สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องอ่านบาร์โค้ดแบบไร้สาย

WIRELESS BARCODE READER

โดย

นางสาวกชกร ขุนชนะ

นางสาวแก้วนภา โพธิ

เลขทะเบียน 82911
เลขเรียกหนังสือ
วัน,เดือน,ปี 29 ก.ค. 2551

b. 11957049
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องอ่านบาร์โค้ดแบบไร้สาย

WIRELESS BARCODE READER

โดย

นางสาวกชกร ขุนชนะ 47010001

นางสาวแก้วนภา โพธิ 47010063

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 **เครื่องอ่านบาร์โค้ดแบบไร้สาย**

Wireless barcode reader

ผู้จัดทำ

1. นางสาวกชกร ขุนชนะ 47010001

2.นางสาวแก้วนภา โพธิ 47010063

อาจารย์ที่ปรึกษา

(รศ.ดร. ปราโมทย์ วาดเขียน)

อาจารย์ที่ปรึกษา

(ผศ.ดรจีรสุดา โกษียาภรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องอ่านบาร์โค้ดแบบไร้สาย

Wireless barcode reader

โดย นางสาว กชกร ขุนชนะ รหัส 47010001

นางสาว แก้วนภา โพธิ รหัส 47010063

อาจารย์ที่ปรึกษา รศ.ดร. ปราโมทย์ วาดเขียน

ผศ.ดรจีรสุดา โกยียาภรณ์

บทคัดย่อ

โครงงานนี้ทำการสร้างเครื่องอ่านบาร์โค้ดแบบไร้สาย ซึ่งนำไปใช้ประโยชน์ในหลายด้าน เช่น ระบบคลังสินค้า และ ระบบห้องสมุด ทำให้เกิดความสะดวกในการพกพากรณีที่เป็นสินค้าขนาดใหญ่หรือในบริเวณกว้าง เครื่องอ่านบาร์โค้ดแบบไร้สายนี้ จะมีการทำงานเป็นดังนี้คือ ข้อมูลที่ได้จากการอ่านจะถูกส่งไปยัง โมดูล TRW 2.4 GHz ซึ่งถูกควบคุมโดยไมโครคอนโทรลเลอร์ โมดูล TRW 2.4 GHz ทางด้านส่ง จะทำการส่งข้อมูลดังกล่าวไปยังโมดูล TRW 2.4 GHz ทางด้านรับ ซึ่งจะเชื่อมต่อกับคอมพิวเตอร์ โดยใช้ MAX 232 คอมพิวเตอร์ทางด้านรับจะมีโปรแกรมทำหน้าที่แสดงข้อมูลบาร์โค้ดที่ได้รับดังกล่าวทางหน้าจอ

Abstract

This project constructs the wireless barcode reader which can be used for warehouse , library and ect. It is convenient to carry for the case of large size of goods or in wide area. The operation of the wireless barcode reader is that the read data is sent to the module TRW 2.4 GHz which is controlled by a microcontroller. The module TRW 2.4 GHz at the transmitter is sent that information to the module TRW 2.4 GHz at the receiving site which is connected with the computer by using MAX 232 . The computer located at the receiving site has a program which will display the received barcode information on the monitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จได้ด้วยดีเนื่องจากได้รับความช่วยเหลือและคำแนะนำจากท่าน รศ.ดร.ปราโมทย์ วาดเขียน และ ผศ.ดร. จีรสุดา โกษียาภรณ์ ตลอดระยะเวลาการทำโครงงาน นอกจากนี้ยังได้รับความช่วยเหลือและกำลังใจจากเพื่อนๆทุกท่าน

สุดท้ายนี้ขอขอบพระคุณครอบครัวที่สนับสนุนและเป็นกำลังใจที่ดีเสมอมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
สารบัญ	II
สารบัญรูป	V
สารบัญตาราง	VIII
บทที่	
1 บทนำ	
1.1ความเป็นมาของปริญญานิพนธ์	1
1.2วัตถุประสงค์ของปริญญานิพนธ์	1
2 ทฤษฎีและหลักการ	
2.1 บาร์โค้ด	2
2.1.1 ความหมาย	2
2.1.2 ส่วนประกอบของบาร์โค้ด	2
2.1.3 โครงสร้างของบาร์โค้ด	2
2.1.4 การอ่านบาร์โค้ด	3
2.1.5 เซนเซอร์อ่านบาร์โค้ด	4
2.1.6 ประเภทของเครื่องอ่านบาร์โค้ด	4
2.1.7 มาตรฐานบาร์โค้ด	5
2.2 ไมโครคอนโทลเลอร์	11
2.2.1 คุณลักษณะพื้นฐานของ MCS-51	11
2.2.2 โครงสร้างหน่วยความจำ	11
2.2.2.1 หน่วยความจำโปรแกรมของMCS-51	11
2.2.2.2 หน่วยความจำข้อมูลของMCS-51	12
2.2.3 รีจิสเตอร์ฟังก์ชันพิเศษ	13
2.2.4. การโปรแกรมพอร์ตอนุกรม	14
2.2.4.1รีจิสเตอร์ต่างๆที่เกี่ยวข้องในการใช้งานพอร์ตอนุกรม	14
2.2.4.2 การส่งข้อมูลแบบอนุกรมในโหมด 3	16
2.3 เครื่องรับ-ส่งสัญญาณ FSK TRW 2.4 GHz	16
2.3.1 การจัดขาของโมดูล	17
2.3.2 โหมดการทำงานของ TRW 2.4 GHz	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.3.2.1 Active modes 17

2.3.2.2 Shock Burst 17

2.3.3 Device configuration 21

2.3.4 Data package Description 26

2.4 การสื่อสารผ่านพอร์ตอนุกรม RS-232 27

2.4.1 การสื่อสารข้อมูล 27

2.4.2 มาตรฐาน RS-232 31

2.5ระบบเครือข่าย Ethernet 32

2.5.1 ส่วนประกอบหลักที่สำคัญของเครือข่าย Ethernet 32

2.5.2 เฟรมบนระบบ Ethernet 33

2.5.3 โครงสร้างของสถาปัตยกรรมรูปแบบของ Protocol TCP/IP 34

3 การคำนวณและการสร้าง

3.1 บล็อกไดอะแกรม 50

3.2 ไมโครคอนโทรลเลอร์ 51

3.3 การออกแบบโฟลวชาร์ทของวงจรรวมทางด้านส่ง 52

3.4 การออกแบบโฟลวชาร์ทของวงจรรวมทางด้านรับ 53

3.5 การออกแบบโฟลว์ชาร์ทในการส่งข้อมูลของ Embeded 54

3.6 การออกแบบโฟลว์ชาร์ทการแสดงข้อมูลออกหน้าจอคอมพิวเตอร์ 55

4 การทดลองและผลการทดลอง

4.1การทดลองวัดค่าสัญญาณจากเครื่องอ่านบาร์โค้ด 56

4.2 การทดลองวัดค่าสัญญาณที่ส่งจาก AT89C51 ไปยัง TRW2.4GHz (ด้านส่ง) 57

4.3 การทดลองวัดค่าสัญญาณที่ส่งจาก TRW2.4 GHz ไปยัง AT89C51 (ด้านรับ) 61

4.4 การทดลองแสดงผลเป็นหมายเลขบาร์โค้ดบนไฮเปอร์เทอร์มินอล 64

4.5การทดลองแสดงผลข้อมูลที่เข้ามาในคอมพิวเตอร์โดยใช้โปรแกรม wireshark 65

4.6 การทดลองแสดงข้อมูลทางหน้าจอคอมพิวเตอร์ 66

5 บทสรุปและวิจารณ์

5.1บทวิจารณ์ 67

5.1.1 ส่วนของเครื่องอ่านบาร์โค้ด 67

5.1.2 ส่วนของไมโครคอนโทรลเลอร์และพอร์ตอนุกรม 67

5.1.3 การรับส่งผ่านคลื่นวิทยุ 67

5.1.4 อีเทอร์เน็ตโมดูล (Embeded) 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

5.2บทสรุป	68
กิตติกรรมประกาศ	69
เอกสารอ้างอิง	70
ภาคผนวก	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 โครงสร้างของบาร์โค้ด 3

รูปที่ 2.2 เครื่องอ่านบาร์โค้ด 5

รูปที่ 2.3 ตัวอย่างบาร์โค้ดระบบ UPC 5

รูปที่ 2.4 ตัวอย่างบาร์โค้ดระบบEAN 6

รูปที่ 2.5 ตัวอย่างบาร์โค้ดระบบCODE39 7

รูปที่ 2.6 ตัวอย่างบาร์โค้ดระบบIFT 2 of 5 8

รูปที่ 2.7 ตัวอย่างบาร์โค้ดระบบ CodaBar 8

รูปที่ 2.8 ตัวอย่างบาร์โค้ดระบบ Code128 9

รูปที่ 2.9 ตัวอย่างบาร์โค้ดระบบ Code93 9

รูปที่ 2.10 ตัวอย่างบาร์โค้ดระบบCode49 9

รูปที่ 2.11 ตัวอย่างบาร์โค้ดระบบCode16K 10

รูปที่ 2.12 ตัวอย่างบาร์โค้ดระบบ ISBN/ISSN 10

รูปที่ 2.13 ตัวอย่างบาร์โค้ดระบบ EAN/UCC128 10

รูปที่ 2.14 การจัดพื้นที่หน่วยความจำของไมโครคอนโทรลเลอร์ 12

รูปที่ 2.15 แสดงลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล 12

รูปที่ 2.16 รูปแบบการรับส่งข้อมูลในโหมด 3 16

รูปที่ 2.17 เครื่องส่งสัญญาณ FSK TRW 2.4 GHz 16

รูปที่ 2.18 จับเวลาข้อมูลโดย CPU ส่งโดยเทคโนโลยี Shock Burst 18

รูปที่ 2.19 การใช้กระแส RF โดยใช้และไม่ใช้เทคโนโลยี Shock Burst 18

รูปที่ 2.20 Flow Chart ShockBurst การส่งของระบบย่อย nRF2401 19

รูปที่ 2.21 Flow Cart ShockBurst การรับระบบย่อย nRF2401 20

รูปที่ 2.22 DATA package set-up 21

รูปที่ 2.23 Configuration data word 26

รูปที่ 2.24 Data package Diagram 27

รูปที่ 2.25 แสดงบล็อกไดอะแกรมรูปแบบการสื่อสารแบบขนาน 28

รูปที่ 2.26 แสดงบล็อกไดอะแกรมรูปแบบการสื่อสารแบบอนุกรม 29

รูปที่ 2.27รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 2.28 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 30

รูปที่ 2.29ลักษณะของคอนเน็คเตอร์แบบ DB-9 31

รูปที่ 2.30 ลักษณะโครงสร้างของเฟรมข้อมูล 33

รูปที่ 2.31 ลักษณะของ Ethernet II Frame 33

รูปที่ 2.32 ภาพแสดงการรับส่งข้อมูลผ่านโปรโตคอล TCP/IP 34

รูปที่ 2.33 แสดงการใช้งาน port ของแต่ละโพรโตคอล 36

รูปที่ 2.34 แสดงการส่งข้อมูลจาก Application ไปยัง Host – to – Host Layer 37

รูปที่ 2.35 แสดงกลไกการส่งข้อมูลด้วยโปรโตคอล UDP 39

รูปที่ 2.36 แสดงโครงสร้างของ UDP Header 39

รูปที่ 2.37 Pseudo Header 42

รูปที่ 2.38 แสดงโครงสร้าง IP Header 44

รูปที่ 2.39 แสดงการ Bind IP address 44

รูปที่ 2.40 โครงสร้างของโปรโตคอล TCP/IP 48

รูปที่ 3.1 บล็อกไดอะแกรม 50

รูปที่ 3.2 แสดงวงจรด้านส่ง 51

รูปที่ 3.3 แสดงวงจรด้านรับ 51

รูปที่ 3.4 โฟลวชาร์ทของวงจรรวมทางด้านส่ง 52

รูปที่ 3.5 โฟลวชาร์ทของวงจรรวมทางด้านรับ 53

รูปที่ 3.6 โฟลวชาร์ทของในการส่งข้อมูลของ Embeded 54

รูปที่ 3.7 โฟลว์ชาร์ทการแสดงข้อมูลออกหน้าจอคอมพิวเตอร์ 55

รูปที่ 4.1 แสดงสัญญาณข้อมูลที่อ่านจากเครื่องอ่านบาร์โค้ด (แบบ Asynchronous) 56

รูปที่ 4.2 แสดงสัญญาณข้อมูลแบบ TTLที่วัดจากขา 9 ของMAX-232 56

รูปที่ 4.3 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 1 รหัส 8) 57

รูปที่ 4.4 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 3 รหัส 5) 57

รูปที่ 4.5 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 5 รหัส 9) 58

รูปที่ 4.6 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 8 รหัส 3) 59

รูปที่ 4.7 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 10 รหัส 0) 60

รูปที่ 4.8 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 13 รหัส 1) 60

รูปที่ 4.9 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 1 รหัส 8) 61

รูปที่ 4.10 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 3 รหัส 5) 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 4.11 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 5 รหัส 9) 62
รูปที่ 4.12 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 8 รหัส 3) 63
รูปที่ 4.13 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 10 รหัส 0) 63
รูปที่ 4.14 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 13รหัส 1) 64
รูปที่ 4.15 แสดงหน้าจอโปรแกรมไฮเปอร์เทอร์มินอลเมื่อได้รับหมายเลขบาร์โค้ด 64
รูปที่ 4.16 แสดงหน้าจอโปรแกรม wireshark เมื่อมีการส่ง packet เข้ามาในคอมพิวเตอร์ 65
รูปที่ 4.17 แสดงข้อมูลผ่านทางหน้าจอคอมพิวเตอร์ 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตารางเปรียบเทียบเซนเซอร์อ่านบาร์โค้ดแบบLaserและแบบCCD 4
ตารางที่ 2.2 รหัสเลขฐาน2ของรหัส2ใน5 7
ตารางที่ 2.3 แสดงบิตภายใน PCON 14
ตารางที่ 2.4 แสดงบิตภายใน SCON 15
ตารางที่ 2.5 แสดงการเลือกโหมด 15
ตารางที่ 2.6แสดงลักษณะการทำงานของขาTRW2.4GHz 17
ตารางที่ 2.7 PLL setting 21
ตารางที่ 2.8จำนวนของบิตใน payload 22
ตารางที่ 2.9 แอดเดรสของตัวรับ 1 และ 2 22
ตารางที่ 2.10 จำนวนบิตที่ต้องคงไว้สำหรับ RX address + CRC setting 22
ตารางที่ 2.11 RF operational setting 23
ตารางที่ 2.12 crystal frequency setting 24
ตารางที่ 2.13 RF output power setting 24
ตารางที่ 2.14 ช่องความถี่และการset Rx / Tx 24
ตารางที่ 2.15 ตารางของ Configuration word 25
ตารางที่ 2.16 รายละเอียดของ Data package 27
ตารางที่ 2.17 UDP Checksum คำนวณโดยใช้วิธีเดียวกันกับ IP Header Checksum 41
ตารางที่ 2.18 UDP Port Numbers 42
ตารางที่ 2.19 ค่าต่างๆของ IP Protocol Field 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปริญญานิพนธ์

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

1. การส่งข้อมูลแบบขนาน ข้อมูลจะส่งออกไปทั้งไบต์ในครั้งเดียว โดยที่ทุกบิตจะถูกส่งออกไปพร้อมกัน ซึ่งต้องใช้บัสในการส่งข้อมูลอย่างน้อยเท่ากับจำนวนบิตใน 1 ไบต์
2. การส่งข้อมูลแบบอนุกรม ข้อมูลจะถูกส่งออกไปทีละบิต ดังนั้นจึงใช้บัสในการส่งข้อมูลเพียงเส้นเดียวเท่านั้น

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

โดยการใช้อินฟราเรดที่มีการพัฒนามาอย่างต่อเนื่อง และได้มีการนำมาผลิตขายแล้ว แต่ยังคงมีราคาสูงอยู่จึงยังไม่เป็นที่แพร่หลายมากเท่าไรนัก และการส่งสัญญาณแบบใช้อินฟราเรดนี้ยังมีข้อจำกัดอยู่หลายๆอย่าง เช่น สามารถรับ-ส่ง สัญญาณได้ในระยะทางสั้นๆเท่านั้น อีกทั้งยังต้องจัดให้อุปกรณ์หันส่วนที่รับ-ส่งอินฟราเรดอยู่ในแนวเดียวกันด้วยจึงจะสามารถรับ-ส่งข้อมูลได้

แต่ในปริญญานิพนธ์นี้เราจะใช้หลักการส่งที่อาศัยคลื่นวิทยุ เนื่องจากประโยชน์คือทำให้สามารถที่จะนำอุปกรณ์เชื่อมต่อคอมพิวเตอร์ไปวางไว้ตรงไหนก็ได้ภายในระยะรัศมีที่กำหนด อีกทั้งไม่จำเป็นต้องวางเครื่องรับและเครื่องส่งให้ตรงกันเหมือนกับระบบที่ใช้อินฟราเรด จึงทำให้เกิดความสะดวกสบายเป็นอย่างมากกับผู้ใช้งานและไม่ต้องเสียค่าใช้จ่ายในการเดินสายและไม่ยุ่งยาก

1.2 วัตถุประสงค์ของปริญญานิพนธ์

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

สำหรับปริญญานิพนธ์ฉบับนี้มีทฤษฎีและหลักการแบ่งออกเป็น 5 ส่วน ได้แก่ ส่วนบาร์โค้ด, ส่วนไมโครคอนโทรลเลอร์, ส่วนTRW 2.4GHz,การสื่อสารผ่านพอร์ทอนุกรมและระบบเครือข่ายEthernet

2.1 บาร์โค้ด

2.1.1 ความหมาย

บาร์โค้ดคือสัญลักษณ์รหัสแท่งที่ใช้แทนข้อมูลตัวเลขมีลักษณะเป็นแถบมีความหนาบางแตกต่างกันขึ้นอยู่กับตัวเลขที่กำกับอยู่ข้างล่าง การอ่านข้อมูลจะอาศัยหลักการสะท้อนแสง เพื่ออ่านข้อมูลเข้าเก็บในคอมพิวเตอร์โดยตรงไม่ต้องผ่านการกดปุ่มที่แป้นพิมพ์ ระบบนี้เป็นมาตรฐานสากลที่นิยมใช้กันทั่วโลก ลักษณะที่เป็นลายเส้นสีขาว - ดำ จะมีขนาดความกว้างของลายเส้นตามมาตรฐานแต่ละชนิดของบาร์โค้ด และมีข้อมูลตัวอักษรเป็นส่วนที่แสดงความหมายของข้อมูลลายเส้นสำหรับให้อ่านเข้าใจได้ และมีแถบว่าง (Quiet Zone) ใช้เป็นส่วนที่เครื่องอ่านบาร์โค้ดกำหนดขอบเขตของบาร์โค้ดและกำหนดค่าให้กับสีขาว (ความเข้มของการสะท้อนแสงในสีของพื้นผิวแต่ละชนิดที่ใช้แทนสีขาว) โดยแต่ละเส้นจะมีความยาวเท่ากันเรียงตามลำดับในแนวนอนจากซ้ายไปขวา ซึ่งจะเป็นประโยชน์ต่อเครื่องอ่านบาร์โค้ด (Barcode Scanner) ในการอ่านข้อมูลที่บันทึกไว้

2.1.2 ส่วนประกอบของบาร์โค้ด

บาร์โค้ดจะมีส่วนประกอบต่างๆ ดังต่อไปนี้

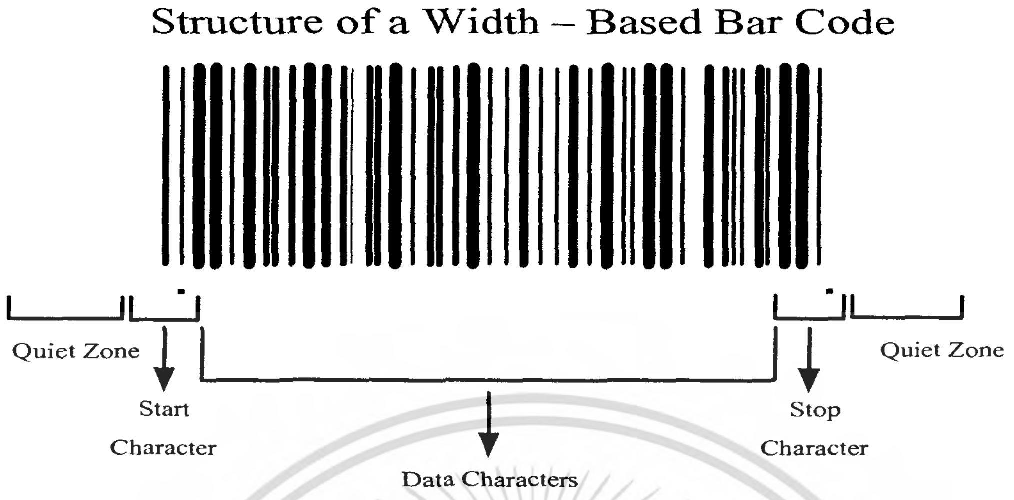
- Quiet Zone เป็นบริเวณที่ว่างเปล่าไม่มีการพิมพ์ข้อความใดๆ โดยจะอยู่ก่อนและหลังบาร์โค้ด
- Start/ Stop Character เป็นบริเวณแถบแท่งหรือช่องว่าง เพื่อเตรียมสั่งให้เซนเซอร์เริ่มต้นหรือหยุดบาร์โค้ด
- Data เป็นบริเวณแถบแท่งหรือช่องว่างที่แทนข้อมูลต่างๆ ที่เราต้องการ
- Check Digit เป็นบริเวณแถบแท่งที่ไว้สำหรับเก็บค่าตัวเลข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

2.1.3 โครงสร้างของบาร์โค้ด

บาร์โค้ดประกอบด้วยแถบสีดำและสีขาว โดยความกว้างของแถบสีดำสลับขาวเป็นรหัสแทนข้อมูล เรียงจากซ้ายไปขวา การถอดรหัสจำเป็นต้องทาบความกว้างของแถบดำและแถบขาวนำไปเทียบกับตารางมาตรฐานเครื่องอ่านบาร์โค้ด ประกอบด้วยหัวอ่านอินฟาเรด แบบปากกาและแบบวงจรถอดรหัส การใช้งานเริ่มต้นด้วยการกวาดหัวอ่านผ่านบาร์โค้ด ซึ่งหัวอ่านจะมีตัวตรวจจับแสงสะท้อนไปจุดชนวนวงจรอิเล็กทรอนิกส์ ทำให้เกิดคลื่นสัญญาณไฟฟ้าแบบพัลส์โดยความกว้างของรูปคลื่นจะเป็นสัดส่วนกับความกว้างของแถบโค้ด ต่อจากนั้นวงจรถอดรหัสจะตรวจสอบความกว้างของรูปคลื่นแล้ว นำไปเปรียบเทียบกับแถบขาวดำทั้งหมด ที่แทนข้อมูลตัวเลขหรือตัวอักษร โดยปกติเครื่องอ่านจะต่อเข้ากับ

พระจอมเกล้าเจ้าคุณทหาร
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ ดังนั้นวงจรภายในเครื่องอ่านจะส่งข้อมูลตัวเลขที่ถอดรหัสได้ไปยังคอมพิวเตอร์เพื่อประมวลผลต่อไป ดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของบาร์โค้ด

2.1.4 การอ่านบาร์โค้ด

ในการอ่านบาร์โค้ดใช้หลักการเปลี่ยนรหัสแถบใช้เป็นรหัสแอสกี้ โดยอาศัยความแตกต่างกันระหว่างแถบเข้มและพื้นที่ว่าง โดยที่พื้นที่ว่าง(ปกติจะเป็นสีขาวหรือสีอ่อน)จะมีการสะท้อนกลับของแสงได้มากกว่าบริเวณที่เป็นแถบเข้ม(ซึ่งใช้สีดำหรือสีอื่น ที่มีความเข้มมาก) ตัวอ่าน(Barcode reader) จะประกอบด้วย ตัวกำเนิดแสงผ่านเลเซอร์ออกมาโดยถูกบังคับทิศทางให้มีจุดรวมแสงที่เล็กที่สุด กับตัวรับแสงที่มีความไวสูง ทั้ง 2 อย่างนี้จะบรรจุไว้ที่ตัวอ่านเดียวกันที่มีหลายรูปแบบ แต่แบบที่เป็นพื้นฐานที่สุดอยู่ในรูปคล้ายปากกาขนาดใหญ่(Wand type) ตัวอ่านจะถูกสแกนผ่านบาร์โค้ด ในขณะที่ตัวกำเนิดแสงจะทำให้เกิดแสงส่งผ่านเลนส์ไปกระทบบนบาร์โค้ดและสะท้อนกลับจากแถบ(แถบและช่องว่าง) กลับไปยังตัวรับแสง(Photosensor) ที่เกิดค่าความแตกต่างขึ้นตามหลักการสะท้อนกลับในแต่ละแถบทำให้เกิดสภาวะลอจิก"0" และลอจิก"1" ขึ้นตามที่กล่าวมาแล้วข้างต้นซึ่งเมื่อรวมสภาวะลอจิก "0" และ"1" ทั้งหมดตลอดความกว้างของทุกแถบแล้วจะตรงกับแบบที่ได้กำหนดไว้แล้ว ในตัวอ่านบาร์โค้ดจะใช้ตัวกำเนิดแสงสีแดงหรือสีขาว แต่ส่วนใหญ่จะใช้แสงสีแดงเนื่องจากแสงสีขาวต้องการพลังงานและความเข้มของแสงสูงมากกว่าแสงสีแดง แสงสีแดงสามารถอ่านรหัสที่พิมพ์ด้วยสีต่างๆได้ยกเว้นรหัสที่พิมพ์ด้วยสีแดง

องค์ประกอบที่สำคัญ 2 ประการที่จำเป็นอย่างมากในการอ่านบาร์โค้ดได้ถูกต้อง ประการแรกคือ พื้นที่ภายในแถบและช่องว่าง จะต้องทำให้เกิดความแตกต่างของการสะท้อนกลับอย่างมาก(Contrast) เช่น แถบสีดำและช่องว่างสีขาว เป็นต้น ซึ่งปกติความแตกต่างนี้ต้องอยู่ในช่วงระหว่างอัตรา 30-90 เปอร์เซ็นต์ขึ้นไป ประการที่สองคือ ความกว้างระหว่างแถบกว้างหรือ ช่องว่างกว้างต่อแถบแคบ หรือ ช่องว่างแคบจะเป็นอัตราส่วน 2:0.5 , 2:1 , และ 3:1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 เซนเซอร์อ่านบาร์โค้ด

เซนเซอร์อ่านบาร์โค้ดส่วนใหญ่แล้วแบ่งออกเป็น 2 ชนิดคือ

1. แบบ Laser จะใช้อ่านบาร์โค้ดที่ติดในสายการผลิต ซุปเปอร์มาร์เก็ต และคลังสินค้า หลักการทำงานคือ ลำแสงเลเซอร์ถูกปล่อยออกจากเลเซอร์ไดโอดมากระทบกับกระจกแบบหลายเหลี่ยมเพื่อที่จะสแกนบาร์โค้ด เมื่อลำแสงเลเซอร์กระทบบาร์โค้ดจะกระจายออก และถูกส่งมาที่โฟโต้ไดโอด ลักษณะของลำแสงที่กระจายตามบาร์โค้ดจะถูกแปลงไปเป็นสัญญาณอนาลอก จากนั้นทำการแปลงสัญญาณเป็นดิจิตอล ลักษณะของสัญญาณดิจิตอลจะขึ้นอยู่กับขนาดของแท่ง และที่ว่างในแถบบาร์โค้ด จากนั้นก็จะแปลงรหัสเป็นข้อมูลผ่านพอร์ทคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์ไปประมวลผลหรือเก็บข้อมูลไว้ใช้

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

ตารางที่ 2.1 ตารางเปรียบเทียบเซนเซอร์อ่านบาร์โค้ด แบบ Laser และแบบ CCD

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

2.1.6 ประเภทของเครื่องอ่านบาร์โค้ด

เครื่องอ่านบาร์โค้ดสามารถแบ่งตามลักษณะการใช้งานได้ดังนี้

1. Moving Bean Scanner เครื่องอ่านอยู่กับที่ แต่แสงฉายกวาดไปที่สินค้า เพื่อหาบาร์โค้ดที่กำกับบนสินค้านั้น

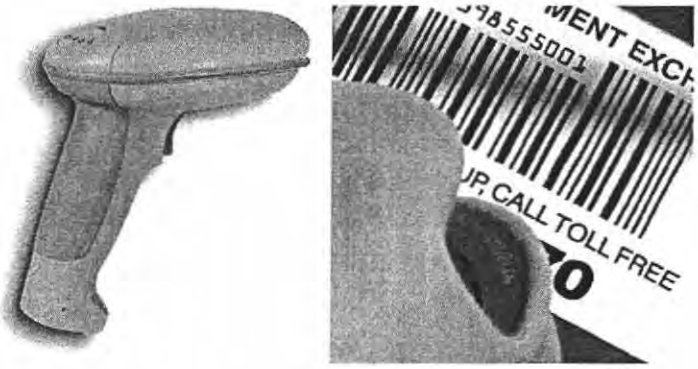
2. Fixed Bean Scanner เครื่องอ่านอยู่กับที่ลำแสงไม่เคลื่อนที่สินค้าเคลื่อนที่ผ่านจุดที่แสงฉาย

3. Hand Held Scanner เครื่องอ่านที่ต้องใช้คนควบคุมและถือได้ เหมาะสำหรับการอ่านบาร์โค้ดของสินค้าที่มีขนาดใหญ่เคลื่อนที่ยาก เช่น ม้วนกระดาษใหญ่ที่ผลิตจากโรงงาน

4. Wand Scanner เครื่องอ่านที่ให้แสงสีแดงอินฟราเรดในการอ่านต้องใช้เครื่องอ่านสัมผัสกับแถบบาร์โค้ด

5. Hand Held Laser Scanner เครื่องอ่านที่มีหลักการทำงานแบบ Moving Bean Scanner ที่ให้แสงเลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 เครื่องอ่านบาร์โค้ด

2.1.7 มาตรฐานบาร์โค้ด

บาร์โค้ดสามารถแบ่งออกตามชนิดของข้อมูลได้ดังนี้

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

บาร์โค้ดมาตรฐานที่ใช้กันอยู่ปัจจุบันมีประมาณ 12 ระบบดังนี้

1. UPC – Uniform Product Code

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

6 90173 80003 5

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

รูปที่ 2.3 ตัวอย่างบาร์โค้ดระบบ UPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรหัสของ UPC รหัสทางซ้ายจะใช้กับรหัสแถบแบบ UPC ในทางโซนซ้าย ส่วนรหัสทางขวาจะใช้ได้กับโซนทางขวาของรหัสแถบชนิด UPC เท่านั้น จะใช้สลับกันไม่ได้ ในส่วนของรหัสทางซ้ายจะขึ้นต้นด้วยบิต 0 และลงท้ายบิต 1 เสมอ จะมีการตรวจสอบเป็นแบบบิตคี่ (Odd parity) ส่วนรหัสทางขวาจะกลับด้านกับรหัสทางซ้าย คือ มีบิต 1 เป็นบิตเริ่มต้น และ 0 เป็นบิตสิ้นสุด การตรวจสอบบิตเป็นแบบคู่ (even parity) นอกจากนี้จากตารางเลขรหัสทางซ้ายและทางขวายังเป็นเลขแบบ 1's complement ซึ่งกันและกัน

2. EAN – European Article Number

เป็นรหัสที่นิยมใช้กับสินค้าที่มาจากต่างประเทศหลายประเภท หรือสินค้าที่ส่งออกไปขายต่างประเทศ ผู้อ่านหลายๆ คนคงได้เคยเห็นรหัสชนิดนี้ปรากฏบนสินค้าที่ซื้อ โครงสร้างของรหัสชนิดนี้ต่างจากรหัสแถบประเภทอื่นๆ โดยสิ้นเชิง

จากรูปที่ 2.4 จะเห็นได้ว่ารหัสแถบชนิดนี้แบ่งเป็น 2 ส่วน ซึ่งถูกแบ่งด้วยแถบสีดำเล็กๆแต่ยาวกว่าแถบอื่น 2 แถบคั่นอยู่ตรงกลาง (เลขรหัสฐาน 2 ของแถบคั่นกลางนี้เป็น 01010) และยังมีแถบลักษณะเดียวกัน 2 ชุด อยู่ทางซ้าย-ขวาสุด(เลขรหัสฐาน 2 ของแถบนี้คือ 101)แถบทั้ง 3 ชุดนี้เรียกว่า Guide bar ซึ่งปกติจะมีความยาวกว่าแถบอื่นๆ เป็นข้อสังเกตทำให้แบ่งรหัสแถบเป็น 2 ส่วน คือ โซนทางซ้ายและโซนทางขวา หลักสุดท้ายทางซ้ายทางขวาสุด เป็นตัวกลางตรวจสอบความถูกต้อง (Check Digit) ซึ่งคำนวณมาจากหลักที่เหลือ โดยตัวตรวจสอบทางซ้ายสุดมาจากเลข 5 หลักที่อยู่ทางด้านขวา ซึ่งแถบสำหรับตรวจสอบนี้บางครั้งก็พิมพ์ยาวเท่ากับส่วนที่เป็นข้อมูล

6 901738 000352

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

รูปที่ 2.4 ตัวอย่างบาร์โค้ดระบบ EAN

3. CODE 39

รหัส 3 ใน 9 เป็นรหัสที่ใช้แทนตัวอักขระทั้งหมด 44 อักขระเป็นอักษรตัวใหญ่ 26 รหัส เลข 0-9 10 รหัส และอักขระพิเศษอีก 3 รหัส เป็นการประยุกต์ ใช้รหัส 2 ใน 5 โดยการนำเอาส่วนที่เป็นแถบดำ 5 แถบและแถบว่าง 4 แถบ รวมเป็น 9 แถบแทน 1 รหัส ในแถบดำ 5 แถบนั้น ประกอบด้วยแถบกว้างที่เป็นบิต 1 อยู่ 2 แถบ และแถบแคบที่เป็นบิต 0 อยู่ 3 แถบ ส่วนแถบว่าง(Space) 4 แถบ ประกอบด้วยแถบกว้างที่เป็นบิต 1 อยู่ 1 แถบและแถบแคบที่เป็นบิต 0 อีก 3 แถบ ดังนั้นเมื่อรวมทั้งหมด 9 แถบจะเป็นบิต 1 อยู่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แถบ และบิต 0 อยู่ 6 แถบ รหัส 3 ใน 9 มีส่วนเริ่มต้น (Start code) และสิ้นสุด (Stop code) ด้วยรหัสเดียวกันคือ (Asterisk) ซึ่งมีรหัสฐาน 2 เป็นแถบ 00110 และช่องว่าง 1000 ข้อดีของรหัสชนิดนี้คือ ใช้งานได้กว้างขวางมากขึ้น เพราะสามารถใช้ตัวเลขปนกับตัวอักษรและเครื่องหมายต่างๆได้

รูปที่ 2.5 ตัวอย่างบาร์โค้ดระบบ CODE 39

4.ชนิดรหัส 2 ใน 5 (2 of 5 code)

เป็นรหัสที่มีใช้ตั้งแต่ ค.ศ. 1960 เป็นแบบที่ง่ายที่สุดในการใช้งาน การที่ชื่อเรียกว่า 2ใน5 เพราะใน 1 รหัส จะประกอบไปด้วยแถบ 5 แถบ(5 บิต)แต่จะมีแถบกว้างที่มีค่า(แถบกว้าง)เพียง 2 แถบ(2 บิต) เท่านั้น ส่วนบิตที่เหลือเป็น 0 ทั้งหมดคือ การแทนด้วยแถบแคบ(Narrow bar) 3 แถบโดยไม่นำส่วนที่เป็นช่องว่าง(Space bar) มาใช้เลย รหัส 2 ใน 5 นี้เป็นรหัสที่ใช้แทนข้อมูลได้เฉพาะตัวเลข 0-9 เพียงแค่ 10 รหัสเท่านั้น เริ่มต้นจาก Start code 3 บิต คือ 110 (แถบกว้าง 2 และแถบแคบ 1 กับปิดท้ายด้วย stop code 3 บิต คือ 101ส่วนรหัสทั้ง 5 บิตที่แทนเลข 0-9 ดูได้ในตารางในรหัส Interleaved 2 of 5)

ตารางที่ 2.2 รหัสเลขฐาน 2 ของรหัส 2 ใน 5

ตัวเลข	เลขฐาน 2 ทั้ง 6 แบบ				
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ITF – INTERLEAVE 2 of 5

รหัสแบบนี้พัฒนามาจากแบบ 2 ใน 5 เนื่องจากรหัส 2 ใน 5 ไม่ได้นำส่วนที่เป็นช่องว่างกว้างและช่องว่างแคบมาใช้ ใช้แต่เพียงแถบกว้างและแถบแคบจึงทำให้ความหนาแน่นของข้อมูลน้อยลงนั่นคือ เมื่อต้องการบรรจุข้อมูลต่อเนื่องหลายตัวเลข จะต้องใช้แถบที่มีความกว้างมากขึ้น รหัส 2 ใน 5 แบบสอดแทรก ได้ดัดแปลงนำส่วนที่เป็นช่องว่างทั้ง 2 ชนิด มาใช้งานด้วยโดยการสอดแทรกรหัสลงไปอีก 1 รหัสทุกๆช่อง 5 แถบของรหัสปกติที่เป็นแถบสีดำแต่ก็ยังแทนรหัสตัวเลข 0-9 ได้เพียง 10 รหัสเท่านั้น ดังรูปที่ 2.6

Interleaved 2 of 5
MEDIUM - NBW .4mm / 2.5:1 Ratio
12345678
TRONIC X / P-105

รูปที่ 2.6 ตัวอย่างบาร์โค้ดระบบ IFT 2 of 5

6. CODABAR

รหัส Codabar ประกอบด้วย 7 บิตโดย 4 บิตเป็นแถบดำและ 3 บิตเป็นช่องว่าง ใช้แทนตัวเลข 0-9 เครื่องหมาย - $: / . + A B C และ D รหัส Codabar ที่สมบูรณ์จะต้องมีรหัสที่ใช้แทนตัวอักษร A B C หรือ D (เช่น A = 00110010) เป็นส่วนเริ่มต้นหรือสิ้นสุด ภายในประกอบด้วยรหัสของ Codabar ที่เป็นตัวเลขและเครื่องหมายซึ่งทำให้มีความยาวที่ไม่แน่นอนเพราะ 12 รหัสแรกมีบิตที่เป็น 1 อยู่ 2 บิต 4 รหัส ต่อมามีบิต 1 อยู่ 3 บิต (Codabar ใช้ทั้งแถบดำและขาวแทนข้อมูลใน 1 รหัส) และ 4 รหัสสุดท้ายเป็นรหัส A, B, C, D กำหนดขึ้นมาเพื่อใช้เป็นรหัสเริ่มต้นและสิ้นสุด (Start / Stop code)ดังตัวอย่างรูปที่ 2.7

CODABAR
0123456789

รูปที่ 2.7 ตัวอย่างบาร์โค้ดระบบ CodaBar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. CODE 128

ได้ถูกพัฒนาขึ้นและยอมรับว่าได้ใช้เป็นทางการในสหรัฐอเมริกา เมื่อปี ค.ศ. 1981 นิยมใช้ในวงการดีไซน์เนอร์และแฟชั่น ปัจจุบันกำลังเริ่มนิยมใช้ในสหรัฐอเมริกา ดังตัวอย่างรูปที่ 2.8

123456789012

Code 128

รูปที่ 2.8 ตัวอย่างบาร์โค้ดระบบ Code 128

8. CODE 93

ได้เริ่มพัฒนาขึ้นในปี ค.ศ. 1982 ปัจจุบันเริ่มนิยมใช้ในสหรัฐอเมริกา ดังตัวอย่างรูปที่ 2.9

CODE93 12345678

รูปที่ 2.9 ตัวอย่างบาร์โค้ดระบบ Code 93

9. CODE 49

ได้เริ่มพัฒนาขึ้นในปี ค.ศ. 1987 โดย Dr. Davis Allais ผู้คิดค้น CODE 39 ได้ปรับปรุงพัฒนาให้บรรจุข้อมูลได้มากขึ้นด้วยพื้นที่เท่าเดิมดังตัวอย่างรูปที่ 2.10

Code 49

รูปที่ 2.10 ตัวอย่างบาร์โค้ดระบบ Code 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. CODE 16K

เหมาะสำหรับอุตสาหกรรมผลิตสินค้าที่มีขนาดเล็กมาก พื้นที่ในการใส่บาร์โค้ดน้อย เช่น อุปกรณ์อะไหล่เครื่องไฟฟ้า ดังตัวอย่างรูปที่ 2.11

Code 16K

รูปที่ 2.11 ตัวอย่างบาร์โค้ดระบบ Code 16K

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

ใช้สำหรับหนังสือและนิตยสาร ดังตัวอย่างรูปที่ 2.12

ISBN 0-330-28987-X

9 780330 289870

รูปที่ 2.12 ตัวอย่างบาร์โค้ดระบบ ISBN/ISSN

12. EAN/ UCC 128 (shipping container code)

เป็นระบบใหม่ ซึ่งเป็นการร่วมมือกันระหว่าง EAN ของยุโรป และ UCC ของสหรัฐอเมริกา โดยนำเอาระบบ EAN มาใช้ร่วมกับ CODE 128 เพื่อบอกรายละเอียดของสินค้ามากขึ้น เช่น วันเดือนปีที่ผลิต ครั้งที่ผลิต วันที่สั่งซื้อ มีกี่สี กี่ขนาด เป็นต้น ดังตัวอย่างรูปที่ 2.13

1 234567 890128

รูปที่ 2.13 ตัวอย่างบาร์โค้ดระบบ EAN/UCC128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ไมโครคอนโทรลเลอร์

2.2.1 คุณลักษณะพื้นฐานของ MCS-51

คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
2. ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช สามารถลบและเขียนใหม่ได้พันครั้ง
3. หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีพรอมเพิ่มเติม
4. ขาพอร์ทเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุทและเอาท์พุท
5. มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
6. ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
7. สามารถรองรับแหล่งกำเนิดอินเตอร์รัพต์ได้ 6 ประเภท
8. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
9. มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

2.2.2 โครงสร้างหน่วยความจำ

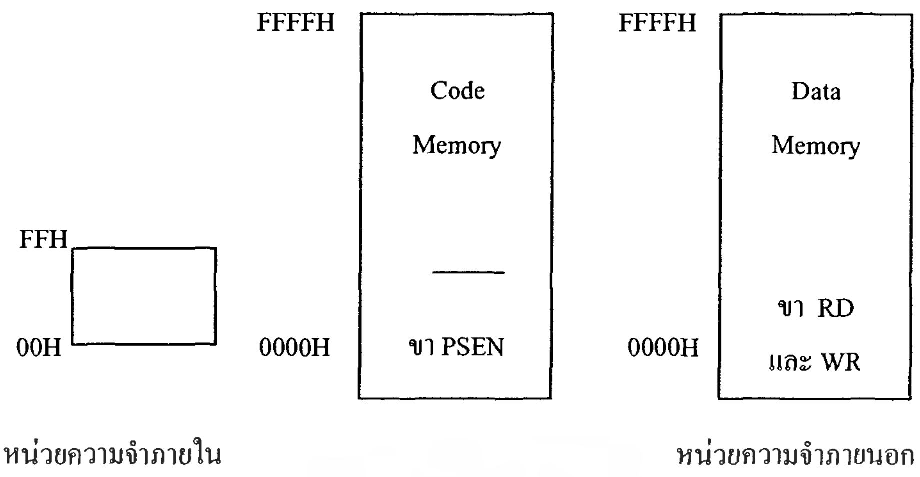
2.2.2.1 หน่วยความจำโปรแกรมของ MCS-51

หน่วยความจำโปรแกรม(Program Memory) มีไว้เพื่อบรรจุคำสั่งหรือโปรแกรมที่ผู้ใช้พัฒนาขึ้น จัดเก็บไว้ในหน่วยความจำ โดยอาจจะประกอบอยู่ภายในตัวของไอซี 89C51 เอง หรือเป็นไอซีหน่วยความจำ EPROM หรือ ROM แยกออกต่างหากได้ ในกรณีหลังจำเป็นต้องมีการใช้พอร์ทอินพุทเอาท์พุท ทำหน้าที่เป็นบัสแอดเดรสและบัสข้อมูลเพื่อให้สามารถติดต่อกับหน่วยความจำมาตรฐานทั่วไปได้

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

การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ 89C51 จะมีการจัดพื้นที่ดังรูปที่ 2.14 โดยที่ไมโครคอนโทรลเลอร์ 89C51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมสูงสุดได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะตามตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ภายในตัวไอซี ไมโครคอนโทรลเลอร์เองและหน่วยความจำภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

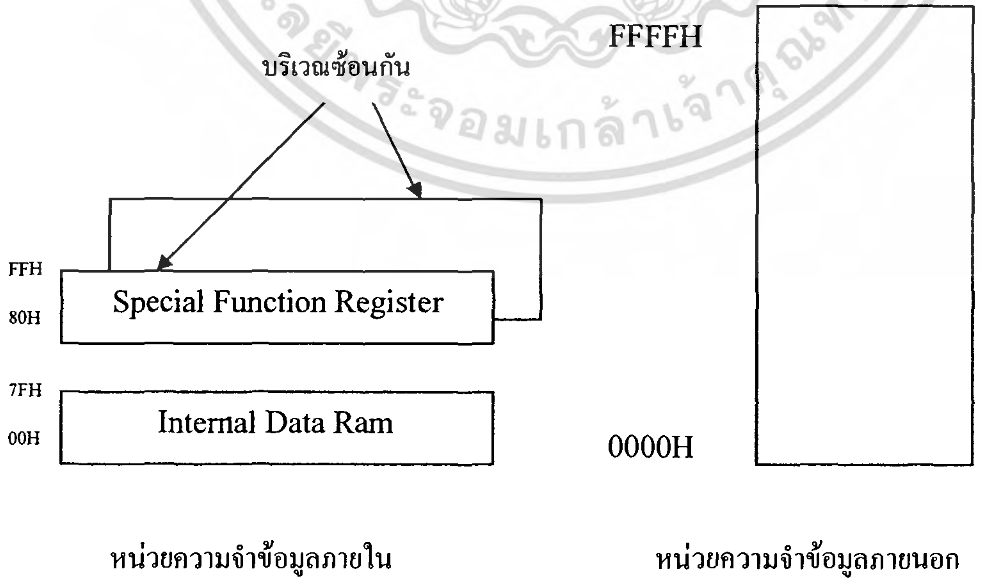
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การจัดพื้นที่หน่วยความจำของไมโครคอนโทรลเลอร์

2.2.2.2 หน่วยความจำข้อมูลของ MCS-51

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยที่หน่วยความจำข้อมูลจะมีลักษณะเป็นหน่วยความจำ RAM แบบสแตติก (Static) ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับระบบ ก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ในหน่วยความจำนี้สูญหายไป พื้นที่ของหน่วยความจำข้อมูลของ 89C51 สามารถมีได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตำแหน่งที่ตั้งของหน่วยความจำนั้นดังลักษณะในรูปที่ 2.15 คือ หน่วยความจำโปรแกรมภายใน (Internal Date Memory) หรือ RAM ที่อยู่ภายในตัวไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 2.15 แสดงลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register)

เป็นรีจีสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์หรือพอร์ทของ 89C51 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจีสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อรีจีสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจีสเตอร์นั้นก็ได้

- แอกคิวมูเลเตอร์ (Accumulator)

เป็นรีจีสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บข้อมูลที่ส่งให้หน่วยทำงานในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การใช้งานในโปรแกรมจะเรียกว่า รีจีสเตอร์ A

- รีจีสเตอร์ B

เป็นรีจีสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณหารตัวเลข ในกรณีที่ไม่ใช้การคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจีสเตอร์ทั่วไปได้

- โปรแกรมเคาน์เตอร์(Program Counter)

เป็นรีจีสเตอร์ที่ใช้สำหรับการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรมซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานในโปรแกรมจะเรียกว่า รีจีสเตอร์ PC

- สแตคพอยน์เตอร์(Stack Pointer)

เป็นรีจีสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ของบริเวณสแตคสำหรับเก็บข้อมูลแอคคิวมูเลเตอร์รีจีสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม ค่าเริ่มต้นของสแตคจะอยู่ที่ตำแหน่ง 07H การใช้งานในโปรแกรมจะเรียกว่า รีจีสเตอร์ SP

- ตัวชี้ข้อมูลหรือดาต้าพอยน์เตอร์(Data Pointer)

เป็นรีจีสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจีสเตอร์ DPTR และสามารถใช้งานแยกออกเป็นรีจีสเตอร์ขนาด 8 บิต สองตัว คือ รีจีสเตอร์ DPH และ DPL เพื่อเก็บค่าแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรม หรืออาจเป็นแอดเดรสของอุปกรณ์ภายนอก

- โปรแกรมสเตตัสเวิร์ด(PSW)

รีจีสเตอร์นี้ทำหน้าที่บอกถึงแฟลกซ์สภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์(Bank) ของรีจิสเตอร์ที่ใช้งานด้วย

- รีจีสเตอร์ที่เกี่ยวข้องกับพอร์ท (Port Register)

รีจีสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ทอินพุทเอาท์พุทโดยตรงซึ่งจะเป็นรีจีสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะการอินพุทหรือการเอาท์พุทข้อมูลได้

- รีจีสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล

- รีจีสเตอร์ PCON

เป็นรีจีสเตอร์ควบคุมการทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรแกรมเซ็นเซอร์ การกำหนดอัตราทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรมและแฟลกซ์ สภาวะการทำงานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีจีสเตอร์ IP,IE,TMOD,SCON

เป็นกลุ่มรีจีสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอินเทอร์รัพต์ต่างๆ

2.2.4 การโปรแกรมพอร์ทอนุกรม

โดยปกติแล้ว MCS-51 จะเป็นไมโครคอนโทรลเลอร์ที่มีความสามารถในการรับข้อมูลจากภายนอกและนำมาประมวลผล พร้อมทั้งสามารถส่งสัญญาณเพื่อทำการควบคุมการทำงานของอุปกรณ์ต่างๆ ได้อย่างดี และในส่วนของการติดต่อสื่อสารข้อมูล (Data Communication) กับระบบภายนอกอื่นๆก็สามารถกระทำโดยผ่านทางพอร์ทอนุกรม (Serial Port) ซึ่งพอร์ทอนุกรมนี้ จะเป็นส่วนที่เหมาะสมในการรับ หรือส่งข้อมูลในระยะทางไกลได้ดีกว่าพอร์ทขนาน

2.2.4.1 รีจิสเตอร์ต่างๆที่เกี่ยวข้องในการใช้งานพอร์ทอนุกรม

1. รีจิสเตอร์ควบคุมไทม์เมอร์เนื่องจากว่าการใช้งานพอร์ทอนุกรมนั้นมีสิ่งที่ต้องคำนึงถึงคือ อัตราการรับ-ส่งข้อมูล หรือ เรียกว่า อัตราบอด (Baud Rate) คือจังหวะการเลื่อนข้อมูลเข้าหรือออกจาก 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. รีจิสเตอร์ควบคุมการลดกำลังเนื่องจาก การสร้างอัตราบอดนั้นจะต้องนำบิตในรีจิสเตอร์ PCON มาใช้ในการคำนวณข้อมูลของ TH1 ดังนั้นรีจิสเตอร์ที่ใช้คือ

- PCON ตำแหน่ง 87H ทำหน้าที่ในการคำนวณข้อมูลที่จะใส่ในรีจิสเตอร์ TH1 ดังนี้

ตารางที่ 2.3 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 แฟลกใช้งานทั่วไป ไม่เกี่ยวข้องกับการควบคุมใด ๆ

GF2 แฟลกใช้งานทั่วไป ไม่เกี่ยวข้องกับการควบคุมใด ๆ

PD Power down Bit 1 = หยุดออสซิลเลเตอร์ของซีพียูสัญญาณรีเซต หรือ อินเตอร์รัพต์ (83C154, 83C154D เท่านั้น) ที่จะยกเลิกโหมดนี้

IDL Idle Mode Bit 1 = หยุดการทำงานของซีพียูสัญญาณรีเซต หรือ อินเตอร์รัพต์เท่านั้นที่จะยกเลิกโหมดนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. รีจิสเตอร์ควบคุมการอินเตอร์รัพต์ เนื่องจากว่า MCS-51 สามารถใช้งานพอร์ทอนุกรมในลักษณะของการอินเตอร์รัพต์ จึงมีรีจิสเตอร์ที่เกี่ยวข้องดังนี้

- IE ตำแหน่ง A8H ทำหน้าที่ยอมให้เกิดการอินเตอร์รัพต์จากพอร์ทอนุกรมได้หรือไม่
- IP ตำแหน่ง B8H ทำหน้าที่จัดลำดับความสำคัญของการอินเตอร์รัพต์

4. รีจิสเตอร์ควบคุมพอร์ทอนุกรม การใช้งานพอร์ทอนุกรมจะขึ้นอยู่กับรีจิสเตอร์โดยตรงคือ

- SBUF ตำแหน่ง 99H ทำหน้าที่เป็นบัฟเฟอร์การรับหรือส่งข้อมูลของพอร์ทอนุกรม
- SCON ตำแหน่ง 98H ทำหน้าที่ควบคุมและกำหนดโหมดการใช้งานพอร์ทอนุกรมทั้งหมดซึ่งมีรายละเอียดของบิตต่าง ๆ ดังนี้

ตารางที่ 2.4 แสดงบิตภายใน SCON

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

ตารางที่ 2.5 แสดงการเลือกโหมด

SM0, SM1 บิตเลือกโหมดการทำงาน

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

SM2 บิตเลือกการทำงานแบบ Single Processor Environment หรือ Multiprocessor Environment

1: เลือก Multiprocessor Environment ใช้ได้กับโหมด 2, 3

0: เลือก Single Processor Environment ใช้ได้กับทุกโหมด

REN บิตควบคุมให้รับหรือไม่รับข้อมูล

1: ให้รับข้อมูลได้ 0: ห้ามรับข้อมูล

TB8 ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2, 3 ให้ใส่ในบิตนี้ได้เลย

RB8 ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมาเก็บในบิตนี้

TI แฟลก TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4.2 การส่งข้อมูลแบบอนุกรมในโหมด 3

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

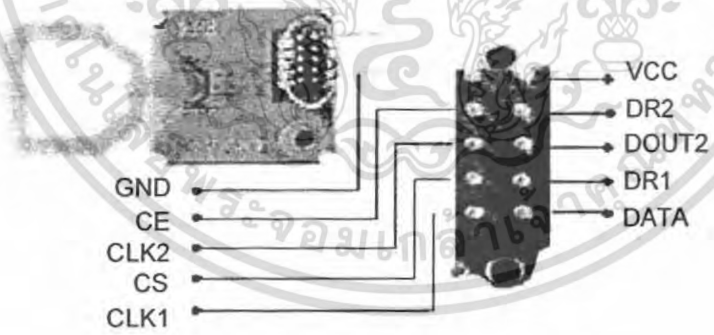
สมการแสดงการคำนวณหาอัตราบอดในโหมด 3

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

รูปที่ 2.16 รูปแบบการรับส่งข้อมูลในโหมด 3

2.3 เครื่องรับ-ส่งสัญญาณ FSK TRW 2.4 GHz

TRW2.4GHzเป็นFSKmoduleซึ่งสามารถกำหนดให้เป็นตัวรับหรือส่งได้โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานและยังสามารถทำการเข้ารหัสและตรวจสอบความผิดพลาดได้ภายในตัวของมันเอง ซึ่งมีลักษณะดังรูปที่ 2.17



รูปที่ 2.17 เครื่องส่งสัญญาณ FSK TRW 2.4 GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3.1 การจัดขาของโมดูล

การจัดขาของโมดูลมีรายละเอียดดังตารางที่ 2.6

ตารางที่ 2.6 แสดงลักษณะการทำงานของขา TRW2.4GHz

ขา	ชื่อ	ลักษณะการทำงาน	รายละเอียด
1	GND	Power	Ground(0V)
2	CE	Input	Chip Enable activates RX or TX mode
3	CLK2	I/O	Clock output/input for RX data channel2
4	CS	Input	Chip Select activates Configuration mode
5	CLK1	I/O	Clock Input(TX)&I/O(RX)for data channel 1
6	DATA	I/O	RX data channel 1/TX data input /3 wire interface
7	DR1	Output	RX data ready at data channel 1 (Shock Burst only)
8	DOUT2	Output	RX data channel 2
9	DR2	Output	RX data ready at data channel 2 (Shock Burst only)
10	VCC	Power	Power Supply(+3V DC)

2.3.2โหมดการทำงานของ TRW 2.4 GHz

2.3.2.1 Active modes

ระบบย่อย nRF2401 มี 2 Active โหมด (Tx / Rx)

- Shock Burst
- Direct Mode (not supported by nRF2401)

ฟังก์ชันการทำงานของอุปกรณ์ในโหมดนี้เลือกโดยเนื้อหาของ Configuration word โดย Configuration word จะแสดงในส่วนของ Configuration

2.3.2.2 Shock Burst

เทคโนโลยี Shock Burst ใช้บนชิพ FIFO เพื่อจับเวลาข้อมูลที่อัตราข้อมูลต่ำและส่งต่อที่อัตราข้อมูลสูงมาก ด้วยเหตุนี้จะสามารถทำให้เกิดการลดทอนของกำลังสูงสุด เมื่อดำเนินการระบบย่อย nRF2401 ใน Shock Burst เราจะได้อัตราข้อมูลสูงสุด (1 Mbps) โดย band 2.4 GHz โดยไม่ต้องใช้ไมโครคอนโทรลเลอร์ความเร็วสูงและราคาสูง (MCU) สำหรับการดำเนินการทางข้อมูล

โดยการใส่การประมวลผลข้อมูลความเร็วสูงสัมพันธ์กับ RF Protocol บนชิพ nRF2401 มีประโยชน์ดังนี้

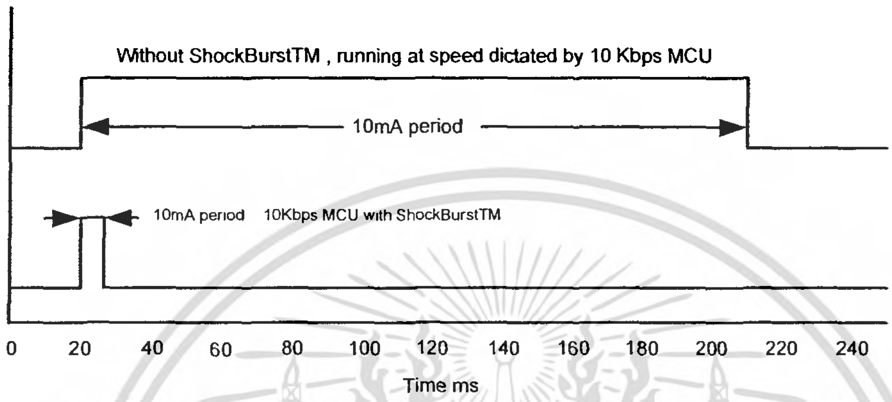
- ลดการใช้กระแส
- ระบบที่มีราคาถูก (ใช้งานได้ง่ายสำหรับ microcontroller ราคาไม่แพง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

82911

- ลดอัตราการเสี่ยงของการชนกันในอากาศ เนื่องจากการส่งสัญญาณในช่วงเวลาสั้นๆได้ดี

ระบบย่อย nRF 2401 สามารถโปรแกรมโดยใช้ สาย interface พื้นฐาน 3 เส้น ซึ่งอัตราของข้อมูลจะเลือกโดยความเร็ว CPU โดยยอมรับส่วนดิจิตอลของ application เพื่อจะ run ที่ความเร็วต่ำขณะที่อัตราข้อมูลบน RF Link มีค่าเพิ่มสูงสุด โหมด Shock Burst จะลดการใช้กระแสเฉลี่ยใน application ซึ่งเปรียบเทียบกับการใช้เฉลี่ยใน direct mode ดังรูปที่ 2.19



รูปที่ 2.19 การใช้กระแส RF โดยใช้และไม่ใช้เทคโนโลยี Shock Burst

การส่ง Shock Burst

CPU interface pins : CE, CLK1, DATA

1. เมื่อ CPU มีข้อมูลที่จะส่ง Set CE high คือ กระตุ้น nRF2401 ให้ประมวลผลบนบอร์ด

2. แอดเดรสของโหมดรับ (Rx) และ payload ข้อมูลจะจับเวลาเข้าสู่ช่วงเวลาระบบย่อย nRF2401 protocol หรือ CPU ตั้งความเร็วน้อยกว่า 1 Mbps

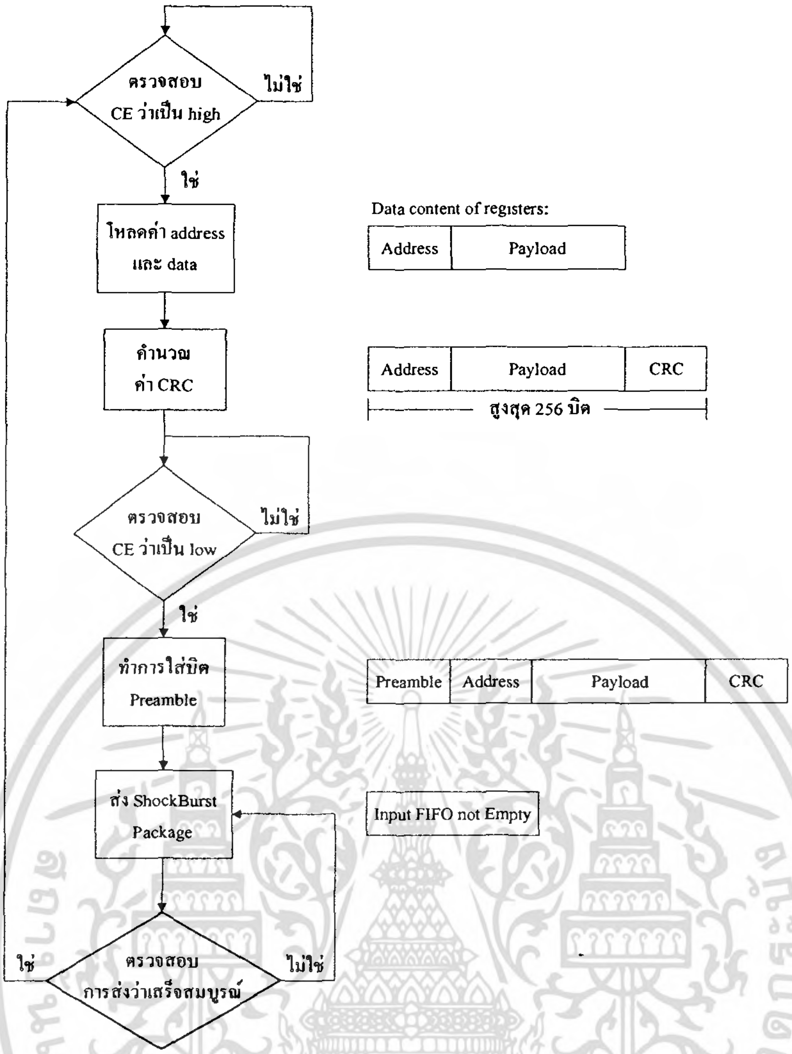
3. CPU ตั้ง CE low คือ กระตุ้นการส่ง Shock Burst

4. Shock Burst

- RF front end จะมีกำลังเพิ่มขึ้น
- RF Package จะสมบูรณ์
- ข้อมูลจะถูกส่งที่ความเร็วสูง (250 bps หรือ 1Mbps แล้วแต่ผู้ใช้)

ซึ่งอธิบายได้ดังรูปที่ 2.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 Flow Chart ShockBurst การส่งของระบบย่อย nRF2401

การรับ Shock Burst

CPU interface pins : CE, DR1, CLK1, DATA (1 ช่องรับ Rx)

1. แอดเดรสถูกต้องและขนาดของ payload ของ RF package ที่เข้ามาจะ set เมื่อระบบย่อย nRF 2401 เป็นโครงร่างของ Shock Burst Rx

2. เมื่อกระตุ้น Rx ; set CE high

3. หลังจากตั้ง 200us nRF 2401 จะตรวจสอบอากาศสำหรับการสื่อสารที่จะเข้ามา

4. เมื่อ package ที่ถูกต้องถูกเรียกมา (แอดเดรสถูกต้องและพบ CRC) nRF 2401 จะทำการย้ายบทนำแอดเดรสและบิต CRC

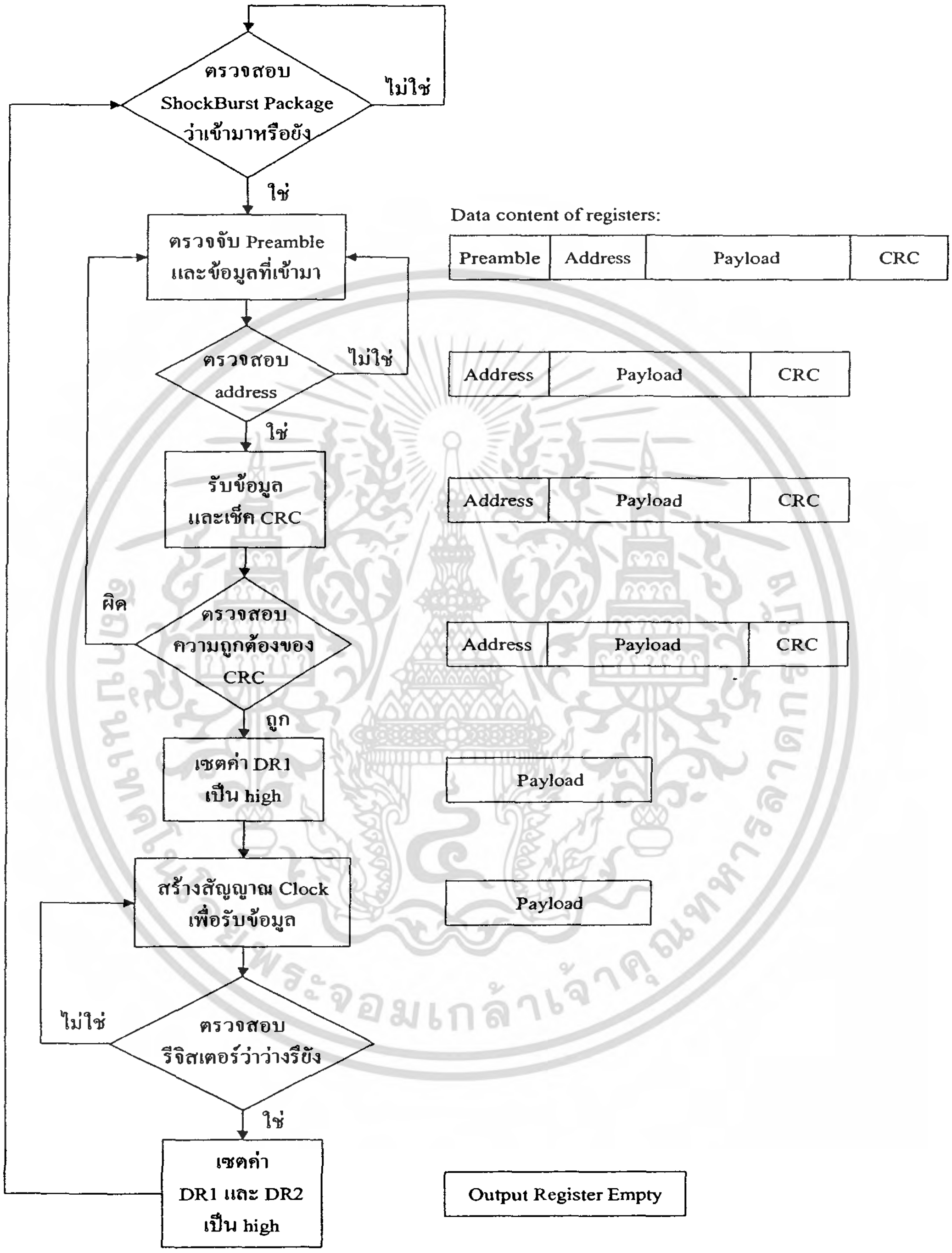
5. จากนั้นระบบย่อย nRF 2401 จะ interrupt CPU โดยการตั้ง DR1 high

6. CPU จะ set CE low เพื่อป้องกัน RF front end (โหมดที่กระแสต่ำ)

7. CPU จะจับเวลาที่หยุด payload data ที่อัตราที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เมื่อ payload data ทั้งหมดถูกเรียกมา nRF 2401 set DR1 low อีกครั้ง และพร้อมสำหรับข้อมูลที่จะเข้ามา ถ้า CE ยัง high ระหว่างที่ download ข้อมูล ถ้า CE set low การลำดับ start up ใหม่จะสามารถเริ่มได้ อธิบายโดยโฟลว์ชาร์ทดังรูปที่ 2.21



รูปที่ 2.21 Flow Cart ShockBurst การรับระบบย่อย nRF2401

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 Device configuration

Configuration ทั้งหมดของระบบย่อย nRF2401 จะสามารถทำผ่านทาง 3-wire interface ของการติดต่อสื่อสาร (CS, CLK1, DATA) กับ รีจิสเตอร์ เดี่ยว โดย configuration word สามารถมีได้ถึง 18 bytes ส่วน configuration bit (DATA) จะต้องจับเวลาโดย clocked (CLK1) โดย MSB มาก่อนขณะที่ CS = 1 ไม่เกิน 18 bytes อาจถูกดาวน์โหลด

Configuration สำหรับ การดำเนินการ ShockBurst

Configuration word ใน ShockBurst ทำให้ระบบย่อย nRF 2401 สามารถจัดการกับ RF protocol ได้ หาก protocol สมบูรณ์และโหลดไประบบย่อย nRF 2401 เพียง 1 byte :bit[7:0] จำเป็นที่จะต้อง update ตลอดช่วงที่ดำเนินการ

Configuration block โดยละเอียด สำหรับ Shock Burst ดังนี้

- ความกว้างของ payload: ระบุจำนวนของบิต payload ใน RF package ซึ่งทำให้ระบบย่อย nRF2401 สามารถแบ่งแยกระหว่างข้อมูล payload และ CRC bytes ในตัวรับ package ได้
- ความกว้าง address: set จำนวนบิตที่แอดเดรสใช้ใน RF package ซึ่งทำให้ระบบย่อย nRF2401 สามารถแบ่งแยกระหว่างแอดเดรสและข้อมูล package ได้
- แอดเดรส (Rx ช่อง 1 และ 2) : กำหนดแอดเดรสสำหรับข้อมูล
- CRC : สามารถทำให้ on-chip CRC generate และถอดรหัสได้

Configuration Block นี้ โดยการยกเว้น CRC จะละเอียดสำหรับ package ซึ่งระบบ nRF2401 ที่จะรับในโหมด Tx CPU ต้อง generate แอดเดรส และส่วน payload ให้เหมาะกับระบบย่อย nRF2401 เพื่อจะรับข้อมูลเมื่อใช้ระบบย่อย nRF2401 on-chip รูปแบบ CRC ต้องแน่ใจว่า CRC นั้นใช้ได้และใช้ที่ความยาวเดียวกันทั้งอุปกรณ์ Tx และ Rx

PRE-AMBLE	ADDRESS	PAYLOAD	CRC

รูปที่ 2.22 DATA package set-up

ช่วงบิต [119:16] ประกอบด้วย segments อย่างละเอียดของ configuration register ไปยัง protocol การทำงานของ ShockBurst บิตรวมของ 120 บิตต้อง shift เข้าเพื่อจะ switch ระหว่าง Rx และ Tx

- PLL_CTRL

ตารางที่ 2.7 PLL setting

PLL_CTRL		
D121	D120	PLL
0	0	Open TX/Closed RX
0	1	Open TX/Open RX
1	0	Closed TX/Closed RX
1	1	Closed TX/Closed RX

Bit 121-120 : PLL_CTRL ควบคุมการ set ของ PLL ใน Tx จะไม่เกิดการเบี่ยงเบน สำหรับโหมดดำเนินการปกติ 2 บิตนี้ต้องเป็น bit low

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DATAx_W

ตารางที่ 2.8 จำนวนของบิตใน payload

DATA2_W							
119	118	117	116	115	114	113	112

DATA1_W							
111	110	109	108	107	106	105	104

Bit 119-112 : DATA2_W ความยาว package ของ RF payload สำหรับสัญญาณช่อง 2

Bit 111-104 : DATA1_W ความยาว package ของ RF payload สำหรับสัญญาณช่อง 1

จำนวนบิตทั้งหมดใน Shock Burst RF package อาจไม่เกิน 256 ความยาวสูงสุดของส่วน payload หาได้จาก DATAx_W (bits) = 256 – ADDR_W – CRC

โดย ADD_W: ความยาวแอดเดรสของ Rx ที่ set ใน configuration word Bit[23:18]

CRC: Check sum 8 หรือ 16 บิต ที่ set ใน configuration word Bit[17]

PRE: Preamble 8 บิตซึ่งรวมอัตโนมัติ

แอดเดรสและCRC ที่สั้นขึ้น จะทำให้มีที่สำหรับ payload data มากขึ้น

- ADDRx

ตารางที่ 2.9 แอดเดรสของตัวรับ 1 และ 2

ADDR2											
103	102	101	.....	71	70	69	68	67	66	65	64

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

Bit 103-64 : ตัวรับ ADDR2 แอดเดรส channel2 มีได้ถึง 40 บิต

Bit 63-24 : ตัวรับ ADDR1 แอดเดรส channel1 มีได้ถึง 40 บิต

บิตใน ADDRx มากกว่าความกว้างที่ set ใน ADDR_W ซึ่งยืดยาวเกินไปและสามารถ set เป็น logic 0 ได้

- ADDR_W & CRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.10 จำนวนบิตที่ต้องคงไว้สำหรับ RX address + CRC setting

ADDR_W						CRC_L	CRC_EN
23	22	21	20	19	18	17	18

Bit 23-18 : ADDR_W :จำนวนบิตที่ต้องสงวนไว้สำหรับแอดเดรส Rx ใน ShockBurst จำนวนบิตแอดเดรสสูงสุดคือ 40 บิต (5 ไบต์) ค่าที่มากกว่า 40 บิต ADDR_W จะไม่สามารถใช้ได้

Bit 17 : CRC_L: ความยาว CRC ที่จะคำนวณใน ShockBurst

Logic 0 : 8 bit CRC

Logic 1 : 16 bit CRC

Bit 16 : CRC_EN: ทำให้ CRC on-chip Tx และ Rx ใช้ได้

Logic 0 : on-chip CRC generation / checking disable

Logic 1 : on-chip CRC generation / checking enable

บิต 8 CRC อาจเพิ่มจำนวน payload ใน ShockBurst ได้แต่จะลดความสมบูรณ์ของระบบลง

General RF configuration

General RF configuration ของ configuration word จะควบคุม RF และ พารามิเตอร์ที่เกี่ยวกับอุปกรณ์

ตารางที่ 2.11 RF operational setting

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

Bit 15 : Rx2_EN

Logic 0 : Channel 1 รับ

Logic 1 : Channel 2 รับ

ในตัวรับทั้ง 2 channel nRF 2401 รับทั้ง 2 channel แยกความถี่ของช่องสัญญาณทันที ความถี่ของตัวรับ channel 1 จะ set ในบิต [7-1] ของ configuration word ตัวรับ channel 2 จะเหนือตัวรับ channel 1 อยู่ 8 channel (8 MHz)

Bit 14 : Communication Mode

Logic 0 : ระบบย่อย nRF 2401 ดำเนินการใน Direct Mode

Logic 1 : ระบบย่อย nRF 2401 ดำเนินการใน Shock Burst Mode

Bit 13 : RF Data Rate:

Logic 0 : 250 kbps

Logic 1 : 1 Mbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ 250 kbps แทน 1 Mbps จะช่วยปรับปรุง sensitivity ของตัวรับ 10 dB 1Mbps ต้องใช้ crystal 16 MHz

Bit 12-10 : XO_F: เลือกความถี่ crystal ที่จะใช้

ตารางที่ 2.12 crystal frequency setting

XO FREQUENCY SELECTION			
D12	D11	D10	Crystal Frequency[MHz]
0	0	0	4
0	0	1	8
0	1	0	12
0	1	1	16
1	0	0	20

Bit 9-8 : RF_PWR set nRF 2401 RF กำลังของเอาท์พุทในโหมดการส่ง

ตารางที่ 2.13 RF output power setting

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

RF channel & direction

ตารางที่ 2.14 ช่องความถี่ และ การ set Rx / Tx

RF_CH#							
7	6	5	4	3	2	1	0

Bit 7-1 : RF_CH# ตั้งความถี่ช่องสัญญาณของ nRF 2401 ให้ดำเนินการความถี่ของช่องสัญญาณในการส่งได้โดย

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

RF_CH#: อาจตั้งไว้ระหว่าง 2400 MHz และ 2527 MHz ช่วงที่มากกว่า 83 สามารถใช้ได้เพียงแค่ในอาณาเขตที่แน่ใจได้

ความถี่ช่วงสัญญาณในช่องข้อมูล 2 หาได้โดย

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

Bit 0 : Active Mode

Logic 0 : transmit mode โหมดส่ง

Logic 1 : receive mode โหมดรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Configuration สำหรับการดำเนินการใน Direct Mode

สำหรับใน Direct mode จะดำเนินการเพียงแค่ใน 2 ไบต์แรกของ Configuration word ที่เกี่ยวข้อง

ตารางที่ 2.15 ตารางของ Configuration word

	Bit position	Number of bits	Name	Function
ShockBurst™ configuration	143:120	24	TEST	Reserved for resting
	119:112	8	DATA2_W	Length of data payload section RX channel 2
	111:104	8	DATA1_W	Length of data payload section RX channel 1
	103:64	40	ADDR2	Up to 5 byte address for RX channel 2
	63:24	40	ADDR1	Up to 5 byte address for RX channel 1
	23:18	6	ADDR_W	Number of address bits (both RX channels)
	17	1	CRC_L	8 or 16 bit CRC
	16	1	CRC_EN	Enable on-chip CRC generation/checking
General device configuration	15	1	RX2_EN	Enable two channel receive mode
	14	1	CM	Communication mode (Direct or ShockBurstTM)
	13	1	RFDR_SB	RF data rate (1Mbps requires 16MHz crystal)
	12:10	3	XO_F	Crystal frequency
	9:8	2	RF_PWR	RF output power
	7:1	7	RF_CH#	Frequency channel
	0	1	RXEN	RX or TX operation

Configuration word จะ shift MSB ก่อนที่ clock ขอบขาขึ้น configuration ใหม่จะเกิดที่ขอบขาลงของCS

configuration word ดังตารางที่ 2.7 อธิบายฟังก์ชันของ 144 บิต (บิต 143 = MSB) ซึ่งใช้ configure ระบบ nRF2401

Configuration อุปกรณ์ทั่วไป: bit [15:0]

Configuration Shock Burst: bit [119:0]

Test Configuration: bit [143:120]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

MSB	TEST																
D135	D134	D133	D133	D132	D131	D130	D129	D128	D127	D126	D125	D124	D123	D122	D121	D120	
Channel#2 Address RX(Up to 40 bit)															Close PLL in Tx		
0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	Default

DATA2_W								
D119	D118	D117	D116	D115	D114	D113	D112	
Data width channel#2 in # of bit excluding adress								
0	0	1	0	0	0	0	0	Default

DATA1_W								
D111	D110	D109	D108	D107	D106	D105	D104	
Data width channel#2 in # of bit excluding adress								
0	0	1	0	0	0	0	0	Default

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

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

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=16bits,0=0bits	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 Frequency			RF Power		Chanel selection							RXEN	
0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	Default

รูปที่ 2.23 Configuration data word

โดย MSB bit ควรจะถูกโหลดสู่ configuration register เป็นอันดับแรก Default Configuration word: h8E08.1C20.2000.0000.00E7.0000.0000.E721.0F04.

2.3.4 Data package Description

ชุดข้อมูลสำหรับการสื่อสารใน Shock Burst mode และ Direct mode จะแบ่งเป็น 4 ส่วน ดังรูปที่ 2.24 และมีรายละเอียดดังตารางที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRE-AMBLE	ADDRESS	PAYLOAD	CRC

รูปที่ 2.24 Data package Diagram

ตารางที่ 2.16 รายละเอียดของ Data package

1. PREMBLE	- ส่วน preamble ต้องมีใน Shock Burst mode และ Direct mode - Preamble มีความยาว 8 บิตและ ขึ้นกับบิตแรกของแอดเดรส PREAMBLE 1st ADDR_BIT 01010101 0 10101010 1 - Preamble จะใส่ไปในชุดข้อมูลโดยอัตโนมัติและดังนั้นจึงมีที่ว่างพิเศษสำหรับ payload ใน Direct mode MCU ต้องควบคุม preamble - ใน ShockBurst mode Rx preamble จะถูกย้ายจากรับข้อมูลเอาต์พุท ใน Direct mode preamble จะเห็นได้ชัดเจนสำหรับข้อมูลเอาต์พุท
2. ADDRESS	- ส่วนของแอดเดรสต้องมีใน ShockBurst โหมด และ Direct โหมด - ความยาว 80 – 40 บิต - แอดเดรสจะถูกย้ายโดยอัตโนมัติจากชุดรับใน ShockBurst mode ใน Direct mode MCU ต้องควบคุมแอดเดรส
3. PAYLOAD	- ข้อมูลถูกส่ง - ใน ShockBurst โหมด payload มีขนาดน้อยที่สุด 256 บิต (แอดเดรส 8-40 บิต +CRC 8หรือ16 บิต - ใน Direct mode ความยาวมากสุดสำหรับ 1 Mbps คือ 4000 บิต
4. CRC	- CRC จะเลือกได้ใน ShockBurst โหมด และไม่ใช้ใน Direct โหมด - ความยาว 8 หรือ 16 บิต - ShockBurst Rx CRC จะถูกย้ายจากรับข้อมูลเอาต์พุท

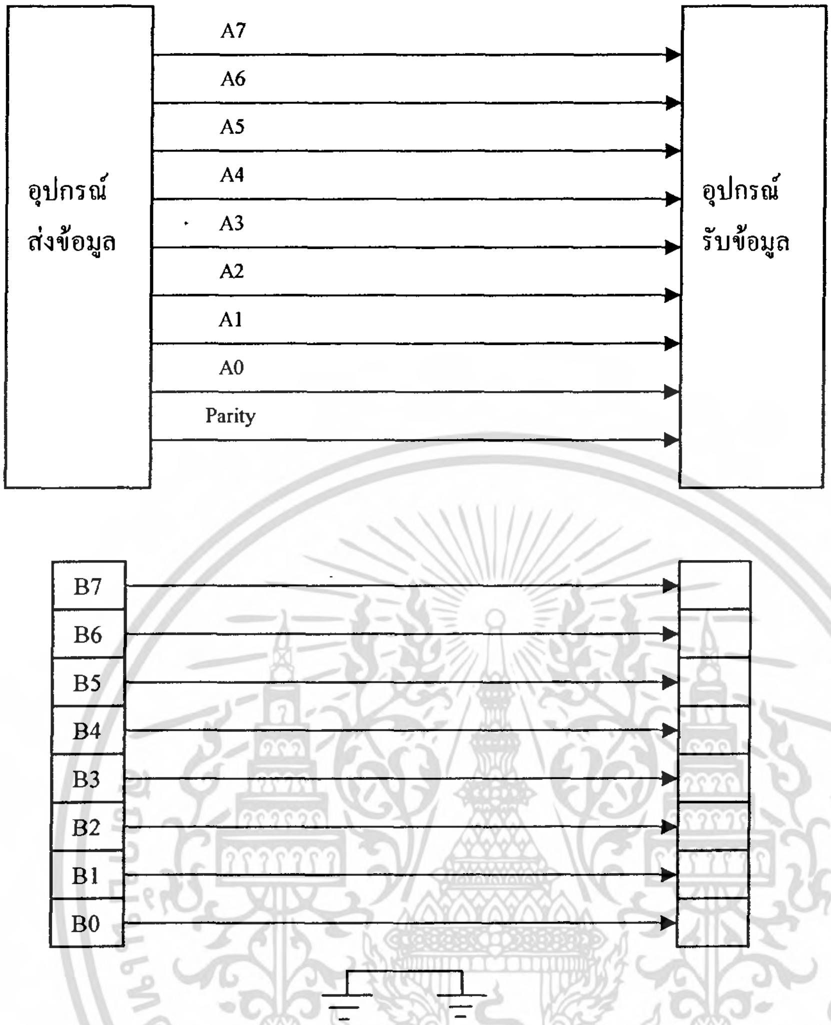
2.4 การสื่อสารผ่านพอร์ทอนุกรม RS-232

2.4.1 การสื่อสารข้อมูล

การสื่อสารข้อมูลแบบขนาน (Parallel Communication)

การสื่อสารข้อมูลแบบขนานคือการสื่อสารแบบที่ส่งข้อมูลพร้อมๆกัน n บิตผ่านสายสัญญาณ n เส้น สามารถแสดงรูปแบบการสื่อสารข้อมูลแบบขนานได้ดังรูป 2.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

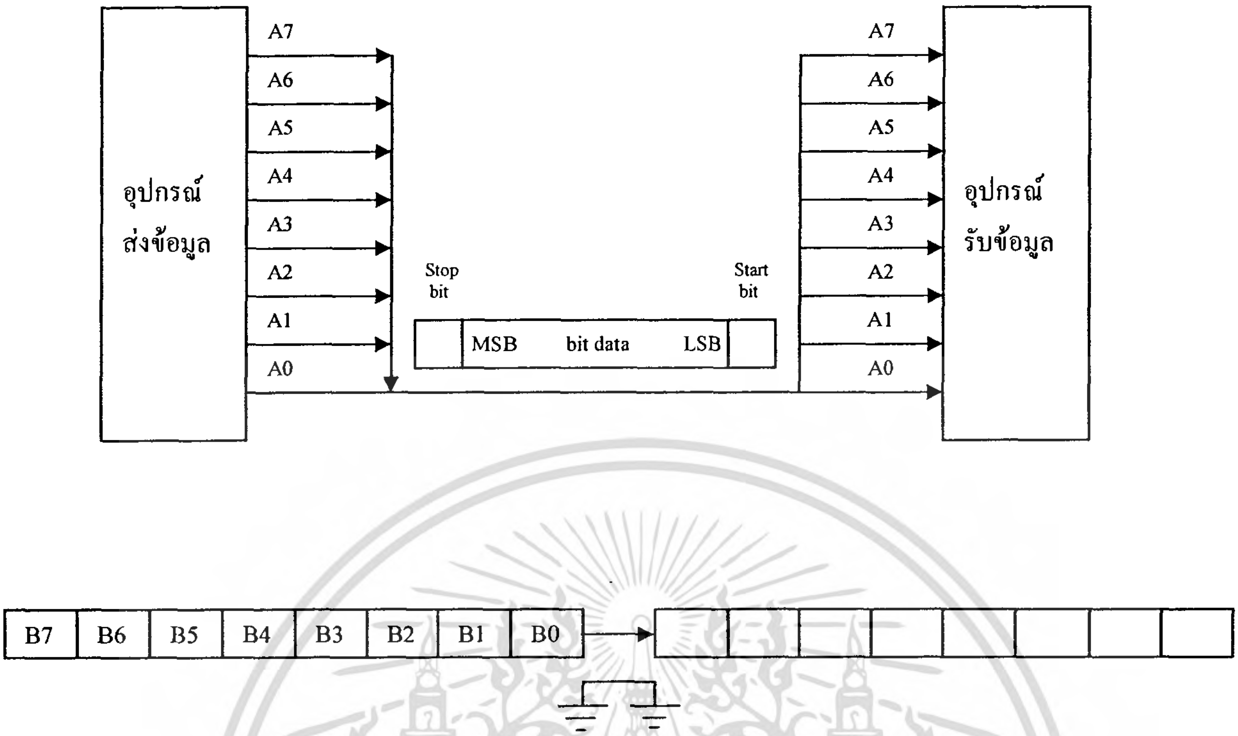


รูปที่ 2.25 แสดงบล็อกไดอะแกรมรูปแบบการสื่อสารแบบขนาน

- การสื่อสารข้อมูลแบบอนุกรม (Serial Communication)

การสื่อสารข้อมูลแบบอนุกรมคือการสื่อสารแบบที่ส่งข้อมูลทีละบิต ผ่านสายสัญญาณเส้นเดียวกันจนครบจำนวนข้อมูลที่ต้องการ โดยเฟรมของการสื่อสารข้อมูลแบบอนุกรมประกอบด้วย สตาร์ทบิต(start bit), สต็อปบิต (stop bit) และบิตข้อมูล (data bit) สามารถแสดงรูปแบบการสื่อสารข้อมูลแบบอนุกรมได้ดังรูปที่ 2.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

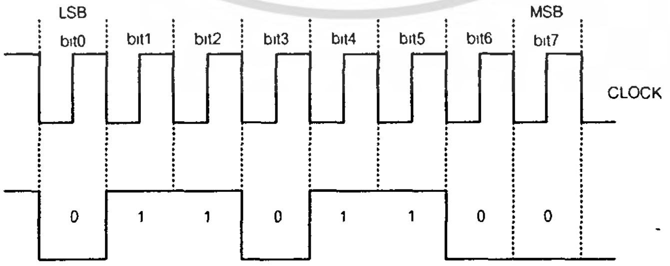


รูปที่ 2.26 แสดงบล็อกไดอะแกรมรูปแบบการสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบ คือ การสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส

1.การสื่อสารข้อมูลแบบซิงโครนัส

การสื่อสารข้อมูลแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสคือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา, ข้อมูล และกราวด์ ดังรูปที่ 2.32



รูปที่ 2.27 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

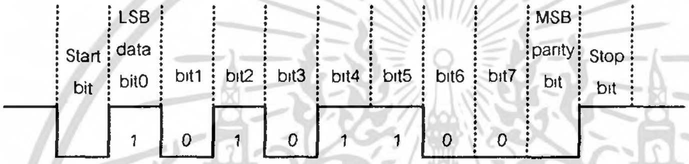
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับส่งข้อมูลไปในสายสัญญาณ โดยไม่จำเป็นต้องส่งสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายเทข้อมูล หรือ อัตราบอด (baud rate) ที่มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

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



รูปที่ 2.28 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

รูปที่ 2.28 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่ง ขาดาต้าจะมีสถานะลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่าสถานะว่าง (Idle stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขาดาต้ามีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไปโดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูลบิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและการส่งข้อมูลแบบอะซิงโครนัส เรียกว่า Universal Asynchronous Receiver / Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส คือ ค่าอัตราบอด ซึ่งก็คือค่าอัตราการเข้ารหัสที่ใช้ในการรับและส่งข้อมูล อัตราบอดมาตรฐานที่ใช้สำหรับพอร์ทอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าอัตราบอดได้สูงถึง 115,200 บิตต่อวินาที เนื่องจากอัตราบอดคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิตและบิตปิดท้าย 1 บิตความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้อัตราบอดในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

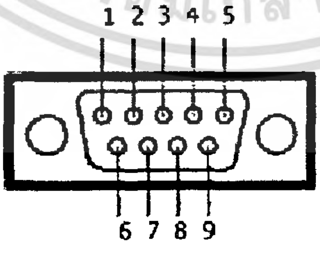
MAX-232 เป็น IC ในวงจรที่ทำหน้าที่เปลี่ยนแรงดันที่เข้ามาจาก Serial Port ซึ่งเป็นแรงดันตามมาตรฐานของ RS-232 โดยเปลี่ยนเป็นระดับแรงดัน TTL เพื่อให้ใช้ได้กับไมโครคอนโทรลเลอร์

ลักษณะของการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะส่งออกมาทีละบิตจากตัวส่งไปตัวรับข้อมูล ช่องสัญญาณในการส่งข้อมูลอาจใช้เพียง 1 หรือ 2 ช่องสัญญาณเท่านั้น ทำให้ค่าใช้จ่ายในการสื่อสารจะถูกกว่าแบบขนาน แต่อัตราการรับ-ส่งข้อมูลจะช้ากว่าแบบขนาน ในการส่งข้อมูลแบบอนุกรมข้อมูลที่ต้องการส่งจะอยู่ในลักษณะเป็นไบต์จะทยอยส่งทีละบิต และทางตัวรับจะต้องรับข้อมูลเข้ามาทีละบิตแล้วมารวมกันเป็นไบต์ซึ่งทางตัวรับ ต้องคอยตรวจสอบว่าบิตใดเป็นบิตเริ่มต้นหรือบิตสุดท้ายของข้อมูล การตรวจสอบนั้นจะขึ้นอยู่กับรูปแบบของรหัสของบิตข้อมูลที่ใช้ ซึ่งในการรับส่งข้อมูลแบบอนุกรมระหว่างไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้นจำเป็นจะต้องมีมาตรฐานในการรับส่งข้อมูล ซึ่งมาตรฐานที่นิยมมากที่สุดก็คือมาตรฐาน RS-232

2.4.2 มาตรฐาน RS-232

เพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันอย่างกว้างขวางที่สุดคือ RS-232 ซึ่งโดยปกติไมโครคอมพิวเตอร์จะมีพอร์ทที่เป็นแบบอนุกรมอยู่ในตัวแล้ว และจะทำหน้าที่รับส่งข้อมูลในแบบอนุกรม

ตามจุดประสงค์ของมาตรฐาน RS-232 นั้นเพื่อจะสามารถเชื่อมต่อกันระหว่างอุปกรณ์รับส่งปลายทาง (Data Terminal Equipment: DTE) เช่น พอร์ทของคอมพิวเตอร์หลักหรืออุปกรณ์ปลายทางกับอุปกรณ์สื่อสาร RS-232 เป็นข้อกำหนดของการอินเตอร์เฟสมาตรฐาน และสามารถใช้เพื่อจุดประสงค์อื่นต่างกันไป เช่นการสื่อสารแบบซิงโครนัส (synchronous communication) และรูปแบบการสื่อสารที่ต้องการสัญญาณนาฬิกา และสัญญาณกำหนดจังหวะเพิ่มเติมขึ้นมา ในความเป็นจริงแล้วเราสามารถทำให้มีการสนทนากันระหว่าง DTE และ DCE โดยการใช้สายสัญญาณเพียง 3 เส้นเท่านั้น คือ ใช้สาย Tx สาย Rx และสายกราวด์เท่านั้น มีลักษณะของคอนเน็คเตอร์ดังรูปที่ 2.29



รูปที่ 2.29 ลักษณะของคอนเน็คเตอร์แบบ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ระบบเครือข่าย Ethernet

ระบบเครือข่าย Ethernet เป็นระบบเครือข่ายท้องถิ่นหรือ LAN (Local Area Network) ประกอบด้วยส่วนที่เป็นฮาร์ดแวร์ที่ทำงานร่วมกันเพื่อการส่งถ่ายข้อมูลในระบบดิจิตอลระหว่าง คอมพิวเตอร์ระบบเครือข่าย Ethernet มีลักษณะพิเศษดังนี้

1. เป็นระบบเครือข่ายที่มีความเร็วในการส่งข้อมูลในรูปแบบดิจิตอลที่มีความเร็วตั้งแต่ 10 Mbps จนถึง 1,000 Mbps (1Gbps)

2. เป็นเครือข่ายที่มีขนาด Diameter ตั้งแต่ 205 เมตรจนถึง 4,000 เมตร

3. ใช้โปรโตคอลการทำงานที่เรียกว่า CSMA/CD (Carrier Sense Multiple Access with Collision Detect) ซึ่งเป็นมาตรฐานของ IEEE802.3 นอกจากนี้ก็ยังมีมาตรฐาน IEEE802.3ab สำหรับ Gigabit Ethernet ที่ใช้สายทองแดง

4. หนึ่งเครือข่าย Ethernet สามารถมีอุปกรณ์เชื่อมต่อ เช่น คอมพิวเตอร์ลูกข่าย อุปกรณ์ Repeater เป็นต้น ได้มากมายถึง 1,024 รายการหรือเรียกว่า Node

5. เป็นเครือข่ายที่สามารถใช้สายสัญญาณได้หลายแบบ เช่น สาย Coaxial ทั้งแบบหนาแบบบาง สาย Twisted Pair ทั้งแบบ Shield และ Unshield รวมทั้งสาย Optical Fiber แบบขนาดต่างๆ นอกจากนี้ยังสามารถใช้สื่อที่ใช้รับส่งข้อมูลแบบไร้สาย เช่น คลื่นวิทยุที่มีความถี่ Spread Spectrum รวมทั้งไมโครเวฟ (Microwave) ที่ใช้ความถี่ในช่วง 14 GHz และอินฟราเรด (infrared) เป็นต้น

6. เป็นระบบเครือข่ายที่มีการเชื่อมต่อในรูปแบบ Bus และ Star Topology

7. อุปกรณ์ราคาประหยัด

8. มีความน่าเชื่อถือสูง โดยเฉพาะหากใช้สื่อที่เป็นสาย Optical Fiber

9. มีเครื่องมือในรูปแบบของซอฟแวร์ที่ให้บริหารจัดการเครือข่ายมากมายที่ทำงานภายใต้ SNMP (Simple Network Management Protocal)

2.5.1 ส่วนประกอบหลักที่สำคัญของเครือข่าย Ethernet

ระบบเครือข่าย Ethernet มีส่วนประกอบหลักซึ่งเมื่อทำงานด้วยกันแล้วจะเป็นเครือข่ายที่มีประสิทธิภาพการทำงานสูงดังนี้

1. ตัวเฟรมเป็นชุดรูปแบบของบิตข้อมูลข่าวสารที่ใช้ส่งผ่านมาบนระบบ หากไม่มีเฟรมเราจะไม่สามารถสื่อสารข้อมูลบนเครือข่ายได้เด็ดขาด การรับส่งข้อมูลข่าวสารบนเครือข่าย Ethernet **จะต้องเป็นรูปแบบเฟรมมาตรฐาน 2 แบบ และเป็นแบบใดแบบหนึ่งเท่านั้น (การ์ด LAN เป็นผู้สร้างเฟรมนี้ขึ้นมา)**

2. **ชุดโปรโตคอลที่ใช้ในการควบคุมการแอกเซสเข้าไปเครือข่าย (Media Access Control Protocal) ซึ่งประกอบด้วยชุดของกฎกติกาที่อยู่ใน Ethernet Interface (เช่นการ์ด LAN เป็นต้น)ซึ่งเป็นกฎกติกามาตรฐานที่จะยอมให้คอมพิวเตอร์ต่างๆ สามารถเข้ามาในเครือข่าย และแบ่งใช้ทรัพยากรต่างๆ บนเครือข่ายได้อย่างมีประสิทธิภาพ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

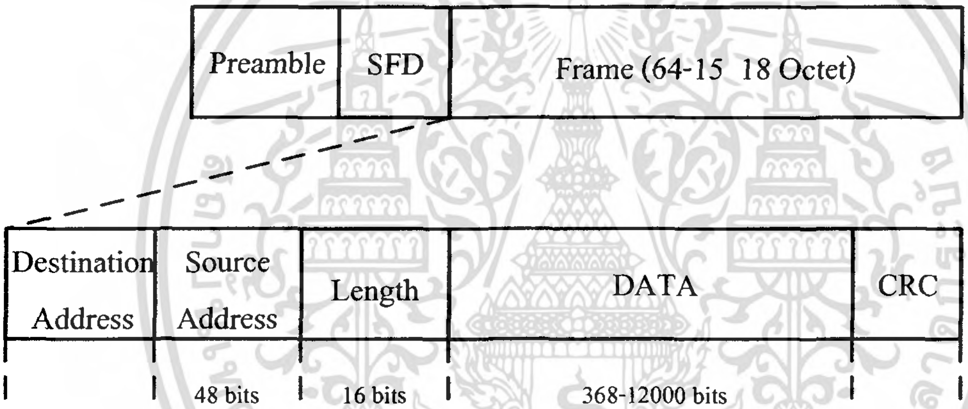
3. อุปกรณ์ที่ใช้รับส่งสัญญาณบนเครือข่าย (Signaling Components) ประกอบด้วยชุดของอุปกรณ์ที่ใช้เชื่อมต่อและส่งสัญญาณเพื่อการรับส่งข้อมูลในเครือข่าย

4. สื่อที่ใช้ในการรับส่งสัญญาณข้อมูลบนเครือข่าย (Physical Medium) ประกอบด้วยสายสัญญาณรวมทั้งอุปกรณ์ทางฮาร์ดแวร์อื่นๆ ที่จะช่วยในการนำพาข้อมูลข่าวสารต่างๆ ในรูปแบบดิจิตอลวิ่งไปมาบนเครือข่าย

2.5.2 เฟรมบนระบบ Ethernet

หัวใจสำคัญของระบบ Ethernet ได้แก่ เฟรมข้อมูลทางข่าวสารและอุปกรณ์ทางฮาร์ดแวร์ที่ใช้เชื่อมต่อสื่อสารบนเครือข่าย ซึ่งได้แก่ การ์ด Ethernet LAN สายสัญญาณและอุปกรณ์เสริมอื่นๆ ที่จะช่วยนำพาข้อมูลในรูปแบบของบิตทางดิจิตอล ที่เรียกว่าเฟรมวิ่งไปมาระหว่างคอมพิวเตอร์บนเครือข่าย

เฟรมข้อมูลสำหรับระบบ Ethernet ประกอบขึ้นด้วยกลุ่มของบิตที่เป็นข้อมูลและข่าวสารสำคัญแบ่งออกเป็นขนาดสัดส่วนที่แน่นอนที่เรียกว่าช่อง Field



รูปที่ 2.30 ลักษณะโครงสร้างของเฟรมข้อมูล

จากรูปที่2.30 ทั้งสองเฟรมจะมีความแตกต่างกันเล็กน้อย ทำให้เครือข่ายที่ใช้เฟรมแตกต่างกันนี้อาจไม่สามารถเข้ากันได้ หมายความว่าระบบเครือข่าย Ethernet ของท่านจะต้องเลือกใช้อุปกรณ์เครือข่ายที่คอยสนับสนุนเฟรมอย่างใดอย่างหนึ่งเท่านั้น แต่ก็เป็นเรื่องที่ดีที่ผู้ผลิตอุปกรณ์สนับสนุนเฟรมทั้งสองแบบในตัวเดียวกันดังรูปที่ 2.31

Preamble	**Destination MAC Address (6Byte)**	**Source MAC Address (6Byte)**	**Type (2Byte)**	**Data Field (1500 Byte Max)**	**Frame Check Sequence (4Byte)**

รูปที่ 2.31 ลักษณะของ Ethernet II Frame

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดเกี่ยวกับขนาดของ Data Frame

ขนาดของ Data Frame มีมาตรฐานดังต่อไปนี้

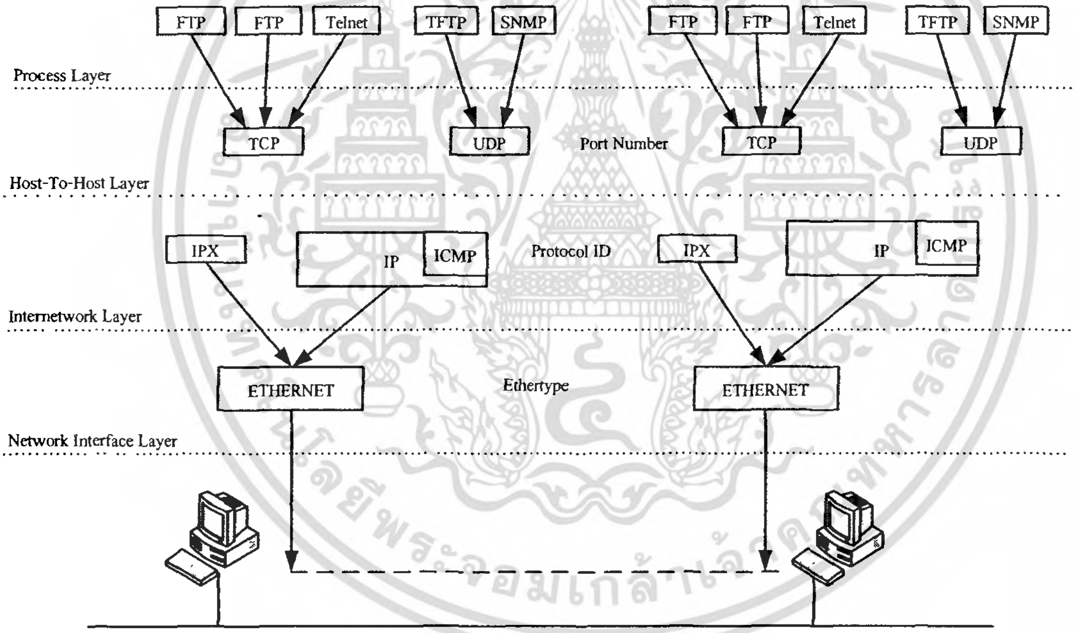
1. ขนาดเล็กที่สุด ต้องไม่น้อยกว่า 64 byte โดยมี 12 byte สำหรับแอดเดรส 2 byte สำหรับช่อง Length 46 byte สำหรับเก็บข้อมูล และ 4 byte สำหรับตรวจสอบความผิดพลาดข้อมูล หรือ Frame Check Sequence

2. ขนาดใหญ่ที่สุดต้องไม่เกิน 1,518 byte โดยแบ่งออกเป็น 12 byte สำหรับแอดเดรส 2 byte สำหรับ Length 1,500 byte สำหรับข้อมูล และ 4 byte สำหรับช่องตรวจสอบความผิดพลาดข้อมูล

3. เฟรมที่มีขนาดเล็กที่สุด 64 byte จะต้องใช้เวลาอยู่ที่ 51.2 ไมโครวินาที

2.5.3 โครงสร้างของสถาปัตยกรรมรูปแบบของ Protocol TCP/IP

สามารถการแบ่งออกเป็น 4 เลเยอร์ และในแต่ละเลเยอร์ได้มีการกำหนดหน้าที่การทำงานไว้ดังรูปที่ 2.32



รูปที่ 2.32 แสดงการรับส่งข้อมูลผ่านโปรโตคอล TCP/IP

Process Layer

เป็นลำดับชั้นการทำงานของโปรโตคอล TCP/IP ตามมาตรฐาน DoD-Reference Model ซึ่งเมื่อนำมาเปรียบเทียบกับมาตรฐาน OSI-Reference Model นั้น ในชั้นบนสุดที่เรียกว่า Process Layer ของ DoD Model จะทำงาน 2 หน้าที่เทียบได้กับ Application Layer และPresentation Layer ของ OSI-Reference Model ในชั้นนี้จะรองรับการทำงานของ Application ต่างๆ อย่างเช่น เมื่อเครื่อง Client ทั่วไป ขอใช้บริการเพื่อจะติดต่อขอ Download File ผ่านทาง Internet โดยอาจจะเรียกใช้โปรแกรม FTP Client ทั่วไป อย่างเช่นโปรแกรม WS_ftp เพื่อติดต่อกับโปรเซส FTP ที่กำลังให้บริการอยู่ที่เครื่อง Server จากนั้นตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรเซส FTP ก็จะเรียกใช้โปรโตคอล FTP (File Transfer Protocol) เพื่อทำการโอนถ่ายไฟล์นี้ไปให้เครื่อง Client เป็นต้น หรือถ้าผู้ใช้ต้องการเรียกใช้งานคอมพิวเตอร์จากเครื่องที่อยู่ห่างไกลออกไป ด้วยการใช้โปรแกรม Telnet ที่เครื่อง Server ให้บริการตัวโปรเซส Telnet ที่ทำงานอยู่ก็จะเรียกใช้โปรโตคอล Telnet เพื่อติดต่อกัน

การทำงานของ Application ต่างๆ จะอยู่ที่ Process Layer นี้ และมีการติดต่อกันตามแต่ละโปรโตคอลเฉพาะแล้วแต่ Application ที่ใช้งาน จากการที่ Process Layer ของ TCP/IP รองรับให้โปรโตคอลอื่นทำงานได้หลายโปรเซส และหลายโปรโตคอลได้พร้อมกันนั้น ทำให้ผู้ใช้สามารถเปิดโปรแกรมใช้งานได้หลายอย่างพร้อมกัน เช่น เปิดโปรแกรม Internet Explorer เพื่อเรียกดูเว็บเพจพร้อมกับใช้งานโปรแกรม Outlook Explorer เพื่อรับส่ง E-mail ไปพร้อมกันได้โดยไม่ต้องรอให้การทำงานอย่างใดอย่างหนึ่งเสร็จไปก่อน หรือในปัจจุบันมีการพัฒนาโปรแกรม Web Browser โอนถ่ายไฟล์ข้อมูลที่ใช้โปรโตคอล FTP ได้โดยไม่ต้องไปหาโปรแกรมอื่นมาใช้เพิ่มเติมอีก

โปรโตคอลหลักที่ทำงานและให้บริการในชั้น Process Layer

1. FTP (File Transfer Protocol)

FTP ที่ใช้ในการรับ-ส่งแฟ้มข้อมูลระหว่างเครื่องลูกข่ายและเครื่อง Server จะต้องมีโปรแกรมให้บริการ FTP (FTP Server) ติดตั้งและทำงานอยู่ เพื่อให้ลูกข่ายที่รันโปรแกรม FTP Client สามารถเข้ามาขอใช้บริการได้

2. Telnet

Telnet เป็นบริการที่ให้ลูกข่ายสามารถเข้าไปใช้เครื่อง Server โดยการจำลองตัวเองให้ทำงานในเทอร์มินัล ผู้ใช้งานจะต้องใส่รหัสผู้ใช้งานและรหัสผ่านเพื่อแจ้งการเข้าใช้เครื่อง เมื่อเข้าไปใช้ได้แล้ว การทำงานจะเหมือนกับการเข้าไปทำงานที่หน้าจอของเครื่อง Server การทำงานแบบนี้เครื่อง Serverจะติดตั้งโปรแกรมการให้บริการ Telnet ซึ่งโดยปกติในระบบปฏิบัติการยูนิกส์ จะมีบริการนี้ติดตั้งไว้แล้วเป็นมาตรฐาน มีศัพท์เรียกโปรแกรมให้บริการบนเครื่องยูนิกส์ว่า daemon เช่น FTP daemon , Telnet daemon เป็นต้น

3. HTTP (Hypertext Transfer Protocol)

HTTP ใช้ในการติดตั้งรับส่งข้อมูลชนิดไฮเปอร์เท็กซ์ (Hypertext) ระหว่างเครื่องลูกข่ายกับ WWW Server (World Wide Web) โดยที่เอกสารนี้จะอยู่ในรูปแบบที่เขียนในภาษา HTML (Hypertext Markup Language) เอกสารแต่ละชิ้นจะสามารถเชื่อมโยงไปยังเอกสารอื่นได้ ซึ่งเอกสารที่ถูกเชื่อมโยงนี้อาจจะอยู่บนเครื่องคอมพิวเตอร์เดียวกัน หรือต่างเครื่องกันก็ได้

4. SMTP (Sample Mail Transfer Protocol)

SMTP เป็นการให้บริการอินเตอร์เน็ตเพื่อรับส่งจดหมายอิเล็กทรอนิคส์ (E-mail) โดยที่ SMTP จะมีตู้ไปรษณีย์เพื่อทำหน้าที่รับจดหมายจากผู้อื่นที่ต้องการส่งให้ และเก็บจดหมายของผู้ใช้ที่ต้องการส่งไปยังผู้ใช้อื่น เมื่อถึงกำหนด เวลาที่ตั้งไว้โปรแกรมจะทำการส่งจดหมายออก และรับจดหมายเข้า ผู้ใช้ก็สามารถจะเปิดอ่านได้เมื่อต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้โปรโตคอล ที่อยู่เบื้องหลัง ซึ่งทำงานโดยที่ผู้ใช้ไม่สามารถมองเห็นได้จากโปรแกรม หรือไม่ได้มีการใช้งานโดยตรง เช่น

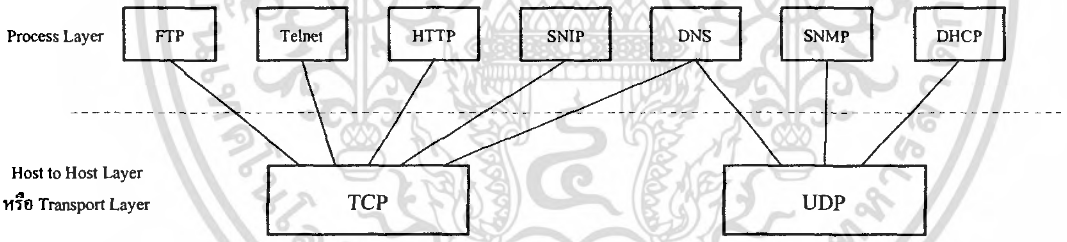
-Protocol DNS (Domain Name System) ที่ทำหน้าที่แปลงข้อมูลชื่อ Domain Name หรือ ชื่อเว็บไซท์ ทั้งหลายให้เป็นหมายเลข IP Address

-Protocol SNMP (Simple Network Management Protocol) ใช้ในการควบคุม และตรวจสอบอุปกรณ์ที่อยู่ในเครือข่าย

-Protocol DHCP (Dynamic Configuration Protocol) ทำหน้าที่แจกจ่ายข้อมูล พารามิเตอร์ ของโครงข่ายให้กับเครื่องลูกข่ายที่เชื่อมต่ออยู่

Host-to-Host Layer

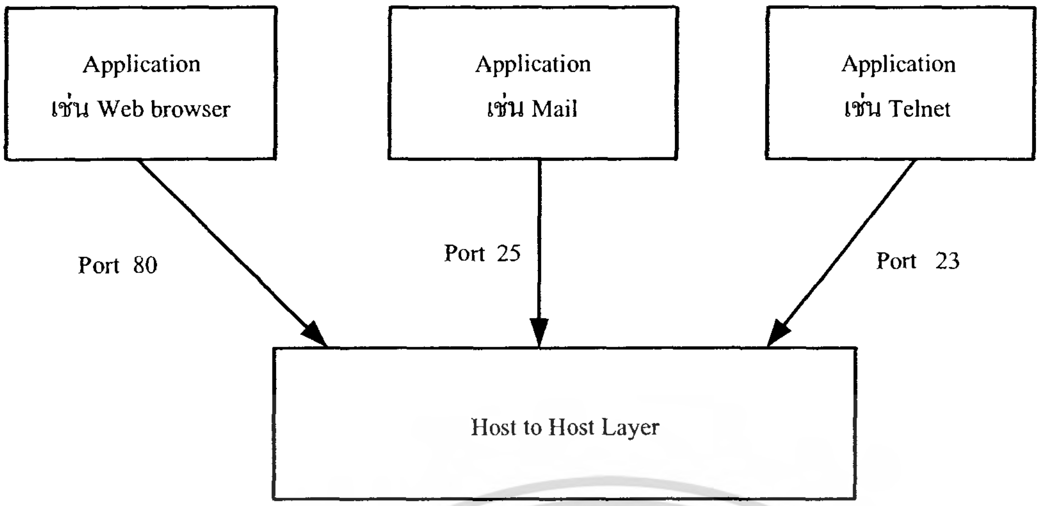
การทำงานที่ชั้นของ Host - to - Host Layer จะมีบทบาทในการจัดการต่อจากชั้นของ Process Layer บางครั้งเรามักเรียกชั้น Host - To - Host Layer ว่าเป็น Transport Layer ซึ่งไม่ใช่ชั้นของ Transport Layer ในมาตรฐาน OSI - Reference Model การทำงานของ Host - To - Host Layer นี้จะมีการสร้าง Connection หรือการเชื่อมต่อกันระหว่างแอพพลิเคชันกับ Host - To - Host Layer โดยจุดที่เชื่อมกันเพื่อรับส่งข้อมูลนี้เรียกว่า Port หรือ Socket (คำว่า Port ในที่นี้ ไม่ได้หมายถึง Port ทาง Hardware) และในแต่ละแอพพลิเคชัน ก็จะสร้างการเชื่อมต่อ ผ่าน Port ได้พร้อมกันหลายแอพพลิเคชัน ซึ่งการใช้งาน Port ของแต่ละแอพพลิเคชันที่อยู่ในชั้น Process Layer จะแตกต่างกันตามหมายเลขที่กำหนดไว้ และแต่ละโปรโตคอลจะมีการใช้งาน Port หมายเลขไม่ซ้ำกัน ดังรูปที่ 2.33



รูปที่ 2.33 แสดงการใช้งาน port ของแต่ละโพรโตคอล

เมื่อแอพพลิเคชันทำงานผ่านโปรโตคอลในชั้น Process layer จะมีการส่งผ่านข้อมูลไปยัง Host-to-Host layer ที่ชั้นนี้จะมีการเชื่อมต่อผ่าน port ที่กำหนดดังรูปที่ 2.33 ทำให้การรับส่งข้อมูลในแต่ละโปรโตคอลทำได้ถูกต้อง ถึงแม้ว่าในเครื่องเซิร์ฟเวอร์ที่ให้บริการ จะมีการทำงานอยู่หลายโปรเซสที่แตกต่างกันก็ตาม หรือมีผู้ใช้บริการเข้ามาใช้งานพร้อมกันจำนวนมากและหลายแอพพลิเคชันในเวลาเดียวกัน ในชั้น Host-to-Host หรือ Transport layer ของ TCP/IP นี้ จะมีโปรโตคอลทำงานอยู่ 2 โปรโตคอลที่แตกต่างกัน คือ โปรโตคอล TCP และโปรโตคอล UDP (User Datagram Protocol) ในการส่งผ่านข้อมูลลงไปที่ชั้นถัดไป จะเห็นว่าโปรโตคอล TCP และ UDP จะถูกผนึกเข้าไปในโปรโตคอล IP อีกทีหนึ่งและส่งต่อไปยังเครือข่ายอินเตอร์เน็ตต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 แสดงการส่งข้อมูลจาก Application ไปยัง Host – to – Host Layer

ตัวโปรโตคอล TCP และโปรโตคอล UDP จะมีแอพพลิเคชันเฉพาะเพื่อเรียกใช้งานแยกกันคือ แอพพลิเคชันที่ใช้โปรโตคอล FTP, Telnet, HTTP และ SMTP จะมีการส่งผ่านข้อมูลโดยเรียกใช้โปรโตคอล TCP ส่วนแอพพลิเคชัน SNMP และ DHCP จะส่งผ่านข้อมูลโดยเรียกใช้โปรโตคอล UDP และสำหรับโปรโตคอล DNS นั้นจะสามารถเรียกใช้งานได้ทั้ง TCP และ UDP ดังรูป ซึ่งเหตุผลที่มีการเรียกใช้โปรโตคอล TCP และ UDP แตกต่างกัน เนื่องจากวิธีการทำงานของทั้งสองโปรโตคอลต่างกัน

โปรโตคอล TCP

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ stream oriented protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อย ๆ ก่อน แล้วจึงจะส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางจะทำหน้าที่จัดเรียงส่วนของข้อมูล datagram ใหม่ให้ต่อเนื่องกันและประกอบกับเป็นข้อมูลทั้งหมดได้ ซึ่งจะแยกข้อมูลที่ไม่ถูกต้องออก ดังนั้น แอพพลิเคชันหรือโปรเซสใดที่อาศัยการส่งผ่านข้อมูลด้วยโปรโตคอล TCP จะต้องใช้หน่วยความจำและขนาดของช่องสัญญาณ (bandwidth) มากกว่า UDP การติดต่อระหว่างกันจะต้องเป็นแบบ connection-oriented คือ ต้องมีการสร้างการติดต่อกันเป็น session ทั้ง 2 ด้านเสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมกัน (full duplex) เหมือนกับการใช้โทรศัพท์ติดต่อกัน เมื่อผู้ติดต่อต้องการเรียกให้ฝ่ายตรงข้ามรับสายแล้ว จึงเริ่มการสนทนา เช่น พูดคำว่า “สวัสดี” หรือ “ฮัลโหล” กันก่อนเพื่อให้แน่ใจว่าอีกฝ่ายตรงข้ามพร้อมที่จะติดต่อด้วย จากนั้นจึงเริ่มที่จะติดต่อกัน และเมื่อต้องการจะเลิกการติดต่อก็จะมีการพูดคำว่า “สวัสดี” ให้ฝ่ายตรงข้ามทราบว่าจะเลิกการติดต่อและวางสายไป ซึ่งในระหว่างการติดต่อกันนั้นแม้ว่าฝ่ายใดฝ่ายหนึ่งหรือทั้งสองฝ่ายจะเงียบไป คือ ไม่พูดอะไรกันเป็นเวลานาน ๆ แต่การเชื่อมโยงระหว่างทั้งสองด้านยังคงมีอยู่ไม่ขาดไปจนกว่าฝ่ายใดฝ่ายหนึ่งจะวางสาย เช่นเดียวกันกับการติดต่อกันด้วยกลไกโปรโตคอล TCP เมื่อแอพพลิเคชันต้องการส่งผ่านข้อมูลจะใช้โปรโตคอลที่เหมาะสมในชั้น Process Layer ติดต่อไปและมีการสร้างช่องส่งข้อมูลผ่าน port ที่กำหนดเพื่อส่งผ่านข้อมูลไปยังโปรโตคอล TCP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระหว่างการรับส่งข้อมูลนี้ โปรโตคอล TCP จะเพิ่มขบวนการสอบทานข้อมูลเพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณสอบทานข้อมูล (acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น

ความน่าเชื่อถือของการส่งผ่านข้อมูลโดยโปรโตคอล TCP จะมีมากกว่า แต่ก็ต้องอาศัยทรัพยากรของระบบมากกว่าในการทำงานเช่นกัน

โปรโตคอล UDP

ใน Host-to-Host layer นอกจากจะมีโปรโตคอล TCP ทำงานแล้ว ก็ยังมีโปรโตคอล UDP (User Datagram Protocol) ที่มีคุณสมบัติแตกต่างกันอยู่ด้วย ในการรับส่งข้อมูลผ่านโปรโตคอล UDP จะเป็นแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (connectionless) ระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการรับส่งข้อมูล เนื่องจากโปรโตคอล UDP ไม่มีสัญญาณสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้แอพพลิเคชันหรือโปรเซสใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการตรวจสอบข้อมูลขึ้นมาเอง

ตัวอย่างขั้นตอนกลไกการทำงานโดยใช้โพรโตคอล UDP มีดังต่อไปนี้

1. ในชั้นของ Process Layer เมื่อโปรแกรมควบคุมอุปกรณ์เครือข่าย เช่น โปรแกรม Network Management ต้องการส่งข้อมูลไปยังอุปกรณ์ที่ต้องการ แอพพลิเคชันนั้นจะติดต่อผ่านโพรโตคอล SNMP ในชั้น Process Layer

2. โพรโตคอล SNMP จะติดต่อกับโพรโตคอล UDP ในชั้นถัดไปเพื่อขอติดต่อผ่าน port ที่กำหนด

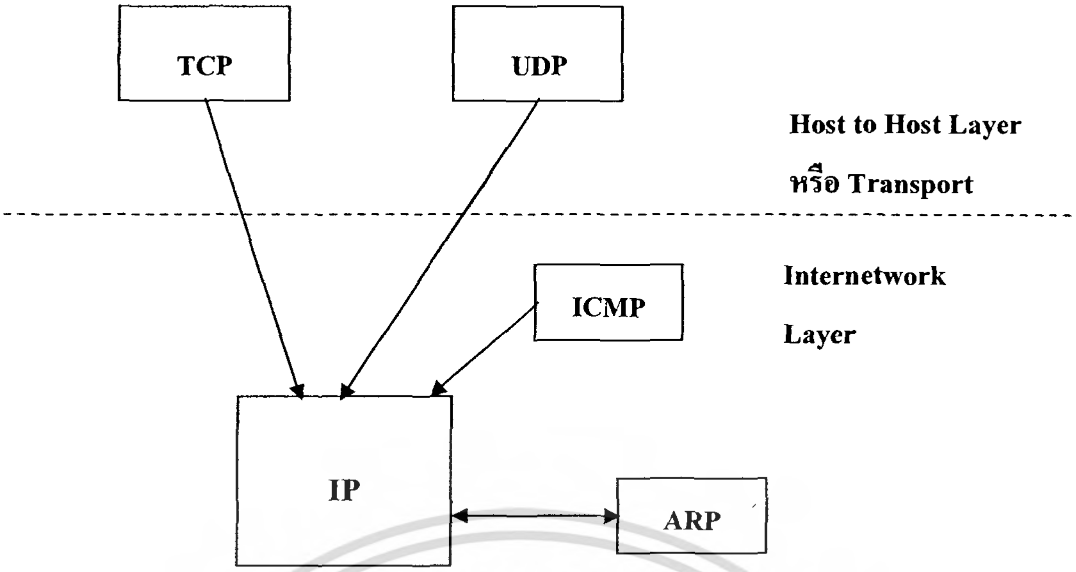
3. โพรโตคอล SNMP เตรียมข้อมูลที่จะส่ง รวมทั้งที่อยู่ปลายทาง

4. โพรโตคอล SNMP ส่งผ่านข้อมูลให้โพรโตคอล UDP ที่อยู่ในชั้น Host–to–Host Layer

5. โพรโตคอล UDP ทำหน้าที่ผนึกข้อมูลหรือ datagram นั้น ให้กับโพรโตคอล IP ในชั้นถัดลงไปเพื่อส่งข้อมูลออกจากเครื่อง

ซึ่งจะเห็นว่ามีกลไกที่ต่างจากการส่งข้อมูลด้วยโพรโตคอล TCP ซึ่งจะต้องมีการติดต่อกันก่อน และทั้งสองฝ่ายรับทราบการรับส่งข้อมูลของช่องการส่งข้อมูลนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 แสดงกลไกการส่งข้อมูลด้วยโปรโตคอล UDP

จากรูปที่ 2.35 จะเห็นว่าโปรโตคอลชั้นบนขึ้นไป ที่ใช้การส่งผ่านข้อมูลโดยโปรโตคอล UDP เช่น โปรโตคอล SNMP (ใช้ควบคุมและจัดการอุปกรณ์ในเครือข่าย) หรือโปรโตคอล DHCP (ใช้ส่งข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายได้ใช้งาน) การส่งข้อมูลเหล่านั้นไม่ต้องรับทราบหรือตรวจสอบว่าข้อมูลไปถึงปลายทางถูกต้องหรือไม่ แต่กลไกการตรวจสอบข้อมูลที่มีการรับส่ง จะไปทำให้ขั้นตอนของโปรโตคอลชั้นที่สูงกว่าแทน

UDP Header

มีขนาด 8 ไบต์ โดยประกอบด้วย 4 ส่วนดังรูปที่ 2.36

Source Port	Destination Port	Length	Checksum
2 ไบต์	2 ไบต์	2 ไบต์	2 ไบต์

รูปที่ 2.36 แสดงโครงสร้างของ UDP Header

- **Source Port** มี 2 ไบต์ใช้ระบุเป็น Source Application Layer Protocol ทำการส่ง UDP Message โดย Source Port เป็น port ที่ใช้ในการเลือก เมื่อใดที่ไม่ได้ใช้มัน มันจะตั้งค่าเป็น 0x00-00 IP multicast traffic เปรียบเสมือน videocasts ใช้ส่ง UDP สามารถใช้ค่า 0x00-00 เพราะจะไม่ตอบรับ video traffic เป็นเพียงการสมมุติ Application Layer ใช้ Source Port ในการนำ UDP Message เข้ามา Destination Port สำหรับการตอบรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Destination Port** มี 2 ไบต์ใช้ระบุเป็น Destination Application Layer Protocol การรวมของ Destination IP Address ของ IP Header และ Destination Port ของ UDP Header จะไม่เหมือนใครสำหรับกระบวนการที่จะส่งข้อมูล

- **Length** มี 2 ไบต์ที่ใช้ในการแสดงความยาวใน UDP Message มีความยาวน้อยที่สุด 8 ไบต์ (ขนาดของ UDP Header) และมากที่สุด 65,515 ไบต์ (ค่าสูงสุด IP Datagram 65,535 ไบต์ น้อยกว่าค่าน้อยที่สุด IP Header 20 ไบต์) ความยาวมากที่สุดที่แท้จริงถูกจำกัดโดย MTU ซึ่งจะทำการเชื่อมโยงโดย UDP Message เป็นตัวส่ง ความยาว UDP สามารถคำนวณได้จากความยาวทั้งหมดและความยาวของ IPHeader field ใน IP Header

- **Checksum** มี 2 ไบต์ โดยจะทำการตรวจระดับของบิตอย่างสมบูรณ์สำหรับ UDP Message โดยที่ UDP Checksum คำนวณโดยใช้วิธีเดียวกันกับ IP Header Checksum

ตารางที่ 2.17 UDP Checksum คำนวณโดยใช้วิธีเดียวกันกับ IP Header Checksum

ตำแหน่ง	**ชื่อ**	**อธิบาย**
บิต 0-15	Source port number	หมายเลขพอร์ทต้นทางที่ส่งดาต้าแกรมนี้ มีความยาว 16 บิต
บิต 16-31	destination port number	หมายเลขพอร์ทปลายทางที่จะเป็นผู้รับดาต้าแกรม มีความยาว 16 บิตเช่นกัน
บิต 32-47	UDP length	ความยาวของดาต้าแกรม ทั้งส่วน Header และ data นั่นหมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
บิต 48-63	Checksum	เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

UDP Port

UDP Port จะแสดงที่ตั้งหรือแถวของ message ที่ชัดเจนสำหรับการส่ง message ถึง Application Layer protocol โดยใช้ UDP services รวมถึงในแต่ละตัวของ UDP message เป็น Source Port และ Destination Port ซึ่ง Internet Assigned Number Authority (IANA) จะเป็นตัวกำหนดหมายเลข Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.18 UDP Port Numbers

Port Numbers	Application Layer Protocol
53	Domain Name System (DNS)
67	BOOTP client (Dynamic Host Configuration Protocol [DHCP])
68	BOOTP server (DHCP)
69	Trivial File Transfer Protocol (TFTP)
137	NetBIOS Name Service
138	NetBIOS Datagram Service
161	Simple Network Management Protocol (SNMP)
520	Routing Information Protocol (RIP)
445	Direct hosting of server Message Block (SMB) datagram over TCP/IP
1812 , 1813	Remote Authentication Dial-In User Service (RADIUS)

UDP Checksum

Checksum เป็น เลข 16 บิตถูกคำนวนด้วยวิธี one's complement โดยนำ Pseudo Header และข้อมูลทั้งหมดใน UDP Datagram มาคำนวณ Pseudo Header เป็นข้อมูลที่อยู่ในส่วนของ IP Header ประกอบด้วยฟิลด์ source IP address destination IP address , zero , protocol , UDP length ดังแสดงในรูปที่ 2.37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16-bit Source IP address		
16-bit Destination IP address		
zero	**8-bit protocol (17 for UDP)**	**16-bit length**

รูปที่ 2.37 Pseudo Header

หากค่า Checksum ที่คำนวณออกมาเป็น 0 ค่า checksum จะถูกเซ็ตเป็น 1 ทั้งหมดแทน (มีค่าเท่ากันในระบบ 1's complement) ทั้งนี้เพราะในบางแอพพลิเคชั่นที่ไม่ต้องการตรวจสอบค่า checksum ในระดับ UDP จะเซ็ตค่านี้เป็น 0 (disable checksum)

Internetwork Layer

ในระดับล่างต่อมาในชั้น Internetwork Layer มีหน้าที่ส่งผ่านข้อมูลในระหว่างเครือข่ายโดยมีโปรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใด ๆ บนอินเตอร์เน็ต คือ โปรโตคอล IP (Internet Protocol) นอกจากนี้ในชั้น Internetwork Layer ยังมีโปรโตคอลที่ทำงานอยู่ด้วยอีก 2 ชนิด คือ โปรโตคอล Internet Control Message Protocol (ICMP) และโปรโตคอล Address Resolution Protocol (ARP)

โปรโตคอล IP (Internet Protocol)

โปรโตคอล IP นี้ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host Layer เพื่อส่งเข้าไปยังเครือข่ายใด ๆ ได้อย่างถูกต้อง แม้ว่าจะมีเครือข่ายเชื่อมต่ออยู่ในอินเตอร์เน็ตเป็นล้าน ๆ เครือข่ายก็ตาม เนื่องจากโปรโตคอล IP มีข้อมูลตำแหน่ง IP ปลายทางที่จะส่งข้อมูลไปให้ โดยทำงานร่วมกับอุปกรณ์ Router เพื่อส่งข้อมูลข้ามเครือข่ายออกไปได้ ตัวโปรโตคอล IP จะทำงานแบบ Packet Switching คือ มีการส่งผ่านข้อมูลสวิทช์ (Switch) ไปยังปลายทาง โดยข้อมูลจะเดินทางไปยังเครือข่ายต่าง ๆ ผ่านสวิทช์นี้ไปเรื่อย ๆ จนกว่าจะถึงปลายทาง ตัววงจรผ่านหรือสวิทช์นี้เป็น Gateway หรือ Router ในระบบเครือข่ายก็ได้ ซึ่งในข้อมูลของโปรโตคอล IP จะมีข้อมูลของหมายเลข IP ปลายทางที่จะส่งข้อมูลไปให้ และเมื่อถึงเครือข่ายปลายทางแล้ว จะมีกลไกแปลงหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์ประจำเครื่องที่ถูกต้องอีกทีหนึ่ง ด้วยโปรโตคอล ARP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลไกของโปรโตคอล IP

ในการส่งผ่านข้อมูลหรือ IP datagram ไปยังเครือข่ายอินเตอร์เน็ตนั้น โปรโตคอล IP จะทำหน้าที่พิจารณาว่าปลายทางในการส่ง IP datagram นั้นเป็นภายในเครือข่ายตนเองหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยการพิจารณานี้โปรโตคอล IP Address ปลายทางว่าส่วนที่เป็นค่าหมายเลขเครือข่าย (network address) จะเหมือนกับค่าเลขหมายเครือข่าย IP Address ต้นทางหรือไม่ ถ้าค่าตรงกันแสดงว่าการส่งข้อมูลภายในเครือข่ายเดียวกัน แต่ถ้าค่าต่างกัน แสดงว่าต้องส่งข้อมูลไปยังปลายทางที่อยู่คนละเครือข่ายกัน

-การส่งข้อมูลภายในเครือข่ายเดียวกัน มีกลไกดังนี้

1. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARP (Address Resolution Protocol) เพื่อแปลงหมายเลข IP ปลายทางให้เป็นค่าหมายเลขฮาร์ดแวร์ เช่น MAC address

2. เมื่อโปรโตคอล IP ได้รับค่าหมายเลขฮาร์ดแวร์แล้ว ก็จะส่งข้อมูลไปยังฮาร์ดแวร์ที่ระบุไว้

-การส่งข้อมูลข้ามเครือข่าย มีกลไกดังนี้

1. โปรโตคอล IP ตรวจสอบพบว่าหมายเลข IP address ที่ปลายทางอยู่คนละเครือข่ายกันโดยโปรโตคอล IP จะอ่านค่า IP address ของ Router เพื่อเตรียมส่งข้อมูลไปที่ Router แทนซึ่งในที่นี้จะมีการกำหนดเป็น default router

2. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARP เพื่อแปลงค่า IP Address ของ Router ให้เป็นค่าหมายเลขฮาร์ดแวร์

3. โปรโตคอล IP ส่งข้อมูล IP datagram ไปยัง Router ส่งข้อมูลข้ามเครือข่ายไปตามขั้นตอน

IP Datagram

IP Datagram ประกอบด้วย IP Header และ IP Payload

IP Header	**IP Payload**

- **IP Header** เป็นขนาดที่เปลี่ยนแปลงได้ระหว่าง 20 และ 60 ไบต์ ในการเพิ่มขึ้น 4 ไบต์ มันจะจัดเตรียมการสนับสนุน routing , การแสดงตัว payload , การชี้ให้เห็นถึงขนาด IP Header และ Datagram , การสนับสนุน fragmentation โดยมีโครงสร้างดังรูปที่ 2.38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Version	IP Header Length	Type of Service	Total Length	Identifier	Flags	Fragment Offset
4 bit	4 bit	8 bit	16 bit	16 bit	3 bit	13 bit

Time-to-Live	Protocol	Header Checksum	Source IP Address	Destination IP Address	IP Option and Padding
8 bit	8 bit	16 bit	32 bit	32 bit	32 bit

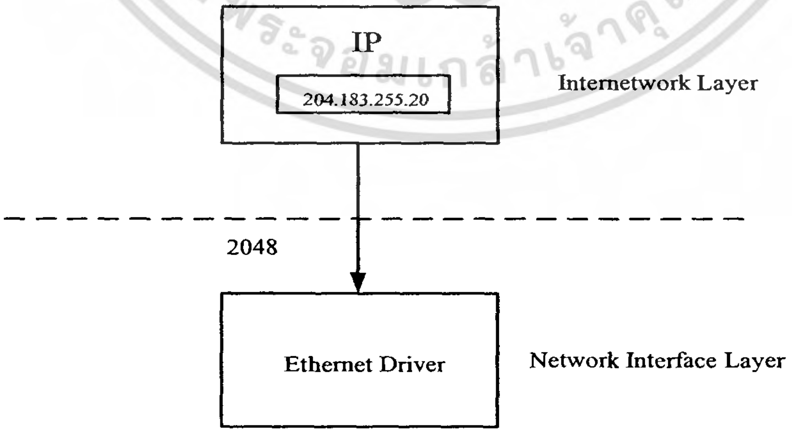
รูปที่ 2.38 แสดงโครงสร้าง IP Header

การกำหนด IP address ให้กับอุปกรณ์

ต้องกำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อเข้ากับเครือข่ายทุกจุด จุดเชื่อมต่อหรือ Interface อาจจะหมายถึง Network Interface card (LAN การ์ด) ที่ติดตั้งในเซร์ฟเวอร์หรือ WAN port , Ethernet port ที่Router ใช้เชื่อมต่อเข้ากับเครือข่ายเป็นต้นการกำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อนี้ทำให้เราเข้าใจได้ว่า ในบางอุปกรณ์ที่มีจุดเชื่อมต่อเข้ากับเครือข่ายมากกว่าหนึ่งจุด ต้องกำหนดหมายเลข IP address ให้ครบ

การ Bind IP address

เมื่อได้กำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อเช่น LAN card แล้วที่เครื่องเซร์ฟเวอร์จะต้องมีการ bindหรือผนวกค่า IP address ดังกล่าวเข้ากับ Ethernet driver เพื่ออ้างอิงหมายเลข IP กับฮาร์ดแวร์ให้ทำหน้าที่ติดต่อส่งข้อมูลในระดับ Network Interface ได้ต่อไป ดังตัวอย่างในรูปที่ 2.39



รูปที่ 2.39 แสดงการ Bind IP address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่2.39 จะแสดงค่า bind IP address 204.183.255.20 เข้ากับ Ethernet driver โปรโตคอล IP จะใช้ค่า IP address นี้ในการติดต่อกันและผ่านฮาร์ดแวร์ที่ถูก bind ไว้อีกต่อหนึ่ง ค่าหมายเลขฮาร์ดแวร์ได้แก่ MAC address ที่มีประจำอยู่บน LAN card ซึ่งจะไม่ได้ใช้งานอ้างอิงโดยตรงแต่จะผ่านหมายเลข IP address แทน

- **IP Payload** เป็นขนาดที่เปลี่ยนแปลงโดยมีลำดับจาก 8 ไบต์ (68 ไบต์ IP Datagram กับ 60ไบต์ IP Header) ถึง 65,515 ไบต์ (65,535 ไบต์ IP Datagram กับ 20 ไบต์ IP Header)

Protocol

Protocol Field มีขนาดยาว 1 ไบต์ และใช้เป็นตัวบ่งบอกในการบรรจุข้อมูลไปยัง Layer ที่สูงขึ้นกับใน IP Payload และยังเป็นตัวบ่งบอกลูกข่ายโปรโตคอลที่ชัดเจน โดยปกติค่าของ IP Protocol Field จะเป็น 1 สำหรับ ICMP6 สำหรับ TCP และ 17 (0x11) สำหรับ UDP Protocol field จะแสดงตัวได้อย่างมากมาย ดังนั้น payload สามารถที่จะผ่านไปยัง Layer ที่สูงขึ้นไปได้ถูกต้อง โดยจะได้รับที่destination

โปรโตคอล ICMP (Internet Control Message Protocol)

หน้าที่หลักของโปรโตคอล ICMP (Internet Control Message Protocol) คือ การแจ้งหรือแสดงข้อความจากระบบเพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้น ในการส่งผ่านข้อมูลนั้น ซึ่งปัญหาส่วนมากที่พบคือ ส่งไปไม่ได้ หรือปลายทางรับข้อมูลไม่ได้ เป็นต้น นากจากนี้ โปรโตคอล ICMP ยังถูกเรียกใช้งานจากเครื่อง Server และ Router อีกด้วย เพื่อแลกเปลี่ยนข้อมูลที่ใช้ควบคุม ส่วนรูปแบบการทำงานของโปรโตคอล ICMP นั้นจะทำควบคู่กับโปรโตคอล IP ในระบบเดียวกัน และข้อความที่แจ้งให้ทราบจะถูกผนึกอยู่ภายในข้อมูล IP (IP datagram) อีกทีหนึ่ง

ข้อความที่โปรโตคอล ICMP ส่งนั้น แบ่งออกได้เป็น 2 แบบ คือ ICMP Error Message หรือข้อความแจ้งข้อผิดพลาด และ ICMP Query หรือข้อความเรียกขอข้อมูลเพิ่ม ตัวอย่างกลไกการทำงานของโปรโตคอล ICMP เช่น เมื่อมีการส่งผ่านข้อมูลจากผู้ใช้ไปยังปลายทางที่ไม่ถูกต้อง หรือขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ ที่ Router จะส่งข้อความแจ้งเป็น ICMP Message ที่ชื่อ Destination Unreachable ให้กับผู้ส่งข้อมูลนั้น นอกจากนี้ตัวข้อมูลที่แจ้งข้อความ ก็จะมีส่วนของข้อมูล IP datagram ที่เกิดปัญหาด้วย ดังนั้น เมื่อผู้ส่งข้อมูลได้รับข้อความแจ้งแล้วก็จะทราบได้ว่า จุดที่เกิดปัญหานั้นอยู่ที่ใด ดังนั้น โปรโตคอล ICMP จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง Ping ที่เรามักใช้ทดสอบว่าเครื่อง Server ที่ให้บริการหรืออุปกรณ์ที่ต่ออยู่ในเครือข่ายอินเตอร์เน็ตนั้นยังทำงานเป็นปกติหรือไม่ แล้วคำสั่ง Ping มีการเรียกใช้งานโปรโตคอล ICMP แจ้งเป็นข้อความให้ทราบอีกด้วยต่อหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.19 ค่าต่างๆของ IP Protocol Field

Value	**Protocol**
0	Reserved
1	Internet Control Message Protocol (ICMP)
2	Internet Group Management Protocol (IGMP)
4	IP in IP encapsulation
6	Transmission Control Protocol (TCP)
8	Exterior Gateway Protocol (EGP)
17	User Datagram Protocol (UDP)
46	Resource Reservation Protocol (RSVP)
47	Generic Routing Protocol (GRE)
50	IP Security Encapsulating Security Payload (ESP)
51	IP Security Authentication Header (AH)
89	Open Shortest Path First (OSPF)

โปรโตคอล ARP (Addressable Resolution Protocol)

โปรโตคอล ARP ถูกเรียกใช้งานโดยโปรโตคอล IP เพื่อช่วยแปลงหมายเลข IP ไปเป็นหมายเลขฮาร์ดแวร์ปลายทาง ตัวอย่างเช่น เว็บเซิร์ฟเวอร์เครื่องหนึ่ง เชื่อมต่ออยู่ในเครือข่ายอินเตอร์เน็ตและในการเชื่อมต่อนี้ ต้องอาศัย LAN Card ที่ติดตั้งอยู่นี่เอง จะมีหมายเลขเฉพาะประจำฮาร์ดแวร์ ที่ไม่ซ้ำกับใคร เพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่มาใช้งานโปรโตคอล TCP/IP ก็จะต้องมีการกำหนดหมายเลข IP Address ประจำตัวเพื่อใช้อ้างอิงกัน และโปรโตคอล ARP จะทำหน้าที่แปลงค่าหมายเลข IP ให้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขฮาร์ดแวร์จริงให้ในระดับการทำงานที่ Internetwork Layer นี้ ซึ่งกลไกการแปลงนี้เรียกว่า Address Resolution

โปรโตคอล ARP ย้อนกลับ หรือ RARP (Reverse Addressable Protocol)

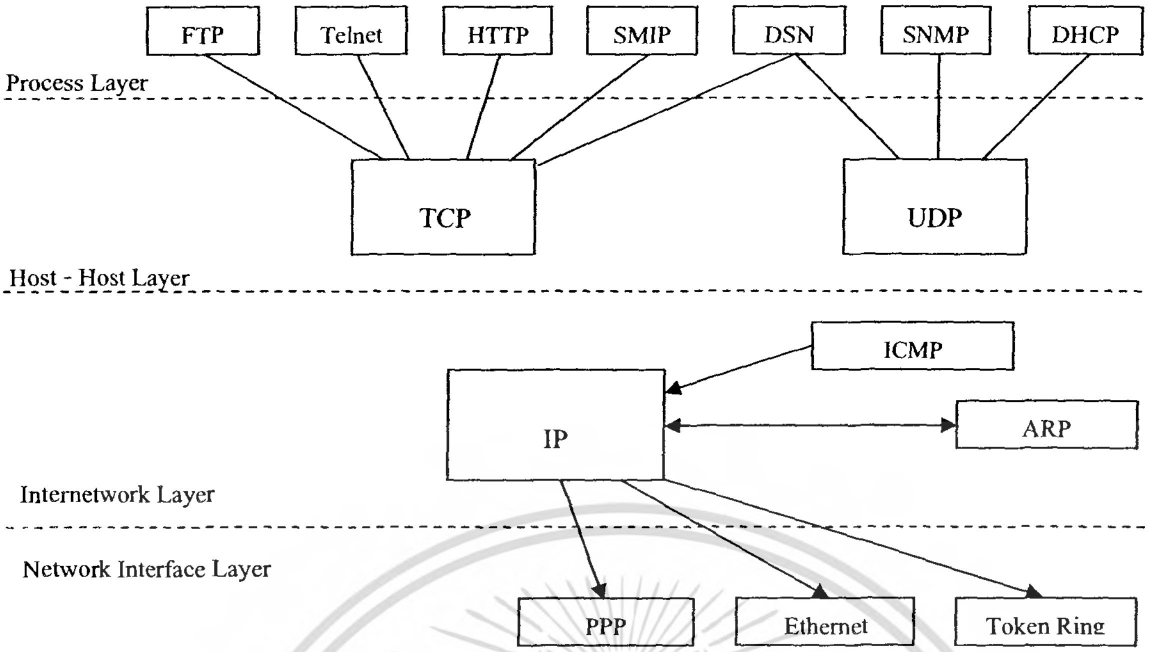
วิธีการ ARP ช่วยแก้ปัญหาในการค้นหาที่อยู่ของข้อมูลที่ใช้การกำหนดที่อยู่ฮาร์ดแวร์แบบ IP แต่ถ้าทราบที่อยู่แบบฮาร์ดแวร์ แล้วต้องการแปลงที่อยู่เป็น IP จะทำอย่างไร ปัญหานี้มักเกิดขึ้นกับเครื่อง Computer ที่เริ่มทำงานด้วยการอ่านข้อมูลทั้งหมดจากเครื่อง Host เครื่องประเภทนี้จะทราบเพียงที่อยู่ของตนเองจากอุปกรณ์สื่อสารเครือข่ายเท่านั้น การค้นหาคำตอบสามารถทำได้โดยวิธีควบคุมการสื่อสารแบบ ARP ย้อนกลับหรือ RARP (Reverse Addressable Protocol) วิธีการนี้ Computer ที่เพิ่งจะเริ่มทำงาน (หรือเครื่องใดก็แล้วแต่) จะส่งคำถามออกไปในทำนอง "ที่อยู่ขนาด 48 Bits แบบฮาร์ดแวร์ ของฉัน คือ 14.04.05.18.01.25 มีใครทราบที่อยู่ IP ของฉันบ้าง" เครื่องที่ให้บริการ RARP จะตรวจดูข้อมูลในตารางข้อมูลของตนเองแล้วจึงส่งหมายเลข IP กลับไปให้ วิธีการนี้ช่วยให้เกิดความอ่อนตัวและเพิ่มประสิทธิภาพในการใช้หมายเลข IP เนื่องจากผู้ใช้ไม่มีหมายเลข IP เนื่องจากผู้ใช้ไม่มีหมายเลข IP เป็นของตัวเอง ผู้ควบคุมระบบสามารถกำหนดหมายเลข IP ที่ไม่มีผู้ใช้งานในขณะนั้นให้ใช้ได้ หมายเลข IP ในที่นี้จึงเป็นเสมือนสมบัติส่วนกลางที่ทุกคนใช้ร่วมกัน

ข้อด้อยของวิธี RARP คือ การที่ผู้ใช้จะส่งคำถามโดยใช้หมายเลข 1 จำนวน 48 ตัว เป็นที่อยู่ของผู้ให้บริการหมายเลขนี้เป็นหมายเลขพิเศษที่ Router จะไม่ยอมส่ง Packet ผ่านไปยังเครือข่ายอื่นเลย ฉะนั้นผู้ให้บริการ RARP จะต้องมีอยู่ประจำทุกเครือข่าย อย่างไรก็ตาม Protocol แบบ BOOTP ได้รับการพัฒนาขึ้นมาเพื่อแก้ปัญหานี้โดยการใช้ Packet UDP แทน Packet ชนิดนี้ สามารถส่งไปได้ทั่วทุกเครือข่าย และยังให้ข้อมูลอื่นเพิ่มเติม เช่น หมายเลข IP ของผู้ให้บริการแฟ้มข้อมูล หมายเลข IP ของ Router อัตโนมัติ และตารางข้อมูลเครือข่ายย่อย เป็นต้น

Network Interface Layer

เนื่องจากในด้านกายภาพของเครือข่ายนั้น มีหลายวิธีการและหลายรูปแบบในการเชื่อมต่อระบบให้เป็นเครือข่าย แต่อย่างไรก็ตามในเครือข่ายอินเตอร์เน็ตนี้ ข้อมูลใน IP datagram จะถูกถ่ายทอดและส่งผ่านไปยังปลายทางโดยไม่คำนึงถึงรูปแบบการเชื่อมต่อทางกายภาพ ไม่ว่าจะเป็นการใช้เครือข่ายใยแก้วนำแสง หรือเครือข่ายสาย Unshielded Twist Pair (UTP) เชื่อมต่อเป็นแบบเครือข่าย Ethernet ธรรมดาหรือเครือข่าย Token Ring, ATM, ISDN ฯลฯ ก็ตาม การทำงานระดับล่างสุดต่อจาก Internetwork Layer จะเป็นการแปลงข้อมูล IP datagram ให้อยู่ในรูปที่เหมาะสม และแปลงสัญญาณไฟฟ้าส่งไปยังเครือข่ายต่อไป ซึ่งในชั้น Network Interface Layer นี้เมื่อเทียบกับมาตรฐาน OSI model แล้วจะเป็นการรวม 2 layer เข้าด้วยกัน คือ Data link layer และ Physical layer กล่าวโดยสรุป คือ การทำงานในชั้นต่าง ๆ ตามโครงสร้างโปรโตคอล TCP/IP จะมีลักษณะดังรูปที่ 2.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.40 โครงสร้างของโปรโตคอล TCP/IP

ในแต่ละชั้นหรือ layer จะมีโปรโตคอลหลักทำหน้าที่ต่าง ๆ และส่งข้อมูลไปยังเครือข่ายและออกสู่อินเตอร์เน็ต

MAC Address

เนื่องจากคอมพิวเตอร์แต่ละเครื่องสามารถแลกเปลี่ยนข้อมูลกันได้ในระบบเครือข่ายเดียวกัน ดังนั้นแต่ละเครื่องควรมีสิ่งที่ชี้ลักษณะเฉพาะตัวของมัน เช่น เราต้องมีบัตรประจำตัวประชาชน ซึ่งในทางคอมพิวเตอร์นี้เราจะใช้เลขฐาน 16 จำนวน 12 digits เป็นตัวบ่งชี้ลักษณะเฉพาะ ซึ่งเราเรียกว่า MAC Address เนื่องจาก MAC Address เป็นตัวบ่งชี้ลักษณะเฉพาะของแต่ละเครื่อง ดังนั้นจึงต้องเป็นค่าที่ไม่ซ้ำกัน (unique) MAC Address เป็นเลข 48 bits โดยแบ่งออกเป็น 2 ส่วน โดย 24 bits แรกเป็นค่าที่แสดงถึงบริษัทที่ผลิตการ์ด ส่วน 24 bits หลังเป็น serial number ที่ทางบริษัทกำหนดให้ ซึ่งแต่ละตัวต้องไม่ซ้ำกัน เราเรียกเลข 24 bit นี้ว่า OUI (Organizationally Unique Identifier) ซึ่ง OUI จะใช้เพียง 22 bits เท่านั้น ส่วนอีก 2 bits ที่เหลือจะถูกใช้เพื่อวัตถุประสงค์อื่น โดย bit หนึ่งจะใช้เพื่อแสดงว่า address นั้นเป็น broadcast/multicast address ส่วนอีก bit หนึ่งนั้นไว้แสดงว่า adapter นั้นถูกกำหนด locally administered address ซึ่ง admin ของระบบจะทำการกำหนด MAC Address เพื่อความเหมาะสมของนโยบายระบบ เช่น MAC Address = 03 00 00 00 00 00 01 ซึ่งจะเห็นว่า byte แรก = 03 = 00000011 นั่นคือ ทั้ง 2 bits ถูก set (reset = 0) ซึ่งเอาไว้กรณี multicast ให้ทุกเครื่องที่ run บนโปรโตคอล NetBEUI

กล่าวโดยสรุป คือ โปรโตคอล TCP/IP ทำงานโดยแบ่งเป็นชั้นเทียบกับ OSI model ได้กลไกในการทำงานของโปรโตคอล TCP/IP มี 4 ชั้น ซึ่งในชั้นแรก คือ Process layer ทำหน้าที่ติดต่อกับแอพพลิเคชันและโปรโตคอลที่แอพพลิเคชันนั้น ๆ ใช้งาน และส่งมาให้ชั้น Host-to-Host layer เพื่อติดต่อกันระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องผู้ขอใช้บริการ ในชั้นนี้จะมีการสร้าง Session หรือการเชื่อมต่อระหว่างระบบขึ้นตามแต่ละโปรโตคอลที่ต้องการ ต่อมาเป็นการผนึกข้อมูลไปเป็น IP datagram ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

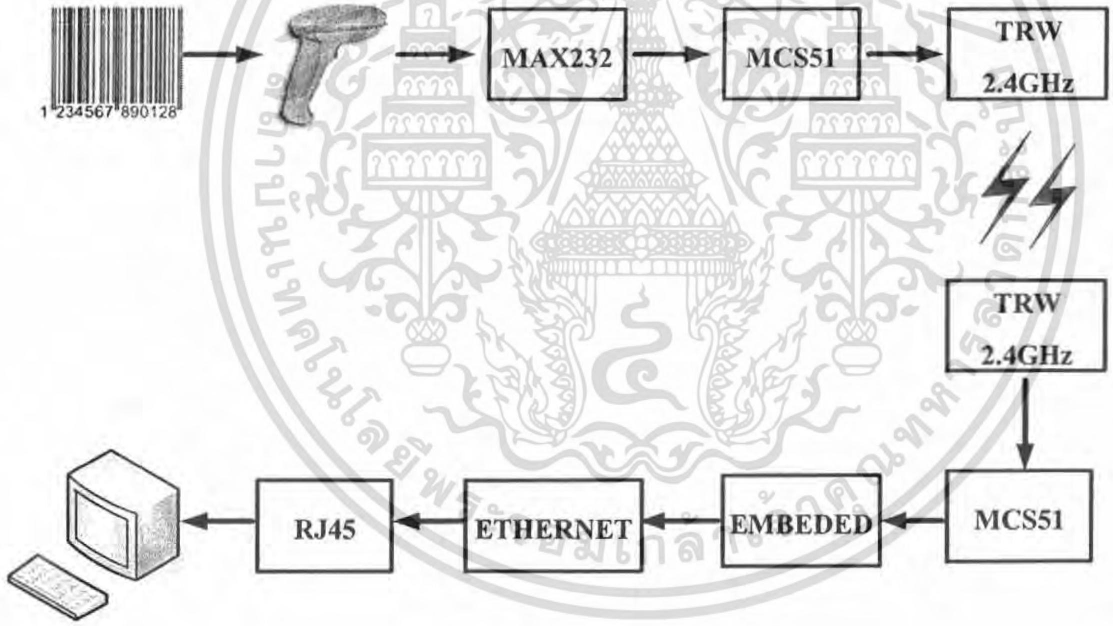
ชั้น Internetwork Layer โดยอาศัยโปรโตคอล IP เพื่อให้สามารถติดต่อข้อมูลข้ามเครือข่ายและเครื่องที่ถูกต้องได้ และสุดท้ายการส่งข้อมูลออกสู่โลกภายนอกต้องอาศัยกลในชั้น Network Interface Layer เพื่อแปลงข้อมูลใหม่ให้เป็นสัญญาณไฟฟ้าส่งออกไปเครือข่ายและอาจจะออกไปยัง Gateway หรือ Router เพื่อข้ามเครือข่ายออกไปยังเส้นทางที่กำหนดไว้ในอินเตอร์เน็ตโปรโตคอล ในแต่ละโปรโตคอลเหล่านี้ก็จะรับผิดชอบหน้าที่ของตนเพื่อผ่านข้อมูลลงไปยังระดับล่าง และออกสู่เครือข่ายอินเตอร์เน็ตในที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

3.1 บล็อกไดอะแกรม

ในโครงงานนี้ใช้เครื่องอ่านบาร์โค้ดแบบซีซีดีในการอ่านบาร์โค้ด แล้วทำการส่งข้อมูลที่ได้จากการอ่านผ่านวงจรMAX232 ที่ทำหน้าที่เปลี่ยนแรงดันที่เข้ามาจาก Serial Port ซึ่งเป็นแรงดันมาตรฐานของ RS-232 โดยเปลี่ยนระดับแรงดันเป็นแบบ TTL เพื่อให้ใช้ได้กับวงจรไมโครคอนโทรลเลอร์ ที่ทำหน้าที่ควบคุมการทำงานของวงจร เพื่อส่งค่าบาร์โค้ดไปที่วงจรส่วน TRW2.4GHz แบบ transmitter ส่งค่าบาร์โค้ดออกไปในรูปของคลื่นวิทยุ ไปยังTRW2.4GHz แบบ receiver อีกตัวหนึ่งที่อยู่ทางด้านรับ ทำหน้าที่เป็นตัวรับสัญญาณเข้า มาแล้วใช้วงจรไมโครคอนโทรลเลอร์ด้านรับเป็นตัวควบคุม โดยนำข้อมูลที่รับได้จากTRW2.4GHz เข้ามาในวงจรEmbeded เพื่อนำข้อมูลที่ได้ส่งเข้าผ่าน RJ45 เพื่อนำข้อมูลไปแสดงผลทางหน้าจอคอมพิวเตอร์ ดังรูปที่ 3.1

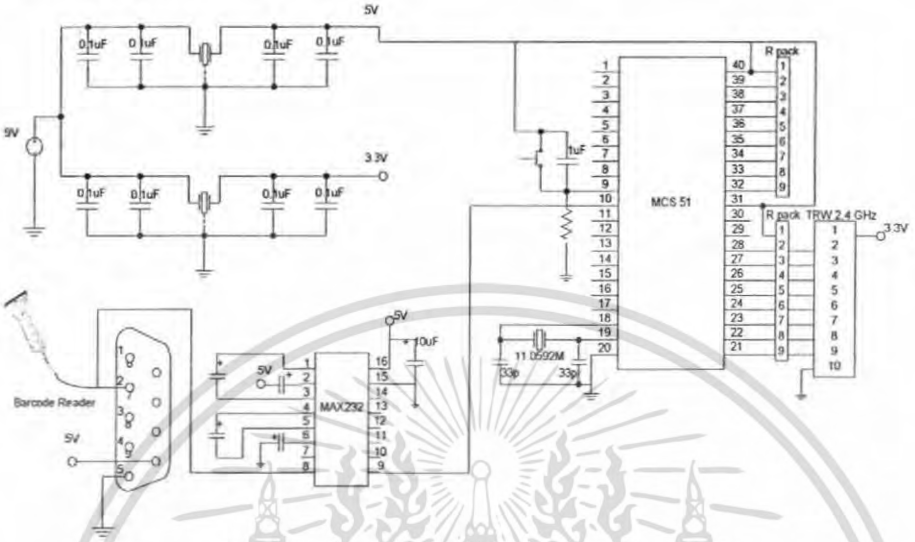


รูปที่ 3.1 บล็อกไดอะแกรมของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

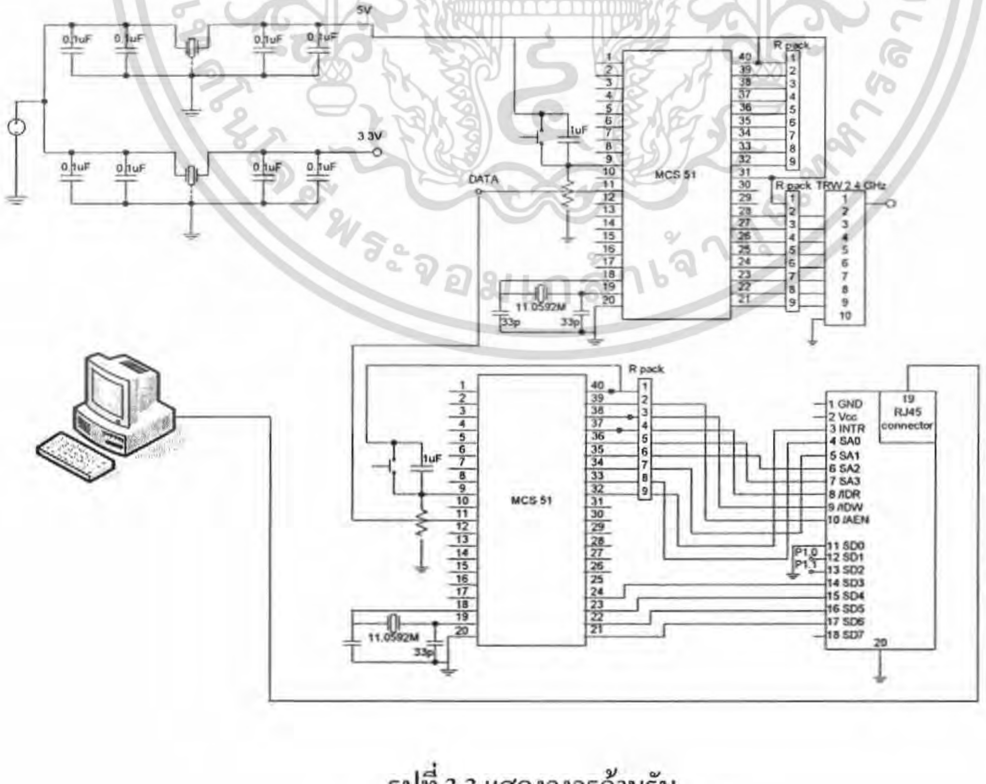
3.2 ไมโครคอนโทรลเลอร์

วงจรรวมทางด้านส่งซึ่งประกอบด้วยเครื่องอ่านบาร์โค้ด , วงจรMAX232 ,ไมโครคอนโทรลเลอร์ และวงจรแปลงไฟฟ้า ดังรูปที่ 3.2



รูปที่ 3.2 แสดงวงจรด้านส่ง

วงจรรวมทางด้านรับซึ่งประกอบด้วยเครื่องอ่านบาร์โค้ด,ไมโครคอนโทรลเลอร์,วงจรแปลงไฟฟ้าและEmbeded ดังรูปที่ 3.3

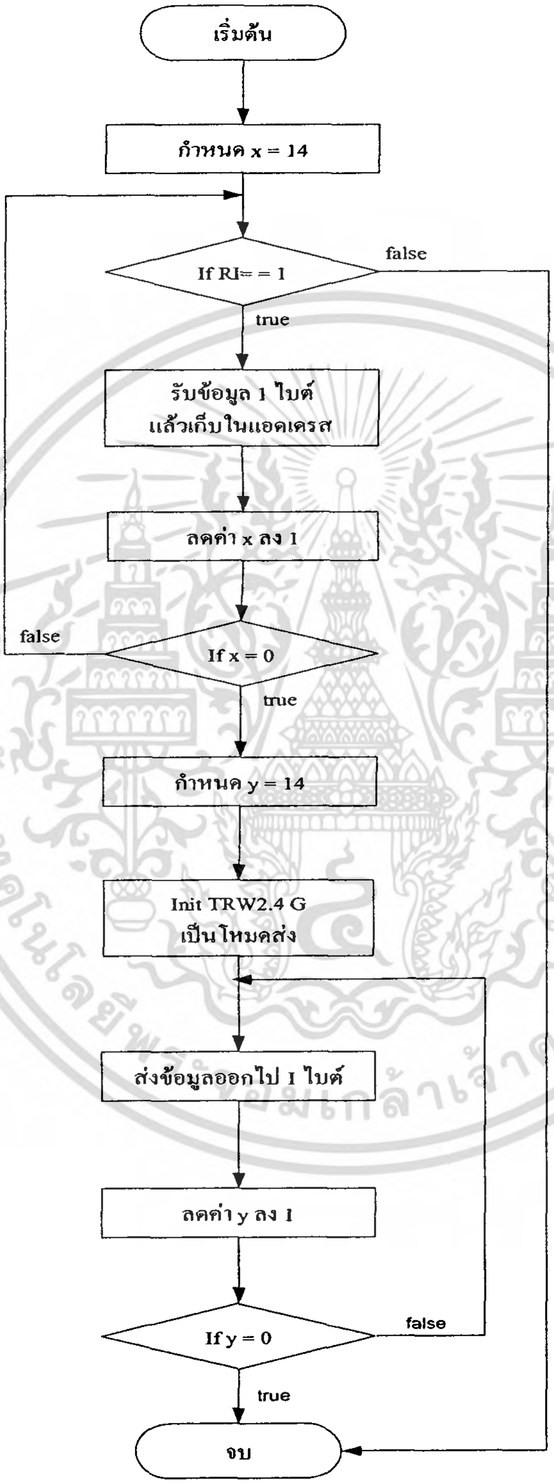


รูปที่ 3.3 แสดงวงจรด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบโฟลวชาร์ทของวงจรรวมทางด้านส่ง

ในการทำงานทางด้านส่งมีการออกแบบขั้นตอนการทำงานดังโฟลวชาร์ทดังรูปที่ 3.4

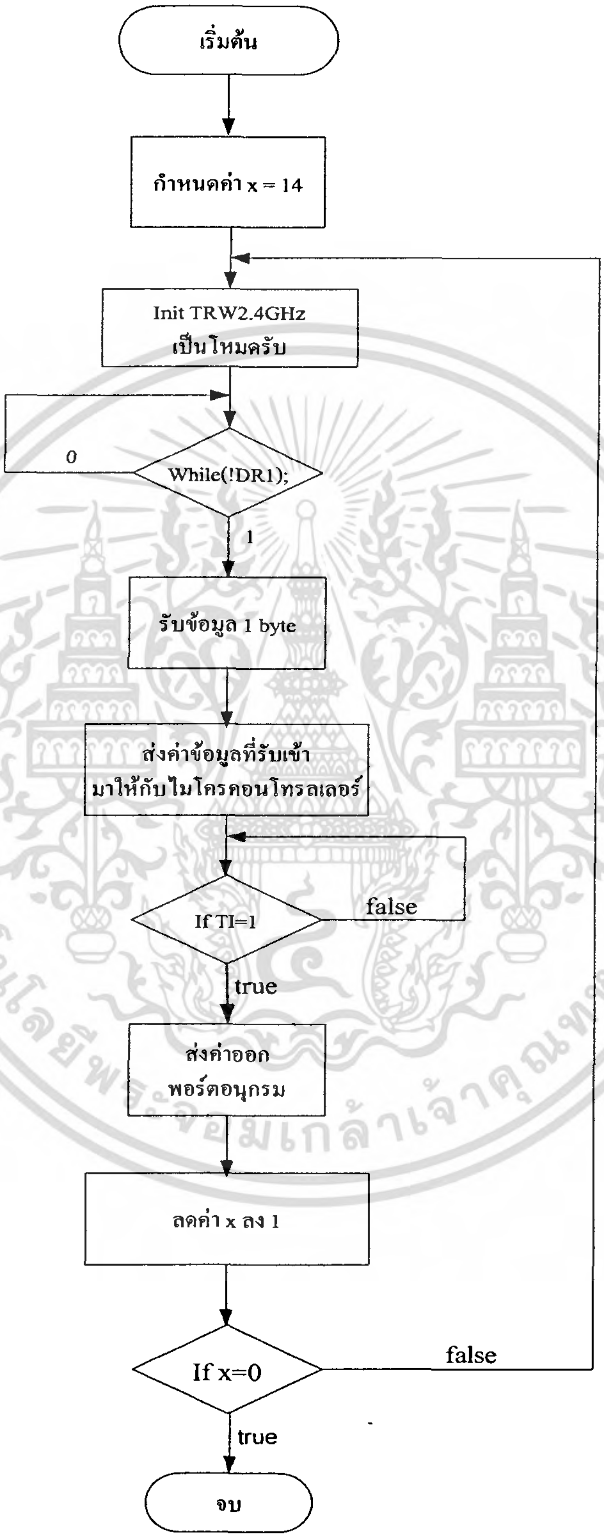


รูปที่ 3.4 โฟลวชาร์ทของวงจรรวมทางด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบโฟลวชาร์ทวงจรรวมทางด้านรับ

ในการทำงานทางด้านส่งมีการออกแบบขั้นตอนการทำงานดังโฟลวชาร์ทดังรูปที่ 3.5

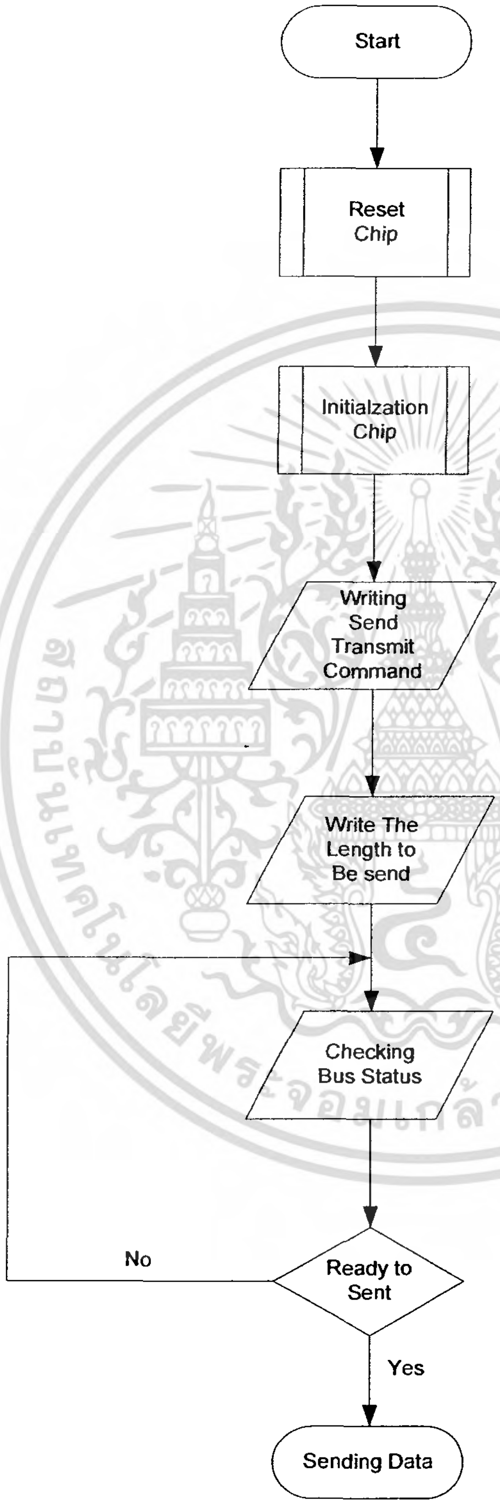


รูปที่ 3.5 โฟลวชาร์ทของวงจรรวมทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การออกแบบโฟลว์ชาร์ทในการส่งข้อมูลของ Embeded

ในการทำงานของ embedded มีการออกแบบขั้นตอนการทำงานดังโฟลวชาร์ทดังรูปที่ 3.6

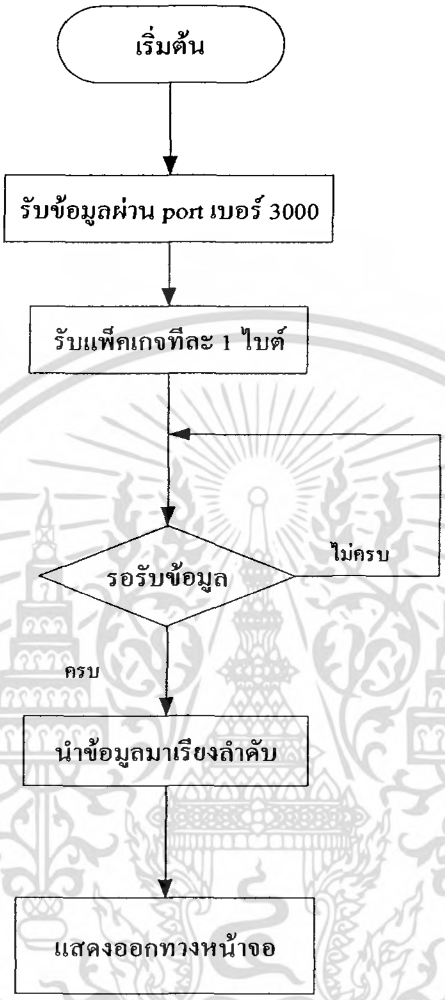


รูปที่ 3.6 โฟลวชาร์ทของในการส่งข้อมูลของ Embeded

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การออกแบบโฟลว์ชาร์ทการแสดงข้อมูลออกหน้าจอคอมพิวเตอร์

ในการออกแบบการแสดงหน้าจอคอมพิวเตอร์มีการทำงานดังรูปที่ 3.7



รูปที่ 3.7 โฟลว์ชาร์ทการแสดงข้อมูลออกหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

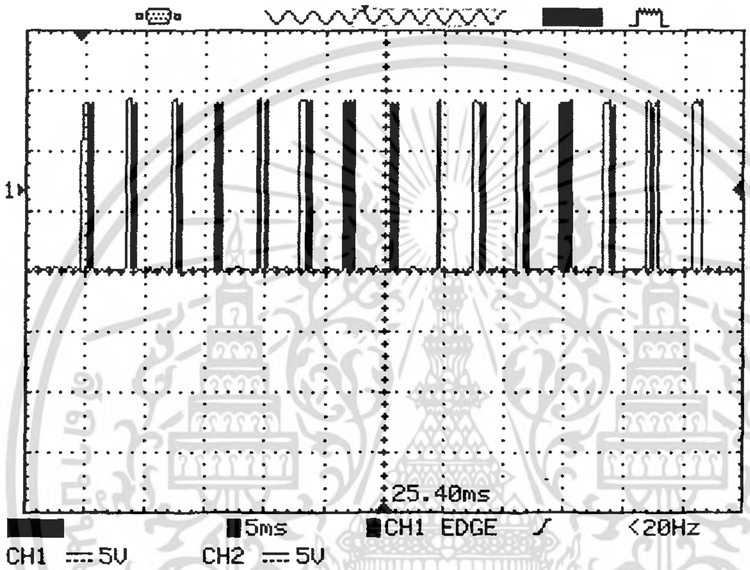
การทดลองและผลการทดลอง

4.1การทดลองวัดค่าสัญญาณจากเครื่องอ่านบาร์โค้ด

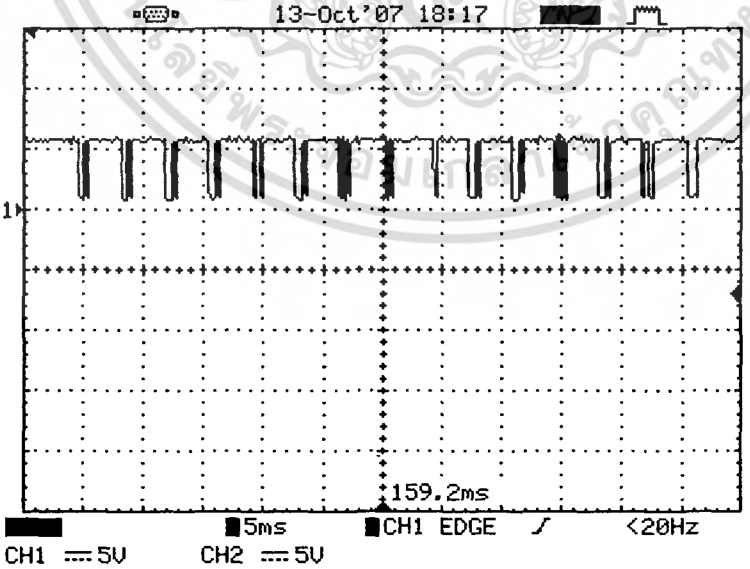
ขั้นตอนการทดลอง

1. จ่ายไฟเลี้ยงให้กับเครื่องอ่านบาร์โค้ดและทำการแสกนบาร์โค้ด
2. ทำการวัดค่าสัญญาณข้อมูลที่ขา 2 ของพอร์ตอนุกรม
3. ทำการวัดค่าสัญญาณเมื่อผ่านวงจร MAX-232

ผลการทดลอง



รูปที่ 4.1 แสดงสัญญาณข้อมูลที่อ่านจากเครื่องอ่านบาร์โค้ด (แบบ Asynchronous)



รูปที่ 4.2 แสดงสัญญาณข้อมูลแบบ TTLที่วัดจากขา 9 ของMAX-232

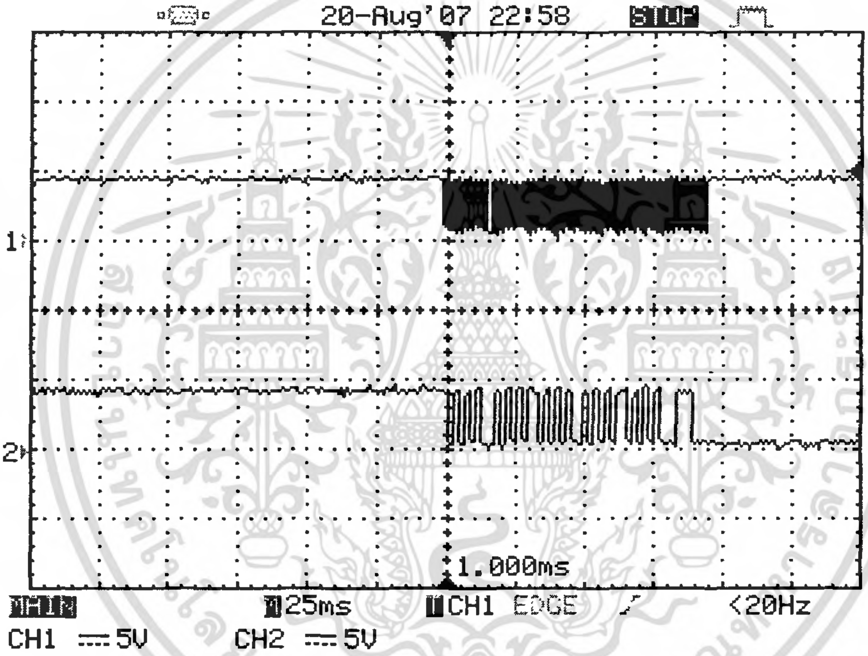
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2การทดลองวัดค่าสัญญาณที่ส่งจาก AT89C51 ไปยัง TRW2.4GHz (ด้านส่ง)

ขั้นตอนการทดลอง

1. เขียนโปรแกรมในไมโครคอนโทรลเลอร์ด้านส่งที่แสดงในภาคผนวก ซึ่งมีโฟลว์ชาร์ทตามรูปที่ 3.4
2. ต่อวงจรตามรูปที่ 3.2
3. จ่ายไฟเลี้ยงให้กับวงจร และทำการสแกนบาร์โค้ด
4. วัดค่าสัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz ให้แสดงในช่องสัญญาณที่ 1
5. ทำการวัดค่าสัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz ด้านส่ง ให้แสดงในช่องสัญญาณที่ 2

ผลการทดลอง



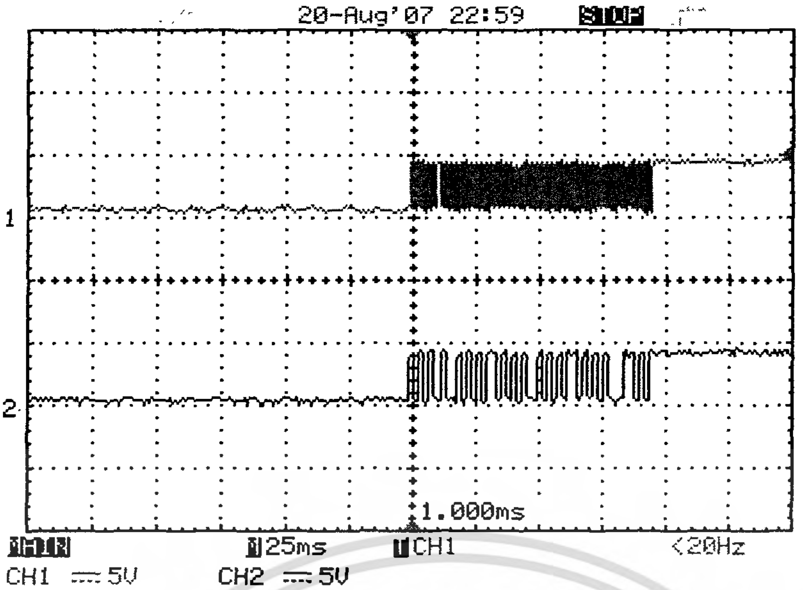
รูปที่ 4.3 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 1 รหัส 8)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.3 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่ประกอบด้วยข้อมูลในส่วนของเฮดเดอร์ 5ไบต์ และ ข้อมูลที่อ่านได้จากเครื่องอ่านบาร์โค้ด 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00111000 ซึ่งตรงกับ เลข 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

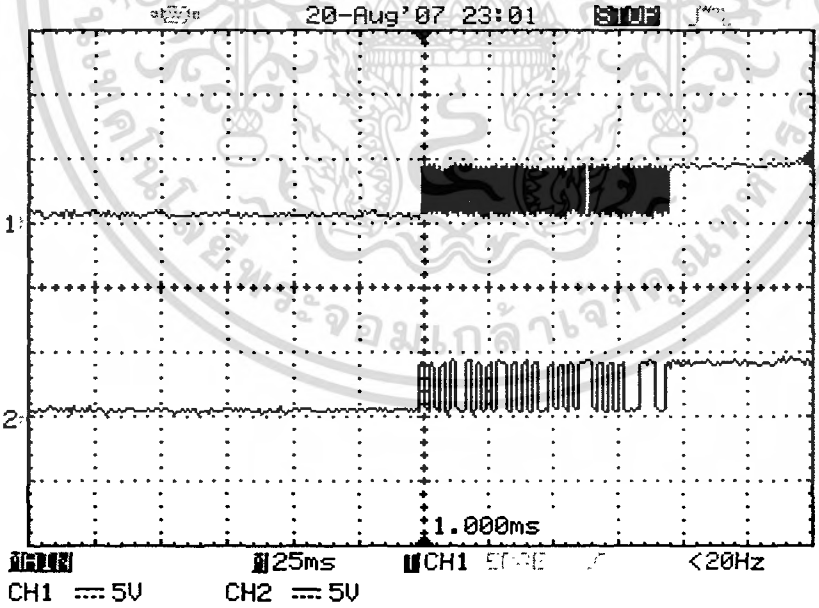


รูปที่ 4.4 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 3 รหัส 5)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.4 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่ประกอบด้วยข้อมูลในส่วนของเฮดเดอร์ 5ไบต์ และ ข้อมูลที่อ่านได้จากเครื่องอ่านบาร์โค้ด 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00110101 ซึ่งตรงกับ เลข 5



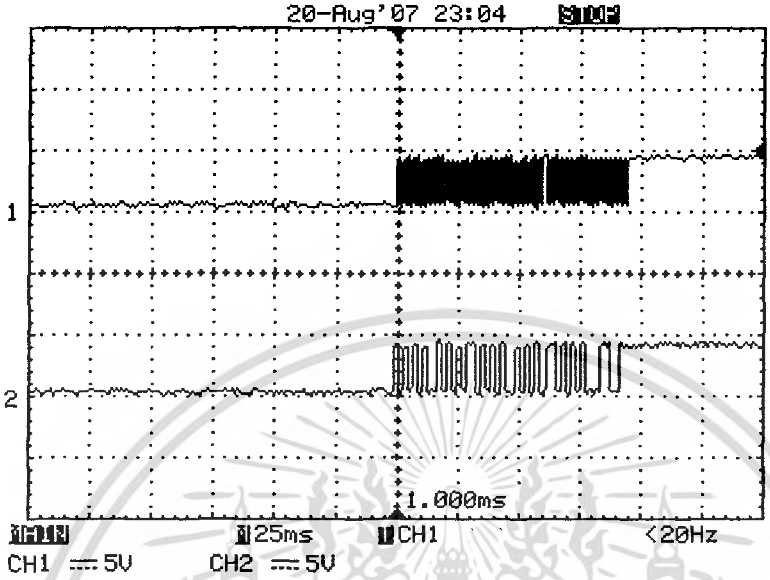
รูปที่ 4.5 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 5 รหัส 9)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

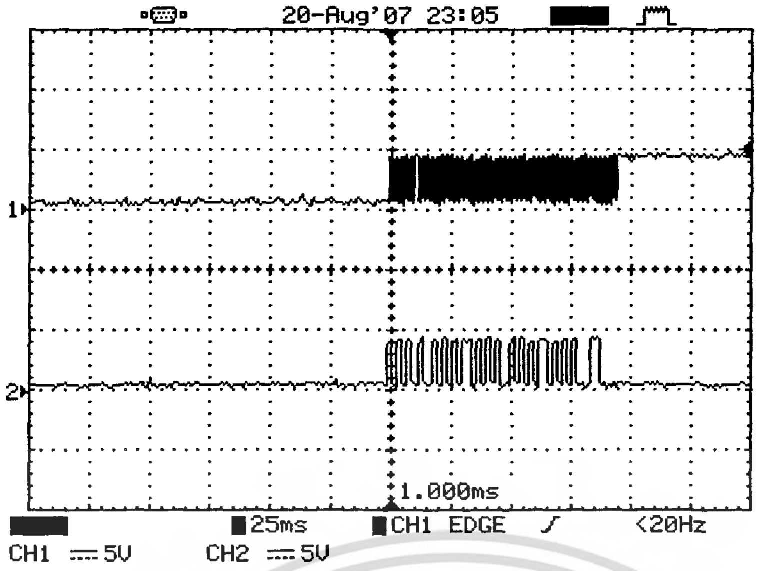
จากรูปที่ 4.4 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่ประกอบด้วยข้อมูลในส่วนของเฮดเดอร์ 5ไบต์ และ ข้อมูลที่อ่านได้จากเครื่องอ่านบาร์โค้ด 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00111001 ซึ่งตรงกับ เลข 9



รูปที่ 4.6 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 8 รหัส 3)
ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz
ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.6 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่ประกอบด้วยข้อมูลในส่วนของเฮดเดอร์ 5ไบต์ และ ข้อมูลที่อ่านได้จากเครื่องอ่านบาร์โค้ด 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00110011 ซึ่งตรงกับ เลข 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

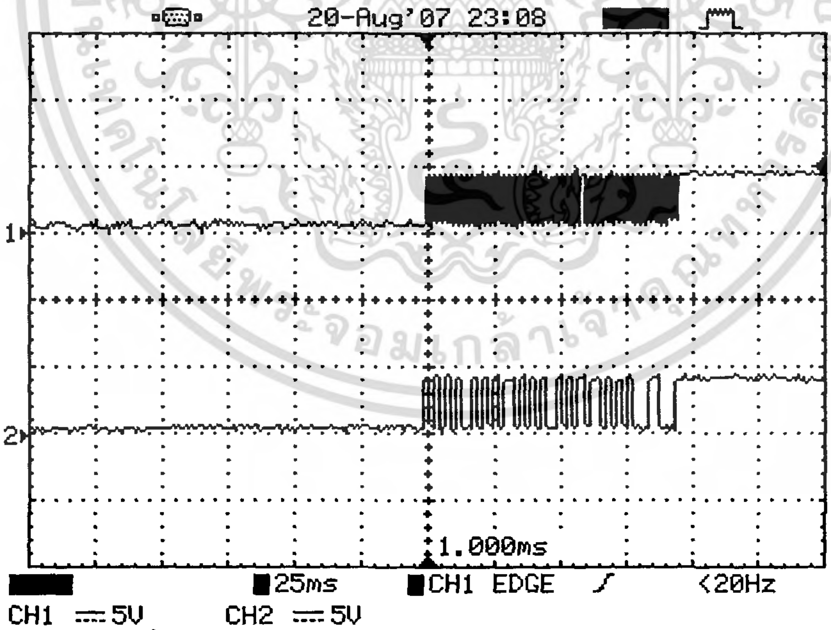


รูปที่ 4.7 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 10 รหัส 0)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.7 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่ประกอบด้วยข้อมูลในส่วนของเฮดเดอร์ 5ไบต์ และ ข้อมูลที่อ่านได้จากเครื่องอ่านบาร์โค้ด 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00110000 ซึ่งตรงกับ เลข 0



รูปที่ 4.8 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์ทำการส่งออก (ไบต์ที่ 13 รหัส 1)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

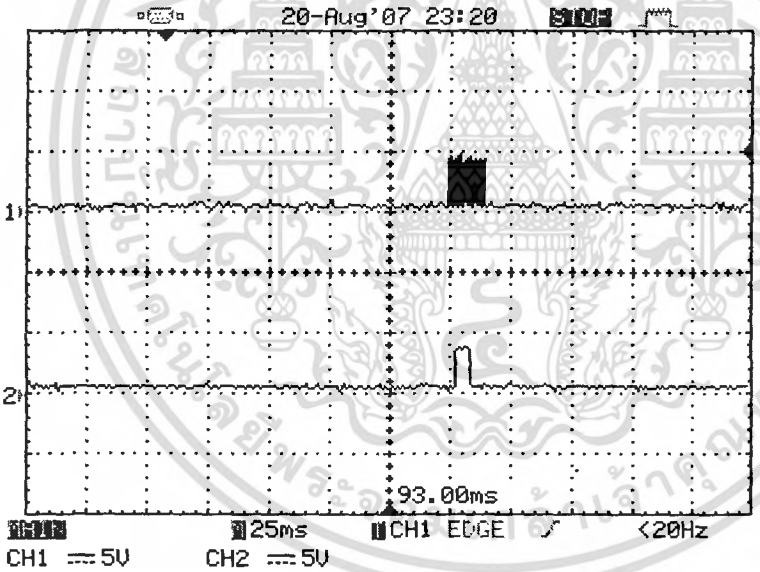
จากรูปที่ 4.8 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่ประกอบด้วยข้อมูลในส่วนของเฮดเดอร์ 5ไบต์ และ ข้อมูลที่อ่านได้จากเครื่องอ่านบาร์โค้ด 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00110001 ซึ่งตรงกับ เลข 1

4.3 การทดลองวัดค่าสัญญาณที่ส่งจาก TRW2.4GHz ไปยัง AT89C51 (ด้านรับ)

ขั้นตอนการทดลอง

1. เขียนโปรแกรมในไมโครคอนโทรลเลอร์ด้านรับที่แสดงในภาคผนวก ซึ่งมีโฟลว์ชาร์ทดังรูปที่ 3.5 เพื่อทำการรับค่าจาก TRW 2.4GHz
2. ต่อวงจรตามรูปที่ 3.3
3. จ่ายไฟให้กับวงจรและทำการสแกนบาร์โค้ด
4. วัดค่าสัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz ให้แสดงในช่องสัญญาณที่ 1
5. ทำการวัดค่าสัญญาณที่ขา 6 ของTRW2.4GHz ด้านรับ ให้แสดงในช่องสัญญาณที่ 2

ผลการทดลอง



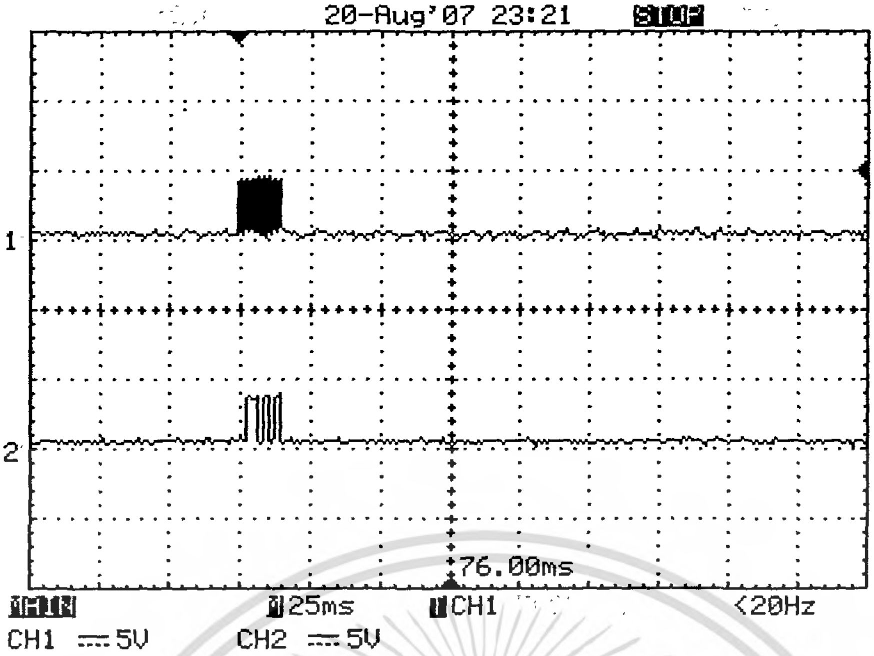
รูปที่ 4.9 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 1 รหัส 8)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.9 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่รับได้ 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00111000 ซึ่งตรงกับ เลข 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

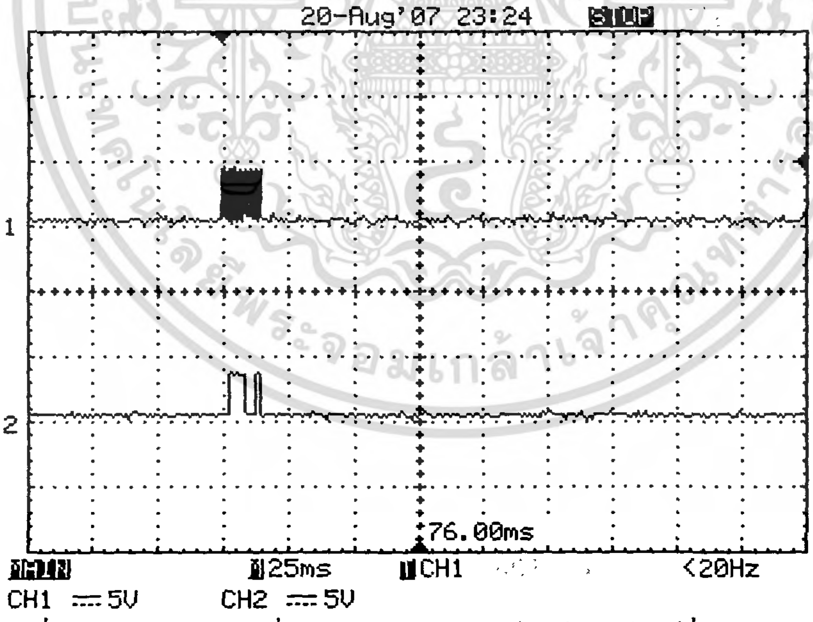


รูปที่ 4.10 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 3 รหัส 5)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.10 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่รับได้ 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00110101 ซึ่งตรงกับ เลข 5



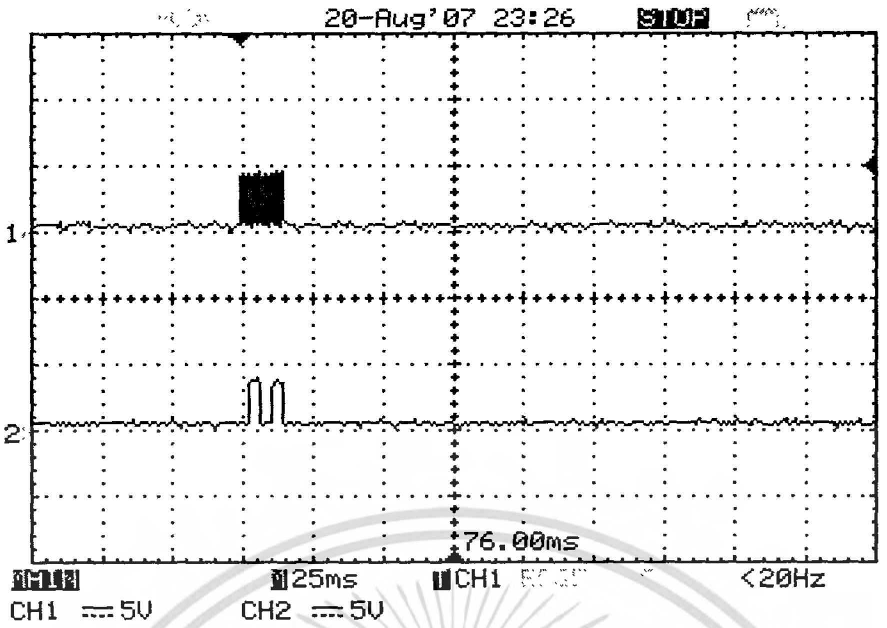
รูปที่ 4.11 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 5 รหัส 9)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.11 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่รับได้ 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00111001 ซึ่งตรงกับ เลข 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

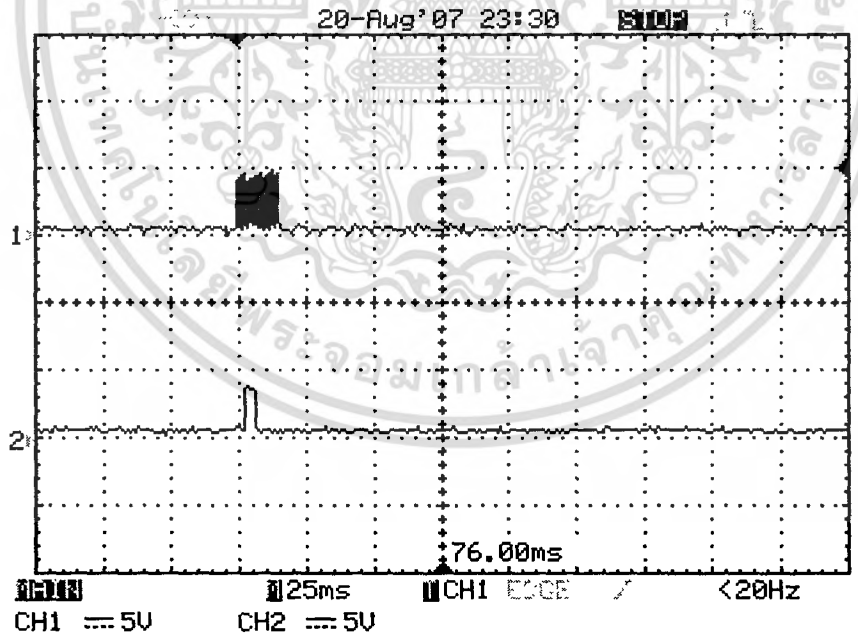


รูปที่ 4.12 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 8 รหัส 3)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.12 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่รับได้ 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00110011 ซึ่งตรงกับ เลข 3



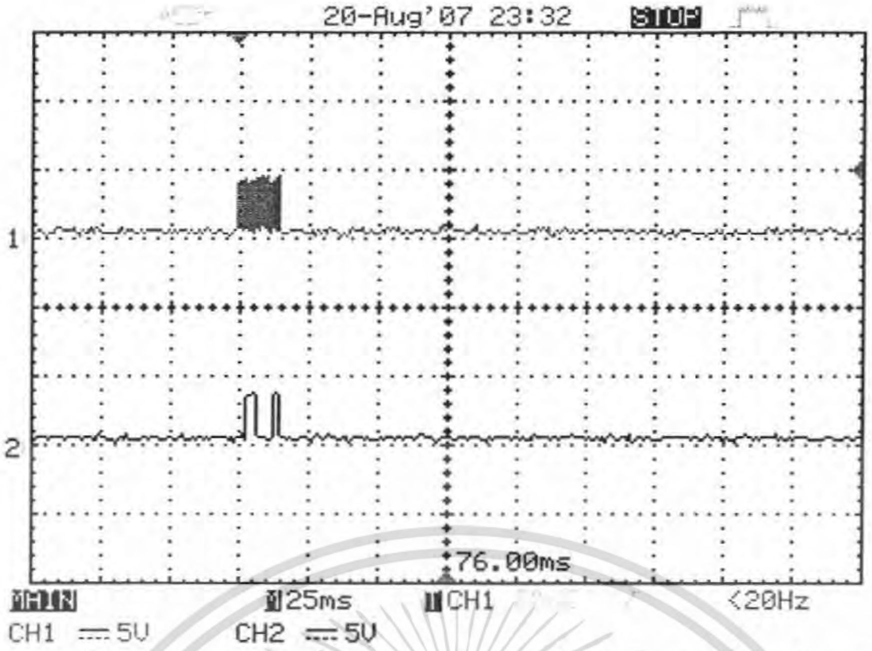
รูปที่ 4.13 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 10 รหัส 0)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

จากรูปที่ 4.13 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่รับได้ 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00110000 ซึ่งตรงกับ เลข 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงสัญญาณที่ไมโครคอนโทรลเลอร์รับข้อมูล (ไบต์ที่ 13 รหัส 1)

ช่องสัญญาณที่ 1 : สัญญาณนาฬิกาที่ขา 5 ของTRW2.4GHz

ช่องสัญญาณที่ 2 : สัญญาณข้อมูลที่ขา 6 ของTRW2.4GHz

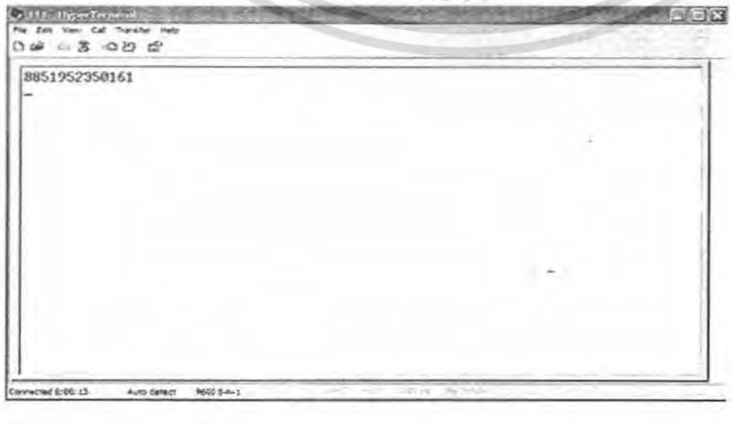
จากรูปที่ 4.14 ที่ช่องสัญญาณที่ 1 มีสัญญาณนาฬิกาเฉพาะช่วงที่ข้อมูลและที่ช่องสัญญาณที่ 2 เป็นสัญญาณข้อมูลที่รับได้ 1 ไบต์ซึ่ง เป็นรหัส ASCII ในรูปนี้คือ 00110001 ซึ่งตรงกับ เลข 1

4.4 การทดลองแสดงผลเป็นหมายเลขบาร์โค้ดบนไฮเปอร์เทอร์มินอล

ขั้นตอนการทดลอง

1. จากวงจรที่ 3.3 ทำการเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม
2. จ่ายไฟเลี้ยงให้กับวงจร และสแกนบาร์โค้ดทางด้านส่ง
3. เปิดโปรแกรมไฮเปอร์เทอร์มินอล เพื่อรับค่าจากพอร์ต COM1

ผลการทดลอง



รูปที่ 4.15 แสดงหน้าจอโปรแกรมไฮเปอร์เทอร์มินอลเมื่อได้รับหมายเลขบาร์โค้ด

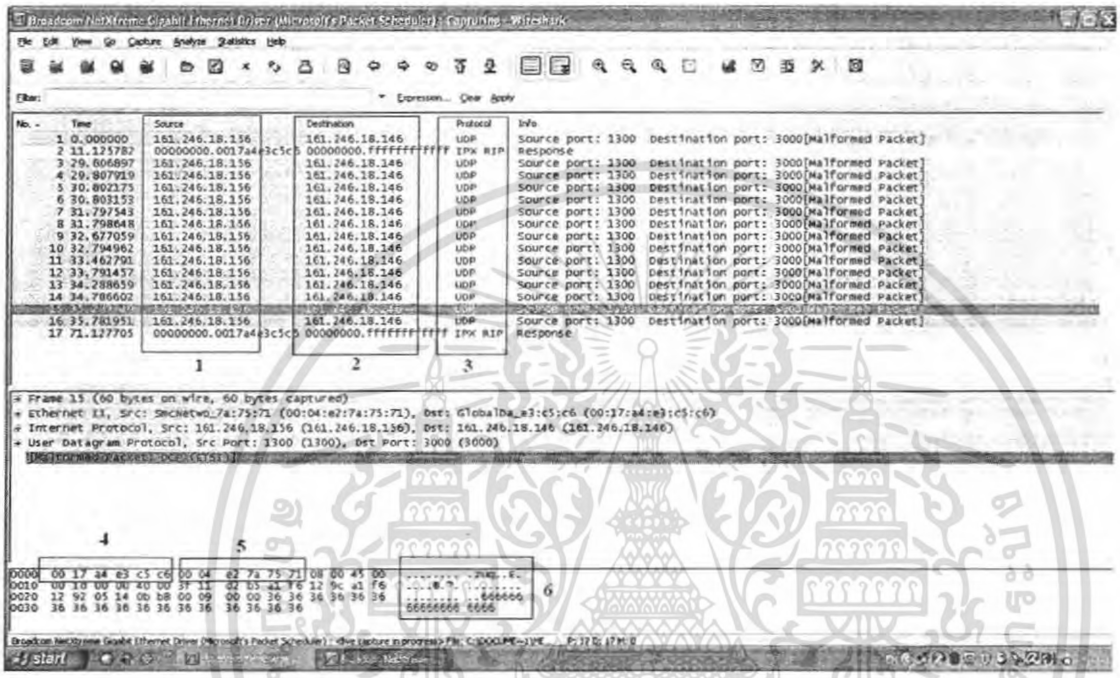
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองแสดงผลข้อมูลที่เข้ามาในคอมพิวเตอร์โดยใช้โปรแกรม wireshark

ขั้นตอนการทดลอง

1. จากวงจรที่ 3.3 ทำการเชื่อมต่อกับคอมพิวเตอร์ผ่านRJ45
2. จ่ายไฟเลี้ยงให้กับวงจร และสแกนบาร์โค้ดทางด้านส่ง
3. เปิดโปรแกรม wireshark เพื่อทำการจับpacketที่เข้ามาในคอมพิวเตอร์ทางด้านรับ

ผลการทดลอง



รูปที่ 4.16 แสดงหน้าจอโปรแกรม wireshark เมื่อมีการส่ง packet เข้ามาในคอมพิวเตอร์

จากรูปที่ 4.16 สามารถอธิบายได้ดังนี้

หมายเลข 1 : IP Address ของต้นทางซึ่งเป็น IP Address ของ Embeded คือ 161.246.18.156

หมายเลข 2 : IP Address ของปลายทางซึ่งเป็น IP Address ของ เครื่องคอมพิวเตอร์ คือ 161.246.18.146

หมายเลข 3 : ต้นทางและปลายทางมีการติดต่อสื่อสาร โดยใช้โปรโตคอล UDP

หมายเลข 4 : MAC Address ของปลายทางซึ่งเป็น MAC Address ของเครื่องคอมพิวเตอร์ คือ 00-17-a4-e3-c5-c6

หมายเลข 5 : MAC Address ของต้นทางซึ่งเป็นเป็น MAC Address ของ Embeded คือ 00-04-e2-7a-75-71

หมายเลข 6 : แสดงข้อมูลอยู่ใน packet ซึ่งเป็นข้อมูลที่ได้จากการอ่านบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การทดลองแสดงข้อมูลทางหน้าจอคอมพิวเตอร์

ขั้นตอนการทดลอง

1. รันโปรแกรมที่เขียนโดยภาษาจาวาตามโฟลว์ชาร์ทรูปที่ 3.7ในCommand Prompt
2. สแกนบาร์โค้ดทางด้านส่ง

ผลการทดลอง

รูปที่ 4.17 แสดงข้อมูลผ่านทางหน้าจอคอมพิวเตอร์

จากรูปที่ 4.17 จะแสดงหมายเลข IP Address ของต้นทางซึ่งเป็น IP Address ของ Embeded คือ 161.246.18.156 ตามด้วยจำนวนไบต์ของข้อมูลคือ 14 ไบต์และข้อมูลที่รับมาคือ 8851952350161 จำนวน 13 ไบต์รวมกับอีก 1 ไบต์สำหรับแสดงการสิ้นสุดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุปผล

5.1 บทวิจารณ์

จากการศึกษาเกี่ยวกับการทำงานของเครื่องอ่านบาร์โค้ด ไมโครคอนโทรลเลอร์ เครื่องรับส่งผ่านคลื่นวิทยุ และส่วนของEmbeded พบปัญหาต่อไปนี้

5.1.1 ส่วนของเครื่องอ่านบาร์โค้ด

สำหรับเครื่องอ่านบาร์โค้ดที่ใช้ในโครงงานนี้ สามารถใช้งานได้ดี โดยเครื่องอ่านมีไฟเลี้ยงอยู่ที่ 5 โวลท์ จึงต้องใช้หม้อแปลง แต่เนื่องจากโครงงานนี้ต้องการทำให้เครื่องอ่านบาร์โค้ดเป็นแบบไร้สาย จึงต้องใช้วงจรแปลงแรงดันจากแหล่งจ่ายไฟซึ่งเป็นแบตเตอรี่แรงดัน 9 โวลท์ ให้เป็น 5 โวลท์ โดยในการใช้งานนั้นต้องระวังไม่ให้เกิดการช็อต เพราะจะทำให้เครื่องอ่านบาร์โค้ดไม่สามารถอ่านข้อมูลได้ตามปกติ

5.1.2 ส่วนของไมโครคอนโทรลเลอร์และพอร์ทอนุกรม

จากการออกแบบระบบสื่อสารไร้สาย ได้ทำการออกแบบโดยใช้ไมโครคอนโทรลเลอร์ทางด้านส่งทำหน้าที่ในการรับค่าจากเครื่องอ่านบาร์โค้ดผ่านทางพอร์ทอนุกรม และไมโครคอนโทรลเลอร์ทางด้านรับทำหน้าที่ในการส่งค่าออกไปยัง Embeded ผ่านทางพอร์ทอนุกรม ซึ่งการส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์และพอร์ทอนุกรมจำเป็นที่ต้องมีการแปลงแรงดันให้เป็นแรงดันที่เหมาะสมกับการใช้งานโดยใช้วงจร MAX232

ในส่วนของโปรแกรมนั้นมีปัญหาในการคำนวณเวลาที่ใช้ในการทำงานของโปรแกรมเพื่อให้สามารถรับส่งข้อมูลได้ทันกัน จึงต้องมีการทดลองหลายครั้ง และในส่วนของโปรแกรมทางด้านรับนั้นพบว่า มีการเขียนโปรแกรมทำการ Initial TRW2.4 GHz ทุกครั้งที่จะรับข้อมูลแต่ละไบต์ ซึ่งไม่มีความจำเป็น ทำให้การรับข้อมูลยังไม่รวดเร็วนัก

5.1.3 การรับส่งผ่านคลื่นวิทยุ

ในส่วนของการรับส่งผ่านคลื่นวิทยุนั้น สามารถส่งและรับข้อมูลได้จริง เนื่องจาก TRW 2.4 GHz มีประสิทธิภาพค่อนข้างสูง ในการใช้งานจะใช้ไฟเลี้ยง 3.3 โวลท์ โดยต้องใช้วงจรแปลงแรงดันและต้องระวังไม่ให้ไฟเลี้ยงเกินข้อจำกัด ไม่อย่างนั้นจะทำให้ TRW 2.4 GHz เกิดความเสียหายจนมาสามารถใช้งานได้ซึ่งทำให้สิ้นเปลืองเพราะมีราคาแพง

5.1.4 อีเทอร์เน็ตโมดูล (Embeded)

ในการบัดกรีวงจรซึ่งมีขนาดเล็ก ต้องมีความระมัดระวังเป็นอย่างมากเพราะจะเกิดความเสียหายได้ง่าย ในการติดต่อสื่อสารระหว่างอีเทอร์เน็ตโมดูลกับเครื่องคอมพิวเตอร์จะใช้โปรโตคอล UDP ซึ่งไม่มีการประกันความถูกต้องของข้อมูล อาจทำให้เกิดความผิดพลาดในข้อมูลที่รับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 บทสรุป

จากการที่ได้ศึกษาและทดลองเพื่อสร้างเครื่องอ่านบาร์โค้ดแบบไร้สายนั้น พบว่าวงจรและโปรแกรมที่สร้างนั้นสามารถส่งข้อมูลจากเครื่องอ่านบาร์โค้ดทางด้านส่งไปยังด้านรับแบบไร้สายแล้วส่งข้อมูลผ่านอีเทอร์เน็ตโมดูลซึ่งแสดงผลออกหน้าจอคอมพิวเตอร์ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1 ชัยมงคล วงษ์พานิช, เช็คข้อมูลสินค้าด้วยบาร์โค้ดแบบไร้สาย ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมบัณฑิต สาขาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2549

2 รัตน์ ตั้งพัฒนาประดิษฐ์,ระบบเครือข่ายควบคุมการใช้เครื่องใช้ไฟฟ้าด้วยบาร์โค้ด ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิคส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2535

3 สมยศ จุณณะปิยะ, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูลMCS-51 ,2550

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมทางด้านส่ง

```
;**********************************************************************
;           Program TRW2.4GHZ Tx Part
;**********************************************************************

CE     BIT    P2.0
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

            ORG 0000H

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

            LCALL INIT
            LCALL SETMODE_TRW24

            SETB   TR1

;**********************************************************************
;           MAIN PROGRAM
;**********************************************************************
MAIN:       MOV    R0,#031H
            MOV    R4,#0FH

RE_ED:             LCALL RECEIVE_BAR

            MOV    @R0,ACC
            INC    R0
            DJNZ   R4,RE_ED

            LCALL SEND_TRW

            LCALL DELAY_1s

            LJMP   MAIN

;**********************************************************************
;           RECEIVE DATA From BARCODE
;**********************************************************************
RECEIVE_BAR:       JNB    RI,$
            MOV    ACC,SBUF
            CLR    RI

            RET

;**********************************************************************
;           INITIAL TRW pin
;**********************************************************************
INIT:       CLR    CE
            CLR    CS
            CLR    DAT
            CLR    CLK

            RET

;**********************************************************************
;           Setmode TRW Config
;**********************************************************************
SETMODE_TRW24:  CLR     CE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SETB   CS
            CLR    A
            MOV    R1,#18
SETMODE_0:  MOV    DPTR,#CONFIG_TEST
            PUSH   ACC
            MOVC   A,@A+DPTR
            CALL   WRITE_TRW24
            POP    ACC
            INC    A
            DJNZ   R1,SETMODE_0
            SETB   DAT
            SETB   DR1
            SETB   CE
            CLR    CS

            RET

;*********************************************************************
;           SEND Data to TRW
;*********************************************************************
SEND_TRW:   MOV    R0,#031H
            MOV    R4,#0FH

            CLR    CS
            SETB   CE

            CLR    A
            MOV    R1,#5
SEND_TRW_0: MOV    DPTR,#CONFIG_ADDR1
            PUSH   ACC
            MOVC   A,@A+DPTR
            CALL   WRITE_TRW24
            POP    ACC
            INC    A
            DJNZ   R1,SEND_TRW_0

SEND_1:            MOV    ACC,@R0

            CALL   WRITE_DATA

            INC    R0
            DJNZ   R4,SEND_1

            CLR    CLK
            CLR    CE
            CLR    DAT

            RET

;*********************************************************************
;           Write 8 Bit to TRW pin
;*********************************************************************

WRITE_DATA:        MOV    R2,#8
WRITE_DATA_0:   JB        ACC.7,WRITE_DATA_1
            CLR    DAT
            JMP    WRITE_DATA_2
WRITE_DATA_1:      SETB   DAT
WRITE_DATA_2:      CALL   CLK_TRW
            RL     A
            DJNZ   R2,WRITE_DATA_0
            RET

;*********************************************************************
;           TRW Clock
;*********************************************************************
CLK_TRW:    CLR    CLK
            CALL   DELAY_1ms
            SETB   CLK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DAT     BIT     P2.4
DR1     BIT     P2.5
DOUT2   BIT     P2.6
DR2     BIT     P2.7

                ORG     0000H

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

                CALL    INIT
                CALL    SETMODE_RX

                SETB    TR1

;*******************************************************************
;               MAIN PROGRAM
;*******************************************************************
MAIN:           MOV     R4,#0FH

PRE_1           LCALL   RECIVE_TRW

                LCALL   TX_SERIAL

                DJNZ    R4,PRE_1

                SJMP    MAIN
;*******************************************************************
;               Get DATA From TRW
;*******************************************************************
RECIVE_TRW:     JNB     DR1,$
                CALL    READ_TRW24
                MOV     P0,A

                RET

;*******************************************************************
;               Send DATA to Serial
;*******************************************************************
TX_SERIAL:      MOV     SBUF,ACC
                JNB     TI,$
                CLR     TI

                RET

;*******************************************************************
;               INITIAL TRW pin
;*******************************************************************
INIT:           CLR     CE
                CLR     CS
                CLR     DAT
                CLR     CLK

                RET

;*******************************************************************
;               Function Config for TRW RX
;*******************************************************************
SETMODE_RX:     CLR     CE
                SETB    CS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CALL  DELAY_1ms

            RET

;*********************************************************************
;           Write DATA to TRW
;*********************************************************************
WRITE_TRW24:      MOV   R2,#8
WRITE_TRW24_0:  JB      ACC.7,WRITE1
            CLR   DAT
            JMP   WRITE_TRW2
WRITE1:           SETB  DAT
WRITE_TRW2:       CALL  CLK_TRW
            RL    A
            DJNZ  R2,WRITE_TRW24_0

            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:      DB 8EH,08H,1CH
CONFIG_LEN2:      DB 78H
CONFIG_LEN1:      DB 78H
CONFIG_ADDR2:     DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1:     DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR: DB 0A3H
CONFIG_RF:  DB 0CFH
CONFIG_CH:  DB 04H ;TX

            END
```

โปรแกรมทางด้านรับ

```
;*********************************************************************
;           Program TRW24GHZ Rx Part
;*********************************************************************

CE      BIT   P20
CLK2    BIT   P21
CS      BIT   P22
CLK     BIT   P23
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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

;**********************************************************************
;           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 DATA to TRW
;**********************************************************************
WRITE_TRW24 MOV     R2,#8
WRITE_TRW24_0 JB    ACC.7,WRITE1
            CLR     DAT
            JMP     WRITE_TRW2
WRITE1:     SETB    DAT
WRITE_TRW2  CALL    CLK_TRW
            RL      A
            DJNZ    R2,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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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 RX for TRW
;*********************************************************************

CONFIG_TEST_RX:      DB 8EH,08H,1CH
CONFIG_LEN2_RX:      DB 78H
CONFIG_LEN1_RX:      DB 78H
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 0CFH
CONFIG_CH_RX:              DB 05H ;RX

             END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแสดงผลหน้าจอคอมพิวเตอร์

```
import java.net.*;
import java.io.*;
//import java.net.*;
import java.awt.*;
import java.awt.event.*;

//Java extension packages
import javax.swing.*;

import java.sql.*;
import java.net.*;
class DgServer
{
      public static void main(String args[]) throws Exception
            {
                  DatagramSocket ds = new DatagramSocket(3000);
                  System.out.println("Server created.");
                  while (true)
                  {
                        DatagramPacket dp = new DatagramPacket(new
byte[100], 100);

                        ds.receive(dp);
                        String header = "\nFrom Host: " +
dp.getAddress() + "\nBarcode: ";
                        String msg = new String(dp.getData()).trim();

                        if (msg.equals ("quit"))
                              break;
                        System.out.println();
                        System.out.println();
                        System.out.println();
                        System.out.println(header + msg);

                  }
                  ds.close();
            }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้