สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมประตูโดยใช้บัตร RFID

DOOR CONTROL SYSTEM BY RFID CARD

โดย

นายกษิดิศ เจริญศรีวิไลวัฒน์

นายแทนเทพ เจติยานุวัตร

ปพ.
ก3335
2550

เลขหมู่ 83292
เลขทะเบียน 11 ส.ค. 2551
วัน,เดือน,ปี

b. 119 62394
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมประตูโดยใช้บัตร RFID

DOOR CONTROL SYSTEM BY RFID CARD

โดย

นายกษิดิศ เจริญศรีวิไลวัฒน์ 47010028

นายแทนเทพ เจติยานุวัตร 47010294

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 **ระบบควบคุมประตูโดยบัตร**

DOOR CONTROL SYSTEM BY RFID CARD

ผู้จัดทำ

1. นายกษิดิศ เจริญศรีวิไลวัฒน์ 47010028

2.นายแทนเทพ เจติยานุวัตร 47010294

.... อาจารย์ที่ปรึกษา

(รศ.ดร. ปราโมทย์ วาดเขียน)

.... อาจารย์ที่ปรึกษา

(ผศ.ดรจีรสุดา โกษียาภรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมประตูโดยบัตร RFID

DOOR CONTROL SYSTEM BY RFID CARD

โดย นายกษิดิศ เจริญศรีวิไลวัฒน์ 47010028

นายแทนเทพ เจติยานุวัตร 47010294

อาจารย์ที่ปรึกษา รศ.ดร. ปราโมทย์ วาดเขียน

ผศ.ดร. จีรสุดา โกษียาภรณ์

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการนำเสนอ ระบบปิด – เปิดประตูโดยใช้ คีย์การ์ดแบบ RFID ซึ่งจะแบ่งออกเป็น 2 ส่วนคือ ส่วนควบคุมการเปิด – ปิดประตู และส่วนตรวจสอบข้อมูล ในส่วนควบคุมการเปิด - ปิดประตูนั้นจะมีไมโครคอนโทรลเลอร์ทำหน้าที่ส่งข้อมูลบัตร RFID ผ่านโมดูล RF TRW 2.4 GHz ไปให้ส่วนตรวจสอบหรือทำการเปิด - ปิด ประตู ตามสัญญาณควบคุมที่จะได้รับมาจากภาคตรวจสอบผ่านโมดูล RFID

สำหรับภาคตรวจสอบ เมื่อไมโครคอนโทรเลอร์รับข้อมูลบัตร RFID จากโมดูล RF จะส่งไปยังโมดูล Embedded เพื่อส่งผ่านไปยังระบบ LAN เข้าสู่คอมพิวเตอร์ส่วนกลางทำการตรวจสอบข้อมูล ข้อมูลควบคุมการเปิด – ปิดจะส่งกลับไปให้ส่วนควบคุมการเปิด – ปิด ดังที่ได้กล่าวมาแล้ว นอกจากนี้ข้อมูลการเข้า – ออก ของผู้ใช้จะถูกตรวจสอบกับฐานข้อมูล

ABSTRACT

This project presents door control system by RFID card, composes of two parts: a door controlling part and a data checking part. In the door controlling part, a microcontroller will either send the information of a RFID card through the RF module to the data checking part or open / lock the door according to the received controlling signal obtained from the data checking part. At the data checking part, the microcontroller will either send the coming RFID information through the embedded module and LAN to the main computer for checking or send out the controlling signal to the door controlling part.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณรองศาสตราจารย์ ดร. ปราโมทย์ วาดเขียน และผู้ช่วยศาสตราจารย์ ดร.จีรสุดา โกษียาภรณ์ ที่ช่วยดูแลเอาใจใส่ คอยให้คำแนะนำ ชี้แจงข้อบกพร่องต่างๆ และเป็นที่ปรึกษาการทำโครงงาน ตลอดจนช่วยดูแลและจัดให้มีการตรวจความคืบหน้าของโปรเจคอยู่เป็นประจำ ทำให้ผลงานสำเร็จออกมาได้ดีมีคุณภาพทันเวลาที่กำหนดส่ง รวมทั้งขอขอบคุณรุ่นพี่อีกหลายคน ในการช่วยเหลือให้ข้อมูลต่างๆ และขอขอบคุณเพื่อนๆที่ช่วยเหลือตลอดมา จึงทำให้สามารถจัดทำรายงานนี้ขึ้นได้อย่างสมบูรณ์

(นายกษิดิศ เจริญศรีวิไลวัฒน์)

(นายแทนเทพ เจติยานุวัตร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	i
สารบัญ	ii
สารบัญรูป	v
สารบัญตาราง	vii
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการศึกษา	1
บทที่ 2 ทฤษฎีและหลักการ	
2.1 RFID	2
2.1.1 ส่วนประกอบของ RFID	2
2.1.2 ป้าย	2
2.1.2.1 ป้ายแบบแพสซิฟ	5
2.1.2.2 ป้ายแบบกึ่งแพสซิฟ	6
2.1.2.3 ป้ายแบบแอ็กทิฟ	7
2.1.3 เครื่องอ่าน	7
2.1.4 หลักการรับและส่งข้อมูลระหว่างป้ายและเครื่องอ่าน	10
2.1.5 RFID ในคลื่นความถี่ต่างๆ	12
2.2 ไมโครคอนโทรลเลอร์	14
2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51	14
2.2.2 ลักษณะเด่นของ MCS – 51	15
2.2.2.1 หน่วยความจำภายใน (Internal Memory)	15
2.2.2.2 การติดต่อและควบคุมอุปกรณ์ภายนอก	18
2.2.3 รีจิสเตอร์ที่ใช้ในการเขียนโปรแกรม	18
2.2.3.1 IE: อินเตอร์รัพต์ อีนาเบิล รีจิสเตอร์	18
2.2.3.2 IP: อินเตอร์รัพต์ ไพออริตี้ รีจิสเตอร์	18
2.2.3.3 TCON: ไทเมอร์/เคาน์เตอร์ คอนโทรล รีจิสเตอร์	18
2.2.3.4 TMOD: ไทเมอร์/เคาน์เตอร์ โหมด คอนโทรล รีจิสเตอร์	19
2.3 RF- MODULE TRW 2.4 GHz	19
2.3.1 ลักษณะของ RF- MODULE TRW 2.4 GHz	19
2.3.2 การประยุกต์ใช้	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.3.3 ลักษณะการจัดขาของโมดูล	21
2.3.4 โหมดการทำงาน	21
2.3.4.1 การส่งข้อมูลโหมดการทำงานแบบ ShockBurst	22
2.3.4.2 หลักการทำงานในโหมด Dircect	26
2.3.5 องค์ประกอบของชุดเฟรมข้อมูล	26
2.3.6 ตำแหน่งบิตข้อมูลของโมดูลความถี่วิทยุ TRW-2.4 GHz	28
2.3.7 การตั้งค่าในโหมด ShockBurst	28
2.3.7.1 การตั้งค่าเกี่ยวกับอุปกรณ์	30
2.3.7.2 การตั้งค่าความถี่ช่องสัญญาณและทิศทางการส่งข้อมูล	32
2.4 TCP/IP โปรโตคอล	33
2.4.1 Encapsulation/Demultiplexing	33
2.4.2 ชั้นโฮสต์-เครือข่าย	35
2.4.3 ชั้นสื่อสารอินเทอร์เน็ต	35
2.4.4 ชั้นสื่อสารนำส่งข้อมูล	38
2.4.5 ชั้นสื่อสารการประยุกต์	42
2.4.6 ระบบของ Ethernet Frame	44
2.4.7 ข้อกำหนดเกี่ยวกับขนาดของ Data Frame	44
2.5 ระบบฝังตัว	45
2.5.1 สถาปัตยกรรม Vortex86	45
บทที่ 3 การคำนวณและการออกแบบ	
3.1 วงจรรับข้อมูลจาก RFID CARD	47
3.2 วงจรรับข้อมูลที่ส่งมาจากโมดูล TRW	51
3.3 วงจร Ethernet Controller	53
3.4 ส่วนการแสดงข้อมูลด้วยโปรแกรม	58
3.5 ส่วนควบคุมการปิด – เปิดประตู	61
บทที่4 การทดลองและผลการทดลอง	
4.1 การทดลองส่งข้อมูลผ่านโมดูล TRW 2.4GHz ด้วยเลขฐาน 2 จำนวน 8 บิต	63
4.2 การทดลองนำข้อมูลจากบัตร RFID ส่งผ่านโมดูล TRW 2.4 GHz	65
4.3 การทดลองนำบัตร RFID ส่งผ่านโมดูล TRW 2.4 GHz เข้าคอมพิวเตอร์	66
4.4 การทดลองนำข้อมูลส่งผ่านโมดูล Embedded เข้าคอมพิวเตอร์	67
4.5 การทดลองนำข้อมูลจากวงจรInternet Controllerเข้าสู่คอมพิวเตอร์แสดงด้วยโปรแกรม	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

4.6 การทดลองนำข้อมูลจากบัตร RFID ผ่านวงจร Ethernet Controller เข้าสู่คอมพิวเตอร์แสดงด้วยโปรแกรม 69

4.7 การทดลองการเปิด – ปิดประตู 71

บทที่ 5 บทวิจารณ์และสรุปผล

5.1 บทวิจารณ์ 74

5.1.1 ส่วนของเครื่องอ่าน RFID 74

5.1.1 ส่วนของเครื่องอ่าน RFID 74

5.1.3 ส่วนของไมโครคอนโทรลเลอร์ 74

5.1.4 ส่วนของ Embedded Module 74

5.1.5 ส่วนของโปรแกรมคอมพิวเตอร์ 74

5.2 บทสรุป 74

บรรณานุกรม 75

ภาคผนวก 76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 ระบบ RFID 2
รูปที่ 2.2 องค์ประกอบทั่วไปของป้าย 3
รูปที่ 2.3 ป้ายในรูปแบบชนิดต่างๆ 4
รูปที่ 2.4 สถาปัตยกรรมภายในไมโครชิพของป้ายแบบแพสซิฟ 5
รูปที่ 2.5 สนามแม่เหล็กจากกระบวนการคู่ควบเหนี่ยวนำ 6
รูปที่ 2.6 หลักการทำงานของ LF, HF และ UHF 6
รูปที่ 2.7 ป้ายแบบแอ็กทิฟ ที่มีแบตเตอรี่ลิเธียม 2 ก้อนอยู่ภายนอก 7
รูปที่ 2.8 โครงสร้างภายในเครื่องอ่าน RFID 8
รูปที่ 2.9 เครื่องอ่าน RFID แบบพกพา 8
รูปที่ 2.10 เครื่องอ่าน RFID แบบติดผนัง 9
รูปที่ 2.11 เครื่องอ่าน RFID แบบอุโมงค์ 9
รูปที่ 2.12 เครื่องอ่าน RFID แบบประตู 10
รูปที่ 2.13 ตัวอย่างการเข้ารหัสแบบต่างๆ 11
รูปที่ 2.14 เครื่องอ่านทำงานกับป้ายหลายๆป้ายพร้อมๆกัน 12
รูปที่ 2.15 ตัวอย่างของอัลกอริทึมในการป้องกันการชนกันของข้อมูลในป้าย 12
รูปที่ 2.16 แสดงความถี่ย่านที่ระบบ RFID ถูกใช้งาน 13
รูปที่ 2.17 ตำแหน่งขา MCS-51 15
รูปที่ 2.18 แสดงส่วนต่าง ๆ ของหน่วยความจำภายในไมโครคอนโทรลเลอร์ 16
รูปที่ 2.19 อินเตอร์รัพต์ อีนาเบิล รีจิสเตอร์ 18
รูปที่ 2.20 อินเตอร์รัพต์ ไพออริตี้ รีจิสเตอร์ 18
รูปที่ 2.21 ไทเมอร์/เคาเตอร์ คอนโทรลรีจิสเตอร์ 18
รูปที่ 2.22 ไทเมอร์/เคาน์เตอร์ โหมดคอนโทรลรีจิสเตอร์ 19
รูปที่ 2.23 แสดงมุมมองของModule TRW 2.4 GHz 19
รูปที่ 2.24 แสดง โมดูล TRW 2.4GHz 21
รูปที่ 2.25 nRF2401 ต่อกับอุปกรณ์ภายนอก 22
รูปที่ 2.26 แสดงลำดับการส่งข้อมูลด้วยวิธี ShockBurst 23
รูปที่ 2.27 การใช้กระแส RF โดยใช้และไม่ใช้เทคโนโลยี Shock Burst 23
รูปที่ 2.28 Flow Chart ShockBurst การส่งของ nRF2401 24
รูปที่ 2.29 Flow Cart การทำงานโหมดShockBurst ทางด้านรับ 25
รูปที่ 2.30 แสดงลำดับเฟรมข้อมูล 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 2.31 จำนวนของบิตใน payload 29

รูปที่ 2.32 แอดเดรสของตัวรับ 1 และ 2 29

รูปที่ 2.33 จำนวนบิตที่ต้องจองไว้สำหรับ RX address + CRC setting 30

รูปที่ 2.34 การเซ็ตการทำงานของโมดูลความถี่วิทยุ 30

รูปที่ 2.35 ช่องความถี่ และ การกำหนดโหมดการทำงาน Rx / Tx 32

รูปที่ 2.36 ตัวอย่างการกำหนดค่าความถี่ให้กับตัวส่ง ความถี่ 2500 เมกะเฮิรตซ์ 32

รูปที่ 2.37 ตัวอย่างการกำหนดค่าความถี่ให้กับตัวรับ ความถี่ 2500 เมกะเฮิรตซ์ 32

รูปที่ 2.38 ขั้นตอนการ Encapsulation และ Demultiplexing 33

รูปที่ 2.39 โครงสร้าง TCP/IP 34

รูปที่ 2.40 เปรียบเทียบ OSI Model กับ TCP/IP Model 34

รูปที่ 2.41 แสดง IP Header 36

รูปที่ 2.42 ICMP Header 38

รูปที่ 2.43 UPD Header 39

รูปที่ 2.44 TCP Header 41

รูปที่ 2.45 แสดงโปรโตคอลที่ใช้งานในโมเดล TCP/IP 44

รูปที่ 2.46 ลักษณะโครงสร้างของเฟรมข้อมูล 44

รูปที่ 2.47 สถาปัตยกรรม Vortex 86 46

รูปที่ 3.1 แสดงบล็อกไดอะแกรมของโครงงานทั้งหมด 47

รูปที่ 3.2 วงจรรับข้อมูลจาก RFID แล้วส่งข้อมูลต่อผ่าน TRW 48

รูปที่ 3.3 แสดงโฟล์วชาร์ตการทำงานของเครื่องอ่าน RFID กับไมโครคอนโทรลเลอร์ 49

รูปที่ 3.4 แสดงโฟล์วชาร์ดการทำงานของโมดูล TRW กับไมโครคอนโทรลเลอร์ทางด้านส่ง 50

รูปที่ 3.5 แสดงวงจรรับข้อมูล Wireless ผ่านโมดูล TRW 2.4GHz 51

รูปที่ 3.6 แสดงโฟล์วชาร์ดการทำงานของโมดูล TRWกับไมโครคอนโทรลเลอร์ทางด้านรับ 52

รูปที่ 3.7 แสดงการเชื่อมต่อของไอซี CS8900A กับ RJ-45ในวงจร Internet Controller 53

รูปที่ 3.8 แสดงการ Encapsulation ข้อมูลที่ใช้เขียนในโปรแกรม 54

รูปที่ 3.9 แสดงการ Decapsulation ข้อมูลที่ฝั่งคอมพิวเตอร์ 55

รูปที่ 3.10 วงจร Internet Controller เชื่อมต่อกับไมโครคอนโทรลเลอร์ 56

รูปที่ 3.11 แสดงโฟล์วชาร์ตการทำงานของ Internet Controller 57

รูปที่ 3.12 แสดงโฟล์วชาร์ตการทำงานของภาคแสดงบนคอมพิวเตอร์ด้วยโปรแกรม 60

รูปที่ 3.13 แสดงวงจรควบคุมการปิด – เปิดประตู 61

รูปที่ 3.14 แสดงโฟล์วชาร์ตการทำงานของส่วนควบคุมประตู 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 4.1 แสดงการวัดข้อมูลด้านส่ง ch 1 ที่ขา clock ch 2 ที่ขา data 63

รูปที่ 4.2 แสดงการวัดข้อมูลด้านรับระหว่างขา ch 1 ที่ขา clock ch 2 ที่ขา DR1 64

รูปที่ 4.3 แสดงการวัดข้อมูลด้านรับระหว่างขา ch 1 ที่ขา clock ch 2 ที่ขา data ซึ่งมีข้อมูลเป็น 10111010 64

รูปที่ 4.4 แสดงสัญญาณด้านส่ง ch 1 ที่ขา data, ch 2 ที่ขา clock 65

รูปที่ 4.5 แสดงสัญญาณด้านรับ ch 1 ที่ขา data, ch 2 ที่ขา clock 66

รูปที่ 4.6 แสดงข้อมูลจากบัตร RFID ผ่านคอมพิวเตอร์ 67

รูปที่ 4.7 แสดงการส่งข้อมูลระหว่างวงจร Embedded Module และคอมพิวเตอร์ 68

รูปที่ 4.8 แสดงการส่งข้อมูลจากวงจรInternet Controllerเข้าคอมพิวเตอร์ด้วยโปรแกรมจาวา 69

รูปที่ 4.9 แสดงข้อมูลจากบัตร RFID ตรวจสอบด้วยซอฟท์แวร์ Wireshark 70

รูปที่ 4.10 แสดงข้อมูลจากบัตร RFID ตรวจสอบด้วยซอฟท์แวร์ Wiresharkและโปรแกรม 70

รูปที่ 4.11 แสดงสัญญาณจาก RS232 มีค่า 00110001 71

รูปที่ 4.12 แสดงสัญญาณเมื่อประตูปิด 72

รูปที่ 4.13 แสดงสัญญาณเมื่อประตูเปิด 73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 ระยะที่สามารถอ่านได้ของแต่ละย่านความถี่ 13

ตารางที่ 2.2 แสดงข้อแตกต่างของ RFID ระบบต่างๆ 14

ตารางที่ 2.3 แสดงบิตต่างๆภายในรีจิสเตอร์ PSW 16

ตารางที่ 2.4 แสดง SFR ของ MCS - 51 17

ตารางที่ 2.5 การเลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์โดยใช้ M0,M1 19

ตารางที่ 2.6 ความเร็วในการรับ-ส่งข้อมูล 20

ตารางที่ 2.7 แสดงรายละเอียดการทำงานของโมดูลแต่ละขา 21

ตารางที่ 2.8 โหมดการทำงานหลักnRF2401 22

ตารางที่ 2.9 แสดงองค์ประกอบของชุดเฟรมข้อมูล 27

ตารางที่ 2.10 แสดงรายละเอียดของตำแหน่งบิตข้อมูลในโมดูลความถี่วิทยุ 28

ตารางที่ 2.11 PLL setting 28

ตารางที่ 2.12 การเซ็ตความถี่คริสตอล 31

ตารางที่ 2.13 RF การเซ็ตกำลังเอาท์พุท 31

ตารางที่ 2.14 แสดง UDP Port Number 40

ตารางที่ 2.15 Flag ชนิดต่างๆ 42

ตารางที่ 3.1 แสดงฐานข้อมูล 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีการระบุหรือชี้เฉพาะโดยใช้คลื่นวิทยุ หรืออาร์เอฟไอดี (RFID: Radio Frequency Identification) เป็นเทคโนโลยีที่เข้ามามีบทบาทต่อการดำเนินชีวิตของผู้คนในแต่ละวัน และได้มีการนำมาประยุกต์ใช้งานในหลากหลายด้าน เช่น การควบคุมการเข้า-ออก/บัตรประจำตัว เป็นระบบการรักษาความปลอดภัยการเข้า-ออกอาคาร แทนการใช้บัตรแบบแม่เหล็ก เนื่องจากบัตรแถบแม่เหล็กเมื่อมีการใช้งานเป็นเวลานานๆ จะมีโอกาสเกิดการชำรุดของบัตรสูง แต่บัตรแบบ RFID ใช้เพียงการสัมผัสหรือแสดงผ่านหน้าเครื่องอ่านเท่านั้น ทั้งนี้ยังทำให้เกิดความสะดวกสบาย นอกจากนี้ยังสามารถระบุได้ว่ามีการเข้า-ออกของบุคคลากรที่ได้รับอนุญาต และยังป้องกันไม่ให้บุคคลภายนอกสามารถเข้ามาได้ด้วย

1.2 วัตถุประสงค์ของการศึกษา

- เพื่อศึกษาถึงกระบวนการทำงานของระบบบัตร RFID ซึ่งเป็นบัตรอัตโนมัติ (Auto ID) ชนิดหนึ่ง
- เพื่อนำไปประยุกต์ใช้ในการพัฒนากับระบบความปลอดภัยภายในหน่วยงานต่างๆ
- เพื่อศึกษาการเชื่อมต่อระบบ RFID กับฐานข้อมูลให้ทำงานร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 RFID

2.1.1 ส่วนประกอบของ RFID

ระบบ RFID ประกอบด้วย 3 ส่วน ดังรูปที่ 2.1 กล่าวคือ

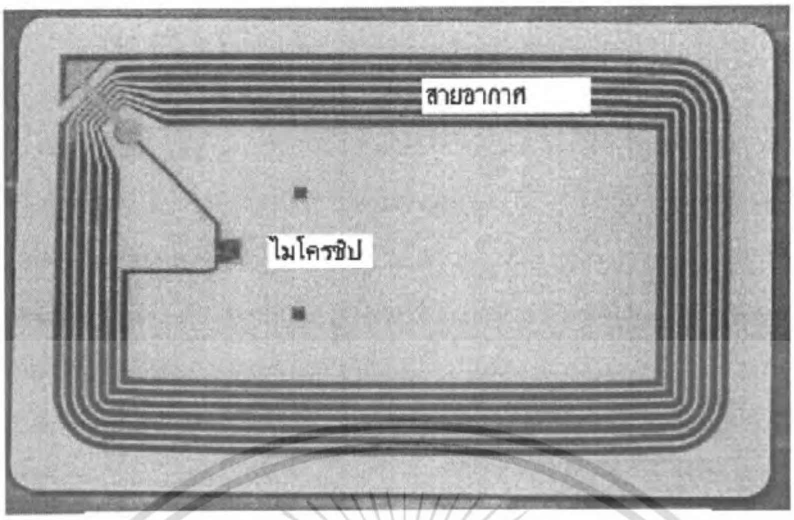
- ทรานสปอนเดอร์ หรือ ป้าย (Transponder/Tag) ที่ส่วนที่ติดอยู่กับวัตถุต่างๆที่เราต้องการ ส่วนประกอบในป้ายนั้นประกอบด้วยสายอากาศและไมโครชิพที่มีการบันทึกของมูลของวัตถุนั้นๆ อยู่
- เครื่องสำหรับอ่าน/เขียน ข้อมูลภายในป้าย (Interrogator/Reader) ใช้ความถี่คลื่นวิทยุในการอ่าน/เขียน ซึ่งทำให้สามารถอ่านรหัสจากป้ายได้โดยไม่ต้องเห็นป้าย และไม่จำเป็นจะต้องอยู่ในแนวเส้นตรงกับคลื่น เพียงแต่ให้อยู่ในรัศมีที่สามารถรับคลื่นวิทยุได้ก็สามารถอ่านข้อมูลได้ ทั้งยังสามารถอ่านได้หลายๆป้ายในเวลาเดียวกันอีกด้วย
- ระบบประยุกต์ใช้งาน

รูปที่ 2.1 ระบบ RFID

2.1.2 ป้าย

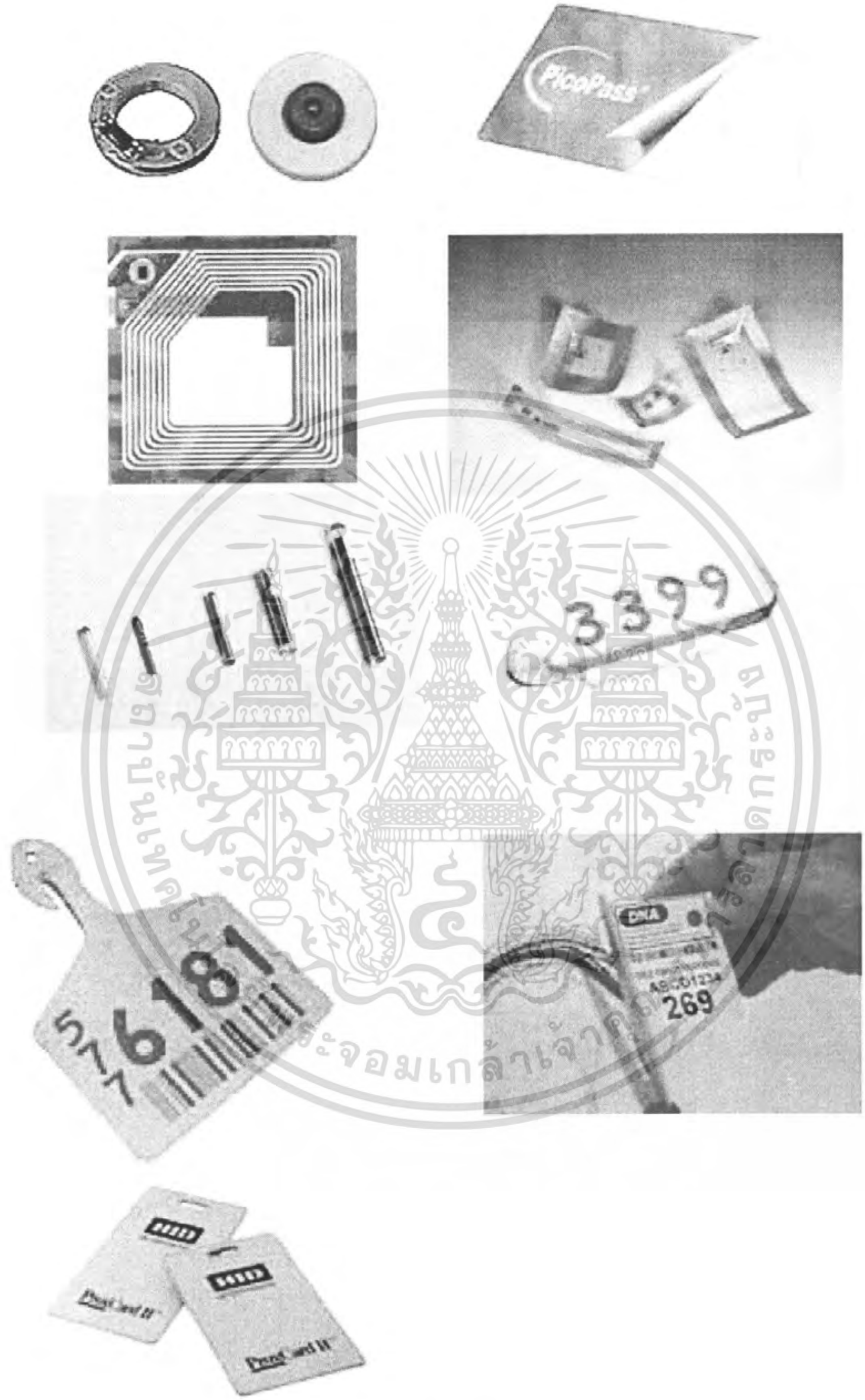
โดยทั่วไปป้ายจะอยู่ได้ในหลายรูปแบบ ไม่ว่าจะเป็น กระดาษ แผ่นฟิล์ม พลาสติก โดยมีขนาดและรูปร่างแตกต่างกันไป โครงสร้างภายในของป้ายนั้นจะประกอบด้วย 2 ส่วนใหญ่ๆ ได้แก่ ส่วนของไมโครชิพ จะทำหน้าที่เก็บข้อมูลของสินค้า และมีขดลวดขนาดเล็กทำหน้าที่เป็นสายอากาศ (Antenna) สำหรับรับ-ส่งสัญญาณความถี่คลื่นวิทยุและสร้างพลังงานป้อนให้กับไมโครชิพ ดังรูปที่ 2.2 และ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 องค์ประกอบทั่วไปของป้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ป้ายในรูปแบบชนิดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

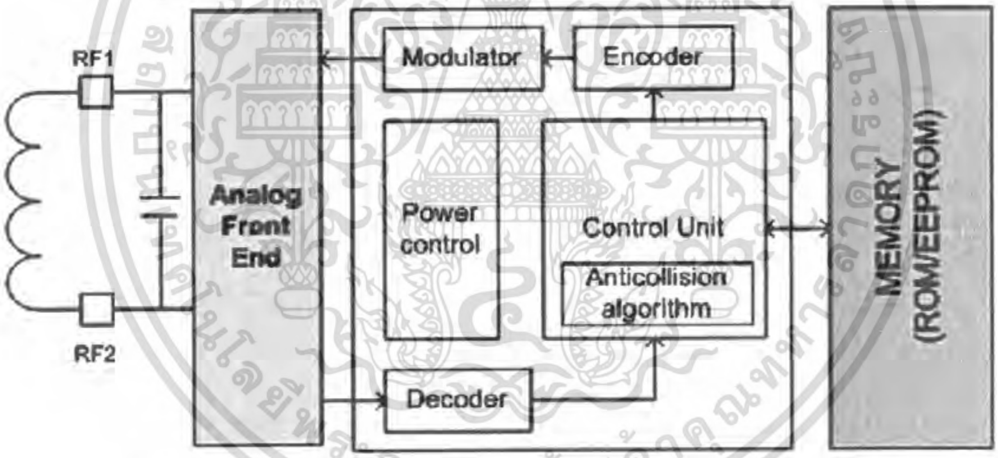
นอกจากนี้สามารถแบ่งป้ายที่มีการใช้งานกันอยู่นั้นได้เป็น 3 ชนิด ได้แก่ ป้ายแบบแพสซิฟ ป้ายแบบกึ่งแพสซิฟ และป้ายแบบแอ็กทิฟ

2.1.2.1 ป้ายแบบแพสซิฟ

ป้ายจะมีวงจรกำเนิดไฟฟ้าเหนี่ยวนำขนาดเล็กอยู่ภายใน ทำให้สามารถทำงานได้โดยไม่ต้องอาศัยแหล่งจ่ายไฟจากภายนอก ดังนั้นในการอ่านข้อมูลจึงสามารถอ่านได้ในระยะทางใกล้ๆ ระยะอ่านสูงสุดประมาณ1 เมตร โดยขึ้นกับกำลังงานของเครื่องส่งและความถี่วิทยุที่ใช้ มีหน่วยความจำขนาดเล็กประมาณ 16-1024 ไบต์

ไมโครชิพของป้ายแบบแพสซิฟจะมีขนาดตั้งแต่เล็กมากๆจนถึงขนาดใหญ่มากๆ ขึ้นอยู่กับความเหมาะสมในการใช้งาน ซึ่งโดยทั่วไปโครงสร้างภายในของไมโครชิพจะประกอบด้วย 3 ส่วนหลักๆ ดังรูปที่ 2.4 ได้แก่

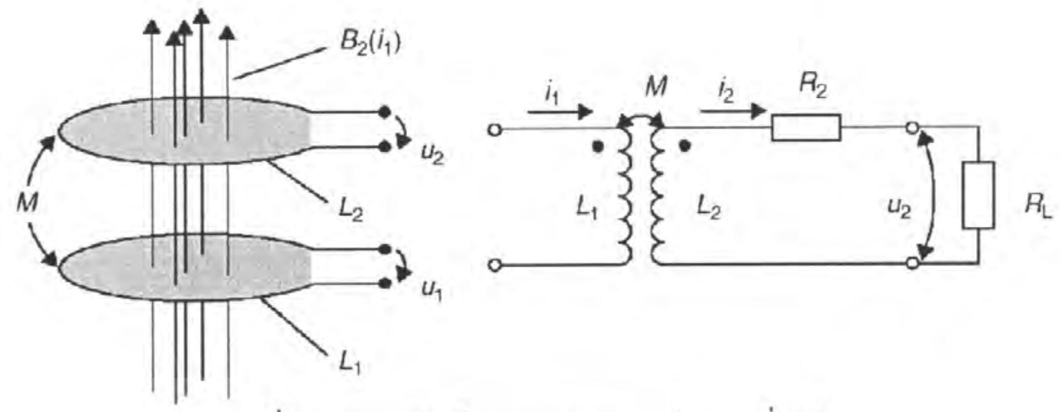
- ส่วนควบคุมการทำงานของภาครับ-ส่งสัญญาณวิทยุ (Analog-Front-End)
- ส่วนควบคุมภาคโลจิก (Digital Control Unit)
- ส่วนหน่วยความจำ อาจเป็นแบบ ROM หรือ EEPROM



รูปที่ 2.4 สถาปัตยกรรมภายในไมโครชิพของป้ายแบบแพสซิฟ

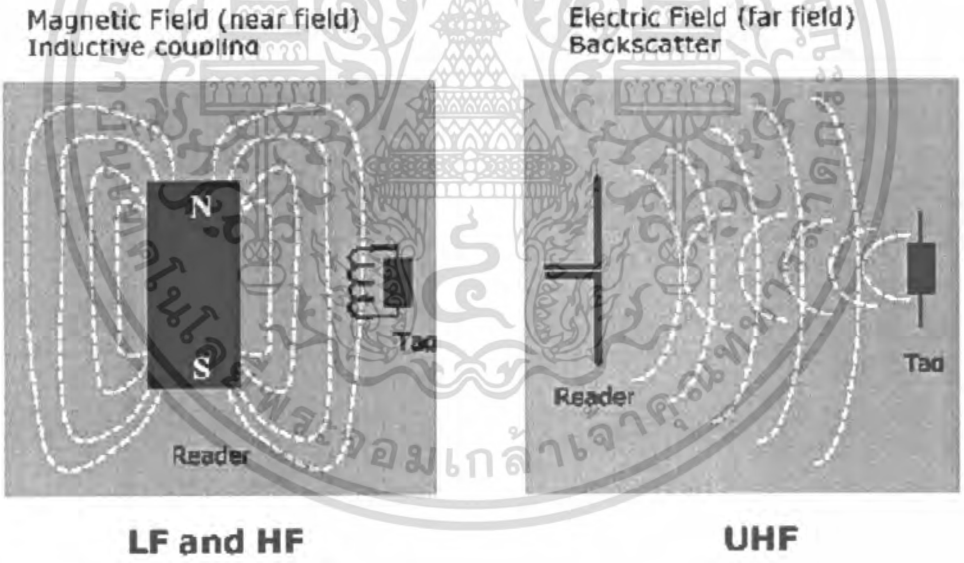
การทำงานของป้าย RFID แบบแพสซิฟโดยทั่วไปทำงานในย่านความถี่ต่ำและสูง (LF และ HF) ใช้หลักการคู่ควบแบบเหนี่ยวนำ (Inductive Coupling) เกิดจากการที่สายอากาศของป้ายมาอยู่ใกล้กับขดลวดของเครื่องอ่านที่กำลังทำงานอยู่ ทำให้เกิดการถ่ายเทพลังงานผ่านสนามแม่เหล็กไฟฟ้าที่เกิดขึ้นจากเครื่องอ่านไปยังไมโครชิพในป้ายไมโครชิพก็จะทำงานตามลักษณะเฉพาะของข้อมูลรหัสประจำตัว เครื่องอ่านก็ทำให้รู้ปฏิกิริยาของไมโครชิพผ่านทางสนามแม่เหล็ก แล้วทำการแปลงให้เป็นสัญญาณดิจิตอลก็จะทำให้ข้อมูลเฉพาะตัวของป้ายมาได้ ดังแสดงในรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 สนามแม่เหล็กจากกระบวนการคู่ควบเหนี่ยวนำ

ส่วนในย่านความถี่สูงยิ่ง (UHF) จะใช้การคู่ควบแบบแผ่กระจาย (Propagation Coupling) โดยที่สายอากาศของเครื่องอ่านจะส่งสัญญาณคลื่นวิทยุออกมา เมื่อป้ายได้รับสัญญาณผ่านทางสายอากาศของป้ายก็จะทำงานสะท้อนสัญญาณกลับไปตามรหัสประจำตัวของตัวเองกลับไปยังเครื่องอ่าน (Backscattering) ดังรูปที่ 2.6



รูปที่ 2.6 หลักการทำงานของ LF, HF และ UHF

2.1.2.2 ป้ายแบบกึ่งแพสซิฟ

ป้ายชนิดนี้อาศัยแหล่งจ่ายไฟจากภายนอก ทำให้สามารถส่งข้อมูลได้ระยะไกลกว่าแบบแพสซิฟ ตัวป้ายจะรอรับสัญญาณกระตุ้นให้ทำงานจากเครื่องอ่านจึงไม่สามารถเป็นผู้เริ่มต้นส่งสัญญาณการสื่อสารได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

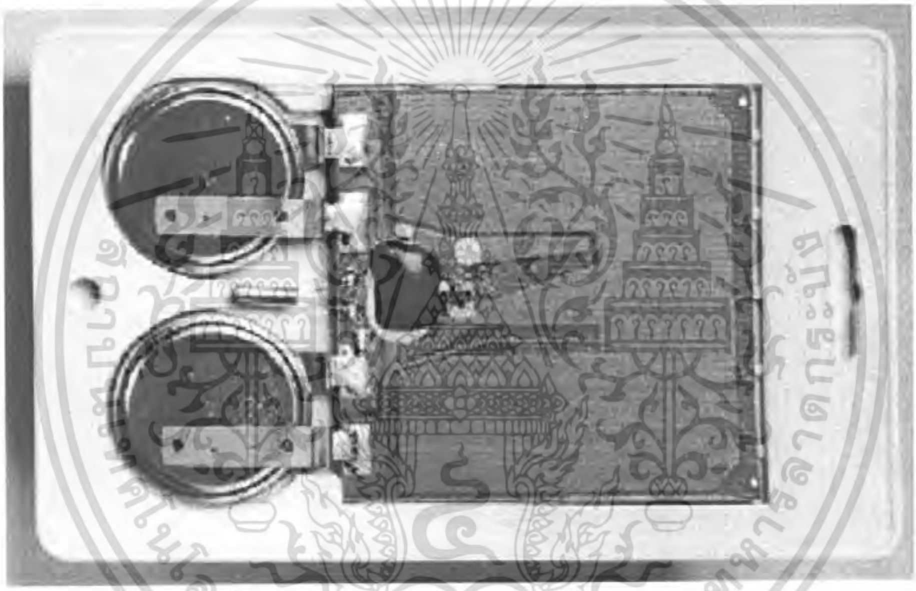
2.1.2.3 ป้ายแบบแอ็กทิฟ

ป้ายชนิดนี้อาศัยแหล่งจ่ายไฟจากภายนอก เพื่อจ่ายให้กับวงจรการทำงานภายใน ทำให้สามารถอ่านได้ในระยะไกลสูงสุดประมาณ 100เมตร มีหน่วยความจำภายในขนาดใหญ่ได้ถึง 1 เมกะไบต์ ทำให้ป้ายชนิดนี้มีราคาสูง ขนาดค่อนข้างใหญ่ และมีอายุการใช้งานจำกัดตามอายุของแบตเตอรี่

ตามหลักการทำงานแล้วแบ่งออกได้เป็น 2 แบบ

-แบบทรานสปอนเดอร์แบบแอ็กทิฟ ซึ่งจะทำการส่งข้อมูลออกก็ต่อเมื่อได้รับสัญญาณจากเครื่องอ่าน พบการใช้งานแบบนี้ได้ใน ระบบจ่ายเงินบนทางด่วน หรือด่านตรวจ

-แบบเครื่องบอกตำแหน่งหรือเบคอน (Beacon) ซึ่งสัญญาณจะถูกปล่อยออกมาเป็นระยะๆ ตลอดเวลา พบได้ในระบบที่ต้องการบ่งชี้พิกัดแบบเวลาจริง (Real-time Locating System: RTLS) เช่น การจัดการการขนส่งสินค้า



รูปที่ 2.7 ป้ายแบบแอกทิฟ ที่มีแบตเตอรี่ลิเธียม 2 ก้อนอยู่ภายนอก

นอกจากการแบ่งชนิดของป้ายต่างๆตามที่ได้กล่าวมาแล้วนั้น เรายังสามารถแบ่งป้ายตามรูปแบบการอ่านหรือบันทึกข้อมูลได้เป็น 3 แบบ คือ ป้ายชนิดที่ถูกอ่านหรือเขียนข้อมูลได้หลายครั้ง (Read-Write) ป้ายชนิดที่เขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้หลายครั้ง (Write-Once Read-Many หรือ WORM) และป้ายชนิดอ่านได้เพียงอย่างเดียว (Read-Only) หรือเรายังสามารถแบ่งชนิดตามความถี่การใช้งาน เช่น ป้ายย่านความถี่ต่ำ 125-134 กิโลเฮิรตซ์ ป้ายย่านความถี่สูง 13.56 เมกะเฮิรตซ์ ป้ายย่านความถี่สูงยิ่ง 433 และ 900 เมกะเฮิรตซ์ และป้ายย่านไมโครเวฟ 2.4 กิกะเฮิรตซ์

2.1.3 เครื่องอ่าน

เครื่องอ่านมีหน้าที่อ่านหรือเขียนข้อมูลลงในป้ายด้วยสัญญาณความถี่วิทยุ ภายในเครื่องอ่านจะประกอบไปด้วยส่วนประกอบหลักดังแสดงในรูปที่ 2.8 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภาครับและภาคส่งสัญญาณวิทยุ

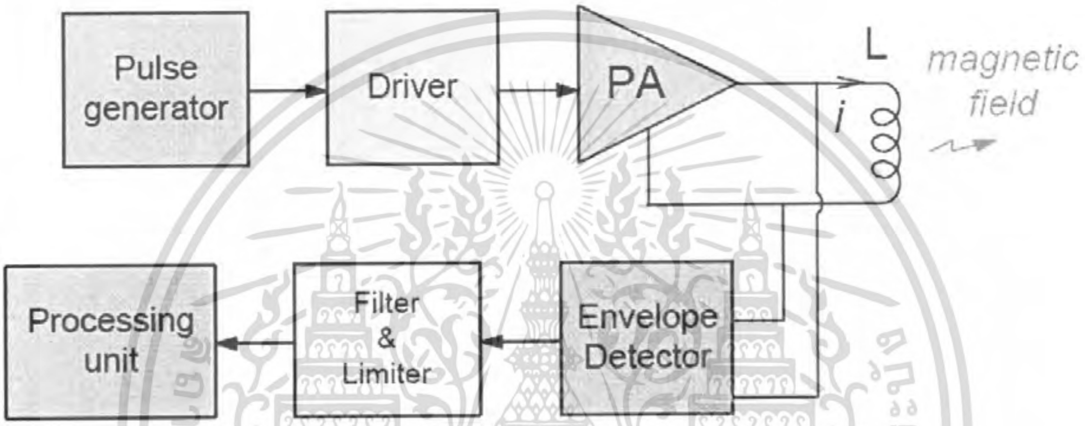
- ภาคสร้างสัญญาณพาหะ

- สายอากาศ ทำจากขดลวดทองแดง

- วงจรจูนสัญญาณ

- หน่วยประมวลผลข้อมูล และภาคติดต่อกับคอมพิวเตอร์ หน่วยประมวลผลโดยทั่วไปมักจะใช้วงจรไมโครคอนโทรลเลอร์ ซึ่งอัลกอริทึมที่อยู่ภายในโปรแกรมจะทำหน้าที่ถอดรหัสข้อมูลที่ได้รับ

โดยลักษณะ ขนาด และรูปร่างของเครื่องอ่านจะแตกต่างกันไปตามลักษณะการใช้งาน เช่น แบบมือถือขนาดเล็ก จนถึงขนาดใหญ่เท่าประตู ดังตัวอย่างแสดงในรูปที่ 2.9 – 2.12



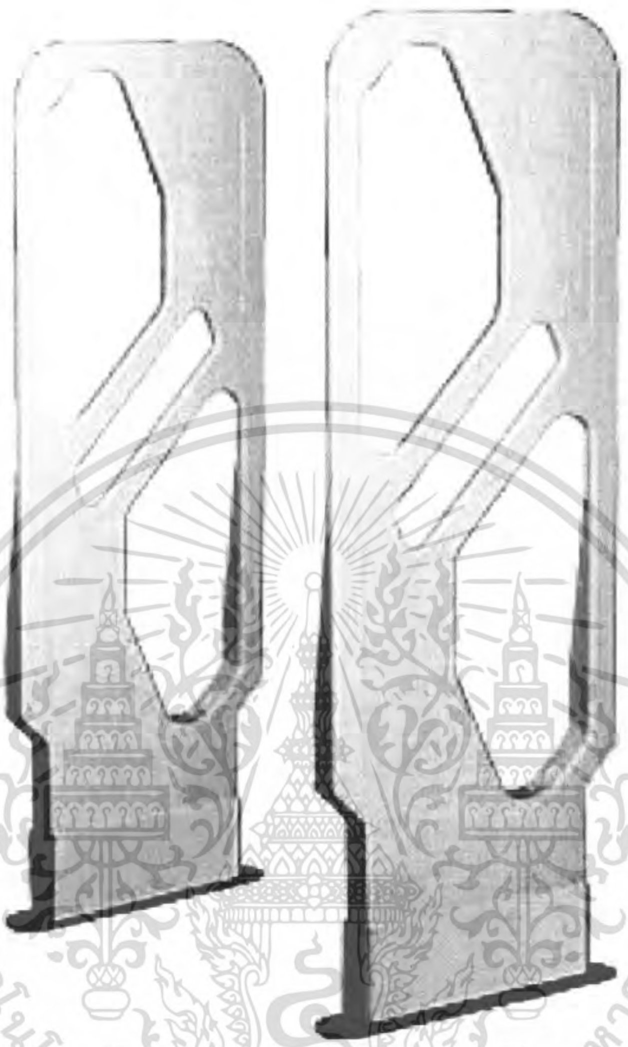
รูปที่ 2.8 โครงสร้างภายในเครื่องอ่าน RFID

รูปที่ 2.9 เครื่องอ่าน RFID แบบพกพา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.10 เครื่องอ่าน RFID แบบติดผนัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.11 เครื่องอ่าน RFID แบบอุโมงค์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

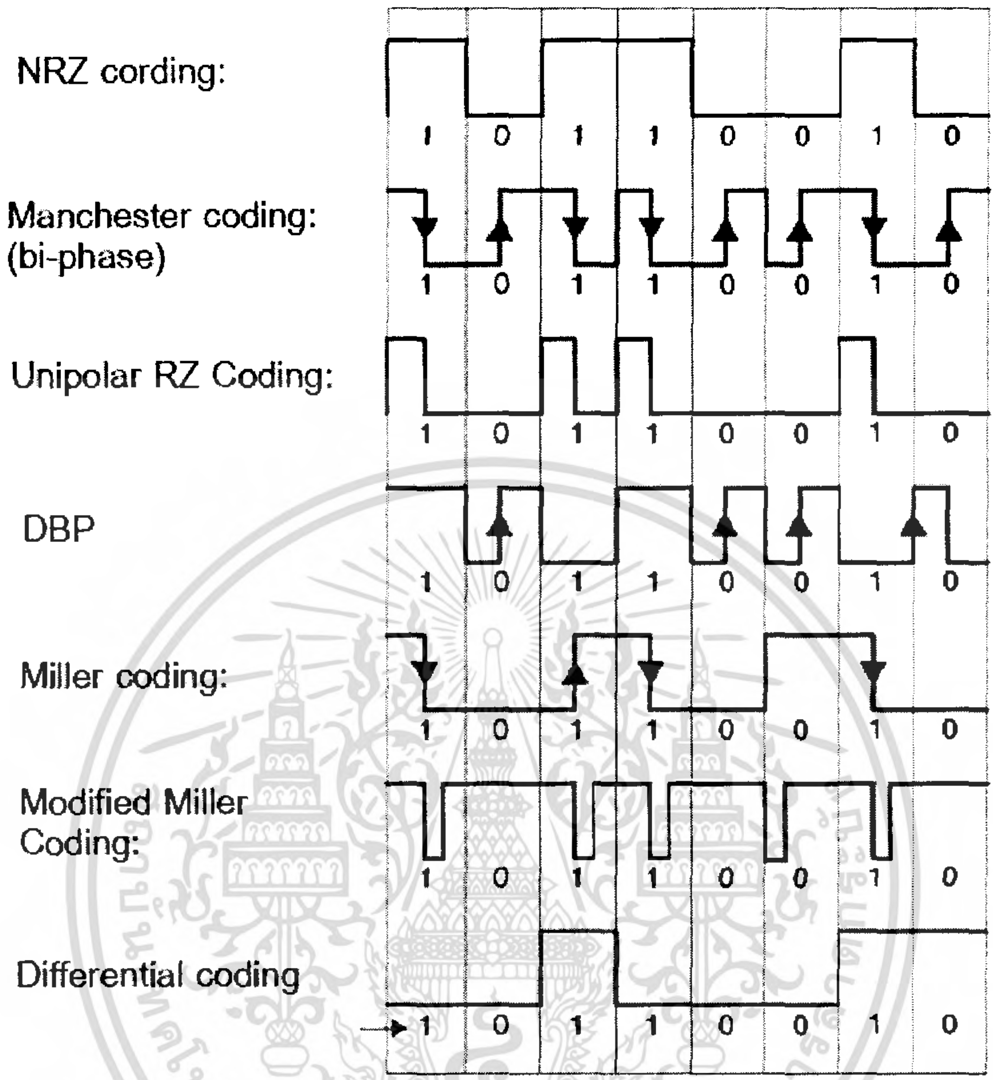


รูปที่ 2.12 เครื่องอ่าน RFID แบบประตู

2.1.4 หลักการรับและส่งข้อมูลระหว่างป้ายและเครื่องอ่าน

การรับและส่งสัญญาณ RFID โดยทั่วไปเป็นกระบวนการสื่อสารระบบดิจิตอล โดยการเข้ารหัสสัญญาณในรูปแบบที่เหมาะสมที่ใช้ ในการส่งผ่านช่องสัญญาณคือการส่งสัญญาณผ่านช่องสัญญาณจะทำให้เกิดสัญญาณรบกวน โดยมีความผิดพลาดให้น้อยที่สุด ซึ่งเทคนิคการเข้ารหัสนั้นมีหลายแบบ เช่น การเข้ารหัสสัญญาณแบบเอ็นอาร์แซด การเข้ารหัสแบบแมนเชสเตอร์ การเข้ารหัสแบบมิลเลอร์ การเข้ารหัสแบบดิฟเฟอเรนเชียล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

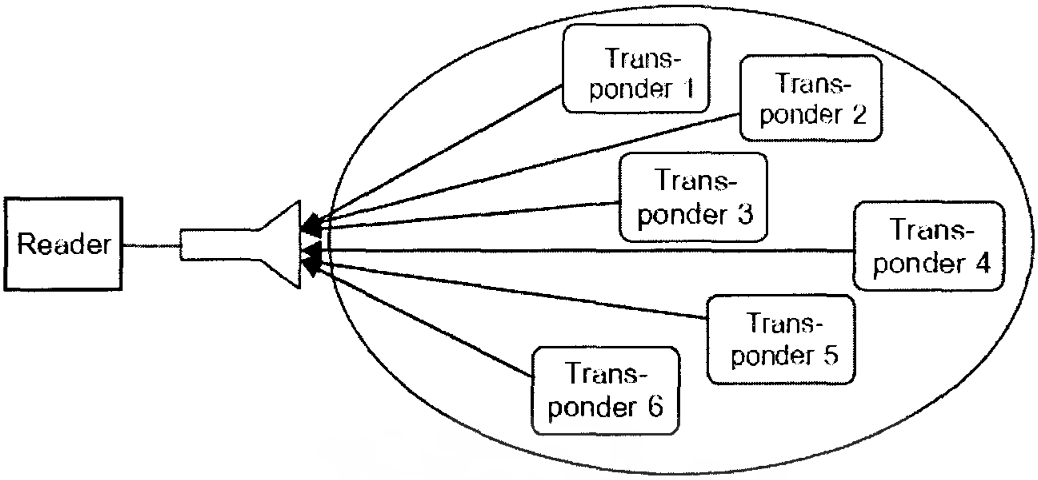


รูปที่ 2.13 ตัวอย่างการเข้ารหัสแบบต่างๆ

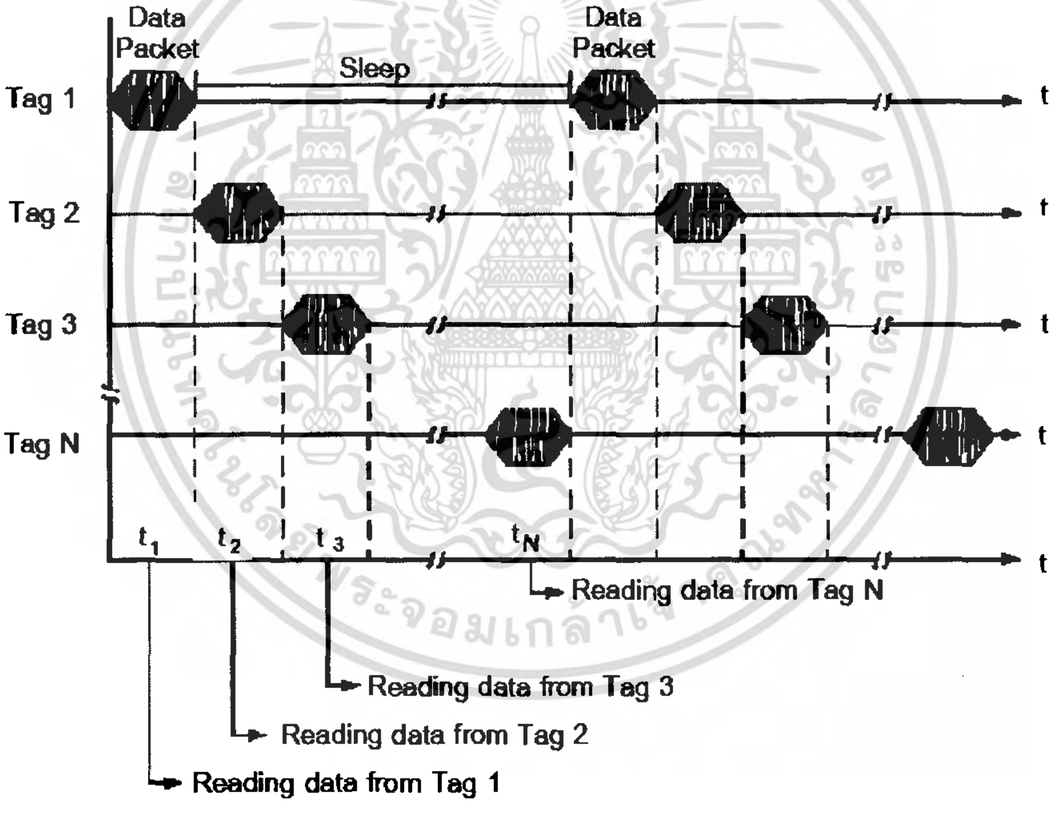
หลังจากทำการเข้ารหัสแล้ว สัญญาณจะถูกทำการมอดูเลชั่น กับคลื่นพาหะเพื่อทำการรับ-ส่งข้อมูลต่อไป การมอดูเลชั่น หมายถึง การปรับเปลี่ยนค่าต่างๆของคลื่นพาหะซึ่งเป็นคลื่นสนามแม่เหล็กไฟฟ้า เช่น แอมปลิจูด เฟส หรือ ความถี่ ตามค่าของข้อมูลที่จะส่ง

ข้อดีอีกส่วนหนึ่งของระบบ RFID คือ การอ่านข้อมูลจากป้ายได้หลายๆป้ายในเวลาเดียวกัน ดังรูปที่ 2.14 โดยมีระบบการป้องกันการชนกันของสัญญาณข้อมูล (Anti-Collision) ซึ่งทำให้การอ่านข้อมูลจากป้ายพร้อมกันในจำนวนมากทำได้รวดเร็ว ตัวอย่างการป้องกันการชนกัน เช่น ใช้เทคนิคทีดีเอ็มเอ (TDMA: Time Division Multiple Access) ซึ่งเป็นการจัดลำดับการอ่านข้อมูลจากป้ายต่างๆ ในเวลาต่างกันไป ดังรูปที่ 2.15 นอกจากนี้ยังมีกระบวนการตรวจสอบความถูกต้องของข้อมูลที่ได้รับ เช่น การทำผลรวมตรวจสอบ (Check-sum) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 เครื่องอ่านทำงานกับป้ายหลายๆป้ายพร้อมๆกัน



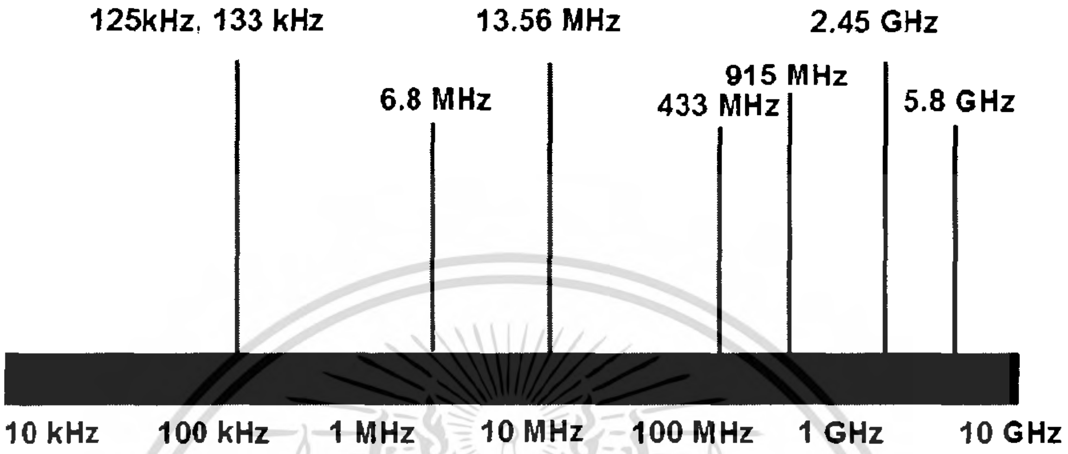
รูปที่ 2.15 ตัวอย่างของอัลกอริทึมในการป้องกันการชนกันของข้อมูลในป้าย

2.1.5 RFID ในคลื่นความถี่ต่างๆ

คลื่นพาหะที่ใช้กันจะอยู่ในย่านความถี่พลเรือน ISM (Industrial-Scientific-Medical) สามารถใช้ได้ในย่านความถี่ที่ไม่ตรงกับที่ใช้ในการสื่อสารโดยทั่วไป คลื่นความถี่ที่ใช้งานกันในระบบ RFID แบ่งออกได้ 4 ย่านใหญ่ๆ ดังแสดงในรูปที่ 2.16 คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ย่านความถี่ต่ำ ต่ำกว่า 150 กิโลเฮิรตซ์
- ย่านความถี่สูง 13.56/27.125 เมกะเฮิรตซ์
- ย่านความถี่สูงยิ่ง 433/868/915 เมกะเฮิรตซ์
- ย่านความถี่ไมโครเวฟ 2.45/5.8 กิกะเฮิรตซ์



รูปที่ 2.16 แสดงความถี่ย่านที่ระบบ RFID ถูกใช้งาน

ถ้าเปรียบเทียบความถี่ที่ใช้งานในแต่ละย่านความถี่ในด้านของระยะการอ่านสามารถสรุปได้ดังตารางต่อไปนี้

ตารางที่ 2.1 ระยะที่สามารถอ่านได้ของแต่ละย่านความถี่

ความถี่	ระยะทางที่อ่านได้
125 – 134 กิโลเฮิรตซ์	น้อยกว่า 1 เมตร (10 เซนติเมตร)
13.56 เมกะเฮิรตซ์	น้อยกว่า 1.5 เมตร (~1 เมตร)
860 – 960 เมกะเฮิรตซ์	2 – 5 เมตร 1 – 100 เมตร (ป้ายแบบแอ็กทิฟ)
2.45 กิกะเฮิรตซ์	น้อยกว่า 1 เมตร (ป้ายแบบแพสซิฟ) 1 – 15 เมตร (ป้ายแบบแอ็กทิฟ)

ในการใช้งาน 2 ย่านความถี่แรก จะเหมาะกับงานที่มีระยะการสื่อสารข้อมูลระยะใกล้ โดยย่านความถี่ต่ำ 125 กิโลเฮิรตซ์ และ 134 กิโลเฮิรตซ์ นิยมใช้สำหรับควบคุมการเข้าออกสถานที่ หรือการลงทะเบียนสัตว์ ส่วนในย่านความถี่สูง 13.56 เมกะเฮิรตซ์ นิยมใช้กับบัตรเอนกประสงค์ไร้สัมผัสและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสือเดินทางอิเล็กทรอนิกส์ ส่วนย่านความถี่สูงยิ่งใช้งานในลักษณะที่มีการสื่อสารระยะไกล เช่น ระบบการเก็บค่าบริการทางด่วน ระบบขนส่งสินค้า เป็นต้น

ตารางที่ 2.2 แสดงข้อแตกต่างของ RFID ระบบต่างๆ

พารามิเตอร์	ย่านความถี่ต่ำ (LF)	ย่านความสูง (HF)			ย่านความถี่สูงยิ่ง (UHF)	ย่านไมโครเวฟ
ความถี่	125-134 kHz	13.56 MHz	13.56 MHz	PJM 13.56 MHz	868-915 MHz	2.45-5.8 GHz
ระยะในการอ่าน	ถึง 1.2 เมตร	0.7-1.2 เมตร	ถึง 1.2 เมตร	ถึง 1.2 เมตร	ถึง 4 เมตร	ถึง 15 เมตร
ความเร็วในการอ่าน	ไม่เร็วมาก	น้อยกว่า 1-5 วินาที (5s for 32KB)	ปานกลาง (0.5m/s)	เร็วมาก (4m/s)	เร็ว	เร็วมาก
สภาวะที่ชื้น	ไม่มีผลกระทบ	ไม่มีผลกระทบ	ไม่มีผลกระทบ	ไม่มีผลกระทบ	มีผลกระทบ	มีผลกระทบ
การประยุกต์ใช้งาน	Access immobilizer, gas, laundry	Smart Cards: identification, electronic ID, ticketing	Library, ticketing for big events, good logistic	Baggage handing at airports, boarding pass, pharmacy	Truck registry, trailer tracking	Road tolling, container tracking

2.2 ไมโครคอนโทรลเลอร์

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

- ต้องมีแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว

- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในชิพ

- มีหน่วยความจำสำหรับโปรแกรมและข้อมูล (RAM) อยู่ภายในชิพจำนวน 128 ไบต์ หรือ 256 ไบต์

- สามารถใช้หน่วยความจำสำหรับโปรแกรมและข้อมูลที่อยู่ภายนอกชิพได้ อย่างละ 64 กิโลไบต์ แยกจากกัน

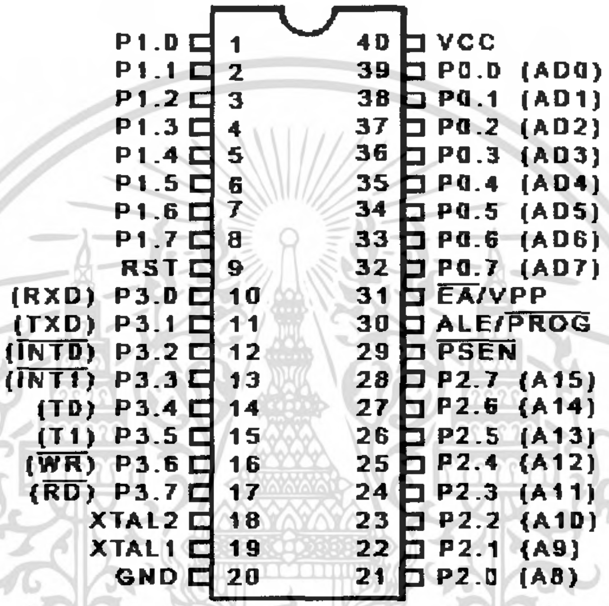
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีรีจิสเตอร์สำหรับใช้งานเป็นไทเมอร์ หรือเคาเตอร์ เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิพ หรือนับเปลี่ยนสถานะของสัญญาณภายนอก 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ วัดความกว้างของพัลส์ หรือใช้วัดช่วงเวลา

- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วน สามารถเข้าถึงข้อมูลได้ในระดับบิต เพื่อให้การออกแบบโปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น

- สามารถประมวลผลแบบบูลีน เพื่อใช้งานควบคุมโดยเฉพาะ

ซึ่งรายละเอียดตำแหน่งขาต่างๆ ของMCS-51 แสดงในรูปที่ 2.17



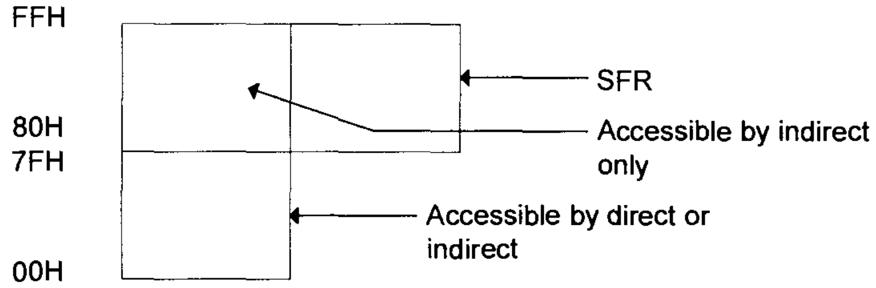
รูปที่ 2.17 ตำแหน่งขา MCS-51

2.2.2 ลักษณะเด่นของ MCS - 51

2.2.2.1 หน่วยความจำภายใน (Internal Memory)

ในคอมพิวเตอร์ทั่วๆไปนั้น ต้องมีหน่วยความจำสำหรับเก็บโปรแกรมเพื่อสั่งงานให้ CPU ทำงานหรือหน่วยความจำสำหรับเก็บข้อมูล ซึ่งอาจอยู่ใน ROM หรือ RAM ก็ได้ สำหรับ MCS-51 จะมีหน่วยความจำภายในสำหรับเก็บโปรแกรม และ หน่วยความจำสำหรับเก็บข้อมูล อย่างไรก็ตาม ถ้าหากหน่วยความจำภายในที่ให้มานี้ไม่พอใช้ ผู้ใช้สามารถต่อหน่วยความจำภายนอกเพิ่มให้กับ MCS-51 นี้ได้ ซึ่งสามารถเชื่อมต่อได้มากที่สุดถึง 64 กิโลไบต์ โดยจะมีการจัดวางหน่วยความจำดังรูปที่ 2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดงส่วนต่าง ๆ ของหน่วยความจำภายในไมโครคอนโทรลเลอร์

ส่วนที่ 1 หน่วยความจำภายใน ตั้งแต่แอดเดรส 00H ถึง 7FH ใช้สำหรับเก็บข้อมูลทั่วไปจำนวน 128 ไบต์

- **Working Register** แอดเดรส 00H ถึง 1FH เป็นจำนวน 32 ไบต์ ถูกแบ่งให้เป็นกลุ่มของรีจิสเตอร์ใช้งานทั่วไป 4 ชุด ชุดละ 8 ตัวคือ R0 -R7 รีจิสเตอร์ทั้ง 4 ชุด การใช้งานจะถูกเลือกโดยการกำหนดใน Register Program Status Word
- **หน่วยความจำที่สามารถเข้าถึงในระดับบิตได้** (Bit Addressable) ที่แอดเดรส 20H ถึง 2FH เป็นจำนวน 16 ไบต์ แต่ละบิตของหน่วยความจำบริเวณนี้จะมีการกำหนดแอดเดรสของแต่ละบิตไว้แน่นอนเพื่อใช้สำหรับการทำงานของชุดคำสั่งที่ประมวลผลแบบบูลีน ได้
- **หน่วยความจำสำหรับการใช้งานทั่ว ๆ ไป** หน่วยความจำส่วนนี้จะอยู่ที่แอดเดรสตั้งแต่ 30H ถึง 7FH เป็นจำนวน 80 ไบต์โดยทั่วๆ ไปแล้วจะกำหนดแอดเดรสเริ่มต้นของข้อมูลที่แอดเดรส 30H นี้

ตารางที่ 2.3 แสดงบิตต่างๆภายในรีจิสเตอร์ PSW

CY	AC	F0	RS1	RS0	OV	-	P
			0	0	เลือกใช้รีจิสเตอร์กลุ่มที่ 1		
			0	1	เลือกใช้รีจิสเตอร์กลุ่มที่ 2		
			1	0	เลือกใช้รีจิสเตอร์กลุ่มที่ 3		
			1	1	เลือกใช้รีจิสเตอร์กลุ่มที่ 4		

ชื่อบิต

RS1 ใช้เลือกกลุ่มรีจิสเตอร์ R0 - R7

RS0 ใช้เลือกกลุ่มรีจิสเตอร์ R0 - R7

CY Carry Flag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

AC Auxiliary Carry Flag

OV บิตแสดงการเกิดโอเวอร์โฟล

\- บิตที่ผู้ใช้กำหนดการใช้งานเองได้

P Parity Flag ถูกเซตหรือเคลียร์โดยวงจรภายใน MCS -51 ในแต่ละไซเกิลของคำสั่งแต่ละคำสั่ง ใช้เป็นตัวบอกให้ทราบว่าในรีจิสเตอร์ ACC มีข้อมูลที่เป็น 1 เป็นจำนวนคู่หรือคี่

ส่วนที่ 2 มีตั้งแต่แอดเดรส 80H - FFH เก็บส่วนของหน่วยความจำที่ใช้สำหรับเก็บข้อมูลของรีจิสเตอร์ ใช้งานเฉพาะสำหรับรีจิสเตอร์ที่ใช้สำหรับงานเฉพาะบางตัว (SFR : Special Function Register) ซึ่งสามารถเข้าถึงโดยวิธีโดยตรงเท่านั้น แอดเดรสต่าง ๆ ของ SFR มีดังตารางต่อไปนี้

ตารางที่ 2.4 แสดง SFR ของ MCS - 51

				8 Byte					
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

ส่วนที่ 3 มีตั้งแต่แอดเดรส 80H ถึง FFH ซึ่งจะเห็นว่าซ้ำกับส่วนที่ 2 แต่การเข้าถึงข้อมูลจะใช้วิธีโดยทางอ้อมเท่านั้นซึ่ง MCS-51 จะทราบเองว่าจะเข้าถึงข้อมูลใดตำแหน่งใดและจะต้องใช้วิธีไหน ในไมโครคอนโทรลเลอร์ เบอร์ AT89C2051 นั้นจะไม่มีหน่วยความจำในส่วนนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

83292

2.2.2.2 การติดต่อและควบคุมอุปกรณ์ภายนอก (I/O Interface Driving Power)

จะเห็นได้ว่าความสามารถทางด้านการติดต่อกับอุปกรณ์อินพุตและอุปกรณ์เอาท์พุตของ MCS-51 นี้สามารถจะนำไปใช้งานได้เลย แต่ถ้าไม่เพียงพอกับความต้องการใช้งานก็สามารถต่ออุปกรณ์อินพุตและเอาท์พุตเพิ่มได้ และมีพอร์ตอนุกรมที่สามารถกำหนดความเร็วได้

2.2.3 รีจิสเตอร์ที่ใช้ในการเขียนโปรแกรม

2.2.3.1 IE: อินเตอร์รัพต์ อีนาเบิล รีจิสเตอร์

EA	-	ET2	ES	ET1	EX1	ET0	EX0

รูปที่ 2.19 อินเตอร์รัพต์ อีนาเบิล รีจิสเตอร์

ในกรณีที่ต้องการให้มีการอินเทอร์รัพท์จากไทเมอร์ 1 และ ไทเมอร์ 0 ให้ใช้คำสั่ง SET ET0 และ SET ET1 แล้วนำเอาโปรแกรมในการรับและส่งไปไว้ที่ตำแหน่ง 000BH (t0 interrupt vector) และ 001B(t1 interrupt vector) แต่ต้องไม่เกิน 8 ไบต์ ซึ่งในความเป็นจริงแล้วโปรแกรมมักจะมากกว่า 8 ไบต์ ดังนั้นที่ตำแหน่ง 000BH และ 001B ส่วนมากจึงเป็นคำสั่ง JUMP

2.2.3.2 IP: อินเตอร์รัพต์ ไพออริตี้ รีจิสเตอร์

-	-	PT2	PS	PT1	PX1	PT0	PX0

รูปที่ 2.20 อินเตอร์รัพต์ ไพออริตี้ รีจิสเตอร์

รีจิสเตอร์ตัวนี้เป็นรีจิสเตอร์ใช้กำหนดความสำคัญในการอินเตอร์รัพต์ ถ้าบิตใดเป็น “1” ลักษณะการอินเตอร์รัพต์ในบิตนั้นจะมีความสำคัญสูงสุด ใช้คำสั่ง SETB PT1

2.2.3.3 TCON: ไทเมอร์/เคาน์เตอร์ คอนโทรล รีจิสเตอร์

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

รูปที่ 2.21 ไทเมอร์/เคาเตอร์ คอนโทรลรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรีจิสเตอร์มีบิตที่เราจะต้องใช้คือ TR0 และ TR1 ในตัวควบคุมไทเมอร์และเคาน์เตอร์ โดยถ้าให้เป็น "0" จะหมายถึง "off" และ "1" จะหมายถึง "on"

2.2.3.4 TMOD: ไทเมอร์/เคาน์เตอร์ โหมด คอนโทรลรีจิสเตอร์

GATE	C/T	M1	M0	GATE	C/T	M1	M0

รูปที่ 2.22 ไทเมอร์/เคาน์เตอร์ โหมดคอนโทรลรีจิสเตอร์

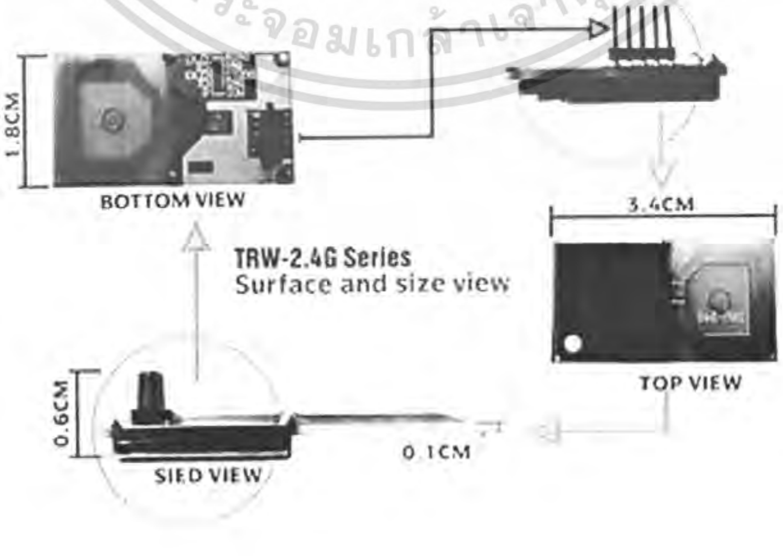
ในการใช้งานพอร์ตสื่อสารอนุกรมหรือไทเมอร์ จะใช้งานรีจิสเตอร์ TMOD นี้บางบิต คือ Gate , C/T , M1 , M0

ตารางที่ 2.5 การเลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์โดยใช้ M0, M1

M0	M1	Mode	คำอธิบาย
0	0	0	13 Bit Timer
0	1	1	16 Bit Timer
1	0	2	8 Bit Auto-reload
1	1	3	Timer/Counter 8 Bit 2 ชุด

2.3 RF- MODULE TRW 2.4 GHz

2.3.1 ลักษณะของ RF- MODULE TRW 2.4 GHz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.23 แสดงมุมมองของModule TRW 2.4 GHz
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ MODULE TRW 2.4 GHz

- เป็นการส่งผ่านข้อมูลแบบ full duplex
- เลือกความเร็วในการส่งข้อมูลได้ 2 แบบ ดังตารางที่ 2.6

ตารางที่ 2.6 ความเร็วในการรับ - ส่งข้อมูล

ด้านส่ง	ด้านรับ	ระยะทาง
TRW-2.4GHz(250kbps)	TRW-2.4GHz(250kbps)	280 m
TRW-2.4GHz(1Mbps)	TRW-2.4GHz(1Mbps)	150m

- ทำงานที่ Voltage 3.0 – 3.6 V
- วิธีส่งการเป็นมอดูเลทแบบ GFSK
- ความถี่ใช้งานที่ 2.4 – 2.524 GHz
- กำลังงานเอาต์พุต 44 dBm
- มีสายอากาศอยู่ภายในโมดูล
- มีขนาดอยู่ที่ 20.0*36.7*2.4mm

2.3.2 การประยุกต์ใช้

- อุปกรณ์ไร้สายเช่น เมาส์ คีย์บอร์ด จอยสติก
- ใช้กับการสื่อสารข้อมูลแบบไร้สาย
- ระบบเตือนภัยและรักษาความปลอดภัย
- หูโทรศัพท์แบบไร้สาย
- ควบคุมอุปกรณ์ภายในบ้านโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 ลักษณะการจัดขาของโมดูล

รูปที่ 2.24 แสดง โมดูล TRW 2.4GHz

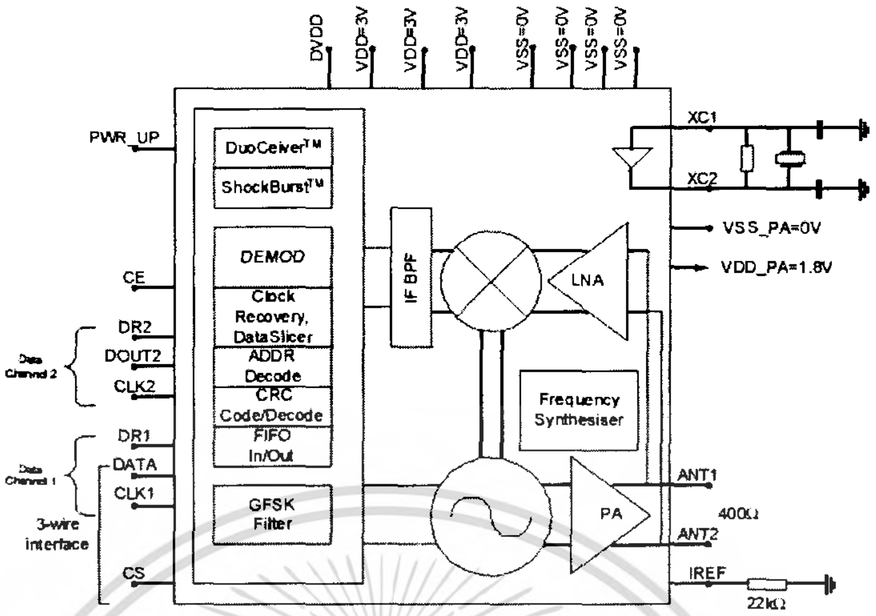
ตารางที่ 2.7 แสดงรายละเอียดการทำงานของโมดูลแต่ละขา

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 3-wire interface
6	DATA	I/O	RX data channel 1/TX data input/3-wire interface
7	DR1	Output	RX data ready at data channel 1(ShockBurst only)
8	DOUT2	Output	RX data channel 2
9	DR2	Output	RX data ready at data channel 2 (ShockBurst only)
10	VCC	Power	Power Supply(+3V DC)

2.3.4 โหมดการทำงาน

ชิพใช้รับและส่งในโมดูล RF TRW 2.4 GHz ถูกออกแบบและพัฒนาโดย NORDIC SEMICONDUCTOR โดยมีชื่อรุ่นระบุไว้ว่า nRF โดยการส่งและรับข้อมูลที่ความถี่ 2.4 GHz จะมีชื่อเรียกว่า “nRF2401A” โดยจะทำงานในโหมด ShockBurst, Duotranceiver ซึ่งมีการต่อ nRF2401 กับอุปกรณ์ภายนอก แสดงได้ดังรูปที่ 2.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 nRF2401 ต่อกับอุปกรณ์ภายนอก

โหมดการทำงานหลักของ nRF2401 จะควบคุมโดยใช้ 3 ขาเป็นตัวควบคุมดังตารางที่ 2.8

ตารางที่ 2.8 โหมดการทำงานหลักของ nRF2401

Mode	PWR_UP	CE	CS
Active	1	1	0
Configuration	1	0	1
Stand by	1	0	0
Power down	0	X	X

โหมดการทำงานแบบ Activeได้เป็น 2 โหมดคือ

- ShockBurst Mode(ใช้ในโมดูล TRW 2.4GHz)
- Direct Mode

2.3.4.1 การส่งข้อมูลโหมดการทำงานแบบ ShockBurst

การทำงานด้วยเทคนิคแบบ FIFO คือการรับและส่งข้อมูลบนชิป แบบเข้าก่อน-ออกก่อน (First in – First out) โดยกำหนดให้สัญญาณข้อมูลมีอัตราส่งบิตข้อมูลความเร็วต่ำและในการส่งข้อมูลจะมีอัตราบิตข้อมูลความเร็วสูงเป็นผลให้ใช้กำลังงานลดลง เมื่อโมดูลทำงานในโหมด ShockBurst จะเพิ่มระดับการส่งข้อมูลได้ถึง 1 Mbps และต้องใช้ไมโครคอนโทรลเลอร์ความเร็วสูง(MCU)ในการประมวลผล โดยการประมวลผลแบบความเร็วสูงนี้จะต้องสัมพันธ์กับโปรโตคอลบนชิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของ nRF2401

- ลดการใช้กระแส

- ระบบมีราคาถูกเนื่องจากไมโครคอนโทรลเลอร์มีราคาถูก

- ลดอัตราการเสี่ยงของการชนกันในอากาศ เนื่องจากการส่งสัญญาณในช่วงเวลาสั้นๆได้ดี

หลักการทำงานในโหมด ShockBurst

โหมดการส่งข้อมูล

ทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์กับขา CE, CLK1, DATA โดยมีขั้นตอนการทำงานดังนี้

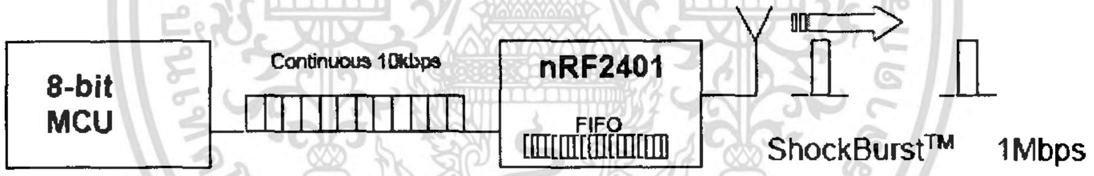
- เมื่อต้องไมโครคอนโทรลเลอร์ส่งข้อมูลจะกำหนดให้ขา CE แสดงสถานะ "high" คือ กระตุ้น nRF2401 ให้ประมวลผลข้อมูลบนโมดูล

- แอดเดรสของด้านรับ และ payload ข้อมูลจะเริ่มทำงานเข้าสู่ nRF2401

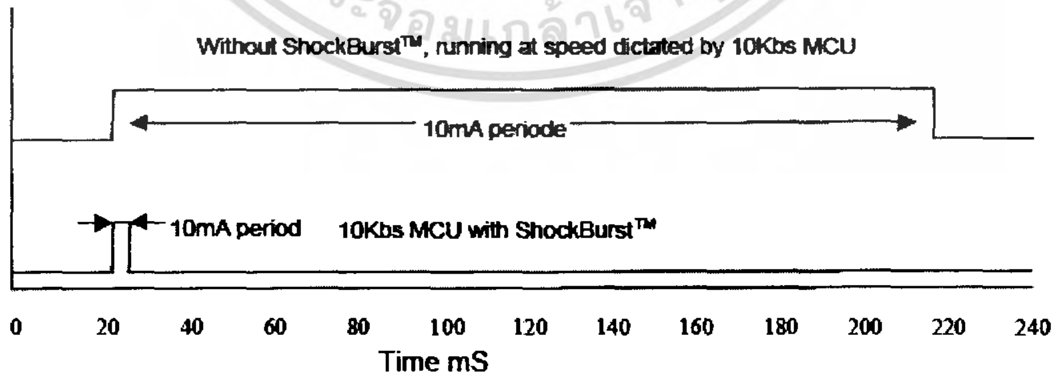
-ไมโครคอนโทรลเลอร์ กระตุ้นให้ขา CE แสดงสถานะ "low" เพื่อกระตุ้นให้โมดูลทำการส่งข้อมูล

- ข้อมูลที่ส่งโดยShockBurst จะมีpackageที่สมบูรณ์เมื่อทำการส่งเรียบร้อยจะกลับมาอยู่ในโหมดที่ stand-by พร้อมที่จะทำงานต่อไป

เมื่อข้อมูลส่งจากไมโครคอนโทรลเลอร์ไปยังปลายทางจะถูกส่งออกเป็น package โดยในตัวอย่างนี้กำหนดความเร็วไว้ที่ 1 Mbps ดังรูปที่ 2.26



รูปที่ 2.26 แสดงลำดับการส่งข้อมูลด้วยวิธี ShockBurst

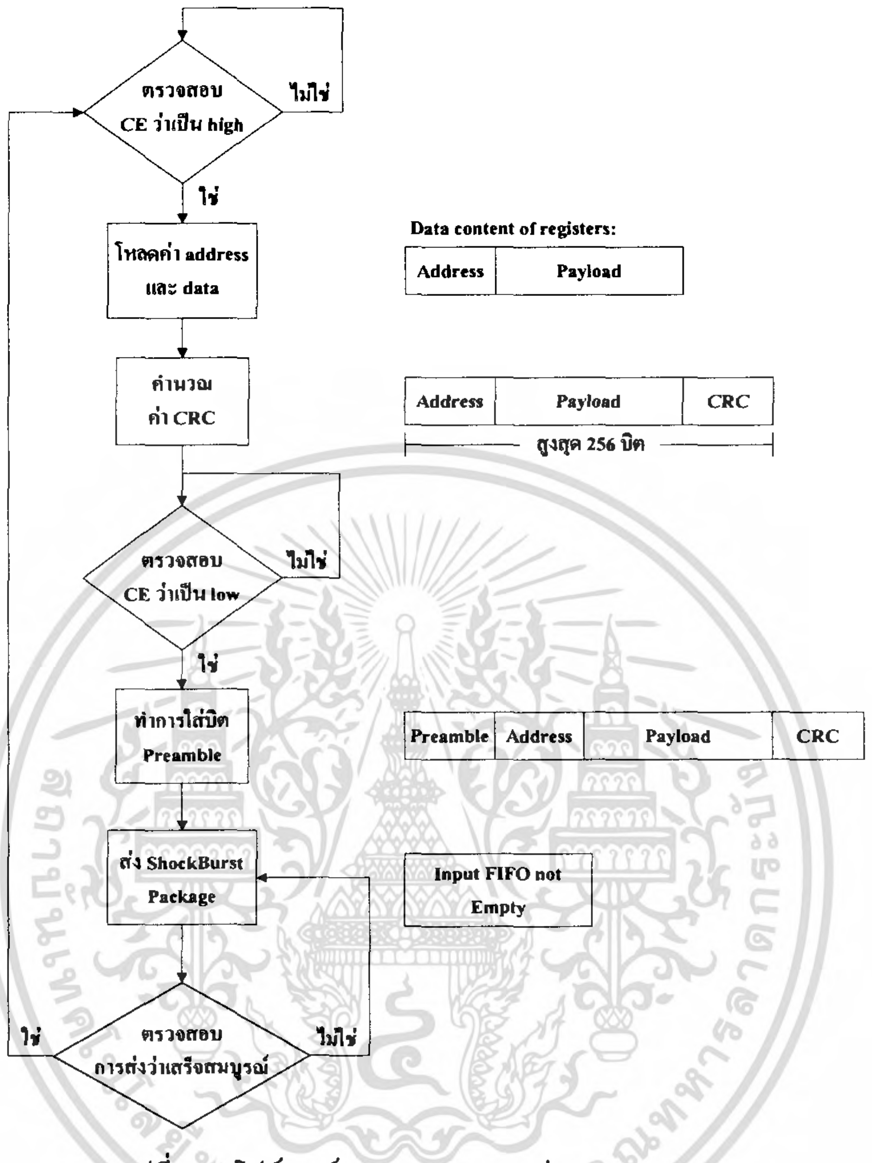


รูปที่ 2.27 การใช้กระแส RF โดยใช้และไม่ใช้เทคโนโลยี Shock Burst

จากรูปที่ 2.27 จะเห็นว่า วิธีการแบบ ShockBurst จะใช้กระแสไฟที่น้อยกว่าเมื่อเทียบกับคาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งลำดับชั้นในการส่งข้อมูลของ nRF2401 ในโหมด ShockBurst แสดงได้ดังรูปที่ 2.28



รูปที่ 2.28 โฟลว์ชาร์ต ShockBurst การส่งของ nRF2401

โหมดการรับข้อมูล

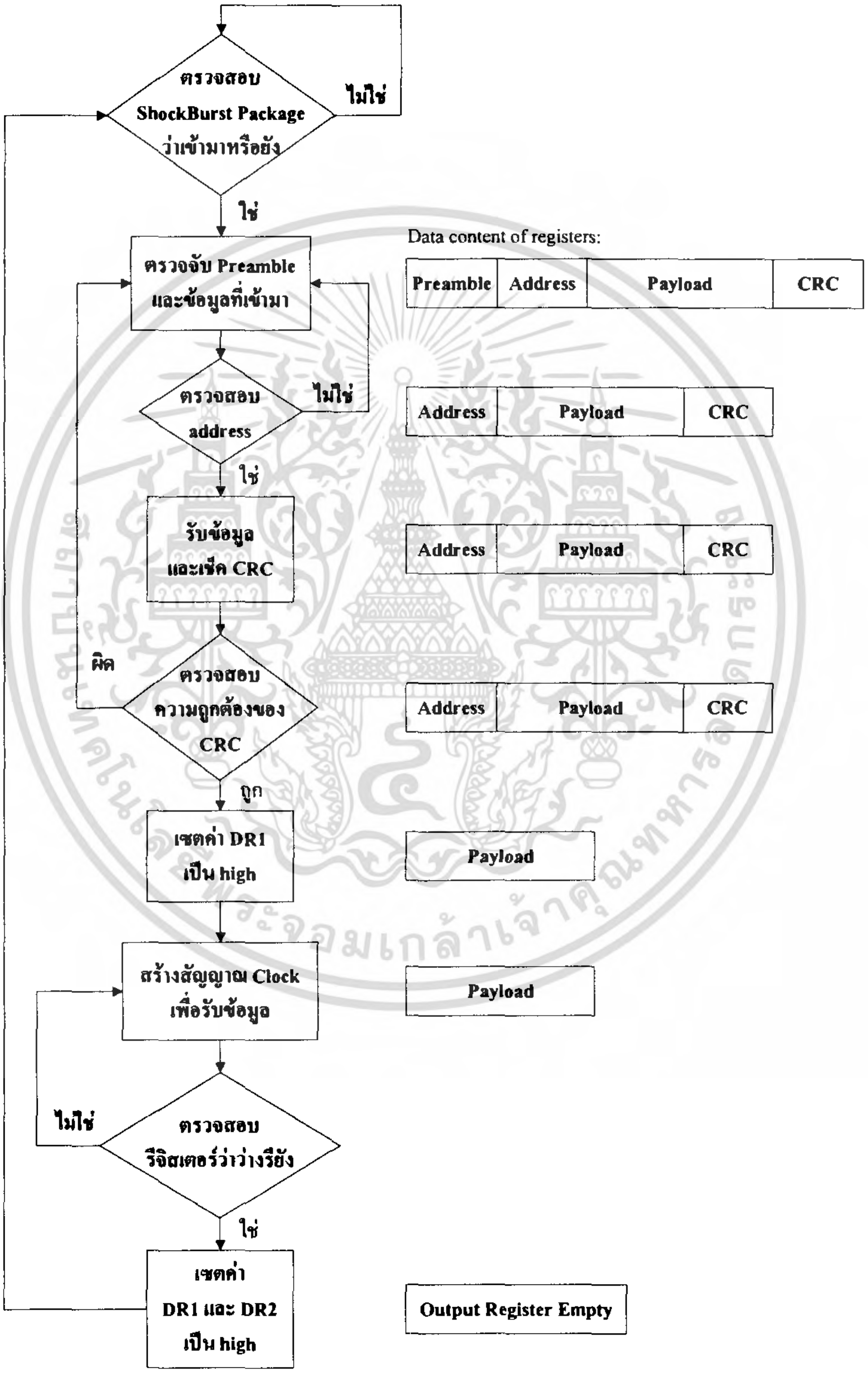
ทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์กับขา CE, CLK1, DR1 และ DATA โดยมีขั้นตอนการทำงานดังนี้

- เมื่อแอดเดรสและขนาดของ payload ที่มีถูกต้อง RF package ที่เข้ามาจะ เซ็ตให้ขา CE มีสถานะ “high” เมื่อชิป nRF 2401 ทำงานในโหมดShock Burst
- หลังจากเวลาผ่านไป 200 μs nRF 2401 จะตรวจสอบอากาศสำหรับการสื่อสารที่จะเข้ามา
- เมื่อ package ที่รับเข้าถูกต้องถูกเรียกมา (แอดเดรสและCRC ถูกต้อง) nRF 2401 จะทำการย้าย Preamble, address และ CRC
- จากนั้นจะแจ้งไปที่ไมโครคอนโทรลเลอร์ให้ทำการเซ็ตขาDR1เป็นสถานะ “high”
- หลังจากนั้นจะเซ็ตขา CE เป็นสถานะ “low” เพื่อป้องกัน RF front end (โหมดที่กระแสต่ำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไมโครคอนโทรลเลอร์ จะจับเวลาอัตราบิตที่เหมาะสมของ payload data ไว้ได้

- เมื่อรับบิต payload data เรียบร้อยแล้ว ชิป nRF2401 จะเซ็ตขา DR1 ให้มีสถานะ "low" อีกครั้ง และพร้อมที่จะรับข้อมูล package โดยที่ขา CE แสดงสถานะ "high" ถ้าขา CE แสดงสถานะ "low" จะไม่สามารถรับข้อมูลใหม่ได้ ซึ่งแสดงดังโฟลว์ชาร์ตในรูปที่ 2.29



รูปที่ 2.29 โฟล์วชาร์ต การทำงานโหมดShockBurst ทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.2 หลักการทำงานในโหมด Dircect

ในโหมด Direct ชิป nRF2401 จะทำงานเหมือนอุปกรณ์ RF ทั่วไป ข้อมูลต้องมีอัตราบิตเร็วอยู่ที่ 100Mps ± 200ppm หรือที่อัตราบิตต่ำสุดที่ 250 ± 200ppm เพื่อให้ด้านรับตรวจจับสัญญาณ

การทำงานด้านส่ง

การเชื่อมต่อกับไมโครคอนโทรลเลอร์ เซ็ตที่ขา CE, DATA

- เมื่อทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ขา CE แสดงสถานะ “high”

- nRF2401 ทำงานและหลังเวลา 200 μs ข้อมูลจะถูกมอดูเลทกับสัญญาณพาหะ

การทำงานด้านรับ

การเชื่อมต่อกับไมโครคอนโทรลเลอร์ เซ็ตที่ขา CE, CLK1, DATA

- ทางด้านรับขา CE จะแสดงสถานะ “high” ข้อมูลจะถูกรบกวนด้วยสัญญาณรบกวนในอากาศ

- ขา CLK1 จะทำให้ชิป nRF2401ทำการล็อกข้อมูลที่เข้ามา

- เมื่อ perамble มาถึง ขา CLK1 และ DATAจะล็อกสัญญาณที่เข้ามา RF package จะปรากฏที่ขา DATA ด้วยความเร็วที่ได้ส่งมา

- ดีมอดูเลทสัญญาณเพื่อนำสัญญาณ clock, permeable ต้องแสดง 8 บิต

- ในโหมดนี้สัญญาณที่ขา DR1 จะไม่มี

การตั้งค่าการทำงานในโหมด *ShockBurst*

การตั้งค่าใน ShockBurstโดยละเอียดมี ดังนี้

- ความกว้างของ Payload: ระบุจำนวนของบิต payload ใน RF package ซึ่งทำให้ชิป nRF2401 สามารถแบ่งแยกระหว่างข้อมูล payload และ CRC ใน package ที่รับเข้ามาได้

- ความกว้าง Address: เซ็ตจำนวนบิตที่แอดเดรสใช้ใน RF package ซึ่งทำให้ชิป nRF2401 สามารถแบ่งแยกระหว่างแอดเดรสและข้อมูล package ได้

- แอดเดรส (Rx ช่อง 1 และ 2) : กำหนดแอดเดรสสำหรับข้อมูลทางด้านรับ

- CRC: สามารถตรวจสอบความผิดพลาดและถอดรหัสได้

2.3.5 องค์ประกอบของชุดเฟรมข้อมูล

ส่วนประประกอบของเฟรมข้อมูลแบ่งเป็น 4 ส่วนดังตารางที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.9 แสดงองค์ประกอบของชุดเฟรมข้อมูล

1. PREAMBLE	- ส่วน preamble ต้องมีใน Shock Burst mode และ Direct mode - Preamble มีความยาว 8 บิตและ ขึ้นกับบิตแรกของแอดเดรสสำหรับ Direct mode PREAMBLE 1st ADDR_BIT 11010101 1 00101010 0 - Preamble จะใส่ไปในชุดข้อมูลโดยอัตโนมัติและทำให้มีที่ว่างสำหรับ payload ใน ShockBurst mode ทางด้านรับ preamble จะถูกย้ายจากข้อมูลเอาต์พุต สำหรับ Direct mode preamble จะไม่ย้ายออกจากข้อมูลเอาต์พุต
2. ADDRESS	- ส่วนของแอดเดรสต้องมีในโหมด ShockBurst และ Direct - ความยาว 80 – 40 บิต -ในShockBurstแอดเดรสจะถูกย้ายโดยอัตโนมัติจากชุดข้อมูลด้านรับสำหรับ Direct mode MCU ต้องควบคุมแอดเดรส
3. PAYLOAD	- ส่งข้อมูล - ในโหมด ShockBurst payload มีขนาดน้อยที่สุด 256 บิต (แอดเดรส 8-40 บิต +CRC 8หรือ16 บิต) - ใน Direct mode ความยาวมากสุดสำหรับ 1 Mbps คือ 4000 บิต
4. CRC	- CRC จะเลือกได้ในโหมด ShockBurst และไม่ใช้ในโหมด Direct - มีความยาว 8 หรือ 16 บิต - CRC จะถูกย้ายจากรับข้อมูลเอาต์พุตทางด้านรับ

ซึ่งมีโครงสร้างของเฟรมข้อมูลดังรูปที่ 2.30

Preamble	Address	Payload	CRC

รูปที่ 2.30 แสดงลำดับเฟรมข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 ตำแหน่งบิตข้อมูลของโมดูลความถี่วิทยุ TRW-2.4 GHz

ตารางที่ 2.10 แสดงรายละเอียดของตำแหน่งบิตข้อมูลในโมดูลความถี่วิทยุ

	ตำแหน่งบิต	จำนวนบิต	ชื่อ	การทำงาน
การตั้งค่า Shock Burst	120-143	24	TEST	จองไว้สำหรับการทดสอบข้อมูล
	112-119	8	DATA2_W	จำนวนความกว้างของบิตข้อมูล ช่องสัญญาณ 2
	104-111	8	DATA1_W	จำนวนความกว้างของบิตข้อมูล ช่องสัญญาณ 1
	64-103	40	ADDR2	แอดเดรสของตัวรับ ช่องสัญญาณ 2
	24-63	40	ADDR1	แอดเดรสของตัวรับ ช่องสัญญาณ 1
	18-23	6	ADDR_W	จำนวนบิตที่จองไว้สำหรับแอดเดรสตัวรับ
	17	1	CRC_L	จำนวนความกว้างของบิต CRC
	16	1	CRC_EN	สร้างCRCสำหรับTXและตรวจสอบบิตCRC (RX)
	15	1	RX2_EN	กำหนดจำนวนสัญญาณด้านรับทั้ง 2 ช่อง
	14	1	CM	เลือกโหมด(Direct หรือ ShockBurst)
	13	1	RFDR_SB	เลือกอัตราการส่งข้อมูล(1 Mbps และ 250 Kbps)
การตั้งค่าอุปกรณ์	10-12	3	XO_F	ความถี่คริสตอล
	8-9	2	RF_PWR	เลือกค่ากำลังตัวส่ง
	1-7	7	RF_CH#	เลือกช่องสัญญาณความถี่
	0	1	RXEN	เลือกโหมดการรับหรือส่ง

2.3.7 การตั้งค่าในโหมด ShockBurst

ช่วงบิต (16-119) ประกอบด้วย ส่วนของการตั้งค่าการทำงานของ Shock Burstไปยัง protocol อย่างละเอียด ในระหว่างการทำงานจะมีเพียงไบต์แรกของช่องสัญญาณความถี่และการเลือกระหว่าง Rx และ Tx เท่านั้นที่ต้องเปลี่ยน

ตารางที่ 2.11 PLL setting

PLL_CTRL		
C121	D120	PLL
0	0	Open TX/Closed RX
0	1	Open TX/Open RX
1	0	Closed TX/Closed RX
1	1	Closed TX/Open RX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตที่120-121 :PLL_CTRL ควบคุมการเซ็ตของ PLL สำหรับการทดสอบด้วยการปิด TX ถ้าในการทำงานที่เป็นปกติบิตทั้ง 2 ต้องแสดงสถานะ “LOW” ทั้งคู่

DATA2_W							
119	118	117	116	115	114	113	112

DATA1_W							
111	110	109	108	107	106	105	104

รูปที่ 2.31 จำนวนของบิตใน payload

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

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

ข้อสังเกต : จำนวนบิตทั้งหมดใน Shock Burst RF package อาจไม่เกิน 256 ความยาวสูงสุดของส่วน payload หาได้จาก

DATAx_W (bits) = 256 – ADDR_W – CRC

โดยที่ ADD_W: ความยาวแอดเดรสของ Rx ที่เซ็ตในบิตที่[18-23]

CRC: Check sum 8 หรือ 16 บิต ที่เซ็ต ในบิตที่ [17]

PRE: Preamble 8 บิตซึ่งรวมโดยอัตโนมัติ

แอดเดรสและCRC ที่สั้นขึ้น จะทำให้มีที่สำหรับ payload data มากขึ้น

- ADDRx

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

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

รูปที่ 2.32 แอดเดรสของตัวรับ 1 และ 2

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

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

ข้อสังเกต : บิตใน ADDRx มากกว่าความกว้างที่ กำหนดไว้ ใน ADDR_W ซึ่งขีดขาวเกินไปและสามารถเซ็ตเป็นลอจิก 0 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ADDR_W & CRC

ADDR_W						CRC_L	CRC_EN
23	22	21	20	19	18	17	16

รูปที่ 2.33 จำนวนบิตที่ต้องจองไว้สำหรับ RX address + CRC setting

บิตที่ 18-23 : ADDR_W: จำนวนบิตที่ต้องจองไว้สำหรับแอดเดรส Rx ใน Shock Burst

ข้อสังเกต : จำนวนบิตแอดเดรสสูงสุดคือ 40 บิต (5 ไบต์) ค่าที่มากกว่า 40 บิต ADDR_W จะไม่สามารถใช้ได้

บิตที่ 17 :CRC_L: ความยาว CRC ที่จะคำนวณใน Shock Burst

Logic 0: 8 bit CRC

Logic 1: 16 bit CRC

บิตที่ 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

ข้อสังเกต : บิต 8 CRC อาจเพิ่มจำนวน payload ใน Shock Burst ได้แต่จะลดความสมบูรณ์ของระบบลง

2.3.7.1 การตั้งค่าเกี่ยวกับอุปกรณ์

การตั้งค่าในส่วนนี้จะมีความสัมพันธ์กับโมดูลความถี่วิทยุและอุปกรณ์ดังนี้

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

รูปที่ 2.34 การเซ็ตการทำงานของโมดูลความถี่วิทยุ

บิตที่ 15 : Rx2_EN

ตำแหน่งบิตที่ 15 เป็นบิตที่เลือกจำนวนช่องสัญญาณของภาครับ

สถานะ 0: จำนวน 1 ช่องสัญญาณ

สถานะ 1: จำนวน 2 ช่องสัญญาณ

บิตที่ 14 : Communication Mode (CM)

ตำแหน่งบิตที่ 14 เป็นบิตที่เลือกโหมดการส่งข้อมูลกำหนดให้มีสถานะดังนี้

สถานะ 0: ทำงานใน Direct Mode

สถานะ 1: ทำงานใน Shock Burst Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตที่ 13 : RF Data Rate (RFDR_SB)

ตำแหน่งบิตที่ 13 โดยเป็นบิตที่เลือกค่าอัตราการส่งผ่านข้อมูลของโมดูลความถี่วิทยุ

สถานะ 0: อัตราการส่งผ่านข้อมูลเท่ากับ 250 กิโลบิตต่อวินาที

สถานะ 1: อัตราการส่งผ่านข้อมูลเท่ากับ 1 เมกะบิตต่อวินาที

ข้อสังเกต : ใช้ 250 kbps แทน 1 Mbps จะช่วยปรับปรุง sensitivity ของตัวรับ 10 db 1Mbps ต้องใช้ crystal 16 MHz

บิตที่ 10-12 : XO_F

เลือกความถี่ crystal ที่จะใช้

ตารางที่ 2.12 การเซ็ตความถี่คริสตอล

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

บิตที่ 8-9 : RF_PWR

เซ็ตกำลังของเอาท์พุทในโหมดการส่งดังตาราง

ตารางที่ 2.13 RF การเซ็ตกำลังเอาท์พุท

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.7.2 การตั้งค่าความถี่ช่องสัญญาณและทิศทางการส่งข้อมูล

RF_CH#							RXEN
7	6	5	4	3	2	1	0

รูปที่ 2.35 ช่องความถี่ และ การกำหนดโหมดการทำงาน Rx / Tx

บิตที่ 1-7 : RF_CH#

เป็นช่วงบิตข้อมูลที่กำหนดความถี่ที่ใช้รับหรือส่งซึ่งมีความถี่ในช่วง 2400 เมกะเฮิรตซ์ (MHz) ถึง 2524 เมกะเฮิรตซ์โดยคำนวณได้จากสูตรนี้

ช่องสัญญาณ โมดูลความถี่วิทยุ = 2400 MHz + [(RF_CH#)(1.0 MHz)]

RF_CH#							RXEN
1	1	0	0	1	0	0	0

รูปที่ 2.36 ตัวอย่างการกำหนดค่าความถี่ให้กับตัวส่ง ความถี่ 2500 เมกะเฮิรตซ์

RF_CH#							RXEN
1	1	0	0	1	0	0	1

รูปที่ 2.37 ตัวอย่างการกำหนดค่าความถี่ให้กับตัวรับ ความถี่ 2500 เมกะเฮิรตซ์

ข้อสังเกต : ในกรณีที่ส่งความถี่ช่องสัญญาณจำนวน 2 ช่องสามารถคำนวณหาความถี่โดยใช้สูตรดังนี้

ช่องสัญญาณความถี่ = 2400 เมกะเฮิรตซ์ + [(RF_CH#)(1.0 เมกะเฮิรตซ์) + 8 เมกะเฮิรตซ์

RF_CH#: อาจตั้งไว้ระหว่าง 2408 MHz และ 2524 MHz

บิตที่ 0 : RXEN

เป็นตำแหน่งบิตที่ 0 ใช้กำหนดค่าของโมดูลความถี่วิทยุโดยกำหนดดังนี้

สถานะ 0: โหมดส่ง

สถานะ 1: โหมดรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 TCP/IP โปรโตคอล

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้ TCP/IP เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน 3 ประการคือ

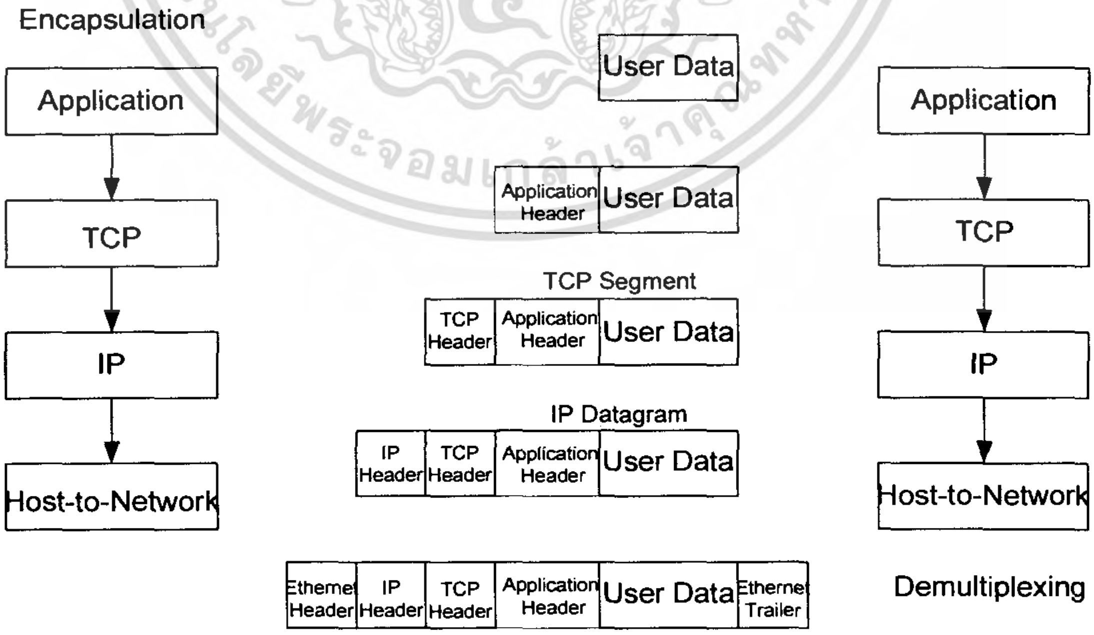
- เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน

- ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ

- มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง และข้อมูล

2.4.1 Encapsulation/Demultiplexing

การส่งข้อมูลผ่านแต่ละเลเยอร์ โดยแต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมากับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่าเฮดเดอร์ ภายในเฮดเดอร์จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือโปรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็นเฮดเดอร์ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing ดังแสดงขั้นตอนในรูปที่ 2.38



รูปที่ 2.38 ขั้นตอนการ Encapsulation และ Demultiplexing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละเลเยอร์มีชื่อเรียกแตกต่างกัน ดังนี้

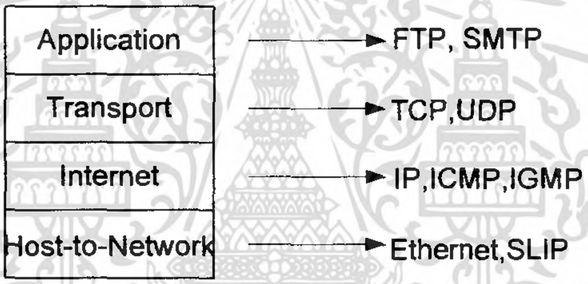
ข้อมูลที่มาจาก User หรือก็คือข้อมูลที่ User เป็นผู้ป้อนให้กับแอปพลิเคชั่น เรียกว่า User Data เมื่อแอปพลิเคชั่นได้รับข้อมูลจาก User ก็จะนำมาประกอบกับส่วนหัวของแอปพลิเคชั่น เรียกว่า แอปพลิเคชั่นดาต้า (Application Data) และส่งต่อไปยังโปรโตคอล TCP

เมื่อโปรโตคอล TCP ได้รับแอปพลิเคชั่นดาต้าก็จะนำมารวมกับเฮดเดอร์ของโปรโตคอล TCP เรียกว่า ทีซีพีเซกเมนต์ (TCP Segment) และส่งต่อไปยังโปรโตคอล IP

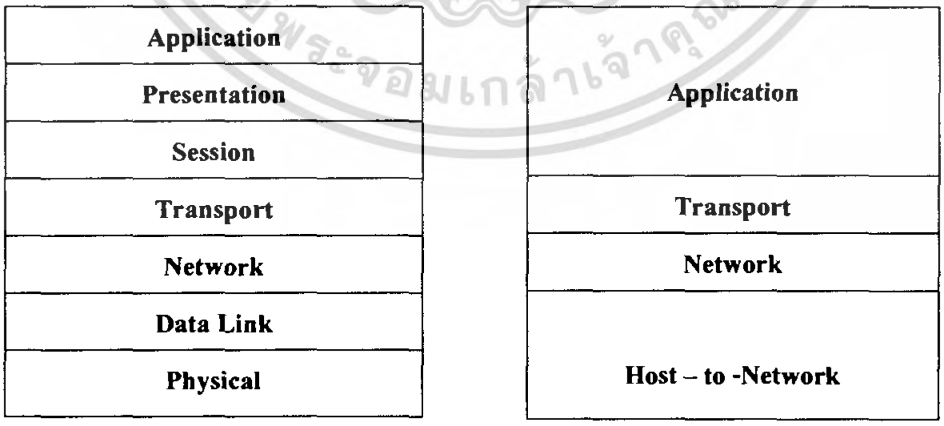
เมื่อโปรโตคอล IP ได้รับทีซีพีเซกเมนต์ก็จะนำมารวมกับเฮดเดอร์ของโปรโตคอล IP เรียกว่า ไอพีดาต้าแกรม (IP Datagram) และส่งต่อไปยังชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

ในระดับชั้นโฮสต์-เครือข่ายจะนำไอพีดาต้าแกรมมาเพิ่มส่วน Error Correction และ flag เรียกว่า อีเทอร์เน็ตแฟรม (Ethernet Frame) ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

ในแต่ละเลเยอร์ของโครงสร้าง TCP/IP สามารถอธิบายได้ดังรูปที่ 2.39 ซึ่งเมื่อเปรียบเทียบกับโครงสร้าง OSI แสดงได้ดังรูปที่ 2.40



รูปที่ 2.39 โครงสร้าง TCP/IP



รูปที่ 2.40 เปรียบเทียบ OSI Model กับ TCP/IP Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ชั้นโฮสต์-เครือข่าย

หรือ Network Access Layer เป็นการรวม Physical Layer และ Data Link Layer ของแบบจำลอง OSI เข้าด้วยกัน ใช้โปรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มา แล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูล ทางด้านผู้รับก็จะทำงานในทางกลับกัน คือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร 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

MAC Address

MAC Address เป็นตัวบ่งชี้ลักษณะเฉพาะของแต่ละ machine ดังนั้นจึงต้องเป็นค่าที่ไม่ซ้ำกัน (unique) MAC Address เป็นเลข 48 bits โดยแบ่งออกเป็น 2 ส่วน โดย 24 bits แรกเป็นค่าที่แสดงถึงบริษัทที่ผลิตการ์ดนั้น ๆ ส่วน 24 bits หลังเป็น serial number ที่ทางบริษัทกำหนดให้ ซึ่งแต่ละตัวต้องไม่ซ้ำกันระบบ เช่น MAC Address = 00 17 A4 E3 C5 C6 ซึ่งจะเห็นว่า 3 byte แรกคือ 00, 17, A4 แสดงถึงบริษัทที่ผลิตการ์ดและ 3 byte หลังเป็น serial number

2.4.3 ชั้นการสื่อสารอินเทอร์เน็ต

เป็นระดับชั้นที่ทำหน้าที่เหมือน Network Layer ของแบบจำลอง OSI ทำหน้าที่หาเส้นทางการส่งข้อมูลให้ถึงปลายทางได้อย่างถูกต้องและรวดเร็ว โปรโตคอลในระดับชั้นนี้ถูกออกแบบมาให้ส่งข้อมูลแบบ Connectionless ซึ่งไม่รับประกันความถูกต้องของข้อมูล แต่ละแพ็กเก็ตสามารถส่งผ่านได้อย่างอิสระ เมื่อแพ็กเก็ตถึงปลายทางอาจไม่เรียงตามลำดับ โปรโตคอลที่ใช้หลักๆ คือ IP

- IP

IP เป็นโปรโตคอลในระดับ Network Layer ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูล และควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็กเก็ต ซึ่งกลไกในการหาเส้นทางของ IP จะมีความสามารถในการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล และมีระบบการแยกและประกอบดาต้าแกรม เพื่อรองรับการส่งข้อมูลระดับดาต้าลิ้ง ที่มีขนาด MTU (*Maximum Transmission Unit*) ที่แตกต่างกัน ทำให้สามารถนำ IP ไปใช้บนโปรโตคอลอื่นได้หลากหลาย เช่น Ethernet, Token Ring หรือ Apple Talk

การเชื่อมต่อของ IP เพื่อทำการส่งข้อมูล จะเป็นแบบ connectionless หรือเกิดเส้นทางการเชื่อมต่อในทุกๆ ครั้งของการส่งข้อมูล 1 ดาต้าแกรม โดยจะไม่ทราบถึงข้อมูลดาต้าแกรมที่ส่งก่อนหน้าหรือส่งตามมา แต่การส่งข้อมูลใน 1 ดาต้าแกรม อาจจะเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (fragmentation) และถูกนำไปรวมเป็นดาต้าแกรมเดิมเมื่อถึงปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4-bit Version	Header Length	8-bit Type of Service	16-bit Total Length in Byte	
16-bit Identification			3-bit Flag	16-bit Fragment Checksum
8-bit Time to Live(TTL)		8 -bit Protocol	16-bit Header Checksum	
32-bit Source IP Address				
32-bit Destination IP Address				
Option				
Data				

รูปที่ 2.41 แสดง IP Header

จากรูปที่ 2.41 สามารถอธิบายรายละเอียดกล่าวคือ

เฮดเดอร์ของ IP โดยปกติจะมีขนาด 20 ไบต์ ยกเว้นในกรณีที่มีการเพิ่มออปชั่นบางอย่าง ฟิลด์ของเฮดเดอร์ IP จะมีความหมายดังนี้

Version: หมายเลขเวอร์ชันของโปรโตคอล ที่ใช้งานในปัจจุบันคือ เวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)

Header Length: ความยาวของเฮดเดอร์โดยทั่วไป ถ้าไม่มีส่วนออปชั่นจะมีค่าเป็น 5 (5×32 บิต)

Type of Service (TOS): ใช้เป็นข้อมูลสำหรับเราท์เตอร์ในการตัดสินใจเลือกการเราท์ข้อมูลในแต่ละดาต้าแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว

Length: ความยาวทั้งหมดเป็นจำนวนไบต์ของดาต้าแกรม ซึ่งด้วยขนาด 16 บิตของฟิลด์ จะหมายถึงความยาวสูงสุดของดาต้าแกรม คือ 65535 ไบต์ (64 Kb) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆ ตามขนาดของ MTU ที่กำหนดในลิงค์เลเยอร์ และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทาง แอปพลิเคชั่นส่วนใหญ่จะมีขนาดของดาต้าแกรมไม่เกิน 512 ไบต์

Identification: เป็นหมายเลขของดาต้าแกรมในกรณีที่มีการแยกดาต้าแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี Identification เดียวกันมารวมกัน

Flag: ใช้ในกรณีที่มีการแยกดาต้าแกรม

Fragment Offset: ใช้ในการกำหนดตำแหน่งข้อมูลในดาต้าแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง

Time to Live (TTL): กำหนดจำนวนครั้งที่มากที่สุดที่ดาต้าแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิกและเราท์เตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิด time out ในระหว่างการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Protocol: ระบุโปรโตคอลที่ส่งในดาต้าแกรม เช่น TCP, UDP หรือ ICMP

Header Checksum: ใช้ในการตรวจสอบความถูกต้องของข้อมูลในเฮดเดอร์

Source IP Address: หมายเลข IP ของผู้ส่งข้อมูล

Destination IP Address: หมายเลข IP ของผู้รับข้อมูล

Data: ข้อมูลจากโปรโตคอลระดับบน

กลไกของโปรโตคอล IP

ในการส่งผ่านข้อมูลหรือ IP datagram ไปยังเครือข่ายอินเตอร์เน็ตนั้น โปรโตคอล IP จะทำหน้าที่พิจารณาว่าปลายทางในการส่ง IP datagram นั้นเป็นภายในเครือข่ายตนเองหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยกรพิจารณานี้โปรโตคอล 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

- ICMP (Internet Control Message Protocol)

ICMP เป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานภาพของดาต้าแกรม ในกรณีที่เกิดปัญหากับดาต้าแกรม เช่น เราท์เตอร์ไม่สามารถส่งดาต้าแกรมไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยังโฮสต์ต้นทางเพื่อรายงานข้อผิดพลาดที่เกิดขึ้น อย่างไรก็ดี ไม่มีอะไรรับประกันได้ว่า ICMP Message ที่ส่งไปจะถึงผู้รับจริงหรือไม่ หากมีการส่งดาต้าแกรมออกไปแล้วไม่มี ICMP Message ฟ้อง Error กลับมา ก็แปลความหมายได้สองกรณีคือ ข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย หรืออาจจะมีปัญหา ในการสื่อสารทั้งการส่งดาต้าแกรม และ ICMP Message ที่ส่งกลับมาก็มีปัญหาระว่างทางก็ได้ ICMP จึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (unreliable) ซึ่งจะเป็นหน้าที่ของโปรโตคอลในระดับสูงกว่า Network Layer ในการจัดการให้การสื่อสารนั้นๆ มีความน่าเชื่อถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ ICMP Message จะประกอบด้วย Type ขนาด 8 บิต Checksum ขนาด 16 บิต และส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตาม Type และ Code ดังรูปที่ 2.42

8-bit Type	8 -bit Code	16-bit Checksum
ICMP Content		

รูปที่ 2.42 ICMP Header

- ARP

ARP (Address Resolution Protocol) เป็นโปรโตคอลสำหรับการจับคู่ (map) ระหว่าง Internet Protocol address (IP address) กับตำแหน่งของอุปกรณ์ในระบบเครือข่าย เช่น IP เวอร์ชัน 4 ใช้การระบุตำแหน่งขนาด 32 บิต ใน Ethernet ของระบบใช้การระบุ ตำแหน่ง 48 บิต (การระบุตำแหน่งของอุปกรณ์รู้จักในชื่อของ Media Access Control หรือ MAC address) ตาราง ARP ซึ่งมักจะเป็น cache จะรักษาการจับคู่ ระหว่าง MAC address กับ IP address โดย ARP ใช้กฎของโปรโตคอล สำหรับการสร้างการจับคู่และแปลงตำแหน่งทั้งสองฝ่าย

การทำงานของ ARP

เมื่อแพ็คเกตนำเข้าที่ระบุเครื่อง host ในระบบเครือข่ายมาถึง Gateway เครื่องที่ Gateway จะเรียกโปรแกรม ARP ให้หาเครื่อง host หรือ MAC address ที่ตรงกับ IP address โปรแกรม ARP จะหาใน ARP cache เมื่อพบแล้วจะแปลงแพ็คเกต เป็นแพ็คเกตที่มีความยาว และรูปแบบที่ถูกต้อง เพื่อส่งไปยังเครื่องที่ระบุไว้ แต่ถ้าไม่พบ ARP จะกระจายแพ็คเกตในรูปแบบพิเศษ ไปยังเครื่องทุกเครื่องในระบบและถ้าเครื่องใดเครื่องหนึ่งทราบว่ามี IP address ตรงกันก็จะตอบกลับมาที่ ARP โปรแกรม ARP จะปรับปรุง ARPและส่งแพ็คเกตไปยัง MAC address หรือเครื่องที่ตอบมา

เนื่องจากแต่ละโปรโตคอลมีรายละเอียดที่แตกต่างกันตามประเภทของ LAN ดังนั้นจึงมี การแยก ARP Request for Comments ตามประเภทของโปรโตคอลสำหรับ Ethernet, asynchronous transfer mode, Fiber Distributed-Data Interface, HIPPI และโปรโตคอลอื่น

ส่วน Reverse ARP สำหรับเครื่อง host ที่ไม่รู้จัก IP address นั้น RARP สามารถให้เครื่อง เหล่านี้ขอ IP address จาก ARP cache ของ Gateway

2.4.4 ชั้นสื่อสารนำส่งข้อมูล

เป็นระดับชั้นที่ทำหน้าที่เหมือน Transport Layer ของแบบจำลอง OSI ทำหน้าที่ควบคุมการส่งข้อมูลระหว่างผู้ส่งและผู้รับมีชื่อเรียกอีกอย่างว่าHost-To-Host Layer ประกอบด้วย 2 โปรโตคอล ประกอบด้วย TCP(Transmission Control Protocol) ให้บริการส่งข้อมูลแบบ Connection-Oriented โดยส่งข้อมูลแบบ Byte Stream จากฝั่งส่งไปยังฝั่งรับผ่านเครือข่ายได้อย่างถูกต้อง มีความน่าเชื่อถือ โดยหน่วยของข้อมูลเรียกว่า เซกเมนต์ และอีกโปรโตคอล คือ UDP(User Datagram Protocol) ให้บริการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ Connectionless โดยไม่ต้องสร้างการเชื่อมต่อก่อน ไม่เน้นความถูกต้องของข้อมูล เน้นความรวดเร็ว โดยหน่วยของข้อมูลเรียกว่า ดาต้าแกรม

- UDP: (User Datagram Protocol)

เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรม และจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล

กลไกการตรวจสอบโดย checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง, หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของ UDP เอง จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด

16-bit Source Port	16-bit Destination Port
Length	Checksum
Data	

รูปที่ 2.43 UPD Header

จากรูปที่ 2.43 สามารถอธิบายรายละเอียดได้ดังนี้

Source Port Number: หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้

Destination Port Number: หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

UDP Length: ความยาวของดาต้าแกรม ทั้งส่วนเฮดเดอร์ และ ดาต้า นั่นหมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของเฮดเดอร์

Checksum: เป็นตัวตรวจสอบความถูกต้องของ UDP Datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 แสดงได้ดังตารางที่ 2.14

ตารางที่ 2.14 แสดง UDP Port Number

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)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TCP: (Transmission Control Protocol)

อยู่ใน Transport Layer เช่นเดียวกับ UDP ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล ซึ่งมีความสามารถและรายละเอียดมากกว่า UDP โดยดาต้าแกรมของ TCP จะมีความสัมพันธ์ต่อเนื่องกัน และมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง และมีการสื่อสารแบบ Connection-oriented ดังแสดงในรูปที่ 2.44

<table>
<tr><td colspan="8">16-bit Source Port Number</td><td>16-bit Destination Port</td></tr>
<tr><td colspan="9">32-bit sequence Number</td></tr>
<tr><td colspan="9">32-bit Ackowledge Number</td></tr>
<tr><td>Header Length</td><td>6-bit Reserved</td><td>URG</td><td>ACK</td><td>PUSH</td><td>RESET</td><td>SYN</td><td>FIN</td><td>16-bit Windows Size</td></tr>
<tr><td colspan="8">16-bit TCP Checksum</td><td>16-bit Urgent Pointer</td></tr>
<tr><td colspan="9">TCP Option</td></tr>
<tr><td colspan="9">Data</td></tr>
</table>

รูปที่ 2.44 TCP Header

จากรูปที่ 2.44 สามารถกล่าวในรายละเอียด ดังนี้

Source Port Number: หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้

Destination Port Number: หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

Sequence Number: ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง

Acknowledgment Number: ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ

Header Length: โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ออปชั่น แต่ต้องไม่เกิน 60 ไบต์

Flag: เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังตารางที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.15 Flag ชนิดต่างๆ

Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent Pointer)
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (reset) เนื่องจากในกรณีที่เกิดการสับสนในกรณีต่างๆ เช่น โฮสต์มีปัญหา ให้เริ่มต้นการสื่อสารกันใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

Flag ในเฮดเดอร์ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP Segment เนื่องจากข้อมูลในเฮดเดอร์ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน flag จะเป็นตัวกำหนดว่าให้ใช้งานฟิลด์ไหน เช่น ฟิลด์ Acknowledgment number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลในฟิลด์ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ ซึ่งถ้าไม่มี flag เป็นตัวกำหนดก็อาจจะมีการนำข้อมูลมาใช้ และก่อให้เกิดความผิดพลาดได้

2.4.5 ชั้นสื่อสารการประยุกต์

ชั้นสื่อสารการประยุกต์ เมื่อนำมาเทียบกับมาตรฐานของ OSI จะทำงาน 2 หน้าที่เทียบได้กับ Application Layer และ Presentation Layer ของ OSI ในชั้นนี้จะรองรับการทำงานของแอปพลิเคชั่นต่าง ๆ อย่างเช่น เมื่อมีเครื่อง Client ขอใช้บริการเพื่อจะติดต่อขอดาวน์โหลดไฟล์ผ่านทางอินเตอร์เน็ต โดยอาจจะเรียกใช้โปรแกรม FTP Client ทั่วไป อย่างเช่นโปรแกรม WS_ftp เพื่อติดต่อกับโปรเซส FTP ที่กำลังให้ บริการอยู่ที่เครื่องเซิร์ฟเวอร์ จากนั้นตัวโปรเซส FTP ก็จะเรียกใช้โปรโตคอล FTP เพื่อทำการโอนถ่ายไฟล์นี้ไปให้เครื่อง Client เป็นต้น หรือถ้าผู้ใช้ต้องการเรียกใช้งานคอมพิวเตอร์จากเครื่องที่อยู่ห่างไกลออกไป ด้วยการใช้โปรแกรม Telnet ที่เครื่องเซิร์ฟเวอร์ให้บริการ ตัวโปรเซส Telnet ที่ทำงานอยู่ก็จะเรียกใช้โปรโตคอล Telnet เพื่อติดต่อกัน หรือกรณีที่มีการเรียกใช้โปรแกรม Web Browser เช่น Internet Explorer เพื่อเรียกดูเว็บเพจในเว็บไซต์ ของ NASA ที่เครื่องให้บริการเว็บของ NASA ก็จะมีโปรเซส HTTP (Hyper Text Transfer Protocol) ทำงานอยู่และจะติดต่อกับผู้ใช้ผ่านโปรโตคอล HTTP เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของแอปพลิเคชั่นต่าง ๆ จะอยู่ที่ชั้นสื่อสารการประยุกต์นี้ และมีการติดต่อกันตามแต่ละโปรโตคอลเฉพาะแล้วแต่แอปพลิเคชั่นที่ใช้งาน จากการที่ชั้นสื่อสารการประยุกต์ของ TCP/IP รองรับให้โปรโตคอลอื่นทำงานได้หลายโปรเซส และหลายโปรโตคอลได้พร้อมกันนั้น ทำให้ผู้ใช้สามารถเปิดโปรแกรมใช้งานได้หลาย ๆ อย่าง พร้อมกัน เช่น เปิดโปรแกรม Internet Explorer เพื่อเรียกดูเว็บเพจพร้อมกับ

ใช้งานโปรแกรม Outlook Express เพื่อรับส่ง E-Mail ไปพร้อม ๆ กัน ได้โดยไม่ต้องรอให้ทำงานอย่างใดอย่างหนึ่งเสร็จไปก่อน หรือในปัจจุบันมีการพัฒนาโปรแกรม Web Browser ให้สามารถเรียกใช้งานโปรโตคอลอื่น ๆ ได้มากขึ้น ทำให้เราสามารถใช้โปรแกรม Web Browser โอนถ่ายไฟล์ข้อมูลที่ใช้โปรโตคอล FTP ได้โดยไม่ต้องไปหาโปรแกรมอื่น ๆ มาใช้เพิ่มเติมอีก

โปรโตคอล ที่ทำงานและให้บริการก็มี อย่างเช่น

- HTTP

HTTP ใช้ในการติดต่อรับส่งข้อมูลชนิดไฮเปอร์เท็กซ์ (Hypertext) ระหว่างเครื่องลูกข่ายกับ WWW เซิร์ฟเวอร์ (World Wide Web) โดยที่เอกสารนี้จะอยู่ในรูปแบบที่เขียนในภาษา HTML (HyperText Markup Language) เอกสารแต่ละชิ้น จะสามารถเชื่อมโยงไปยังเอกสารชิ้นอื่นได้ ซึ่งเอกสารที่ถูกเชื่อมโยงนี้ อาจจะอยู่บนเครื่องคอมพิวเตอร์เครื่องเดียวกัน หรือต่างเครื่องกันก็ได้

- FTP

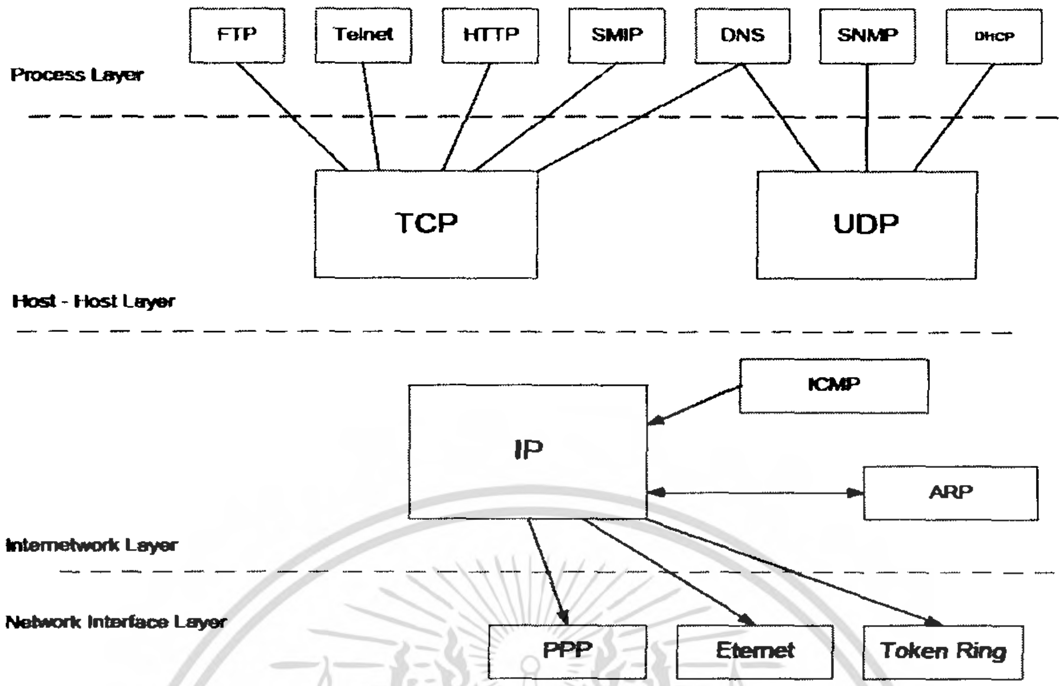
FTP ใช้ในการรับ-ส่งแฟ้มข้อมูลระหว่างเครื่องลูกข่ายและเครื่องเซิร์ฟเวอร์ โดยที่เครื่องเซิร์ฟเวอร์ จะต้องมีโปรแกรมให้บริการ FTP (FTP Server) ติดตั้งและทำงานอยู่เพื่อให้เครื่องลูกข่ายที่รัน (Run)โปรแกรม FTP Client สามารถเข้ามาขอใช้บริการได้

- Protocol DNS

ทำหน้าที่แปลงข้อมูลชื่อ Domain Name หรือ ชื่อเว็บไซต์ ทั้งหลายให้เป็นหมายเลข IP Address โปรโตคอล UDP ทำหน้าที่นำส่งข้อมูลจากโปรโตคอลประยุกต์ไปยัง IP ข้อมูลรวบรวม UDP Header เรียกว่า UDP Datagram หรือ User Datagram

จากเลเยอร์ทั้ง 4 ชั้นของโมเดล TCP/IP สามารถสรุปเป็นโปรโตคอลได้ดังรูปที่ 2.45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.45 แสดงโปรโตคอลที่ใช้งานในโมเดล TCP/IP

2.4.6 ระบบของ Ethernet Frame

หัวใจสำคัญของระบบ Ethernet ได้แก่ เฟรมข้อมูลทางข่าวสารและอุปกรณ์ทางฮาร์ดแวร์ที่ใช้เชื่อมต่อสื่อสารบนเครือข่าย ซึ่งได้แก่ การ์ด Ethernet LAN สายสัญญาณและอุปกรณ์เสริมอื่นๆ ที่จะช่วยนำพาข้อมูลในรูปแบบของบิตทางดิจิตอล ที่เรียกว่าเฟรมวิ่งไปมาระหว่างคอมพิวเตอร์บนเครือข่าย

เฟรมข้อมูลสำหรับระบบ Ethernet ประกอบขึ้นด้วยกลุ่มของบิตที่เป็นข้อมูลและข่าวสารสำคัญแบ่งออกเป็นขนาดสัดส่วนที่แน่นอนที่เรียกว่าช่อง Field ดังรูปที่ 2.46

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

รูปที่ 2.46 ลักษณะโครงสร้างของเฟรมข้อมูล

2.4.7 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 สำหรับตรวจสอบความผิดพาดข้อมูล หรือ Frame Check Sequence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

3. เฟรมที่มีขนาดเล็กที่สุด 64 byte จะต้องใช้เวลาอยู่ที่ 51.2 ไมโครวินาที

2.5 ระบบฝังตัว

ระบบฝังตัว คือ ระบบคอมพิวเตอร์ขนาดเล็กที่บรรจุอยู่ภายในอุปกรณ์เครื่องใช้ไฟฟ้า หรือเครื่องมืออิเล็กทรอนิกส์ต่างๆ เพื่อเพิ่มความสามารถให้กับอุปกรณ์เหล่านี้ให้ทำงานได้อย่างมีประสิทธิภาพมากกว่าอุปกรณ์ทั่วไป ในการพัฒนาระบบฝังตัวนี้จะต้องพัฒนาทางด้านซอฟต์แวร์และฮาร์ดแวร์ร่วมกัน ซึ่งเทคโนโลยีของระบบฝังตัวสามารถแบ่งออกได้เป็น 3 ระดับ ดังนี้

- ระบบฝังตัวขนาดเล็ก เป็นไมโครโปรเซสเซอร์ขนาดเล็ก จะใช้ไมโครคอนโทรลเลอร์ที่มีหน่วยความจำและส่วนติดต่ออินพุต/เอาท์พุต รวมทั้งส่วนประกอบอื่นๆ ที่จำเป็นรวมอยู่ภายในตัวไมโครคอนโทรลเลอร์เองเพื่อความสะดวกในการพัฒนา ระบบฝังตัวชนิดนี้นิยมใช้ในงานควบคุมอุปกรณ์อิเล็กทรอนิกส์พื้นฐานที่มีการทำงานไม่ซับซ้อน มีขนาด 4 บิต หรือ 8 บิต หน่วยความจำประมาณ 10-120 กิโลไบต์ มีพอร์ตอินพุต/เอาท์พุต 1-4 พอร์ต สามารถติดต่อสื่อสารแบบอนุกรมหรือขนานได้ทันที ตัวอย่างของไมโครคอนโทรลเลอร์เหล่านี้ได้แก่ Z80 , MCS-51 และ PIC เป็นต้น

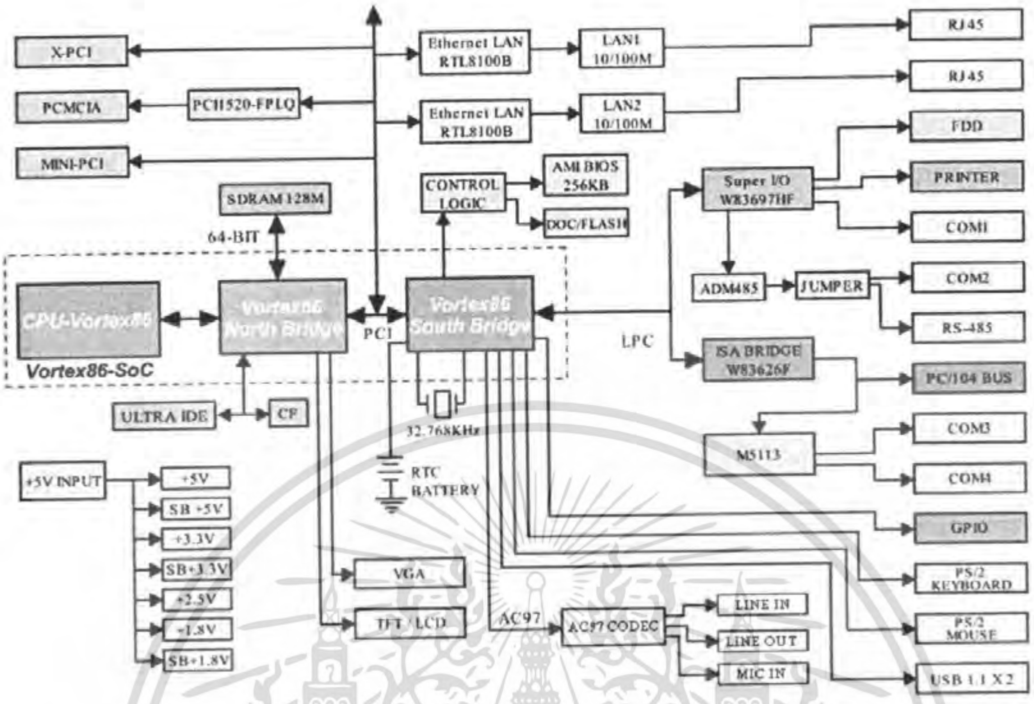
- ระบบฝังตัวขนาดกลาง เป็นไมโครโปรเซสเซอร์ที่มีความสามารถสูงขึ้น มีหน่วยความจำมากขึ้น มีความเร็วในการประมวลผลสูงขึ้น มักใช้ในงานที่ต้องการความสามารถพิเศษซึ่งไมโครโปรเซสเซอร์ขนาดเล็กไม่สามารถทำงานได้ ไมโครโปรเซสเซอร์เหล่านี้มักมีขนาด 16 บิต และ 32 บิต ดังเช่น ตระกูล x86 ของบริษัท Intel และตระกูล ARM7 และ TMS320 ของบริษัท AMD เป็นต้น ไมโครโปรเซสเซอร์เหล่านี้มักมีคุณสมบัติพิเศษเฉพาะทาง เช่น การสื่อสารหรือการประมวลผลชนิดพิเศษ

- ระบบฝังตัวขนาดใหญ่ เป็นไมโครโปรเซสเซอร์ที่มีความสามารถในการประมวลผลมากเป็นพิเศษ ส่วนใหญ่จะเป็นระบบฝังตัวที่ใช้ไมโครโปรเซสเซอร์ที่ได้รับการออกแบบมาสำหรับประมวลผลเป็นหลัก หรืออาจจะเป็นระบบฝังตัวที่ใช้คอมพิวเตอร์เป็นพื้นฐาน(PC-Based) อาจจะใช้ระบบปฏิบัติการพื้นฐานไม่ว่าจะเป็นระบบ DOS หรือ Linux ในการทำงาน เพื่อให้ง่ายในการพัฒนาและบำรุงรักษาซอฟต์แวร์ เนื่องจากสามารถเพิ่มเติมหรือแก้ไขซอฟต์แวร์ได้โดยง่ายและรวดเร็วกว่า ที่สำคัญคือ ออกแบบให้ทำงานหลายๆ อย่างพร้อมกันได้ ระบบฝังตัวประเภทนี้มักจะนำไปประยุกต์ใช้งานกับอุปกรณ์สำหรับระบบเครือข่าย เช่น เราท์เตอร์ อินเตอร์เน็ต หรือเว็บเซิร์ฟเวอร์ เป็นต้น

2.5.1 สถาปัตยกรรม Vortex86

Vortex86 เป็นชิพเซ็ตที่รวมตัวประมวลผล อย่างเช่น X86 เข้าด้วยกัน โดยมีส่วนประกอบต่างๆ ตามรูปที่ 2.45 ซึ่งจะมีอินพุตและเอาต์พุตต่างๆ ใกล้เคียงกับเครื่องคอมพิวเตอร์ทั่วไป ไม่ว่าจะเป็น COM Port , LAN Interface , Audio Engine , PCI , ATA หรือแม้กระทั่งส่วนของการต่อคีย์บอร์ด เมาส์ ส่วนการแสดงผล เมื่อระบบทำงานด้วยระบบปฏิบัติการจะทำให้เราสามารถพัฒนาแอพพลิเคชั่นได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

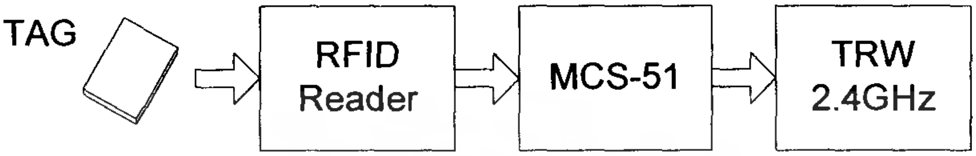


รูปที่ 2.47 สถาปัตยกรรม Vortex 86

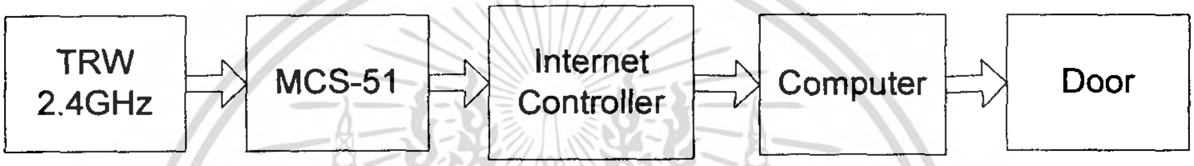
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการออกแบบ

บทนี้ได้แสดงถึงการออกแบบการใช้งานของวงจร โดยแบ่งเป็นส่วนต่างๆตามบล็อคไดอะแกรมได้ดังแสดงในรูปที่ 3.1



(ก) ด้านส่ง



(ข) ด้านรับ

รูปที่ 3.1 แสดงบล็อคไดอะแกรมของโครงงานทั้งหมด

3.1 วงจรรับข้อมูลจาก RFID CARD

หลักการออกแบบวงจร

ในส่วนนี้ประกอบด้วยอินพุตที่มาจากบัตร RFID และส่วนที่เป็นอุปกรณ์ wireless ซึ่งอุปกรณ์ที่ใช้คือโมดูล TRW 2.4 GHz โดยใช้ไมโครคอนโทรลเลอร์ เป็นตัวควบคุมการส่งผ่านข้อมูล สำหรับโปรแกรมที่ใช้ในการเขียนเพื่อควบคุมการทำงานจะใช้ภาษาแอสเซมบลี รายละเอียดของอุปกรณ์ทั้งหมดมีดังนี้

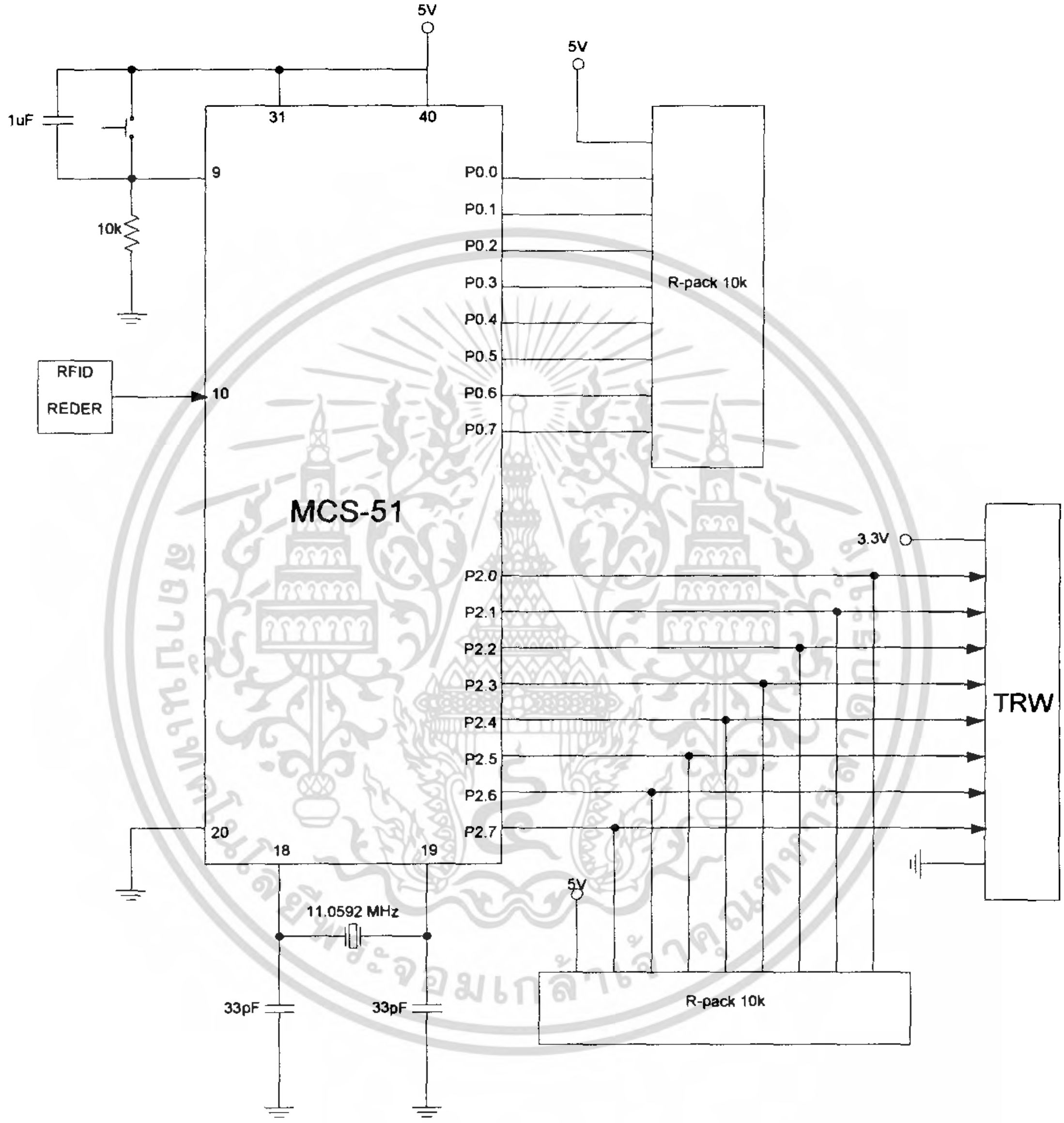
- บัตร RFID
- เครื่องอ่านบัตร RFID ชนิดแพสซิฟ
- ไมโครคอนโทรลเลอร์ เบอร์ 89C51
- โมดูล TRW 2.4 GHz
- วงจร Regulator ไฟเลี้ยง 3.3 v
- วงจร Regulator ไฟเลี้ยง 5 v

หลักการทำงาน

ในวงจรนี้จะ ทำการรับข้อมูลจากหัวอ่าน RFID ที่ขา 10 ซึ่งไว้ใช้สำหรับส่งข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ เบอร์ 89C51 โดยทำการรับข้อมูลทีละ 8 บิต เมื่อรับข้อมูลเข้ามาแล้ว ทำการ

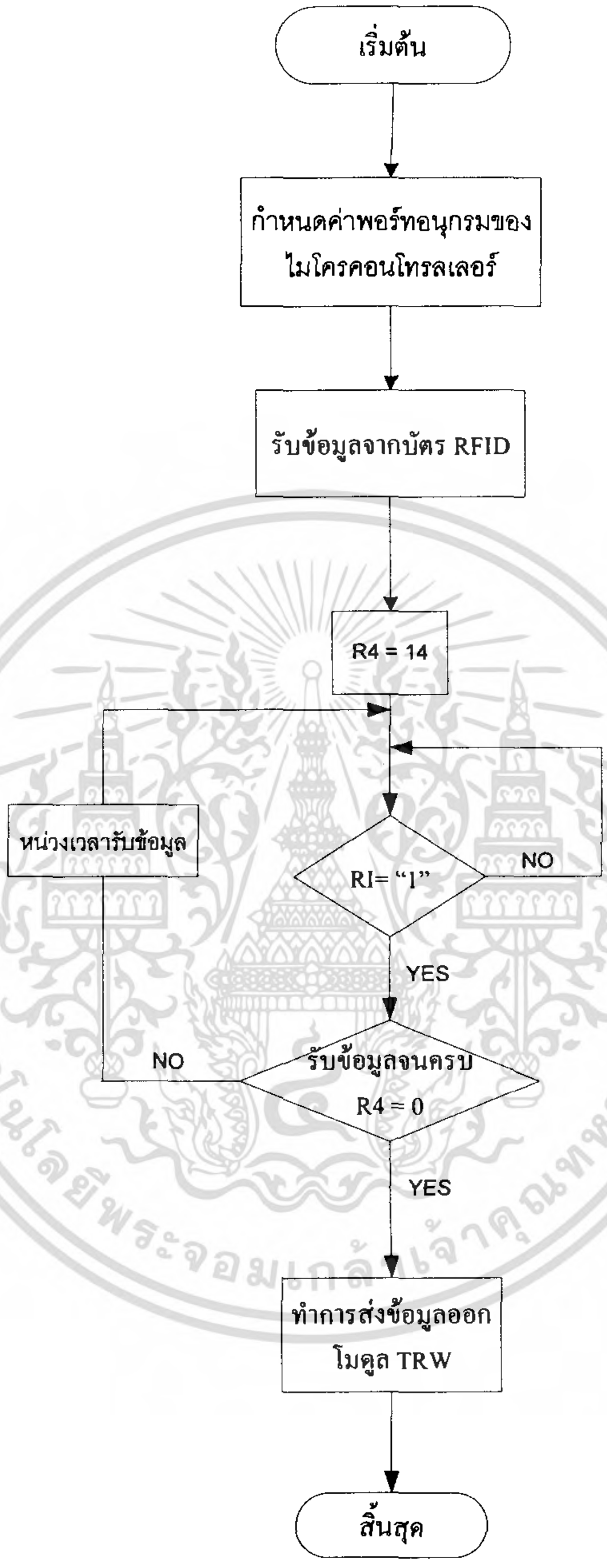
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บในรีจิสเตอร์ก่อน ข้อมูลที่เข้ามาจะถูกป้อนข้อมูลส่วนของ header เข้าไปที่ส่วนหัวของแพ็กเกจแล้วจึงตามด้วยข้อมูลจากบัตร RFID ซึ่งถูกกำหนดโดยโปรแกรม จากนั้นข้อมูลนำส่งออกทาง TRW 2.4 GHz ทางพอร์ต 2 ของไมโครคอนโทรลเลอร์ โมดูล TRW-2.4 GHz จะใช้สายอากาศแบบไมโครสติปส์ในการส่งข้อมูลไปยังด้านรับต่อไป วงจรทางด้านส่งแสดงได้ดังรูปที่ 3.2



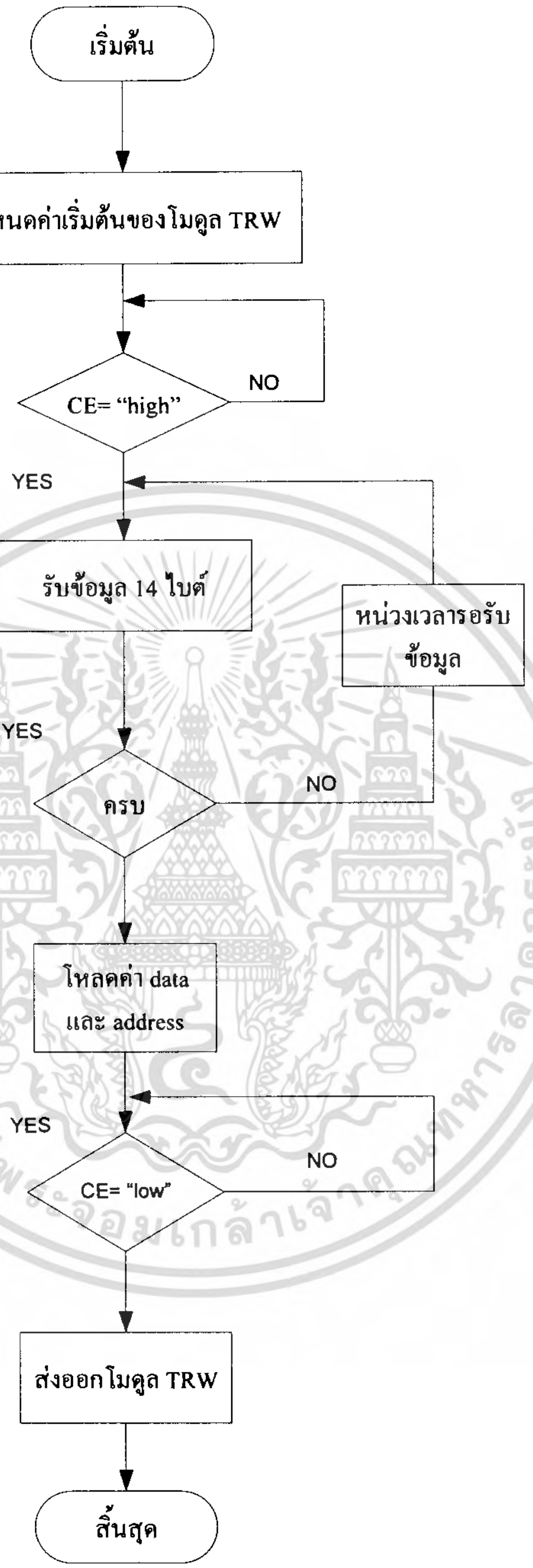
รูปที่ 3.2 แสดงวงจรรับข้อมูลจาก RFID แล้วส่งข้อมูลต่อผ่านโมดูล TRW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงโฟล์วชาร์ตการทำงานของเครื่องอ่าน RFID กับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



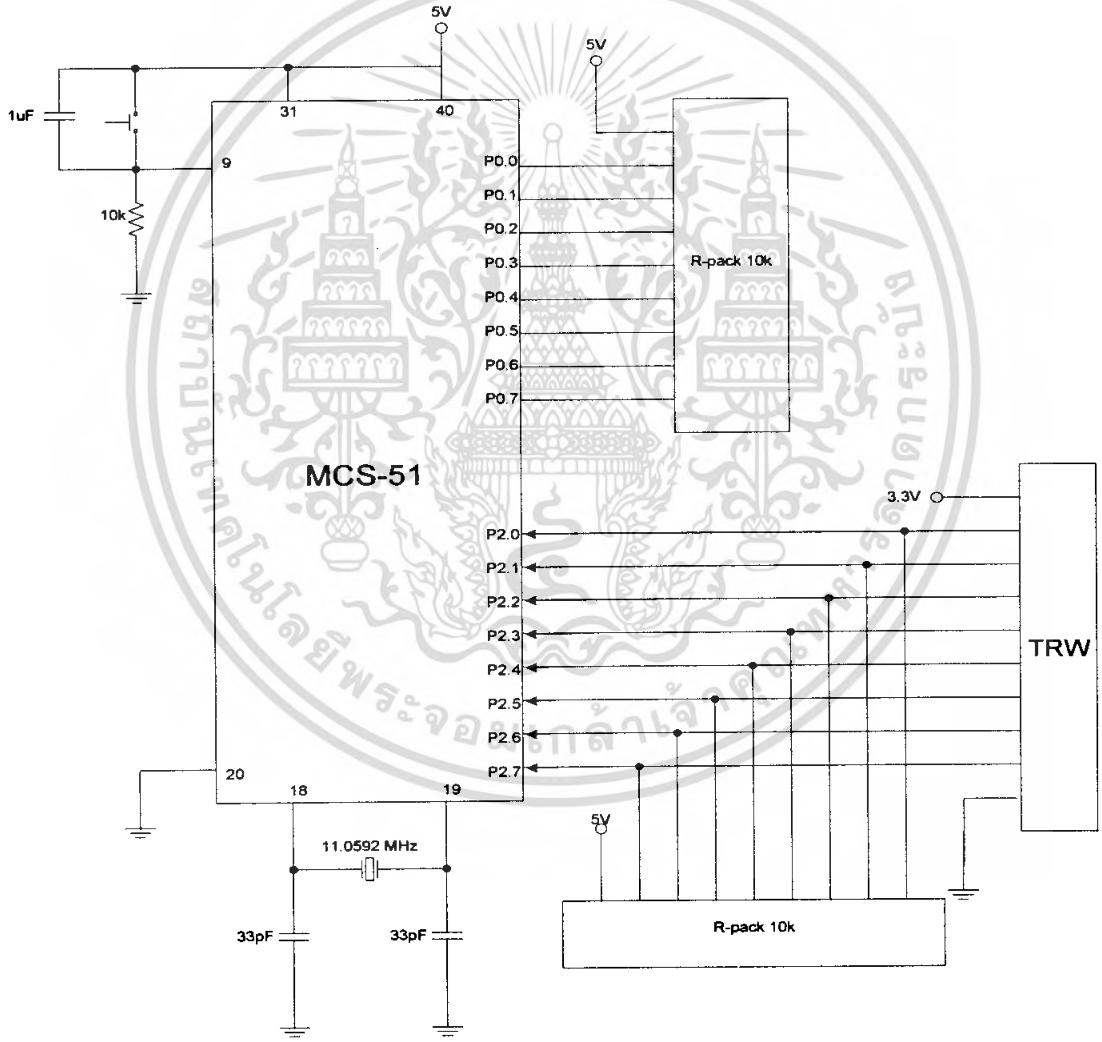
รูปที่ 3.4 แสดงโฟล์วชาร์ตการทำงานของโมดูล TRW 2.4 GHz กับไมโครคอนโทรลเลอร์ทางด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจรรับข้อมูลที่ส่งมาจากโมดูล TRW

สำหรับวงจรในส่วนนี้จะมีหลักการออกแบบคล้ายกับวงจรส่งข้อมูลออกโมดูล TRW เนื่องจากการส่งข้อมูลแบบนี้ใช้โมดูล TRW-2.4 GHz ตัวเดียวกัน โดยใช้ไมโครคอนโทรลเลอร์ ในการควบคุมเช่นเดียวกับในวงจรด้านส่ง เบอร์ที่ใช้ก็เป็น 89C51 เช่นเดียวกัน โปรแกรมที่เขียนควบคุมจะทำการรับข้อมูลเข้ามาเป็นแพ็กเกจ หลังจากนั้นข้อมูลจะถูกส่งเข้าไปที่พอร์ต 2 ของไมโครคอลโทรลเลอร์ ซึ่งข้อมูลในส่วนนี้จะถูกตัดส่วนของ header ซึ่งเป็นส่วนที่ระบุเกี่ยวกับคุณสมบัติของตัวโมดูลทิ้งไป ทำให้เหลือข้อมูลที่ถูกต้องจากบัตร RFID

หลังจากนั้นไมโครคอนโทรลเลอร์จะเป็นตัวเก็บข้อมูลไว้ชั่วคราว ก่อนที่จะส่งไปยังวงจร Ethernet controller ทางซีเรียลพอร์ตต่อไป



รูปที่ 3.5 แสดงวงจรรับข้อมูล Wireless ผ่านโมดูล TRW 2.4 GHz

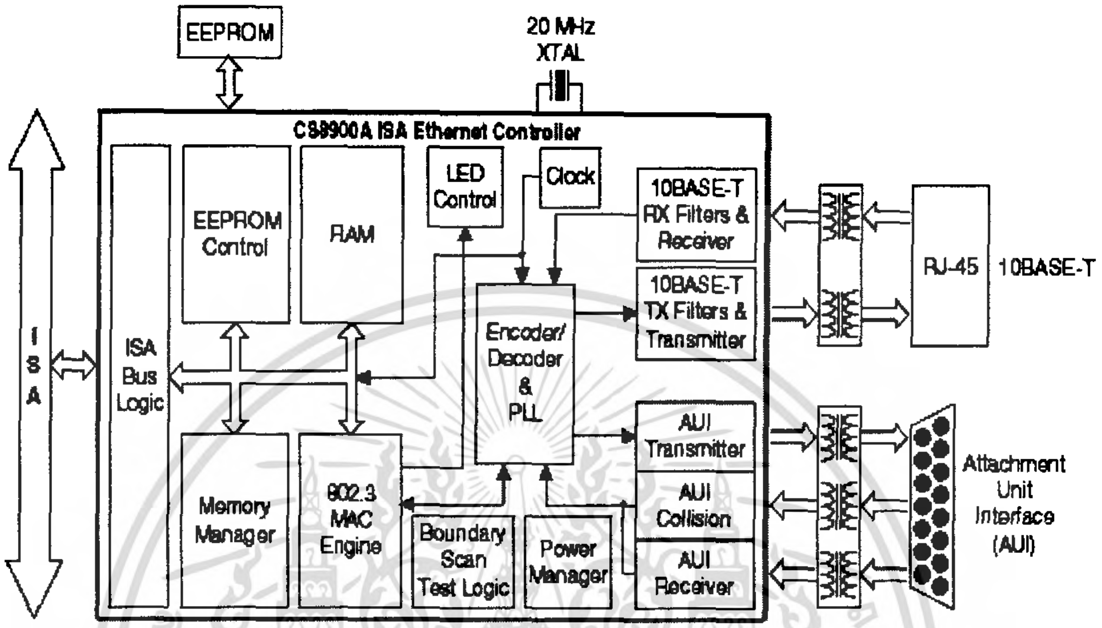
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 แสดงโฟล์วชาร์ตการทำงานของโมดูล TRW 2.4 GHz กับไมโครคอนโทรลเลอร์ทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจร Ethernet Controller

วงจร Ethernet Controller อาศัยการทำงานของ Embedded module ซึ่งประกอบด้วย ไอซี crystal CS 8900A ซึ่งเชื่อมต่อกับพอร์ท RJ-45 ในการส่งผ่านข้อมูล แสดงได้ดังรูปที่ 3.7



รูปที่ 3.7 แสดงการเชื่อมต่อของไอซี CS8900A กับ RJ-45ในวงจร Ethernet Controller

ในการควบคุมการรับส่งข้อมูลในวงจรนี้จะใช้ไมโครคอนโทรลเลอร์ในการควบคุม Ethernet Controller ด้วยภาษาซี หลังจากที่ข้อมูลส่งออกมาจากพอร์ตซีเรียล (ขา10) ของไมโครคอนโทรลเลอร์ที่ควบคุมโมดูล TRW2.4 GHz ในส่วนนี้ไมโครคอนโทรลเลอร์ของวงจร Ethernet controller จะทำการรับข้อมูลทางพอร์ตซีเรียล (ขา 11) โดยรับข้อมูลเข้ามาทีละไบต์จนครบ

วงจรด้าน Ethernet Controller จะอาศัยการเชื่อมต่อแบบระบบ LAN (Local Area Network) โดยจะอาศัย TCP/IP โปรโตคอลซึ่งสามารถนำมาใช้กับการส่งข้อมูลผ่านระบบ LAN

ในการออกแบบโปรแกรมจึงต้องอาศัย TCP/IP โปรโตคอลมาช่วยซึ่งจะต้องทำการ Encapsulation เพื่อให้ได้แพ็กเกจนั้นสามารถส่งไปยังคอมพิวเตอร์ปลายทางได้ สำหรับโปรโตคอลที่ใช้ในการเลือกส่งข้อมูลจะใช้ UDP (User Datagram Protocol) ซึ่งเป็นการส่งที่มีความรวดเร็ว ถึงแม้จะมีข้อผิดพลาดแต่สำหรับการส่งข้อมูล RFID ที่มีความยาวข้อมูลสั้นจะไม่ค่อยมีผลเท่าไหร่ ยิ่งในทางปฏิบัติแล้วถ้าทำการเขียนโปรแกรมรับส่งให้ดีก็จะไม่มีการผิดพลาดของข้อมูลเลย จึงเลือกใช้โปรโตคอลUDP ซึ่งส่งได้เร็วกว่า TCPโดยจะแบ่งส่วนของโปรแกรมที่เป็นแพ็กเกจได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

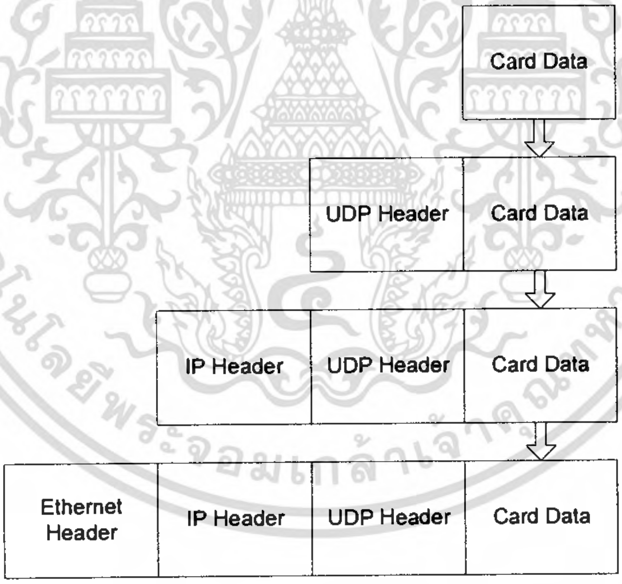
1. ส่วนของข้อมูล
2. ส่วนของ UDP Header
3. ส่วนของ IP Header
4. ส่วนของ Ethernet Header

ซึ่งเมื่อข้อมูลถูกทำ Encapsulation ซึ่งแสดงได้ในรูปที่ 3.8 จะทำการรวมข้อมูลของแต่ละเลเยอร์ ใน TCP/IP ซึ่งประกอบด้วยข้อมูลดังนี้ Ethernet Header (14) + IP Header (20) + UDP Header (8) + Data (22) ซึ่งจะส่งแพ็กแกจไปทั้งสิ้น 64 ไบต์

ส่วนของ UDP Header จะประกอบด้วย Source Port (2 bytes), Destination Port (2 bytes), Message Length (2 bytes), Checksum (2 bytes)

ส่วนของ IP Header จะประกอบด้วย Version Header Length (1 byte), Service (1 byte), Length (2 bytes), Identifier (2 bytes), Flags Fragment offset (2 bytes)(no flag), Time to Live (1 byte), Protocol (1 byte), Checksum IP Header (2 bytes), Source Address (4 bytes), Destination Address (4 bytes)

ส่วนของ Ethernet Header จะประกอบด้วย Send Destination (6 bytes), Send Source (6 bytes), Send Protocol Type (2 bytes)



รูปที่ 3.8 แสดงการ Encapsulation ข้อมูลที่ใช้เขียนในโปรแกรม

นอกจากนี้ในการที่จะส่งข้อมูลออกไปได้จะต้องอาศัย ARP (Address Resolution Protocol)ในการค้นหาหมายเลข IP address เพื่อที่จะสามารถส่งข้อมูลไปยังปลายทางได้ถูกต้อง ซึ่งประกอบด้วย Header 2 ส่วนคือ

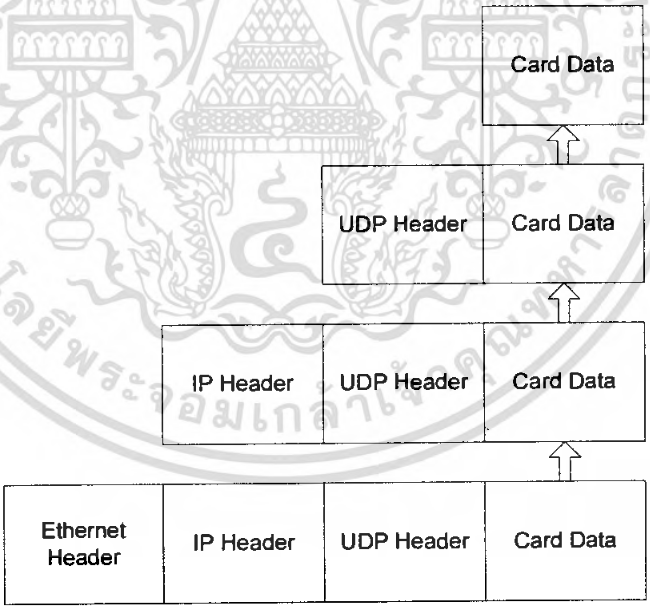
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนของ Ethernet Header

2. ส่วนของ ARP Header

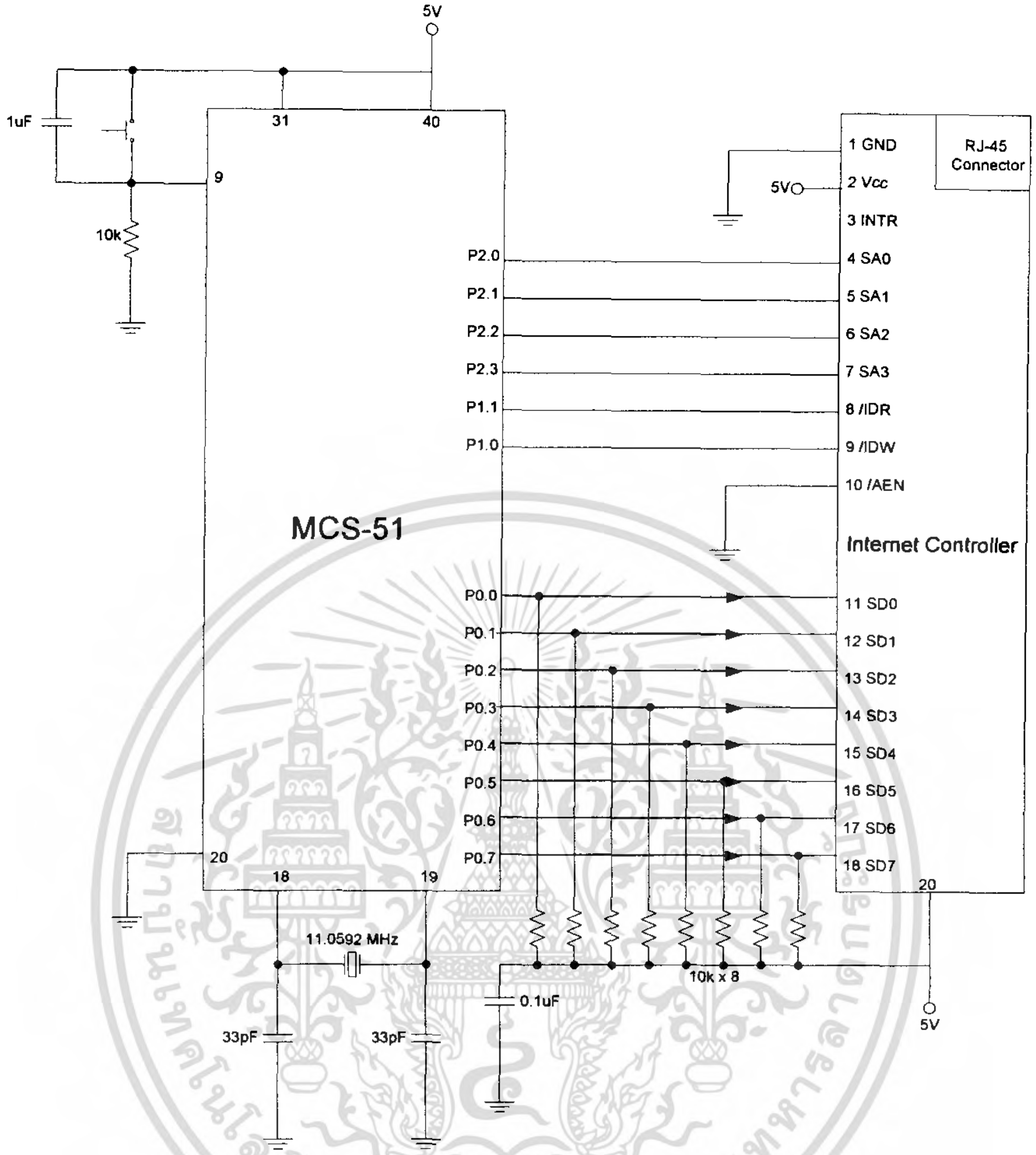
ในส่วนของ ARP Header จะประกอบด้วย Hardware Type (2 byte), Protocol type (2 bytes), Hardware size (1 bytes), Opcode (2 bytes), sender MAC hardware (6bytes), Source Address (4 bytes), Dest MAC computer (6bytes), Destination Address (4 bytes) สำหรับส่วนของ Ethernet Header จะเหมือนเดิม

การทำงานจะเริ่มจากการตั้งค่า IP address และ Mac address ไว้โดยปลายทางจะกำหนด IP address และ Mac address ไว้ที่คอมพิวเตอร์ ส่วน IP address และ Mac address ต้นทางจะกำหนดไว้ที่ Module Ethernet ข้อมูลจะส่งโดยใช้โปรโตคอล UDP(User Datagram Protocol) ด้วยการค้นหาเส้นทางปลายทางซึ่งใช้โปรโตคอล ARP (Address Resolution Protocol) ไปยัง Mac address ปลายทางเพื่อจะได้ IP address ส่งกลับไปยังต้นทาง หลังจากนั้นต้นทางก็จะพร้อมส่งเฟรมข้อมูลลงไปซึ่งประกอบด้วย Ethernet Header(14) + IP Header(20) + UDP Header(8) + Data ที่ได้ Encapsulation เข้าสู่คอมพิวเตอร์ ซึ่งจะทำการ Decapsulation ภายในคอมพิวเตอร์โดยอาศัย TCP/IP เช่นเดียวกัน ดังแสดงได้ในรูปที่ 3.9 ซึ่งสามารถตรวจดูแพ็คเกจข้อมูลด้วยซอฟท์แวร์ เช่น ETHEREAL, WIRESHARK



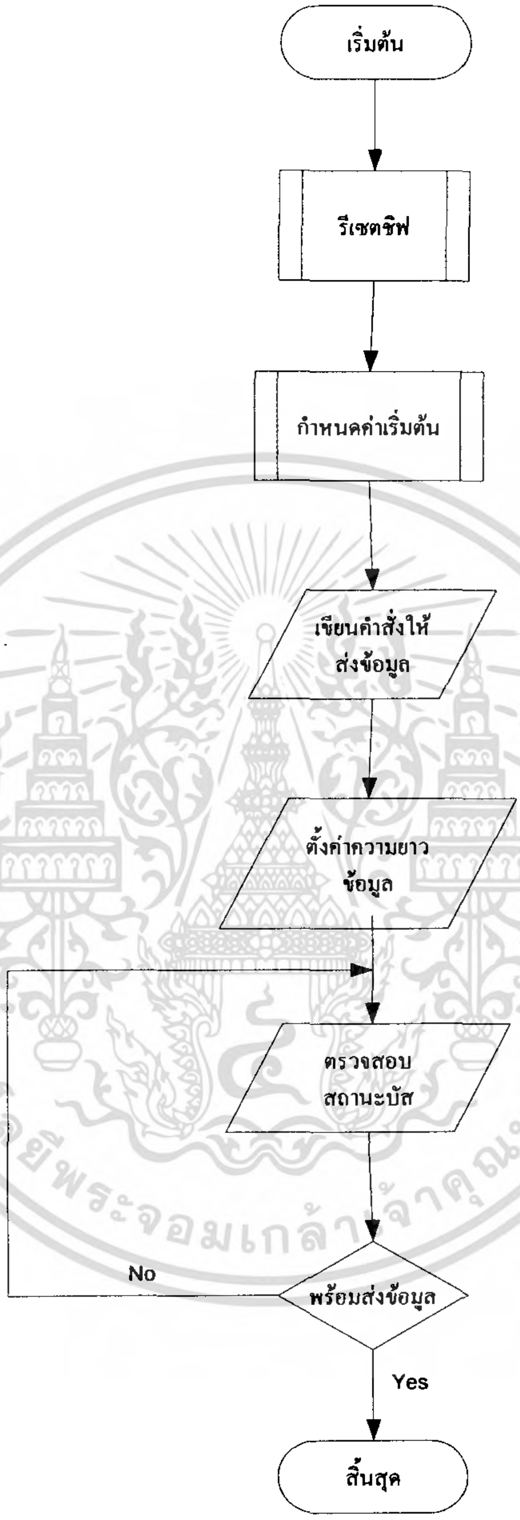
รูปที่ 3.9 แสดงการ Decapsulation ข้อมูลที่ฝั่งคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงวงจร Internet Controller เชื่อมต่อกับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงโฟล์วชาร์ตการทำงานของEthernet Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนการแสดงข้อมูลด้วยโปรแกรม

ภาคของการแสดงข้อมูล ในส่วนนี้จะแสดงผลด้วยโปรแกรมภาษาจาวา ซึ่งโปรแกรมจะทำการรับข้อมูลทีละ 1 ไบต์จากไมโครคอนโทรลเลอร์ ที่ได้ทำการควบคุมวงจร Ethernet Controller โดยเริ่มแรกต้องทำการเชื่อมต่อด้วยวิธี UDP กำหนดค่าหมายเลขพอร์ทให้ตรงกัน โดยส่งข้อมูลผ่านทาง socket

หลักการออกแบบ ในส่วนรับข้อมูล

ในโปรแกรมภาษาจาวาจะสามารถรองรับข้อมูลด้วย UDP โปรโตคอลด้วยกัน 2 คลาสที่พบอยู่ในแพ็กเกจ java.net ซึ่งประกอบด้วย

1. คลาส DatagramPacket ซึ่งใช้ไว้สำหรับเก็บข้อมูลที่เป็นชนิดไบต์เป็นแพ็กเกจ สำหรับส่งและรับจาก UDP

2. คลาส DatagramSocket ซึ่งโปรแกรมทั้งผู้ส่งและผู้รับ จะต้องใช้ไว้สำหรับส่งและรับ

จากนั้นนำคลาสทั้ง 2 มาใช้ประกอบ ในการเขียนโปรแกรม โดยเราจะใช้เขียนโปรแกรมสำหรับรับข้อมูลโดยจะอาศัย คลาส datagram socket สำหรับไว้รับ datagram packet ที่ส่งมาแล้วจึงดึงข้อมูลที่เป็นไบต์ออกมาใช้งานต่อ รายละเอียดของโปรแกรมแสดงอยู่ในภาคผนวก

หลักการออกแบบ ในส่วนเปรียบเทียบกับฐานข้อมูล

ระบบฐานข้อมูลโดยทั่วไปจะมีขั้นตอนการสร้างการติดต่อรวมทั้งการส่งคำสั่ง SOL ออกไปและผลลัพธ์กลับมาเหมือนกัน เพียงมีชื่อ drivers และวิธีอ้างชื่อ database ที่แตกต่างกันไปแต่ละยี่ห้อโดยในที่นี้จะใช้การเชื่อมต่อกับฐานข้อมูลของ Microsoft Access

ขั้นตอนแรกจะต้องสร้างฐานข้อมูลขึ้นมาเพื่อเก็บค่าของหมายเลขบัตร RFID รวมทั้งชื่อของผู้ใช้แสดงได้ตารางที่ 3.1

ตารางที่ 3.1 แสดงฐานข้อมูล

ID	Number Card	First Name	Last Name
1	02001D1DB7	Pung	Jame
2	0900F5E20B	Doctor	Prof
3	0B003546AB	Ben	Klas
4	0A00685A5B	Steven	Yoyo
5	040441525AA	Henry	Hew
6	0B00357D93	JK	Hello
7	095DE54619	Ronaldo	Ww

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการเชื่อมต่อฐานข้อมูลที่สร้างขึ้นไปยัง driver โดยใช้ Microsoft ODBC ซึ่งเป็น APIใช้อย่างแพร่หลาย สามารถติดต่อฐานข้อมูลได้หลายยี่ห้อ โดยมีขั้นตอนการติดตั้งไฟล์เข้ากับ database เข้ากับ ODBC ดังนี้

1. เรียกโปรแกรม ODBC Data Source Administrator ให้ทำงาน(ใน Window xp) โดยเรียกจาก Control Panel> Administrative Tools> DataSources(ODBC)

2. เลือก tab User DSN แล้วเลือกปุ่ม Add จะปรากฏ Create New Data Source dialog

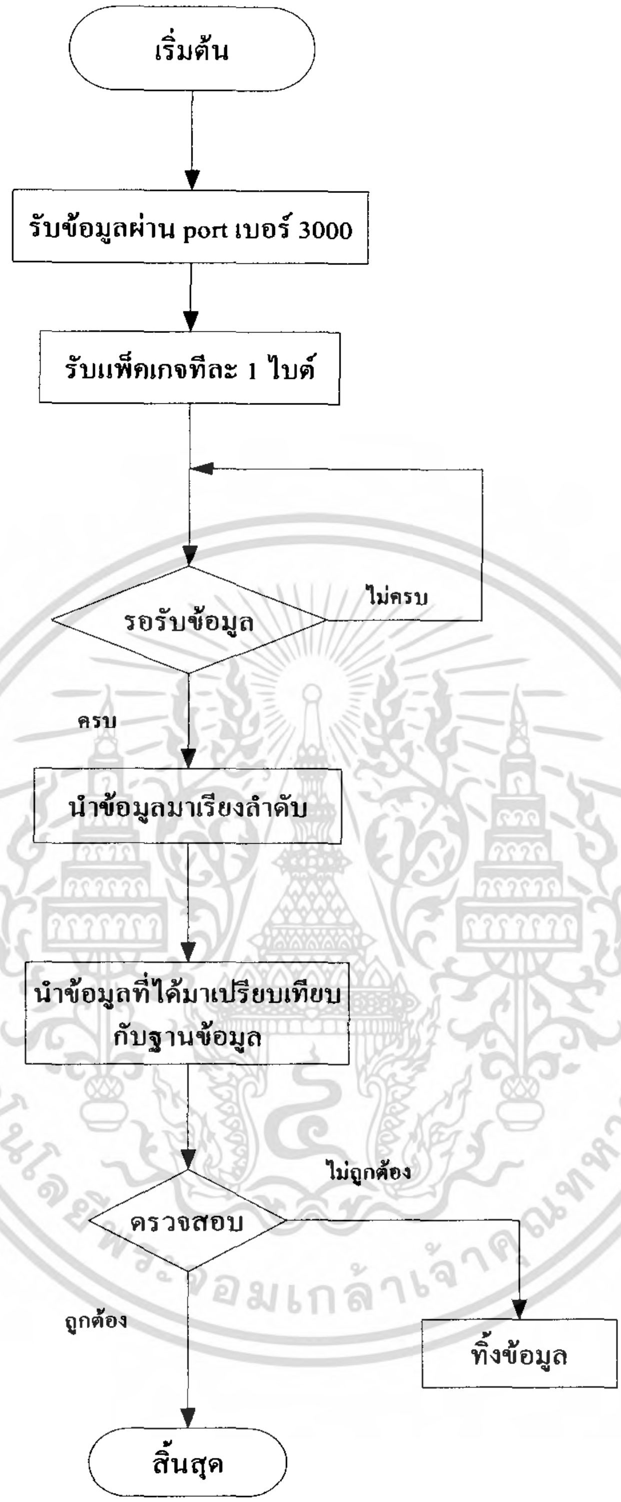
3. เลือกกดปุ่ม Driver do Microsoft Access Driver(*.mdb) แล้วกดปุ่ม Finish

4. ตั้งชื่อ Data Source Name เพื่อใช้อ้างอิงผ่านทาง JDBC-ODBC Bridge ให้เป็นชื่อ table ที่ถูกเชื่อมต่อ จากนั้นเลือกปุ่ม Select จะปรากฏ Select Database dialog ขึ้น

5. จากนั้นเลือก Directions ไปที่เซฟ database แล้วเลือก Database Name เป็นชื่อ DataSources Name คลิกปุ่ม OK เพื่อสิ้นสุดการติดตั้ง

ในส่วนของโปรแกรมจาวาจะต้องใช้แพ็กเกจ java.sql ซึ่งใช้ไว้สำหรับการติดต่อกับdatabaseโดยจะต้องกำหนดที่อยู่ของ driver ให้ถูกต้องโดยกำหนดเป็นชื่อ “sun.jdbc.odbc.JdbcOdbcDriver”รายละเอียดของโปรแกรมแสดงในภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

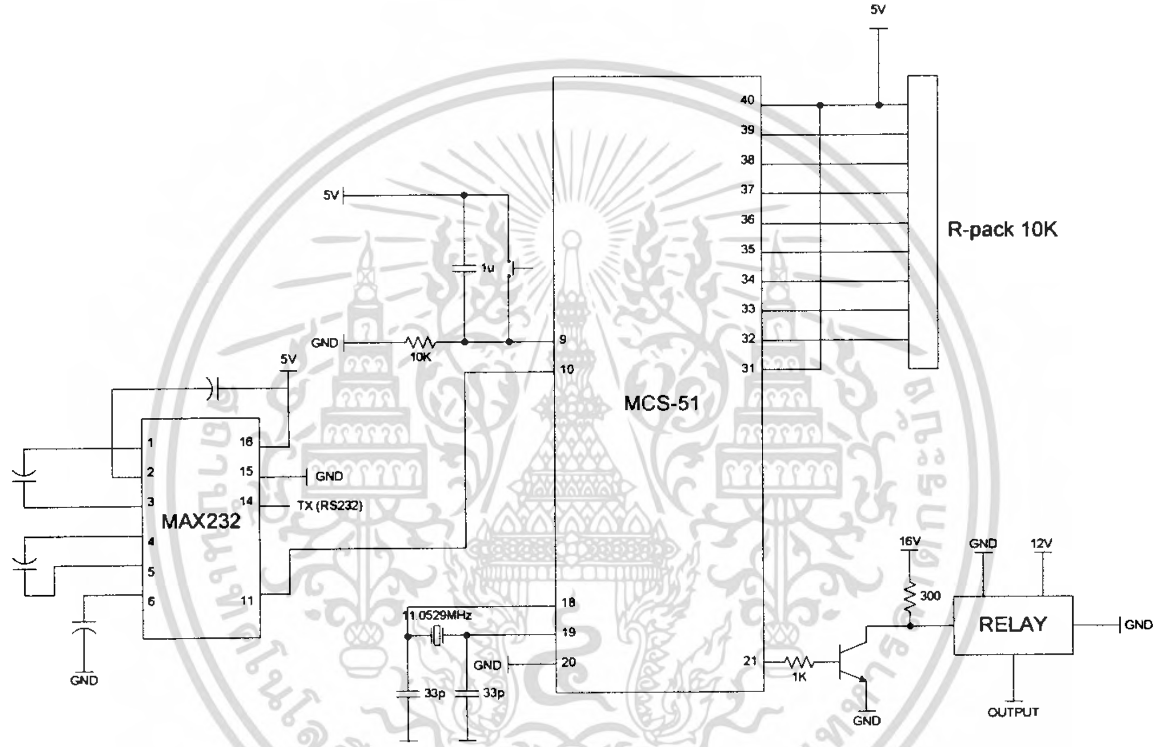


รูปที่ 3.12 แสดงโฟล์วชาร์ตการทำงานของภาคแสดงบนคอมพิวเตอร์ด้วยโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

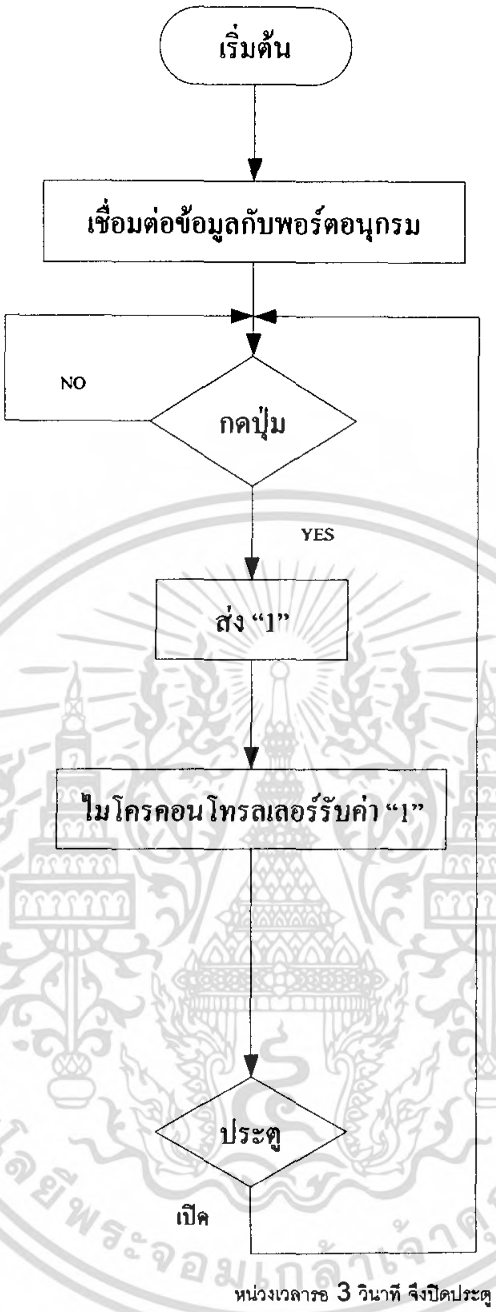
3.5 ส่วนควบคุมการปิด-เปิดประตู

ในส่วนของระบบควบคุมการปิด-เปิดประตูจะประกอบด้วยวงจรที่ควบคุมโดยไมโครคอนโทรลเลอร์ เบอร์ 89C51 โดยทำการรับข้อมูลจากคอมพิวเตอร์ผ่านทาง RS232 โดยเมื่อต้องการที่จะเปิดประตู จะส่งค่า"1"มายังไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม โดยค่า"1"ที่รับมานั้นจะเป็นค่าแอสกี้ (ASCII) ซึ่งแปลงเป็นเลขฐานสองคือ 00110001 จากนั้นจึงนำค่าที่ได้ออกทางพอร์ต 2 ซึ่งบิตสุดท้ายที่มีค่า "1" จะมีค่าเป็น +5V ซึ่งจะทำหน้าที่ไบอัสทรานซิสเตอร์ซึ่งทำหน้าที่เป็นสวิตซ์ใช้ในการเหนี่ยวนำขดลวดที่ รีเลย์ (relay) ในการเลือกว่าจะให้จ่ายไฟให้กับแม่เหล็กหรือไม่ เพื่อทำการเปิดหรือปิดประตูรูปของวงจรควบคุมประตูแสดงได้ดังนี้



รูปที่ 3.13 แสดงวงจรควบคุมการปิด - เปิดประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงโฟล์วชาร์ตการทำงานของส่วนควบคุมประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4

การทดลองและผลการทดลอง

4.1 การทดลองส่งข้อมูลผ่านโมดูล TRW 2.4GHz ด้วยเลขฐาน 2 จำนวน 8 บิต

ขั้นตอนการทดลอง

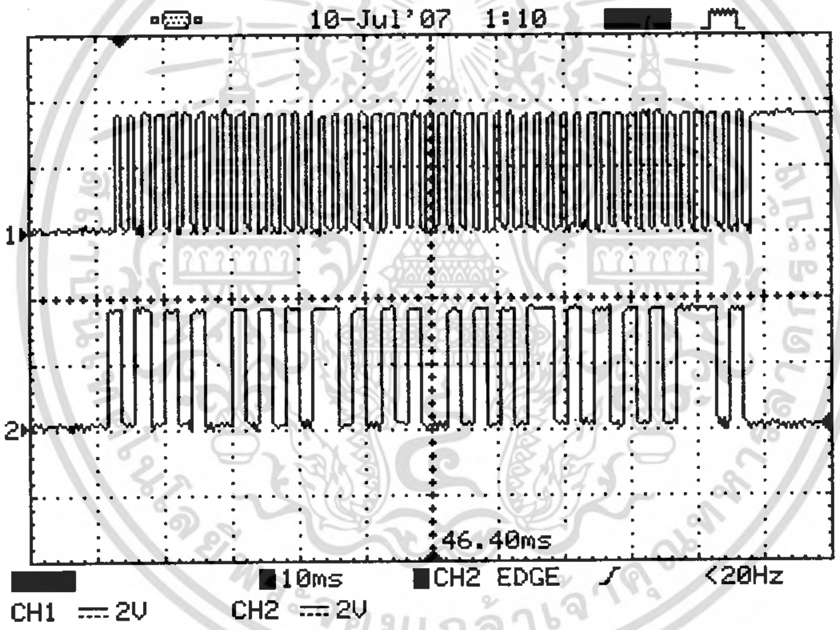
1. เขียนโปรแกรมลงไมโครคอนโทรลเลอร์ โดยใช้โปรแกรมดัง Flow Chart รูปที่ 3.3 เพื่อให้รับค่าผ่านพอร์ทอนุกรม

2. ต่อไฟ LED เพื่อทำการทดสอบที่พอร์ท 0

3. ป้อนไฟเลี้ยง 5 โวลต์ให้กับไมโครคอนโทรลเลอร์และ 3.3 โวลต์สำหรับโมดูล TRW 2.4 GHz

4. ส่งข้อมูลด้วยเลข 10111010

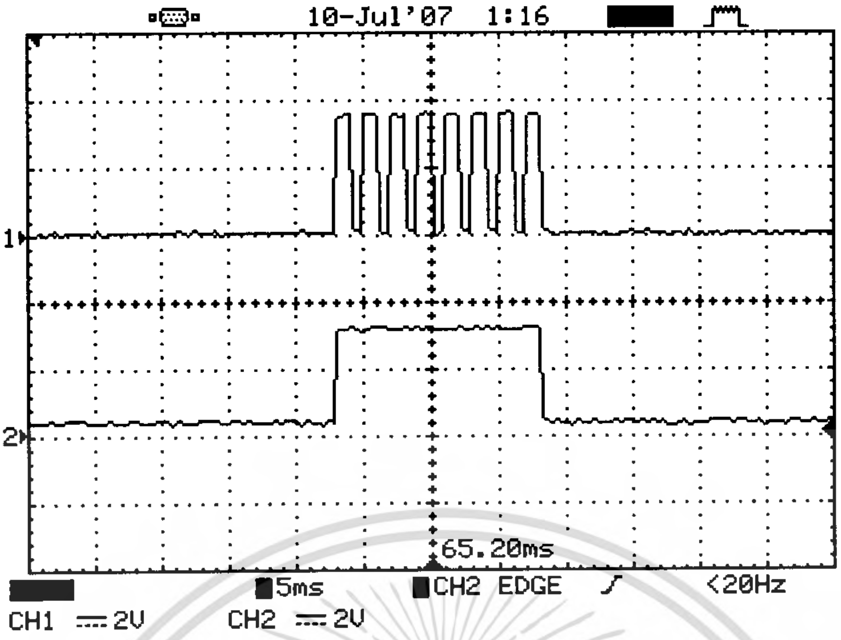
5. วัดสัญญาณนาฬิกาเทียบกับขา DR และขา DATA ของโมดูล TRW 2.4 GHz



รูปที่ 4.1 แสดงการวัดข้อมูลด้านส่ง

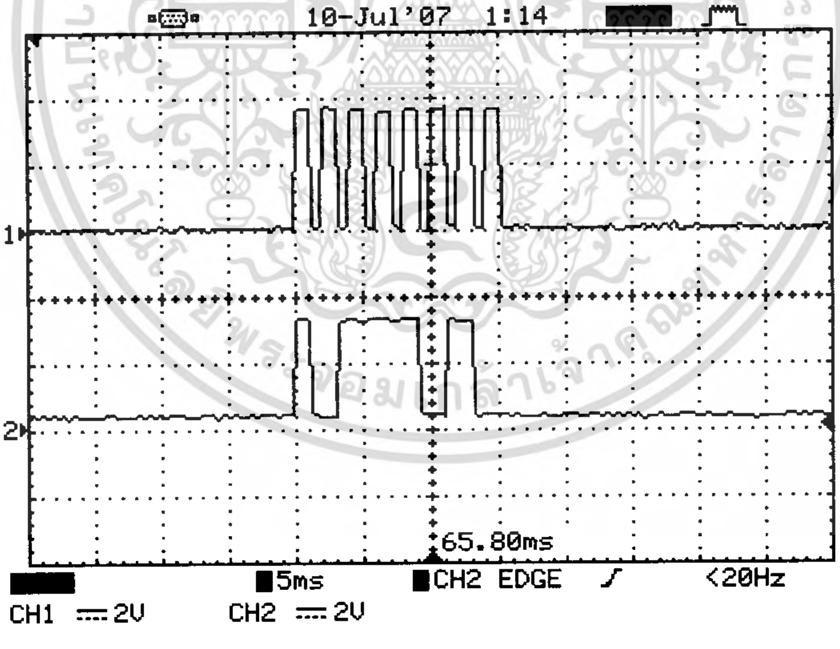
ch 1 แสดงสัญญาณที่ขา clock ch 2 แสดงสัญญาณที่ขา data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงการเปรียบเทียบสัญญาณทางด้านรับระหว่างขา

ch 1 แสดงสัญญาณที่ขาclock ch 2 แสดงสัญญาณที่ขา DR1



รูปที่ 4.3 แสดงการเปรียบเทียบสัญญาณทางด้านรับระหว่างขา

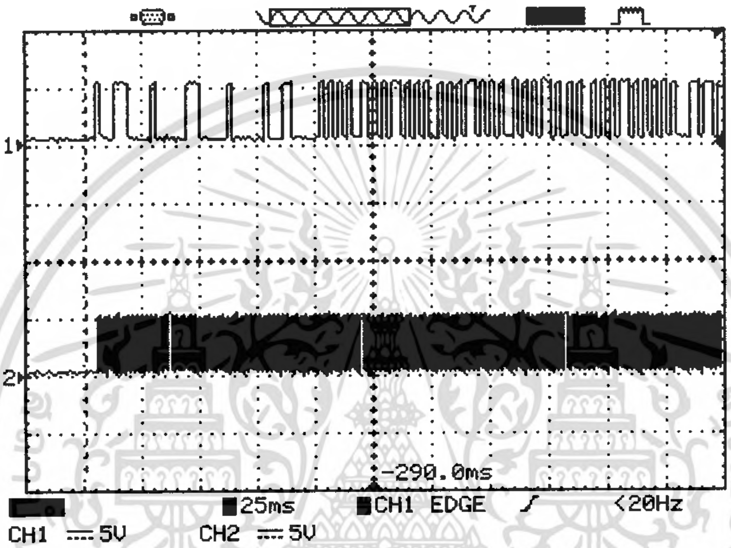
ch 1 แสดงสัญญาณที่ขาclock ch 2 แสดงสัญญาณที่ขา data ซึ่งมีข้อมูลเป็น 10111010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

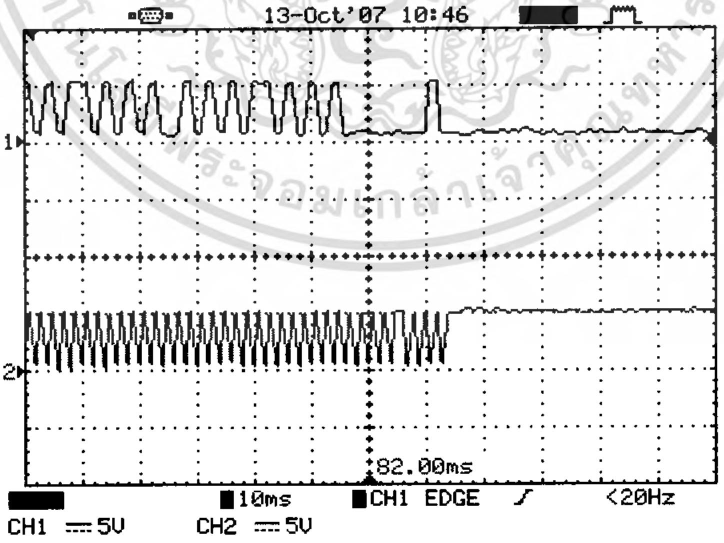
4.2 การทดลองนำข้อมูลจากบัตร RFID ส่งผ่านโมดูล TRW 2.4 GHz

ขั้นตอนการทดลอง

1. เขียนโปรแกรมลงไมโครคอนโทรลเลอร์ สำหรับรับข้อมูลจากบัตร RFID
2. ต่อขาของเครื่องอ่านบัตรเข้ากับไมโครคอนโทรลเลอร์
3. ป้อนไฟเลี้ยง 5 โวลต์ให้กับไมโครคอนโทรลเลอร์และ 3.3 โวลต์สำหรับโมดูล TRW 2.4 GHz
4. วัดสัญญาณที่ขา CLOCK กับขา DATA ตามจำนวนไบต์ข้อมูลที่เข้ามาโดยวัดทีละไบต์



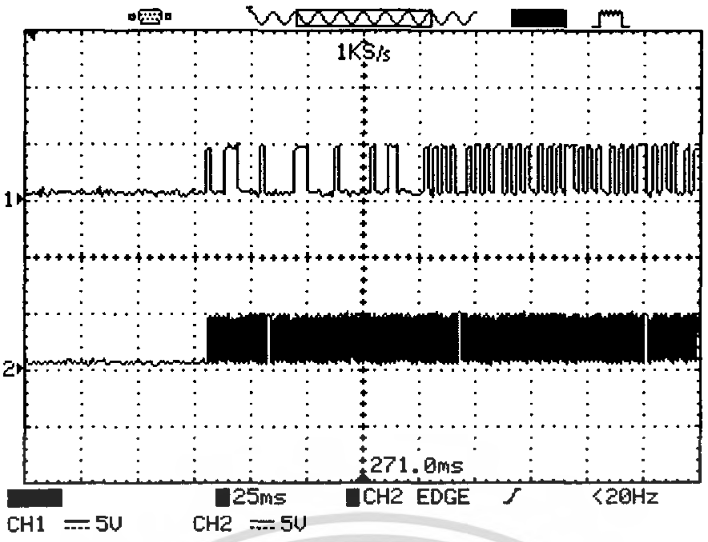
(ก)



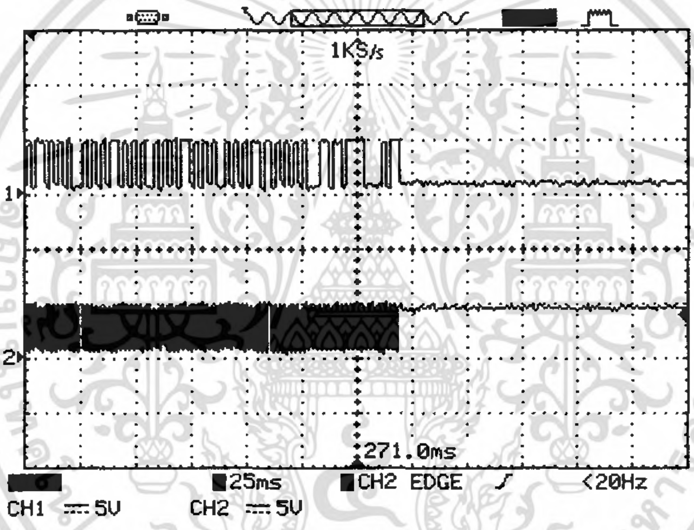
(ข)

รูปที่ 4.4 แสดงสัญญาณด้านส่งที่ขา data (ch1), และที่ขา clock (ch2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

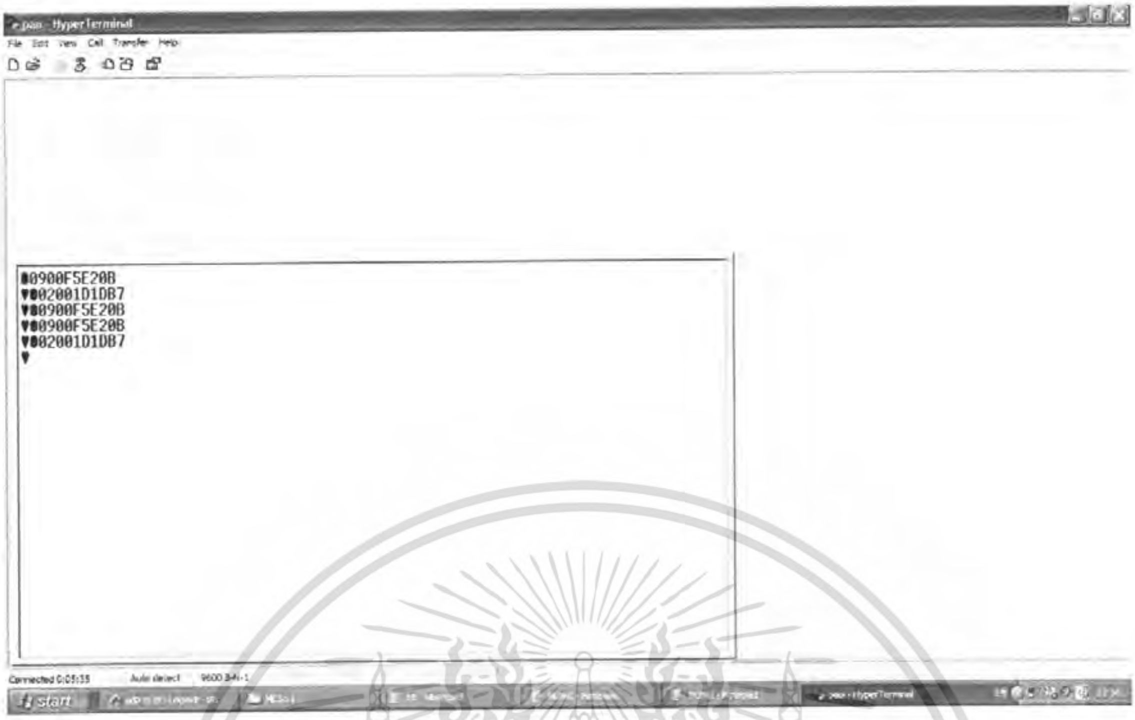
รูปที่ 4.5 แสดงสัญญาณด้านรับที่ขา data (ch1), และที่ขา clock (ch2)

4.3 การทดลองนำบัตร RFID ส่งผ่านโมดูล TRW 2.4 GHz เข้าคอมพิวเตอร์

ขั้นตอนการทดลอง

1. เขียนโปรแกรมแอสเซมบลีเพื่อเชื่อมต่อโมดูล TRW
2. ใช้ไอซี MAX232 แปลงระดับสัญญาณ TTL เข้าสู่คอมพิวเตอร์
3. ป้อนไฟเลี้ยงให้วงจรพร้อมทำการทดสอบส่งค่าจากบัตรRFID
4. ใช้ซอฟท์แวร์ hyper terminal เพื่อแสดงค่าของบัตร RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงข้อมูลจากบัตร RFID ผ่านคอมพิวเตอร์

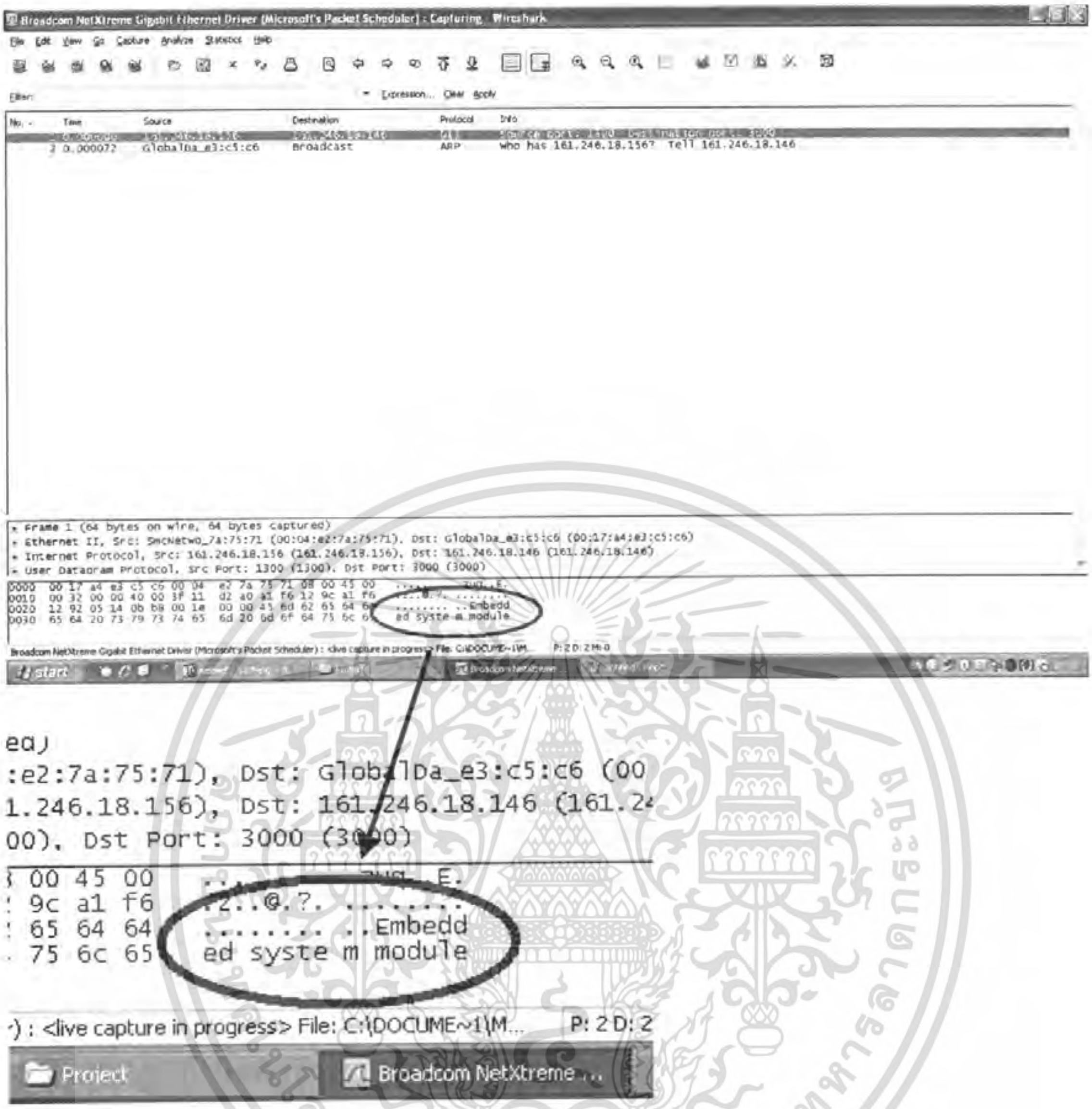
จากรูปที่ 4.6 ค่าตัวเลขที่แสดงออกมานั้น แสดงรหัสเริ่มต้น STX (02HEX) แล้วจึงตามด้วยข้อมูลเลขภายในบัตร RFID 10 ไบต์ โดยอีก 2 ไบต์ถัดมาจะเป็นการเลื่อนตำแหน่งการรับข้อมูลถัดไปเลื่อนกลับไปทางซ้ายมือ และเลื่อนลงมาบรรทัดต่อไป และไบต์สุดท้าย ETX (03HEX) แสดงถึงการสิ้นสุดการส่งข้อมูล

4.4 การทดลองนำข้อมูลส่งผ่านวงจร Ethernet Controller เข้าคอมพิวเตอร์

ขั้นตอนการทดลอง

1. เขียนโปรแกรมภาษาซีลงบนไมโครคอนโทรลเลอร์ เพื่อควบคุมวงจร Embedded
2. ทำการเชื่อมต่อโดยใช้สาย UTP แบบ straight to straight ระหว่างวงจรEmbedded และ พอร์ท RJ-45 ของคอมพิวเตอร์
3. ใช้ซอฟท์แวร์ Wire Shark ในการตรวจจับแพ็คเกจข้อมูล
4. ทำการป้อนข้อมูล “Embedded system module” ลงบนโปรแกรมส่วนแพ็คเกจ
5. วัดผลจาก ซอฟท์แวร์ Wire Shark

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



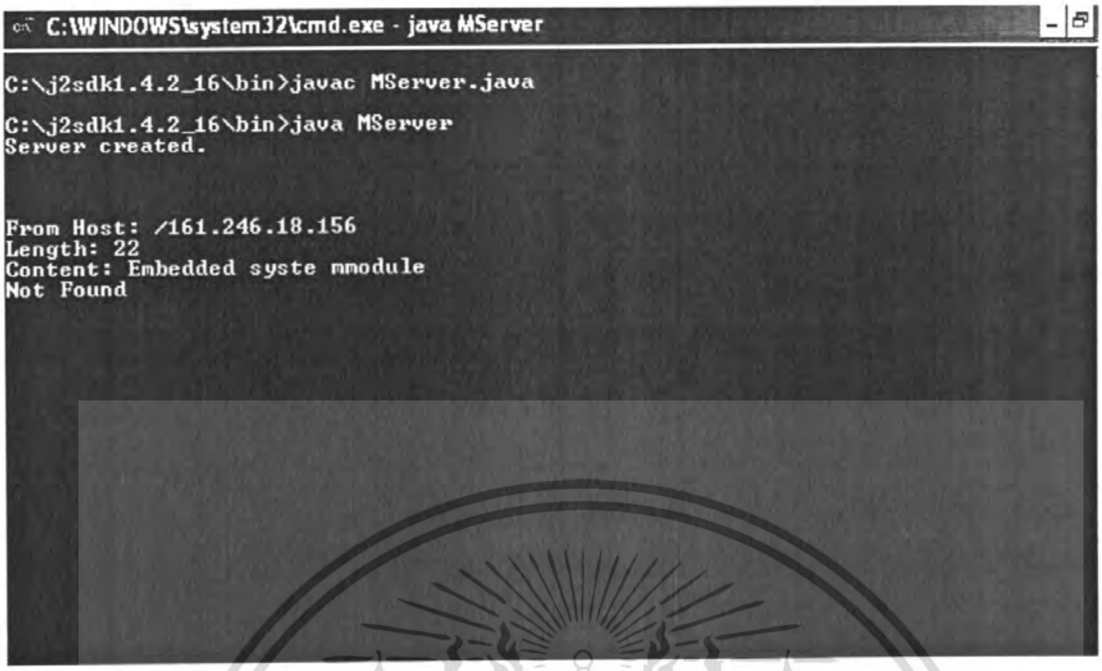
รูปที่ 4.7 แสดงการส่งข้อมูลระหว่างวงจร Embedded Module และคอมพิวเตอร์

4.5 การทดลองนำข้อมูลจากวงจร Ethernet Controller เข้าสู่คอมพิวเตอร์แสดงด้วยโปรแกรม

ขั้นตอนการทดลอง

1. เปิดโปรแกรมภาษาจาวาเพื่อรันรอรับข้อมูล
2. ส่งข้อมูลจากวงจร Ethernet Controller
3. ตรวจดูข้อมูลที่แสดงบนหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



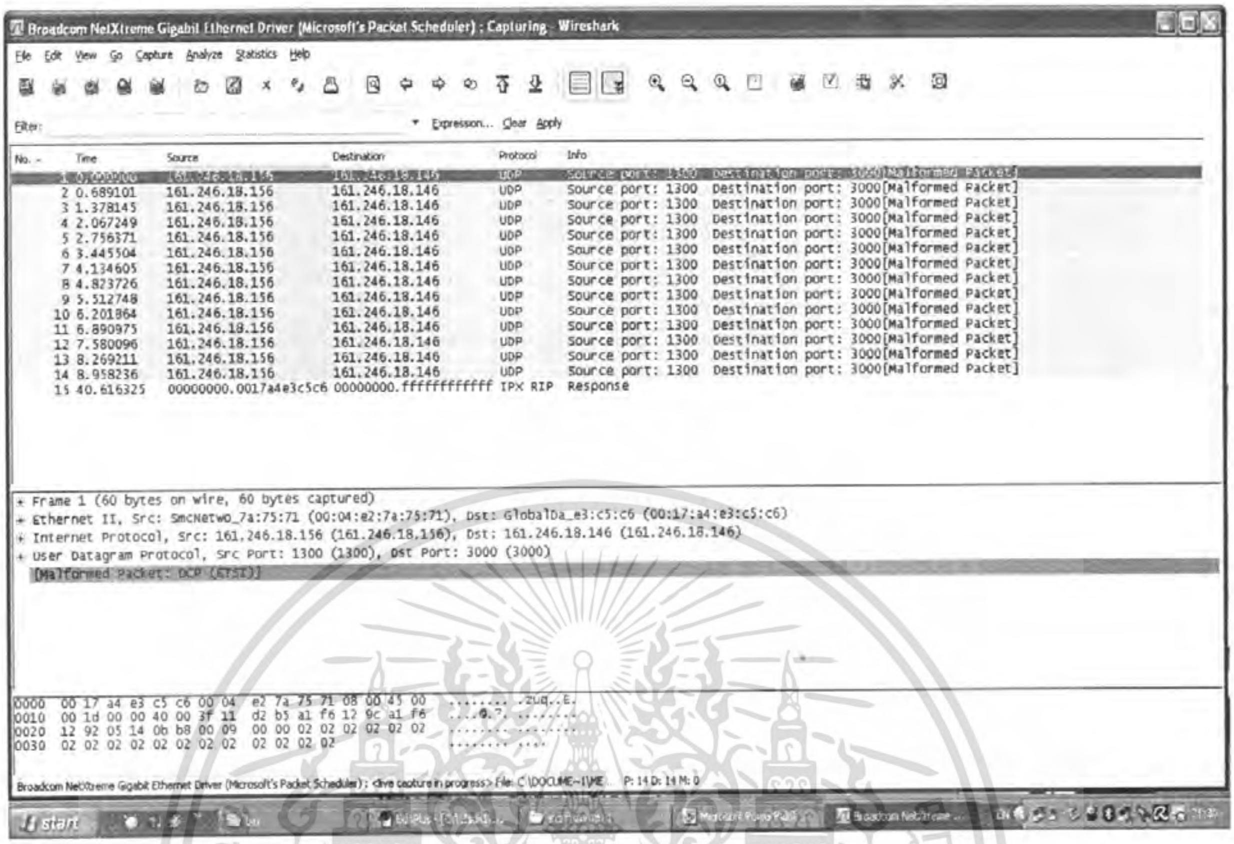
รูปที่ 4.8 แสดงการส่งข้อมูลจากวงจรInternet Controllerเข้าคอมพิวเตอร์ด้วยโปรแกรมจาวา

4.6 การทดลองนำข้อมูลจากบัตร RFID ผ่านวงจร Ethernet Controller เข้าสู่คอมพิวเตอร์แสดงด้วยโปรแกรม

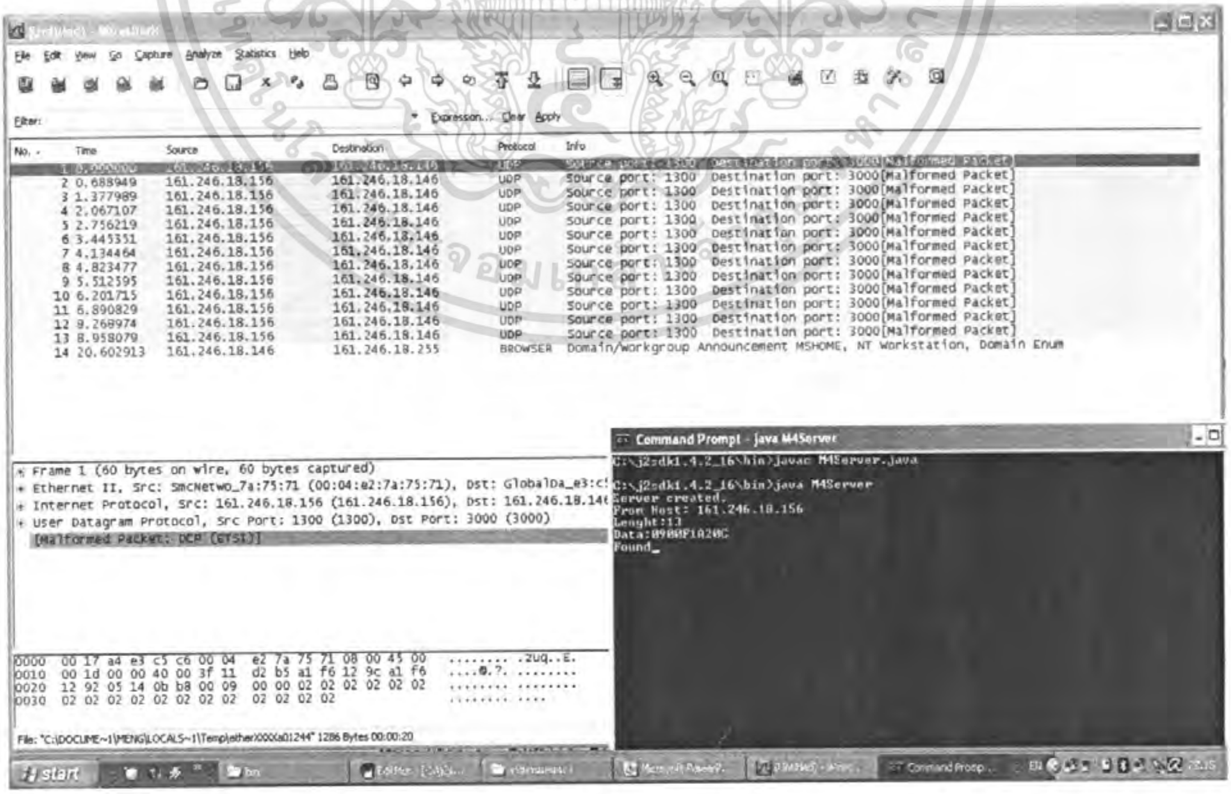
ขั้นตอนการทดลอง

1. ทำการทาบบัตรเพื่อทำการส่งข้อมูล
2. ตรวจสอบการส่งข้อมูลที่ฝั่งรับด้วยไฟ LED ที่วงจร Ethernet Controller
3. run โปรแกรมจาวา เพื่อรับข้อมูลและทำการเปรียบเทียบกับ database
4. ทำการตรวจสอบผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงข้อมูลจากบัตร RFID ตรวจสอบด้วยซอฟท์แวร์ Wireshark

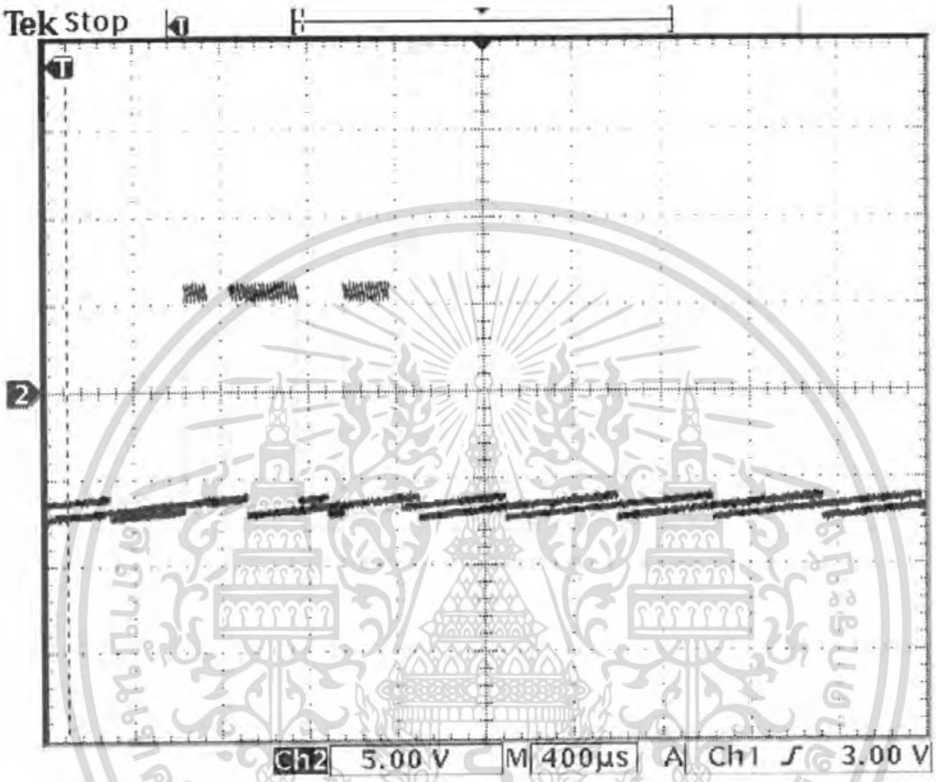


รูปที่ 4.10 แสดงข้อมูลจากบัตร RFID ตรวจสอบด้วยซอฟท์แวร์ Wiresharkและโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การทดลองการเปิด - ปิดประตู

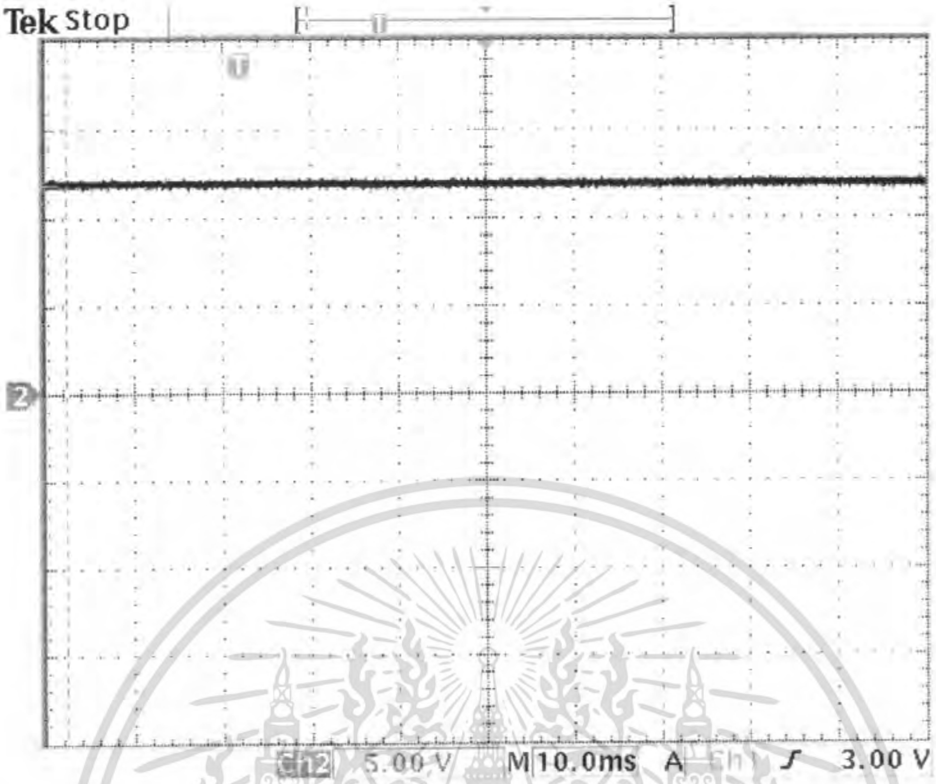
1. ทำการเขียนโปรแกรมเพื่อส่งข้อมูลเป็น "1" ออกจากคอมพิวเตอร์
2. เชื่อมต่อวงจรกับคอมพิวเตอร์ด้วย RS232 และเชื่อมต่อประตูที่เอาท์พุตของวงจร
3. ทำการวัดค่าสัญญาณต่างๆ



รูปที่ 4.11 แสดงสัญญาณจาก RS232 มีค่า 00110001

จากรูปเป็นค่าแอสกี้ "1" ซึ่งแปลงเป็นเลขฐานสองได้ คือ 00110001 เป็นสัญญาณ TTL ค่าที่ได้ดังรูปนั้นจะอ่านค่าจากหลังมาหน้า โดยบิตแรกนั้นเป็นสตาร์ทบิต จากนั้นจึงเป็นค่า 10001100 ตามลำดับ

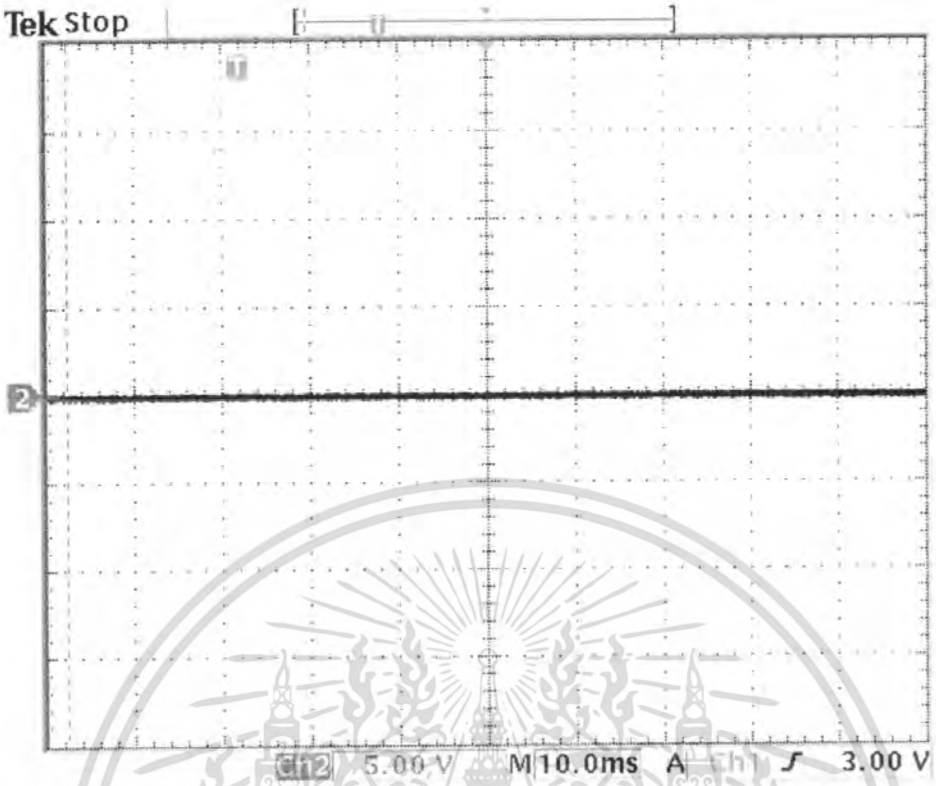
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงสัญญาณเมื่อประตูปิด

จากรูปแสดงสัญญาณกระแสไฟตรง 12V โดยเริ่มต้นรีเลย์ทำการสับสวิตซ์ไปที่สัญญาณกระแสไฟตรง 12V จะทำให้ชุดแม่เหล็กทำงาน ทำให้ไม่สามารถเปิดประตูได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 แสดงสัญญาณเมื่อประตูเปิด

จากรูปแสดงสัญญาณกระแสไฟตรง 0V เมื่อมีกระแสไบอัสมาที่ทรานซิสเตอร์จะทำให้รีเรย์สับสวิตซ์มาที่กระแสไฟตรง 0V จะทำให้ชุดแม่เหล็กไม่ทำงาน ทำให้สามารถเปิดประตูได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุปผล

5.1 บทวิจารณ์

จากการศึกษาเกี่ยวกับการทำงานของเครื่องอ่าน RFID, ไมโครคอนโทรลเลอร์, โมดูล TRW 2.4 GHz, Embedded Module และโปรแกรม ได้พบกับปัญหาต่างๆเหล่านี้

5.1.1 ส่วนของเครื่องอ่าน RFID

เครื่องอ่านบัตร RFID ที่ใช้ต้องทำการตั้งค่าความเร็วให้ตรงตามรุ่นของเครื่องอ่าน โดยต้องทำการกำหนดความเร็วที่โปรแกรมลงบนไมโครคอนโทรลเลอร์ให้ตรงกัน

5.1.2 ส่วนของโมดูล TRW 2.4 GHz

เนื่องจากโมดูล TRW 2.4 GHz มีความนิยมใช้กันมาก ซึ่งโมดูลนี้จะทำงานได้ในช่วงความถี่หนึ่ง ซึ่งหากมีการใช้งานพร้อมกันมากๆ อาจเกิดการซ้อนทับของข้อมูลกัน จึงต้องระวังเวลาที่จะตั้งค่าความถี่ควรตั้งให้มีค่าต่างกัน เพื่อป้องกันการชนกันของข้อมูล

5.1.3 ส่วนของไมโครคอนโทรลเลอร์

ในส่วนนี้จะต้องคำนึงถึงการคำนวณเวลาที่จะใช้ ในการทำงานของโปรแกรมเพื่อให้สามารถรับและส่งข้อมูลกันได้ทัน จึงต้องทำการตั้งค่าช่วงเวลาให้เหมาะสม เริ่มตั้งแต่การรับข้อมูลทีละ 8 บิตจากบัตร RFID จนส่งข้อมูลผ่านโมดูล TRW 2.4 GHz

5.1.4 ส่วนของ Embedded Module

จะมีปัญหาในการใช้โปรแกรมเพื่อเชื่อมต่อระหว่างข้อมูลที่ออกจากโมดูล TRW 2.4 GHz ไปยังวงจร Embedded เนื่องจากเป็นโปรแกรมต่างภาษากัน โดยที่วงจร Embedded ต้องใช้ภาษาซีโปรแกรมลงไมโครคอนโทรลเลอร์และอีกหนึ่งปัญหาที่พบคือข้อมูลที่ทดลองใช้ส่ง เมื่อใช้ซอฟท์แวร์ตรวจจับแพ็คเกจโปรโตคอลที่ได้อาจจะได้ไม่ตรงตามที่ตั้งไว้ถึงแม้จะได้ข้อมูลที่ถูกต้องก็ตาม เป็นผลมาจากการใช้พอร์ทระหว่างต้นทางหรือปลายทางและข้อมูลที่ใช้ไปซ้อนทับกับโปรโตคอลอื่น

5.1.5 ส่วนของโปรแกรมคอมพิวเตอร์

การเขียนโปรแกรมจากไมโครคอมพิวเตอร์ออกเข้าสู่วงจร Ethernet Controller ทำได้ยาก การที่จะหา source ที่เกี่ยวข้องยังมีอยู่น้อย

5.2 บทสรุป

จากการศึกษาและการทดลองการส่งผ่านข้อมูลผ่านระบบไร้สายรวมทั้งระบบแลนนั้นสามารถส่งข้อมูลได้จริง สามารถรับค่าข้อมูลจากบัตร RFID ได้ จากด้านส่งไปยังด้านรับซึ่งส่งข้อมูลได้รวดเร็วพอด้านรับสามารถรับข้อมูลส่งผ่านไปแสดงผลยังคอมพิวเตอร์ สำหรับส่วนที่ต้องส่งข้อมูลออกระบบ LAN นั้นได้ทำการทดสอบแล้วและในการส่งข้อมูลทั้งหมดโดยนำส่วนต่างๆมาเชื่อมต่อกันจากบัตร RFID ออกสู่ LAN ได้เป็นที่ต้องการ โดยเมื่อแก้โปรแกรมภาษาซี เข้าสู่คอมพิวเตอร์ทำให้การส่งข้อมูลได้ถูกต้องแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1 อานันท์ สีห์พิทักษ์เกียรติ, TCP/IP โครงสร้างและหลักการทำงานเบื้องต้น สาขาวิชาคอมพิวเตอร์ วิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่, ธันวาคม 2540.

2 สมยศ จุฌณะปิยะ, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูลMCS-51 ,2550

3 ศูนย์พัฒนาธุรกิจออกแบบวงจรรวม, รู้จักกับเทคโนโลยีอาร์เอไอดี ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, 2548

4 ดร.วีระศักดิ์ ซึงถาวร, JAVA PROGRAMMING VolumeIII บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2547

5 ดร.วีระศักดิ์ ซึงถาวร, JAVA PROGRAMMING VolumeI บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2547

6 อุดม รานอก, ภาษาสำหรับงานควบคุมไมครคอนโทรลเลอร์ MCS-51 บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์, 2548

7 http://www.nordicsemi.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมด้านส่งโมดูล TRW-2.4GHz

```
CH     EQU   030H
;*******************************************************
;  Set  port  TRW
;*******************************************************
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

;*******************************************************
;  Set   initial  UART
;*******************************************************
ORG 0000H

       MOV   P0,#00000000B

       MOV   PCON,#00H ; bit rate 1 time
       MOV   SCON,#50H ; control mode uart
       MOV   TMOD,#20H ; send 8 bit data
       MOV   TH1,#0FDH ; set baud rate

       SETB  TR1

       MOV   P0,#11110000B
       LCALL DELAY_DATA

       MOV   R0,#040H
       MOV   R4,#0EH

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

       MOV   R0,#040H
       MOV   R4,#0EH

OUT:   MOV   ACC,@R0
       MOV   P0,ACC
       LCALL DELAY_DATA
       INC   R0
       DJNZ  R4,PRE


       MOV   CH,#040H
       MOV   R4,#0EH

MAIN:
       LCALL INIT
       LCALL SETMODE_TX
       MOV   R0,CH
       MOV   ACC,@R0
       MOV   P0,ACC
       MOV   ACC,P0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL SEND_TRW
        INC   CH
        DJNZ  R4,MAIN


        SJMP  $

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

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

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


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

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            DJNZ  R1,SETMODE_0_TX
            SETB  DAT
            SETB  DR1
            SETB  CE
            CLR   CS
            RET

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

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;**********************************************************************
**********

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

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

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

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

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

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

CLK_TRW:    CLR   CLK
        CALL  DELAY_1ms
        SETB  CLK
        CALL  DELAY_1ms
        RET

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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

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

CONFIG_TEST_TX:   DB 8EH,08H,1CH
CONFIG_LEN2_TX:   DB 08H
CONFIG_LEN1_TX:   DB 08H
CONFIG_ADDR2_TX:  DB  0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_TX:  DB  0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_TX:      DB 0A3H
CONFIG_RF_MODETX:             DB 6FH
CONFIG_RF_CH_TX:        DB C8H ;TX


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

CONFIG_TEST_RX:   DB 8EH,08H,1CH
CONFIG_LEN2_RX:   DB 08H
CONFIG_LEN1_RX:   DB 08H
CONFIG_ADDR2_RX:  DB  0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_RX:  DB  0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_RX:      DB 0A3H
CONFIG_RF_MODERX:             DB 6FH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CONFIG_RF_CH_RX:        DB C9H ;RX


END
```

โปรแกรมด้านรับโมดูล TRW-2.4GHz

```
;*********************************************************************
**********
;            Program TRW2.4GHZ Rx
;*********************************************************************
**********

;*********************************************************************
**********
;  Set  port  TRW
;*********************************************************************
**********

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

ORG    0000H

       MOV    P0,#00000000B      ;CLR P0

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

       SETB   TR1

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

MAIN_TX:     MOV    R4,#0EH


REC_1:              LCALL RECIEVE
             MOV    P0,A
             LCALL TX

             DJNZ   R4,REC_1


             SJMP   $
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************************************
**********

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

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

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

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

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

SETMODE_TX:
               CLR   CE
               SETB  CS
               CLR   A
               MOV   R1,#18                    ;ALL18*8 = 144bit
SETMODE_0_TX:        MOV   DPTR,#CONFIG_TEST_TX      ;Set Header
               PUSH  ACC
               MOVC  A,@A+DPTR
               CALL  WRITE_TRW24
               POP   ACC
               INC   A
               DJNZ  R1,SETMODE_0_TX
               SETB  DAT
               SETB  DR1
               SETB  CE
               CLR   CS
               RET

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

SETMODE_RX:
               CLR   CE
               SETB  CS
               CLR   A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

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

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

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

READ_TRW24:         CLR     A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

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

CLK_TRW:    CLR    CLK
            CALL   DELAY_1ms
            SETB   CLK
            CALL   DELAY_1ms
            RET

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            DJNZ  R6,DELAY_100ms_2
            DJNZ  R7,DELAY_100ms_1
            RET

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

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

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

CONFIG_TEST_TX:    DB 8EH,08H,1CH
CONFIG_LEN2_TX:    DB 08H
CONFIG_LEN1_TX:    DB 08H
CONFIG_ADDR2_TX:   DB  0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_TX:   DB  0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_TX:       DB 0A3H
CONFIG_RF_TX:            DB 6FH
CONFIG_RF_CH_TX:         DB C8H ;TX


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

CONFIG_TEST_RX:    DB 8EH,08H,1CH
CONFIG_LEN2_RX:    DB 08H
CONFIG_LEN1_RX:    DB 08H
CONFIG_ADDR2_RX:   DB  0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_RX:   DB  0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_RX:       DB 0A3H
CONFIG_RF_RX:            DB 6FH
CONFIG_RF_CH_RX:         DB C9H ;RX

END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาษาจาวาสำหรับแสดงข้อมูลในคอมพิวเตอร์และเปรียบเทียบดาต้าเบส

```
import java.sql.*;
import java.net.*;
class M4Server
{
	public static void main(String args[]) throws Exception
		{
			DatagramSocket ds = new DatagramSocket(3000);
			System.out.println("Server created.");
			String msgg = "";
			int i = 1;
			int x = 0;
			while (true)
			{
				DatagramPacket dp = new DatagramPacket(new
				byte[100], 100);
				ds.receive(dp);
				String header = "\nFrom Host: " +
		dp.getAddress() + "\nLength: " + dp.getLength();
				String msg = new String(dp.getData()).trim();
				if (msg.equals ("quit"))
						break;
				x = x + dp.getLength();
				msgg = msgg + msg;

					i = i+1;
				if (i>10)
				{
					System.out.println("From Host:161.246.18.156");
					System.out.println("Lenght:" + x);
					System.out.println("Data:" + msgg);
					compareDB(msgg);
				}
			}
			ds.close();
		}
	public static void compareDB(String tt)throws Exception
	{
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		Connection c =
DriverManager.getConnection("jdbc:odbc:Meng" , "","");
		Statement s = c.createStatement();
		ResultSet r = s.executeQuery("SELECT * FROM Meng");

		boolean found = false;
		while(r.next()&& !found)
		{	if(tt.equals(r.getString(2)))
			{
				System.out.print("Found");
				found = true;
			}
		}
		if(!found)

			System.out.print("Not Found");
		s.close();
		c.close();
	}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้