถึงขนาดใหญ่มากๆ ซึ่งต่างก็มีความเหมาะสมกับชนิดงานที่แตกต่างกัน โดยทั่วไปโครงสร้างภายในส่วนที่เป็นไอซีของแทกส์นั้นก็จะประกอบด้วย 3 ส่วนหลัก ๆ ได้แก่ ส่วนของหน่วยความจำ, ส่วนควบคุมภาคลอจิก และส่วนของควบคุมการทำงานของภาครับส่งสัญญาณวิทยุ

2. แทกส์ของ RFID แบบแอกทีฟ (Active RFID Tags)

แทกส์ชนิดนี้จะใช้แหล่งจ่ายไฟจากแบตเตอรี่ขนาดเล็ก มีหน่วยความจำภายในขนาดใหญ่ได้ถึง 1 เมกกะไบต์ มีระยะการสื่อสารข้อมูลที่ทำได้สูงสุดถึง 6 เมตร แม้ว่าแทกส์ชนิดนี้จะมีข้อดีอยู่หลายข้อแต่ก็มีข้อเสียอยู่ด้วยเหมือนกัน เช่น มีราคาต่อหน่วยแพง มีขนาดค่อนข้างใหญ่ และมีระยะเวลาในการทำงานที่จำกัด นอกจากการแบ่งจากชนิดที่ว่ามาแล้ว แทกส์ ก็ยังถูกแบ่งประเภทจากรูปแบบในการใช้งานได้แบบ 3 แบบ คือ แบบที่สามารถถูกอ่านและเขียนข้อมูลได้อย่างอิสระ (Read-write), แบบเขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้อย่างอิสระ (Write-Once, Read-Many หรือ WORM) และแบบอ่านได้เพียงอย่างเดียว (Read-Only) ด้วย

2.1.1.2 เทคโนโลยีของหน่วยความจำ

1. RAM (Random Access Memory)

เป็นหน่วยความจำที่ใช้เก็บข้อมูลชั่วคราว ข้อมูลจะสูญหายเมื่อไม่มีไฟเลี้ยง ดังนั้นหน่วยความจำชนิดนี้จึงจำเป็นต้องมีแบตเตอรี่สำรองอยู่ภายในแทกส์ด้วย ข้อดีของหน่วยความจำชนิดนี้คือ สามารถเก็บข้อมูลได้มากเวลาในการอ่านหรือเขียนข้อมูลนั้นสั้น

2. EEPROM (Electric Erasable PROM)

เป็นหน่วยความจำที่ไม่ต้องใช้แบตเตอรี่ในการสำรองไฟเพื่อรักษาข้อมูล เพราะภายในวงจรจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คาปาซิเตอร์ชนิดพิเศษทำหน้าที่ชาร์ตประจุไฟฟ้าให้กับวงจรของหน่วยความจำซึ่งมีอายุยาวนานถึง 10 ปี จำนวนครั้งในการเขียนข้อมูลมากกว่า 100,000 ครั้ง แต่เวลาในการอ่านหรือเขียนข้อมูล นานกว่า RAM และความจุก็น้อยกว่า RAM ด้วย

3. FRAM(Ferroelectric RAM)

EEPROM มีข้อเด่นคือเป็นหน่วยความจำที่ไม่ต้องมีแบตเตอรี่ แต่ข้อเสียคือใช้เวลาในการเขียนข้อมูลมากและใช้พลังงานในการเขียนข้อมูล FRAM จึงถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาข้อบกพร่องของ EEPROM คือสามารถเขียนข้อมูลได้เร็วกว่า ใช้พลังงานน้อยกว่า การเขียนข้อมูลซ้ำได้มากกว่า 1,000,000,000 ครั้ง และไม่ต้องมีแบตเตอรี่เหมือนกัน แต่เทคโนโลยีของ FRAM นั้นจะมีความซับซ้อนมากกว่า และเนื่องจากการพัฒนา FRAM ยังมีปัญหาอยู่บางประการทำให้หน่วยความจำชนิดนี้ยังไม่ถูกนำมาใช้งานแพร่หลาย

2.1.2 เครื่องอ่านสัญญาณ (Reader)

จะประกอบไปด้วยภาครับ/ส่งสัญญาณวิทยุ ส่วนควบคุมและเสาอากาศ (Antenna) ทำหน้าที่สอดคล้องสัญญาณกับแทกส์ และส่วนของการติดต่อสื่อสารกับอุปกรณ์ควบคุมภายนอก (RS232C) ขึ้นอยู่กับลักษณะการสื่อสารที่เราต้องการ เครื่องอ่านสัญญาณนั้น นอกจากสามารถจะอ่านข้อมูลได้แล้วยังสามารถเขียนข้อมูลได้ด้วย

2.1.3 ความถี่ของคลื่นพาห์

ในปัจจุบันได้มีการรวมกลุ่มระหว่างแต่ละประเทศ เพื่อทำการกำหนดมาตรฐานความถี่คลื่นพาห์ของระบบ RFID โดยมีสามกลุ่มใหญ่ ๆ คือ กลุ่มประเทศในยุโรปและอาฟริกา (Region 1) กลุ่มประเทศอเมริกาเหนือและอเมริกาใต้ (Region 2) และสุดท้ายคือกลุ่มประเทศตะวันออกไกลและออสเตรเลีย (Region 3) ซึ่งแต่ละกลุ่มประเทศจะกำหนดแนวทางในการเลือกใช้ความถี่ต่าง ๆ ให้แก่บรรดาประเทศสมาชิก อย่างไรก็ตาม ความถี่ของคลื่นพาห์ที่นิยมใช้งานในย่านความถี่ต่ำ ย่านความถี่ปานกลาง และย่านความถี่สูง ก็คือ 125 kHz , 13.56 MHz และ 2.45 GHz ตามลำดับดังที่แสดงไว้ในตารางที่ 2.1 นอกจากนี้รัฐบาลของแต่ละประเทศ โดยทั่วไปจะมีการออกกฎหมายเกี่ยวกับระเบียบการใช้งานย่านความถี่ต่าง ๆ รวมถึงกำลังส่งของระบบ RFID ด้วย

2.1.4 หลักการทำงานของระบบ RFID

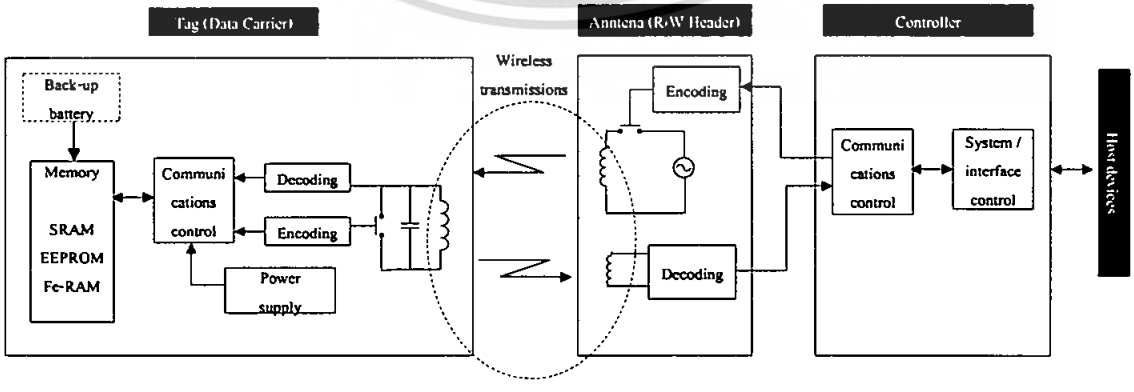
จากวงจรเทียบเคียง แสดงในรูปที่ 2.1 ส่วนควบคุมและติดต่อสื่อสาร (Control and Interface) จะได้รับคำสั่ง (Command) จากส่วนควบคุมที่สูงกว่า (Host) เช่นคอมพิวเตอร์หรือ PLC (Programmable Logic Controller) จากนั้นตัวควบคุมจะทำการประมวลผลคำสั่งจาก Host หลังจากนั้นก็จะสั่งให้ส่วนของภาครับ/ส่งวิทยุที่มีส่วนของวงจรเข้ารหัส (Coding) ในส่วนของวงจรผสมสัญญาณ (Modulation) ทำการผสมข้อมูลเข้ากับคลื่นพาห์แล้วทำการส่งออกไปทางเสาอากาศ ขนาดของพื้นที่ที่มีสัญญาณอยู่นั้นจะขึ้นอยู่กับขนาดของเสาอากาศและพลังงาน (Watt) ของเสาอากาศ เมื่อแทกส์เข้ามาในพื้นที่ที่มีสัญญาณแล้วเสาอากาศในแทกส์จะได้รับการคล้องสัญญาณทำให้แทกส์ทำงานได้ตามที่กล่าวแล้ว วงจรถอดรหัส (Demodulation) จะทำการแยกสัญญาณข้อมูลที่ถูกผสมมาจากเครื่องอ่านออกจากคลื่นพาห์แล้วทำการแปลงรหัส (Decoding) จากนั้นCPUของแทกส์จะรับคำสั่งไปประมวลผล ถ้าเป็นคำสั่งเขียน แทกส์จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกข้อมูลที่ส่งมาลงในหน่วยความจำของแทกส์ แต่ถ้าเป็นคำสั่งอ่าน แทกส์จะถึงข้อมูลจากหน่วยความจำที่ระบุไว้จากคำสั่ง แล้วทำการผสมข้อมูลที่วงจรผสมข้อมูลภายในแทกส์กับคลื่นพาห์ แล้วจึงส่งออกไปทางเสาอากาศ เมื่อเครื่องอ่านได้รับสัญญาณจากแทกส์ วงจรถอดรหัสของเครื่องอ่านก็จะถอดเอาข้อมูลออกจากคลื่นพาห์และส่งไปที่ Host Unit การสื่อสารระหว่างเครื่องอ่านกับแทกส์แสดงได้ดังรูปที่ 2.2

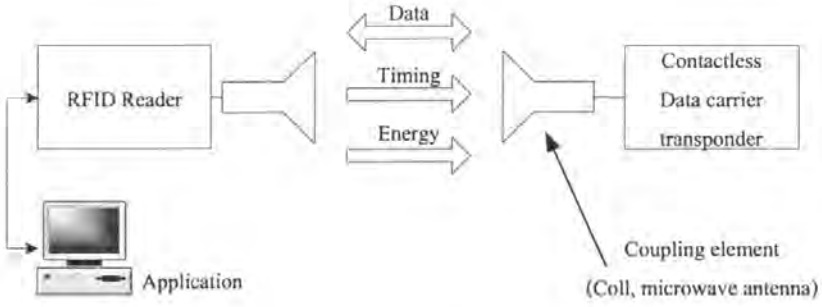
ตารางที่ 2.1 ย่านความถี่ต่าง ๆ ของระบบ RFID และการใช้งาน

ย่านความถี่	คุณลักษณะ	การใช้งาน
ย่านความถี่ต่ำ100-500 kHz ความถี่มาตรฐานที่ใช้งานทั่วไปคือ 125 kHz	-ระยะการรับส่งข้อมูลใกล้ -ต้นทุนไม่สูง -ความเร็วในการอ่านข้อมูลต่ำ -ความถี่ในย่านนี้เป็นที่แพร่หลายทั่วโลก	-Access Control -ปศุสัตว์ -ระบบคงคลัง -รถยนต์
ย่านความถี่กลาง 10-15 MHz ความถี่มาตรฐานที่ใช้งานทั่วไป คือ 13.56 MHz	-ระยะการรับส่งข้อมูลปานกลาง -ราคามีแนวโน้มถูกลงในอนาคต -ความเร็วในการอ่านข้อมูลปานกลาง -ความถี่ในย่านนี้เป็นที่แพร่หลายทั่วโลก	-Access Control -สมาร์ทการ์ด
ย่านความถี่สูง 850-950 MHz 2.4-5.8 GHz ความถี่มาตรฐานที่ใช้ งานทั่วไปคือ 2.45 GHz	-ระยะการรับส่งข้อมูลไกล (10 เมตร) -ความเร็วในอ่านข้อมูลสูง -ราคาแพง	-รถไฟ -ระบบเก็บค่าผ่านทาง



รูปที่ 2.1 วงจรเทียบเคียงของระบบ RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การสื่อสารระหว่างเครื่องอ่านกับแทกส์

2.1.5 ประเภทของ ระบบ RFID

RFID ถูกจำแนกออกเป็นประเภทได้หลายอย่างขึ้นอยู่กับว่าจะถูกจำแนกจากคุณสมบัติอะไร เช่น ความถี่
ที่ใช้งาน ชนิดของแทกส์หรือขนาดของหน่วยความจำของแทกส์ที่ใช้

2.1.5.1 RFID ที่จำแนกโดยขนาดของหน่วยความจำ

1) RFID ชนิด 1 บิต (2 Bit Type)

RFID ชนิดนี้เรียกอีกอย่างหนึ่งว่า EAS (Electronic Article Surveillmce) ดังรูปที่ 2.3 เป็น RFID ที่ใช้แทกส์ที่ไม่มีไมโครชิพ RFID ระบบนี้จะตรวจสอบเฉพาะว่าแทกส์อยู่ในพื้นที่สัญญาณหรือไม่ ดังนั้นสถานะจึงแสดงเพียงแค่มีหรือไม่มี ซึ่งเป็นรหัสดิจิตอล 0 หรือ 1 นั่นเอง

รูปที่ 2.3 แทกส์ของระบบ EAS จะมีเฉพาะเสาอากาศเพียงอย่างเดียว

เนื่องจากการทำงานของ RFID ระบบนี้ง่ายไม่ซับซ้อนและแทกส์มีราคาถูกมาก ระบบจึงถูกนำมาใช้กับการป้องกันสินค้าถูกขโมยในห้างร้านต่างๆ โดยแทกส์จะติดอยู่กับตัวสินค้าหรือซ่อนๆไว้ข้างหลังบาร์โค้ดอีกที เครื่องอ่านจะออกแบบเป็นโครงเสาอากาศสูงประมาณ 1-12 เมตร (ดังรูปที่ 2.4) อยู่ที่ทางออกของห้างร้านนั้นๆ เมื่อสินค้าที่มีแทกส์ติดอยู่ผ่านมาในบริเวณเสาอากาศนี้ก็จะถูกตรวจจับได้จากการคล้องสัญญาณจากเสาอากาศและแทกส์ จึงสามารถตรวจจับสินค้าที่ถูกนำออกไปได้ แทกส์จะถูกดึงออกหรือใช้เครื่องทำลายความเป็นสนามแม่เหล็กไฟฟ้าที่เคาน์เตอร์แคชเชียร์ เมื่อสินค้าถูกนำมาชำระ ซึ่งเราจะพบเห็นระบบ EAS (ดังรูปที่ 2.3) ในซุปเปอร์มาร์เก็ตหรือดิสเคาน์สโตร์ต่าง ๆ เช่น คาร์ฟูร์ โลตัส บิ๊กซี หรือร้านขายเสื้อผ้า ร้านขาย CD ต่างๆ ช่วงความถี่ที่ใช้งานจะเป็นช่วงความถี่ไมโครเวฟ เนื่องจากระยะในการสื่อสารค่อนข้างไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 เสาอากาศของระบบ EAS

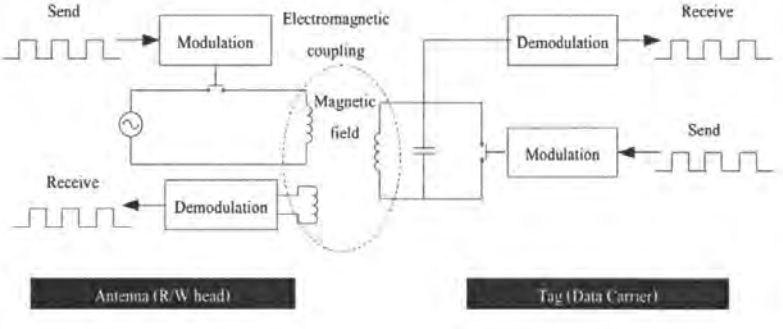
2) RFID ชนิดหน่วยความจำมากกว่า 1 บิต (Data Carrier Type)

RFID ชนิดนี้จะใช้แทกส์ที่มีไมโครชิพและหน่วยความจำที่เป็นส่วนประกอบสำคัญ มีราคาสูงกว่าแทกส์ชนิด EAS โดยบางชนิดสามารถเก็บข้อมูลได้สูงสุดถึง 64 กิโลไบต์ ใช้ในอุตสาหกรรมหรืองานทั่วไปที่ต้องใช้แทกส์ในการเก็บข้อมูล

2.1.5.2 RFID ที่จำแนกโดยลักษณะการคล้องของสัญญาณ

1) Close Coupling

เป็น RFID ที่มีระยะในการอ่าน/เขียนข้อมูลสั้นมากประมาณ 0-1 เซนติเมตร ดังนั้นแทกส์จะต้องอยู่ใกล้หรือวางอยู่บนเครื่องอ่าน Close Coupling นี้จะสามารถใช้คลื่นความถี่ได้ตั้งแต่ 0 Hz จนถึง 50 MHz เนื่องจากการทำงานของแทกส์ไม่อาศัยการส่งพลังงานจากการแผ่คลื่นแม่เหล็กไฟฟ้าจากเครื่องอ่านแต่อาศัยการเหนี่ยวนำเหมือนหลักการของหม้อแปลงไฟฟ้า ทำให้เกิดพลังงานที่ทำให้วงจรภายในแทกส์ทำงานได้ดังรูปที่ 2.5 ระบบ Close-Coupling จะนิยมมาใช้งานที่ต้องการความปลอดภัยค่อนข้างสูง แต่ไม่ต้องการติดต่อได้ไกล เช่น ประตูอัตโนมัติหรือสมาร์ทการ์ดแบบไร้สัมผัส (Contactless Smart Cards)

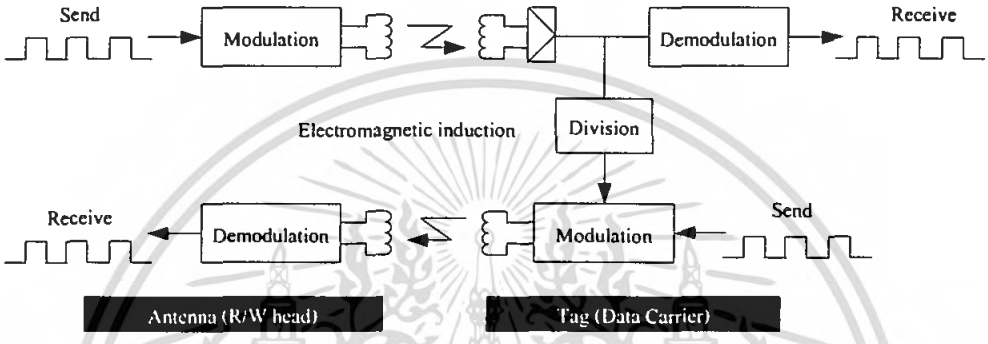


รูปที่ 2.5 วงจรเทียบเคียงของ Close-Coupling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Remote Coupling

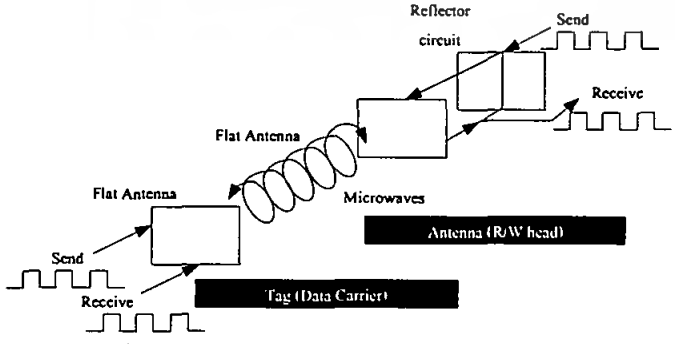
เป็นระบบที่มีระยะการอ่าน/เขียนสูงถึง 1 เมตร ระบบนี้จะใช้หลักการคล้องสัญญาณแบบ Inductive (Magnetic) Coupling ระหว่างเครื่องอ่านกับแทกส์ซึ่งโดยประมาณ 90-95% ของระบบ RFID ที่ใช้งานในปัจจุบันใช้หลักการของ Remote Coupling นี้ โดยความถี่ที่ใช้งานมีหลายความถี่ตั้งแต่ต่ำกว่า 135 kHz หรือ 13.56 MHz และ 27.125 MHz พลังงานไฟฟ้าจะถูกส่งโดยหลักการแผ่ของคลื่นแม่เหล็กไฟฟ้าไปให้แทกส์ ทำให้แทกส์ได้รับพลังงานสามารถทำงานได้ ดังรูปที่ 2.6 ระบบ Remote Coupling นี้จะพบมากในลักษณะงานอุตสาหกรรมเช่น รถยนต์ เครื่องใช้ไฟฟ้าหรืออิเล็กทรอนิกส์



รูปที่ 2.6 วงจรเทียบเคียงของ Remote Coupling หรือ Inductive Coupling

3) Long Range

ระบบนี้จะมีการอ่าน/เขียนอยู่ระหว่างระยะ 1 ถึง 10 เมตรหรือบางระบบอาจจะระยะไกลกว่านี้ ความถี่ที่ใช้ในระบบนี้จะเป็นย่านที่มีความถี่สูงมากหรือไมโครเวฟ (Microwave range) ซึ่งปกติที่ความถี่ 2.45 GHz หรือบางครั้งจะพบที่ 915 MHz, 5.8 GHz และ 24.125 GHz แต่การส่งพลังงานจากตัวเครื่องอ่านไปยังแทกส์ทำได้ยาก ดังนั้นแทกส์ที่ใช้งานจะเป็นชนิดที่มีแบตเตอรี่ในตัวซึ่งจะใช้สำหรับเป็นไฟเลี้ยงที่ทำให้ไมโครชิพทำงานและเก็บรักษาข้อมูล มีวงจรเทียบเคียงดังรูปที่ 2.7 ลักษณะงานที่พบจะเห็นเป็นลักษณะงานที่ต้องการการสื่อสารระยะไกล เช่น ในกระบวนการผลิตรถยนต์ ระบบชำระเงินอัตโนมัติของทางด่วน



รูปที่ 2.7 วงจรเทียบเคียงของ Long Range

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.3 RFID ที่จำแนกตามความสามารถของระบบ

1) ระบบอ่านอย่างเดียว (Read Only System)

แทกส์มีข้อมูลซึ่งจะอยู่ในรูปของซีเรียลนัมเบอร์ และไม่สามารถเขียนข้อมูลใหม่ลงไปได้ เหมาะกับงานที่ต้องการอ่านอย่างเดียว เพื่อแยกแยะความแตกต่างของสินค้าหรือบุคคล ระบุชนิดของสินค้าหรือตู้คอนเทนเนอร์ มีราคาต่ำ ความถี่ที่ใช้งานจะอยู่ที่ต่ำกว่า 135 kHz หรือ 2.45 GHz

แทกส์ชนิดนี้เมื่อเข้าสู่บริเวณการตรวจสอบของเครื่องอ่านสัญญาณ จะทำการเริ่มส่งตัวเลขไอดีของตัวเองอย่างต่อเนื่อง โดยปกติแล้วผู้ผลิตชิพจะรับรองว่าแต่ละ Serial Number นั้นจะใช้แค่ครั้งเดียว ตัวเลขเฉพาะของแทกส์จะถูกบรรจุลงไปในแทกส์ในระหว่างการผลิตชิพ ผู้ใช้ไม่สามารถเปลี่ยนแปลง Serial Number หรือข้อมูลบนชิพได้

เมื่อแทกส์เข้าสู่บริเวณการตรวจสอบของเครื่องอ่านสัญญาณ การนับจะเริ่มนับทุกแอดเดรสที่อยู่ในหน่วยความจำอย่างต่อเนื่อง เอาท์พุทของข้อมูลของหน่วยความจำจะเชื่อมต่อกับโหลดมอดูเลเตอร์ที่เซตเป็นรหัสเบสแบนด์ของรหัสไบนารี่ ในกระบวนการนี้ข้อมูลทั้งหมดในหน่วยความจำ (128 บิตซีเรียลนัมเบอร์) สามารถส่งไปได้อย่างต่อเนื่อง การติดต่อสื่อสารกับเครื่องอ่านสัญญาณจะเป็นแบบทิศทางเดียวโดยแทกส์จะส่งตัวเลขไอดีไปยังเครื่องอ่านสัญญาณ อย่างต่อเนื่อง

2) ระบบอ่านเขียน (Read-Write System)

แทกส์สามารถเขียนข้อมูลซ้ำได้ โดยความจุอยู่ที่ 16 ไบต์ จนถึงมากกว่า 16 กิโลไบต์ การอ่านและเขียนข้อมูลในแทกส์จะเป็นแบบบล็อกข้อมูล ในบล็อกจะมีรูปแบบเป็นจำนวนไบต์ที่สามารถอ่านและเขียนได้ การที่จะเปลี่ยนแปลงข้อมูลของแต่ละบล็อก บล็อกทั้งหมดจะถูกอ่านจากแทกส์ หลังจากนั้นในบล็อกเดียวกันรวมกับบล็อกที่แก้ไขแล้ว จะถูกเขียนกลับลงไปในแทกส์หน่วยความจำที่ใช้จะเป็นชนิด EEPROM หรือ SRAM ความถี่ที่ใช้งานจะเป็น 135 kHz, 13.56 MHz, 27.125 MHz และ 2.45 GHz

2.1.6 กระบวนการรับข้อมูลของ RFID

2.1.6.1 กระบวนการ Half Duplex (HDX) ข้อมูลจะส่งจากตัวส่งไปยังตัวอ่าน และสลับเปลี่ยนข้อมูลส่งจากตัวอ่านไปยังตัวส่ง

2.1.6.2 กระบวนการ Full Duplex (FDX) ข้อมูลจะส่งจากตัวส่งไปยังตัวอ่าน และในเวลาเดียวกันตัวอ่านจะส่งข้อมูลไปยังตัวส่งด้วย

ข้อมูลจะส่งจากตัวเครื่องอ่านสัญญาณไปยังแทกส์ที่เรียกว่า Downlink และจากแทกส์ไปยังเครื่องอ่านสัญญาณเรียกว่า Uplink ซึ่งทั้งสองกระบวนการแสดงดังรูปที่ 2.8

2.1.7 พลังงานที่จ่ายให้แทกส์

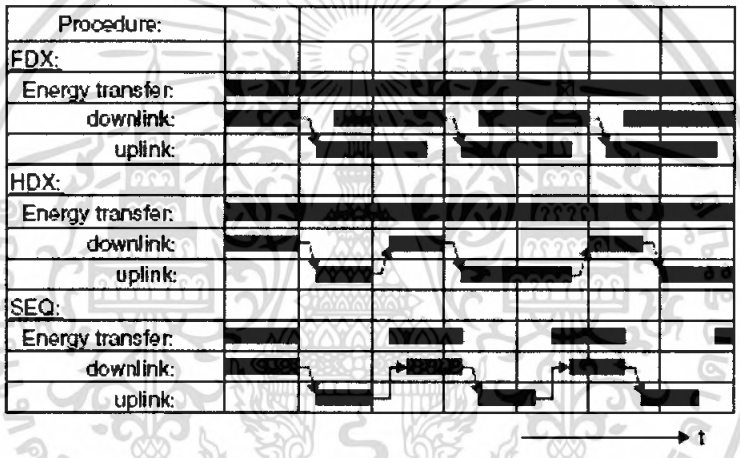
การคล้องสัญญาณโดยการเหนี่ยวนำของแทกส์ประกอบด้วยเครื่องมือเก็บข้อมูลทางไฟฟ้า ส่วนใหญ่ใช้ไมโครชิพเพียงตัวเดียว และขดลวดขนาดกว้างเป็นตัวทำเสาอากาศ

การคล้องสัญญาณโดยการเหนี่ยวนำของแทกส์ ส่วนใหญ่เป็นแทกส์แบบพาสซีฟ ความหมายคือพลังงานทั้งหมดใช้สำหรับการทำงานของไมโครชิพ เสาอากาศของตัวอ่านนั้นจะสร้างความถี่คลื่นสนามแม่เหล็กสูงซึ่งจะแพร่ไปรอบๆขดลวด เนื่องจากความยาวคลื่นของช่วงความถี่ที่ใช้ (<135kHz:

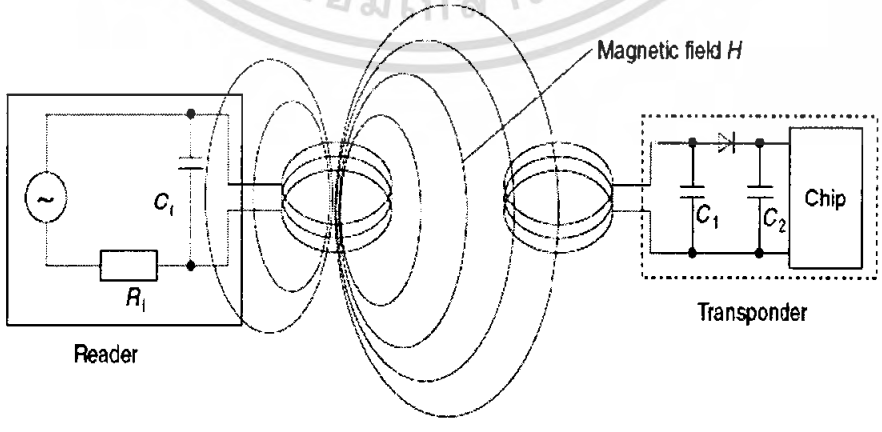
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2400m, 13.56MHz: 22.1m) มากกว่าระยะระหว่างเสาอากาศของตัวอ่านกับตัวแทกส์ สนามแม่เหล็กไฟฟ้าอาจเปลี่ยนเป็นสนามแม่เหล็กซึ่งขึ้นอยู่กับระยะทางของเสาอากาศของตัวอ่านกับตัวแทกส์

ส่วนของการส่งสนามแพร่ไปทั่วเสาอากาศของตัวแทกส์ ซึ่งคือระยะทางจากขดลวดของตัวอ่าน โวลต์ (Ui) ถูกสร้างขึ้นที่เสาอากาศของตัวแทกส์โดยการเหนี่ยวนำ โวลต์นี้จะถูกแปลงและใช้เป็นพลังงานสำหรับในไมโครชิพ ตัวเก็บประจุ Cr ถูกต่อขนานกับเสาของเครื่องอ่านสัญญาณ การเก็บประจุของตัวเก็บประจุนี้จะถูกเลือกซึ่งจะทำงานกับขดลวดเหนี่ยวนำของเสาอากาศจากวงจร parallel resonant ด้วยความถี่ resonant ซึ่งตรงกับความถี่ของเครื่องอ่านสัญญาณ กระแสที่สูงมากถูกสร้างขึ้นในเสาขดลวดของตัวเครื่องอ่านสัญญาณโดยการเข้ากันของวงจร parallel resonant ซึ่งจะสร้างสนามแม่เหล็กที่ต้องการในการทำงานของแทกส์ เสาอากาศของตัวแทกส์และ C1 จากวงจร resonant จะส่งความถี่ไปยังตัวเครื่องอ่านสัญญาณ โวลเตจ Ui ที่ขดลวดในตัวแทกส์จะมากที่สุดก็ขึ้นอยู่กับขั้นตอน resonant ในวงจร parallel resonant ดังรูปที่ 2.9



รูปที่ 2.8 แสดง Full Duplex, Half Duplex และ Sequential System บนแกนเวลา



รูปที่ 2.9 แสดงพลังงานการเหนี่ยวนำที่เกิดจากการคล้องสัญญาณของแทกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พลังงานการเหนี่ยวนำที่เกิดจากการคล้องสัญญาณของแทกส์เนื่องจากพลังงานของสนามแม่เหล็กที่เปลี่ยนแปลงและสร้างโดยตัวอ่านสัญญาณ

รูปของขดลวด 2 ขดคล้ายกับหม้อแปลง ซึ่งกรณีนี้เป็นการคล้องสัญญาณซึ่งได้สัญญาณที่อ่อนมาก ประสิทธิภาพของพลังงานในการส่งระหว่างเสาอากาศของตัวส่งและตัวรับเป็นสัดส่วนของความถี่ ตัวเลขจำนวนรอบการพันของขดลวด n, พื้นที่ A ล้อมรอบด้วยขดลวดของแทกส์, มุมของ 2 ขดลวด, ความแตกต่างของแต่ละตัวและระยะห่างระหว่าง 2 ขดลวด

ความถี่จะแปรผกผันกับจำนวนขดลวดเหนี่ยวนำของแทกส์ กล่าวคือจำนวนขดลวดจะลดลงเมื่อความถี่มีค่าสูงขึ้น (135 kHz: typical 100-1000 windings, 13.56 MHz: typical 3-100 windings) เนื่องจากโวลต์ที่เกิดขึ้นในแทกส์นั้นเป็นสัดส่วนของความถี่ด้วย ดังนั้นการลดจำนวนรอบของขดลวดจะส่งผลต่อประสิทธิภาพของพลังงานในการส่งความถี่ที่สูงขึ้น

ตารางที่ 2.2 แสดงการใช้พลังงานของหลายๆ RFID-ASIC

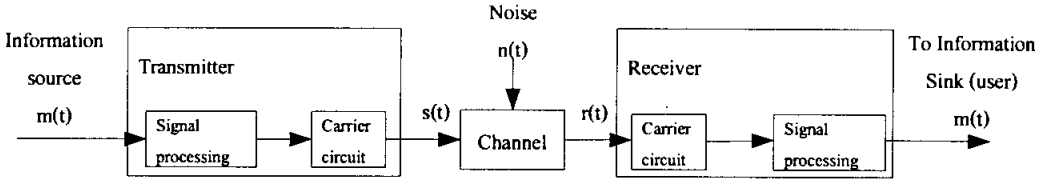
	Memory/Byte	Write/Read distance	Power consumption	Frequency	Application
ASIC#1	6	15 cm	10 uA	120 kHz	Animal ID
ASIC#2	32	13 cm	600 uA	120 kHz	Goods flow, access check
ASIC#3	256	2 cm	6 uA	128 kHz	Public transport
ASIC#4	256	0.5 cm	< 1 mA	4 MHz*	Goods flow, public transport
ASIC#5	256	< 0.2 cm	~ 1 mA	4/13.56 MHz	Good flow
ASIC#6	256	100 cm	500 uA	125 kHz	Access check
ASIC#7	2048	0.3 cm	< 10 mA	4.91 MHz*	Contactless chip cards
ASIC#8	1024	10 cm	~ 1 mA	13.56 MHz	Public transport
ASIC#9	8	100 cm	< 1mA	125 kHz	Goods flow
ASIC#10	128	100 cm	< 1 mA	125 kHz	Access check

*Close coupling system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8 เทคโนโลยีการเข้ารหัสและถอดรหัส

กระบวนการเข้ารหัสและถอดรหัสแสดงดังรูปที่ 2.10



รูปที่ 2.10 การไหลของสัญญาณและข้อมูลในระบบการสื่อสารแบบดิจิตอล

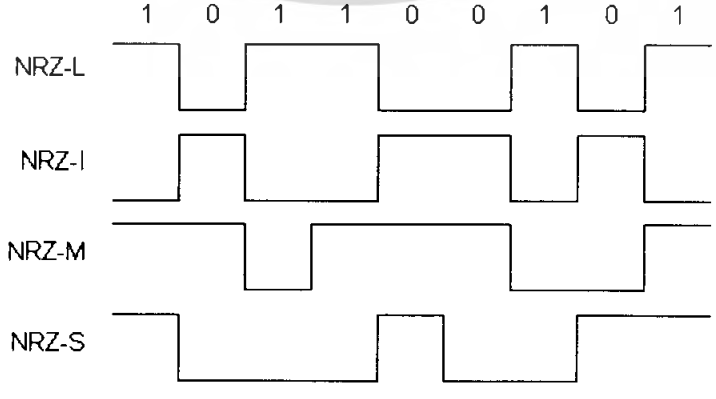
จากรูปแสดงกระบวนการสื่อสารแบบดิจิตอล ซึ่งก็คือการส่งข้อมูลระหว่างเครื่องอ่านกับแทกส์ในระบบ RFID นั่นเอง กระบวนการสื่อสารจะอาศัยหลักการต่างๆประกอบไปด้วยการเข้ารหัสข้อมูล (Signal Coding) การผสมรหัสข้อมูลเข้ากับคลื่นพาห์ (Modulation) การส่งคลื่นสัญญาณออกไป (Transmission) และการถอดข้อมูลออกจากคลื่นพาห์ (Demodulation)

การเข้ารหัสข้อมูล (Signal Coding)

การรับส่งข้อมูลแบบตรงไปตรงมาจะทำให้ข้อมูลอาจถูกรบกวนและทำให้การซิงโครไนซ์ของข้อมูลเกิดการคลาดเคลื่อน (โดยปกติวงจรดิจิตอลจะปรับการซิงโครไนซ์ของข้อมูลได้เฉพาะในช่วงที่มีการเปลี่ยนแปลงระดับของข้อมูลจาก 1 เป็น 0 หรือจาก 0 เป็น 1) ทำให้ได้รับข้อมูลผิดพลาด เพื่อป้องกันปัญหาดังกล่าวจึงต้องมีการนำสัญญาณดิจิตอลปกติไปผ่านการเข้ารหัสเสียก่อนๆที่รหัสจะถูกส่งไปมอดูเลต แต่การเข้ารหัสบางอย่างก็มีข้อเสียคือช่วงความถี่ที่ใช้ในการส่งข้อมูลอาจต้องเพิ่มขึ้นเป็น 2 เท่า การเข้ารหัสมักเป็นที่นิยมใช้กันอย่างกว้างขวางในการสื่อสารแบบดิจิตอล ในระบบการสื่อสารแบบดิจิตอลนั้นจะใช้สัญลักษณ์ 0 กับ 1 แทนข้อมูล โดยความแตกต่างของข้อมูลจะเป็นตัวกำหนดค่า 0 กับ 1 ของแต่ละบิตข้อมูล ข้อมูลจะถูกจัดเรียงเป็นแนวนอนหรือเส้นตรง (Line Code) ซึ่งมาตรฐานของ Line Code จะมีหลายมาตรฐานและมาตรฐานที่นำมาใช้ในระบบ RFID มีดังนี้คือ

1) NRZ coding

การเข้ารหัส NRZ จะมีลักษณะสัญญาณดังรูปที่ 2.11



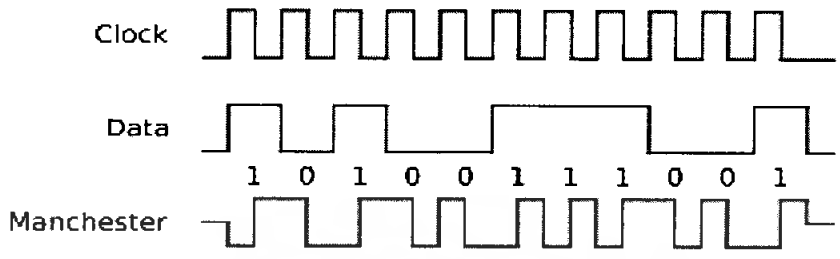
รูปที่ 2.11 รหัสสัญญาณ NRZ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์ "1" จะแทนด้วยช่วงสัญญาณที่เป็น High, สัญลักษณ์ "0" จะแทนด้วยช่วงสัญญาณที่เป็น Low

2) Biphase coding

การเข้ารหัส Biphase แบบ Manchester จะมีลักษณะสัญญาณดังรูปที่ 2.12



รูปที่ 2.12 รหัสสัญญาณ Biphase

ข้อกำหนดของลักษณะการเข้ารหัสแต่ละแบบนี้อาจจะแตกต่างกันออกไปจากที่กล่าวมาข้างต้นตามชนิดของแทกส์ของแต่ละบริษัทผู้ผลิตแทกส์

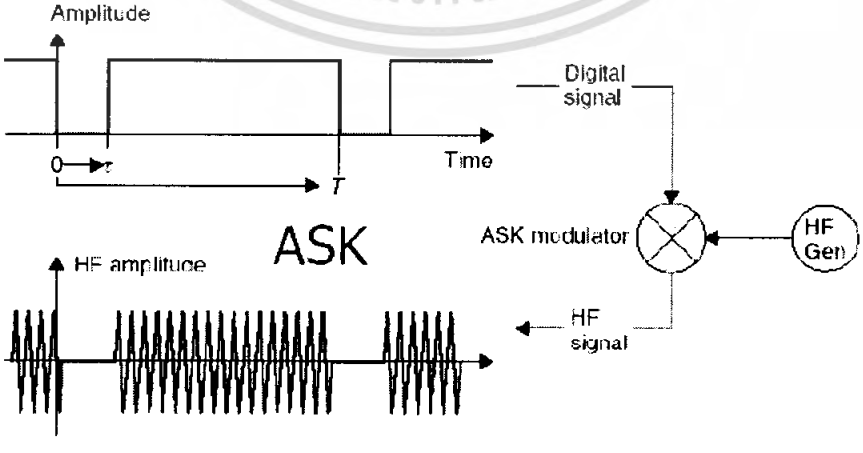
การผสมรหัสข้อมูลเข้ากับรหัสคลื่นพาห์ (Modulation)

เนื่องจากหลักการพื้นฐานของ RFID คือใช้คลื่นสนามแม่เหล็กไฟฟ้าในการสื่อสารรับส่งข้อมูล ดังนั้นก่อนการสื่อสารจะต้องมีกระบวนการผสมข้อมูลที่เข้ารหัสไว้แล้วไปกับคลื่นพาห์ (Modulation) แต่เนื่องจากลักษณะข้อมูลเป็นแบบดิจิตอลจึงจำเป็นต้องใช้วิธีการผสมแบบดิจิตอลซึ่งจะแตกต่างจากแบบ

แอนาลอกที่ใช้กันอยู่หลายระบบในปัจจุบันเช่น FM และ AM การผสมข้อมูลแบบดิจิตอลมี 3 วิธี ดังนี้

1) Amplitude Shift Keying (ASK)

เป็นการผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงของยอดคลื่นพาห์ (Amplitude) และจะแสดงลักษณะข้อมูลโดยความถี่ของคลื่นพาห์ไม่เปลี่ยนแปลง ความสูงของยอดคลื่นจะเปลี่ยนสถานะอยู่ 2 สถานะขึ้นอยู่กับข้อมูลที่ผสมเข้ามา มีกระบวนการดังรูปที่ 2.13

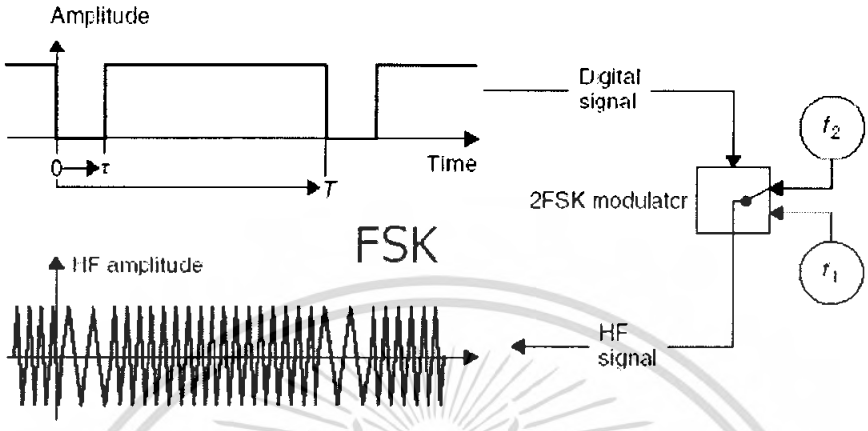


รูปที่ 2.13 แสดงการผสมข้อมูลแบบ BinaryASK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Frequency Shift Keying (FSK)

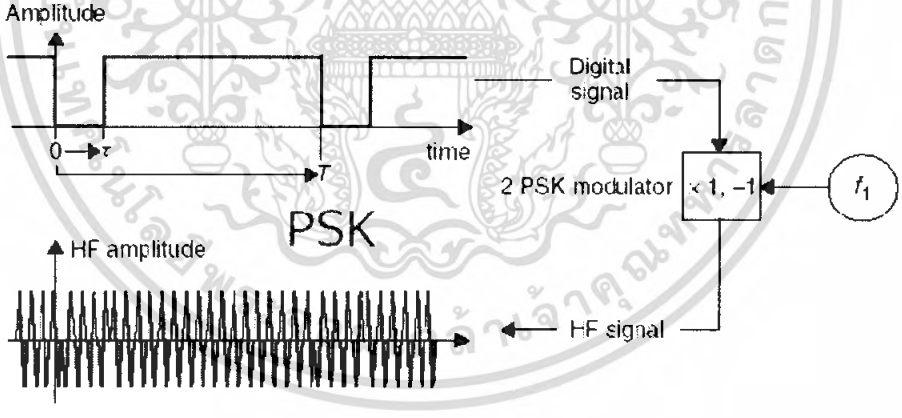
วิธีนี้จะอาศัยการเปลี่ยนแปลงความถี่ของคลื่นพาห์ขึ้นอยู่กับสถานะของบิตข้อมูล โดยความสูงของยอดคลื่น (Amplitude) ไม่เปลี่ยนแปลง มีกระบวนการดังรูปที่ 2.14



รูปที่ 2.14 แสดงการผสมข้อมูลแบบ Binary FSK

3) Phase Shift Keying (PSK)

วิธีนี้จะใช้หลักการเปลี่ยนเฟสของคลื่นขึ้นอยู่กับสถานะของบิตข้อมูล จะกลับเฟสทุกครั้งเมื่อมีการเปลี่ยนสถานะของข้อมูล มีกระบวนการดังรูปที่ 2.15



รูปที่ 2.15 แสดงการผสมข้อมูลแบบ Binary PSK

แต่ในระบบ RFID จะใช้เพียงการมอดูเลต สัญญาณที่เข้ารหัสมาแล้วกับสัญญาณคลื่นพาห์แบบ ASK เท่านั้นในการรับส่งข้อมูลระหว่างเครื่องอ่านสัญาณกับแทกส์

การส่งคลื่นสัญญาณ (Transmission)

เครื่องอ่านจะเป็นตัวส่งคลื่นสนามแม่เหล็กไฟฟ้าออกไปด้วยความถี่ที่จะใช้งาน เมื่อแทกส์อยู่ในระยะที่เหมาะสมจะสามารถเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้านั้นมาแปลงเป็นพลังงานที่ใช้ในการทำงานภายในได้ และแทกส์จะนำสัญญาณข้อมูลดิจิตอลที่เข้ารหัสแล้วมาทำการมอดูเลตกับคลื่นแม่เหล็กไฟฟ้าที่รับได้แบบ ASK ส่งกลับไปยังเครื่องอ่านซึ่งเรียกสัญญาณนี้ว่า "สัญญาณ backscattering"

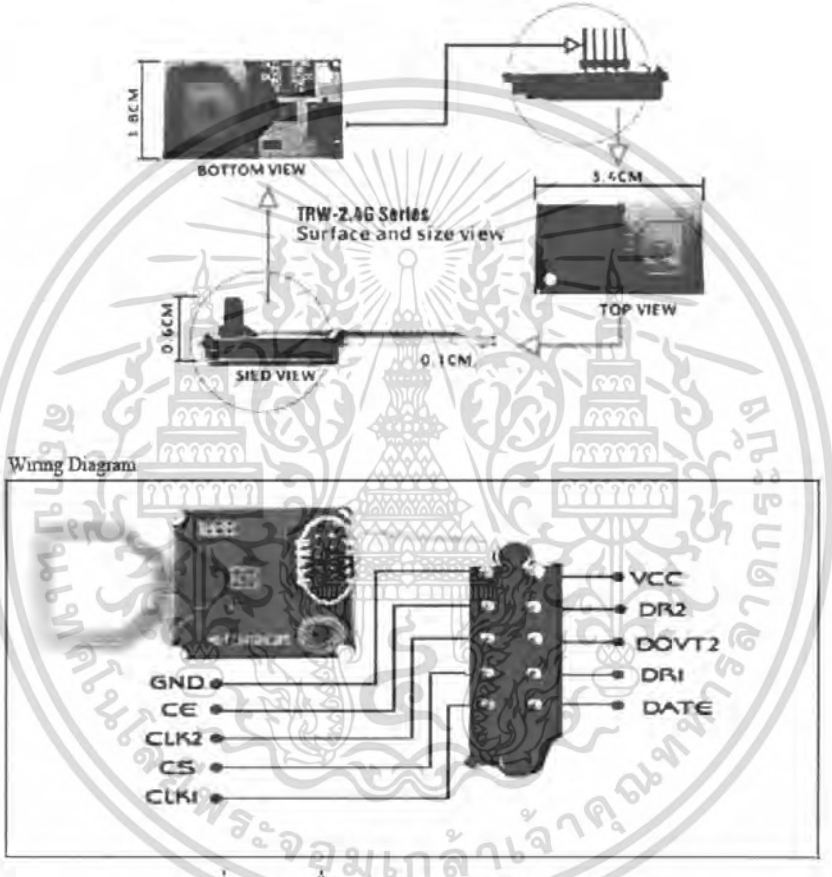
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถอดข้อมูลออกจากคลื่นพาห์ (Demodulation)

สัญญาณที่ทางเครื่องอ่านรับได้นั้น จะเป็นสัญญาณข้อมูลที่ถูกเข้ารหัสมาแล้วมอดูเลตกับคลื่นพาห์แบบ ASK เพราะฉะนั้นเราสามารถนำสัญญาณเดิมกลับมาได้โดยการดีมอดูเลตด้วยวิธีการจับกรอบสัญญาณจะเป็นสัญญาณข้อมูลที่ถูกเข้ารหัสโดยแทกส์

2.2 คู่มือการใช้งานเครื่องรับ-ส่ง TRW 2.4GHz

ลักษณะของ TRW 2.4GHz แสดงดังรูปที่ 2.16



รูปที่ 2.16 เครื่องส่งสัญญาณ TRW 2.4 GHz

2.2.1 โหมดการทำงานของ TRW 2.4GHz

โหมดในการใช้งานของโมดูลความถี่วิทยุ 2.4 จิกะเฮิรตซ์ มีอยู่ 2 โหมดคือ

1. ShockBurst Mode
2. Direct Mode

2.2.2 ShockBurst

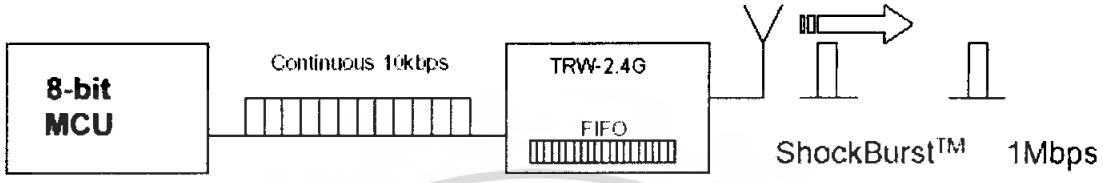
เทคโนโลยี ShockBurst นั้น ใช้ระบบ FIFO (First-In First-Out) โดยเริ่มการทำงานในอัตราการรับส่งข้อมูลต่ำ จากนั้นจะส่งข้อมูลออกไปในอัตราการรับส่งข้อมูลที่สูงมาก ด้วยเหตุนี้ทำให้สามารถลดการใช้พลังงานลงไปได้อย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

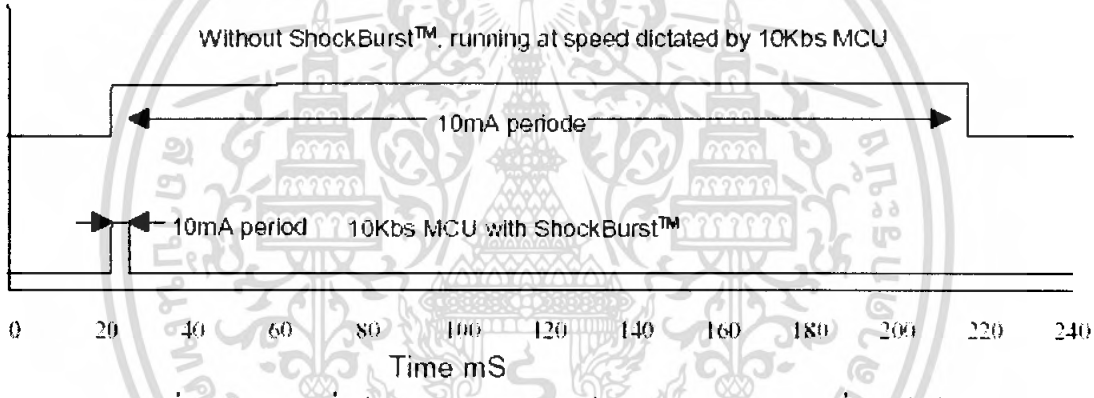
เมื่อ TRW 2.4GHz ทำงานในโหมด ShockBurst จะทำให้ใช้งานได้ในอัตราการรับส่งข้อมูลสูงถึง (1 Mbps) ในย่านความถี่ 2.4 GHz โดยไม่จำเป็นต้องใช้ไมโครคอนโทรลเลอร์ที่มีความเร็วสูง

หลักการของ ShockBurst

เมื่อ TRW 2.4GHz ถูกตั้งค่าการทำงานในโหมด ShockBurst, การทำงานของ TX หรือ RX จะเป็นดังรูปที่ 2.17 (อัตราการรับส่งที่ 10 kbps นั้นใช้สำหรับเป็นตัวอย่างเท่านั้น)



รูปที่ 2.17 แสดงการทำงานของ ShockBurst



รูปที่ 2.18 กระแสที่ใช้ในการทำงานขณะใช้ ShockBurst และขณะที่ไม่ได้ใช้

จากรูปที่ 2.18 จะเห็นว่า วิธีการแบบ ShockBurst จะใช้กระแสไฟที่น้อยกว่าเมื่อเทียบกับคาบเวลา

ShockBurst ขณะทำการส่ง

ขาที่ใช้เชื่อมต่อกับไมโครคอนโทรลเลอร์: CE, CLK1, DATA

- เมื่อไมโครคอนโทรลเลอร์มีข้อมูลที่จะส่งออกไป ให้ตั้งค่า CE เป็น high จะทำให้ภายใน TRW 2.4GHz มีการประมวลผลข้อมูลเพื่อเตรียมการส่งข้อมูล

- Address ของการรับ (RX address) และ Payload data นั้นจะเริ่มการทำงานของ TRW 2.4GHz จากนั้นไมโครคอนโทรลเลอร์ทำการกำหนดค่าความเร็วให้ต่ำกว่า 1 Mbps (ในตัวอย่างคือ 10 kbps)

- ไมโครคอนโทรลเลอร์ตั้งค่า CE เป็น low ซึ่งเป็นการสั่งให้ TRW 2.4GHz เริ่มทำการส่งในโหมด ShockBurst

TRW 2.4GHz ในโหมด ShockBurst

- RF front end (ส่วนที่ป้องกันการเข้าถึงของข้อมูล) นั้นจะมีความสามารถเพิ่มขึ้น

- RF package นั้นจะครบสมบูรณ์ (preamble จะถูกเพิ่ม และ CRC จะถูกสร้างในขั้นตอนนี้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข้อมูลจะถูกส่งออกไปด้วยความเร็วสูง (250 kbps หรือ 1 Mbps ขึ้นอยู่กับการตั้งค่าของผู้ใช้งาน)
- TRW 2.4GHz จะเข้าสู่สภาวะ stand-by เมื่อทุกขั้นตอนเสร็จสิ้น

ShockBurst ขณะทำการรับ

ขาที่ใช้เชื่อมต่อกับไมโครคอนโทรลเลอร์: CE, DR1, CLK1, DATA (การรับข้อมูลช่องทางเดียว)

- เมื่อตั้งค่า CE เป็น high จะทำให้ภายใน TRW 2.4GHz มีการประมวลผลข้อมูลเพื่อเตรียมการรับข้อมูล
- มีการตรวจสอบ address และขนาดของ Payload ของ RF package ที่เข้ามา
- หลังจาก TRW 2.4GHz เริ่มทำงาน 200 uSec TRW 2.4GHz จะตรวจสอบในอากาศว่ามีการติดต่อสื่อสารเข้ามาหรือไม่
- เมื่อได้รับ package ที่ถูกต้อง (address ถูกต้อง และมีการสร้าง CRC) TRW 2.4GHz จะทำการกำจัด preamble, address และบิต CRC
- TRW 2.4GHz ในเวลานั้นจะแจ้ง (interrupts) ไปที่ไมโครคอนโทรลเลอร์ว่าได้ทำการรับข้อมูลที่ถูกต้อง โดยการทำให้ DR1 เป็น high
- ไมโครคอนโทรลเลอร์อาจจะทำการตั้งค่า CE ให้เป็น low เพื่อยกเลิกระบบ RF front end (โหมดกินกระแสต่ำ)
- ไมโครคอนโทรลเลอร์นั้นจะสิ้นสุดการทำงาน โดยทำการปรับอัตราการรับส่งข้อมูลของ payload data ให้เหมาะสม (ในตัวอย่างนี้คือ 10 Kbps)
- เมื่อ payload data ทั้งหมดถูกทำให้กลับสภาพเดิม TRW 2.4GHz จะตั้งค่า DR1 เป็น low อีกครั้ง
- ถ้า CE เป็น high อยู่ตลอดการถ่ายข้อมูล TRW 2.4GHz นั้นจะพร้อมสำหรับการรอรับชุดข้อมูลชุดใหม่
- ถ้า CE เป็น low จะเป็นการเริ่มต้นการทำงานใหม่ตามลำดับ

DuoCeiver โหมดรับข้อมูล 2 ช่องทาง

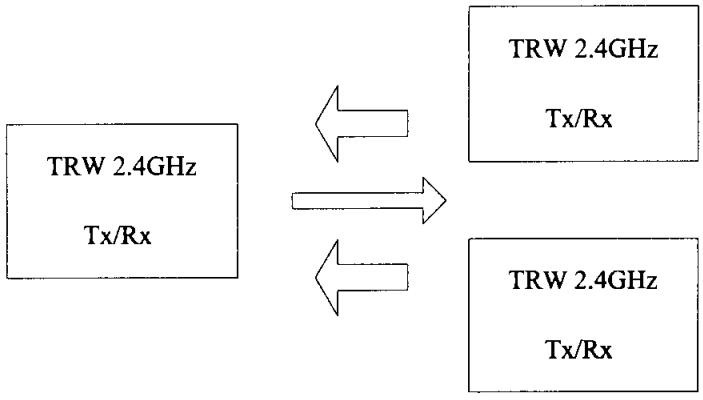
TRW 2.4GHz สามารถรับข้อมูลจาก ตัวส่ง 1 Mbps สองตัว โดยทั้งสองตัวต้องมีช่องสัญญาณห่างกัน 8 MHz ซึ่ง TRW 2.4GHz ตัวรับจะสามารถรับข้อมูลได้ทั้งสองช่องสัญญาณโดยใช้เสาอากาศของตัวรับเพียงตัวเดียว แสดงดังรูปที่ 2.19

การเชื่อมต่อกับไมโครคอนโทรลเลอร์จะใช้ขาสัญญาณดังนี้

- Data Channel 1: CLK1, DATA, and DR1
- Data Channel 2: CLK2, DOUT2, and DR2
- DR1 and DR2 จะใช้ใน ShockBurst mode เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

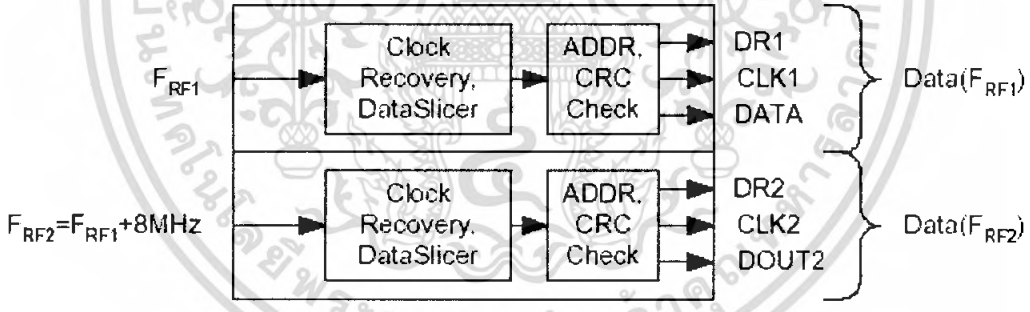


รูปที่ 2.19 การรับข้อมูล 2 ช่องสัญญาณของ TRW 2.4GHz

การจะใช้งานช่องสัญญาณที่ 2 นั้น จะต้องกำหนดให้มีช่องสัญญาณความถี่สูงกว่าช่องสัญญาณที่ 1 อยู่ 8 MHz แสดงได้ดังรูปที่ 2.20

ใน Direct mode นั้นไมโครคอนโทรเลอร์สามารถรับข้อมูลของ TRW 2.4GHz ที่เข้ามาจาก2ช่องสัญญาณได้พร้อมกัน โดยไม่ต้องทำการสลับรับข้อมูลจาก 2 ช่องสัญญาณ

ใน ShockBurst mode สามารถให้ไมโครคอนโทรลเลอร์หยุดการทำงานของช่องสัญญาณหนึ่ง เพื่อไปรับข้อมูลจากอีกช่องสัญญาณหนึ่งได้ โดยปราศจากการสูญเสียของข้อมูล การทำเช่นนี้จะทำให้ลดภาระการทำงานของไมโครคอนโทรลเลอร์ได้



รูปที่ 2.20 การรับข้อมูล 2 ช่องสัญญาณพร้อมกัน

การตั้งค่าให้กับอุปกรณ์

การตั้งค่าสำหรับการทำงานแบบ ShockBurst จะมีส่วนประกอบดังรูปที่ 2.21 และจะประกอบไปด้วยค่าต่างๆดังนี้

- Payload section width: เป็นการระบุจำนวนของ payload บิตใน RF package ซึ่งจะทำให้ TRW 2.4GHz นั้นสามารถจำแนกความแตกต่างของ payload data และ CRC ไบท์ในแพคเกจที่รับมาได้
- Address width: เป็นการกำหนดจำนวนบิตที่ใช้สำหรับ address ใน RF package ซึ่งจะทำให้ TRW 2.4GHz นั้นสามารถจำแนกความแตกต่างระหว่าง address และ payload data
- Address (RX Channel 1 และ 2): เป็นการกำหนดปลายทางของ address สำหรับการรับข้อมูล
- CRC: ทำให้ TRW 2.4GHz สามารถสร้างและถอดรหัส CRC ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

83302

PRE-AMBLE	ADDRESS	PAYLOAD	CRC

รูปที่ 2.21 ส่วนประกอบของ Data package

Configuration Word Overviews

ตารางที่ 2.3 แสดง Configuration Word

ตำแหน่งบิต	จำนวนบิต	ชื่อ	ความหมาย
143 : 120	24	TEST	จองไว้สำหรับทดสอบข้อมูล
119 : 112	8	DATA2_W	จำนวนความกว้างของบิตข้อมูล channel 2
111 : 104	8	DATA1_W	จำนวนความกว้างของบิตข้อมูล channel 1
103 : 64	40	ADDR2	แอดเดรสของตัวรับ channel 2
63 : 24	40	ADDR1	แอดเดรสของตัวรับ channel 1
23 : 18	6	ADDR_W	จำนวนบิตที่จองไว้สำหรับตัวรับ
17	1	CRC_L	จำนวนความกว้างของบิต CRC
16	1	CRC_EN	ยอมให้สร้าง CRC สำหรับ TX และมีการตรวจสอบบิต CRC สำหรับ RX
15	1	RX2_EN	กำหนดจำนวนสัญญาณ RX
14	1	CM	เลือกโหมด (Direct หรือ ShockBurst)
13	1	REDR_SB	เลือดอัตราการส่งผ่านข้อมูล (1 Mbps และ 250 Kbps)
12 : 10	3	XO_F	ความถี่คริสตอล (Crystal)
9 : 8	2	RF_PWR	เลือกค่ากำลังของตัวส่ง
7 : 1	7	RF_CH#	เลือกความถี่
0	1	RXEN	เลือกโหมดรับ/ส่ง

รายละเอียดของ Configuration Word

ในการตั้งค่าการทำงานให้กับ TRW 2.4GHz จะใช้ 144 บิต (บิตที่ 143 เป็น MSB) โดย

- การตั้งค่าการทำงานในแบบ Direct จะใช้งานบิต [15:0]
- การตั้งค่าการทำงานในแบบ ShockBurst จะใช้งานบิต [119:0]
- การตั้งค่าการทำงานสำหรับการทดสอบ จะใช้งานบิต [143:120]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดง Configuration data word

MSB	TEST						
D143	D142	D141	D140	D139	D138	D137	D136
Reserved for testing							
1	0	0	0	1	1	1	0

MSB	TEST														
D135	D134	D133	D132	D131	D130	D129	D128	D127	D126	D125	D124	D123	D122	D121	D120
Reserved for testing														Close PLL in TX	
0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0

DATA2_W							
D119	D118	D117	D116	D115	D114	D113	D112
Data width channel#2 in # of bits excluding addr/crc							
0	0	1	0	0	0	0	0

DATA1_W							
D111	D110	D109	D108	D107	D106	D105	D104
Data width channel#1 in # of bits excluding addr/crc							
0	0	1	0	0	0	0	0

ADDR2											
D103	D102	D101	....	D71	D70	D69	D68	D67	D66	D65	D64
Channel#2 Address RX (up to 40bit)											
0	0	0	...	1	1	1	0	0	1	1	1

ADDR1											
D63	D62	D61	....	D31	D30	D29	D28	D27	D26	D25	D24
Channel#1 Address RX (up to 40bit)											
0	0	0	...	1	1	1	0	0	1	1	1

ADDR_W						
D23	D22	D21	D20	D19	D18	
Address width in # of bits (both channels)						
0	0	1	0	0	0	Default

CRC		
D17	D16	
CRC Mode 1 – 16bit, 0 – 8bit	CRC 1 – enable; 0 – disable	
0	1	Default

RF-Programming															LSB
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Two Ch.	BUF	OD	XO			RF Power		Channel selection							RXEN
0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0

Default

MSB บิตจะถูกโหลดเข้าไปในคอนฟิกูเรชัน รีจิสเตอร์เป็นอันดับแรกเสมอ

Default configuration word: h8E08.1C20.2000.0000.00E7.0000.0000.E721.0F04

การตั้งค่าสำหรับการทำงานแบบ ShockBurst

ในส่วนของบิต[119:16] ประกอบด้วยส่วนของคอนฟิกูเรชัน รีจิสเตอร์ที่ใช้สำหรับการทำงานแบบ ShockBurst หลังจากที่ V_{DD} เปิดการทำงานของ ShockBurst การตั้งค่าการทำงานจะเสร็จสิ้นทันทีที่ V_{DD} ปรากฏขึ้น และระหว่างการทำงานนั้นที่ไบต์แรกของ frequency channel จะมีการเปลี่ยนเพื่อเป็น การสับเปลี่ยน RX/TX (การรับ/การส่ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PLL_CTRL

ตารางที่ 2.5 แสดงการตั้งค่า PLL

PL_CTRL		
D121	D120	PL
0	0	Open Tx/Closed Rx
0	1	Open Tx/Open Rx
1	0	Closed Tx/Closed Rx
1	1	Closed Tx/Open Rx

บิต 121-120:

PLL_CTRL: เพื่อควบคุมการตั้งค่าการทำงานของ PLL สำหรับการทดสอบ พร้อมกันกับปิดการทำงานของ PLL ในขณะที่มีการส่ง (TX) ที่ไม่ถูกต้อง

DATAx_W

ตารางที่ 2.6 แสดงจำนวนบิตของ payload

DATA2_W							
119	118	117	116	115	114	113	112

DATA1_W							
111	110	109	108	107	106	105	104

บิต 119-112:

DATA2_W: ขนาดของ RF package ในส่วนของ payload สำหรับการรับ channel ที่ 2

บิต 111-104:

DATA1_W: ขนาดของ RF package ในส่วนของ payload สำหรับการรับ channel ที่ 1

NOTE:

จำนวนบิตทั้งหมดของ RF package ในโหมด ShockBurst จะต้องไม่เกิน 256 บิต ซึ่งจำนวนสูงสุดของส่วน payload จะหาได้จาก

DATAx_W (bits) = 256 – ADDR_W – CRC

โดย ADDR_W : ความยาวของ RX address ที่จะกำหนดใน configuration word บิต [23:18]

CRC : check sum, 8 บิตหรือ 16 บิต ที่จะกำหนดใน configuration word บิต [17]

PRE : preamble, 4 บิตหรือ 8 บิต ส่วนนี้จะถูกเพิ่มเข้าไปโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDRx

ตารางที่ 2.7 Address ของตัวรับตัวที่ 2 และตัวรับตัวที่ 1

ADRR2											
103	102	101	…	71	70	69	68	67	66	65	64

ADDR1											
63	62	61	…	31	30	29	28	27	26	25	24

บิต 103-64:

ADDR2: Receiver address channel 2, สูงสุด 40 บิต

บิต 63-24:

ADDR1: Receiver address channel 1, สูงสุด 40 บิต

ADDR_W & CRC

ตารางที่ 2.8 แสดงจำนวนบิตสำรองของ RX Address + CRC

		ADDR_W				CRC_L	CRC_EN
23	22	21	20	19	18	17	16

บิต 23-18:

ADDR_W: จำนวนบิตสำรองของ RX address ใน ShockBurst package สูงสุด 40 บิต (5 ไบต์)

บิต 17:

CRC_L: ความยาวของ CRC ที่จะถูกประมวลผลใน TRW 2.4GHz ใน ShockBurst

Logic 0: 8 bit CRC

Logic 1: 16 bit CRC

บิต 16:

CRC_EN: เปิดความสามารถในการสร้าง CRC (TX) และตรวจสอบ CRC (RX)

Logic 0: ปิด On-chip CRC

Logic 1: เปิด On-chip CRC

การตั้งค่าสำหรับการทำงานแบบ Direct

Modes:

ตารางที่ 2.9 แสดงการตั้งค่าการทำงานของ RF

RX2_EN	CM	RFDR_SB	XO_F			RF_POWER	
15	14	13	12	11	10	9	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 15:

RX2_EN: Logic 0: 1 channel receive

Logic 1: 2 channel receive

ในการใช้ 2 channel receive นั้น การกำหนดความถี่ของ channel 2 จะต้องมากกว่าความถี่ของ channel 1 อยู่ 8 MHz เสมอ

บิต 14:

Communication Mode:

Logic 0: Direct Mode

Logic 1: ShockBurst Mode

บิต 13:

RF Data Rate:

Logic 0: 250 kbps

Logic 1: 1 Mbps

บิต 12-10:

Crystal Frequency:

ตารางที่ 2.10 แสดงการตั้งค่าของ Crystal

D12	D11	D10
0	1	1

บิต 9-8:

RF_PWR: ตั้งค่ากำลังในการส่ง RF output

ตารางที่ 2.11 แสดงการตั้งค่าของ RF output power

RF OUTPUT POWER		
D9	D8	P [dBm]
0	0	-20
0	1	-10
1	0	-5
1	1	0

บิต 7-1:

RF Channel & Direction

ตารางที่ 2.12 แสดงการตั้งค่าช่องสัญญาณความถี่ และ RX/TX

RF_CH#							RXEN
7	6	5	4	3	2	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RF_CH#: ตั้งค่าช่องสัญญาณความถี่ที่จะใช้ในการทำงานให้กับ TRW 2.4GHz

Channel frequency ในการส่ง:

$$Channel_{RF} = 2400MHz + [(RF_CH\#)(1.0MHz)]$$

RF_CH#: จะอยู่ระหว่าง 2400MHz และ 2527MHz

Channel frequency ใน data channel 1:

$$Channel_{RF} = 2400MHz + [(RF_CH\#)(1.0MHz)]$$

RF_CH#: จะอยู่ระหว่าง 2400MHz และ 2524MHz

Channel frequency ใน data channel 2:

$$Channel_{RF} = 2400MHz + [(RF_CH\#)(1.0MHz)] + 8MHz$$

RF_CH#: จะอยู่ระหว่าง 2400MHz และ 2524MHz

บิต 0:

Set Active Mode: Logic 0: transmit mode

Logic 1: receive mode

คำอธิบาย DATA package

Data package ของทั้ง ShockBurst mode และ Direct mode จะประกอบไปด้วย 4 ส่วนดังนี้

ตารางที่ 2.13 อธิบาย Data package

1. PREAMBLE	- ส่วน preamble นั้นจำเป็นต้องใช้สำหรับ ShockBurst mode
2. ADDRESS	- ส่วน address นั้นจำเป็นต้องใช้สำหรับ ShockBurst mode - มีขนาด 8 ถึง 40 บิต - address นั้นจะถอดออกจากแพคเกจที่ได้รับมาโดยอัตโนมัติใน ShockBurst mode
3. PAYLOAD	- เป็นข้อมูลที่จะส่งออกไป - ใน ShockBurst mode ขนาด payload คือ 256 บิต โดยไม่รวมค่าต่อไปนี้ (Address: 8 ถึง 40 บิต รวมกับ CRC 8 หรือ 16 บิต)
4. CRC	- มีขนาด 8 หรือ 16 บิต - CRC จะถูกถอดออกจากข้อมูลที่ได้รับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ไมโครคอนโทรลเลอร์ MCS-51

2.3.1 คุณสมบัติพื้นฐานของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
2. ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
3. หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรมในบางเบอร์ จะมีหน่วยความจำแบบ EPROM เพิ่มเติม
4. ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุทและเอาต์พุท
5. มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
6. ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
7. สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
8. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
9. มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิพ

2.3.2 พอร์ทของ MCS-51

- พอร์ท 0 (P0.0 – P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้ปืนได้ทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ท 0 ขาใดขาหนึ่งเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ทที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ทนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุทอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ทอินพุทได้ นอกจากนั้นขาพอร์ทนี้ถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0 - A7) และขาข้อมูล (D0 – D7) โดยใช้กระบวนการมัลติเพลกซ์เข้าช่วยเพื่อสลับการทำงาน เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

- พอร์ท 1 (P1.0 - P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ทใดเป็นอินพุท สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ทที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุทสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุททริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

- พอร์ท 2 (P2.0 - P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ทใดเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ทที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ทนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุทอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ทอินพุทได้ นอกจากนั้นขาพอร์ทนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

- พอร์ท 3 (P3.0 - P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ทใดเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ทที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ทนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุทอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ทอินพุทได้ นอกจากนั้นขาพอร์ท3 ยังเป็นขาที่มีหน้าที่การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้งานพิเศษดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.1 ใช้เป็นขาอินพุทสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา INTO

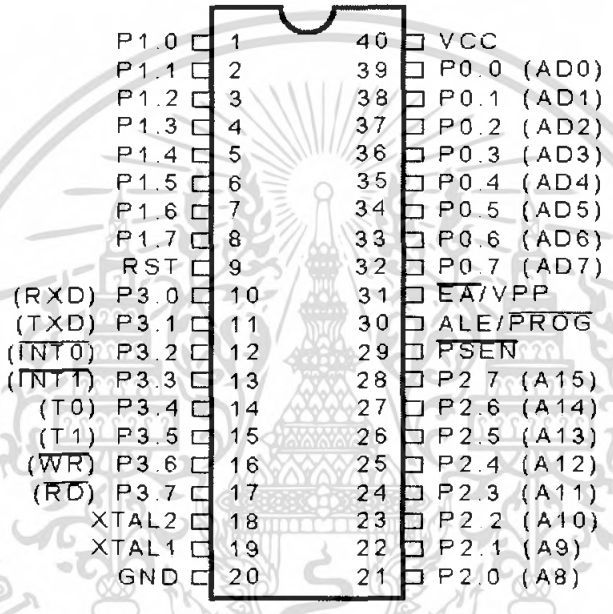
P3.3 ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา INT1

P3.4 ใช้เป็นขาอินพุทสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือ T0

P3.5 ใช้เป็นขาอินพุทสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือ T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก



รูปที่ 2.22 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51

- ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเคิลโดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

- ขา ALE/PROG (Address Latch Enable/Program Pulse Input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

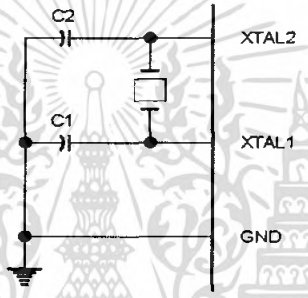
- ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภาย นอกตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดออกมา

- ขาEA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้มีสถานะเป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุทสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดัน สำหรับการโปรแกรม +12V

- ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์



รูปที่ 2.23 แสดงการต่อสัญญาณนาฬิกาที่ขา XTAL1 และ XTAL2

2.3.3 ประเภทของหน่วยความจำ

หน่วยความจำที่ใช้กับ MCS-51 มี 2 ชนิด คือ

หน่วยความจำโปรแกรม (Program Memory)

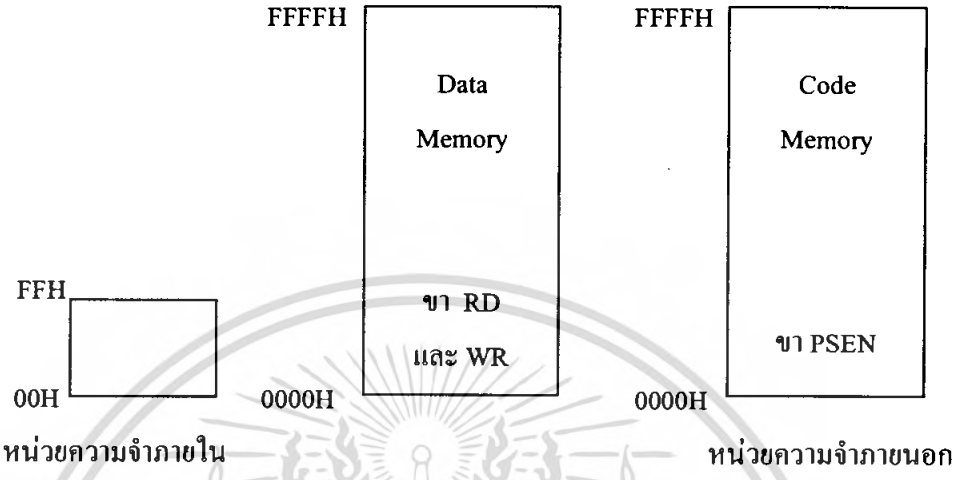
หน่วยความจำโปรแกรม(Program Memory) มีไว้เพื่อบรรจุคำสั่งหรือโปรแกรมที่ผู้ใช้พัฒนาขึ้นจัดเก็บไว้ในหน่วยความจำ โดยอาจจะประกอบอยู่ภายในตัวของไอซี 89C51 เองหรือเป็นไอซีหน่วยความจำ EPROM หรือ ROM แยกออกต่างหากได้ ในกรณีหลังจำเป็นต้องมีการใช้พอร์ทอินพุทเอาท์พุททำหน้าที่เป็นบัสแอดเดรส และบัสข้อมูลเพื่อให้สามารถติดต่อกับหน่วยความจำมาตรฐานทั่วไปได้

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

การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ 89C51 จะมีการจัดพื้นที่ดังรูปที่ 2.24 โดยที่ไมโครคอนโทรลเลอร์ 89C51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมสูงสุดได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะตามตำแหน่งของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

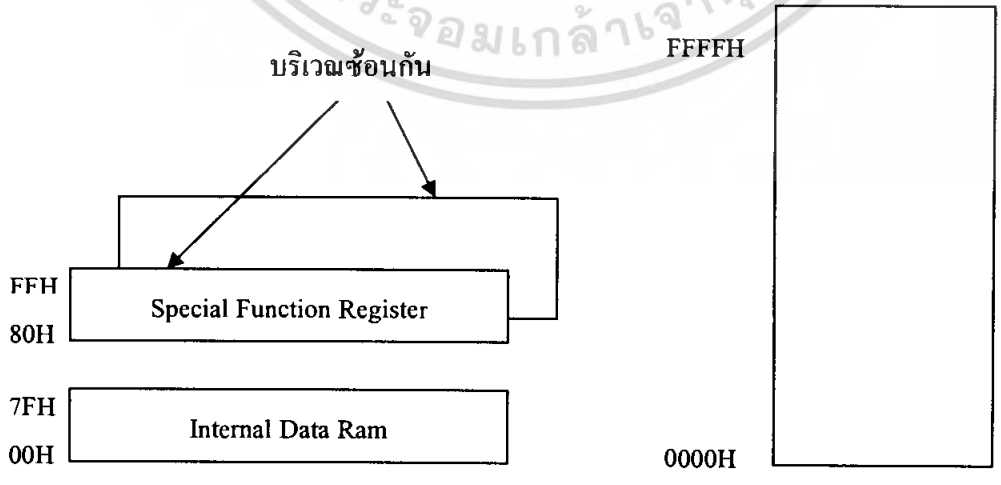
นั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ภายในตัวไอซี ไมโครคอนโทรลเลอร์เองและหน่วยความจำภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ



รูปที่ 2.24 การจัดพื้นที่หน่วยความจำของไมโครคอนโทรลเลอร์

หน่วยความจำข้อมูลของ MCS-51

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยที่หน่วยความจำข้อมูลจะมีลักษณะเป็นหน่วยความจำ RAM แบบสแตติก (Static) ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับระบบ ก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ในหน่วยความจำนี้สูญหายไป พื้นที่ของหน่วยความจำข้อมูลของ 89C51 สามารถมีได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตำแหน่งที่ตั้งของหน่วยความจำนั้นดังลักษณะในรูปที่ 2.25 คือ หน่วยความจำโปรแกรมภายใน (Internal Date Memory) หรือ RAM ที่อยู่ภายในตัวไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจรลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 2.25 แสดงลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register: SFR)

ใน AT89C51 มี SFR ทั้งหมด 21 ตัวโดยเริ่มจากตำแหน่ง 80H ถึง FFH ประกอบไปด้วย

1. Accumulator: ACC

มีขนาด 8 บิต เป็นแอกคิวมูเลเตอร์คำสั่งส่วนใหญ่จะอ้างถึงตัวรีจีสเตอร์นี้ โดยถือค่าภายในเป็นค่าตัวตั้ง และรับค่าผลลัพธ์ที่ได้จากคำสั่งทางคณิตศาสตร์ มาเก็บไว้ ตัว ACC ยังสามารถใช้เป็นตัวแหล่งกระทำหรือถูกกระทำในการทำงานทางตรรกะ และใช้เป็นตัวกลางในการถ่ายเทข้อมูลในการติดต่อกับอุปกรณ์ภายนอก I/O และ หน่วยความจำภายนอก รวมถึงการตรวจสอบตารางข้อมูล

2. Program Status Word: PSW

จะอยู่ที่ตำแหน่ง D0H สามารถเข้าถึงข้อมูลระดับบิตได้ โดยรีจีสเตอร์ตัวนี้จะเป็นตัวบอกสภาวะต่างๆ ของ ไมโครคอนโทรลเลอร์ โดยจะมีทั้งหมด 8 บิต

3. Register B

เป็นรีจีสเตอร์พิเศษที่ใช้งานสำหรับคำสั่งของการคูณและการหาร โดยใช้เป็นที่เก็บตัวคูณหรือตัวและเป็นตัวเก็บผลลัพธ์ตัวที่สองหลังการคูณและหาร

4. Stack Pointer: SP

เป็นรีจีสเตอร์ขนาด 8 บิตอยู่ที่ตำแหน่ง 81H การเขียนค่าเข้าไปในตำแหน่งที่ SP ชี้อยู่นี้เรียกว่า pushing และการอ่านค่าที่ SP ชี้อยู่เรียกว่า Popping ค่าของ SP จะเพิ่มขึ้นหนึ่งก่อนที่จะเขียนข้อมูลลงไปและจะลดลงหนึ่งเมื่ออ่านข้อมูลออกมาแล้ว หากโปรแกรมทำคำสั่ง CALL จะใช้รีจีสเตอร์ SP นี้เป็นค่าตำแหน่งเดิมของโปรแกรม (PC) ก่อนที่จะทำโปรแกรมย่อยเมื่อทำโปรแกรมย่อยเสร็จแล้ว จะคืนค่าในสแตกให้กับ PC ตามเดิม โดยปกติค่า PC จะ กำหนดให้อยู่ใน RAM ภายใน

5. Data Pointer Register: DPTR

DPTR เป็นรีจีสเตอร์ขนาด 16 บิตที่ประกอบไปด้วยไบต์สูง (DPH) และไบต์ต่ำ (DPL) ที่สามารถเลือก แบ่งออกเป็นรีจีสเตอร์ 8 บิตสองตัวใช้ได้อย่างอิสระหรือจะใช้รวมกันทั้ง 16 บิตก็ได้ ในการ Increment หรือ Decrement เพื่อประโยชน์ในการใช้เป็นฐานของเลขที่อยู่ในรีจีสเตอร์ในการกระโดดโดยทางอ้อมในการใช้คำสั่ง เกี่ยวกับตารางข้อมูลและชี้ตำแหน่งของหน่วยความจำภายนอก

6. Port Register

รีจีสเตอร์ P0, P1, P2 และ P3 ของ SFR เป็นรีจีสเตอร์ที่ใช้ในการ Latch ค่าของ port0 ถึง port3

7. Program Counter : PC

เป็นรีจิสเตอร์ 16 บิตที่ทำหน้าที่ระบุตำแหน่งของ Program Memory

8. Serial Port Register : SBUF

บัฟเฟอร์ข้อมูลอนุกรมแบ่งออกเป็นรีจีสเตอร์ 2 ตัว ตัวหนึ่งเป็นบัฟเฟอร์การส่งอีกตัวหนึ่งเป็นบัฟเฟอร์การรับเมื่อมีขนาด 8 บิต

9. Power Control Register: PCON

จะอยู่ที่ตำแหน่ง 87H ใช้ในการหยุดการทำงาน MCS-51 โดยจะหยุดจ่ายสัญญาณนาฬิกาให้ระบบทำให้ข้อมูลต่างๆภายใน MCS-51 ไม่มีการเปลี่ยนแปลงนอกจากนี้ยังช่วยลดพลังงานไฟฟ้าที่จ่ายให้ MCS-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

51 ลงด้วย

2.3.5 การรับส่งข้อมูลอนุกรม (UART)

รีจิสเตอร์ต่างๆที่เกี่ยวข้องในการใช้งานพอร์ทอนุกรม

1. รีจิสเตอร์ควบคุมไทม์เมอร์เนื่องจากว่าการใช้งานพอร์ทอนุกรมนั้นมีสิ่งที่ต้องคำนึงถึงคือ อัตราการรับ-ส่งข้อมูลหรือเรียกว่าอัตราบอด (Baud Rate) จริงๆแล้วก็คือจังหวะการเลื่อนข้อมูลเข้าหรือออกจาก MCS-51 นั่นเอง โดยอัตราบอดนี้สามารถสร้างขึ้นภายในชิพของ MCS-51 ได้จากไทม์เมอร์แชนแนล 1 โดยทำงานในโหมด 2 คือ โหลดค่ากลับอัตโนมัติ 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 2.14 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 แฟลกซ์ใช้งานทั่วไป ไม่เกี่ยวข้องกับการควบคุมใด ๆ

GF2 แฟลกซ์ใช้งานทั่วไป ไม่เกี่ยวข้องกับการควบคุมใด ๆ

PD Power down Bit 1 = หยุดออสซิลเลเตอร์ของซีพียูสัญญาณรีเซตหรืออินเตอร์รัพต์ (83C154, 83C154D เท่านั้น) ที่จะยกเลิกโหมดนี้

IDL Idle Mode Bit 1 = หยุดการทำงานของซีพียูสัญญาณรีเซตหรืออินเตอร์รัพต์เท่านั้นที่จะยกเลิกโหมดนี้

3. รีจิสเตอร์ควบคุมการอินเตอร์รัพต์ เนื่องจากว่า MCS-51 สามารถใช้งานพอร์ทอนุกรมในลักษณะของการอินเตอร์รัพต์ จึงมีรีจิสเตอร์ที่เกี่ยวข้องดังนี้

- IE ตำแหน่ง A8H ทำหน้าที่ยอมให้เกิดการอินเตอร์รัพต์จากพอร์ทอนุกรมหรือไม่
- IP ตำแหน่ง B8H ทำหน้าที่จัดลำดับความสำคัญของการอินเตอร์รัพต์

4. รีจิสเตอร์ควบคุมพอร์ทอนุกรม การใช้งานพอร์ทอนุกรมจะขึ้นอยู่กับรีจิสเดอร์โดยตรงคือ

- SBUF ตำแหน่ง 99H ทำหน้าที่เป็นบัฟเฟอร์การรับหรือส่งข้อมูลของพอร์ทอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SCON ตำแหน่ง 98H ทำหน้าที่ควบคุมและกำหนดโหมดการใช้งานพอร์ทอนุกรมทั้งหมดซึ่งมีรายละเอียดของบิตต่าง ๆ ดังนี้

ตารางที่ 2.15 แสดงบิตภายใน SCON

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

SM0, SM1 บิตเลือกโหมดการทำงาน

ตารางที่ 2.16 แสดงการเลือกโหมดการใช้งานพอร์ทอนุกรม

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

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 (Transmit bit D8) ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2, 3 ให้ใส่ในบิตนี้

RB8 (Receive bit D8) ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมาเก็บในบิตนี้

(ข้อมูลบิตที่ 9 ก็คือค่าใน TB8 ทางด้านส่งนั่นเอง)

TI แฟลกซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

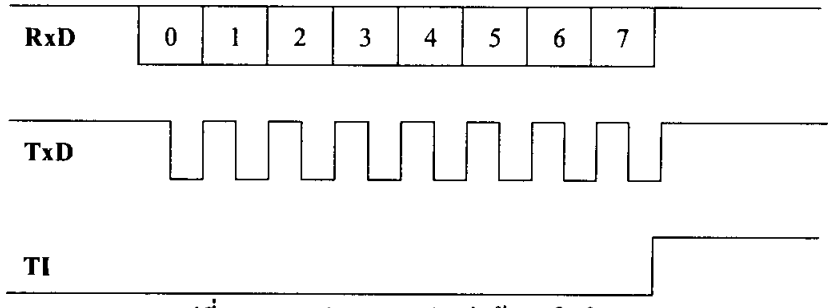
RI แฟลกซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมจะต้องเคลียร์เอง)

โหมดในการส่งข้อมูลแบบอนุกรม

พอร์ทสื่อสารอนุกรมของ MCS-51 สามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกที่บิต SM1 และ SM2 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมด

- โหมด 0: พอร์ทสื่อสารอนุกรม 8 บิต โดยจะส่งข้อมูลเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อน ทางขา RxD เนื่องจากไม่มีการส่ง Start bit แต่จะส่ง shift clock ทางขา TxD [ความเร็ว (1/12) เท่าของ CPU Clock]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

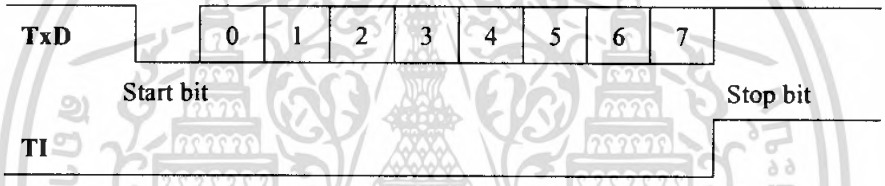


รูปที่ 2.26 รูปแบบการรับส่งข้อมูลในโหมด 0

- โหมด 1: พอร์ทสื่อสารอนุกรม 10 บิต ข้อมูล 8 บิต 1 start bit และ 1 start bit และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นอยู่กับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของ Timer 1, 2

$$BaudRateMode1,3 = \frac{2^{SMOD} \times CPUOSC}{32 \times 12 \times [256 - (TH1)]}$$ โดยใช้ Timer 1

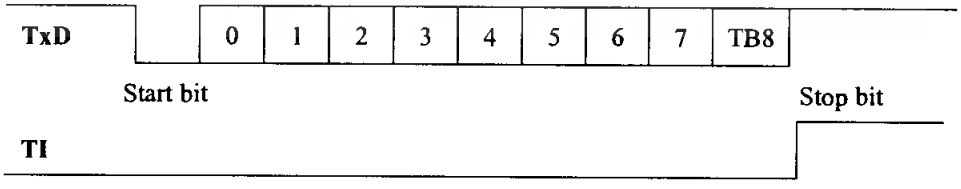
$$BaudRateMode1,3 = \frac{CPUOSC}{32 \times [65536 - (RCAP2H, RCAP2L)]}$$ โดยใช้ Timer 2



รูปที่ 2.27 รูปแบบการรับส่งข้อมูลในโหมด 1

- โหมด 2: พอร์ทสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 start bit และ 1 stop bit (TB8 นิยมนำมาใช้ส่ง Parity bit) ความเร็วในการรับส่งข้อมูลเท่ากับ (1/32) หรือ (1/64) เท่าของ CPU OSC โดยขึ้นกับบิต SMOD ใน PCON

- $BaudRate(Mode2) = (1/32)CPUOSC$ เมื่อ SMOD = 1
- $BaudRate(Mode2) = (1/64)CPUOSC$ เมื่อ SMOD = 0



รูปที่ 2.28 รูปแบบการรับส่งข้อมูลในโหมด 2

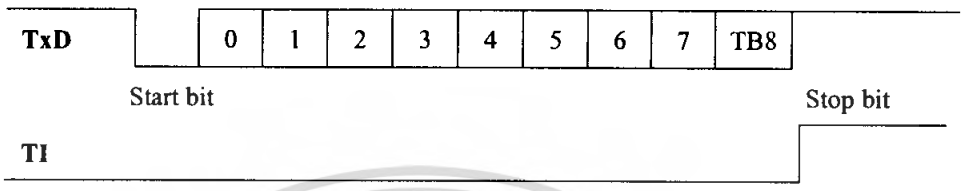
- โหมด 3: พอร์ทสื่อสารอนุกรม 11 bit UART โดย DATA 8 bit 1 start bit และ 1 stop bit เหมือนโหมด 2 ยกเว้นอัตราเร็ว จะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของ Timer 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ 8051

$$BaudRateMode3 = \frac{2^{SMOD} \times CPUOSC}{32 \times 12 \times [256 - (TH1)]}$$ โดยใช้ Timer 1

$$BaudRateModde = \frac{CPUOSC}{32 \times [65536 - (RCAP2H, RCAP2L)]}$$ โดยใช้ Timer 2



รูปที่ 2.29 รูปแบบการรับส่งข้อมูลในโหมด 3

2.4 แอลซีดี

2.4.1 รายละเอียดเกี่ยวกับโมดูลแอลซีดี

ในโมดูลแอลซีดีจะมีส่วนประกอบหลักๆ 3 ส่วน ดังนี้

ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอแอลซีดี

ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล แอลซีดี เช่น ลบจอภาพ แสดงตัวอักษรหรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิพควบคุมโดยเฉพาะ ชิพที่นิยมใช้คือเบอร์ HD44780 และ HD61830 โดย HD44780 ใช้ควบคุมแอลซีดีแบบอักขระ ส่วน HD61830 ใช้ควบคุมแอลซีดีกราฟฟิก

ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิพที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

2.4.2 โครงสร้างภายในของตัวควบคุมโมดูลแอลซีดี

ในการใช้งานโมดูลแอลซีดีจำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อนในที่นี้จะขอยกตัวอย่างโมดูลแอลซีดีแบบอักขระ เพราะสามารถทำความเข้าใจได้ง่าย ประกอบด้วย

บัฟเฟอร์อินพุทเอาท์พุท เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register: IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมการแสดงผล

รีจิสเตอร์ข้อมูล (Data Register: DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอก เพื่อส่งต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผลหรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัว อักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

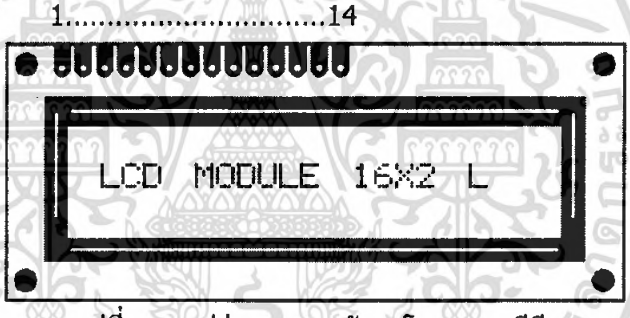
แรมเก็บข้อมูลแสดงผล (Display Data RAM: DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up-label) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงผลที่ตัวแสดงผล

รอมเก็บตัวอักษร (Character Generator ROM: CGROM) เป็นหน่วยความจำที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร (Character Generator RAM: CGRAM) เป็นหน่วยความจำแรมความจุ 512 บิต ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือ เขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

โมดูลแอลซีดี ขนาด 16 ตัวอักษร 2 บรรทัด (LCD 16x2) มีลักษณะดังรูปที่ 2.30



รูปที่ 2.30 รูปร่างและการจัดขาโมดูลแอลซีดี\

โมดูลแอลซีดีขนาด 16x2 มีขาต่อใช้งานทั้งสิ้น 14 ขา รายละเอียดการทำงานของแต่ละขามีดังนี้

V_{SS} (ขา 1): ต่อกราวด์

V_{DD} (ขา 2): ต่อไฟเลี้ยง +5V

V_O (ขา 3): เป็นขาอินพุทรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา 4): เป็นขาอินพุทใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น "0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/W (ขา 5): เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับโมดูลแอลซีดีถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น "1" จะเป็นการอ่านข้อมูล

E (ขา 6): เป็นขาสำหรับรับสัญญาณพัลส์เอ็นเอเบิลโมดูลแอลซีดีให้ทำงาน

D0-D7 (ขา 7-14): เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่างแอลซีดีกับอุปกรณ์ภายนอกขนาด 8 บิต อนึ่งขา RS, R/W และ E จะใช้งานร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งควบคุมโมดูลแอลซีดี

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น "0" แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูลแอลซีดีของชิพควบคุม HD44780 ที่สำคัญมี 10 คำสั่งดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล (Clear Display)

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่างหรือ space เข้าไปใน

DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิคิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D ให้เป็น "1"

2. คำสั่ง Return Home

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น "1" เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode Set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

ตารางที่ 2.17 รายละเอียดรูปแบบโหมดการป้อนข้อมูลของแอลซีดี

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	1	I/D	S

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น "1" เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น "0" เมื่อเกิดข้อมูลใหม่ดัวเคอร์เซอร์จะเลื่อนไปทางซ้ายมือ

บิต I/D เป็นบิตที่ใช้กำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น "1" แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น "0" แอดเดรสจะลดลง

ดังนั้น ข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H (ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

4. คำสั่งควบคุมการแสดงผล

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

ตารางที่ 2.18 รายละเอียดรูปแบบคำสั่งควบคุมการแสดงผล

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	1	D	C	B

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น "1" จะเป็นการเปิดจอแสดงผล ถ้าเป็น "0" จะเป็นการปิดจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น "1" ถ้ากำหนดให้เป็น "0" จะเป็นการปิดเคอร์เซอร์หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น "1" เคอร์เซอร์จะกระพริบ ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H ถึง 0FH (8 คำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กระพริบ

5. คำสั่งควบคุมการเลื่อนคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

ตารางที่ 2.19 รายละเอียดรูปแบบคำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผล ขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถกำหนดได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H ถึง 13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H ถึง 17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H ถึง 1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH ถึง 1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

ตารางที่ 2.20 รายละเอียดรูปแบบคำสั่งกำหนดฟังก์ชันการทำงาน

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	1	DL	N	F	*	*

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น "0" จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น "1" จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล

ถ้าเป็น "0" จะแสดงผล 1 บรรทัด

ถ้าเป็น "1" จะแสดงผล 2 บรรทัดในกรณีจอแสดงผลสามารถได้มากกว่า 2 บรรทัดและต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดที่น่าสังเกตคือ โมดูลแอลซีดีแบบ 16 ตัวอักษร 1 บรรทัด ที่มีขายในประเทศไทย แม้จะแสดงผลเพียง 1 บรรทัดแต่ต้องกำหนด N ให้เป็น "1" เนื่องจากแอดเดรสของ DDRAM มี 2 ช่วงคือ 00H และ 40H

บิต F ใช้เลือกความละเอียดของตัวอักษรให้การแสดงผล ถ้าบิตนี้เป็น "0" จะเป็นการแสดงผลแบบ 5x7 จุด และถ้าเป็น "1" จะแสดงผลเป็นแบบ 5x10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดูลแอลซีดีทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5x7 จุด

7. คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น "0" บิต 6 เป็น "1" ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

8. คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น "1" และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH ถึง 0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น "0" แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H ถึง 0CFH และถ้าบิต N เป็น "1" แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 8CH ถึง 87H และ 0C0H ถึง 0CFH

9. คำสั่งอ่านแฟลก BUSY และแอดเดรส

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

ตารางที่ 2.21 รายละเอียดรูปแบบคำสั่งอ่านค่าแฟลก BUSY และแอดเดรส

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
BF	A	A	A	A	A	A	A
	← แอดเดรสไบต์ต่ำ →			← แอดเดรสไบต์สูง →			

เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุมโมดูลแอลซีดีว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น "0" แสดงว่าตัวควบคุมโมดูลแอลซีดีพร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น "1" แสดงว่า ขณะนี้ตัวควบคุมแอลซีดียังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น "1" ด้วย แต่สัญญาณที่ RS ยังต้องเป็น "0" อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง นอกจากนี้ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต 0-6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.4.3 การเขียนคำสั่งและข้อมูลให้แก่โมดูลแอลซีดี

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูลแอลซีดีแสดงผลตามที่ผู้ใช้งานต้องการ ต้องส่งคำสั่ง (instruction) แล้วกำหนดโหมดการทำงานให้แก่โมดูลแอลซีดีก่อน จากนั้นจึงค่อยส่งข้อมูล (data) ที่ต้องการแสดงผล เนื่องจากบัสข้อมูลของโมดูลแอลซีดีมี 8 เส้นคือ D0 ถึง D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ได้ลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"0" หมายความว่าข้อมูลที่ป้อนให้แก่โมดูลแอลซีดีขณะนั้นเป็นคำสั่ง ในทางตรงข้าม หากขา RS ได้รับลอจิก "1" ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น "1" เพื่อแจ้งให้ตัวควบคุมภายในโมดูลแอลซีดีทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีอ่านข้อมูล ต้องกำหนดให้ขา R/W เป็น "1" ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

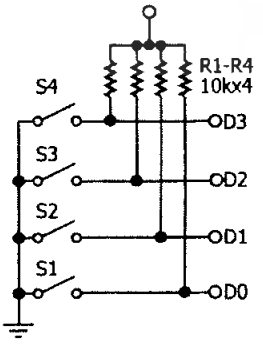
ในกรณีเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก "1" ให้ขา RS แล้ว ต้องกำหนดให้ขา R/W เป็น "0" ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

2.5 Matrix – Switch

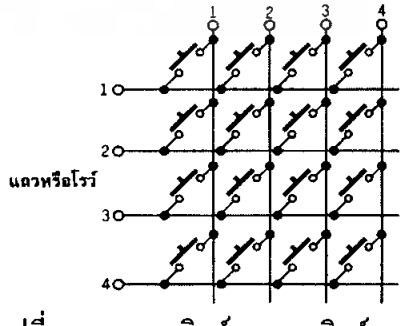
ทฤษฎีเบื้องต้น

การอ่านค่าหรือรับค่าการกดสวิตช์เป็นอีกอย่างหนึ่งที่ไมโครคอนโทรลเลอร์ต้องสามารถรองรับและเชื่อมต่อใช้งานร่วมด้วยได้วงจรของสวิตช์มีด้วยกัน 2 ลักษณะใหญ่ๆ คือ ต่อเข้ากับไฟเลี้ยงหรือกราวด์โดยตรง เมื่อสวิตช์ตัวใดต่อวงจรสามารถอ่านค่าได้โดยตรง ดังในรูป 2.31 วงจรในลักษณะนี้ไม่มีความซับซ้อนสามารถอ่านค่าของสวิตช์ได้ง่ายและรวดเร็ว แต่มีข้อเสียคือถ้าหากจำนวนของสวิตช์มีมากจำนวนของสายข้อมูลก็จะมีมากตาม ทำให้ระบบหรือวงจรโดยรวมมีขนาดใหญ่และสิ้นเปลือง

วงจรของสวิตช์อีกลักษณะหนึ่งคือ การต่อวงจรแบบเมตริกซ์ (Matrix Switch) ดังในรูปที่ 2.32 สวิตช์จะถูกต่อกันในแนวแกนตั้งและแกนนอน จะเรียกแนวตั้งว่า หลักหรือคอลัมน์ (column) ในขณะที่แนวนอนจะเรียกว่า แถวหรือโรว์ (row) ดังนั้นค่าของสวิตช์จะต้องประกอบด้วย ตำแหน่งในแนวหลักและแถว กระบวนการที่จะทำให้ได้มาซึ่งค่าของสวิตช์มีขั้นตอนซับซ้อนพอสมควร แต่วงจรของสวิตช์แบบนี้มีข้อดีคือสามารถรองรับการเพิ่มของสวิตช์ได้อย่างสะดวก เพียงเพิ่มเติมจำนวนสวิตช์และแก้ไขซอฟต์แวร์อีกเล็กน้อยเท่านั้น ทำให้วงจรสวิตช์แบบเมตริกซ์เป็นที่นิยมใช้มากในระบบควบคุมอัตโนมัติที่มีจำนวนสวิตช์มากกว่า 8 ตัว ในการใช้งานทั่วไปจะเรียกสวิตช์แบบเมตริกซ์นี้ว่า คีย์แพด (Keypad)



รูปที่ 2.31 วงจรสวิตช์แบบต่อเข้ากับไฟเลี้ยงและกราวด์

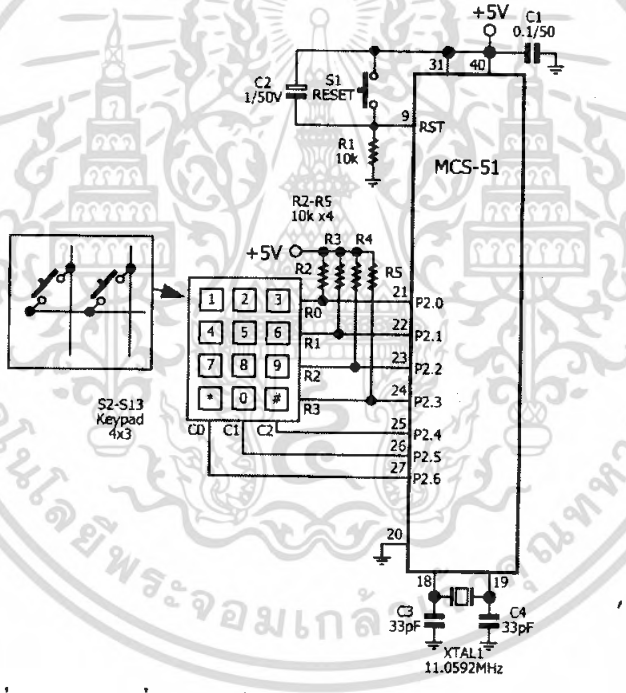


รูปที่ 2.32 วงจรสวิตช์แบบเมตริกซ์หรือคีย์แพด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

มีตัวอย่างวงจรแสดงในรูปที่ 2.33 จะใช้พอร์ท 2 ของไมโครคอนโทรลเลอร์เชื่อมต่อเข้ากับคีย์แพดทั้ง 7 เส้นคือ สายของคอลัมน์ 3 สายของ C0-C2 และสายทางโรว์หรือแถวอีก 4 สายคือ R0-R3 โดยเฉพาะที่ขาพอร์ท P2.0-P2.3 จะต้องต่อตัวต้านทานพูลอัปไว้เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์ โดยไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล "0" ไปยัง P2.6, P2.5 และ P2.4 ตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายคอลัมน์ของคีย์แพดไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ P2.0-P2.3 เข้ามาด้วย หากไม่มีการกด ค่าของ P2.0-P2.3 ก็จะเป็น "1" ทั้งหมด ถ้าหากมีการกดคีย์ ค่าของ P2.0-P2.3 ก็จะไม่เป็น 1111 อีกต่อไป เป็นการแจ้งให้ทราบว่า มีการกดคีย์แพดขึ้นแล้ว จากนั้นไมโครคอนโทรลเลอร์ก็จำการค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งสิ่งที่จะได้มาอย่างแรกคือ ค่าตำแหน่งของคีย์นั้น จากนั้นก็จะนำค่าตำแหน่งนั้นไปเปิดตารางข้อมูล เพื่อจะได้หมายเลขของคีย์ที่กดอย่างแท้จริง ยกตัวอย่างจากวงจรในรูปที่ 2.33 หากคีย์ 0 ถูกกด ไมโครคอนโทรลเลอร์จะได้ค่าตำแหน่งของคีย์ 0 เป็น 0BH จากนั้นนำค่า 0BH นี้ไปเปิดตารางก็จะได้ข้อมูลเป็น 0 หมายถึงคีย์ 0 ซึ่งกระบวนการหลังนี้จะเป็นหน้าที่ของซอฟท์แวร์



รูปที่ 2.33 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

2.6 Visual Studio .NET

Visual Studio .NET ถือเป็นเครื่องมือพัฒนาแอพพลิเคชันตัวแรกที่ออกมารองรับการทำงานตามแนวความคิดของสถาปัตยกรรม .NET Framework ซึ่งประกอบไปด้วยเครื่องมือต่าง ๆ ดังนี้

- Visual Basic .NET
- Visual C++ .NET
- Visual C# .NET
- ASP .NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงงานนี้จะใช้ภาษา Visual C# ซึ่งถือเป็นภาษาที่เกิดขึ้นมาพร้อมกับแนวความคิดของการเขียนโปรแกรมในยุค .NET อาจจะกล่าวได้ว่าภาษา Visual C# คือ ภาษาต้นแบบของการเขียนโปรแกรมใน .NET นั่นเอง โดยที่ภาษาอื่น ๆ ที่เกิดขึ้นมาก่อนหน้านี้ จะต้องปรับตัวเข้าหา .NET ทั้งหมด จะสังเกตได้ว่าไวยากรณ์การใช้งานของแต่ละภาษานั้น ล้วนแล้วแต่ถูกปรับเปลี่ยนไปจากเวอร์ชันก่อนหน้านี้

สำหรับภาษา Visual C# การเขียนโปรแกรมเชิงวัตถุจะไม่ใช่เรื่องที่ยุ่งยากอีกต่อไป โดยที่ภาษา Visual C# จะอาศัยไวยากรณ์ที่ปรับปรุงมาจาก C/C++ ร่วมกับความง่ายของภาษา Visual Basic จนกลายมาเป็นภาษา Visual C#

2.6.1 ความสามารถของ Visual C#

Visual C# นั้นมีความสามารถมากมาย ได้รับการต้อนรับเป็นอย่างดีจากนักพัฒนาแอพพลิเคชันทั่วโลก รวมทั้งเมืองไทยด้วย ซึ่งจะเห็นได้จากการนำ Visual C# ไปประกอบการเรียนการสอน การฝึกอบรม ตลอดจนการนำไปสร้างเป็นซอฟต์แวร์เชิงพาณิชย์จำนวนมาก

2.6.1.1 สร้างแอพพลิเคชันสำหรับ Windows

Visual C# ได้ชื่อว่าเป็นเครื่องมือสำหรับพัฒนาแอพพลิเคชันที่ทำงานบน Windows ที่ใช้งานกันแพร่หลายมาก สามารถสร้างงานได้อย่างรวดเร็ว นิยมนำไปสร้างแอพพลิเคชันทั้งเพื่อการศึกษา และแอพพลิเคชันที่ใช้ในระบบงานจริง ๆ ในโลกธุรกิจ

2.6.1.2 สร้างระบบงานด้านฐานข้อมูล

ถ้าจะถามว่า Visual C# ถูกนำไปสร้างผลงานอะไรมากที่สุด หนีไม่พ้นเรื่องของการพัฒนาระบบงานด้านฐานข้อมูล โดย Visual C# มีจุดเด่นในการสร้างแอพพลิเคชันที่ติดต่อกับฐานข้อมูลได้หลายรูปแบบ มีวิธีการและเครื่องไม้เครื่องมือในการสร้างระบบงาน ระบบบริหารงานบุคคลหรือแม้แต่ระบบระบบคลังสินค้า

แอพพลิเคชันที่เกี่ยวกับฐานข้อมูลที่สร้างจาก Visual C# สามารถนำไปใช้งานกับระบบฐานข้อมูลชั้นนำแทบทุกชนิดทั่วโลก ทั้ง Access, Oracle และ SQL เซิร์ฟเวอร์ รวมถึงระบบไฟล์ชนิดต่าง ๆได้ด้วย

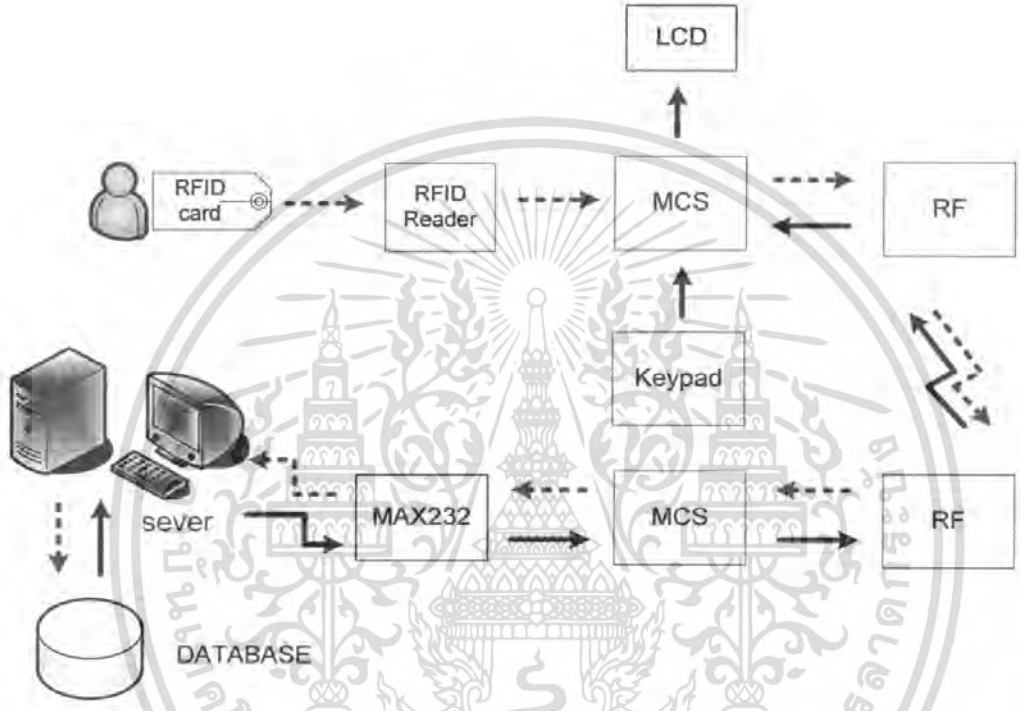
2.6.1.3 สร้างแอพพลิเคชันรองรับ .NET Web Service

แนวโน้มของการพัฒนาแอพพลิเคชันในยุคหน้าที่นี้ต้องเตรียมตัวให้พร้อมก็คือ การพัฒนาแอพพลิเคชันด้วยเทคโนโลยี .NET ซึ่งจะช่วยให้แอพพลิเคชันชนิดต่าง ๆไม่ว่าจะทำงานบนพีซี โน๊ตบุ๊ค PDA หรือแม้แต่โทรศัพท์มือถือ สามารถเชื่อมโยงข้อมูล และคุยกันได้ใน Visual C# นั้นรองรับการพัฒนาแอพพลิเคชันที่เรียกว่า .NET Web Service แล้ว โดยสามารถใช้เครื่องมือชนิดต่าง ๆ พัฒนาแอพพลิเคชันตามแนวความคิดของ .NET เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

3.1 บล็อกไดอะแกรม



รูปที่ 3.1 บล็อกไดอะแกรมของปริญญานิพนธ์

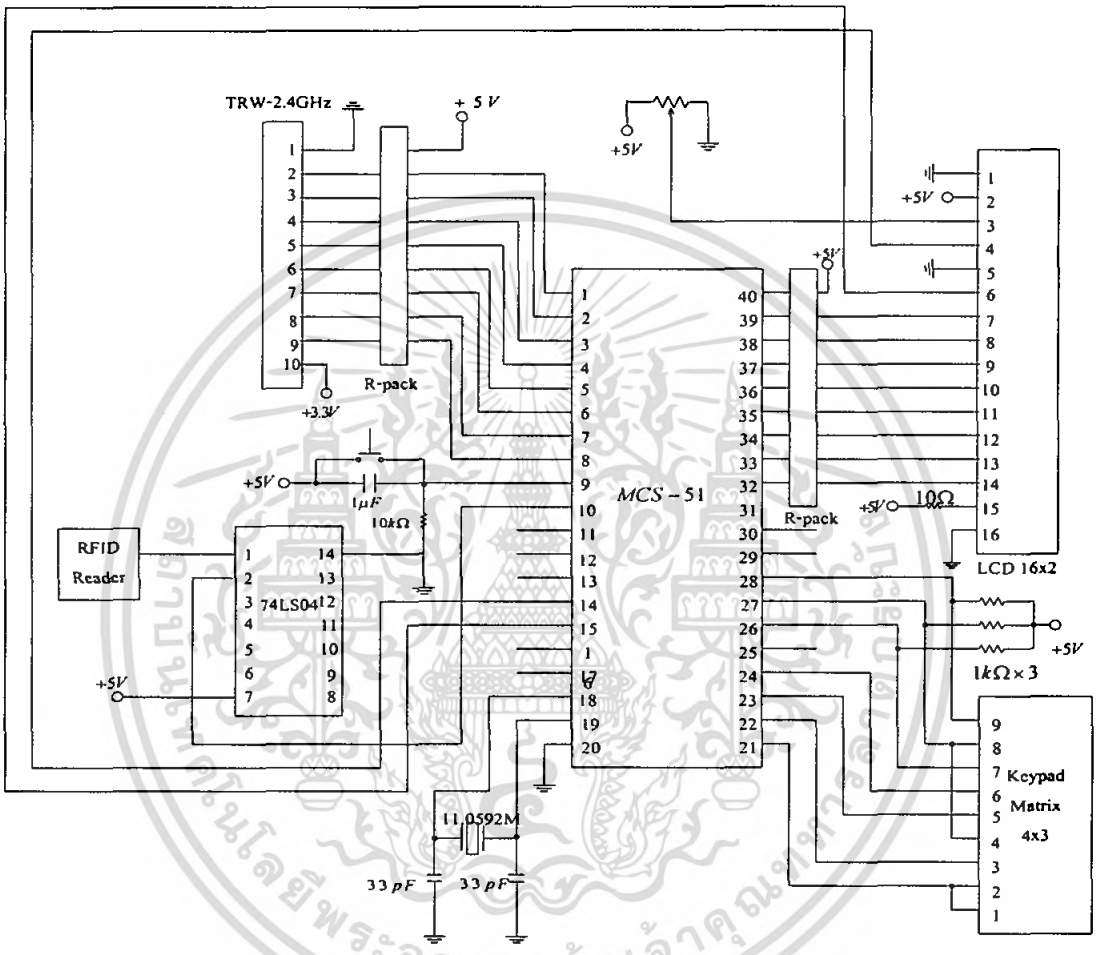
ในปริญญานิพนธ์นี้จะใช้หัวอ่าน RFID เป็นตัวอ่านรหัสที่อยู่ในบัตร แล้วทำการส่งข้อมูลไปให้วงจรซึ่งใช้ไมโครคอนโทรลเลอร์ด้านร้านค้า เบอร์ AT89C51 ซึ่งเป็นไอซี 40 ขา ทำหน้าที่ควบคุมการทำงานของวงจร เพื่อส่งข้อมูลไปที่วงจรส่วน RF module แบบ transceiver ที่ทำหน้าที่เป็นตัวส่งสัญญาณออกไปในรูปคลื่นวิทยุไปยัง RF module แบบ transceiver อีกตัวหนึ่งที่อยู่ทางด้านเครื่องเซิร์ฟเวอร์ที่จะทำหน้าที่เป็นตัวรับสัญญาณเข้ามา แล้วใช้วงจรไมโครคอนโทรลเลอร์ด้านเครื่องเซิร์ฟเวอร์เป็นตัวควบคุมเพื่อทำการส่งข้อมูลที่ได้รับเข้ามาไปยังวงจรเข้าพอร์ทอนุกรม เมื่อเซิร์ฟเวอร์ได้รับรหัสข้อมูลแล้วจะทำการเชื่อมต่อไปยังฐานข้อมูลที่มีรายละเอียดต่างๆของข้อมูลในบัตร แล้วทำการเรียกค้น (Query) ข้อมูลคือ รหัสบัตร ราคาบัตร และทำการหักลบราคาสินค้าทางคาด้านแบสจากการกดคีย์แพดมาแสดงที่เครื่องเซิร์ฟเวอร์ ในขณะเดียวกันก็จะทำการส่งค่าที่ได้จากฐานข้อมูลกลับไปยังฝั่งร้านค้าเพื่อแสดง ผลออกทางหน้าจอแอลซีดี (LCD) โดยข้อมูลทั้งหมดจะถูกส่งออกไปผ่านทางพอร์ทอนุกรมผ่านวงจรเพื่อ ให้ไมโครคอนโทรลเลอร์ส่งค่าเหล่านี้ไปยังวงจรส่วน RF module แบบ transceiver ที่ทำหน้าที่เป็นตัวส่งสัญญาณออกไปในรูปคลื่นวิทยุไปยัง RF module แบบ transceiver อีกตัวที่อยู่ทางด้านฝั่งร้านค้า ซึ่งจะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ในการรับสัญญาณแล้วส่งผ่านไปยังไมโครคอนโทรลเลอร์ที่ใช้การประมวลผลเพื่อแสดงรายละเอียดข้อมูลที่ได้รับมาจากฐานข้อมูลออกทางหน้าจอแอลซีดีแบบ 16 x 2

3.2 ไมโครคอนโทรลเลอร์

3.2.1 วงจรไมโครคอนโทรลเลอร์ฝั่งร้านค้า

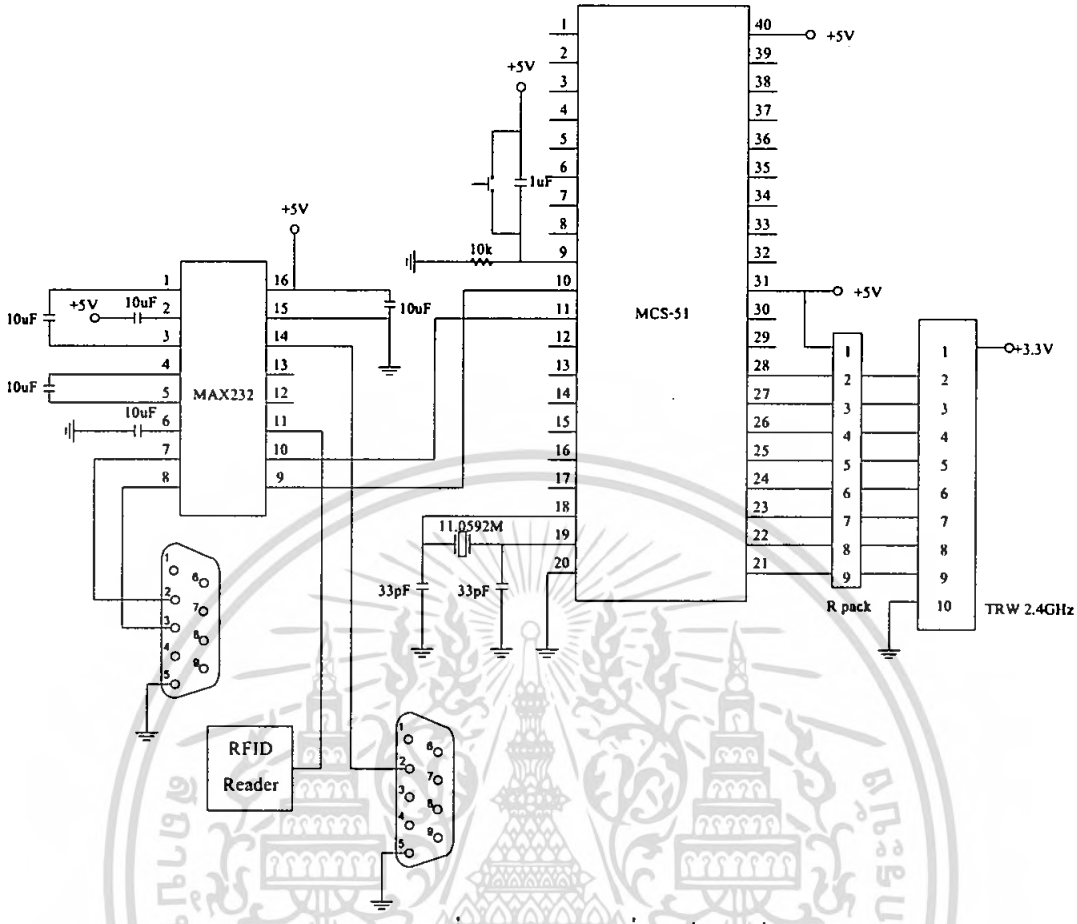


รูปที่ 3.2 แสดงวงจรฝั่งร้านค้า

ประกอบไปด้วยจอแอลซีดี คีย์แพด ไมโครคอนโทรเลอร์ TRW 2.4 GHz และหัวอ่าน RFID เมื่อลูกค้าต้องการทานอาหารร้านแห่งนี้ลูกค้าก็สั่งอาหารแล้วก็ให้บัตรแม่ค้าไปทาบที่หัวอ่าน RFID ข้อมูลในบัตรก็จะถูกส่งไปฝั่งเซิร์ฟเวอร์ ประกอบกับแม่ค้าต้องกดราคาอาหารผ่านทางคีย์แพด ซึ่งตัวคีย์แพดจะทำการตรวจสอบที่แถวและหลักของคีย์แพดที่เชื่อมกับไมโครคอนโทรเลอร์เพื่อนำค่าหมายเลขรหัสที่กดนั้นไปแสดงผลที่หน้าจอแอลซีดี และเก็บค่าไว้ที่หน่วยความจำ เพื่อทำการส่งต่อไปยังวงจรTRW-2.4GHz ในหน้าปัดของคีย์แพดนั้นนอกจากจะมีปุ่มกดหมายเลข 0-9 แล้ว ยังมีปุ่ม # ที่ทำหน้าที่ในการส่งราคาอาหารที่แม่ค้าต้องกดส่งไป ส่วนในหน้าจอแอลซีดีการแสดงผลจะเป็นราคาแม่ค้าที่กดไป พร้อมทั้งมียอดคงเหลือในบัตรนั้น ซึ่งการทำงานทั้งหมดนั้นจะถูกควบคุมโดยไมโครคอนโทรเลอร์

เอกสารนี้มีลิขสิทธิ์ ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

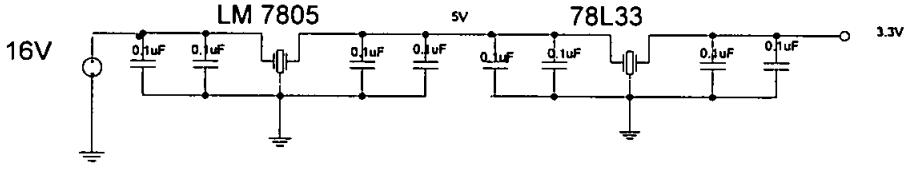
3.2.2 วงจรไมโครคอนโทรลเลอร์ฝั่งเซิร์ฟเวอร์



รูปที่ 3.3 แสดงวงจรฝั่งเซิร์ฟเวอร์

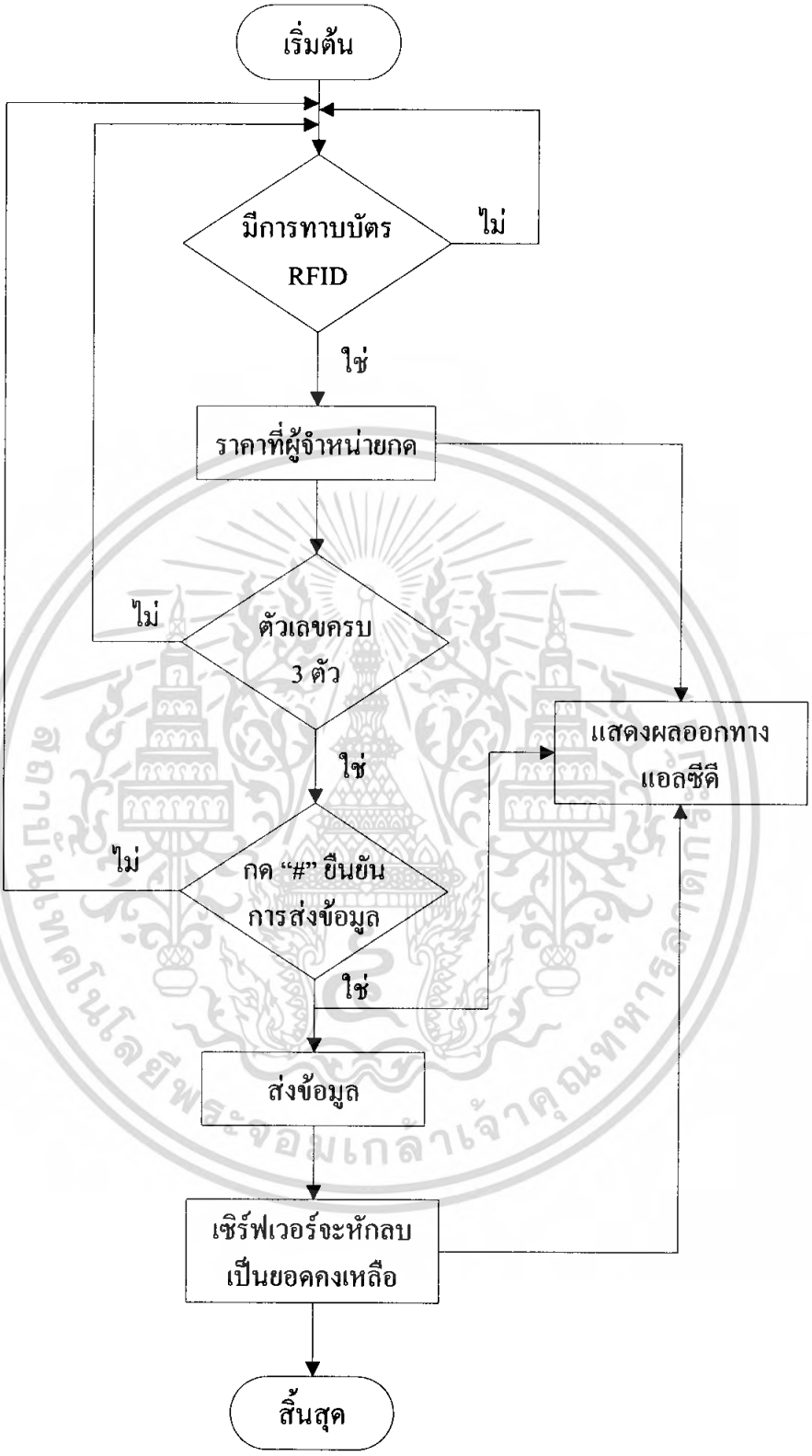
ในส่วนของวงจรเซิร์ฟเวอร์นี้จะทำการรับข้อมูลที่ส่งมาจาก TRW 2.4 GHz ด้านส่ง ซึ่งจะส่งออกทางขา Tx ของไมโครคอนโทรเลอร์ไปยังวงจรส่งผ่านพอร์ทอนุกรม ซึ่งไอซี MAX232 ทำหน้าที่แปลงระดับสัญญาณไฟฟ้า แล้วส่งข้อมูลผ่าน DB9 เข้าสู่คอมพิวเตอร์หรือเซิร์ฟเวอร์เพื่อทำการหักลบราคาให้เป็นยอดคงเหลือ จากนั้นเมื่อเซิร์ฟเวอร์ทำการประมวลผลกลับมาแล้วส่งต่อไปยังฝั่งร้านค้าขึ้นหน้าจอแอลซีดี

3.2.3 วงจรจ่ายไฟ



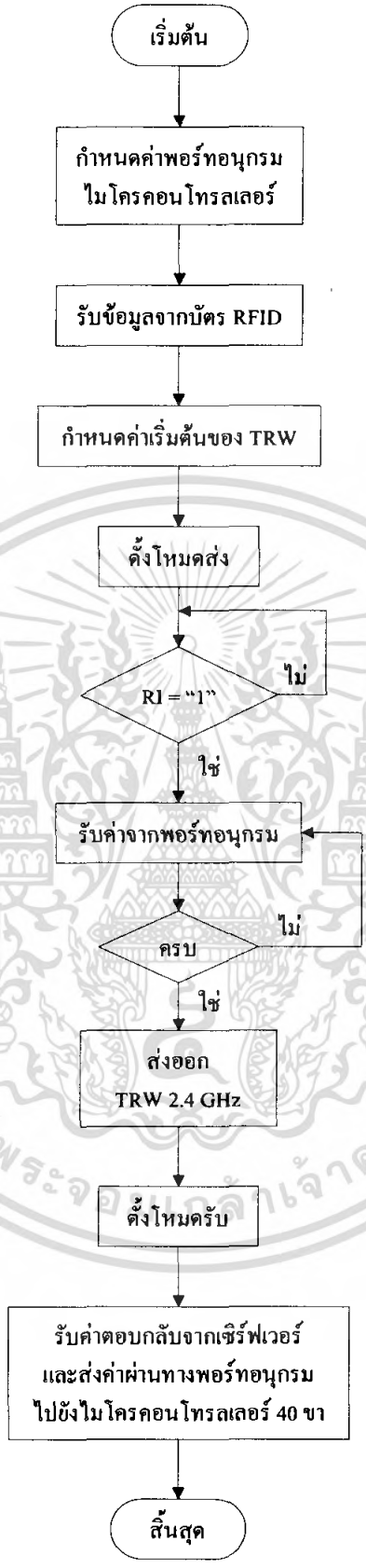
รูปที่ 3.4 แสดงวงจรภาคจ่ายไฟ (Power Supply)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



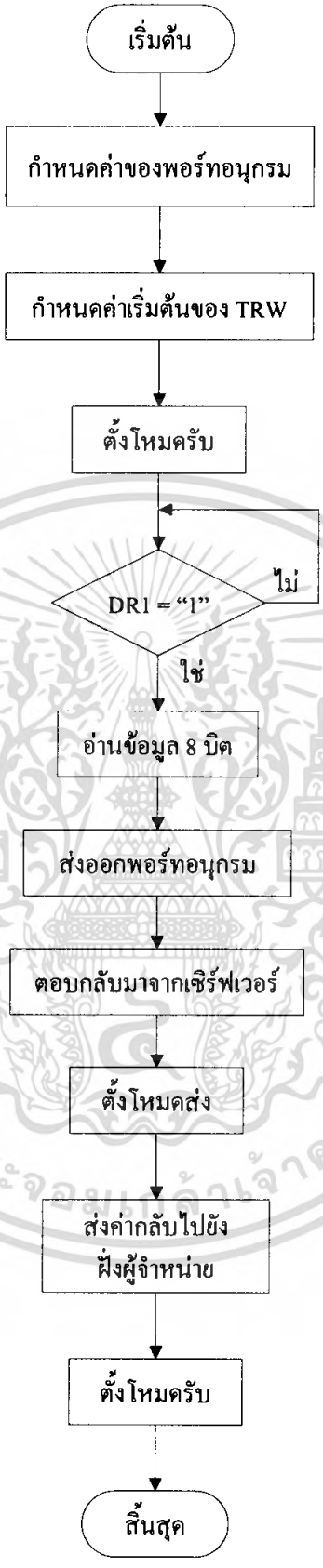
รูปที่ 3.5 แสดงโฟลวชาร์ทการทำงานในไมโครคอนโทรลเลอร์ของฝั่งร้านค้าและเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงโฟลวชาร์ทการทำงานของ TRW 2.4 GHz ทางฝั่งร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงโฟลวชาร์ทการทำงานของ TRW 2.4 GHz ทางฝั่งเครื่องเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนของเครื่องเซิร์ฟเวอร์

หน้าที่ของเครื่องเซิร์ฟเวอร์

1. รับข้อมูลจากบัตร RFID ที่มาจากการทาบบัตรและราคาอาหารที่กดคีย์แพดของแม่ค้าผ่านทางพอร์ทอนุกรม

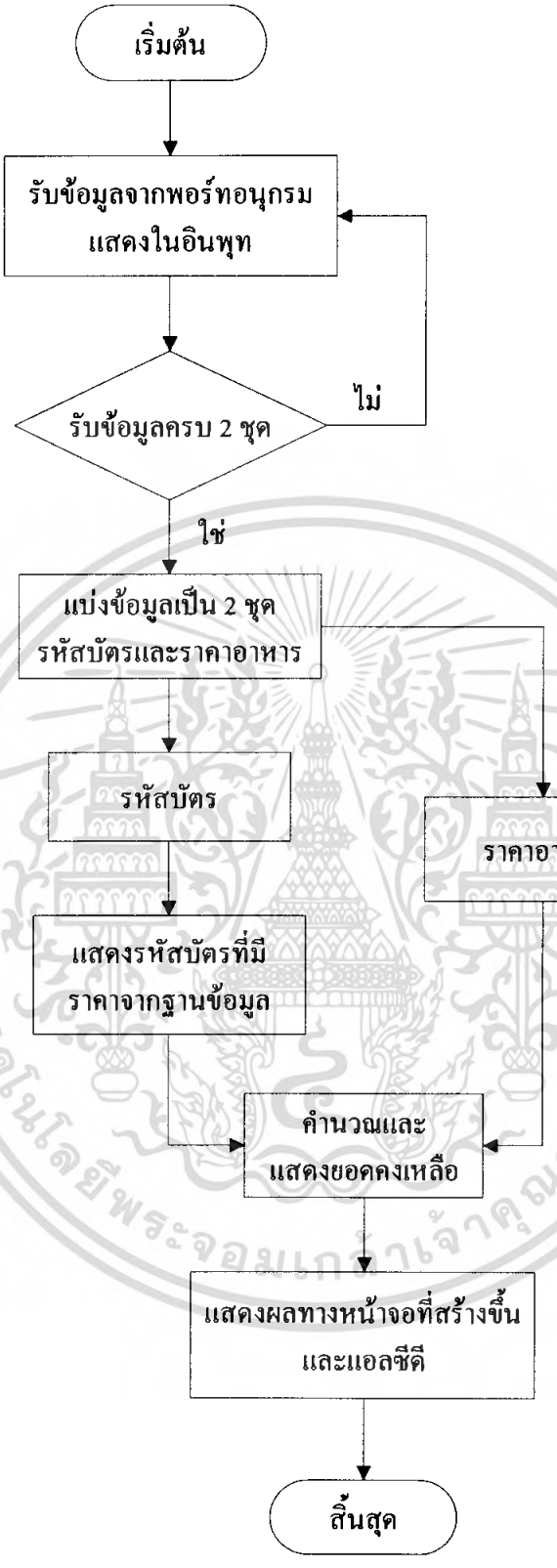
2. แสดงรายละเอียดของรหัสบัตรที่เข้า ราคาของอาหารของแต่ละชนิด และยอดคงเหลือ โดยแสดงผ่านหน้าจอแสดงผลที่สร้างขึ้นจากโปรแกรม Visual C#

3. ส่งค่ากลับไปยังฝั่งร้านค้า แสดงออกโดย หน้าจอแอลซีดี

โปรแกรมของเครื่องเซิร์ฟเวอร์มีลักษณะการทำงาน ดังนี้

เมื่อโปรแกรมของเครื่องเซิร์ฟเวอร์อยู่ในสถานะพร้อมทำงาน โปรแกรมจะทำการรอรับข้อมูลที่ถูกส่งเข้ามาทางพอร์ทอนุกรม โดยลักษณะของข้อมูลที่ต้องการของโปรแกรมจะเป็นชุดรหัสบัตรพร้อมทั้งราคาที่แม่ค้า (ตัวเลข 3 ตัว) เท่านั้น โปรแกรมจะทำการรอรับค่าข้อมูลตัวเลขจนกว่าจะครบจำนวนดังกล่าวจึงจะนำชุดตัวเลขเหล่านี้ไปประมวลผล แล้วแสดงยอดคงเหลือออกทางหน้าจอแสดงผลที่สร้างขึ้นจากโปรแกรม Visual C# พร้อมทั้งแสดงออกหน้าจอแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

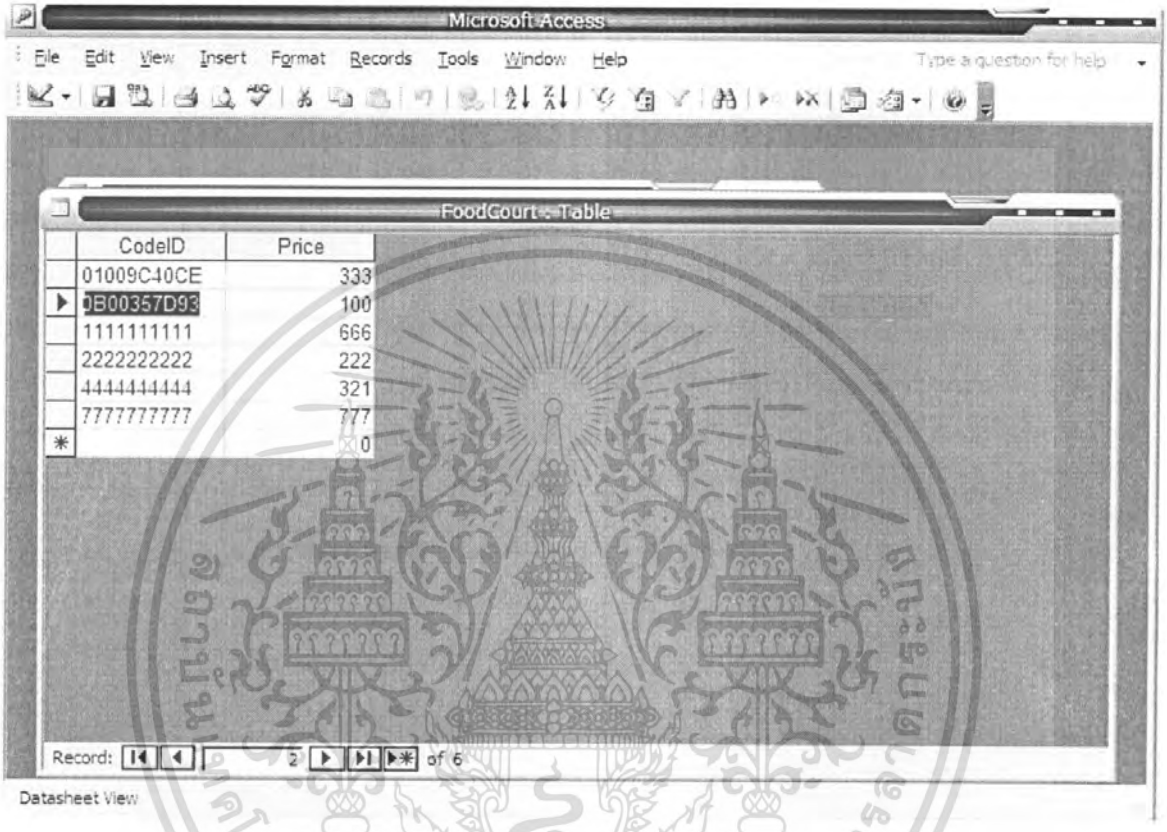


รูปที่ 3.8 แสดงโฟลวชาร์ทการทำงานในส่วนของเครื่องเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ฐานข้อมูล

ฐานข้อมูลจะรวบรวมข้อมูลที่สัมพันธ์กันโดยข้อมูลเหล่านี้จะถูกเก็บในรูปตารางซึ่งจะประกอบด้วยคอลัมน์ และแถวของตารางสำหรับปริญญานิพนธ์นี้จะทำการเก็บตัวอย่างชนิดราคาบัตร 6 ชนิด



รูปที่ 3.9 แสดงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

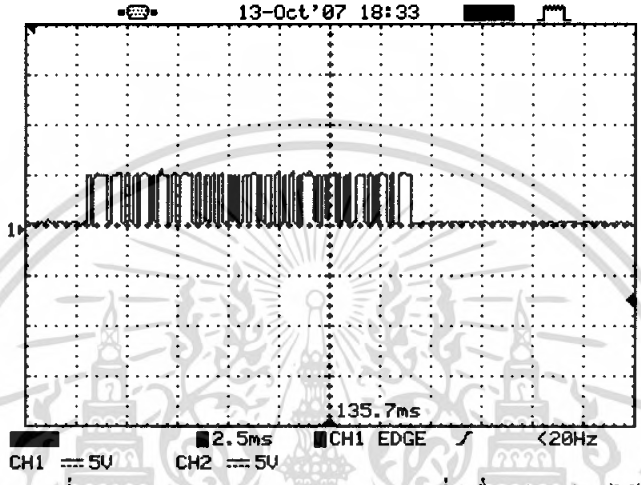
บทที่ 4

การทดลองและผลการทดลอง

4.1 ทดลองอ่านค่าบัตร RFID

ขั้นตอนการทดลอง

1. ต่อเครื่องอ่านเข้ากับ oscilloscope
2. อ่านค่าบัตร RFID แล้ววัดสัญญาณที่ได้



รูปที่ 4.1 สัญญาณของรหัสบัตร RFID ที่เครื่องอ่านอ่านได้

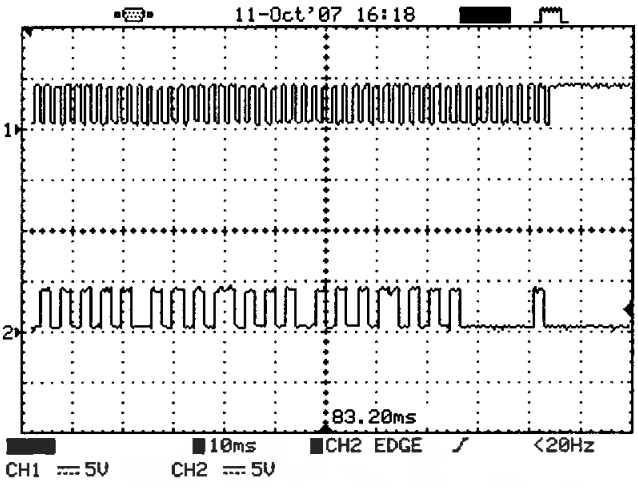
4.2 การทดลองอ่านค่ารหัสบัตร RFID แล้วส่งผ่านโมดูล TRW 2.4GHz

ขั้นตอนการทดลอง

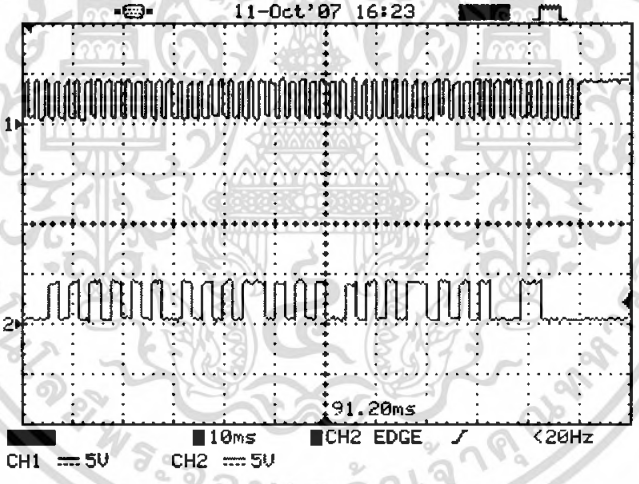
1. เขียนโปรแกรมลงไมโครคอนโทรลเลอร์ AT89C51 เพื่อรับค่ารหัสบัตร RFID ตามโฟลวชาร์ท รูปที่ 3.6 และโปรแกรมในภาคผนวก
2. ต่อวงจรตามรูปที่ 3.2
3. ทำการอ่านค่าบัตร RFID จากเครื่องอ่านบัตร
4. วัดสัญญาณ Clock (ขา 5) เทียบกับ Data (ขา 6) ทีละไบต์ ทั้ง 14 ไบต์

โดยในที่นี้จะแสดงให้ดูไบต์แรก (STX รหัส 02H) ไบต์ที่สอง (รหัส 0) และไบต์สุดท้าย (EXTรหัส 03H)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

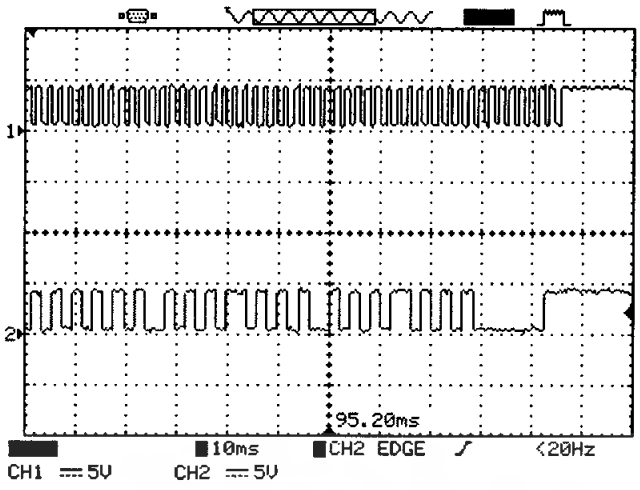


รูปที่ 4.2 สัญญาณ Clock และสัญญาณของข้อมูลในไบต์ที่ 1 (STX)
ช่องสัญญาณที่ 1: สัญญาณ Clock ที่ขา 5 ของ TRW 2.4GHz
ช่องสัญญาณที่ 2: สัญญาณข้อมูล ที่ขา 6 ของ TRW 2.4GHz



รูปที่ 4.3 สัญญาณ Clock และสัญญาณข้อมูลในไบต์ที่สอง
ช่องสัญญาณที่ 1: สัญญาณ Clock ที่ขา 5 ของ TRW 2.4GHz
ช่องสัญญาณที่ 2: สัญญาณข้อมูล ที่ขา 6 ของ TRW 2.4GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 สัญญาณ Clock และสัญญาณของข้อมูลในไบต์ที่ 14 (EXT)

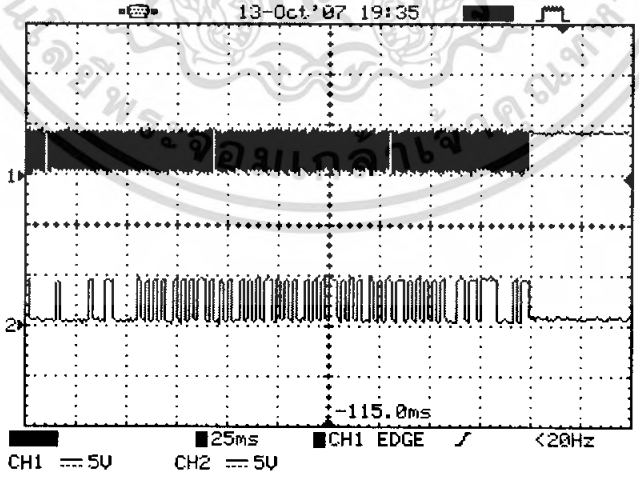
ช่องสัญญาณที่ 1: สัญญาณ Clock ที่ขา 5 ของ TRW 2.4GHz

ช่องสัญญาณที่ 2: สัญญาณข้อมูล ที่ขา 6 ของ TRW 2.4GHz

4.3 การทดลองรับค่าสัญญาณที่ส่งมาจากด้านส่ง

ขั้นตอนการทดลอง

1. เขียนโปรแกรมลงไมโครคอนโทรเลอร์ AT89C51 เพื่อรับค่าที่ส่งมา ตามโฟลวชาร์ทรูปที่ 3.7
2. ต่อวงจรดังรูปที่ 3.3
3. วัดสัญญาณ Clock (ขา 5) เทียบกับ Data (ขา 6)
4. เปิดโปรแกรม HyperTerminal เพื่อรับค่าที่ส่งมา



รูปที่ 4.5 สัญญาณ Clock และสัญญาณของข้อมูลทางด้านรับ

ช่องสัญญาณที่ 1: สัญญาณ Clock ที่ขา 5 ของ TRW 2.4GHz

ช่องสัญญาณที่ 2: สัญญาณข้อมูล ที่ขา 6 ของ TRW 2.4GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

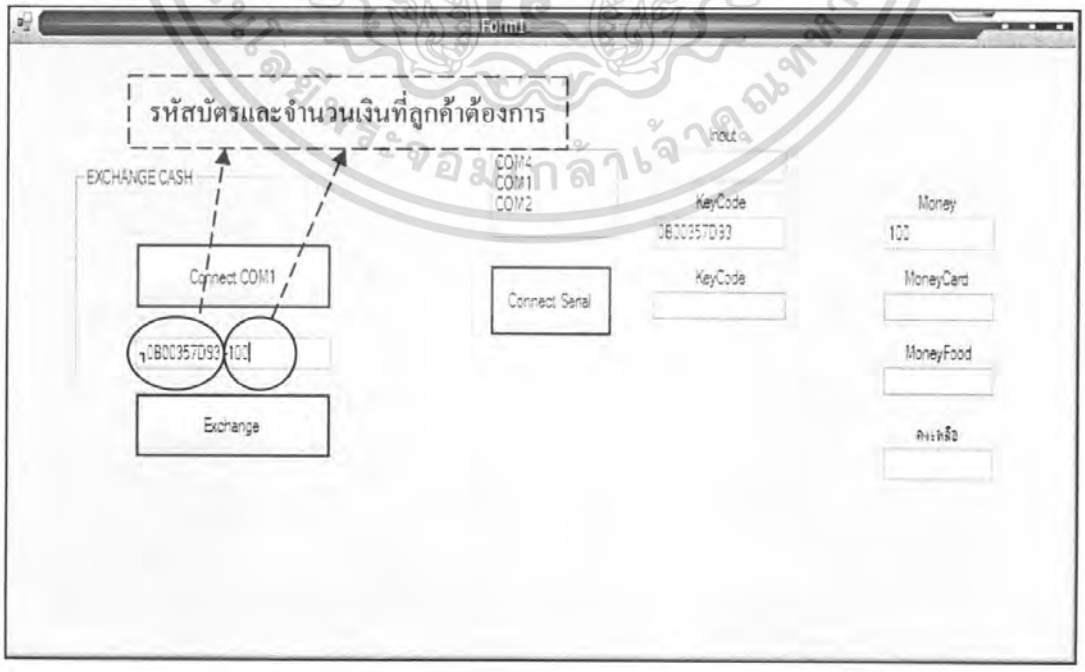
รูปที่ 4.6 รหัสบัตร RFID ที่ทางด้านรับอ่านค่าได้

จากรูปจะมีไบต์ของ STX (02H) แสดงเป็นตัวแรก ตามด้วยไบต์ข้อมูล 10 ไบต์และในไบต์ที่ 13 และ 14 จะเป็นการเลื่อนเคอร์เซอร์ไปทางซ้ายและถัดลงมา 1 บรรทัดเมื่อส่งไบต์ของข้อมูลมาครบ 10 ไบต์ ท้ายสุดจะเป็นไบต์ของ EXT (03H) ทำให้แต่ละบัตรจะแสดงเพียง 12 ไบต์

4.4 การทดลองอ่านค่ารหัสบัตร RFID พร้อมทำการระบุจำนวนเงินของบัตรไว้ในส่วนของ Database

ขั้นตอนการทดลอง

1. กดปุ่ม Connect COM1
2. นำบัตรมาทาบที่เครื่องอ่านบัตรทางฝั่งเซิร์ฟเวอร์
3. ระบุจำนวนเงินที่ลูกค้าต้องการ



รูปที่ 4.7 รหัสบัตรที่รับเข้ามาและจำนวนเงินที่ลูกค้าต้องการใช้

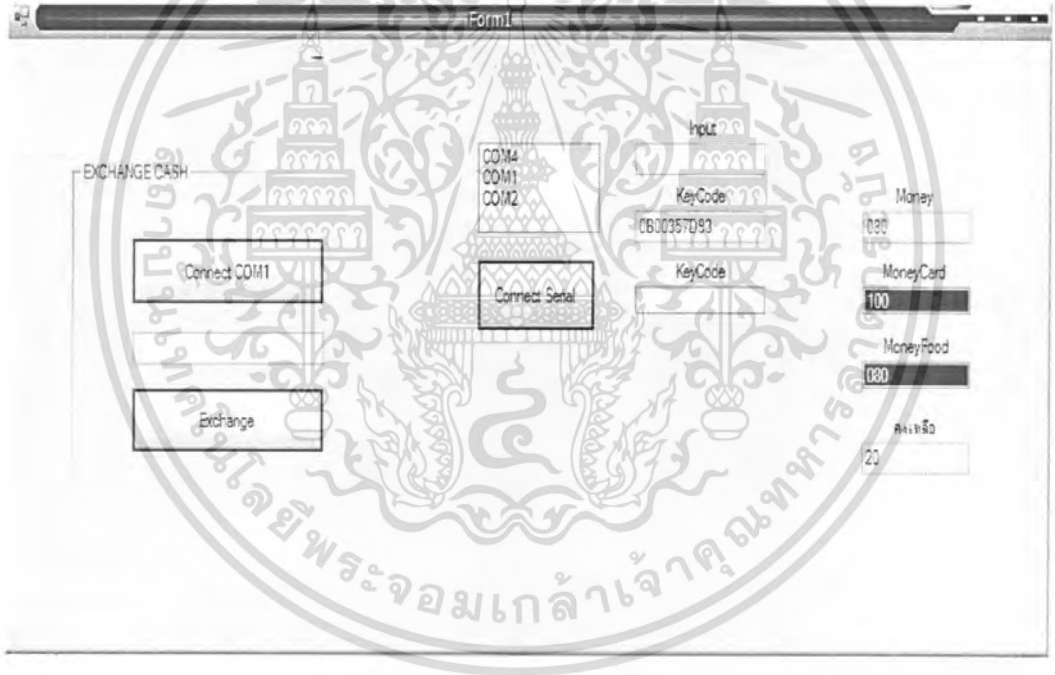
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.7 จะมีข้อมูล 2 ชุด ชุดแรกคือรหัสบัตร ชุดที่สองคือจำนวนเงินที่ลูกค้าต้องการ ซึ่งเมื่อรหัสบัตรเข้ามาแล้วเจ้าหน้าที่ทางฝั่งเซิร์ฟเวอร์ก็จะทำการกดค่าของจำนวนเงินที่ลูกค้าต้องการ ซึ่งข้อมูลส่วนนี้จะถูกเก็บไว้ในส่วนของ Database

4.5 การทดลองอ่านค่ารหัสบัตร RFID ส่งผ่านโมดูล TRW-2.4GHz แล้วส่งรหัสที่ได้เข้าสู่ในส่วนของ Database

ขั้นตอนการทดลอง

1. ต่อวงจรตามรูปที่ 3.2 และ 3.3
2. อ่านค่าบัตร RFID และกดคีย์แพดป้อนจำนวนเงินที่ต้องการ
3. Run โปรแกรม Visual C# หน้าฟอร์ม Form1 เพื่อรอรับข้อมูลที่จะแสดงผล



รูปที่ 4.8 รหัสบัตร ราคาบัตร จำนวนเงินที่กดเข้ามาและเงินคงเหลือของบัตร แสดงที่จอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 จำนวนเงินคงเหลือที่ส่งกลับมาแสดงที่จอแอลซีดีทางด้านส่ง

4.6 การทดลองทำการคืนเงินลูกค้าทางด้านฝั่งเซิร์ฟเวอร์

ขั้นตอนการทดลอง

1. กดปุ่ม Connect COM1
2. นำบัตรมาทาบที่เครื่องอ่าน
3. กดปุ่ม Exchange

รูปที่ 4.10 แสดงจำนวนเงินคงเหลือของบัตรคืนแก่ลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกดปุ่ม Exchange ก็จะแสดงหน้าต่างยอดคงเหลือขึ้นมาดังรูปที่ 4.10

บทที่ 5
บทวิจารณ์และสรุปผล

5.1 บทวิจารณ์

จากการศึกษาเกี่ยวกับการทำงานของเครื่องอ่าน RFID ไมโครคอนโทรลเลอร์ TRW 2.4GHz และระบบฐานข้อมูล พบปัญหาดังต่อไปนี้

5.1.1 ส่วนของเครื่องอ่าน RFID

ความเร็วข้อมูลที่โปรแกรมลงบนไมโครคอนโทรลเลอร์ต้องทำการตั้งค่าให้ตรงกับความเร็วข้อมูลของเครื่องอ่านบัตร RFID ที่ใช้

5.1.2 ส่วนของไมโครคอนโทรลเลอร์

จากการออกแบบระบบสื่อสารไร้สาย ได้ทำการออกแบบโดยใช้ไมโครคอนโทรลเลอร์ 2 ตัว คือ AT89C5I ทั้งฝั่งร้านค้า และฝั่งเซิร์ฟเวอร์ นอกจากนี้ในขั้นตอนการออกแบบ ได้มีการใช้การแสดงผลออกทางหน้าจอแอลซีดี แบบ 16 x 2

ในส่วนนี้จะต้องทำการคำนวณเวลาที่ใช้ในการทำงานของโปรแกรมเพื่อให้สามารถรับและส่งข้อมูลกันได้ทัน จึงต้องทำการตั้งค่าช่วงเวลาให้เหมาะสม เริ่มตั้งแต่การรับข้อมูลจาก บัตร RFID จนถึงการส่งข้อมูลผ่านโมดูล TRW 2.4GHz

5.1.3 โมดูล TRW 2.4GHz

ในส่วนของการรับส่งผ่านคลื่นวิทยุนั้น สามารถส่งและรับรหัสข้อมูลบัตรได้ แต่ควรระวังเรื่องการตั้งค่าความถี่ ควรตั้งให้มีค่าคนละความถี่ เพื่อป้องกันการซ้อนทับของข้อมูล

5.1.4 ส่วนของฐานข้อมูล

การเขียนโปรแกรม Visual C# และบันทึกค่าต่างๆลงในฐานข้อมูลสำหรับโครงงานนี้สามารถทำงานได้อย่างถูกต้อง โดยสามารถทำการเรียกค้นรหัสบัตรและราคาบัตรได้จากฐานข้อมูลที่มีอยู่ เมื่อมีรหัสบัตรเข้ามาทางพอร์ทอนุกรม

5.2 บทสรุป

จากการที่ได้ศึกษาและทดลองสร้างระบบศูนย์อาหารโดยใช้ RFID สามารถส่งรหัสในบัตรและราคาอาหารที่ผู้จำหน่ายอาหารกดจากทางด้านส่งไปยังด้านรับได้และสามารถเรียกค้นข้อมูลของรหัสในบัตรที่อยู่ในฐานข้อมูลเพื่อแสดงผลของยอดเงินคงเหลือบนหน้าจอทางฝั่งเซิร์ฟเวอร์ได้ พร้อมทั้งยังสามารถแสดงผลที่หน้าจอแอลซีดีทางฝั่งร้านค้าได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. เทคโนโลยี RFID นวัตรกรรมกรรมแห่งการเพิ่มผลผลิต. Omron Electronics

2. "RFID Present".[Online].Available:http//www.nectec.com

3. "RFID: Radio Frequency Identification".[Online].Available:http//www.rfidjournal.com

4. "RFID: Radio Frequency Option for Passive Chip Tags".[Online].Available:http//www.IDtechex.com

5. "GP-8 Reader".[Online].Available:http//www.promag.com , www.silaresearch.com

6. สมยศ จุณณะปิยะ, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูลMCS-51 ,2550

7. วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ฉบับ AT89C5x/AT89Sxxxx", บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, กรุงเทพฯ

8. ศุภชัย สมพานิช, "คู่มือการเขียนโปรแกรมและใช้งาน Visual C# .NET ฉบับสมบูรณ์", สำนักพิมพ์อินโฟเพรส, นนทบุรี, 2546

9. ศุภชัย สมพานิช, "Database Programming ด้วย VB 2005 & VC# 2005 ฉบับสมบูรณ์", สำนักพิมพ์อินโฟเพรส, นนทบุรี, 2549

10. Karli Watson, David Espinosa, Zach Greenvoss, Jacob Hammer Pedersen, Christian Nagel, Jon D. Reid, Matthew Reynolds, Morgan Skinner, Eric White, สุวัฒนา สุขสมจินตน์ เรียบเรียง, "คัมภีร์การใช้ Visual C# ฉบับสมบูรณ์", บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), กรุงเทพฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมด้านส่ง

```
;*********************************************************************
*
;                   SHOP
;*********************************************************************
*
            CH      EQU   030H

            CE      BIT   P1.0
            CLK2    BIT   P1.1
            CS      BIT   P1.2
            CLK     BIT   P1.3
            DAT     BIT   P1.4
            DR1     BIT   P1.5
            DOUT2   BIT   P1.6
            DR2     BIT   P1.7

            CH_1    EQU   030H
            CH_2    EQU   031H
            CH_3    EQU   032H
            STA     EQU   033H
            NUM     EQU   034H


            ORG 0000H

            LCALL INIT_LCD

            MOV   LCD_ADDR,#00H
            ACALL SET_ADDR_LCD
            MOV   DPTR,#TITTLE_1
            ACALL WR8CHAR
            ACALL DELAY_10ms

            CLR   P3.6
            CLR   P3.7

;*************************************************************

MAIN:       MOV   NUM,#00H
            MOV   A,#00H

            MOV   PCON,#00H
            MOV   SCON,#50H
            MOV   TMOD,#20H
            MOV   TH1,#0FDH

            SETB  TR1
      ;     LJMP  X0
            MOV   R0,#031H
            MOV   R4,#0EH

RES:        LCALL RX
            MOV   @R0,ACC
            INC   R0
            DJNZ  R4,RES

            MOV   CH,#031H
            MOV   R4,#0EH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PRE_9:          LCALL INIT
                LCALL SETMODE_TX
                MOV   R0,CH
                MOV   ACC,@R0
                LCALL SEND_TRW
                INC   CH
                DJNZ  R4,PRE_9

                SETB  P3.6

                MOV   A,#00H

X0:             MOV   P2,#11110111B
                JB    P2.7,X1
                JNB   P2.7,$
                CJNE  A,#00H,Y0
                MOV   CH_1,#31H
                INC   ACC
                LJMP  X0
Y0:             CJNE  A,#01H,Y1
                MOV   CH_2,#31H
                INC   ACC
                LJMP  X0
Y1:             CJNE  A,#02H,X1
                MOV   CH_3,#31H
                LJMP  X11

X1:             JB    P2.6,X2
                JNB   P2.6,$
                CJNE  A,#00H,Y2
                MOV   CH_1,#32H
                INC   ACC
                LJMP  X0
Y2:             CJNE  A,#01H,Y3
                MOV   CH_2,#32H
                INC   ACC
                LJMP  X0
Y3:             CJNE  A,#02H,X2
                MOV   CH_3,#32H
                LJMP  X11

X2:             JB    P2.5,X3
                JNB   P2.5,$
                CJNE  A,#00H,Y4
                MOV   CH_1,#33H
                INC   ACC
                LJMP  X0
Y4:             CJNE  A,#01H,Y5
                MOV   CH_2,#33H
                INC   ACC
                LJMP  X0
Y5:             CJNE  A,#02H,X3
                MOV   CH_3,#33H
                LJMP  X11

X3:             MOV   P2,#11111011B
                JB    P2.7,X4
                JNB   P2.7,$
                CJNE  A,#00H,Y6
                MOV   CH_1,#34H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            INC   ACC
            LJMP  X0
Y6:         CJNE  A,#01H,Y7
            MOV   CH_2,#34H
            INC   ACC
            LJMP  X0
Y7:         CJNE  A,#02H,X4
            MOV   CH_3,#34H
            LJMP  X11

X4:         JB    P2.6,X5
            JNB   P2.6,$
            CJNE  A,#00H,Y8
            MOV   CH_1,#35H
            INC   ACC
            LJMP  X0
Y8:         CJNE  A,#01H,Y9
            MOV   CH_2,#35H
            INC   ACC
            LJMP  X0
Y9:         CJNE  A,#02H,X5
            MOV   CH_3,#35H
            LJMP  X11

X5:         JB    P2.5,X6
            JNB   P2.5,$
            CJNE  A,#00H,Y10
            MOV   CH_1,#36H
            INC   ACC
            LJMP  X0
Y10:        CJNE  A,#01H,Y11
            MOV   CH_2,#36H
            INC   ACC
            LJMP  X0
Y11:        CJNE  A,#02H,X6
            MOV   CH_3,#36H
            LJMP  X11

X6:         MOV   P2,#11111101B
            JB    P2.7,X7
            JNB   P2.7,$
            CJNE  A,#00H,Y12
            MOV   CH_1,#37H
            INC   ACC
            LJMP  X0
Y12:        CJNE  A,#01H,Y13
            MOV   CH_2,#37H
            INC   ACC
            LJMP  X0
Y13:        CJNE  A,#02H,X7
            MOV   CH_3,#37H
            LJMP  X11

X7:         JB    P2.6,X8
            JNB   P2.6,$
            CJNE  A,#00H,Y14
            MOV   CH_1,#38H
            INC   ACC
            LJMP  X0
Y14:        CJNE  A,#01H,Y15
            MOV   CH_2,#38H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            INC   ACC
            LJMP  X0
Y15:        CJNE  A,#02H,X8
            MOV   CH_3,#38H
            LJMP  X11

X8:         JB    P2.5,X10
            JNB   P2.5,$
            CJNE  A,#00H,Y16
            MOV   CH_1,#39H
            INC   ACC
            LJMP  X0
Y16:        CJNE  A,#01H,Y17
            MOV   CH_2,#39H
            INC   ACC
            LJMP  X0
Y17:        CJNE  A,#02H,X10
            MOV   CH_3,#39H
            LJMP  X11

X10:        MOV   P2,#11111110B
            JB    P2.6,X12
            JNB   P2.6,$
            CJNE  A,#00H,Y20
            MOV   CH_1,#30H
            INC   ACC
            LJMP  X0
Y20:        CJNE  A,#01H,Y21
            MOV   CH_2,#30H
            INC   ACC
            LJMP  X0
Y21:        CJNE  A,#02H,X12
            MOV   CH_3,#30H
            LJMP  X11

X12:        LJMP  X0

X11:        SETB  P3.7
            MOV   P2,#11111110B
            JB    P2.7,X9
            JNB   P2.7,$
            MOV   STA,#0FFH
            LJMP  MAIN

X9:         MOV   P2,#11111110B
            JB    P2.5,X11
            JNB   P2.5,$
            MOV   STA,#00H
            LJMP  SUM


SUM:        MOV   A,#00H


            LCALL INIT_LCD

            MOV   P0,#0FFH

            LCALL PP

            MOV   A,CH_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL TX
                LCALL DELAY_100ms
                MOV   A,CH_2
                LCALL TX
                LCALL DELAY_100ms
                MOV   A,CH_3
                LCALL TX
                LCALL DELAY_100ms

                LCALL INIT
                LCALL SETMODE_TX
                MOV   ACC,CH_1
                LCALL SEND_TRW
                LCALL INIT
                LCALL SETMODE_TX
                MOV   ACC,CH_2
                LCALL SEND_TRW
                LCALL INIT
                LCALL SETMODE_TX
                MOV   ACC,CH_3
                LCALL SEND_TRW

                LCALL RECIVE
                MOV   CH_1,ACC
                LCALL RECIVE
                MOV   CH_2,ACC
                LCALL RECIVE
                MOV   CH_3,ACC


                LCALL PP
                SJMP  $

;*****************************************

TX:             MOV   SBUF,ACC
                JNB   TI,$
                CLR   TI
                RET

RX:             JNB   RI,$
                MOV   ACC,SBUF
                CLR   RI
                RET


DELAY3:         MOV   R5,#0E0H
LOOP3:                MOV   R6,#0FFH
LOOP2:                MOV   R7,#07H
LOOP1:                DJNZ  R7,LOOP1
                      DJNZ  R6,LOOP2
                      DJNZ  R5,LOOP3
                      RET
;*********************************************************************

PP:             MOV   LCD_ADDR,#00H
                ACALL SET_ADDR_LCD
                MOV   LCD_DATA,CH_1
                ACALL WRCHAR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ACALL DELAY_10ms
            MOV   LCD_ADDR,#01H
            ACALL SET_ADDR_LCD
            MOV   LCD_DATA,CH_2
            ACALL WRCHAR_LCD
            ACALL DELAY_10ms
            MOV   LCD_ADDR,#02H
            ACALL SET_ADDR_LCD
            MOV   LCD_DATA,CH_3
            ACALL WRCHAR_LCD
            ACALL DELAY_10ms
            RET

;*********************************************************************
;           Initial     for use TRW Channel1
;*********************************************************************

INIT:       CALL  DELAY_100ms
            CLR   CE
            CLR   CS
            CLR   DAT
            CLR   CLK
            CALL  DELAY_100ms
            RET

;*********************************************************************
;           Function Config for TRW TX
;*********************************************************************

SETMODE_TX:
            CLR   CE
            SETB  CS
            CLR   A
            MOV   R1,#18                    ;18*8 = 144bit
SETMODE_0_TX:     MOV   DPTR,#CONFIG_TEST_TX    ;Set Header
            PUSH  ACC
            MOVC  A,@A+DPTR
            CALL  WRITE_TRW24
            POP   ACC
            INC   A
            DJNZ  R1,SETMODE_0_TX
            SETB  DAT
            SETB  DR1
            SETB  CE
            CLR   CS
            RET

;*********************************************************************
;           Function Config for TRW RX
;*********************************************************************

SETMODE_RX:
            CLR   CE
            SETB  CS
            CLR   A
            MOV   R1,#18                    ;18*8 = 144bit
SETMODE_0_RX:     MOV   DPTR,#CONFIG_TEST_RX    ;Set Header
            PUSH  ACC
            MOVC  A,@A+DPTR
            CALL  WRITE_TRW24
            POP   ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            INC   A
            DJNZ  R1,SETMODE_0_RX
            SETB  DAT
            SETB  DR1
            SETB  CE
            CLR   CS
            RET

;*********************************************************************
;           Function Send Address+DATA to Receiver
;*********************************************************************

SEND_TRW:
            CLR   CS
            SETB  CE
            PUSH  ACC
            CLR   A
            MOV   R1,#5
SEND_TRW_0: MOV   DPTR,#CONFIG_ADDR1_TX
            PUSH  ACC
            MOVC  A,@A+DPTR
            CALL  WRITE_TRW24
            POP   ACC
            INC   A
            DJNZ  R1,SEND_TRW_0
            POP   ACC
            CALL  WRITE_TRW24
            CALL  DELAY_100ms
            CLR   CLK
            CLR   CE
            CLR   DAT
            RET

;*********************************************************************
;           MAIN RX : Get DATA      From TRW and Action
;*********************************************************************

RECIVE:     CALL  INIT
            CALL  SETMODE_RX
            JNB   DR1,$
            CALL  READ_TRW24
;           MOV   P0,A
            RET

;*********************************************************************
;           READ 8 bit DATA from TRW
;*********************************************************************

READ_TRW24:       CLR   A
            MOV   R0,#8
READ_TRW24_0:     RL    A
            SETB  CLK
            JB    DAT,READ_1
            CLR   ACC.0
            JMP   READ_TRW24_1
READ_1:     SETB  ACC.0
READ_TRW24_1:     CLR   CLK
            DJNZ  R0,READ_TRW24_0
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************************************
;           Write 8 bit to DATA pin
;*********************************************************************

WRITE_TRW24:       MOV    R0,#8
WRITE_TRW24_0:  JB        ACC.7,WRITE1
            CLR    DAT
            JMP    WRITE_TRW2
WRITE1:            SETB   DAT
WRITE_TRW2:        CALL   CLK_TRW
            RL     A
            DJNZ   R0,WRITE_TRW24_0
            RET

;*********************************************************************
;           Function Generate Clock to pin CLK
;*********************************************************************

CLK_TRW:    CLR    CLK
            CALL   DELAY_1ms
            SETB   CLK
            CALL   DELAY_1ms
            RET

;*********************************************************************
;           Delay time 1ms
;*********************************************************************

DELAY_1ms:  MOV    R6,#0E6H     ; Each loop = 1 ms
DELAY_1ms_1:       NOP
            NOP
            DJNZ   R6,DELAY_1ms_1
            RET

;*********************************************************************
;           Delay time 100ms
;*********************************************************************

DELAY_100ms:       MOV    R7,#100      ; Do 100 times
DELAY_100ms_1:     MOV    R6,#0E6H     ; Each loop = 1 ms
DELAY_100ms_2:     NOP
            NOP
            DJNZ   R6,DELAY_100ms_2
            DJNZ   R7,DELAY_100ms_1
            RET

;*********************************************************************
;           Delay time 1s
;*********************************************************************

DELAY_1s:   MOV    R5,#0AH
DELAY_1s_1: CALL   DELAY_100ms
            DJNZ   R5,DELAY_1s_1
            RET

;*********************************************************************
;           Define Config Header TX for TRW
;*********************************************************************

CONFIG_TEST_TX:          DB  8EH,08H,1CH
CONFIG_LEN2_TX:  DB 08H
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CONFIG_LEN1_TX:    DB 08H
CONFIG_ADDR2_TX:   DB  0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_TX:   DB  0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_TX:       DB 0A3H
CONFIG_RF_TX:            DB 6FH
CONFIG_CH_TX:            DB 0AH ;TX
;CONFIG_CH:              DB 0BH ;RX

;*********************************************************************
;            Define Config Header RX for TRW
;*********************************************************************

CONFIG_TEST_RX:    DB 8EH,08H,1CH
CONFIG_LEN2_RX:    DB 08H
CONFIG_LEN1_RX:    DB 08H
CONFIG_ADDR2_RX:   DB  0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_RX:   DB  0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_RX:       DB 0A3H
CONFIG_RF_RX:            DB 6FH
;CONFIG_CH:              DB 0AH ;TX
CONFIG_CH_RX:            DB 0BH ;RX




;********************
;Set LCD Address
;********************
SET_ADDR_LCD:      CLR   LCD_RS
            MOV    A,LCD_ADDR
            SETB   ACC.7
            MOV    P0,A
            ACALL LCD_CLK
            RET

;********************
;LCD CLK
;********************
LCD_CLK:     CLR   LCD_EN
             ACALL DL_LCD
             SETB  LCD_EN
             ACALL DL_LCD
             CLR   LCD_EN
             ACALL DL_LCD
             RET

;********************
;LCD Display On
;********************
LCD_ON:             CLR   LCD_RS
             MOV    P0,#00001100B
             ACALL LCD_CLK
             RET

;********************
;LCD Initialize
;********************
INIT_LCD:    ACALL DELAY_100ms
             CLR   LCD_RS
             MOV   P0,#3CH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ACALL LCD_CLK
            ACALL DELAY_10ms
            ACALL LCD_OFF
            ACALL LCD_CLR
            CLR   LCD_RS
            MOV   P0,#00000110B
            ACALL LCD_CLK
            ACALL LCD_HOME
            RET

;********************
;LCD Clear Display
;********************
LCD_CLR:    CLR   LCD_RS
            MOV   P0,#00000001B
            ACALL LCD_CLK
            RET
;********************
;LCD Return Home
;********************
LCD_HOME:   CLR   LCD_RS
            MOV   P0,#00000010B
            ACALL LCD_CLK
            RET
;********************
;LCD Display Off
;********************
LCD_OFF:    CLR   LCD_RS
            MOV   P0,#00001000B
            ACALL LCD_CLK
            RET

;********************
;LCD Left Shift
;********************
LSHT:               CLR   LCD_RS
                    MOV   P0,#10H
                    LCALL       LCD_CLK
                    RET


TITTLE_1:   DB      'PAO NONG'
LCD_EN              BIT   P3.5
LCD_RS              BIT   P3.4
LCD_ADDR    EQU     040H
LCD_DATA    EQU     041H


DL_LCD:     MOV     R7,#0F0H
DEL00:              MOV   R6,#03H
DEL11:              DJNZ  R6,DEL11
                    DJNZ  R7,DEL00
                    RET

DELAY_10ms:         MOV   R7,#0FFH
DEL2:               MOV   R6,#013H
DEL3:               DJNZ  R6,DEL3
                    DJNZ  R7,DEL2
                    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WR8CHAR:        MOV         R0,#0
WR8CHAR_1:      SETB        LCD_RS
                CLR         A
                MOVC        A,@A+DPTR
                MOV         P0,A
                LCALL       LCD_CLK
                INC         DPTR
                INC         R0
                CJNE        R0,#8,WR8CHAR_1
                LCALL       LCD_ON
                RET


WRCHAR_LCD: SETB  LCD_RS
            MOV   P0,LCD_DATA
            ACALL LCD_CLK
            ACALL LCD_ON
            MOV   LCD_DATA,#00H
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านภารค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมด้านรับ

```
;*****************************************************************
;                       SERVER
;*****************************************************************

            CE     BIT   P2.0  ;Set TRW Pin
            CLK2   BIT   P2.1
            CS     BIT   P2.2
            CLK    BIT   P2.3
            DAT    BIT   P2.4
            DR1    BIT   P2.5
            DOUT2  BIT   P2.6
            DR2    BIT   P2.7

            CH_1   EQU   30H
            CH_2   EQU   31H
            CH_3   EQU   32H

            ORG    0000H
            MOV    P0,#00000000B     ;CLR P0

            MOV    PCON,#00H
            MOV    SCON,#50H
            MOV    TMOD,#20H
            MOV    TH1,#0FDH

            SETB   TR1

;*****************************************************************

MAIN_TX:    MOV    R4,#0EH

PRE_1:      LCALL RECIVE
            MOV    P0,A
            LCALL TX
            DJNZ   R4,PRE_1

            LCALL RECIVE
            LCALL TX
            LCALL RECIVE
            LCALL TX
            LCALL RECIVE
            LCALL TX

            LCALL RX
            MOV    CH_1,ACC
            LCALL RX
            MOV    CH_2,ACC
            LCALL RX
            MOV    CH_3,ACC

            LCALL INIT
            LCALL SETMODE_TX
            MOV    ACC,CH_1
            LCALL SEND_TRW
            LCALL INIT
            LCALL SETMODE_TX
            MOV    ACC,CH_2
            LCALL SEND_TRW
            LCALL INIT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL SETMODE_TX
            MOV   ACC,CH_3
            LCALL SEND_TRW

            SJMP  $

;*****************************************************************

KK:         MOV   B,#010H
            DIV   AB
            MOV   R1,A
            MOV   R2,B

            MOV   A,R1
            LCALL COMPAR
            LCALL TX

            MOV   A,R2
            LCALL COMPAR
            LCALL TX

            MOV   A,#20H
            LCALL TX
            RET

;*****************************************************************

COMPAR:     CJNE  A,#00H,CP_1
            MOV   A,#30H
            RET
CP_1:       CJNE  A,#01H,CP_2
            MOV   A,#31H
            RET
CP_2:       CJNE  A,#02H,CP_3
            MOV   A,#32H
            RET
CP_3:       CJNE  A,#03H,CP_4
            MOV   A,#33H
            RET
CP_4:       CJNE  A,#04H,CP_5
            MOV   A,#34H
            RET
CP_5:       CJNE  A,#05H,CP_6
            MOV   A,#35H
            RET
CP_6:       CJNE  A,#06H,CP_7
            MOV   A,#36H
            RET
CP_7:       CJNE  A,#07H,CP_8
            MOV   A,#37H
            RET
CP_8:       CJNE  A,#08H,CP_9
            MOV   A,#38H
            RET
CP_9:       CJNE  A,#09H,CP_10
            MOV   A,#39H
            RET
CP_10:            CJNE  A,#0AH,CP_11
            MOV   A,#41H
            RET
CP_11:      CJNE  A,#0BH,CP_12
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   A,#42H
            RET
CP_12:      CJNE  A,#0CH,CP_13
            MOV   A,#43H
            RET
CP_13:      CJNE  A,#0DH,CP_14
            MOV   A,#44H
            RET
CP_14:      CJNE  A,#0EH,CP_15
            MOV   A,#45H
            RET
CP_15:      CJNE  A,#0FH,CP_16
            MOV   A,#46H
            RET

CP_16:      RET

;*****************************************************************

TX:         MOV   SBUF,ACC
            JNB   TI,$
            CLR   TI
            RET

RX:         JNB   RI,$
            MOV   ACC,SBUF
            CLR   RI
            RET

;*****************************************************************
;           Initial     for use TRW Channell
;*****************************************************************

INIT:       CALL  DELAY_100ms
            CLR   CE
            CLR   CS
            CLR   DAT
            CLR   CLK
            CALL  DELAY_100ms
            RET

;*****************************************************************
;           Function Config for TRW TX
;*****************************************************************

SETMODE_TX:       CLR   CE
            SETB  CS
            CLR   A
            MOV   R1,#18                    ;18*8 = 144bit
SETMODE_0_TX:     MOV   DPTR,#CONFIG_TEST_TX    ;Set Header
            PUSH  ACC
            MOVC  A,@A+DPTR
            CALL  WRITE_TRW24
            POP   ACC
            INC   A
            DJNZ  R1,SETMODE_0_TX
            SETB  DAT
            SETB  DR1
            SETB  CE
            CLR   CS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET

;*****************************************************************
;           Function Config for TRW RX
;*****************************************************************

SETMODE_RX:
            CLR   CE
            SETB  CS
            CLR   A
            MOV   R1,#18                    ;18*8 = 144bit
SETMODE_0_RX:     MOV   DPTR,#CONFIG_TEST_RX    ;Set Header
            PUSH  ACC
            MOVC  A,@A+DPTR
            CALL  WRITE_TRW24
            POP   ACC
            INC   A
            DJNZ  R1,SETMODE_0_RX
            SETB  DAT
            SETB  DR1
            SETB  CE
            CLR   CS
            RET

;*****************************************************************
;           Function Send Address+DATA to Receiver
;*****************************************************************

SEND_TRW:
            CLR   CS
            SETB  CE
            PUSH  ACC
            CLR   A
            MOV   R1,#5
SEND_TRW_0: MOV   DPTR,#CONFIG_ADDR1_TX
            PUSH  ACC
            MOVC  A,@A+DPTR
            CALL  WRITE_TRW24
            POP   ACC
            INC   A
            DJNZ  R1,SEND_TRW_0
            POP   ACC
            CALL  WRITE_TRW24
            CALL  DELAY_100ms
            CLR   CLK
            CLR   CE
            CLR   DAT
            RET

;*****************************************************************
;           MAIN RX : Get DATA      From TRW and Action
;*****************************************************************

RECIVE:           CALL  INIT
            CALL  SETMODE_RX
            JNB   DR1,$
            CALL  READ_TRW24
            MOV   P0,A
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************************
;           READ 8 bit DATA from TRW
;*****************************************************************

READ_TRW24:         CLR   A
            MOV     R0,#8
READ_TRW24_0:       RL    A
            SETB    CLK
            JB      DAT,READ_1
            CLR     ACC.0
            JMP     READ_TRW24_1
READ_1:             SETB  ACC.0
READ_TRW24_1:       CLR   CLK
            DJNZ    R0,READ_TRW24_0
            RET

;*****************************************************************
;           Write 8 bit to DATA pin
;*****************************************************************

WRITE_TRW24:        MOV   R0,#8
WRITE_TRW24_0:      JB    ACC.7,WRITE1
            CLR     DAT
            JMP     WRITE_TRW2
WRITE1:             SETB  DAT
WRITE_TRW2:         CALL  CLK_TRW
            RL      A
            DJNZ    R0,WRITE_TRW24_0
            RET

;*****************************************************************
;           Function Generate Clock to pin CLK
;*****************************************************************

CLK_TRW:    CLR     CLK
            CALL    DELAY_1ms
            SETB    CLK
            CALL    DELAY_1ms
            RET

;*****************************************************************
;           Delay time 1ms
;*****************************************************************

DELAY_1ms:  MOV     R6,#0E6H    ; Each loop = 1 ms
DELAY_1ms_1:        NOP
            NOP
            DJNZ    R6,DELAY_1ms_1
            RET

;*****************************************************************
;           Delay time 100ms
;*****************************************************************

DELAY_100ms:        MOV   R7,#100       ; Do 100 times
DELAY_100ms_1:      MOV   R6,#0E6H      ; Each loop = 1 ms
DELAY_100ms_2:      NOP
            NOP
            DJNZ    R6,DELAY_100ms_2
            DJNZ    R7,DELAY_100ms_1
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************************
;            Delay time 1s
;*****************************************************************

DELAY_1s:    MOV    R5,#0AH
DELAY_1s_1:  CALL   DELAY_100ms
             DJNZ   R5,DELAY_1s_1
             RET

;*****************************************************************
;            Define Config Header TX for TRW
;*****************************************************************

CONFIG_TEST_TX:    DB 8EH,08H,1CH
CONFIG_LEN2_TX:    DB 08H
CONFIG_LEN1_TX:    DB 08H
CONFIG_ADDR2_TX:   DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_TX:   DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_TX:      DB 0A3H
CONFIG_RF_TX:           DB 6FH
CONFIG_CH_TX:           DB 0AH ;TX
;CONFIG_CH:             DB 0BH ;RX

;*****************************************************************
;            Define Config Header RX for TRW
;*****************************************************************
CONFIG_TEST_RX:    DB 8EH,08H,1CH
CONFIG_LEN2_RX:    DB 08H
CONFIG_LEN1_RX:    DB 08H
CONFIG_ADDR2_RX:   DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_RX:   DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_RX:      DB 0A3H
CONFIG_RF_RX:           DB 6FH
;CONFIG_CH:             DB 0AH ;TX
CONFIG_CH_RX:           DB 0BH ;RX

             END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วน Database

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Data.OleDb;

namespace WindowsApplication3
{
    public partial class Form1 : Form
    {
        string RxString;
        SerialPort serial;

        public Form1()
        {
            InitializeComponent();

            serial = new SerialPort();
            serial.DataReceived += new
SerialDataReceivedEventHandler(serial_DataReceived);
        }

        string strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\\RFID.mdb";
        //string strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\\Northwind.mdb";
        OleDbConnection Conn = new OleDbConnection();
        DataSet ds = new DataSet();
        string Subtract;
        string x, y, z;

        private void Form1_Load(object sender, EventArgs e)
        {

listBox5.Items.AddRange(System.IO.Ports.SerialPort.GetPortNames());

            serial.PortName = "COM1";
            serial.BaudRate = 9600;
            serial.Parity = Parity.None;
            serial.DataBits = 8;
            serial.StopBits = StopBits.One;
            serial.Open();
        }

        private void serial_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            this.Invoke(new EventHandler(DisplayText));
        }

        private void DisplayText(object sender, EventArgs e)
        {
            try
            {
                RxString = serial.ReadExisting();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                textBox2.AppendText(RxString);
            }
            catch
            {
                MessageBox.Show("ไม่สามารถทำงานได้!!!", "ข้อผิดพลาด");
            }
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            if (textBox2.TextLength == 17)
            {
                listBox1.DataSource = null;
                textBox1.Clear();
                textBox3.Clear();
                textBox4.Clear();
                listBox2.Items.Clear();
                listBox4.Items.Clear();
                string Code = textBox2.Text;
                string Key, Pri;
                Key = Code.Substring(1, 10);
                textBox1.Text = Key;
                Pri = Code.Substring(14);
                textBox3.Text = Pri;
                string sqlProducts = "";
                string KeyID = Key;

                sqlProducts = "SELECT * FROM FoodCourt WHERE (CodeID
= '" + KeyID + "')";
                try
                {
                    if (Conn.State == ConnectionState.Open)
                    {
                        Conn.Close();
                    }
                    Conn.ConnectionString = strConn;
                    Conn.Open();

                    DataRow dr;
                    DataColumn[] pk = new DataColumn[1];
                    OleDbDataAdapter da = new
OleDbDataAdapter(sqlProducts, Conn);
                    da.Fill(ds, "FoodCourt");

                    pk[0] = ds.Tables["FoodCourt"].Columns["CodeID"];
                    ds.Tables["FoodCourt"].PrimaryKey = pk;
                    listBox1.DataSource = ds.Tables["FoodCourt"];
                    listBox1.DisplayMember = "CodeID";
                    dr =
ds.Tables["FoodCourt"].Rows.Find(listBox1.Text);

                    listBox4.Items.Add(dr["Price"].ToString());
                    listBox2.Items.Add(Pri);
                    string a, b;
                    listBox4.SetSelected(0, true);
                    a = listBox4.SelectedItem.ToString();
                    listBox2.SetSelected(0, true);
                    b = listBox2.SelectedItem.ToString();
                    Subtract = Convert.ToString(Convert.ToInt32(a) -
Convert.ToInt32(b));
                    textBox4.Text = Subtract;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    if (textBox4.TextLength == 3)
                    {
                        x = Subtract.Substring(0, 1);
                        y = Subtract.Substring(1, 1);
                        z = Subtract.Substring(2, 1);
                        serial.Write(x);
                        //serial.WriteTimeout = 200;
                        serial.Write(y);
                        //serial.WriteTimeout = 200;
                        serial.Write(z);
                    }
                    else if (textBox4.TextLength == 2)
                    {
                        y = Subtract.Substring(0, 1);
                        z = Subtract.Substring(1, 1);
                        serial.Write("0");
                        serial.Write(y);
                        //serial.WriteTimeout = 200;
                        serial.Write(z);
                    }
                    else
                    {
                        z = Subtract;
                        serial.Write("0");
                        serial.Write("0");
                        serial.Write(z);
                    }
                    textBox2.Clear();
                }
                catch (Exception errToAccess)
                {
                    MessageBox.Show("ไม่สามารถอ่านข้อมูลจากตารางได้!!!", "ข้อผิดพลาด");
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            serial.PortName = listBox5.SelectedItem.ToString();
            serial.BaudRate = 9600;
            serial.Parity = Parity.None;
            serial.DataBits = 8;
            serial.StopBits = StopBits.One;
            serial.Open();
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้