

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบตรวจสอบสินค้าด้วย RFID

Stock Checking System by RFID Technology



เลขหมู่.....  
เลขทะเบียน..... 62535  
วัน,เดือน,ปี..... 19 ส.ค. 2549

b. 11629922  
i. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมโทรคมนาคม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องตรวจสอบข้อมูลสินค้าด้วย RFID  
Stock Checking Devices by RFID Technology

โดย

นาย เป็ลย์ เจียบหญู 45010481  
นางสาว พงนา พรหมแสง 45010491  
นางสาว พรทิพย์ อร่ามรัตนา 45010497

อาจารย์ที่ปรึกษา

ผศ. นภัทร สระเอี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมโทรคมนาคม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2548

ผ่านการตรวจชิ้นงานแล้ว  
(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของสิทธิ์  
ผ่านการตรวจรูปเล่มแล้ว  
(ลงชื่อ).....ผู้ตรวจ

ปริญญาโทปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจสอบสินค้าด้วย RFID

Stock Checking System by RFID Technology

ผู้จัดทำ

1. นายเป็ชัย เจียบหญ่ 45010481
2. นางสาว พงนา พรหมแสง 45010491
3. นางสาว พรทิพย์ อร่ามรัตนา 45010497

.....  อาจารย์ที่ปรึกษา

( ผศ. นภัทร สระเอี่ยม )



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบตรวจสอบสินค้าด้วย RFID

### Stock Checking System by RFID Technology

โดย นาย เป็ชัย                      เจี๊ยบหนู      45010481

นางสาว พงนา                      พรมแสง      45010491

นางสาว พรทิพย์      อร่ามรัตนา      45010497

อาจารย์ที่ปรึกษา ผศ. นภัทร สระเอี่ยม

#### บทคัดย่อ

โครงการนี้เป็นโครงการที่ใช้ระบบ RFID ในการตรวจสอบสินค้า โดยระบบ RFID นี้ จะใช้ ตัว Tag และ ตัว Reader ในการเก็บและตรวจสอบสินค้า ซึ่งโครงการนี้จะใช้ Tag ในการเก็บข้อมูลของ สินค้า และ ใช้ Reader เพื่อตรวจสอบสินค้า และสามารถควบคุม Reader ผ่านอุปกรณ์เชื่อมต่อกับโครงข่าย

#### Abstract

This project presents RFID (Radio Frequency Identification) technology which applies for stock checking. The system includes tag and reader to store and checks stock's data. In this project, tag will be assigned to keep stock's data and reader will be assigned to check stock's data and Reader can be controlled by Network Interface Card.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
<b>บทที่ 1 บทนำ</b>	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีดำเนินงาน	2
<b>บทที่ 2 ทฤษฎีและหลักการ</b>	3
2.1 หลักการทำงานของ RFID	3
2.1.1 ฉลากอิเล็กทรอนิกส์ ( Transponder )	3
2.1.1.1 ชนิดของแท็กส์	3
2.1.1.2 เทคโนโลยีของหน่วยความจำ	4
2.1.2 เครื่องอ่านสัญญาณ ( Reader )	5
2.1.3 ความถี่ของคลื่นพาหะ	5
2.1.4 RFID ทำงานอย่างไร	6
2.1.5 ประเภทของ ระบบ RFID	7
2.1.5.1 RFID ที่จำแนกโดยขนาดของหน่วยความจำ	7
2.1.5.2 RFID ที่จำแนกโดยลักษณะการเคลื่อนของสัญญาณ	9
2.1.5.3 RFID ที่จำแนกตามความสามารถของระบบ	11
2.1.6 กระบวนการรับส่งข้อมูลของ RFID	13
2.1.6.1 กระบวนการ Half Duplex (HDX)	13
2.1.6.2 กระบวนการ Full Duplex(FDX)	13
2.1.7 พลังงานที่จ่ายให้แท็กส์	14
2.1.8 เทคโนโลยีการเข้ารหัสและถอดรหัส	16
2.1.8.1 การเข้ารหัสข้อมูล (Signal Coding)	16
2.1.8.2 การผสมรหัสข้อมูลเข้ากับคลื่นพาหะ (Modulation)	18
2.1.8.3 การส่งคลื่นสัญญาณออกไป (Transmission)	19
2.1.8.4 การถอดข้อมูลออกจากคลื่นพาหะ (Demodulation)	19
2.1.9 การป้องกันการเกิดการชนกันของข้อมูล (ANTICOLLISION)	19
2.1.9.1 วิธีป้องกันข้อมูลชนกันแบบใช้การอ้างอิง ความถี่(frequency domain anti-collision)	21

	หน้า
2.1.9.2.2 วิธีป้องกันข้อมูลชนกันแบบใช้การอ้างอิงเวลา(time domain anti-collision)	21
2.2 ฐานข้อมูล MySQL	22
2.3 Delphi 7	24
2.3.1 ข้อเด่นของ Delphi 7	24
2.3.2 องค์ประกอบของ Delphi 7	24
2.3.3 รายละเอียดของคอม โพนেন্টที่สำคัญ	25
2.4 Ethernet IO Bord	29
<b>บทที่ 3 การคำนวณและการออกแบบ</b>	<b>33</b>
3.1 การออกแบบส่วนของ RFID	33
3.1.1 EM4095 : อ่าน/เขียน อนุภาคสำหรับ 125 kHz RFID เบสสเดชั่น	33
3.1.2 Tag : Multifunctional 330 bit Read/Write RF-Identification IC	36
3.1.3 โปรโตคอลการติดต่อสื่อสาร	39
3.2 ระบบฐานข้อมูล	41
3.2.1 จัดการฐานข้อมูล Mysql ด้วย phpmyadmin	41
3.2.2 การออกแบบระบบฐานข้อมูล	43
3.3 การออกแบบโปรแกรม Delphi	45
3.3.1 ขั้นตอนการติดต่อกับซีเรียลพอร์ต	45
3.3.2 หลักการทำงานของระบบ	47
3.4 การควบคุมการปิดเปิดเครื่องอ่านสัญญาณ	48
<b>บทที่ 4 การทดลองและผลการทดลอง</b>	<b>54</b>
4.1 การวัดสัญญาณของเครื่องอ่านสัญญาณ	54
4.1.1 สัญญาณของวงจร ไอซี EM4095 ขณะไม่มีแท็กส์	54
4.1.1 สัญญาณของวงจร ไอซี EM4095 ขณะมีแท็กส์	56
4.1.1 สัญญาณที่ผ่านวงจรของ ไอซี EM4095 แล้ว	57
4.1.4 การวัดสัญญาณส่วนหัวของข้อมูล	58
4.2 ส่วนของโปรแกรมที่ใช้ในการติดต่อกับเครื่องอ่านสัญญาณ	59
4.3 แสดงรายการสินค้าผ่านเว็บไซต์	62
4.4 ผลการทดลองส่วนควบคุมการปิดเปิดผ่านเครือข่าย	62
<b>บทที่ 5 สรุปและวิเคราะห์ผล</b>	<b>65</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

	หน้า
บทที่ 1 บทนำ	
บทที่ 2 ทฤษฎีและหลักการ	
รูปที่ 2.1 รูปสัญญาณ sequence terminator ของรหัส Manchester	5
รูปที่ 2.2 การสื่อสารระหว่างเครื่องอ่านกับ แทกส์	7
รูปที่ 2.3 วงจรเทียบเคียงของระบบ RFID	7
รูปที่ 2.4 แทกส์ ของระบบ EAS จะมีเฉพาะเสาอากาศอย่างเดียว	8
รูปที่ 2.5 เสาอากาศของระบบ EAS	8
รูปที่ 2.6 วงจรเทียบเคียงของ Close – Coupling	9
รูปที่ 2.7 วงจรเทียบเคียงของ Remote Coupling หรือ Inductive Coupling	10
รูปที่ 2.8 วงจรเทียบเคียง Long Range	10
รูปที่ 2.9 ระบบ RFID ที่แบ่งตามฟังก์ชันของระบบ	11
รูปที่ 2.10 บล็อกไดอะแกรมของแทกส์แบบอ่านอย่างเดียว เมื่อแทกส์เข้าสู่บริเวณการตรวจสอบของ reader การนับจะเริ่มนับทุกแอดเดรสที่อยู่ในหน่วยความจำอย่างต่อเนื่อง เอาท์พุทของข้อมูลของหน่วยความจำจะเชื่อมต่อกับ โหมดมอดูเลเตอร์ที่เซตเป็นรหัสเบสแบนด์ของรหัสไบนารี ในกระบวนการนี้ข้อมูลทั้งหมดในหน่วยความจำ (128 บิตซีเรียลนัมเบอร์) สามารถส่งไปได้อย่างต่อเนื่อง	11
รูปที่ 2.11 บล็อกไดอะแกรมของแทกส์ที่เขียนได้ที่มีฟังก์ชัน cryptological ระหว่างแทกส์กับ reader	13
รูปที่ 2.12 แสดง Full Duplex , Half Duplex และ Sequential System บนเวลา ข้อมูลจะส่งจากตัวReader ไปยัง Transponder เรียกว่า downlink และจากTransponder ไปยัง Reader เรียกว่า uplink	14
รูปที่ 2.13 พลังงานการเหนี่ยวนำที่เกิดจากการคล่องสัญญาณของ Tag จากพลังงานของสนามแม่เหล็กที่เปลี่ยนแปลงและสร้างโดยตัว Reader	15
รูปที่ 2.14 การออกแบบแทกส์	15
รูปที่ 2.15 การไหลของสัญญาณและข้อมูลในระบบการสื่อสารแบบดิจิทัล	16
รูปที่ 2.16 รหัสสัญญาณ NRZ	17
รูปที่ 2.17 รหัสสัญญาณ Biphase	17
รูปที่ 2.18 ใน ASK ความสูงของขดคลื่นจะเปลี่ยนสถานะอยู่ 2 สถานะ ขึ้นอยู่กับข้อมูลที่ผสมเข้ามา	18
รูปที่ 2.19 2FSK จะเปลี่ยนแปลงความถี่ของคลื่นตามข้อมูลที่ผสมเข้ามา	18
รูปที่ 2.20 PSK จะกลับเฟสทุกครั้งเมื่อมีการเปลี่ยนสถานะของข้อมูล	19

รูปที่ 2.21 แพทส์หลายๆตัวในบริเวณทำงานของเครื่องอ่านสัญญาณตัว เดียวเป็นสาเหตุของการชนกันของข้อมูล	20
รูปที่ 2.22 รูปแบบในการจัดเก็บฐานข้อมูล	22
รูปที่ 2.23 แสดงการเชื่อมต่อระหว่างภาษา PHP กับ Mysql	24
รูปที่ 2.24 คอมโพเนนต์ในเพจ MySQL Access	25
รูปที่ 2.25 คอมโพเนนต์ในเพจ Varian Freeware	28
รูปที่ 2.26 Ethernet IO บอร์ด	29
รูปที่ 2.27 แสดงตัวอย่างการเชื่อมต่อ Ethernet IO	30
รูปที่ 2.28 แสดงโครงสร้างของซอฟต์แวร์	31

### บทที่ 3 การคำนวณและการออกแบบ

รูปที่ 3.1 แสดงแผนผังการทำงานระบบ	33
รูปที่ 3.2 ขาของไอซี EM4095	34
รูปที่ 3.3 วงจรที่ใช้ต่อ EM4095 กับไมโครคอนโทรลเลอร์	34
รูปที่ 3.4 หลักการของระบบ RFID	35
รูปที่ 3.5 ระบบ RFID โดยใช้ T5557	36
รูปที่ 3.6 บล็อกไดอะแกรม	36
รูปที่ 3.7 การเริ่มต้นต่อระหว่าง Reader กับ Tag	37
รูปที่ 3.8 รูปร่างสัญญาณขณะเขียนข้อมูล	38
รูปที่ 3.9 รูปแบบในการเขียน	38
รูปที่ 3.10 โปรแกรมสร้างฐานข้อมูล	41
รูปที่ 3.11 ระบบฐานข้อมูลของรายการสินค้า	42
รูปที่ 3.12 ฐานข้อมูลของข้อมูลสินค้า	43
รูปที่ 3.13 แสดงขั้นตอนการทำงานของโปรแกรม Read	45
รูปที่ 3.14 แสดงขั้นตอนการทำงานของโปรแกรม Write	45
รูปที่ 3.15 ขั้นตอนการทำงานของระบบการซื้อ-ขายสินค้า	47
รูปที่ 3.16 บล็อกไดอะแกรมของส่วนควบคุมการเปิดปิดเครื่องอ่านสัญญาณ	48
รูปที่ 3.17 บอร์ดอีเธอร์เน็ต ไอ / โอ	49
รูปที่ 3.18 บล็อกไดอะแกรมของบอร์ดอีเธอร์เน็ต ไอ / โอ	49
รูปที่ 3.19 วงจรเชื่อมต่อ MAX232 หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และ ไมโครคอนโทรลเลอร์ MCS – 51	50
รูปที่ 3.20 แสดงการทำงานของไมโครคอนโทรลเลอร์	51
รูปที่ 3.21 แสดงวงจรในส่วนควบคุมไมโครคอนโทรลเลอร์	52
รูปที่ 3.22 การประยุกต์ใช้งานของการ์ด Ethernet IO	53

#### บทที่ 4 การทดลองและผลการทดลอง

รูปที่ 4.1 วงจร ไอซี EM4095	54
รูปที่ 4.2 ความถี่และขนาดของสัญญาณที่สายอากาศขาที่ 1 ของวงจรของไอซี EM4095	54
รูปที่ 4.3 ความถี่และขนาดของสัญญาณที่สายอากาศขาที่ 2 ของวงจรของไอซี EM4095	55
รูปที่ 4.4 ความถี่ของสัญญาณที่สายอากาศขาที่ 1 และ 2 เปรียบเทียบกันของวงจรของไอซี EM4095	55
รูปที่ 4.5 ขนาดของสัญญาณที่สายอากาศขาที่ 1 และ 2 เปรียบเทียบกันของวงจรของไอซี EM4095	56
รูปที่ 4.6 สัญญาณที่สายอากาศขาที่ 1 ของวงจรของไอซี EM4095 ขณะมีแผ่นป้ายข้อมูล	56
รูปที่ 4.7 สัญญาณที่สายอากาศขาที่ 1 เมื่อนำมาขยายออกของวงจรของไอซี EM4095	57
รูปที่ 4.8 สัญญาณที่เปรียบสัญญาณก่อนที่จะเข้าไอซี EM4095 ที่วัดจากสายอากาศที่ 1 และหลังจากที่ผ่าน ไอซี EM4095 ซึ่งวัดจากขา DEMOD_OUT	57
รูปที่ 4.9 ส่วนหัวของสัญญาณที่วัดจากสายอากาศขาที่ 1 ของวงจรของไอซี EM4095	58
รูปที่ 4.10 การเปรียบเทียบสัญญาณส่วนหัว ก่อนเข้าและหลังเข้าวงจรของไอซี EM4095	58
รูปที่ 4.11 แสดงฟอร์มการขายสินค้า	59
รูปที่ 4.12 แสดงฟอร์มการเพิ่มสินค้าลงในฐานข้อมูล	60
รูปที่ 4.13 แสดงฟอร์มรายละเอียดของลูกค้า	60
รูปที่ 4.14 แสดงฟอร์มข้อมูลของสินค้า	61
รูปที่ 4.15 แสดงฟอร์มการแก้ไขข้อมูลสินค้า	61
รูปที่ 4.16 แสดงรายการสินค้าผ่านเว็บไซต์	62
รูปที่ 4.17 หน้าต่างควบคุมอุปกรณ์เครื่องอ่านสัญญาณ เมื่อทำการรัน โปรแกรม	62
รูปที่ 4.18 หน้าต่างควบคุมอุปกรณ์เครื่องอ่านสัญญาณ เมื่อทำการเชื่อมต่อกับบอร์ด	63
รูปที่ 4.19 หน้าต่างโปรแกรมที่สามารถควบคุมเครื่องอ่านสัญญาณได้	63
รูปที่ 4.20 หน้าต่างโปรแกรมที่ถูกหยุดการเชื่อมต่อเรียบร้อยแล้ว	64

## สารบัญตาราง

	หน้า
<b>บทที่ 2 ทฤษฎีและหลักการ</b>	
ตารางที่ 2.1 ย่านความถี่ต่างๆ ของระบบ RFID และการใช้งาน	6
ตารางที่ 2.2 แสดงการใช้พลังงานของหลายๆ RFID-ASIC พลังงาน โวลเตจที่น้อยที่สุดที่ไม่โครชิพสามารถทำงานได้คือ 1.8V และมากที่สุดที่10V	16
<b>บทที่ 3 การคำนวณและการออกแบบ</b>	
ตารางที่ 3.1 การถอดรหัสข้อมูลที่เขียน	37
ตารางที่ 3.2 แสดงการส่ง Command Read ของ GUI	39
ตารางที่ 3.3 แสดงการส่ง Command Read กรณีที่ไม่มีความผิดพลาด	39
ตารางที่ 3.4 แสดงการส่ง Command Read กรณีที่มีความผิดพลาดเกิดขึ้น	39
ตารางที่ 3.5 แสดงการส่ง Comand Write ของ GUI	39
ตารางที่ 3.6 แสดงสถานะของการอ่านข้อมูล	39
ตารางที่ 3.7 แสดงโค้ดคำสั่ง	39
ตารางที่ 3.8 แสดงโค้ดของเซคเตอร์และสถานะ	40
ตารางที่ 3.9 แสดงโค้ดเซกเมนต์	40
ตารางที่ 3.10 รายละเอียดของข้อมูลในฐานข้อมูล	45

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

ปัจจุบันการพัฒนาทางด้านเทคโนโลยีถือว่าการด้ว้งซึ่งถึงความก้าวหน้าของประเทศ ดังนั้นจึงมีการแข่งขันทางการพัฒนาเทคโนโลยีอย่างมากเพื่อให้เทคโนโลยีนั้นๆเป็นที่ยอมรับและถูกนำมาใช้อย่างแพร่หลาย ณ ปัจจุบันในประเทศไทยนั้นมีเทคโนโลยีเกี่ยวกับการสื่อสารอยู่หลายเทคโนโลยี ซึ่งหนึ่งในเทคโนโลยีที่น่าสนใจนั้นก็คือ Radio Frequency Identification (RFID)

เทคโนโลยี RFID เป็นเทคโนโลยีไร้สายที่ใช้คลื่นความถี่วิทยุในการระบุลักษณะเฉพาะของวัตถุแต่ละชิ้น โดยการติดป้ายฉลากด้วยแผ่นอิเล็กทรอนิกส์ (Tags) ที่มีการลงโปรแกรมควบคุมที่ระบุอย่างเฉพาะเจาะจง โดยติดไปกับวัตถุที่ต้องการตรวจสอบ และระบุถึงข้อมูลของวัตถุนั้นๆ กล่าวคือแนวคิดในการทำงานนั้นเช่นเดียวกับบาร์โค้ด ซึ่งเทคโนโลยี RFID นั้นจะประกอบด้วย 2 ส่วนดังนี้

1. Label หรือ RFID Tag คือแผงวงจรวิทยุขนาดเล็กบรรจุข้อมูลความจำ (Memory chip) บนแผ่นกระดาษขนาด 2 ตารางนิ้ว ประกอบด้วยแผงวงจรไมโครชิพกับเสาอากาศขนาดเล็กที่ฝังเป็นส่วนหนึ่งของแท็ก
2. Reader หรือ เครื่องอ่านสัญญาณ

เราสามารถอธิบายระบบได้ดังนี้

1. ส่วนของคอมพิวเตอร์เป็นส่วนที่ใช้แสดงของข้อมูลต่างๆที่ได้จากการอ่านแท็กและเป็นส่วนที่ส่งผ่านข้อมูลที่เราต้องการใส่ไว้ในแท็กไปให้เครื่องอ่านสัญญาณ นอกจากนี้ยังเชื่อมต่อกับฐานข้อมูลในระบบ และยังทำหน้าที่เป็นเครื่องเซิร์ฟเวอร์ให้กับระบบอีกด้วย
2. ส่วนของเครื่องอ่านสัญญาณจะสร้างคลื่นสัญญาณวิทยุขึ้นซึ่งความถี่ที่สร้างขึ้นจะมีขนาดเท่ากับที่แท็ก(RFID Tags) เพื่อที่จะอ่านข้อมูลและป้อนข้อมูลให้กับแผ่นป้ายระบุข้อมูล
3. ส่วนของแท็ก (RFID Tags) เป็นส่วนที่เก็บข้อมูลของสินค้าเอาไว้โดยที่จะสามารถอ่านออกมาและป้อนข้อมูลเข้าไปได้โดยผ่านทางเครื่องอ่านสัญญาณ

และในปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทในการทำเป็นที่เก็บข้อมูลหรือจะใช้ติดต่อกันผ่านอินเทอร์เน็ตรวมไปถึงการควบคุมและใช้งานอุปกรณ์ต่าง ๆ ซึ่งในปัจจุบันนี้เราไม่จำเป็นต้องใช้คอมพิวเตอร์ในการเชื่อมต่อเพียงอย่างเดียวแล้ว โดยได้มีอุปกรณ์ที่เรียกว่า อิเทอร์เน็ต ไอโอ บอร์ด ซึ่งจะช่วยให้เราสามารถประหยัดคอมพิวเตอร์ไปได้หนึ่งเครื่อง และเรายังสามารถที่จะพัฒนาให้สามารถควบคุมการเปิดปิดของเครื่องอ่านสัญญาณได้ที่ละหลาย ๆ เครื่องอีกด้วย

### 1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาการส่งผ่านข้อมูลระหว่างแท็กกับเครื่องอ่านสัญญาณ
2. ศึกษาการส่งผ่านข้อมูลจากเครื่องอ่านสัญญาณไปยังคอมพิวเตอร์โดยผ่านพอร์ตอนุกรม
3. ศึกษาการสร้างระบบฐานข้อมูลและการเชื่อมต่อฐานข้อมูลผ่านอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. พัฒนาโปรแกรม Delphi เพื่อใช้ในการเชื่อมต่อระหว่างเครื่องอ่านสัญญาณกับคอมพิวเตอร์
5. ศึกษาการควบคุมการเปิดการปิดเครื่องอ่านสัญญาณผ่านอิเทอร์เน็ตไอโอบอร์ด

### 1.3 ขอบเขตของโครงการ

1. ในส่วนของการติดต่อกันระหว่างเครื่องอ่านสัญญาณกับแทกส์ เราจะใช้อุปกรณ์สำเร็จรูป
2. ในส่วนการเชื่อมต่อระหว่างเครื่องอ่านสัญญาณกับคอมพิวเตอร์นั้นเราจะใช้โปรแกรม Delphi มาเป็นตัวเชื่อมต่อเพื่อส่งผ่านข้อมูลต่างๆผ่านเครื่องอ่านสัญญาณไปยังแทกส์
3. ในส่วนของระบบฐานข้อมูล เราจะนำข้อมูลที่เรานำข้อมูลทั้งหมดของระบบเก็บไว้ เพื่อที่จะสามารถเรียกนำมาตรวจสอบได้ทั้งทางเครื่องคอมพิวเตอร์ที่เป็นเซิร์ฟเวอร์และทางอินเทอร์เน็ต โดยผ่านทางเว็บไซต์ที่เราสร้างขึ้น ซึ่งโปรแกรมที่ใช้สร้างระบบฐานข้อมูลจะ ประกอบด้วย โปรแกรม MySQL, PHP โดยนำมาใช้ในรูปแบบต่างๆดังนี้

#### MySQL

- บริหารจัดการด้านฐานข้อมูล เพื่อให้ระบบสามารถเรียกใช้ข้อมูล ได้อย่างสะดวกขึ้น

#### PHP

- เชื่อมต่อข้อมูลจากระบบฐานข้อมูลและนำมาแสดงผลผ่านทางเว็บไซต์

4. การควบคุมการปิดเปิดเครื่องอ่านสัญญาณผ่านเครือข่ายอินเทอร์เน็ต โดยควบคุมผ่านการัด Ethernet IO

### 1.4 วิธีดำเนินงาน

ในโครงการนี้เริ่มต้นโดยการศึกษาทฤษฎีต่างๆของเทคโนโลยี RFID รวมทั้งทำการทดลองเกี่ยวกับวงจรภายในเครื่องอ่านสัญญาณเพื่อวัดสัญญาณต่างๆ ต่อมาจึงศึกษาการติดตั้ง เซิร์ฟเวอร์ให้สามารถเชื่อมต่อกับอินเทอร์เน็ตได้และสามารถประมวลผลผ่านทางเว็บไซต์ได้ แล้วจึงศึกษาทฤษฎีการเขียน โปรแกรมเพื่อติดต่อกันระหว่างเครื่องอ่านสัญญาณกับคอมพิวเตอร์ เมื่อสามารถติดต่อกันได้แล้วจะทำการทดสอบการส่งและอ่านข้อมูลของเครื่องอ่านสัญญาณกับแทกส์

หลังจากนั้นจะนำข้อมูลที่ส่งและอ่านได้จากแทกส์เข้าไปเก็บไว้ในระบบฐานข้อมูลแล้วข้อมูลต่างๆจะสามารถเรียกดูได้โดยผ่านทางเว็บไซต์

และเชื่อมต่อเครื่องอ่านสัญญาณผ่านระบบเครือข่าย เพื่อให้สามารถควบคุมการปิดเปิดของเครื่องอ่านสัญญาณผ่านระบบเครือข่ายได้นั้น เราจะใช้ อิเทอร์เน็ต ไอโอ

## บทที่ 2

### ทฤษฎีหรือหลักการ

ระบบตรวจสอบข้อมูลสินค้าในปริมาณนี้ ในการสร้างระบบดังกล่าวจะประกอบด้วยส่วนหลักๆ ด้วยกัน 4 ส่วน คือ

#### 1. ส่วนของการเชื่อมต่อระหว่างเครื่องอ่านสัญญาณและคอมพิวเตอร์

ในส่วนนี้เราต้องศึกษาไปโคคอลลในการรับและส่งข้อมูลจากเครื่องอ่านสัญญาณไปยังคอมพิวเตอร์และนำมาพัฒนาบนโปรแกรม Delphi เพื่อให้ได้แอปพลิเคชันที่สามารถรับส่งข้อมูลได้

#### 2. ส่วนของระบบฐานข้อมูล

ในส่วนนี้จะใช้ MySQL มาใช้ในการสร้างฐานข้อมูลบนเครื่องเซิร์ฟเวอร์และใช้ PHP เป็นตัวเชื่อมฐานข้อมูลมาแสดงผลบนเวปไซต์

#### 3. ส่วนของการเชื่อมต่อข้อมูลต่างๆ ไปยังระบบฐานข้อมูล

ในส่วนนี้จะเป็นส่วนที่นำข้อมูลที่อ่านได้จากเครื่องอ่านสัญญาณผ่านคอมพิวเตอร์และนำไปเก็บไว้ในฐานข้อมูล ซึ่งเราจะต้องนำแอปพลิเคชันของโปรแกรม Delphi ที่เราได้พัฒนาขึ้นมาเชื่อมต่อกับฐานข้อมูลที่สร้างขึ้นด้วยโปรแกรม MySQL เช่น

#### 4. ส่วนของการควบคุมการเปิด- การปิดเครื่องอ่านสัญญาณผ่านระบบเครือข่าย

ในส่วนนี้เราจะใช้การ์ดอินเตอร์เฟซควบคุม

### 2.1 หลักการทำงานของ RFID

โดยทั่วไประบบ RFID จะประกอบไปด้วย 2 ส่วนสำคัญคือ

#### 2.1.1 ฉลากอิเล็กทรอนิกส์ ( Transponder ) หรือเรียกง่าย ๆ ว่า แท็กส์

จะเป็นส่วนของการเก็บข้อมูลและถูกติดตั้งอยู่กับวัตถุที่เราต้องการชี้บ่ง ( Identify ) แท็กส์จะประกอบไปด้วยเสาอากาศทำหน้าที่คล่องสัญญาณที่ส่งมาจาก เครื่องอ่านและส่วนของ ไมโครชิพ

##### 2.1.1.1 ชนิดของแท็กส์

แท็กส์ที่มีการใช้งานอยู่นั้นจะมีอยู่ 2 ชนิดใหญ่ๆ โดยแต่ละชนิดก็จะมี ความแตกต่างกันในแง่ของ การใช้งาน ราคา โครงสร้างและหลักการทำงานอยู่ ซึ่งจะขอล่าวและอธิบายแยกเป็นหัวข้อดังนี้

##### 1. แท็กส์ของ RFID แบบพาสซีฟ ( Passive RFID Tags )

แท็กส์ ชนิดนี้ไม่จำเป็นต้องรับแหล่งจ่ายไฟใดๆ เพราะมีวงจรกำเนิดไฟฟ้าเหนี่ยวนำขนาดเล็กเป็น แหล่งจ่ายไฟในตัวอยู่แล้ว ระยะการสื่อสารข้อมูลที่ทำได้สูงสุด 1.5 เมตร มีหน่วยความจำขนาดเล็ก (ทั่วไป ประมาณ 32 – 128 บิต ) มีขนาดเล็กและน้ำหนักเบา ราคาต่อหน่วยต่ำ ไอซีของแท็กส์ ชนิดพาสซีฟที่มีการผลิตออกมาจะมีทั้งขนาดและรูปร่างเป็นได้ตั้งแต่แท่งหรือแผ่นขนาดเล็กจนแทบไม่สามารถมองเป็นไปจนถึงขนาดใหญ่จนสะดุดตา ซึ่งต่างก็มีความเหมาะสมกับชนิดงานที่แตกต่างกัน โดยทั่วไป โครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในส่วนที่เป็นไอซีของ แทกส์ นั้นก็จะประกอบด้วย 3 ส่วนหลักๆ ได้แก่ ส่วนของหน่วยความจำ , ส่วนควบคุมภาคลอจิก และส่วนของควบคุมการทำงานของภาครับส่งสัญญาณวิทยุ

## 2. แทกส์ของ RFID แบบแอคทีฟ (Active RFID Tags)

แทกส์ ชนิดนี้จะใช้แหล่งจ่ายไฟจากแบตเตอรี่ขนาดเล็ก มีหน่วยความจำภายในขนาดใหญ่ได้ถึง 1 เมกะไบต์ มีระยะการสื่อสารข้อมูลที่ได้สูงสุดถึง 6 เมตร แม้ว่าแทกส์ชนิดนี้จะมีข้อดีอยู่หลายข้อแต่ก็มีข้อเสียอยู่ด้วยเหมือนกัน เช่น มีราคาต่อหน่วยแพง มีขนาดค่อนข้างใหญ่ และมีระยะเวลาในการทำงานที่จำกัด นอกจากการแบ่งจากชนิดที่นำมาแล้ว แทกส์ ก็ยังถูกแบ่งประเภทจากรูปแบบในการใช้งานได้แบบ 3 แบบ คือ แบบที่สามารถถูกอ่านและเขียนข้อมูลได้อย่างอิสระ (Read – write) , แบบเขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้อย่างอิสระ ( Write- Once , Read – Many หรือ WORM ) และแบบอ่านได้เพียงอย่างเดียว ( Read – only ) ด้วย

### 2.1.1.2 เทคโนโลยีของหน่วยความจำ

#### 1. RAM (Random Access Memory)

เป็นหน่วยความจำที่ใช้เก็บข้อมูลชั่วคราว ข้อมูลจะสูญหายเมื่อไม่มีไฟเลี้ยง ดังนั้นหน่วยความจำชนิดนี้จึงจำเป็นต้องมีแบตเตอรี่สำรองอยู่ภายใน แทกส์ ด้วย ข้อดีของหน่วยความจำชนิดนี้คือ สามารถเก็บข้อมูลได้มาก เวลาในการอ่านหรือเขียนข้อมูลนั้นสั้น

#### 2. EEPROM (Electric Erasable PROM)

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

#### 3. FRAM (Ferroelectric RAM)

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

แทกส์ของบริษัท microchip เบอร์ T5551 มีรายละเอียดคร่าวๆดังนี้

1. แทกส์จะทำการเก็บข้อมูลเป็นบล็อก บล็อกละ 32 บิต จำนวน 8 บล็อก
2. แทกส์จะทำการส่งข้อมูลทั้ง 8 บล็อกออกมาอย่างต่อเนื่อง วนไปเรื่อยๆ โดยเมื่อวนมาครบ 1

รอบ แล้วกำลังจะทำการส่งข้อมูลในบล็อกที่ 1 จะมีสัญญาณ sequence terminator เพื่อบอกจุด  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นของชุดข้อมูล

3. สัญญาณ sequence terminator นี้ จะกำหนดรูปแบบคองที่ดังนี้

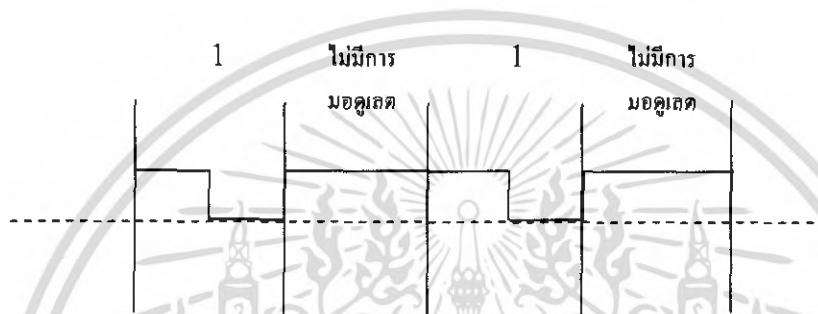
บิตที่ 1 เป็นสัญญาณ “1”

บิตที่ 2 เป็นสัญญาณที่ไม่ถูกมอดูเลต เพราะฉะนั้นจะเป็นการส่งคลื่นพาห่อออกไป

บิตที่ 3 เป็นสัญญาณ “1”

บิตที่ 4 มีค่าเหมือนบิตที่ 2

รูปสัญญาณ sequence terminator ของรหัส Manchester แสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 รูปสัญญาณ sequence terminator ของรหัส Manchester

4. สามารถมอดูเลตสัญญาณส่งออกได้หลายแบบและกำหนดบิตเรตในการส่งข้อมูลได้

#### 2.1.2 เครื่องอ่านสัญญาณ (Reader)

จะประกอบไปด้วยภาครับ/ส่งสัญญาณวิทยุ ส่วนควบคุมและเสาอากาศ (Antenna) ทำหน้าที่คล้องสัญญาณ กับ แทกส์ และส่วนของการติดต่อสื่อสารกับอุปกรณ์ควบคุมภายนอก (RS232C) ขึ้นอยู่กับลักษณะการสื่อสารที่เราต้องการ เครื่องอ่านสัญญาณนั้น นอกจากสามารถจะอ่านข้อมูลได้แล้วยังสามารถเขียนข้อมูลได้ด้วย

#### 2.1.3 ความถี่ต่าง ๆ ของคลื่นพาหะ

ในปัจจุบันได้มีการรวมกลุ่มระหว่างแต่ละประเทศ เพื่อทำการกำหนดมาตรฐานความถี่คลื่นพาหะของระบบ RFID โดยมีสามกลุ่มใหญ่ๆ คือ กลุ่มประเทศในยุโรปและแอฟริกา (Region 1), กลุ่มประเทศอเมริกาเหนือและอเมริกาใต้ (Region 2) และสุดท้ายคือกลุ่มประเทศตะวันออกไกลและออสเตรเลีย (Region 3) ซึ่งแต่ละกลุ่มประเทศจะกำหนดแนวทางในการเลือกใช้ความถี่ต่างๆ ให้แก่บรรดาประเทศสมาชิก อย่างไรก็ตาม ความถี่ของคลื่นพาหะที่นิยมใช้งานในย่านความถี่ต่ำ ย่านความถี่ปานกลาง และย่านความถี่สูงก็คือ 125 kHz, 13.56 MHz และ 2.45 GHz ตามลำดับดังที่แสดงไว้ในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้รัฐบาลของแต่ละประเทศ โดยทั่วไปจะมีการออกกฎหมายเกี่ยวกับระเบียบการใช้งานย่านความถี่ต่างๆ รวมถึงกำลังส่งของระบบ RFID ด้วย

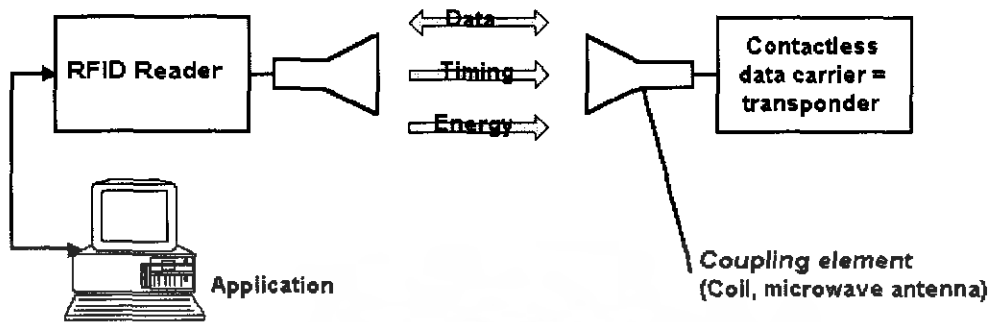
ย่านความถี่	คุณลักษณะ	การใช้งาน
ย่านความถี่ต่ำ 100-500 kHz ความถี่มาตรฐานที่ใช้งานทั่วไปคือ 125 kHz	-ระยะการรับส่งข้อมูลใกล้ -ต้นทุนไม่สูง -ความเร็วในการอ่านข้อมูลต่ำ -ความถี่ในย่านนี้เป็นที่แพร่หลายทั่วโลก	-Access Control -ปลุสตัดว์ -ระบบคลัง -รอยนต์
ย่านความถี่กลาง 10-15 MHz ความถี่มาตรฐานที่ใช้งานทั่วไปคือ 13.56 MHz	-ระยะการรับส่งข้อมูลปานกลาง -ราคามีแนวโน้มถูกลงในอนาคต -ความเร็วในการอ่านข้อมูลปานกลาง -ความถี่ในย่านนี้เป็นที่แพร่หลายทั่วโลก	-Access Control -สมาร์ทการ์ด
ย่านความถี่สูง 850-950 MHz 2.4-5.8 GHz ความถี่มาตรฐานที่ใช้งานทั่วไปคือ 2.45 GHz	-ระยะการรับส่งข้อมูลไกล (10 เมตร) -ความเร็วในการอ่านข้อมูลสูง -ราคาแพง	-รถไฟ -ระบบเก็บค่าผ่านทาง

ตารางที่ 2.1 ย่านความถี่ต่างๆ ของระบบ RFID และการใช้งาน

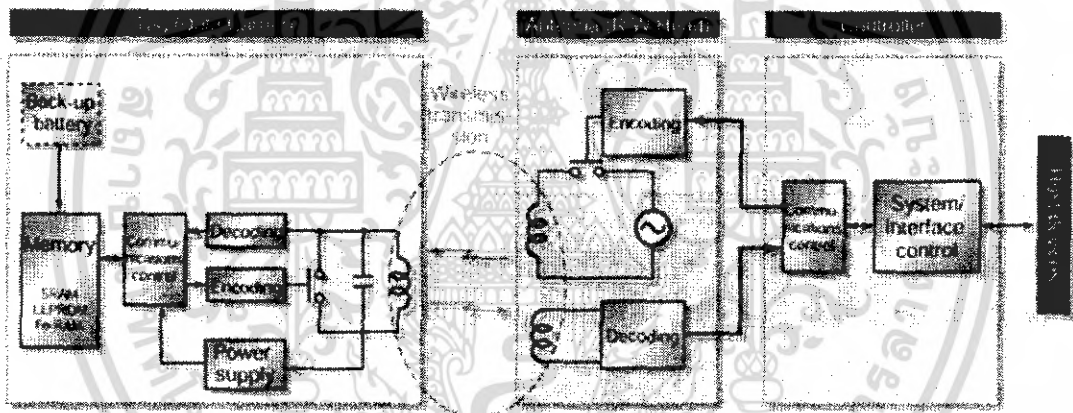
#### 2.1.4 RFID ทำงานอย่างไร

จากวงจรเทียบเคียงส่วนควบคุมและติดต่อสื่อสาร ( Control and Interface ) จะได้รับคำสั่ง (Command ) จากส่วนควบคุมที่สูงกว่า ( Host ) เช่น คอมพิวเตอร์ หรือ PLC ( Programmable Logic Controller ) จากนั้นตัวควบคุมจะทำการประมวลผลคำสั่งว่า Host ต้องการให้ทำอะไร จากนั้นก็จะส่งให้ส่วนของภาครับ / ส่งวิทยุ ที่มีส่วนของวงจรเข้ารหัส ( Coding ) จากนั้นส่วนของวงจรผสมสัญญาณ ( Modulation ) ทำการผสมข้อมูลเข้ากับคลื่นพาหะแล้วทำการส่งออกไปทางเสาอากาศ ขนาดของพื้นที่ที่มีสัญญาณอยู่นั้นจะขึ้นอยู่กับขนาดของเสาอากาศและพลังงาน ( Watt ) ของเสาอากาศ เมื่อแท็กเข้ามาในพื้นที่ที่มีสัญญาณแล้ว เสาอากาศภายในแท็กจะได้รับการคล้องสัญญาณทำให้แท็กทำงานได้ตามที่กล่าวมาแล้ว วงจรถอดรหัส ( Demodulation ) จะทำการแยกสัญญาณข้อมูล ที่ถูกผสมมาจากเครื่องอ่านออกจากคลื่นพาหะแล้วทำการแปลรหัส ( Decoding ) จากนั้น CPU ของแท็กจะรับคำสั่งไปประมวลผลถ้าเป็นคำสั่งเขียนแท็กจะบันทึกข้อมูลที่ส่งมาลงในหน่วยความจำของแท็ก แต่ถ้าเป็นคำสั่งอ่านแท็ก จะดึงข้อมูลจากหน่วยความจำที่ระบุไว้จากจากคำสั่ง แล้วทำการผสมข้อมูลที่วงจรผสมเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลภายในแท็กส์ กับคลื่นพาหะ แล้วส่งออกไปทางเสาอากาศเหมือนกัน เมื่อเครื่องอ่านได้รับสัญญาณจากแท็กส์วงจรถอดรหัสของเครื่องอ่านก็จะถอดเอาข้อมูลออกจากคลื่นพาหะและส่งไปที่ Host Unit



รูปที่ 2.2 การสื่อสารระหว่างเครื่องอ่านกับ แท็กส์



รูปที่ 2.3 วงจรเทียบเคียงของระบบ RFID

## 2.1.5 ประเภทของ ระบบ RFID

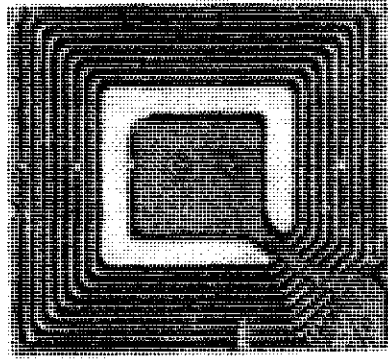
RFID ถูกจำแนกออกเป็นประเภทได้หลายอย่างขึ้นอยู่กับว่าจะถูกจำแนกจากคุณสมบัติอะไร เช่น ความถี่ที่ใช้งาน ชนิดของ แท็กส์ หรือ ขนาดของหน่วยความจำของ แท็กส์ ที่ใช้

### 2.1.5.1 RFID ที่จำแนกโดยขนาดของหน่วยความจำ

#### 1) RFID ชนิด 1 บิต ( 2 Bit Type)

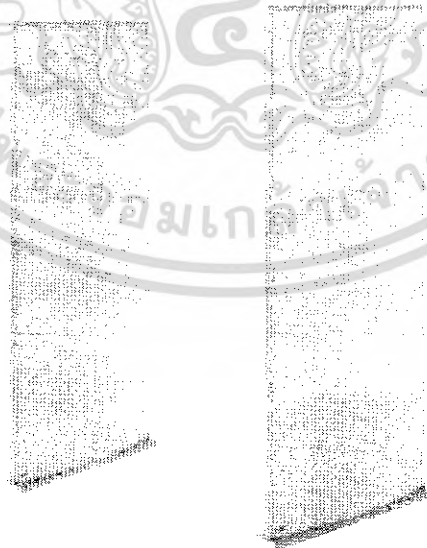
RFID ชนิดนี้หรือเรียกอีกอย่างหนึ่งว่า EAS (Electronic Article Surveillance) เป็น RFID ที่ใช้แท็กส์ที่ไม่มีไมโครชิพ RFID ระบบนี้จะตรวจสอบเฉพาะว่ามีแท็กส์อยู่ในพื้นที่สัญญาณหรือไม่ ดังนั้นสถานะจึงแสดงเพียงแอมป์หรือไม่มี ซึ่งเป็นรหัสดิจิทัล 0 หรือ 1 นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แทงส์ ของระบบ EAS จะมีเฉพาะเสาอากาศอย่างเดียว

เนื่องจากการทำงานของ RFID ระบบนี้ง่ายไม่ซับซ้อนและ แทงส์ มีราคาถูกมาก ระบบจึงถูกนำมาใช้กับการป้องกันสินค้าถูกขโมยในห้างร้านต่างๆ โดย แทงส์ จะติดอยู่กับตัวสินค้าหรือซ่อนไว้ข้างหลังบาร์โค้ดอีกที เครื่องอ่านจะออกแบบเป็นโครงเสาอากาศสูงประมาณ 1-12 เมตร อยู่ที่ทางออกของห้างร้านนั้นๆ เมื่อสินค้าที่มี แทงส์ติดอยู่ผ่านมาในบริเวณเสาอากาศนี้ก็จะถูกตรวจจับได้จากการคลั่งสัญญาณจากเสาอากาศและแทงส์ จึงสามารถตรวจจับสินค้าที่ถูกนำออกไปได้ แทงส์จะถูกดึงออกหรือใช้เครื่องทำลายความเป็นสนามแม่เหล็กไฟฟ้าที่เกาน์เตอร์แคชเชอร์ เมื่อสินค้าถูกนำมาชำระ ซึ่งเราจะพบเห็นระบบ EAS ในซูเปอร์มาร์เก็ตหรือ ดิสเคาน์ตโร้ตต่างๆ เช่น คาร์ฟูร์ โลตัส BigC หรือร้านขายเสื้อผ้า CD ต่างๆ ช่วงความถี่ที่ใช้งานจะเป็นช่วงความถี่ microwave เนื่องจากระยะในการสื่อสารค่อนข้างไกล (ขนาดระหว่างเสาอากาศ)



รูปที่ 2.5 เสาอากาศของระบบ EAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

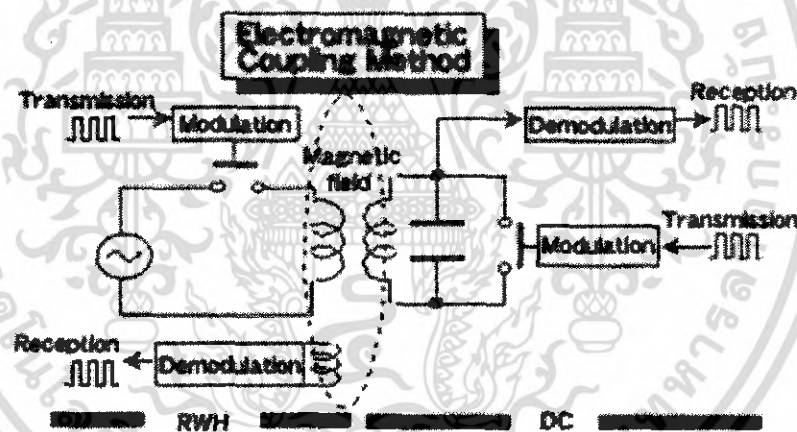
## 2) RFID ชนิดหน่วยความจำมากกว่า 1 บิต (Data Carrier Type)

RFID ชนิดนี้จะใช้แท็กที่มีไมโครชิพและหน่วยความจำที่เป็นส่วนประกอบสำคัญ มีราคาสูงกว่าแท็กชนิด EAS โดยบางชนิดสามารถเก็บข้อมูลได้สูงสุดถึง 64 Kbytes ใช้ในอุตสาหกรรมหรืองานทั่วไปที่ต้องใช้แท็กในการเก็บข้อมูล

### 2.1.5.2 RFID ที่จำแนกโดยลักษณะการสื่อสารของสัญญาณ

#### 1) Close Coupling

เป็น RFID ที่มีระยะในการอ่าน /เขียนข้อมูลสั้นมากประมาณ 0-1 เซนติเมตร ดังนั้น แท็กจะต้องอยู่ใกล้หรือวางอยู่บนเครื่องอ่าน Close Coupling นี้จะสามารถใช้คลื่นความถี่ได้ตั้งแต่ 0 Hz จนถึง 50 MHz เนื่องจากการทำงานของแท็ก ไม่อาศัยการส่งพลังงานจากการแผ่คลื่นแม่เหล็กไฟฟ้าจากเครื่องอ่านแต่อาศัยการเหนี่ยวนำเหมือนหลักการของหม้อแปลงไฟฟ้า ทำให้เกิดพลังงานที่ทำให้วงจรภายในแท็กทำงานได้ ระบบ Close-Coupling จะนิยมนำมาใช้งานที่ต้องการความปลอดภัยค่อนข้างสูง แต่ไม่ต้องการติดต่อกับไอทีไอ เช่น ประตูอัตโนมัติ หรือ สมาร์ทการ์ด ไร้สัมผัส ( Contactless Smart Cards )

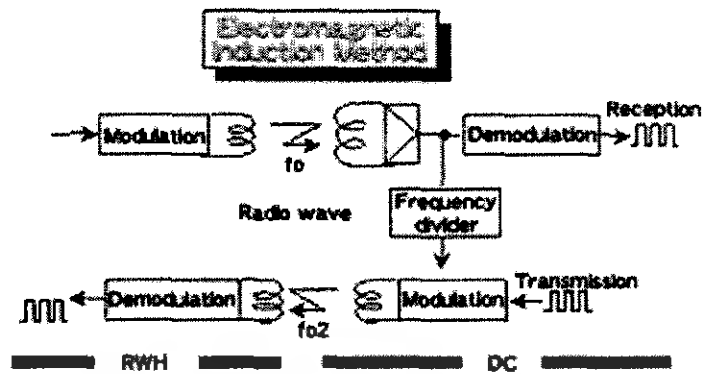


รูปที่ 2.6 วงจรเทียบเคียงของ Close - Coupling

#### 2) Remote Coupling

เป็นระบบที่มีระยะการอ่าน /เขียนสูงถึง 1 เมตร ระบบนี้จะใช้หลักการสื่อสารสัญญาณแบบ Inductive ( Magnetic ) Coupling ระหว่างเครื่องอ่านกับ แท็ก ประมาณ 90 - 95% ของระบบ RFID ในปัจจุบัน ใช้หลักการ Remote Coupling นี้โดยความถี่ที่ใช้งานมีหลายความถี่ตั้งแต่ต่ำกว่า 135 KHz หรือ 13.56 MHz และ 27.125 MHz พลังงานไฟฟ้าจะถูกส่งโดยหลักการแผ่ของคลื่นแม่เหล็กไฟฟ้าไปให้แท็ก ทำให้แท็กได้รับพลังงานสามารถทำงานได้ ระบบ Remote Coupling นี้จะพบมากในลักษณะงานอุตสาหกรรมเช่น รถยนต์ เครื่องใช้ไฟฟ้าหรือ อิเล็กทรอนิกส์

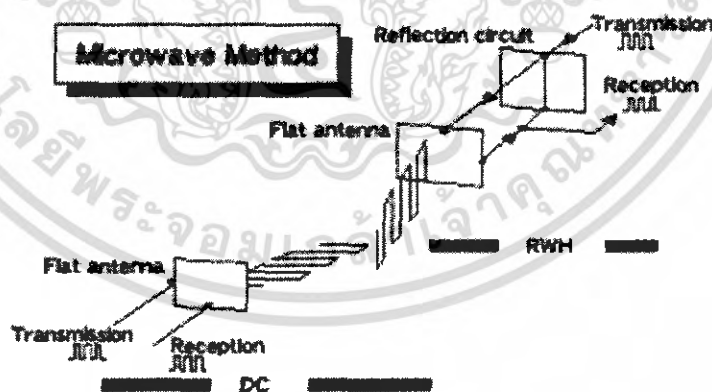
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 วงจรเทียบเคียงของ Remote Coupling หรือ Inductive Coupling

3) Long Range

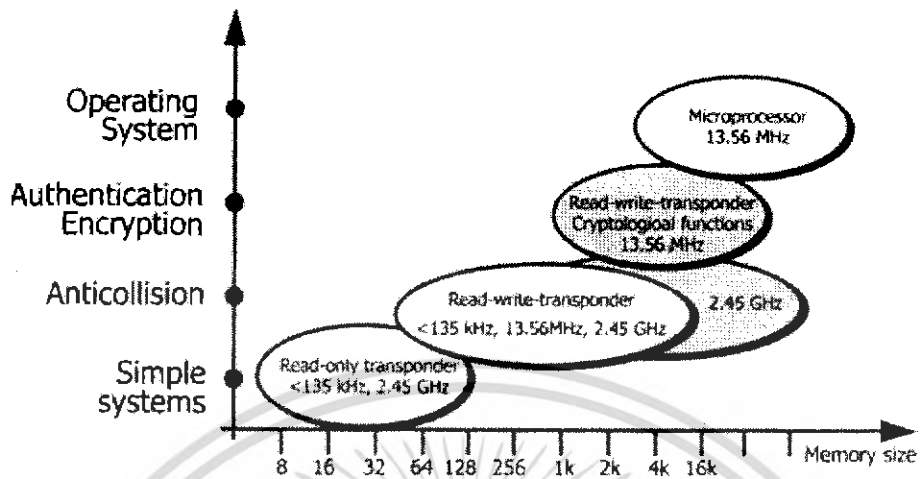
ระบบนี้จะมีระยะการอ่าน / เขียนอยู่ระหว่าง 1 ถึง 10 เมตรหรือบางระบบอาจสูงกว่านี้ ความถี่ที่ใช้ในระบบนี้จะเป็นย่านที่มีความถี่สูงมากหรือ ไมโครเวฟ (Microwave range) ซึ่งปกติที่ความถี่ 2.45 GHz หรือบางครั้งจะพบที่ 915 MHz, 5.8 GHz และ 24.125 GHz แต่การส่งพลังงานจากตัวเครื่องอ่านไปยังแท็กทำได้ยาก ดังนั้นแท็กที่ใช้งานจะเป็นชนิดที่มีแบตเตอรี่ในตัวซึ่งจะใช้สำหรับเป็นไฟเลี้ยงที่ทำให้ไมโครชิปทำงานและเก็บรักษาข้อมูล ลักษณะงานที่พบเห็นจะเป็นลักษณะงานที่ต้องการการสื่อสารระยะไกล เช่น ในกระบวนการผลิตรถยนต์ ระบบชำระเงินอัตโนมัติของทางด่วน



รูปที่ 2.8 วงจรเทียบเคียง Long Range

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.3 RFID ที่จำแนกตามความสามารถของระบบ

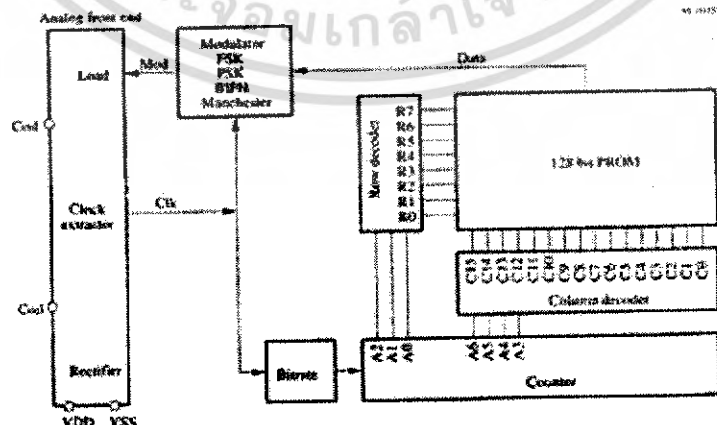


รูปที่ 2.9 ระบบ RFID ที่แบ่งตามฟังก์ชันของระบบ

1) ระบบอ่านอย่างเดียว ( Read Only System )

แท็กมีข้อมูลซึ่งจะอยู่ในรูปของ Serial Number และไม่สามารถเขียนข้อมูลใหม่ลงไป ได้เหมาะกับงานที่ต้องการอ่านอย่างเดียว เพื่อแยกแยะความแตกต่างของสินค้าหรือบุคคล ระบุ ชนิดของสินค้า หรือตู้คอนเทนเนอร์ มีราคาต่ำความถี่ที่ใช้งานจะอยู่ที่ต่ำกว่า 135KHZ หรือ 2.45 GHz

แท็กชนิดนี้จะเข้าสู่บริเวณการตรวจสอบของเครื่องอ่านสัญญาณแล้วมันเริ่มส่งตัวเลข ไอดีของมันอย่างต่อเนื่อง โดยปกติแล้วผู้ผลิตชิพจะรับรองว่าแต่ละ Serial Number นั้นจะใช้แค่ ครั้งเดียว ตัวเลขเฉพาะของแท็กจะถูกบรรจุลงไปในแท็กในระหว่างการผลิตชิพ ผู้ใช้ไม่สามารถเปลี่ยนแปลง Serial Number หรือข้อมูลบนชิพได้



รูปที่ 2.10 บล็อกไดอะแกรมของแท็กแบบอ่านอย่างเดียว เมื่อแท็กเข้าสู่บริเวณการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของเครื่องอ่านสัญญาณ การนับจะเริ่มนับทุกแอดเดรสที่อยู่ในหน่วยความจำอย่างต่อเนื่อง เอาท์พุทของข้อมูลของหน่วยความจำจะเชื่อมต่อกับโพลคมออดูเลเตอร์ที่เซตเป็นรหัสเบสแบนด์ของรหัสไบนารี ในกระบวนการนี้ข้อมูลทั้งหมดในหน่วยความจำ (128 บิตซีเรียลนัมเบอร์) สามารถส่งไปได้อย่างต่อเนื่อง

การติดต่อสื่อสารกับเครื่องอ่านสัญญาณจะเป็นแบบทิศทางเดียวโดยแทกซ์จะส่งตัวเลขไอดีของมันไปยัง reader อย่างต่อเนื่อง

## 2) ระบบอ่านเขียน ( Read-Write System )

จะจัดอยู่ Mid- range ของระบบ RFID Tags สามารถเขียนข้อมูลเข้าได้ โดยความจุอยู่ที่ 16 ไบต์ จนถึงมากกว่า 16 กิโลไบต์ การอ่านและเขียนข้อมูลในแทกซ์จะเป็นแบบบล็อกข้อมูล ในบล็อกจะมีรูปแบบเป็นจำนวนไบต์ที่สามารถอ่านและเขียนได้ การที่จะเปลี่ยนแปลงข้อมูลของแต่ละบล็อกบล็อกทั้งหมดจะถูกอ่านจากแทกซ์หลังจากนั้นในบล็อกเดียวกันรวมกับบล็อกที่แก้ไขแล้วจะถูกเขียนกลับลงไปในแทกซ์หน่วยความจำที่ใช้จะเป็นชนิด EEPROM หรือ SRAM ความถี่ที่ใช้งานจะเป็น 135KHz, 13.56MHz, 27.125MHz และ 2.45GHz

## 3) ระบบไมโครโพรเซสเซอร์ ( Microprocessor System )

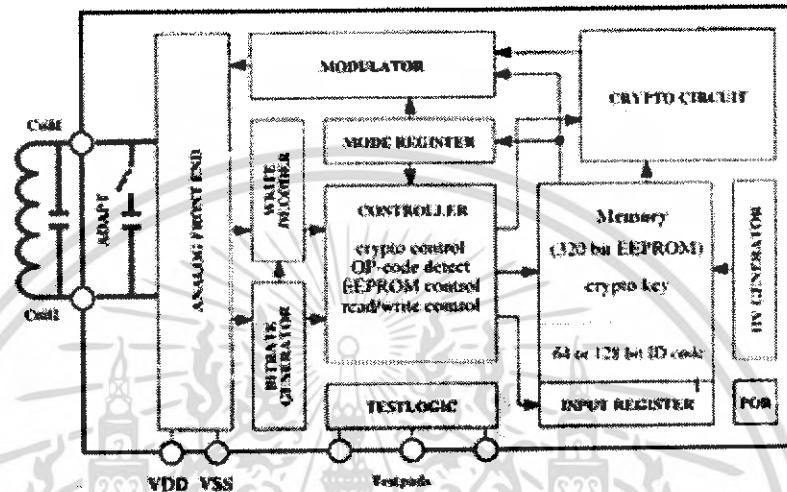
ระบบนี้จะจัดอยู่ประเภท High end เพราะว่ามีไมโครโพรเซสเซอร์เป็นตัวประมวลผลในแทกซ์สามารถประยุกต์ใช้งานที่หลากหลายและมีฟังก์ชันการสร้างรหัสลับ ( Cryptological Functions ) สามารถนำไปใช้งานที่เกี่ยวข้องกับการรักษาความปลอดภัย ส่วนมากระบบนี้จะใช้ย่านความถี่ที่ 13.56MHz หน่วยความจำที่ใช้งานจะมีขนาดตั้งแต่อย่างน้อยจนถึง 16 กิโลไบต์ และหน่วยความจำเป็นแบบ EEPROM

ถ้าแทกซ์ที่สามารถเขียนได้ไม่ได้ป้องกันเอาไว้เครื่องอ่านสัญญาณที่เป็นระบบ RFID สามารถอ่านและเขียนข้อมูลได้ซึ่งจะไม่ต้องการเพราะอาจจะเกิดความเสียหายได้โดยการอ่านหรือเขียนข้อมูลที่ไม่ได้รับอนุญาตในแทกซ์ตัวอย่างที่เป็นแทกซ์ชนิดนี้คือแทกซ์ในตั๋วเดินทางและแทกซ์ในกุญแจรถ

มีวิธีการแก้ปัญหาในกรณีที่มีการเข้าถึงแทกซ์โดยไม่ได้รับอนุญาต วิธีที่ง่ายที่สุดในการป้องกันการอ่านและเขียนข้อมูลคือการเช็คพาสเวิร์ด วิธีการก็คือการที่จะเปรียบเทียบพาสเวิร์ดที่ส่งกับพาสเวิร์ดอ้างอิง และสามารถเข้าถึงในหน่วยความจำได้ถ้าพาสเวิร์ดตรงกัน

อย่างไรก็ตามการตรวจสอบจะจำเป็นต้องตรวจสอบทั้งสององค์ประกอบของแอปพลิเคชันเดียวกัน พื้นฐานของวิธีการตรวจสอบจะรวมกับการเปรียบเทียบของ 2 secret key ที่ไม่ได้ถูกส่งผ่านอินเตอร์เฟซ cryptological ที่แท้จริงจะถูกรวมกับ encryption ของข้อมูลที่ถูกส่ง จะมีผลทำให้ป้องกันการลักลอบข้อมูลโดยแทกซ์ที่ใช้ตัวรับเป็นวิทยุ

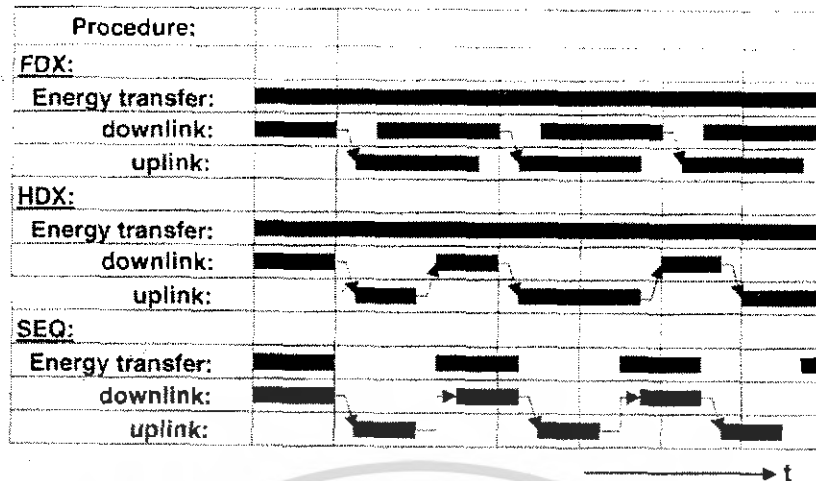
พื้นที่ในหน่วยความจำที่บรรจุในแอปพลิเคชันแทกส์ที่มีฟังก์ชันการทำงานแบบ cryptological จะมีพื้นที่ในหน่วยความจำสำหรับเก็บ secret key และ configuration register (access register, Acc) secret key จะถูกเขียนไว้ที่ key memory โดยผู้ผลิตก่อนที่แทกส์จะส่งไปให้ผู้ใช้ ซึ่ง key memory จะไม่สามารถอ่านได้



รูปที่ 2.11 บล็อกไดอะแกรมของแทกส์ที่เขียนได้ที่มีฟังก์ชัน cryptological ระหว่างแทกส์กับเครื่องอ่านสัญญาณ

## 2.1.6 กระบวนการรับส่งข้อมูลของ RFID

- 2.1.6.1 กระบวนการ Half Duplex (HDX) ข้อมูลจะส่งจากตัวส่งไปยังตัวอ่าน และสลับเปลี่ยนข้อมูลส่งจากตัวอ่านไปยังตัวส่ง
- 2.1.6.2 กระบวนการ Full Duplex (FDX) ข้อมูลจะส่งจากตัวส่งไปยังตัวอ่าน และในเวลาเดียวกันตัวอ่านจะส่งข้อมูลไปยังตัวส่งด้วย



รูปที่ 2.12 แสดง Full Duplex , Half Duplex และ Sequential System บนเวลา ข้อมูลจะส่งจากตัวเครื่องอ่านสัญญาณ ไปยังแท็กเรียกว่า downlink และจากแท็กไปยังเครื่องอ่านสัญญาณเรียกว่า uplink

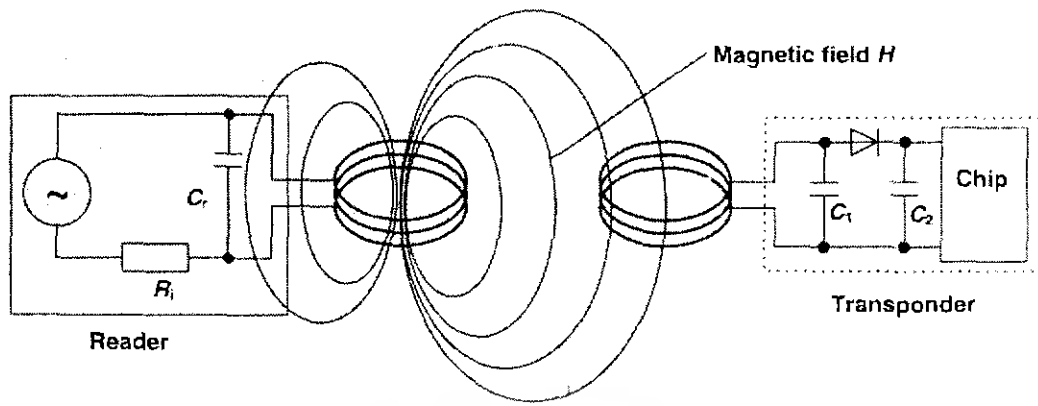
### 2.1.7 พลังงานที่จ่ายให้แท็ก

การคล้องสัญญาณ โดยการเหนี่ยวนำของแท็กประกอบด้วยเครื่องมือเก็บข้อมูลทางไฟฟ้า ส่วนใหญ่ใช้ไมโครชิพตัวเดียว และขดลวดขนาดกว้างเป็นตัวทำเสาอากาศ

การคล้องสัญญาณ โดยการเหนี่ยวนำของแท็ก ส่วนใหญ่เป็นแท็กแบบพาสซีฟ ความหมายคือพลังงานทั้งหมดใช้สำหรับการทำงานของไมโครชิพ เสาอากาศของตัวอ่านนั้นจะสร้างความถี่คลื่นสนามแม่เหล็กสูงซึ่งจะแพร่ไปรอบๆขดลวด เนื่องจากความยาวคลื่นของช่วงความถี่ที่ใช้ (<math>135\text{kHz}:2400\text{m}</math> , <math>13.56\text{MHz}:22.1\text{m}</math>) มากกว่าระยะระหว่างเสาอากาศของตัวอ่านกับตัวแท็ก สนามแม่เหล็กไฟฟ้าอาจจะเปลี่ยนเป็นสนามแม่เหล็กซึ่งขึ้นอยู่กับระยะทางของเสาอากาศของตัวอ่านกับตัวแท็ก ซึ่งกล่าวภายหลัง

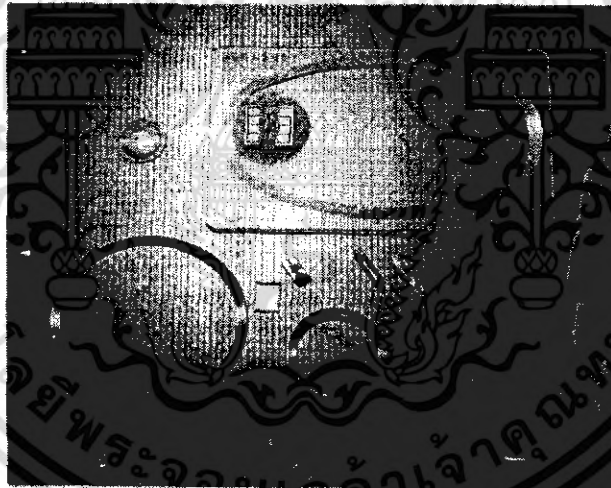
ส่วนเล็กของการส่งสนามแพร่ไปทั่วเสาอากาศของตัวแท็ก ซึ่งคือระยะทางจากขดลวดของตัวอ่าน โวลต์(Ui)ถูกสร้างขึ้นที่เสาอากาศของตัวแท็กโดยการเหนี่ยวนำ โวลต์นี้จะถูกแปลงและใช้เป็นพลังงานสำหรับไมโครชิพ ตัวเก็บประจุ Cr ถูกต่อขนานกับเสาของ reader การเก็บประจุของตัวเก็บประจุนี้จะถูกเลือกซึ่งมันจะทำงานกับขดลวดเหนี่ยวนำของเสาอากาศจากวงจร parallel resonant ด้วยความถี่ resonant ซึ่งตรงกับความถี่ของ reader กระแสที่สูงมากถูกสร้างขึ้นในเสาขดลวดของตัวเครื่องอ่านสัญญาณ โดยการเข้ากันของวงจรparallel resonant ซึ่งจะใช้สร้างสนามแม่เหล็กที่ต้องการในการทำงานของแท็ก

เสาอากาศของตัวแท็ก และ CI จากวงจร Resonant จะส่งความถี่ไปยังตัวเครื่องอ่านสัญญาณ โวลเตจ U ที่ขดลวดในตัวแท็กจะมากที่สุด ก็ขึ้นอยู่กับ ขั้นตอน Resonant ในวงจร Parallel Resonant



รูปที่ 2.13 พลังงานการเหนี่ยวนำที่เกิดจากการคล้องสัญญาณของแท็กส่งจากพลังงานของสนามแม่เหล็กที่เปลี่ยนแปลงและสร้างโดยตัวเครื่องอ่านสัญญาณ

รูปของขดลวด 2 ขด คล้ายกับหม้อแปลง ซึ่งกรณีนี้เป็นการคล้องสัญญาณซึ่งได้สัญญาณที่อ่อนมาก ประสิทธิภาพของพลังงานในการส่งระหว่างเสาอากาศของตัวส่งและตัวรับเป็นสัดส่วนของความถี่ตัวเลขของจำนวนรอบการพันของขดลวด  $n$ , พื้นที่  $A$  ล้อมรอบด้วยขดลวดของ Tag มุมของ 2 ขดลวด ความแตกต่างของแต่ละตัวและระยะห่างระหว่าง 2 ขดลวด



รูปที่ 2.14 การออกแบบแท็ก

ความถี่เพิ่มขึ้นนั้นต้องการขดลวดเหนี่ยวนำของขดลวดของตัว Tag ด้วยเหตุนี้ตัวเลขรอบของขดลวดลดลง (135KHz: typical 100-1000 windings, 13.56 MHz: typical 3-10 windings) เพราะโวลต์ที่เกิดขึ้นใน tag นั้นเป็นสัดส่วนของความถี่การลวดขดลวดนั้นส่งผลกับประสิทธิภาพของพลังงานในการส่งความถี่สูงขึ้น

อย่างไรก็ตามมันยังคงมีประสิทธิภาพน้อยมาก การเหนี่ยวนำระบบนั้นเพียงทำให้มันเหมาะสมกันสำหรับวงจรที่กระแสดำ อ่านเพียงแท็ก (<135KHz) อย่างเดียว ด้วยพลังงานที่ใช้ต่ำที่สุดที่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

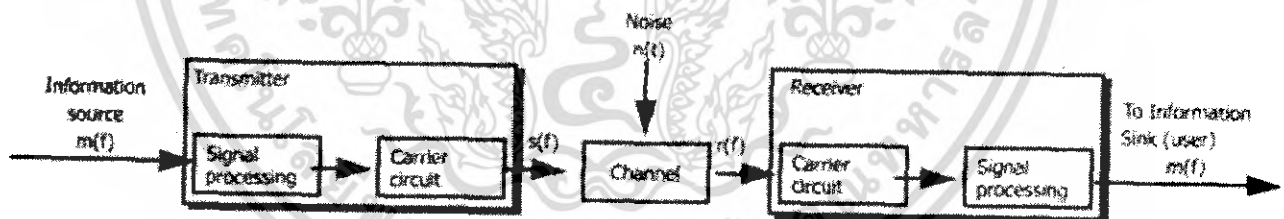
ใช้ได้สำหรับช่วง > 1m เนื่องจากใช้พลังงานสูงแท่งจะเขียนฟังก์ชันและระบบเตือนภัยจะทำงานที่ระยะ 25 cm แม้ว่ามีบ้างที่ระยะเกิน 80 cm

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

\* Close coupling system

ตารางที่ 2.2 แสดงการใช้พลังงานของหลายๆ RFID-ASIC พลังงานโวลเตจที่น้อยที่สุดที่ไม่โครชิพสามารถทำงานได้คือ 1.8V และมากที่สุดที่ 10V

### 2.1.8 เทคโนโลยีการเข้ารหัสและถอดรหัส



รูปที่ 2.15 การไหลของสัญญาณและข้อมูลในระบบการสื่อสารแบบดิจิทัล

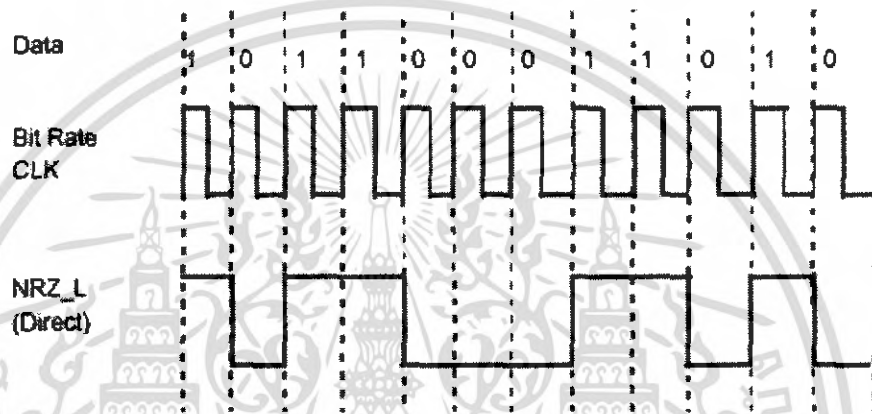
จากรูปแสดงกระบวนการสื่อสารแบบดิจิทัล ซึ่งคือการส่งข้อมูลระหว่างเครื่องอ่านสัญญาณกับแท่งในระบบ RFID นั่นเอง กระบวนการสื่อสารจะอาศัยหลักการดังนี้

#### 2.1.8.1 การเข้ารหัสข้อมูล (Signal Coding)

การรับส่งข้อมูลแบบตรงไปตรงมาจะทำให้ข้อมูลอาจถูกรบกวนและทำให้การชิงโครโนซ์ของข้อมูลเกิดการคลาดเคลื่อน (โดยปกติวงจรดิจิทัลจะปรับการชิงโครโนซ์ของข้อมูลได้เฉพาะในช่วงที่มีการเปลี่ยนระดับของข้อมูลจาก 1 เป็น 0 หรือจาก 0 เป็น 1) ทำให้รับข้อมูลผิดพลาด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อป้องกันปัญหาดังกล่าวจึงจะต้องมีการนำสัญญาณดิจิทัลปกติไปผ่านเข้ารหัสเสียก่อน ก่อนที่รหัสจะถูกส่งไปมอดูเลต แต่การเข้ารหัสบางอย่างก็มีข้อเสียคือช่วงเวลาที่ใช้ในการส่งข้อมูลอาจต้องเพิ่มขึ้นเป็น 2 เท่า การเข้ารหัสมักเป็นที่นิยมใช้กันอย่างกว้างขวางในการสื่อสารแบบดิจิทัล ในระบบสื่อสารแบบดิจิทัลนั้นจะใช้สัญลักษณ์ 0 กับ 1 แทนข้อมูลโดยความแตกต่างของข้อมูลจะเป็นตัวกำหนดค่า 0 กับ 1 ของแต่ละบิตข้อมูล ข้อมูลจะถูกจัดเรียงเป็น แนวนอนหรือเส้นตรง (Line Code) ซึ่งมาตรฐานของ Line Code จะมีหลายมาตรฐาน และมาตรฐานที่นำมาใช้ในระบบ RFID มีดังนี้คือ

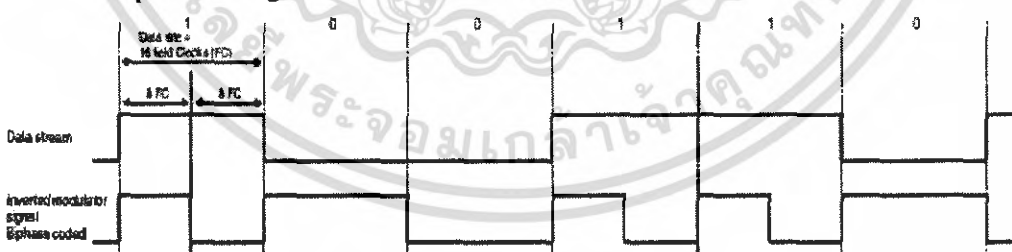
1) NRZ coding



รูปที่ 2.16 รหัสสัญญาณ NRZ

สัญลักษณ์ “1” จะแทนด้วยช่วงสัญญาณที่เป็น High, สัญลักษณ์ “0” จะแทนด้วยช่วงสัญญาณที่เป็น Low

2) Biphas coding



รูปที่ 2.17 รหัสสัญญาณ Biphas

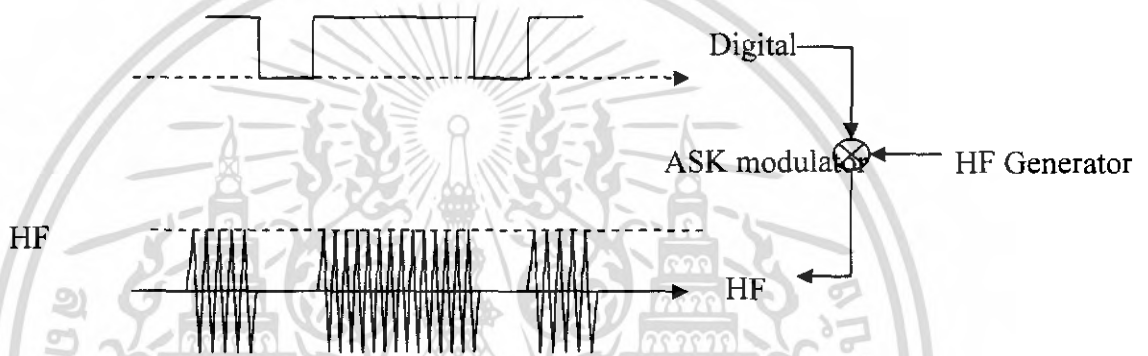
ข้อกำหนดของลักษณะการเข้ารหัส code แต่ละแบบนี้อาจจะแตกต่างกันออกไปจากที่ได้กล่าวมาข้างต้น ตามชนิดของ แทกต์ ของแต่ละบริษัทผู้ผลิตแทกต์

### 2.1.8.2 การผสมรหัสข้อมูลเข้ากับคลื่นพาหะ (Modulation)

เนื่องจากหลักการพื้นฐานของ RFID คือ ใช้คลื่นสนามแม่เหล็กไฟฟ้าในการสื่อสารรับส่งข้อมูล ดังนั้นก่อนการสื่อสารจะต้องมีกระบวนการผสมข้อมูลที่เข้ารหัสไว้แล้วไปกับคลื่นพาหะ (Modulation) แต่เนื่องจากลักษณะข้อมูลเป็นแบบดิจิทัลจึงจำเป็นต้องใช้วิธีการผสมแบบดิจิทัลซึ่งจะแตกต่างจากการผสมแบบอะนาล็อกที่ใช้กันอยู่หลายระบบในปัจจุบันเช่น FM และ AM การผสมข้อมูลแบบดิจิทัลมี 3 วิธีดังนี้

#### 1) Amplitude Shift Keying (ASK)

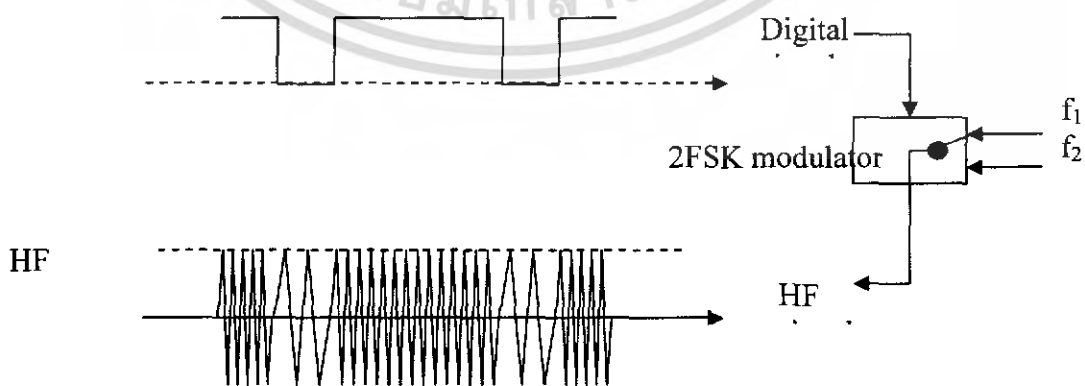
เป็นการผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงของยอดคลื่นพาหะ (Amplitude) เป็นตัวแสดงลักษณะข้อมูลโดยความถี่ของคลื่นพาหะไม่เปลี่ยนแปลงดังรูป



รูปที่ 2.18 ใน ASK ความสูงของยอดคลื่นจะเปลี่ยนสถานะอยู่ 2 สถานะขึ้นอยู่กับข้อมูลที่ผสมเข้ามา

#### 2) Frequency Shift Keying (FSK)

วิธีนี้จะอาศัยการเปลี่ยนแปลงความถี่ของคลื่นพาหะระหว่าง 2 ความถี่ขึ้นอยู่กับสถานะของข้อมูล 0 กับ 1 โดยความสูงของยอดคลื่น (Amplitude) ไม่เปลี่ยนแปลง

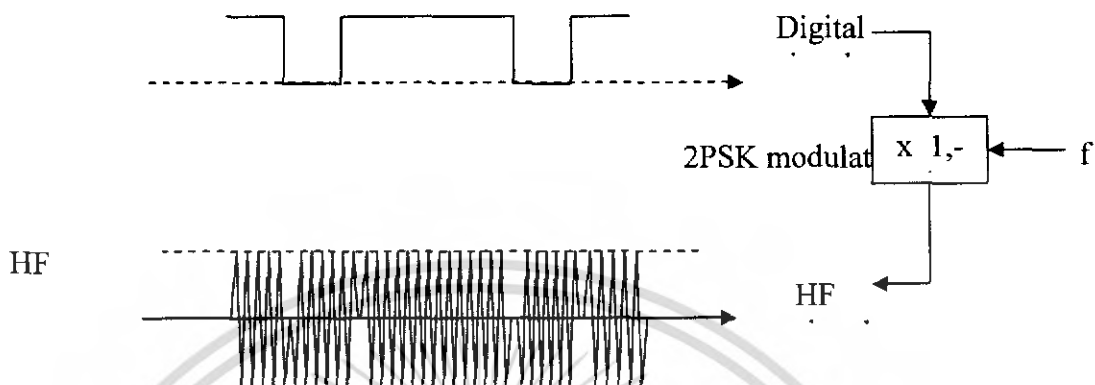


รูปที่ 2.19 2FSK จะเปลี่ยนแปลงความถี่ของคลื่นตามข้อมูลที่ผสมเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3) Phase Shift Keying ( 2PSK )

วิธีนี้จะใช้หลักการเปลี่ยนเฟสของลูกคลื่นเป็นตรงกันข้าม (0 องศา กับ 180 องศา )  
เมื่อมีการเปลี่ยนสถานะของข้อมูล ดังแสดงในรูป



รูปที่ 2.20 PSK จะกลับเฟสทุกครั้งเมื่อมีการเปลี่ยนสถานะของข้อมูล

แต่ในระบบ RFID จะใช้เพียงการ modulate สัญญาณที่เข้ารหัสมาแล้วกับสัญญาณคลื่นพาห้แบบ ASK เท่านั้น ในการรับส่งข้อมูลระหว่างเครื่องอ่านสัญญาณกับแท็กส์

#### 2.1.8.3 การส่งคลื่นสัญญาณออกไป (Transmission)

เครื่องอ่านจะเป็นตัวส่งคลื่นสนามแม่เหล็กไฟฟ้าออกไปด้วยความถี่ที่เราจะใช้งาน เมื่อแท็กส์ อยู่ในระยะที่เหมาะสมจะสามารถเหนี่ยวนำคลื่นสนามแม่เหล็กไฟฟ้านั้นมาแปลงเป็นพลังงานที่ใช้ในการทำงานภายในได้และแท็กส์จะนำสัญญาณข้อมูล ดิจิตอลที่เข้ารหัสแล้วมาทำการ มอดูเลตกับคลื่นสนามแม่เหล็กไฟฟ้าที่รับได้แบบ ASK ส่งกลับไปยังเครื่องอ่านซึ่งเรียกสัญญาณนี้ว่า “ สัญญาณ backscattering ”

#### 2.1.8.4 การถอดข้อมูลออกจากคลื่นพาห้ (Demodulation)

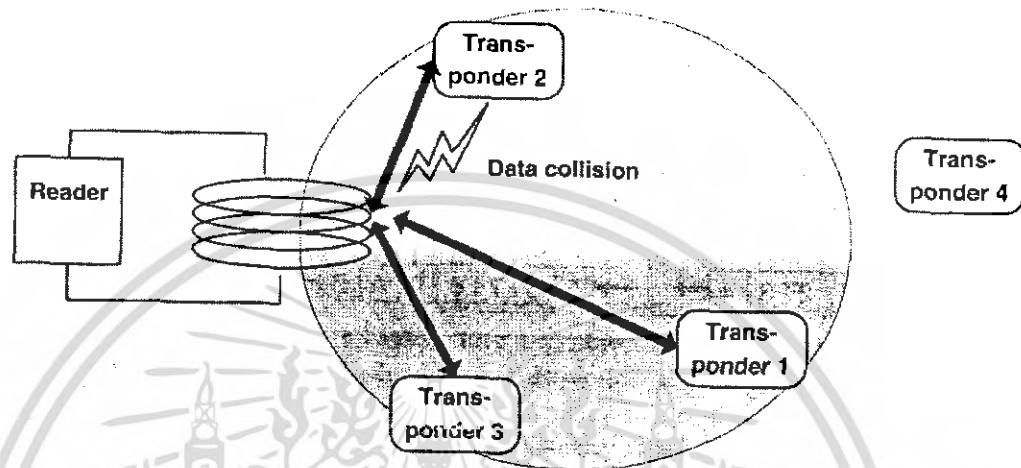
สัญญาณที่ทางเครื่องอ่านรับได้นั้นจะเป็นสัญญาณข้อมูลที่ถูกเข้ารหัสแล้วมอดูเลตกับคลื่นพาห้แบบ ASK เพราะฉะนั้นเราสามารถนำสัญญาณเดิมกลับมาได้โดยการดีมอดูเลตด้วยวิธีการตรวจจับกรอบสัญญาณ สัญญาณที่ได้จากการตรวจจับกรอบสัญญาณจะเป็นสัญญาณข้อมูลที่ถูกเข้ารหัสโดยแท็กส์

### 2.1.9 การป้องกันการเกิดการชนกันของข้อมูล (ANTICOLLISION)

การ collision เกิดขึ้นได้อย่างไร ขณะที่ระบบ RFID ทำงานนั้นเราไม่สามารถตัดโอกาส

ในการที่จะใช้แท็กส์จำนวนหลายตัวโดยอยู่ในขอบเขตการทำงานของเครื่องอ่านสัญญาณ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพียงตัวเดียวได้ เมื่อเครื่องอ่านสัญญาณส่งคำสั่งออกไปมันจะถูกดำเนินงานโดยtransponder ทั้งหมดที่อยู่ในขอบเขตการทำงานของเครื่องอ่านสัญญาณมันคือสาเหตุที่ทำให้สันนิษฐานได้ว่า แทกส์ทั้งหมดจะพยายามส่งคำตอบสำหรับคำสั่งที่ส่งมากลับไปยังเครื่องอ่านสัญญาณพร้อมกัน กรณีนี้ส่วนใหญ่จะเกิดจากการที่แทกส์หลายๆตัวส่งข้อมูลออกมาพร้อมกันซึ่งจะทำให้เกิดการรบกวนซึ่งกันและกันและทำให้เกิดการสูญหายของข้อมูล



รูปที่ 2.21 แทกส์หลายๆตัวในบริเวณทำงานของเครื่องอ่านสัญญาณตัวเดียวเป็นสาเหตุของการชนกันของข้อมูล

### 2.1.9.1 วิธีป้องกันข้อมูลชนกันแบบใช้การอ้างอิงความถี่(frequency domain anticollision)

ในการทำงานวิธีนี้คือ การรวมความถี่(frequency multiplex) หรือขยายสเปกตรัมไว้สำหรับส่งข้อมูลจากแทกส์ไปเครื่องอ่านสัญญาณ การป้องกันข้อมูลชนกันวิธีนี้ค่อนข้างยากเนื่องจากราคาสูง

ผู้สร้างแค่เพียงรู้ระบบที่ใช้สำหรับการ ID ของ tag ที่ติดกับสัมภาระในสนามบิน transponder จะบรรจุตัวเลขดิจิทัลจำนวนมากเช่นเดียวกับบาร์โค้ดบนป้ายติดกระเป๋าเดินทาง ในวิธีปฏิบัติการกระเป๋าจะถูกส่งผ่านอุโมงค์ตัวอ่านโดยสายพาน โดยกระเป๋าหลายๆใบอาจอยู่ในบริเวณการอ่านของ reader ในเวลาเดียวกัน แหล่งจ่ายไฟให้กับแทกส์มาจากการเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้าที่ความถี่ 125kHz และการส่งข้อมูลจากแทกส์ไปยัง reader แทนที่ความถี่ค่าใดค่าหนึ่งในช่วง 2.7-4.2MHz เครื่องอ่านสัญญาณจะแพร่กระแสของข้อมูลแบบต่อเนื่อง ในกรณีความถี่ที่แพร่อยู่ถูกรบกวนโดยแทกส์ลดลง แทกส์ที่เข้าไปในบริเวณทำงานของ reader จะถอดรหัสกระแสของข้อมูลนี้ และส่งตัวเลข ID ของมันไปที่เครื่องอ่านสัญญาณบนความถี่ที่ยังไม่ถูกรบกวน

### 2.1.9.2 วิธีป้องกันข้อมูลชนกันแบบใช้การอ้างอิงเวลา(time domain anticollision)

วิธีป้องกันข้อมูลชนกันแบบใช้การอ้างอิงเวลาเป็นวิธีการป้องกันข้อมูลชนกันที่มีกลุ่มใหญ่ที่สุดโดยใช้ช่องว่างบางส่วน กลุ่มนี้เป็นวิธีการแบ่งออกเป็นกลุ่มย่อยๆ ซึ่งจะถูกรวบรวมโดยเครื่องอ่านสัญญาณ(interrogator driven) และแท็กส์(transponder driven)

กระบวนการขับเคลื่อนของแท็กส์จะทำการจัดการอย่างไม่ป้องกันในเรื่องของเวลา เนื่องจากไม่ได้ถูกรวบรวมโดยตัวอ่าน ตัวอย่างหนึ่งในกระบวนการที่ กลุ่มของแท็กส์ส่ง ID ไปที่ตัวอ่านโดยวนเป็นรอบๆ เวลาที่ใช้ส่งข้อมูลจะเป็นส่วนย่อยๆ ของเวลาที่ซ้ำๆ กัน ซึ่งมีการหยุดรอสั้นๆ อย่างเป็นสัดส่วนของการส่งสัญญาณ ยังมีส่วนเพิ่มเติมที่ว่า เวลาที่ซ้ำๆ นี้มีความแตกต่างน้อยมากสำหรับแท็กส์แต่ละอันมันจึงเหมือนกับว่าแท็กส์ 2 ตัว จะส่ง ID ไปกันคนละเวลาและข้อมูลก็จะไม่ชนกัน ความแตกต่างนี้ถูกควบคุมโดยกระบวนการ "switched off" และ "non switched", ซึ่งขึ้นอยู่กับสถานะของแท็กส์ที่ switched off โดยตัวอ่านหลังจากที่ส่งข้อมูลกันเสร็จแล้ว

วิธีtransponder driven โดยปรกติแล้วจะซ้ำและไม่ยึดหยุ่นดังนั้นการประยุกต์ใช้งานส่วนมากจะใช้วิธีที่ถูกควบคุมโดยเครื่องอ่านสัญญาณ(interrogator driven) วิธีนี้สามารถพิจารณาได้ว่าเป็นการ synchronous เพราะแท็กส์ทุกตัวจะถูกขับและควบคุมโดยเครื่องอ่านสัญญาณในเวลาเดียวกัน วิธีinterrogator driven นั้นถูกแบ่งออกเป็นกลุ่มย่อยๆ แบ่งเป็นวิธีการ polling และวิธี binary search ทั้งสองวิธีอ้างอิงขึ้นอยู่กับ ID ของแท็กส์ที่ใช้หมายเลขชุดแบบพิเศษ

วิธีการpolling ต้องการรายการทั้งหมดของตัวเลขลำดับของแท็กส์ที่สามารถถูกใช้ ตัวเลขลำดับทั้งหมดจะถูกร้องขอโดยเครื่องอ่านสัญญาณจนกว่าแท็กส์พร้อมด้วย ID ตัวเลขลำดับจะมีผลตอบรับ ถึงแม้ว่าวิธีการนี้อาจจะซ้ำและขึ้นอยู่กับจำนวนของแท็กส์และนั่นอาจเหมาะสมกับเฉพาะการประยุกต์ใช้งานที่มีแท็กส์ที่รู้จักเพียงเล็กน้อยอยู่ในบริเวณสนามแม่เหล็ก

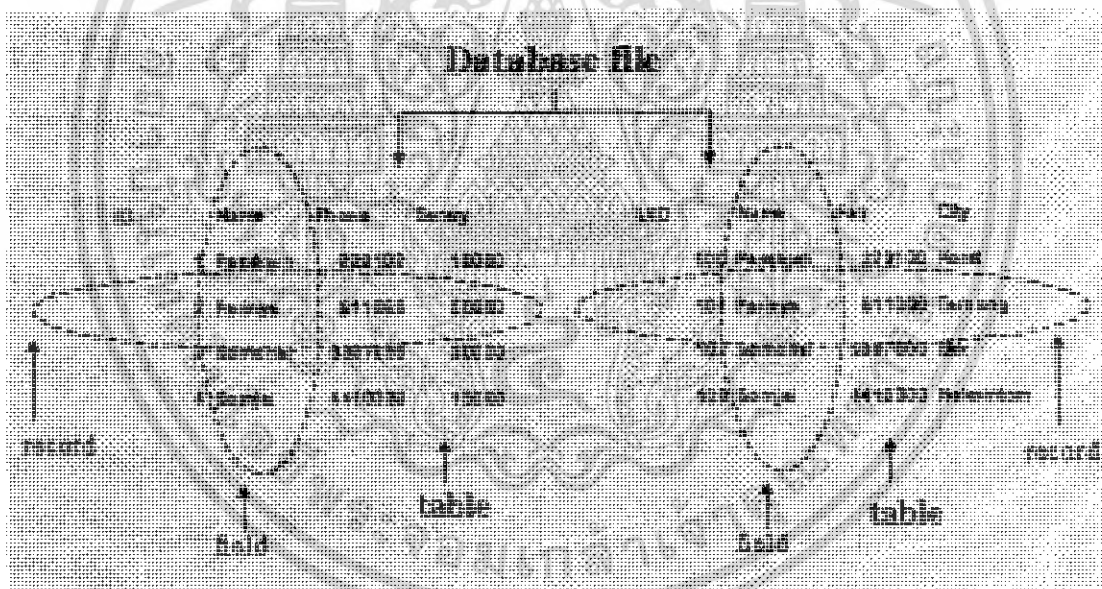
กระบวนการค้นหาแบบ binary search เป็นกระบวนการที่ยืดหยุ่นที่สุด และยังเป็นกระบวนการที่รู้จักกันดี ในขั้นตอนการค้นหาแบบ binary search หนึ่งๆ แท็กส์ตัวหนึ่งจะถูกเลือกออกมาจากกลุ่มโดยตั้งใจทำให้เกิดการชนกันของข้อมูลเกี่ยวกับ transponder serial numbers ที่ส่งไปที่ตัวอ่าน ตามความต้องการของคำสั่งจากตัวอ่าน ถ้ากระบวนการนี้สำเร็จ มันจะเป็นสิ่งสำคัญยิ่งที่ตัวอ่านจะมีความสามารถที่จะพิจารณาดำแหน่งที่แน่นอนของบิทที่การชนกันที่ใช้ระบบเข้ารหัสสัญญาณที่เหมาะสม นอกจากนี้ระบบยังต้องสามารถเลือกได้ก่อนว่ากลุ่มของที่อยู่ของแท็กส์เมื่อเวลาที่คำสั่งร้องขอข้อมูลถูกส่งออกไป คำสั่งนี้จะถูกส่งออกไปเรื่อยๆ โดยที่แต่ละครั้งมันจะทำการลดกลุ่มของจำนวนที่ถูกเลือก จนกระทั่งเหลือแท็กส์เพียงตัวเดียวที่ตอบกลับมา เพื่อให้กระบวนการนี้กระจ่าย กระบวนการ binary search ถูกนำมาอธิบายรายละเอียดในส่วนของการติดตามตัวอย่างของ application

## 2.2 ฐานข้อมูล MySQL

MySQL เป็นโปรแกรมฐานข้อมูลที่ใช้จัดเก็บข้อมูล โปรแกรมหนึ่ง ทำงานในลักษณะ Client Server ทำงานบนระบบ Telnet บน Linux Redhad หรือ Unix System (ฟรี) และบน Win32 (เสียค่าใช้จ่าย) ทั่วไปบนระบบเครือข่าย Inter&Intranet นั้นหมายความว่าเราสามารถเรียกใช้ MySQL ได้ทั่วโลกกรณีเป็น Internet และ ทับบริเวณที่เป็น Intranet และยังสามารถเรียกใช้บน Web Browser ได้กรณีใช้ language เป็น Interface ในการเชื่อม language ที่ใช้เป็น Interface เช่น PHP, Perl, C, C++ ฯ

MySQL เป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) RDBMS คือสามารถทำงานกับตารางข้อมูลหลายตารางพร้อมๆ กัน โดยสามารถแสดงความสัมพันธ์ของตารางเหล่านั้นด้วย field ที่ใช้ร่วมกัน

ไม่ว่าจะจัดเก็บโดยผู้บรรยายหรือจัดเก็บโดยคอมพิวเตอร์ จำเป็นจะต้องมีรูปแบบในการจัดเก็บที่เหมาะสม ซึ่งรูปแบบการจัดเก็บทั้งสองอย่างรวมทั้ง MySQL ด้วยไม่แตกต่างกันนัก โดยเราสามารถแบ่งได้ตามภาพข้างล่างดังนี้



รูปที่ 2.22 รูปแบบในการจัดเก็บฐานข้อมูล

- โดยรวมข้อมูลทั้งหมดที่จัดเก็บเข้าพวกเดียวกันเราเรียกว่า Database file (ไฟล์ฐานข้อมูล)
- จากนั้น ภายใน Database file ก็จะประกอบไปด้วย Table (ตาราง) ต่างๆ ที่จัดเก็บข้อมูล
- ภายใน Table ก็จะประกอบไปด้วย field (คอลัมน์) ซึ่งจะบ่งบอกลักษณะของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากนั้นก็เป็น Record (แถว) ซึ่งเป็นส่วนของข้อมูลที่ถูกจัดเข้าจำพวกด้วยกันตามลักษณะการแบ่งพวกของ field

## PHP กับฐานข้อมูล MySQL

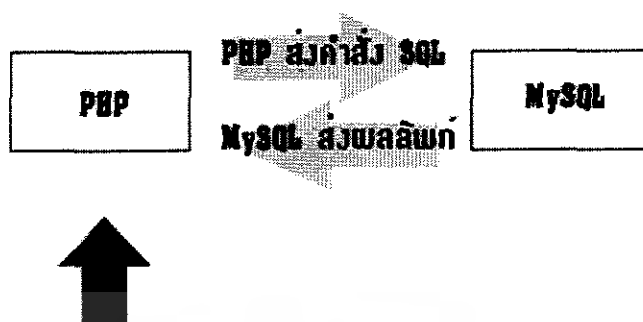
PHP เป็นภาษาจำพวก scripting language คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า สคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริปต์ก็เช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมาเพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่งซึ่งช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

เนื่องจากว่า PHP ไม่ได้เป็นส่วนหนึ่งของตัว Web Server ดังนั้นถ้าจะใช้ PHP ก็จะต้องดูก่อนว่า Web server นั้นสามารถใช้สคริปต์ PHP ได้หรือไม่ ยกตัวอย่างเช่น PHP สามารถใช้ได้กับ Apache WebServer และ Personal Web Server (PWP) สำหรับระบบปฏิบัติการ Windows 95/98/NT ในกรณีของ Apache เราสามารถใช้ PHP ได้สองรูปแบบคือ ในลักษณะของ CGI และ Apache Module ความแตกต่างอยู่ตรงที่ว่า ถ้าใช้ PHP เป็นแบบโมดูล PHP จะเป็นส่วนหนึ่งของ Apache หรือเป็นส่วนขยายในการทำงานนั่นเอง ซึ่งจะทำงานได้เร็วกว่าแบบที่เป็น CGI เพราะว่า ถ้าเป็น CGI แล้ว ตัวแปลชุดคำสั่งของ PHP ถือว่าเป็นแค่โปรแกรมภายนอก ซึ่ง Apache จะต้องเรียกขึ้นมาทำงานทุกครั้ง ที่ต้องการใช้ PHP ดังนั้น ถ้ามองในเรื่องของประสิทธิภาพในการทำงาน การใช้ PHP แบบที่เป็นโมดูลหนึ่งของ Apache จะทำงานได้มีประสิทธิภาพมากกว่า

### ลักษณะเด่นของ PHP

- ใช้งานได้ฟรี
- PHP เป็น โปรแกรมวิ่งข้าง Sever ดังนั้นขีดความสามารถไม่จำกัด
- PHP วิ่งบนเครื่อง UNIX, Linux, Windows ได้หมด
- เรียนรู้ง่าย เนื่องจาก PHP ผังเข้าไปใน HTML และใช้โครงสร้างและไวยากรณ์ภาษาต่างๆ
- เร็วและมีประสิทธิภาพ โดยเฉพาะเมื่อใช้กับ Apache Xerve เพราะไม่ต้องใช้โปรแกรมภายนอก
- ใช้ร่วมกับ XML ได้ทันที
- ใช้กับระบบเพิ่มข้อมูลได้
- ใช้กับข้อมูลตัวอักษรได้อย่างมีประสิทธิภาพ
- ใช้กับโครงสร้างข้อมูลใช้ได้แบบ Scalar, Array, Associative array
- ใช้กับการประมวลผลภาพได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### คำสั่ง SQL ผ่านคำสั่ง PHP

รูปที่ 2.23 แสดงการเชื่อมต่อระหว่างภาษา PHP กับ MySQL

ปกติแล้ว จะทำงานกับ PHP ก็ต้องใช้คำสั่ง PHP แล้วพอจะทำงานกับ MySQL ก็ต้องใช้คำสั่ง SQL เหมือนอย่างในภาพ พอเราจะใช้ PHP ทำงานกับ MySQL ให้เราดูที่จุดหลัก คือ PHP ส่วน MySQL เป็นตัวเสริม การทำงานจึงออกมาในรูปแบบ ใช้ Function ใน PHP เพื่อส่งคำสั่ง SQL ไปยัง MySQL เพื่อทำงานอีกทอดหนึ่ง เมื่อได้ผลลัพธ์ทุกอย่างก็ค่อยส่งผลลัพธ์นั้น ๆ กลับมายัง PHP แล้วที่เหลือก็อยู่ที่เราว่า การนำผลลัพธ์นั้นมาทำอะไร ลองดูตัวอย่างจากในภาพ

## 2.3 Delphi 7

Delphi 7 คือซอฟต์แวร์ที่เรานำมาใช้ในการเขียนโปรแกรมเพื่อสร้างแอปพลิเคชัน หรือซอฟต์แวร์อีกที โดยมันจะประกอบด้วยเครื่องมือชนิดต่างๆ ที่ใช้ให้การเขียนโปรแกรมทำได้อย่างสะดวก

Delphi 7 จัดเป็นเครื่องมือเขียนโปรแกรมชนิด Visual Programming เช่นเดียวกับ Visual Basic หรือ Visual C++ โดยมีข้อดีคือ สามารถเขียนโปรแกรมได้ง่ายและให้ผลงานออกมาอย่างรวดเร็ว ซึ่งจะแตกต่างจากเครื่องมือเขียนโปรแกรม ดังนั้นจึงจัดให้ Delphi 7 เป็นซอฟต์แวร์ประเภท RAD หรือ Rapid Application Development ซึ่งสามารถสร้างแอปพลิเคชันได้อย่างรวดเร็ว

### 2.3.1 ข้อเด่นของ Delphi 7

โปรแกรมที่ได้จากการเขียนโปรแกรมมีขนาดเล็กสามารถสร้างแอปพลิเคชันที่ทำงานบนวินโดวส์ได้ดีเหมือนเดิม โดยมีการปรับปรุงให้สามารถสร้างแอปพลิเคชันตามแนวความคิดของ .NET ซึ่งจะช่วยทำให้สามารถเขียนโปรแกรมครั้งเดียว และนำไปใช้งานบนอุปกรณ์ต่างๆ ไม่ว่าจะเป็น PDA, โทรศัพท์มือถือ และบนเว็บได้

ขณะเดียวกัน Delphi 7 ก็ได้รับการพัฒนาให้สามารถพัฒนาแอปพลิเคชันแบบข้ามแพลตฟอร์มได้นั้นคือสามารถพัฒนาแอปพลิเคชันที่ทำงานได้ทั้งบน วินโดวส์และลินุกซ์นั่นเอง

### 2.3.2 องค์ประกอบของ Delphi 7

หลังจากการแต่งตั้งแล้วเสร็จ เราสามารถเรียกใช้ Delphi 7 ได้โดยคลิกที่ strat > All Program > Borland > Delphi 7 > Delphi 7 จากนั้นก็จะเข้าสู่การทำงานของ Delphi 7

องค์ประกอบที่สำคัญต่างๆ สามารถอธิบายได้ต่อไปนี้

- **เมนูบาร์** เป็นส่วนที่เก็บคำสั่งเพื่อสั่งงาน Delphi 7 โดยมีลักษณะเป็นเมนูให้เลือกใช้งานเหมือนกับโปรแกรมอื่นๆทั่วไป
- **Form Designer** เป็นส่วนที่ใช้ในการออกแบบโปรแกรมที่สร้างขึ้นมาด้วย Delphi 7 ที่เราเรียกว่าฟอร์ม ซึ่งมีลักษณะเหมือนกับวินโดวส์ทั่วไป
- **Component Palette** เป็นส่วนที่เก็บคอมโพเนนต์ ซึ่งจะเป็นส่วนสำคัญที่จะเป็นองค์ประกอบภายในฟอร์ม เช่น ปุ่มคลิก, ปุ่มเลือก, รายการตัวเลือก เป็นต้น
- **Object Tree View** เป็นส่วนที่ใช้อ้างอิงถึงออบเจกต์ และคอมโพเนนต์ต่างๆ ที่นำมาใช้ในระหว่างการพัฒนาแอปพลิเคชัน
- **Object Inspector** เป็นส่วนที่กำหนดค่าให้กับฟอร์ม หรือคอมโพเนนต์ที่อยู่บนฟอร์มค่าที่กำหนดนี้จะทำให้หน้าตา หรือความสามารถของฟอร์มคอมโพเนนต์เปลี่ยนไป เช่น เปลี่ยนสีพื้น, เปลี่ยนข้อความบนปุ่ม, ปรับเปลี่ยนขนาดกว้างยาวของฟอร์ม เป็นต้น

นอกจากนี้ยังมีองค์ประกอบอื่นอีก ซึ่งจะพบเมื่อเราเริ่มเขียนโปรแกรมกับ Delphi 7 ดังนี้

- **Code Editor** เป็นส่วนที่เราจะใช้เขียนคำสั่งเพื่อให้โปรแกรมหรือแอปพลิเคชันที่สร้างขึ้นมานั้นทำงานได้ตามที่เราตั้งใจไว้ ซึ่งเราจะเขียนคำสั่งต่างๆด้วยภาษา Object Pascal

เราเรียกองค์ประกอบต่างๆที่กำลังจะใช้งานเพื่อเขียนโปรแกรมนี้ว่า IDE (Integrate Development Environment)

### 2.3.3 รายละเอียดของคอมโพเนนต์ที่สำคัญ

#### 2.3.3.1 คอมโพเนนต์ต่างๆ ในเพจ MySQL Access

คอมโพเนนต์ในเพจนี้เป็นคอมโพเนนต์ที่สามารถทำให้เราติดต่อกับข้อมูลภายในฐานข้อมูล MySQL ผ่านทาง MySQL Access ซึ่งแสดงได้ดังรูป



รูปที่ 2.24 คอมโพเนนต์ในเพจ MySQL Access

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คอมโพเนนต์ MyConnection

คอมโพเนนต์ Myconnection เป็นออบเจกต์ Connection ทำหน้าที่ในการติดต่อกับแหล่งข้อมูลรวมทั้งควบคุมเงื่อนไข และสถานการณ์ติดต่อกับฐานข้อมูลนั้น

### พรีพเพอร์ติวี่ที่สำคัญ

- **Connection** กำหนดเชื่อมต่อระหว่างคอมโพเนนต์ MyConnection กับแหล่งข้อมูล ถ้ากำหนดเป็น True แสดงว่ามีการเชื่อมต่อกับแหล่งข้อมูลอยู่ แต่ถ้าเป็น False จะไม่มีการเชื่อมต่อเกิดขึ้น
- **DataBase** กำหนดตารางของฐานข้อมูลที่ต้องการติดต่อ
- **LoginPrompt** ใช้กำหนดว่าจะให้แสดงไดอะล็อกซ์ เพื่อรับข้อมูลชื่อผู้ใช้งาน และรหัสผ่านในการเชื่อมต่อกับแหล่งข้อมูลหรือไม่ถ้าต้องการให้มีค่าที่กำหนดคือ True และในทางตรงกันข้ามจะกำหนดเป็น False

### เมธอดที่สำคัญ

- **Open** ใช้เพื่อเปิดการเชื่อมต่อกับคอมโพเนนต์ MyConnection กับแหล่งข้อมูล โดยเมื่อ เรียกใช้เมธอดนี้ เท่ากับว่าเรากำหนดพรีพเพอร์ติวี่ Connected ให้เป็น True
- **Close** ใช้เพื่อปิดการเชื่อมต่อกับคอมโพเนนต์ MyConnection กับแหล่งข้อมูล
- **GetFieldname** ใช้ดึงรายชื่อฟิลด์ทั้งหมดในตารางที่กำหนด
- **GetTableNmaes** ใช้ดึงรายชื่อในตารางทั้งหมดที่มีในแหล่งข้อมูลที่มีการติดต่อ

## คอมโพเนนต์ MyTable

ใช้ในการเข้าถึงข้อมูลในตาราง และดึงกลุ่มข้อมูลจากตารางในฐานข้อมูล MySQL โดยสามารถอ้างถึงตารางได้ 1 ตาราง

### พรีพเพอร์ติวี่ที่สำคัญ

- **ACTIVE** ถ้าเป็น True เราจะสามารถใช้งานข้อมูลในตารางได้ ในทางกลับกัน ถ้าค่า เป็น False ตารางเราจะไม่สามารถใช้งานข้อมูลในตารางนั้นได้
- **BOF** ถ้ามีค่าเป็น True แสดงว่าตัวชี้เร็คคอร์ดอยู่ที่ตำแหน่งแรกของตาราง มาจากคำว่า Begin of File
- **DataSource** กำหนดแหล่งข้อมูลของตารางที่เราต้องการใช้ในฐานข้อมูล
- **EOF** ถ้ามีค่าเป็น True แสดงว่าตัวชี้เร็คคอร์ดอยู่ที่ตำแหน่งสุดท้ายของตาราง มาจากคำว่า End of File
- **FileCount** นับจำนวนฟิลด์ที่มีอยู่ในตาราง
- **FileValue** กำหนดค่าให้กับฟิลด์ของตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Filter** กำหนดค่าเพื่อใช้ในการกรองข้อมูล
  - **Filtered** ถ้าค่าของพรีอเพอร์ติ์นี้เป็น True ข้อมูลจะถูกกรองตามเงื่อนไขที่กำหนดในพรีอเพอร์ติ์Filter แต่ถ้าเป็น False จะไม่มีการกรองข้อมูล
  - **TableName** กำหนดชื่อตารางที่ต้องการนำมาใช้จากแหล่งข้อมูลที่กำหนด
- เมธอดและอีเวนต์ที่สำคัญ**
- **เมธอด FieldByName** เป็นเมธอดที่ใช้ในการดึงรายละเอียดของฟิลด์ โดยใช้ชื่อฟิลด์ในการอ้างอิงข้อมูลนั้น
  - **เมธอด FindKey** ใช้ในการค้นหาข้อมูล จากตารางที่มีการสร้างอินเด็กซ์ไว้ โดยการกำหนดค่าที่ต้องการค้นหา
  - **เมธอด FindNearest** เมธอดนี้คล้ายกับเมธอด FindKey แต่ถ้าค้นหาข้อมูลไม่พบตัวชี้จะชี้ไปยังเรคคอร์ดแรกของข้อมูลที่มีค่ามากกว่าค่าที่กำหนดไว้
  - **เมธอด Locate** เป็นเมธอดอีกตัวหนึ่งที่ใช้ในการค้นหาข้อมูลที่ต้องการ แต่ไม่จำเป็นที่ตารางนั้นจะต้องมีอินเด็กซ์ก็ได้ โดยมีตัวแปรที่ใช้กำหนดรูปแบบการค้นหา โดยแบ่งเป็น 2 รูปแบบคือ
    - **LocaseInsensitive** เป็นการกำหนดให้ค้นหาข้อมูล โดยไม่สนใจเรื่องประเภทของตัวอักษร ไม่ว่าจะเป็นตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ จะถือว่าเป็นตัวเดียวกัน
    - **LoPartialKey** ข้อมูลที่หาเจอจะเป็นข้อมูลที่มีบางส่วนของข้อมูลเหมือนกับค่าที่กำหนดไว้ เช่น ต้องการค้นหาข้อมูลที่มี 'BAT' ข้อมูลที่ค้นหาเจออาจจะเป็น 'BAT', 'DEBATE' หรือ 'PROBATION' เป็นต้น
- เมธอดในการจัดการกับข้อมูล**
- **Append** เพิ่มข้อมูลเข้าไปในตารางที่ละเรคคอร์ด
  - **Delete** ใช้ลบข้อมูลออกจากตารางที่ละเรคคอร์ด
  - **Edit** แก้ไขข้อมูลทีละเรคคอร์ด
  - **Post** บันทึกการเปลี่ยนแปลงที่เกิดขึ้นในตาราง
  - **Cancel** ยกเลิกการเปลี่ยนแปลงที่เกิดขึ้นในตาราง

### คอมโพเนนต์ MyQuery

เป็นคอมโพเนนต์ที่ใช้เข้าถึงข้อมูลในฐานข้อมูลเช่นเดียวกับคอมโพเนนต์ตาราง แต่สามารถเข้าถึงข้อมูลได้ตั้งแต่ 1 ตารางขึ้นไป ด้วยการ ใช้คำสั่ง SQL ในการดึงข้อมูลที่ต้องการออกมา

**พรีอเพอร์ติ์ที่สำคัญ**

- **Active** ถ้ากำหนดค่าเป็น True จะสามารถใช้งานข้อมูลจากคิวรีได้ แต่ถ้าเป็น False จะใช้งานไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **DataSource** กำหนดแหล่งข้อมูลที่เราต้องการค้นหาข้อมูลจากฐานข้อมูล
- **Params** เป็นการกำหนดค่าเพื่อที่จะส่งเข้าไปแทนที่ตัวแปรใน SQL เพื่อให้คำสั่ง SQL สมบูรณ์
- **RecordCount** นับเร็คคอร์ดที่ใช้ในการดึงข้อมูล
- **SQL** ใช้เก็บคำสั่ง SQL ที่ใช้ในการค้นหาข้อมูล โดยสามารถเก็บคำสั่ง SQL ได้เพียง 1 ชุดคำสั่งเท่านั้น

#### เมธอดที่สำคัญ

- **paramByName** เราจะใช้เมธอดนี้ในการกำหนดค่ากับพารามิเตอร์ที่ถูกกำหนดขึ้นในคำสั่ง SQL
- **ExecSQL** เป็นเมธอดที่สั่งให้คำสั่ง SQL ในคิวรีทำงาน เราจะใช้เมธอดนี้ในกรณีที่ไม่มีการส่งค่ากลุ่มข้อมูลกลับมา เช่น การเพิ่มเร็คคอร์ด (INSERT), การลบเร็คคอร์ด (DELETE) เป็นต้น ส่วนประโยคที่ใช้ SELECT เราจะเรียกใช้เมธอด Open แทน

#### 2.3.3.2 คอมโพเนนต์ในเพจ Varian Freeware

คอมโพเนนต์ในเพจนี้เป็นคอมโพเนนต์ที่สามารถทำให้เราติดต่อกับพอร์ตอนุกรม ซึ่งแสดงได้ดัง

รูป



รูปที่ 2.25 คอมโพเนนต์ในเพจ Varian Freeware

#### คอมโพเนนต์ Comm(Asyn32)

คอมโพเนนต์ ที่ใช้ติดต่อกับพอร์ตอนุกรม โดยสามารถที่จะกำหนดค่าต่างๆ เช่น ค่า BuadRate, Databits, stopbits เป็นต้น

##### พอร์พเพอร์ตีที่สำคัญ

- **BuadRate** ใช้กำหนดค่าBuadRate
- **Databits** ใช้กำหนดค่าdatabits
- **DeviceName** ใช้กำหนดคอมพอร์ตที่เราต้องการติดต่อ
- **Parity** ใช้กำหนดค่าParity bit
- **Stopbits** ใช้กำหนดค่าStopbitsของข้อมูล

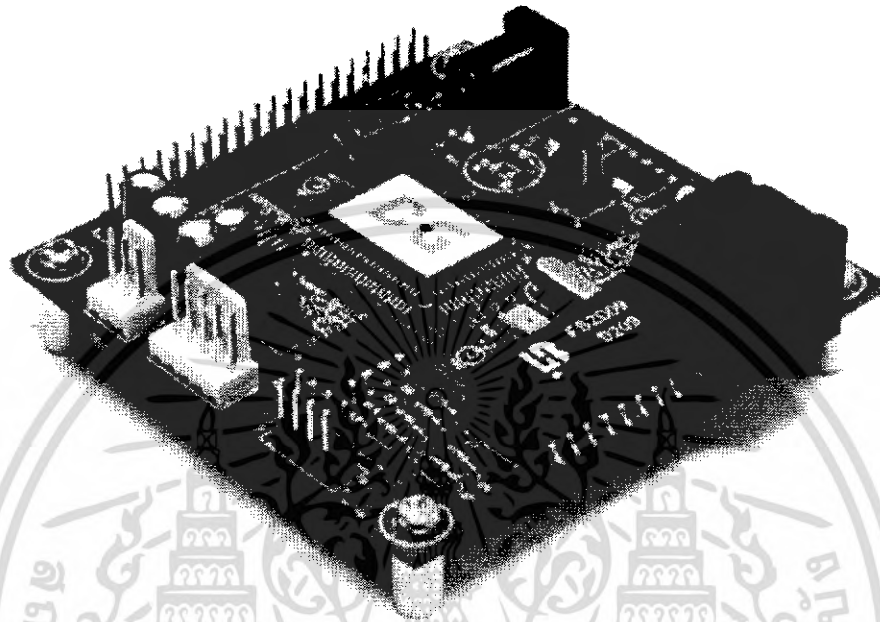
##### เมธอดที่สำคัญ

- **Open** ใช้เพื่อเปิดการเชื่อมต่อคอมโพเนนต์ Comm(Asyn32) กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Close      พอร์ตอนุกรม  
ใช้เพื่อเปิดการเชื่อมต่อคอมโพเนนต์ Comm(Asyn32) กับ  
พอร์ตอนุกรม

## 2.4 Ethernet IO Board



รูปที่ 2.26 Ethernet IO บอร์ด

อีเธอร์เน็ต ไอโอ บอร์ด เป็นเน็ตเวิร์ค ไอโอ บอร์ด พร้อมด้วย Software Library สำหรับทำกาควบคุม I/O ผ่านเครือข่าย Ethernet ซึ่งมีให้ เลือกใช้ทั้ง UART Port และ Parallel I/O เหมาะสำหรับผู้ที่ไม่มีความคุ้นเคยเกี่ยวกับการเขียน โปรแกรมผ่าน Network แต่สามารถควบคุมการทำงานผ่านเครือข่ายได้ รับส่งข้อมูลผ่าน LAN ซึ่งใช้โปรโตคอล TCP/IP Stack สามารถควบคุม GPIO และสามารถวัดสัญญาณอะนาล็อกมาเป็นดิจิทัลด้วย ADC และเปลี่ยนการรับส่งข้อมูลจาก RS 232 ไปเป็นการรับส่งผ่าน LAN

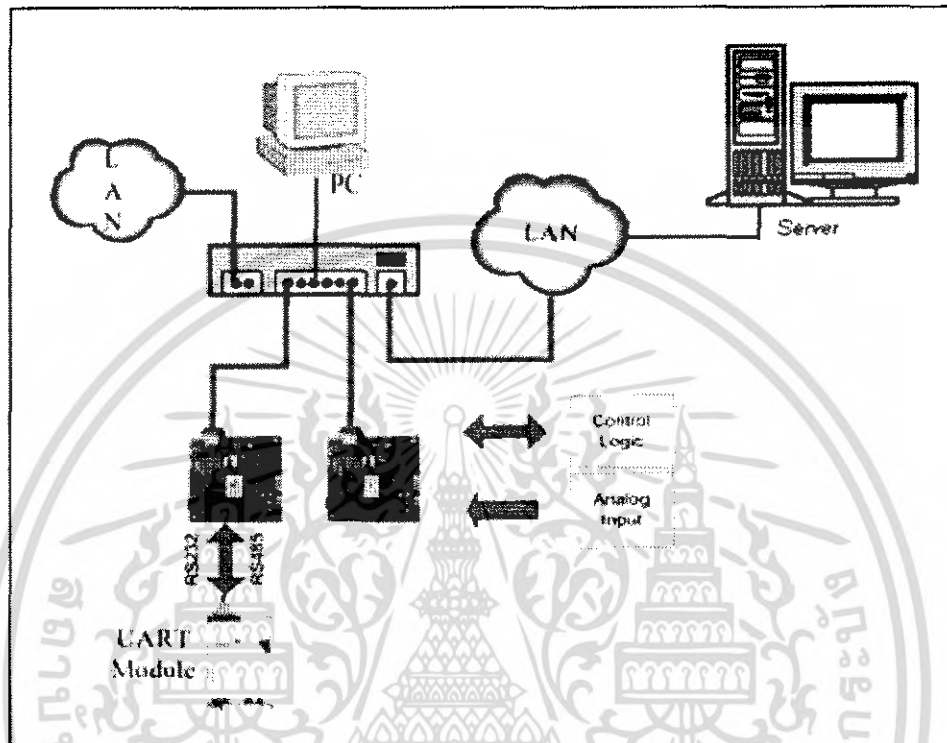
### คุณลักษณะเด่นของ Ethernet IO Board

- ลดระยะเวลาในการพัฒนาไคร้เวอร์,ซอฟต์แวร์ ในส่วนของการจัดการเกี่ยวกับ TCP/IP Protocol Stack
- สามารถควบคุมอุปกรณ์ผ่าน Ethernet ได้โดยไม่ต้องรู้ลึกเกี่ยวกับ Protocol Stack และ socket programming
- สามารถเปลี่ยนการเชื่อมต่อแบบ RS232/RS485 เป็นการเชื่อมต่อแบบ LAN ได้เมื่อต้องการเชื่อมต่ออุปกรณ์จาก RS232 หรือ RS485
- สามารถควบคุมลอจิกของ GPIO และการอ่านค่าจาก ADC จาก Ethernet IO Board ผ่าน LAN ได้โดยใช้โปรแกรม VB6.0 และ VC++6.0 เรียกใช้งานผ่าน DLL ของ Ethernet IO
- สามารถใช้งานแบบ Virtual Comport

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-อุปกรณ์มีขนาดเล็กและประหยัดไฟเนื่องจากใช้ไฟเลี้ยง 5 V ใช้กระแสไฟ 173uA

### โครงสร้างทางฮาร์ดแวร์ของ Ethernet IO Board



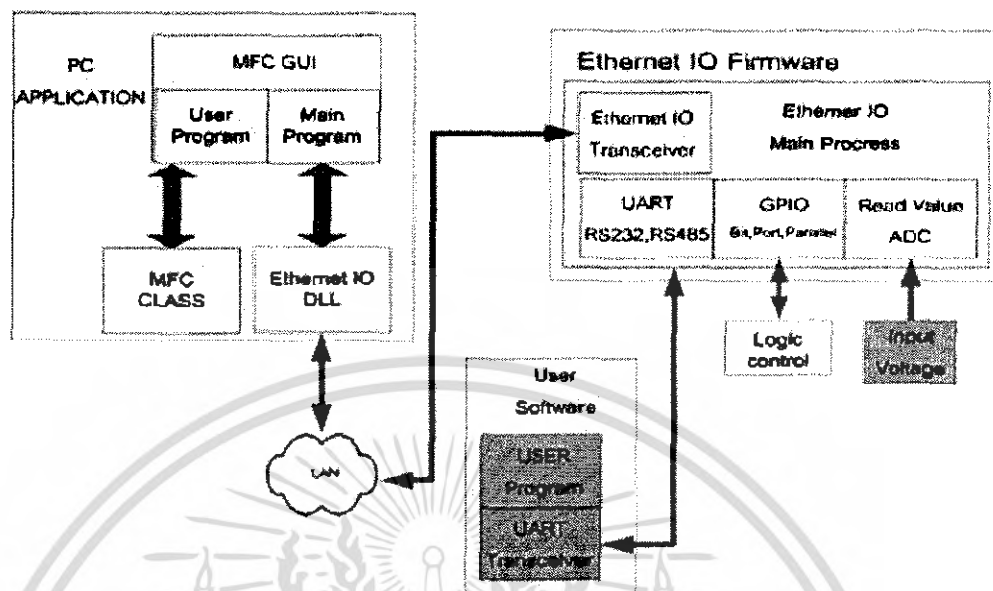
รูปที่ 2.27 แสดงตัวอย่างการเชื่อมต่อ Ethernet IO

Ethernet IO Board จะประกอบไปด้วย

- Embedded network 8- bit RISC microcontroller, speed 120 MIPS
- การเชื่อมต่อแบบ 10base-T (Ethernet 10 Mbit)
- 64 Kbytes flash memory
- 16 Kbytes SRAM program memory
- UART port RS232/RS485, อัตราเร็วในการรับส่งสูงสุด 115200 bps
- 35 bits General Purpose I/O , สามารถทนแรงดันไฟฟ้าได้ +5V และขับแรงดันไฟฟ้าได้+2.5 V
- 10- bit ,6-channel ADC,ค่าสูงสุดในการ sampling rate 48 kHz,รับค่าแรงดันสูงสุด 2.5V
- มีพื้นที่ของ flash memory 400 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โครงสร้างของซอฟต์แวร์ของ Ethernet IO Board



รูปที่ 2.28 แสดงโครงสร้างของซอฟต์แวร์

โครงสร้างของซอฟต์แวร์จะแบ่งออกเป็น 3 ส่วน คือ

1. APPLICATION เป็นส่วนซอฟต์แวร์ที่ใช้ในการติดต่อรับส่งข้อมูลระหว่าง PC กับ Ethernet IO ผ่าน DLL ซึ่งการติดต่อ DLL สามารถใช้ได้กับ Visual Basic และ Visual C++

ในการพัฒนา APPLICATION

2. Ethernet IO Firmware เป็นส่วนซอฟต์แวร์ที่ใช้ในการควบคุม Ethernet IO ในการเชื่อมต่อระหว่าง PC กับบอร์ด Ethernet IO สำหรับการควบคุมใช้งานของ UART, GPIO และ ADC

3. User Software เป็นส่วนซอฟต์แวร์ ของผู้ใช้งานที่ใช้ทำการส่งข้อมูลผ่าน UART เพื่อทำการแปลงการรับส่งข้อมูลจาก UART ไปเป็น LAN ควบคุม GPIO และการกำหนดค่าต่างๆ ของ Ethernet IO Board

### ลักษณะวิธีการ ใช้งาน Ethernet IO Board

การนำ Ethernet IO ไปใช้งานสามารถทำได้ 3 วิธี

1. ทำการติดต่อโดยการเรียกผ่าน socket programming โดยสามารถเลือกใช้โปรแกรมภาษาอะไรก็ได้ เช่น C/C++, VB, VC++ หรือจะใช้โปรแกรมอื่นๆที่สามารถติดต่อข้อมูลผ่าน LAN โดยใช้ TCP Protocol ในการส่งข้อมูลผ่าน Port 4025 เมื่อ Ethernet IO รับข้อมูลจาก RS232 Ethernet IO จะทำการส่งข้อมูลผ่าน LAN ไปทาง port 4025 โดยใช้ TCP Protocol และเมื่อ Ethernet IO รับข้อมูลจาก LAN โดยทาง port 4025 และใช้ส่งข้อมูลแบบ TCP Protocol แล้ว Ethernet IO ก็จะทำการส่งข้อมูลออกไปยัง RS232 โดยอัตโนมัติ

2. ทำการติดต่อโดยใช้ Ethernet IO DLL (Dynamic Link Library) โดยทำการเขียนแอบพริเคชันสำหรับการติดต่อซึ่งในการเขียนแอบพริเคชันติดต่อการใช้งาน Ethernet IO ผ่าน DLL นั้น Ethernet IO DLL เป็น DLL แบบ Regular DLL จึงสามารถเรียกใช้งานได้จาก VB6 และ VC6 ได้
3. ทำการติดต่อในลักษณะ Virtual Serial Comport โดยใช้โปรแกรม HW VSP ในการแปลงการเชื่อมต่อแบบ LAN ให้ PC มองเป็น Comport และสามารถใช้งานได้เหมือน serial port โดยเชื่อมต่อผ่าน port 4025 สามารถกำหนดใช้ได้ตั้งแต่ Comport 2-20

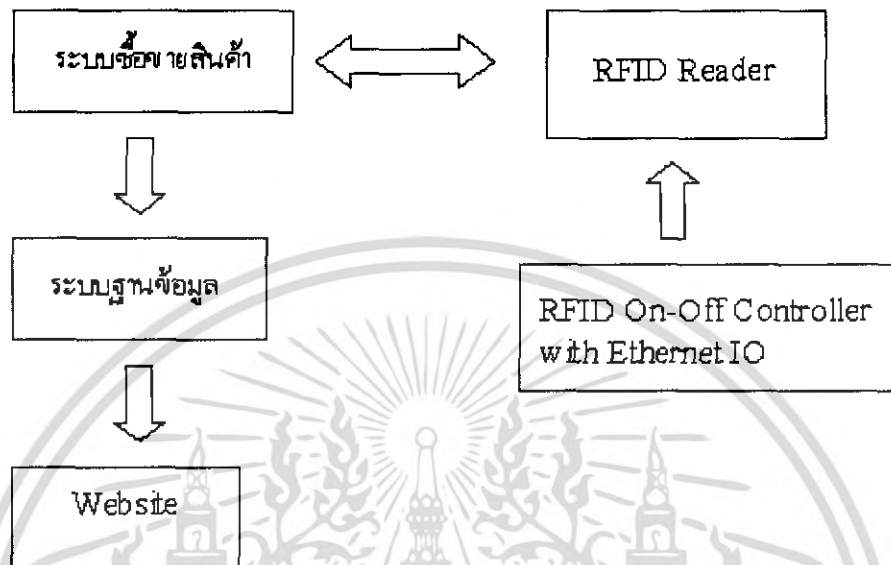


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### การคำนวณและการสร้าง

แผนผังการทำงานทั้งหมด



รูปที่ 3.1 แสดงแผนผังการทำงานระบบ

#### 3.1 การออกแบบส่วนของ RFID

ระบบ RFID จะมีในส่วนของเครื่องอ่านสัญญาณและส่วนของแท็กส์ ในแต่ละส่วนจะมีไอซีที่ใช้ทำงานแตกต่างกันไป โดยในส่วนของเครื่องอ่านสัญญาณเราจะใช้ไอซี EM4095 และในส่วนของแท็กส์ เราจะเป็นแบบ T5557

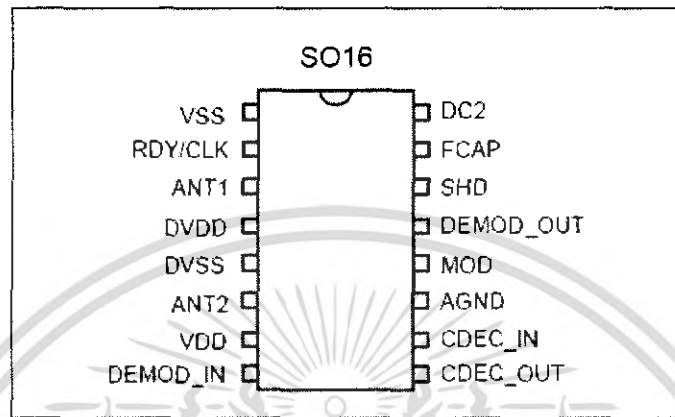
##### 3.1.1 EM4095 : อ่าน/เขียน อนาคตสำหรับ 125 kHz RFID เบลสตัน

ไอซี EM4095 (หรือ P4095) เป็นไอซีแบบวงจรรวมซีมอสสร้างขึ้นมาเพื่อใช้ในเครื่องอ่านสัญญาณซึ่งมีคุณสมบัติดังนี้

- ใช้ในย่านความถี่ 125 kHz - 134 kHz
- มีเสาอากาศขับเคลื่อนด้วยความถี่คลื่นพาห์
- การมอดคูเลตสัญญาณแบบเอเอ็มสำหรับเขียนข้อมูล
- การดีมอดคูเลตสัญญาณแบบเอเอ็มจากเสาอากาศที่ตรวจจับได้
- ใช้กับระบบเฟสล็อกเพื่อให้อ่านความถี่พาหะที่ใช้กับเสาอากาศ
- ใช้บริทจ์ไดรเวอร์ขับเคลื่อนเสาอากาศโดยตรง
- ส่งผ่านข้อมูลแบบแอมพลิฟายด์ มอดูเลชัน กับ คิซนี่การมอดคูเลตแบบปรับได้ภายนอก โดยใช้ไดรเวอร์แบบซิงเกิลเอน

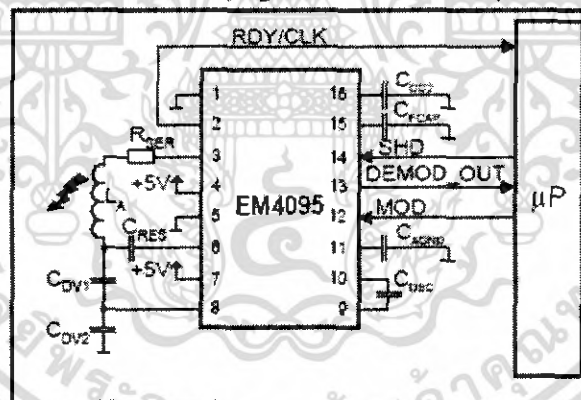
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีโปรโตคอลแท็กส์แบบอื่นๆที่สามารถใช้ร่วมกันได้ เช่น EM400X, EM4050, EM4150, EM4070, EM4170, EM4069 เป็นต้น
- ช่วงอุณหภูมิ 40 ถึง 85 องศาเซลเซียส



รูปที่ 3.2 ขาของไอซี EM4095

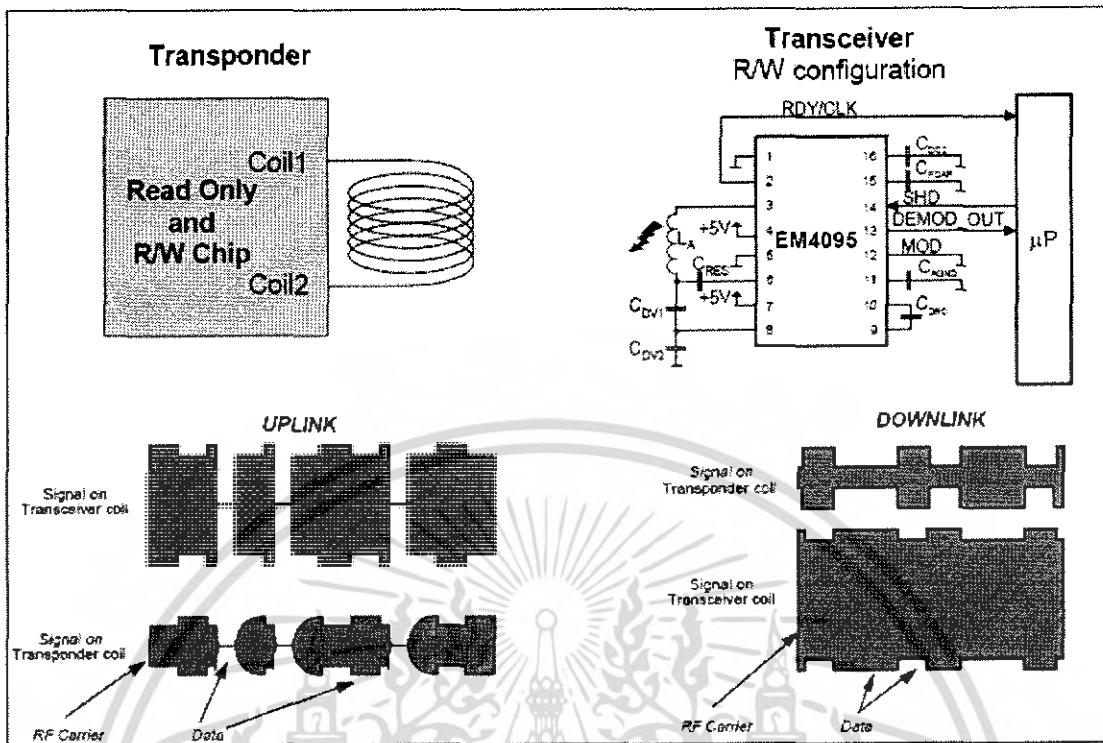
Read/Write mode (High Q factor antenna)



รูปที่ 3.3 วงจรที่ใช้ต่อ EM4095 กับไมโครคอนโทรลเลอร์

เหตุผลที่ใช้วงจรนี้ก็เพราะว่าวงจรนี้มีประสิทธิภาพสูง เนื่องจากวงจรนี้มีการต่อค่าความต้านทานอนุกรมกับขดลวดทำให้กระแสที่ไหลผ่านขดลวดมีค่าลดลง เมื่อกระแสลดลงก็จะมีผลให้ค่ากำลังมีค่าลดลงด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

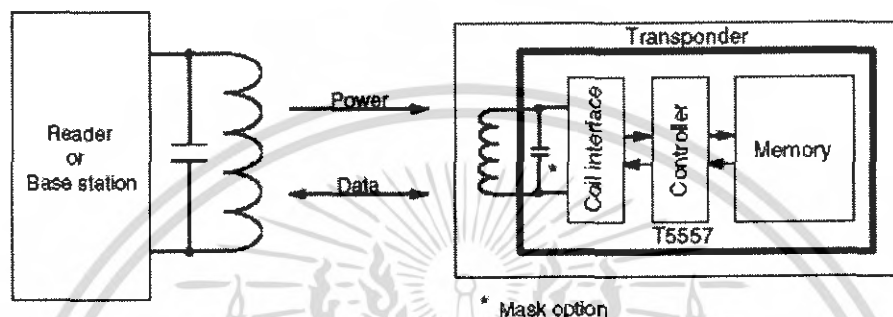


รูปที่ 3.4 หลักการของระบบ

EM4095 นั้นถูกนำไปใช้กับวงจรเสาอากาศและไมโครคอนโทรลเลอร์ จำเป็นต้องได้รับองค์ประกอบภายนอกคือตัวกรองไฟฟ้ากระแสตรง และ คลื่นวิทยุ, การเซนเซอร์กระแส และการดีคัปปีงแหล่งจ่ายไฟตรง ส่วนระดับของแหล่งจ่ายไฟตรงที่ใช้เลี้ยงตัวชิพนั้นต้องอ้างอิงจากคำอธิบายการใช้ EM4095 การทำงานของของไอซีนี้ถูกควบคุมโดยลอคจิกอินพุทที่ขา SHD และ MOD ของไอซี เมื่อ SHD เป็น High ไอซี EM4095 จะทำงานในสลับโหมดใช้กระแสไฟฟ้าน้อยที่สุด เมื่อมีพลังงานเพิ่มขึ้นที่ขา SHD ทำให้ขา SHD จะมีสถานะเป็นไฮท์เพื่อให้สามารถเริ่มต้นได้อย่างถูกต้อง เมื่อ SHD เป็นโลว์ไอซี EM4095 จะสามารถปล่อยสนามคลื่นวิทยุออกมา ซึ่งมันจะเริ่มต้นทำการดีมอดูเลตทุกๆสัญญาณเอเอ็มที่พบบนเสาอากาศ สัญญาณดิจิตอลนี้มาจากบล็อกของการดีมอดูเลตของสัญญาณเอเอ็มที่ส่งผ่านขา DEMOD\_OUT ของไอซีไปยังไมโครคอนโทรลเลอร์เพื่อทำการถอดรหัสและประมวลผล ถ้าขา MOD ของไอซี EM4095 เป็นไฮท์จะไปมีผลทำให้ไครเวอร์ของเสาอากาศหลักซิงโครนัสกับคลื่นพาห์ ขณะที่ขา MOD นั้นเป็นไฮท์ซึ่ง และการดีมอดูเลตแบบเอเอ็มจะถูกคงสภาพการทำงานไว้เหมือนเดิมดังก่อนที่ขา MOD ของไอซีจะเป็นไฮท์ซึ่งจะทำให้วีซีโอและการดีมอดูเลตแบบเอเอ็มนั้นจะทำงานเข้าไป 41 อาร์เอฟล๊อค หลังจากขา MOD ของไอซีเป็นไฮท์การทำงานของวีซีโอ และการดีมอดูเลตแบบ เอเอ็มในจุดนี้จะไม่ถูกทำให้ยุ่งยากโดยวงจรเรโซแนนซ์ของเสาอากาศ

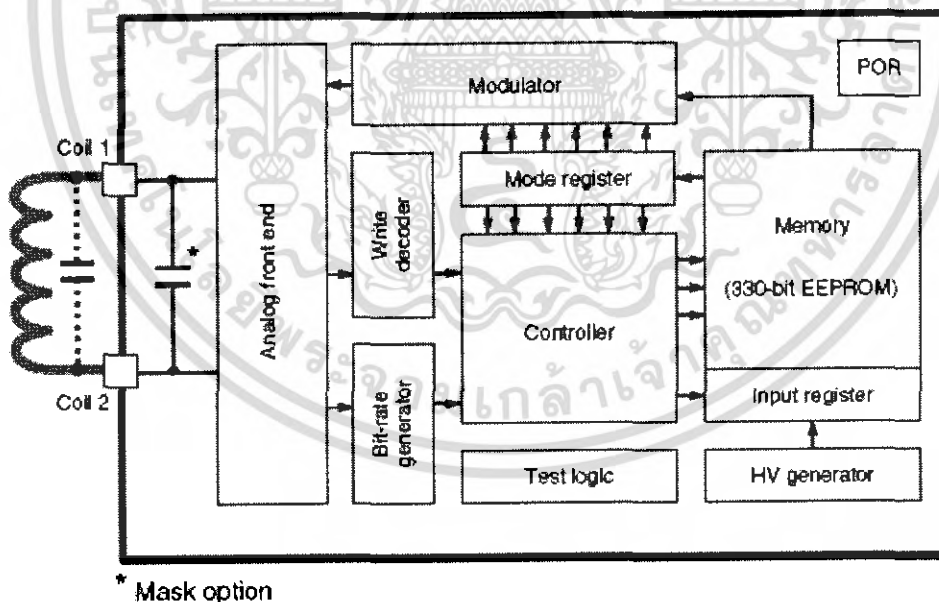
### 3.1.2 Tag : Multifunctional 330 bit Read/Write RF-Identification IC

T5557 เป็นไอซีเฉพาะ แบบไร้สัมผัส สำหรับการประยุกต์ใช้ในย่านความถี่ 125 kHz มีขดลวดต่ออยู่กับชิพ (chip) ใช้เป็นแหล่งจ่ายพลังงาน และรับส่งข้อมูลระหว่างตัวเองกับเครื่องอ่านสัญญาณ (Reader or Base station) ใน T5557 นี้ มีหน่วยความจำเป็นแบบ EEPROM 330-bit (แบ่งเป็น 10 บล็อก บล็อกละ 33 บิต) สามารถอ่านและเขียนบล็อกได้จาก Reader โดยบล็อก 0 จะถูกสงวนไว้ในการตั้งค่า โหมมการทำงานของ T5557 และ บล็อก 7 อาจใช้สำหรับเก็บรหัสผ่าน เพื่อป้องกันผู้อื่นมาเขียน



รูปที่ 3.5 ระบบ RFID โดยใช้ T5557

#### T5557 Building Blocks



รูปที่ 3.6 บล็อกไดอะแกรม

#### คุณสมบัติของ T5557

- ส่งผ่านข้อมูลแบบไร้สัมผัส
- ความถี่คลื่นวิทยุตั้งแต่ 100 kHz ถึง 150 kHz
- E5550 เข้ากันได้กับ T5557 extended mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขนาดเล็ก,เหมาะสมกับ ISO/IEC 11784/785
- มีตัวเก็บประจุเรโซแนนซ์ 75 pF บนชิพ
- มีหน่วยความจำข้อมูลแบบ EEPROM ขนาด  $7 \times 32$ บิต รวมทั้งพาสเวิร์ดขนาด 32บิต
- แบ่งความจำ 64 บิตไว้สำหรับข้อมูล
- อีก 32บิตของรีจิสเตอร์ใน EEPROM นั้นกำหนด

-อัตราข้อมูล

- RF/2 ถึง RF/8, Binary Selectable หรือ

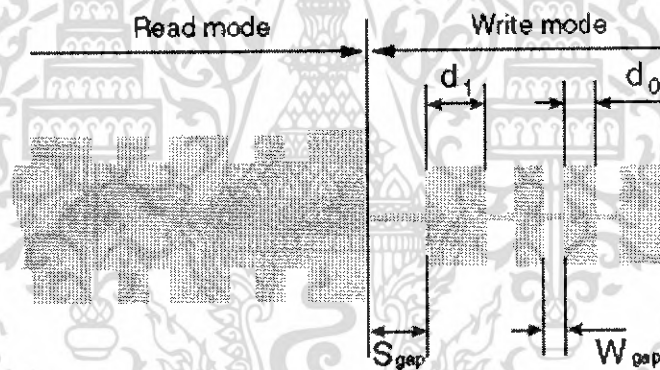
- Fixed E5550 Data Rates

- การมอดูเลต/การเข้ารหัส

- FSK, PSK, แมนเชสเตอร์, ไบเฟส, NRZ

### 3.1.2.1 การติดต่อระหว่าง Reader กับ Tag

เราสามารถเขียนข้อมูลไปที่แท็กได้โดยการรบกวน RF field ด้วย field gaps ดังนี้ ดังรูปและตารางด้านล่าง



รูปที่ 3.7 การเริ่มติดต่อระหว่างเครื่องอ่านสัญญาณกับแท็ก

Parameters	Remark	Symbol	Min.	Max.	Unit
Start gap		$S_{gap}$	10	50	FC
Write gap	Normal write mode	$W_{gap}$	8	30	FC
Write data in normal mode	'0' data	$d_0$	16	31	FC
	'1' data	$d_1$	48	63	FC

### ตารางที่ 3.1 การถอดรหัสข้อมูลที่เขียน

จากรูปและตารางจะเห็นว่า เมื่อเราจะเขียนข้อมูลจะต้องเริ่มจากช่วงเริ่มต้นเสมอซึ่งมีค่าประมาณ 30 FC ( $(10+50)/2$ ) หลังจากนั้นก็จะตามด้วยข้อมูลที่ต้องการเขียน (ถ้าต้องการเขียน 1 ก็จะต้องใช้ประมาณ 54 FC ถ้าต้องการเขียน 0 ใช้ประมาณ 24 FC) หลังจากนั้นจะตามด้วย

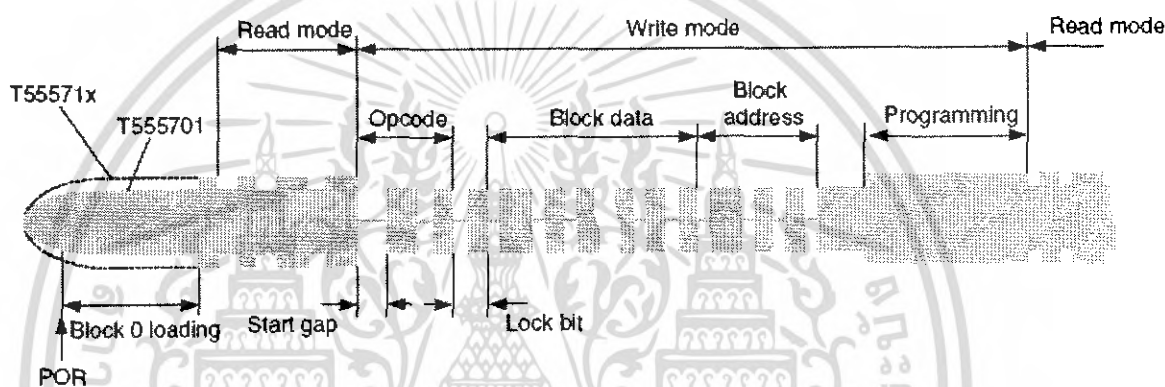
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Write gap ซึ่งมีค่าประมาณ 18 FC อย่างนี้ไปเรื่อยๆ จนกระทั่งไม่มี gap มากกว่า 64 FC ก็จะออก จาก Write mode

### 3.1.2.2 หลักในการเขียนข้อมูล

- ในการเขียนทั่วไปต้องการ

1. opcode
2. lock bit
3. บิตข้อมูลอีก 32 บิต
4. 3 bit address ใช้เลือก block ที่จะเขียน



รูปที่ 3.8 รูปร่างสัญญาณขณะเขียนข้อมูล

	Opcode						
Standard write	1p*	L	1	Data	32	2	Addr 0
Protected write	1p*	1	Password	32	L	1	Data 32 2 Addr 0
AOR (wake-up command)	10	1	Password	32			
Direct access (PWD = 1)	1p*	1	Password	32	0	2	Addr 0
Direct access (PWD = 0)	1p*	0	2	Addr 0			
Page 0/1 regular read	1p*						
Reset command	00						

\* p = page selector

รูปที่ 3.9 รูปแบบในการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3 โปรโตคอลการติดต่อสื่อสาร

โปรโตคอลการติดต่อสื่อสารระหว่างแท็กส์กับเครื่องอ่านสัญญาณ ซึ่งจะใช้สำหรับอ่านข้อมูลจากแท็กส์หรือเขียนข้อมูลลงในแท็กส์

#### 3.1.3.1 คำสั่งการอ่านแท็กส์

การอ่านข้อมูลจากแท็กส์จะอ่านโดยระบุเซ็กเมนต์โดยเครื่องอ่านจะดึงข้อมูลที่อ่านจากแท็กส์ตามที่ระบุส่งข้อมูลกลับมาให้ตามตำแหน่งที่ระบุไว้

การส่งของ GUI

1 byte	1 byte	1 byte	1 byte	1 byte
STR	Command Read	Segment	CRC	END

ตารางที่ 3.2 แสดงการส่ง Command Read ของ GUI

ซึ่งในการตอบสนองของเครื่องอ่านกรณีไม่มีความผิดพลาดในการตรวจสอบ CRC หรือความผิดพลาดในการอ่านแท็กส์

1 byte	1 byte	1 byte	28/4 bytes	1 byte	1 byte
STR	Command Read	Segment	DATA	CRC	END

ตารางที่ 3.3 แสดงผลตอบสนองของ Command Read กรณีที่ไม่มีความผิดพลาดแต่ในกรณีที่มีความผิดพลาดเกิดขึ้น เครื่องอ่านจะตอบสนองด้วยการส่ง

1 byte	1 byte	1 byte	1 byte	1 byte
STR	Command Read	Status	CRC	END

ตารางที่ 3.4 แสดงผลตอบสนองของ Command Read กรณีที่มีความผิดพลาดเกิดขึ้น หลังจากส่งเฟรมเพื่อบอกความผิดพลาดแล้วจะกลับไปทำงานในสถานะเริ่มต้น เพื่อรอคำสั่งใหม่

#### 3.1.3.2 คำสั่งการเขียน

การส่งของ GUI

1 byte	1 byte	1 byte	4 bytes	1 byte	1 byte
STR	Command Write	Segment	DATA	CRC	END

ตารางที่ 3.5 แสดงการส่ง Command Write ของ GUI

ซึ่งในการตอบสนองของเครื่องอ่าน

1 byte	1 byte	1 byte	1 byte	1 byte
STR	Command Write	Status	CRC	END

ตารางที่ 3.6 แสดงการตอบสนองของการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3.3 โทด์คำสั่ง

คำสั่ง	โทด์	ทิศทาง
		แทกส์ <--> เครื่องอ่าน
อ่าน	0x02	สัญญาณ
เขียน	0x04	แทกส์ <--> เครื่องอ่านสัญญาณ

ตารางที่ 3.7 แสดงโทด์คำสั่ง

### 3.1.3.4 โทด์ของเฮดเดอร์และสถานะ

คำอธิบาย	โทด์
STR	0xFB
END	0xFC
CRC BAD	0x88
OK	0x99
Write not complete	0xA1

ตารางที่ 3.8 แสดงโทด์ของเฮดเดอร์และสถานะ

### 3.1.3.5 โทด์เซ็กเมนต์

คำอธิบาย	โทด์
ตำแหน่งของข้อมูลบล็อก 0 หน้า 0	0x00
ตำแหน่งของข้อมูลบล็อก 1 หน้า 0	0x01
ตำแหน่งของข้อมูลบล็อก 2 หน้า 0	0x02
ตำแหน่งของข้อมูลบล็อก 3 หน้า 0	0x03
ตำแหน่งของข้อมูลบล็อก 4 หน้า 0	0x04
ตำแหน่งของข้อมูลบล็อก 5 หน้า 0	0x05
ตำแหน่งของข้อมูลบล็อก 6 หน้า 0	0x06
ตำแหน่งของข้อมูลบล็อก 7 หน้า 0	0x07
ตำแหน่งของข้อมูลบล็อก 1 หน้า 1	0x08
ตำแหน่งของข้อมูลบล็อก 2 หน้า 1	0x09

ตารางที่ 3.9 แสดงโทด์เซ็กเมนต์

### 3.1.3.6 การหา CRC ของแทกส์

การหา CRC ของแทกส์สามารถหาได้โดย XOR ข้อมูล เริ่มตั้งแต่ STR จนถึงข้อมูลก่อนหน้า

CRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3.7 คุณสมบัติในการติดต่อสื่อสาร

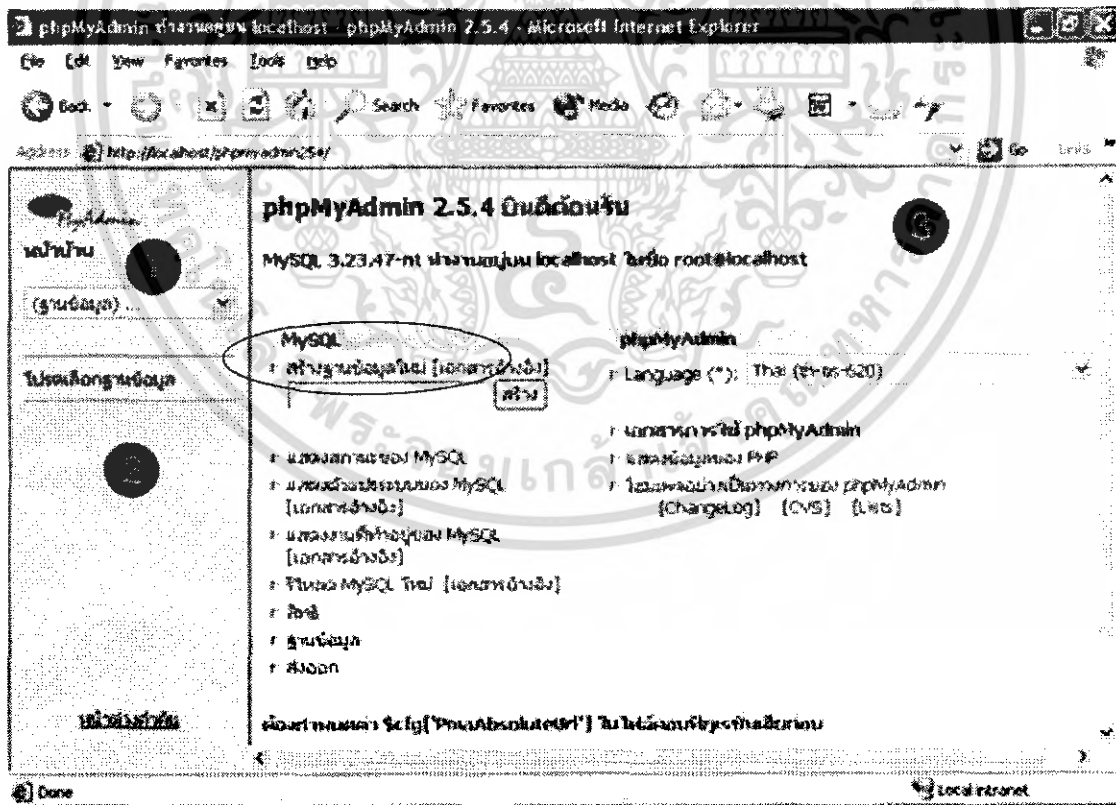
- บอรรถเร่งในการติดต่อ 19.2 Kbps
- ช่วงเวลาในการติดต่อสื่อสาร 5 วินาที หลังจากช่วงเวลาในการติดต่อสื่อสารหมดแล้ว ให้กลับไปอยู่ในสถานะรอคำสั่งใหม่เสมอ

การติดต่อกับแท่งสปีนแบบต่อเนื่อง คือสามารถส่งคำสั่งต่างกันได้ โดยไม่ต้องยกแท่งสอกระหว่างแต่ละคำสั่ง

## 3.2 การออกแบบระบบฐานข้อมูล

### 3.2.1 จัดการฐานข้อมูล Mysql ด้วย phpmyadmin

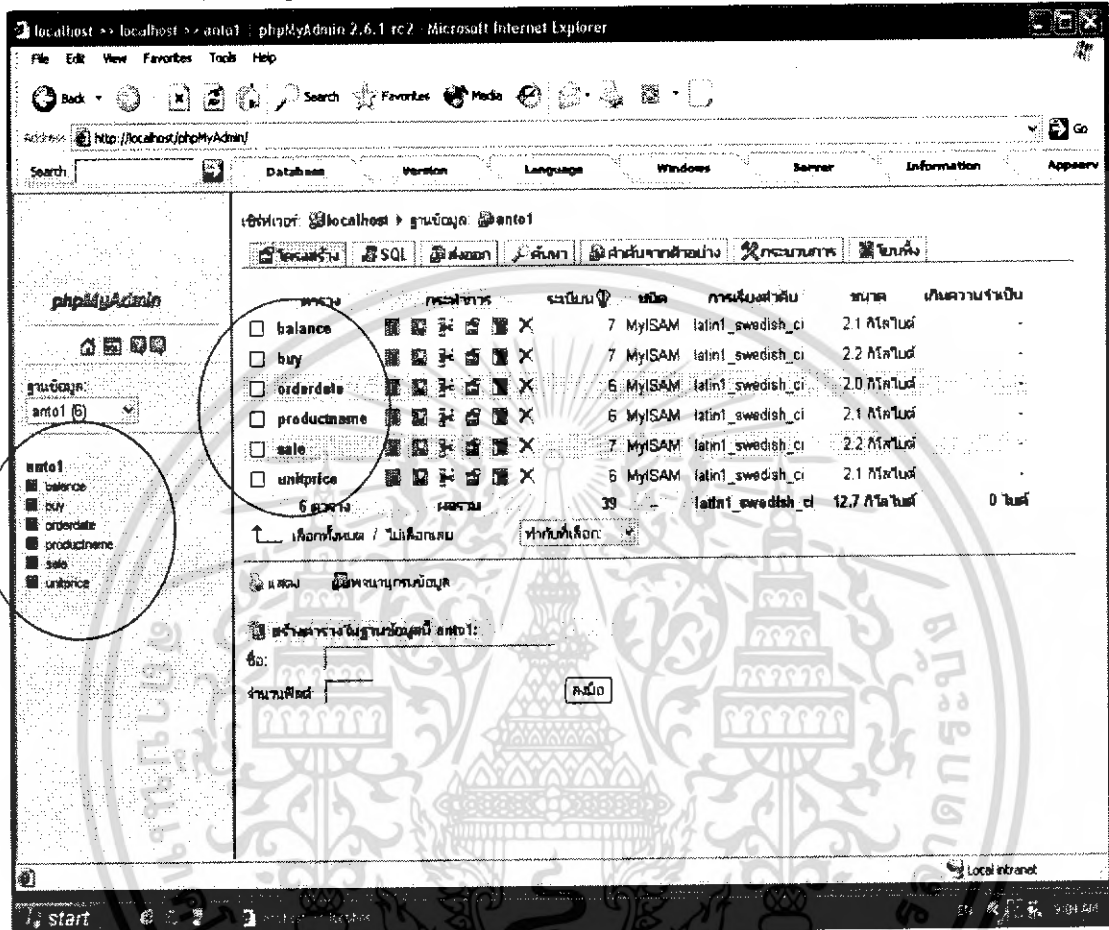
การใช้งานข้อมูล MySQL Database เป็นเรื่องลำบากมากสำหรับผู้เริ่มต้นใช้งานเกี่ยวกับคำสั่ง SQL จึงมีผู้ผลิตโปรแกรมสำหรับเข้าไปจัดการฐานข้อมูลให้ง่ายขึ้น เป็นแบบ Web Interface ก็ไม่ต้องติดตั้งโปรแกรมอะไรเพิ่มเติมที่เครื่องคอมพิวเตอร์ของเรา ขอเพียงเรามีโปรแกรม Web Browser ทั่วไปก็ใช้งานได้ โปรแกรมนี้เรียกว่า phpMyAdmin เมื่อได้ทำการติดตั้งเสร็จแล้ว ก็เปิดด้วย Browser จะได้นหน้าต่างแบบรูปข้างล่างนี้



รูปที่ 3.10 โปรแกรมสร้างฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราต้องการสร้างฐานข้อมูล เราสามารถพิมพ์ชื่อฐานข้อมูลที่เราต้องการสร้างได้เลยแล้ว โปรแกรมนี้ก็จะสร้างฐานข้อมูลขึ้นมา โดยที่เราสามารถนำไปสร้างเป็นตารางย่อยได้ตามต้องการ โดยที่เรากำหนด field ขนาดของข้อมูล ประเภทของข้อมูล และสามารถใส่ค่าของข้อมูลได้ตามต้องการ ซึ่งจะช่วยให้เราจัดการกับฐานข้อมูล Mysql ได้อย่างง่ายดาย

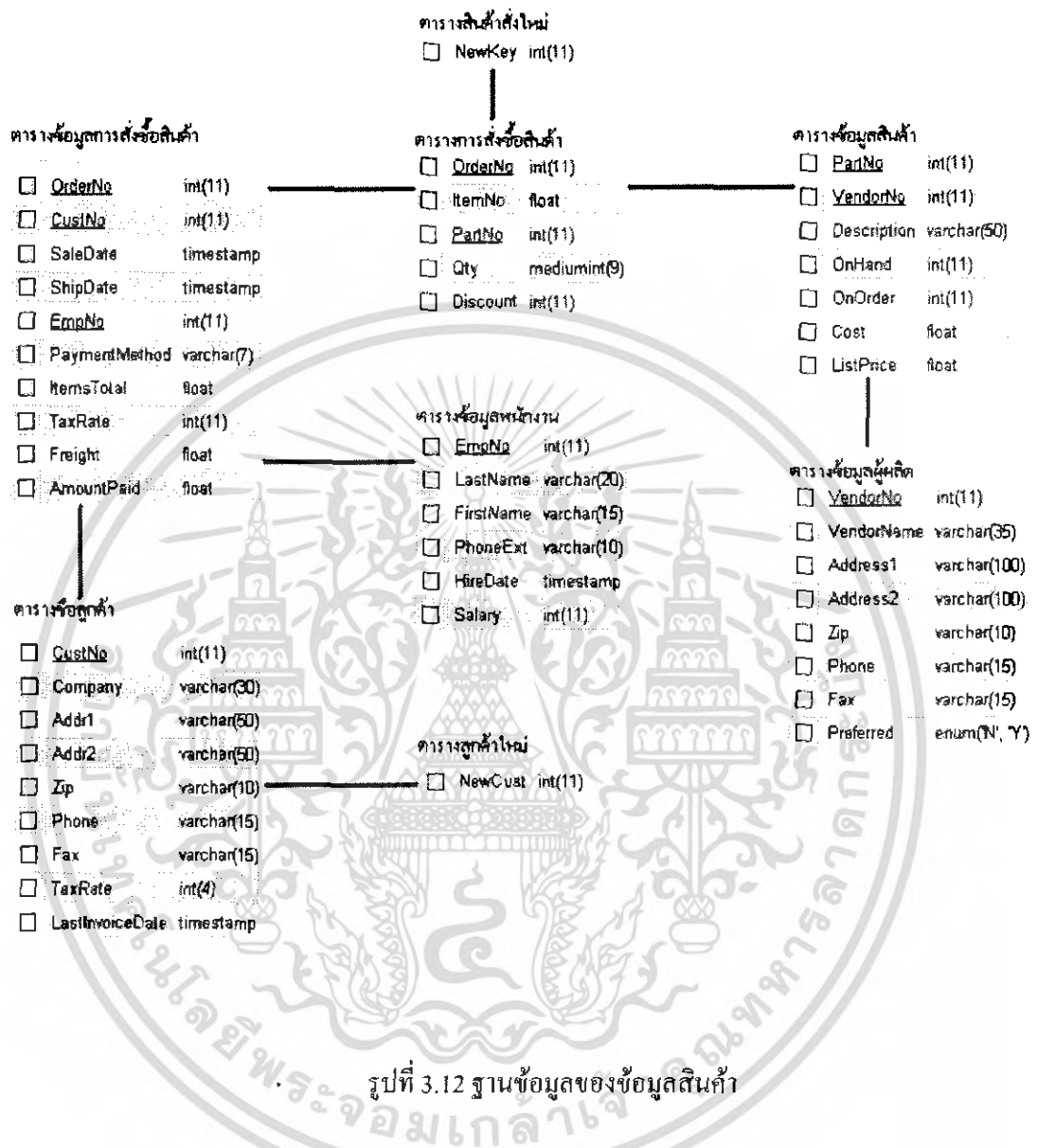


รูปที่ 3.11 ระบบฐานข้อมูลของรายการสินค้า

รูปที่ 3.11 เป็นระบบฐานข้อมูลของรายการสินค้าที่ถูกจำลองขึ้น ซึ่งประกอบด้วยกลุ่มของตารางทั้งหมด 2 ตารางด้วยกัน ตารางของการซื้อสินค้าและตารางขายสินค้า โดยมีโครงสร้างของตารางและความสัมพันธ์ระหว่างตารางดังรูปที่ 3.2 และรายละเอียดของข้อมูลในฐานข้อมูลได้แสดงไว้ในตารางที่ 3.1 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2รูปแบบความสัมพันธ์ของฐานข้อมูล



3.2.3รายละเอียดของข้อมูลในฐานข้อมูล

ตารางข้อมูลสินค้า	<u>PartNo</u>	รหัสสินค้า
	<u>VendorNo</u>	รหัสบริษัทที่ผลิต
	Description	รายการ
	OnHand	สินค้าที่มีอยู่
	OnOrder	สินค้าที่ถูกสั่ง
	Cost	ราคา
	ListPrice	ราคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรณีใช้งานที่องค์กรคือหน่วยงานนั้น ไม่ได้อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางข้อมูลผู้ผลิต	<u>VendorNo</u> VendorName Address1 Address2 Zip Phone Fax Preferred	รหัสบริษัทที่ผลิต ชื่อบริษัท ที่อยู่ 1 ที่อยู่ 2 รหัสไปรษณีย์ โทรศัพท์ แฟกซ์ ยืนยันการส่ง
ตารางการสั่งซื้อสินค้า	<u>OrderNo</u> ItemNo PartNo Qty Discount	รหัสสินค้าที่สั่ง รหัสการสั่ง รหัสสินค้า ปริมาณสินค้า -
ตารางข้อมูลการสั่งซื้อ สินค้า	<u>OrderNo</u> CustNo SaleDate ShipDate EmpNo PaymentMethod ItemsTotal TaxRate Freight AmountPaid	รหัสสินค้าที่สั่ง รหัสลูกค้า วันที่ขาย วันที่ส่งสินค้า รหัสพนักงานขาย วิธีชำระเงิน รวมรายการ ภาษี ค่าส่งสินค้า จำนวนเงินที่จ่าย
ตารางชื่อลูกค้า	<u>CustNo</u> Company Addr1 Addr2 Zip Phone TaxRate LastInvoiceDate	รหัสลูกค้า รหัสบริษัทที่สั่ง ที่อยู่ 1 ที่อยู่ 2 รหัสไปรษณีย์ โทรศัพท์ ภาษี วันที่ซื้อสินค้าล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

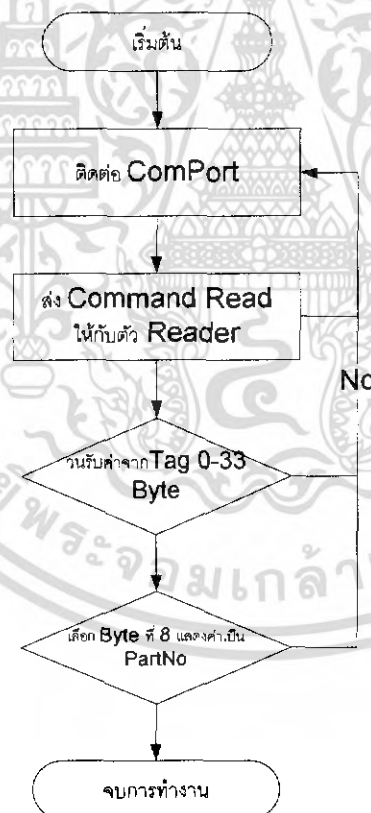
ตารางชื่อพนักงาน	<u>EmpNo</u> LastName FirstName PhoneExt HireDate Salary	รหัสพนักงาน นามสกุล ชื่อ โทรศัพท์ - เงินเดือน
ตารางสินค้าสั่งใหม่	NewKey	
ตารางลูกค้าใหม่	NewCust	

ตารางที่ 3.10 รายละเอียดของข้อมูลในฐานข้อมูล

### 3.3 การออกแบบโปรแกรม Delphi

#### 3.3.1 ขั้นตอนการติดต่อกับ Serial Port

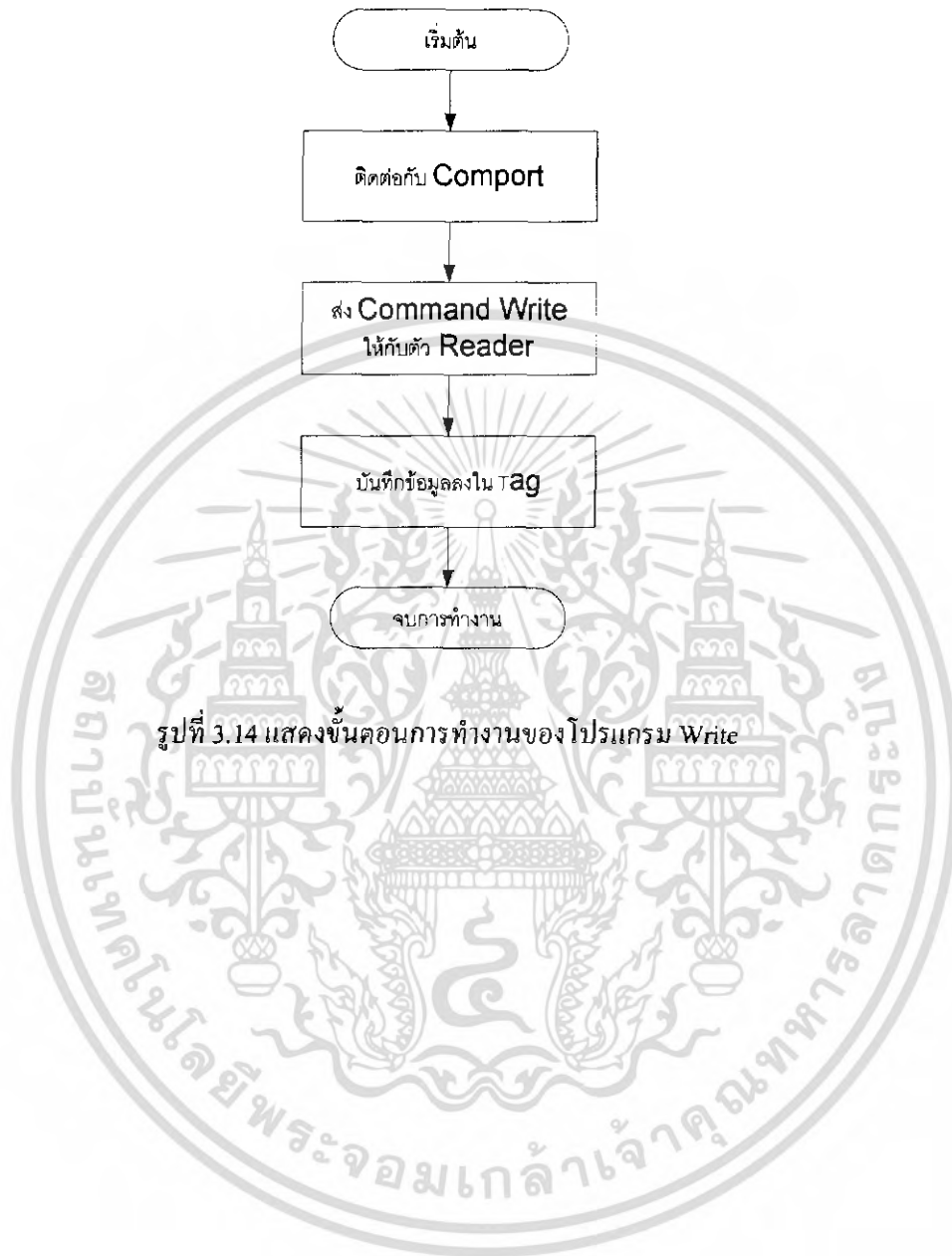
##### แผนผังแสดงการทำงานของโปรแกรม Read



รูปที่ 3.13 แสดงขั้นตอนการทำงานของโปรแกรม Read

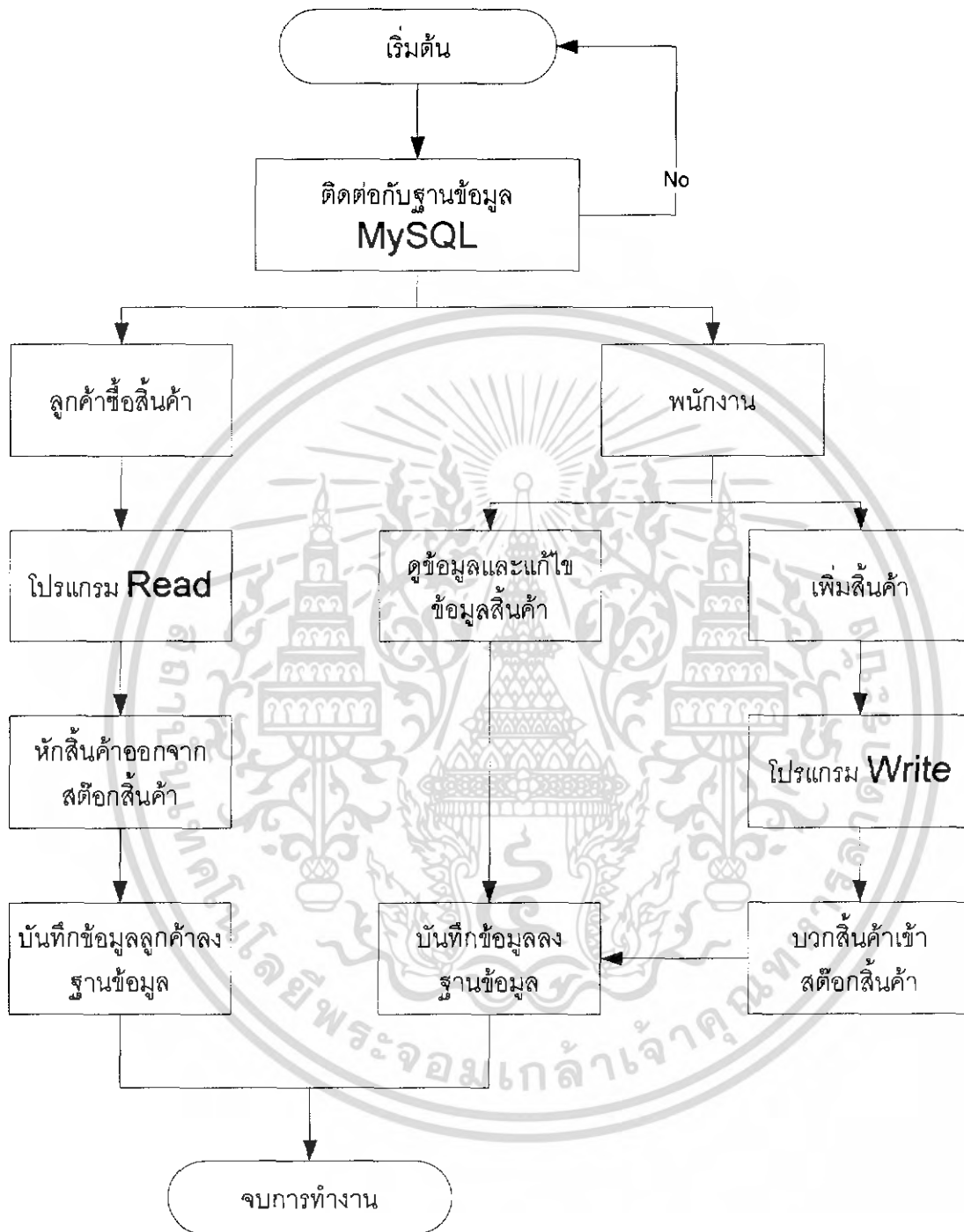
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แผนผังแสดงการทำงานของโปรแกรม Write



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.3.2 หลักการทำงานของระบบ

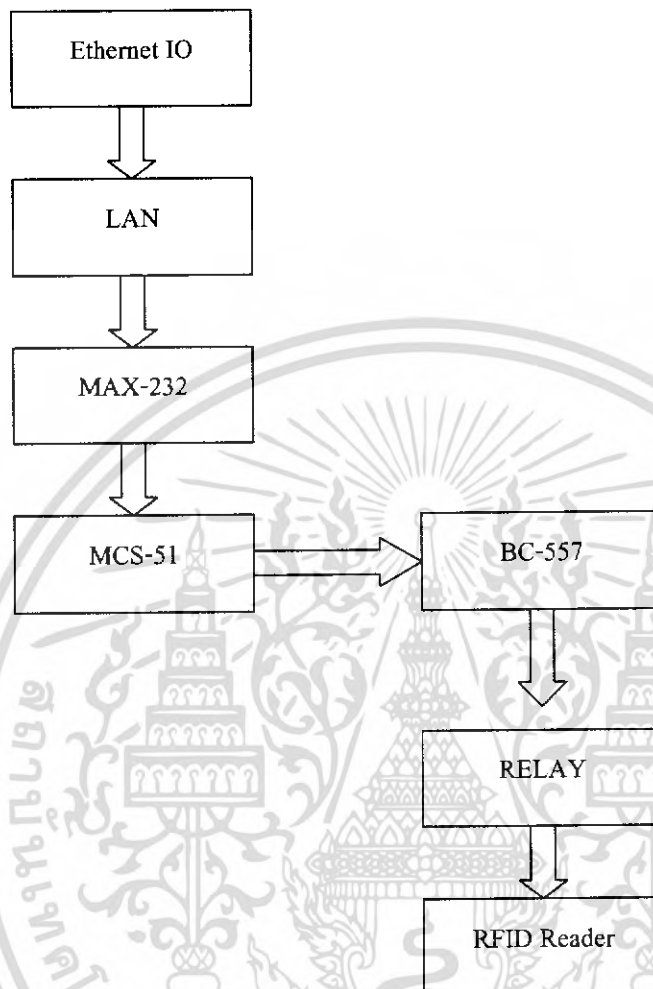


รูปที่ 3.15 ขั้นตอนการทำงานของระบบการขายสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การควบคุมการปิดเปิดเครื่องอ่านสัญญาณ

บล็อกไดอะแกรมของส่วนควบคุมการปิดเปิดเครื่องอ่านสัญญาณ

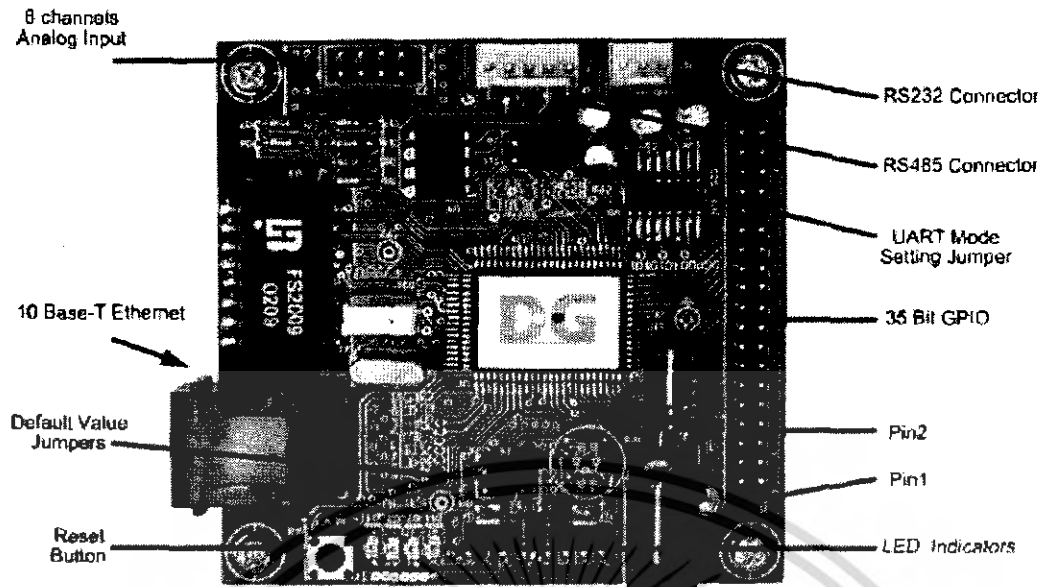


รูปที่ 3.16 บล็อกไดอะแกรมของส่วนควบคุมการปิดเปิดเครื่องอ่านสัญญาณ

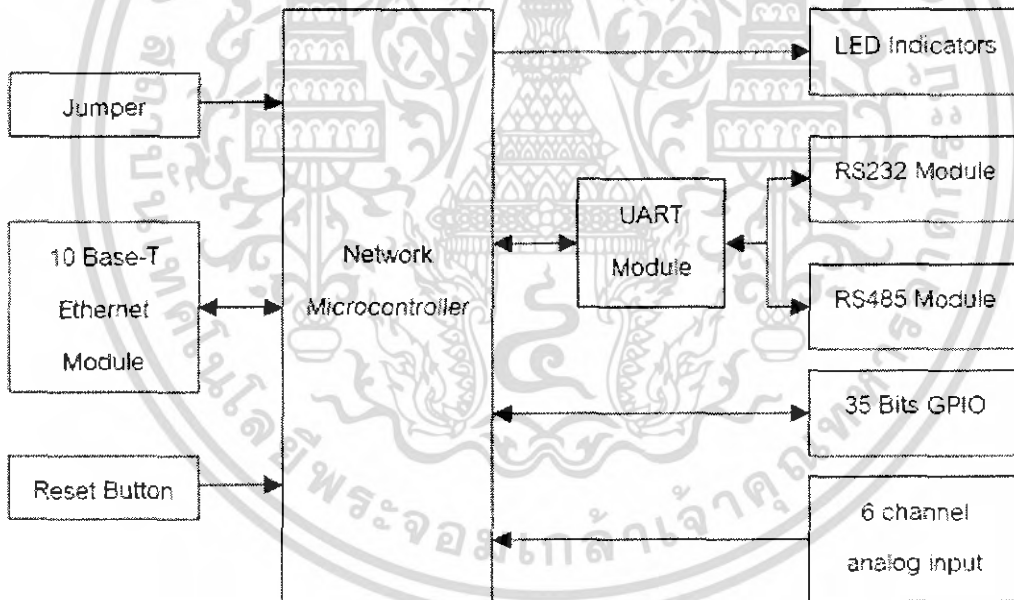
#### บอร์ด Ethernet IO

การใช้บอร์ด Ethernet IO เพื่อควบคุมควมโบโครคอนโทรลเลอร์นั้น เราจะใช้ภาษา C++ ในการโปรแกรมควบคุมการทำงานของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 บอร์ดอีเทอร์เน็ต ไอ / โอ



รูปที่ 3.18 บล็อกไดอะแกรมของบอร์ดอีเทอร์เน็ต ไอ / โอ

### 3.4.1 รายละเอียดต่างๆของบอร์ด อีเทอร์เน็ต ไอ / โอ

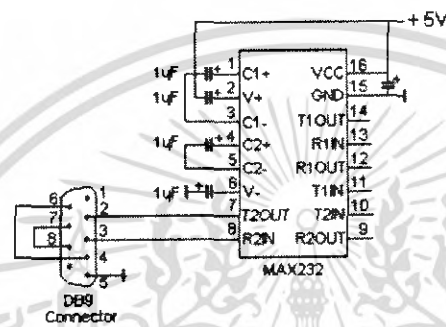
#### 3.4.1.1 ส่วนที่ติดตั้งติดต่อกับอีเทอร์เน็ต แบบ 10 Base-T

3.4.1.2 ส่วนที่ใช้ติดต่อกับส่วนของ UART โดยในส่วนนี้นั้นสามารถใช้ติดต่อกับส่วนของ RS232 หรือ RS485 ได้โดยที่สามารถเลือกใช้ได้เพียงอย่างใดอย่างหนึ่ง ไม่สามารถใช้พร้อมกันทั้งสองตัวได้ โดยการเลือกใช้งานตัวใดตัวหนึ่งนั้นสามารถตั้งค่าได้ที่ Jumper Module ซึ่งในโปรเจกต์นี้เราจะใช้การส่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลผ่านทางพอร์ต RS232 ซึ่งในส่วนนี้จะมีหน้าที่ไว้ใช้ในการสื่อสารระหว่างตัวบอร์ดกับส่วนของไมโครคอนโทรลเลอร์ โดยมีการใช้งานภายในบอร์ดตามตาราง

J13 Pin #	Signal Name	Description
1	TX	Transmitted data from Ethernet IO board
2	RX	Received data to Ethernet IO board
3	Gnd	Ground

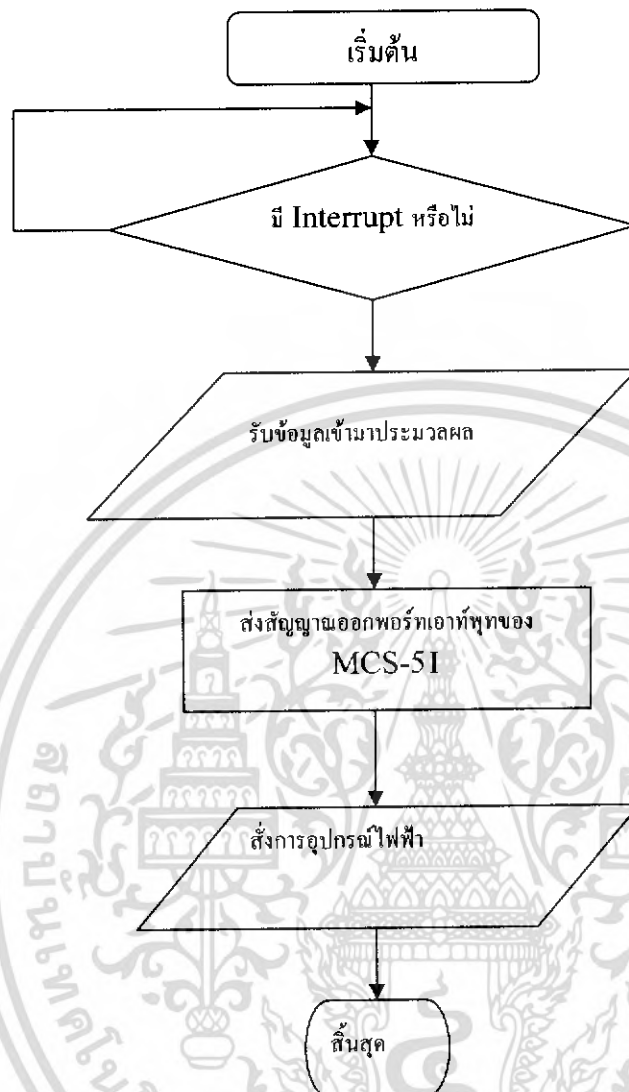
ตารางที่ 3.12 การใช้งานการสื่อสาร โดยใช้ RS232



รูปที่ 3.19 วงจรเชื่อมต่อ MAX232 หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

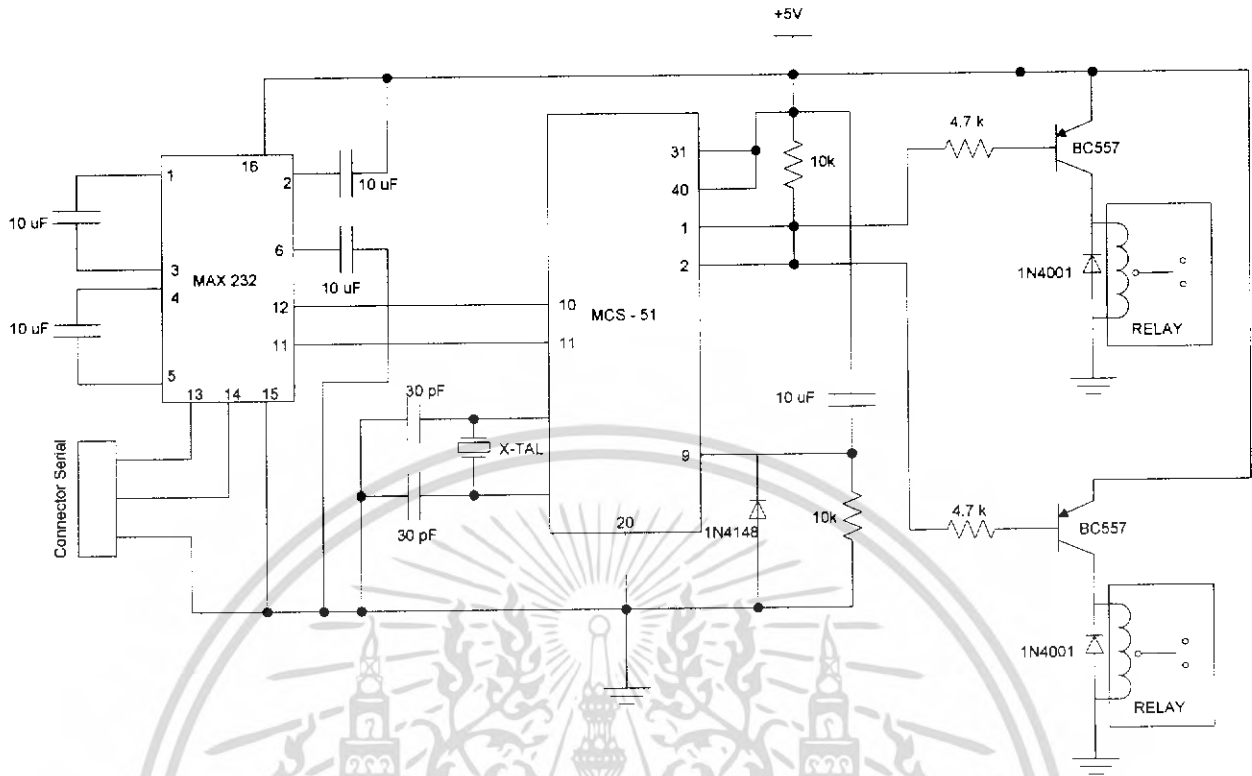
### ผังการทำงานของไมโครคอนโทรลเลอร์



รูปที่ 3.20 แสดงการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### วงจรในส่วนควบคุมไมโครคอนโทรลเลอร์



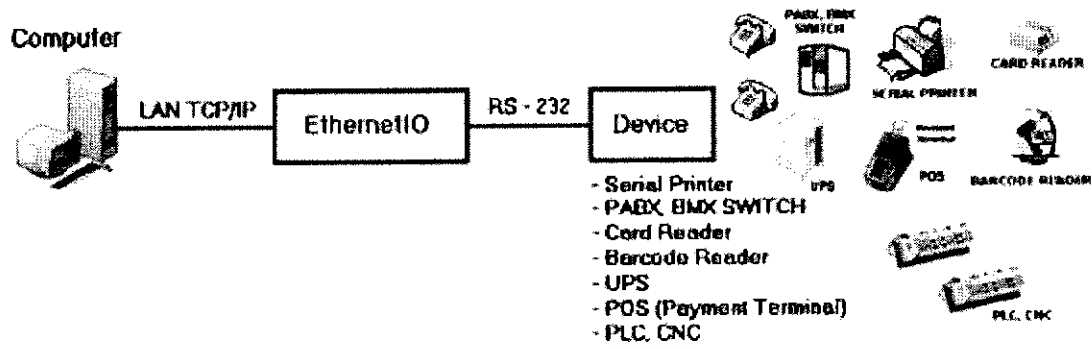
รูปที่ 3.21 แสดงวงจรในส่วนควบคุมไมโครคอนโทรลเลอร์

### การโปรแกรมควบคุมการทำงานของบอร์ด Ethernet IO

ในการโปรแกรมการทำงานของบอร์ด Ethernet IO นี้ จะใช้ภาษา VC++ มนการพัฒนาโปรแกรม ในการควบคุม โดยการเรียกใช้ผ่าน DLL ซึ่งใช้โปรแกรม VC++ ในการเรียกใช้เพื่อจัดการควบคุมการใช้งานของ UART, GPIO, ADC โดยเราจะสามารถกำหนดค่าต่างๆก่อนที่จะรับส่งข้อมูลผ่านพอร์ตของบอร์ดได้ ไม่ว่าจะเป็นค่า Baud rate, Data bit, Stop bit, Parity bit และ (Network setting) IP, Subnet, Gateway, MAC Address ซึ่งสามารถเขียนขึ้นมาเพื่อใช้งานใหม่ได้เองทุกอย่างโดยทางผู้ผลิตได้จัดเตรียม DLL Function เพื่อความสะดวกในการโปรแกรมควบคุม

ในการใช้งานของบอร์ดเราสามารถใช้งานในรูปแบบของ Virtual Comport ได้ จะทำงานโดยการจำลองตัวเองให้เป็น Comport ของ PC โดยสามารถกำหนดการใช้งานได้ตั้งแต่ Comport 2-20 และสามารถกำหนดค่าไอพีแอดเดรสได้เพื่อที่จะใช้ในการส่งแบบ TCP/IP โดยสามารถนำไปต่อใช้งานกับอุปกรณ์ต่างๆ ที่ใช้การเชื่อมต่อเข้ากับ Comport ของ PC โดยเปลี่ยนมาเชื่อมต่อผ่าน UART ของ Ethernet IO แทน ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 การประยุกต์ใช้งานของการ์ด Ethernet IO



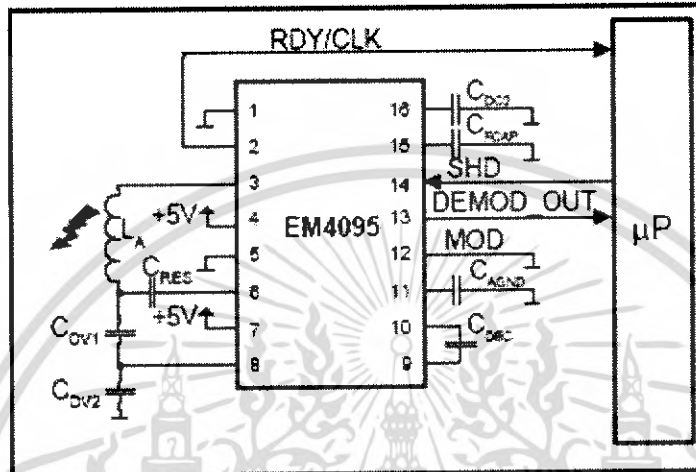
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

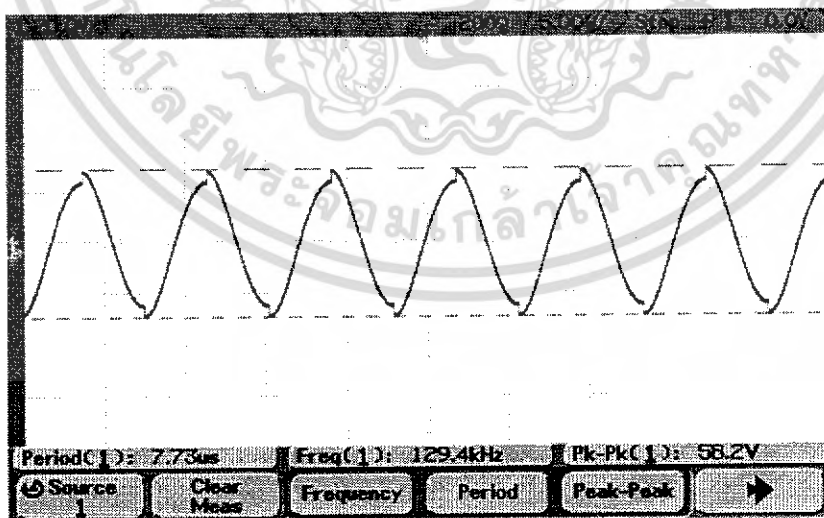
#### 4.1 การวัดสัญญาณของเครื่องอ่านสัญญาณ

##### 4.1.1 สัญญาณ ของวงจรของไอซี EM4095 ณะไม่มีแทกส์



รูปที่ 4.1 วงจร ไอซี EM4095

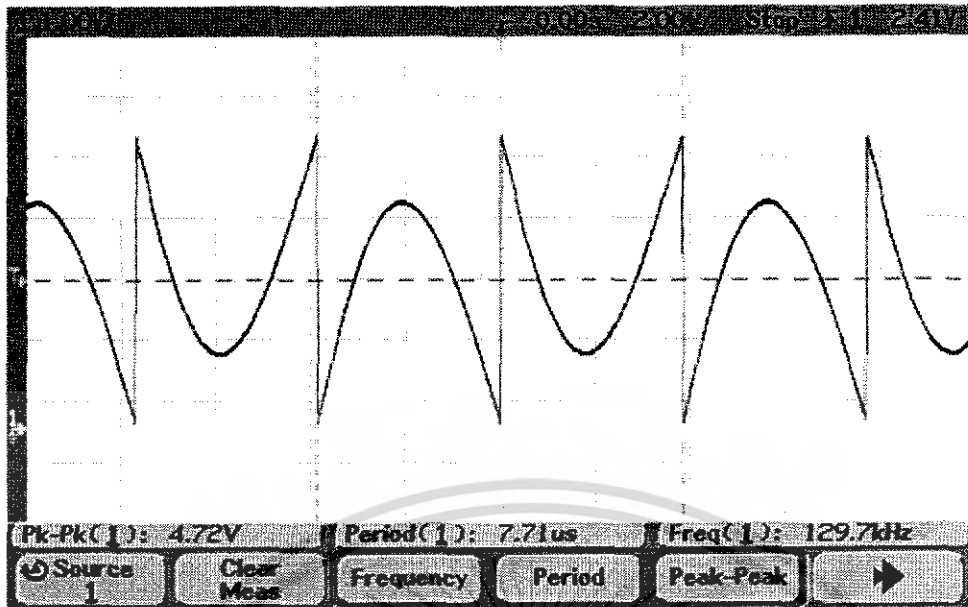
จากการวัดความถี่และขนาดของสัญญาณที่สายอากาศขาที่ 1 ซึ่งเป็นขาที่ 3 และสายอากาศขาที่ 2 ซึ่งเป็นขาที่ 6 ของวงจร ไอซี EM4095 รวมทั้งขา DEMOD\_OUT ซึ่งเป็นขาที่ 13 ของไอซี EM4095 ในรูปที่ 4.1 จะได้ผลตามรูปด้านล่างดังนี้



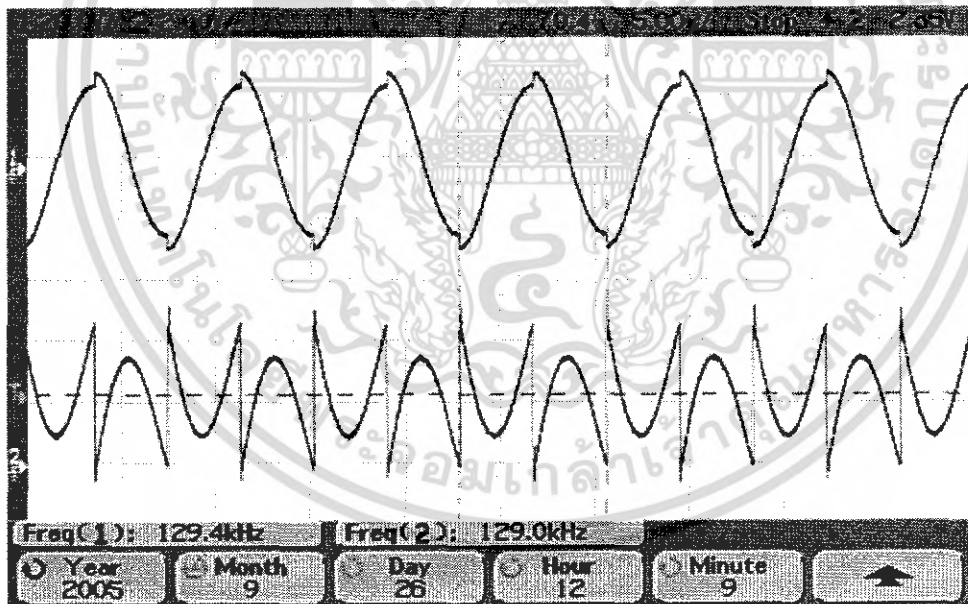
รูปที่ 4.2 ความถี่และขนาดของสัญญาณที่สายอากาศขาที่ 1 ของวงจรของไอซี EM4095

ในรูปที่ 4.2 เป็นรูปความถี่และขนาดของสัญญาณจากสายอากาศขาที่ 1 ซึ่งเป็นขาที่ 3 ที่วัดได้จากวงจรในรูปที่ 4.1 ซึ่งจะมีค่าเท่ากับ 129.4 kHz และขนาดของสัญญาณเท่ากับ 58.2 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

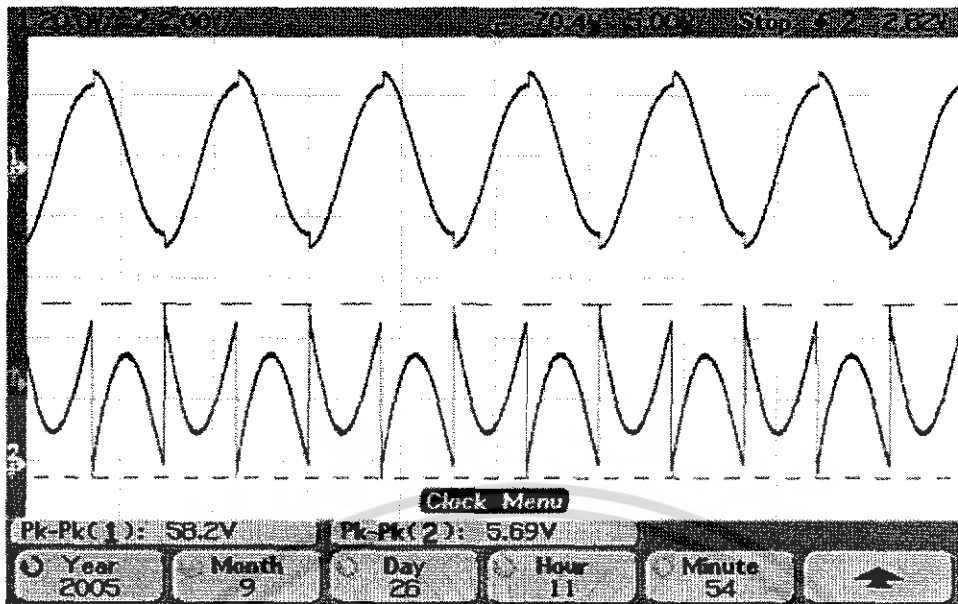


รูปที่ 4.3 ความถี่และขนาดของสัญญาณที่สายอากาศขาที่ 2 ของวงจรของไอซี EM4095  
 ในรูปที่ 4.3 เป็นรูปความถี่และขนาดของสัญญาณจากสายอากาศขาที่ 2 ซึ่งเป็นขาที่ 6 ที่วัดได้จากวงจรใน  
 รูปที่ 4.1 ซึ่งจะมีความถี่เท่ากับ 129.7 kHz และขนาดของสัญญาณเท่ากับ 4.72 V



รูปที่ 4.4 ความถี่ของสัญญาณที่สายอากาศขาที่ 1 และ 2 เปรียบเทียบกันของวงจรของไอซี EM4095  
 ในรูปที่ 4.4 เป็นรูปความถี่ของสัญญาณจากสายอากาศขาที่ 1 และ 2 เปรียบเทียบกัน ซึ่งจะมีความถี่เท่ากับ  
 129.4 kHz และ 129.0 kHz ตามลำดับ

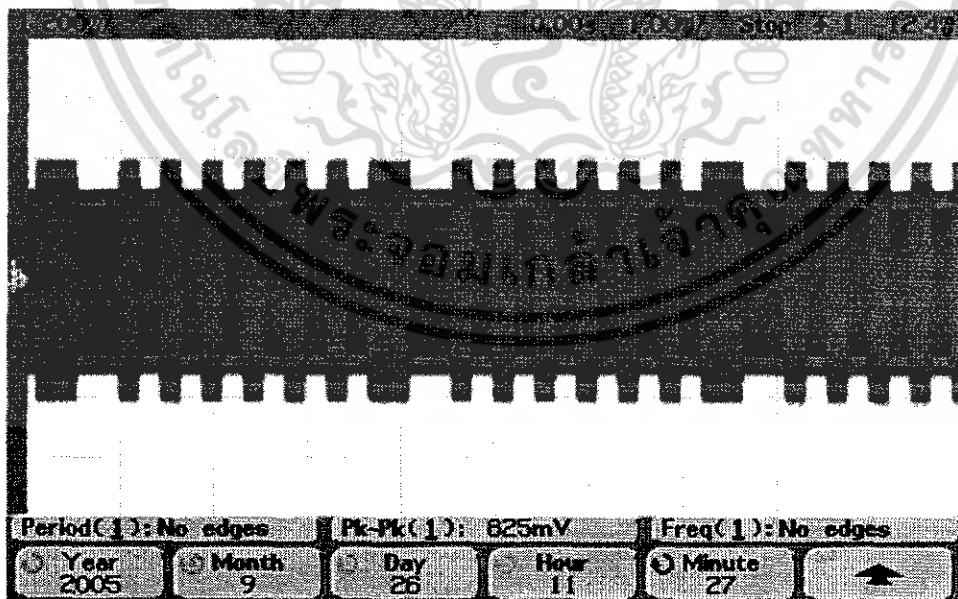
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ขนาดของสัญญาณที่สายอากาศขาที่ 1 และ 2 เปรียบเทียบกันของวงจรของไอซี EM4095 ในรูปที่ 4.5 เป็นรูปขนาดของสัญญาณจากสายอากาศขาที่ 1 และ 2 เปรียบเทียบกัน ซึ่งจะมีความถี่เท่ากับ 58.2 V และ 5.69 V ตามลำดับ

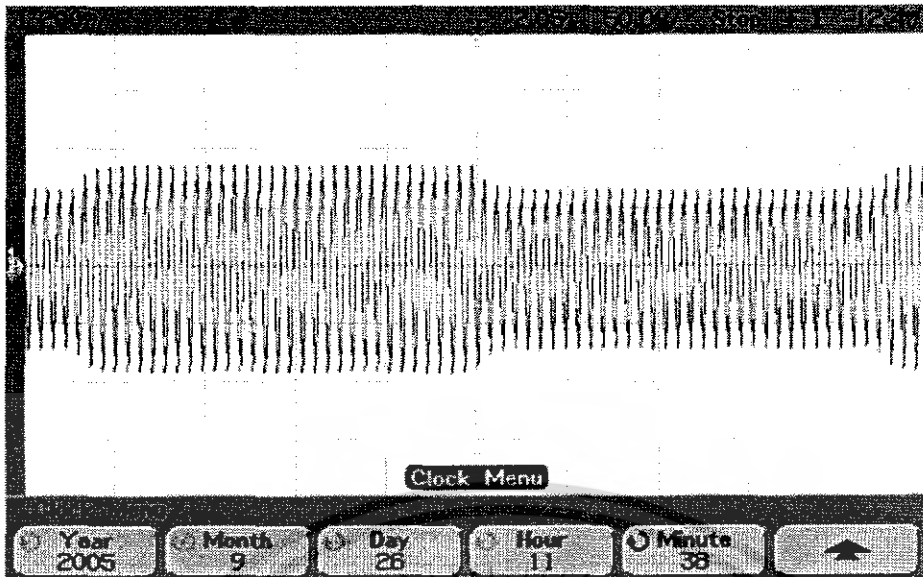
#### 4.1.2 สัญญาณ ของวงจรของไอซี EM4095ขณะมีแทกส์

เป็นการวัดความถี่และขนาดของสัญญาณของสายอากาศขาที่ 1 ของไอซี EM4095 ขณะมีแทกส์ที่อยู่ ในระหว่างการอ่านสัญญาณจะได้ผลดังรูปด้านล่าง



รูปที่ 4.6 สัญญาณที่สายอากาศขาที่ 1 ของวงจรของไอซี EM4095 ขณะมีแทกส์

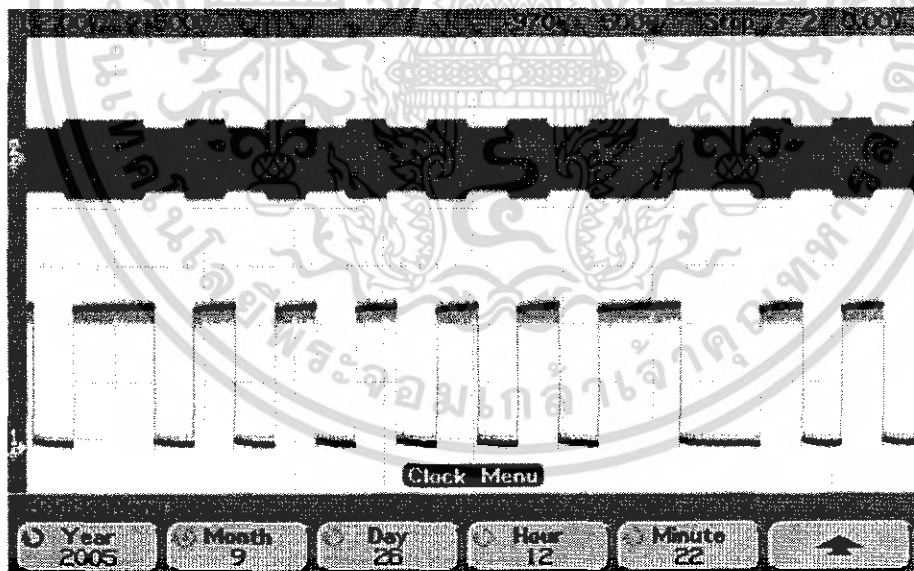
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 สัญญาณที่สายอากาศขาที่ 1 เมื่อนำมาขยายออกของวงจรของไอซี EM4095

#### 4.1.3 สัญญาณที่ผ่านวงจรของไอซี EM4095 แล้ว

เป็นการวัดสัญญาณของสายอากาศขาที่ 1 ที่รับเข้ามาจากเทกส์และสัญญาณที่ขา DEMOD\_OUT ซึ่งผ่านวงจรของไอซี EM4095 ขณะมีเทกส์ที่อยู่ในระยะการอ่านสัญญาณ จะได้สัญญาณดังรูปด้านล่าง

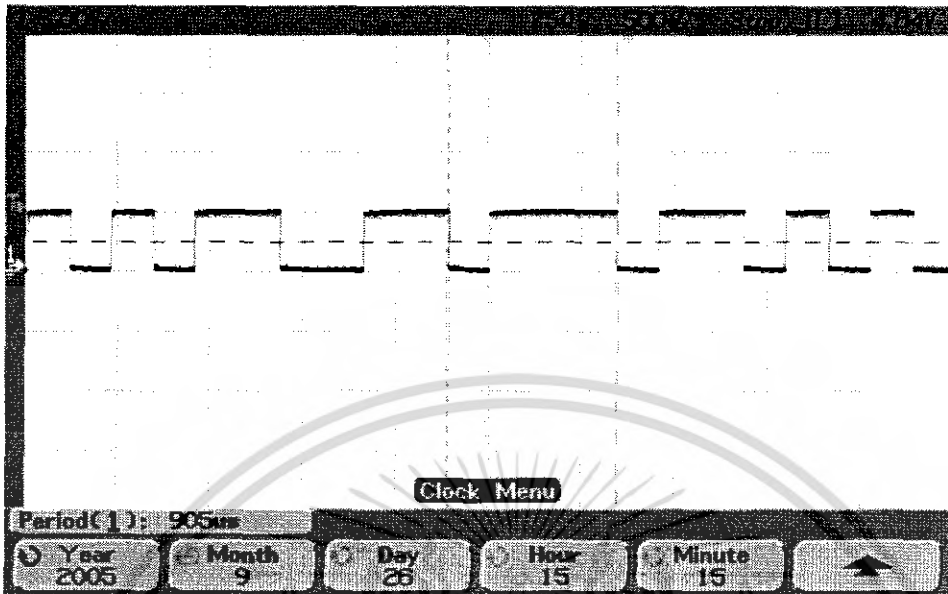


รูปที่ 4.8 สัญญาณที่เปรียบสัญญาณก่อนที่จะเข้าไอซี EM4095 ที่วัดจากสายอากาศที่ 1 และหลังจากที่ผ่านไอซี EM4095 ซึ่งวัดจากขา DEMOD\_OUT

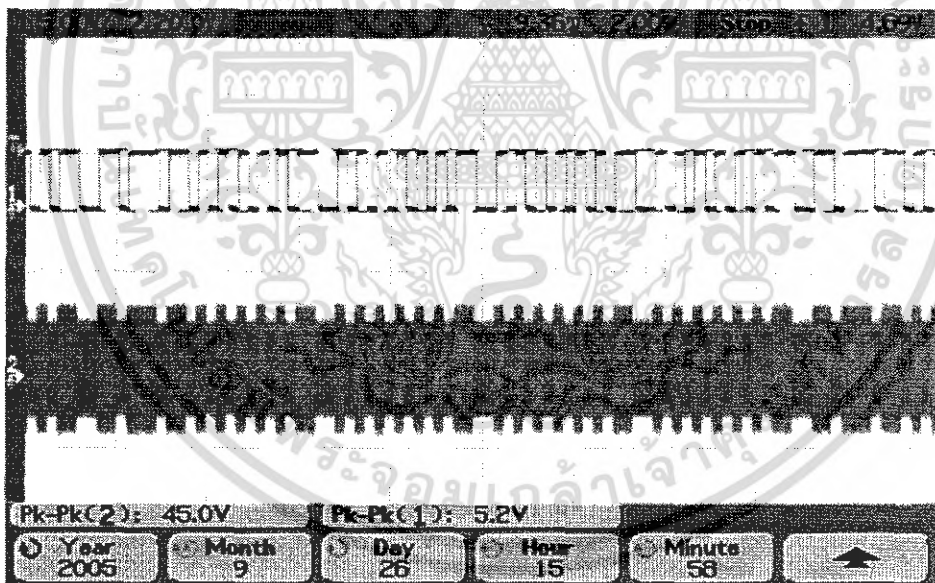
ในรูปที่ 4.8 แสดงการเปรียบเทียบสัญญาณจากสายอากาศที่ 1 ที่ผ่านวงจรไอซี EM 4095 แล้วสัญญาณที่วัดได้ด้านล่างจะวัดสัญญาณจากขาที่ 13 ซึ่งเป็นขา DEMOD\_OUT ของวงจรไอซี EM 4095

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.4 การวัดสัญญาณส่วนหัวของข้อมูล



รูปที่ 4.9 ส่วนหัวของสัญญาณที่วัดจากสายอากาศสาขาที่ 1 ของวงจรของไอซี EM4095



รูปที่ 4.10 การเปรียบเทียบสัญญาณส่วนหัว ก่อนเข้าและหลังเข้าวงจรของไอซี EM4095 จะเห็นได้ว่าขนาดของส่วนหัวของข้อมูลนั้นมีขนาดกว้างกว่าส่วนของข้อมูล ดังจะเห็นได้ในรูปที่ 4.9 และ 4.10 ซึ่งจะเป็นรูปที่แสดงให้เห็นถึงช่วงของข้อมูลที่จะเห็นส่วนของหัวของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2 ขั้นตอนการทำงานในส่วนของการซื้อขายสินค้า

ในส่วนนี้จะมีการเพิ่มเติมคอมพิวเตอร์บางตัวในโปรแกรม Delphi คือ คอมพิวเตอร์ Asyn 32 ใช้เพื่อติดต่อระหว่างตัวReader กับ Computer สามารถแสดงทำงานของโปรแกรมได้ดังนี้

##### 1. ส่วนที่ทำการติดต่อกับฐานข้อมูลกับระบบซื้อ-ขายสินค้าโดยใช้โปรแกรมDelphi

ในส่วนนี้จะเป็นการเพิ่มสินค้า และ ลบสินค้าออกจากฐานข้อมูลเมื่อมีการซื้อสินค้าเข้ามาในร้าน และมีลูกค้าซื้อสินค้า โดยมีขั้นตอนดังต่อไปนี้

1.1 เมื่อลูกค้าใช้บริการ โดยพนักงานจะทำการอ่านข้อมูลสินค้า(tag)ของลูกค้าลงยังฟอร์มขายสินค้าดังรูปที่ สินค้าที่ลูกค้าซื้อจะถูกนำมาลบกับจำนวนสินค้าในฐานข้อมูล และทำการบันทึกข้อมูลผู้ใช้บริการลงในฐานข้อมูล และฟอร์มนี้สามารถเชื่อมโยงไปยังฟอร์มต่างๆ ได้อีกด้วย

The screenshot shows a Delphi application window titled "MainForm" with a menu bar (File, View, Help) and a toolbar with buttons for "Add Goods", "Browse", "Parts", and "Close". The main area is titled "KMITL BOOK" and contains the following fields and controls:

- Date: 1/21/2006
- Bill To: Klamakpom kamthenthong
- CustNo: 1
- PartNo: 3
- Description: Follow Me Book3
- SoldBy: Pitchaya Worasil
- Payment Method: Cash

A table displays the items being purchased:

PartNo	Description	SellPrice	Qty	ExtPrice
1	Follow Me Book1	350	1	350
2	Follow Me Book2	175	1	175
3	Follow Me Book3	320	1	320
*	Change--55	0	1	0

Below the table, there are navigation buttons (left, right, plus, minus, refresh) and a "Buy" button. To the right, a summary shows: Total 845 BAHT, Cash 900 BAHT, and Change 55 BAHT. A "Change--55" dialog box is open over the table, with an "OK" button.

รูปที่ 4.11 แสดงฟอร์มการขายสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 แสดงฟอร์มการเพิ่มสินค้าในร้าน โดยสินค้าที่เพิ่มจะนำมาบวกกับจำนวนสินค้าใน  
ฐานข้อมูล

**Add Information**

Add Goods

Follow Me Book1       American Headway STARTER B

Follow Me Book2       ENGLISH FILE Intermediate student's book

Follow Me Book3       คู่มือทบทวน MySQL สำหรับบริษัท

Follow Me Book4       ตามตาม Visual Basic

PHOTOSHOP RETOUCHING       คู่มือการใช้งานโปรแกรม JAVA ฉบับผู้เขียน

PartNo:  Qty:

Description:

PartNo	Description	OnHand
2	Follow Me Book2	99
1	Follow Me Book1	101
6	American Headway STARTER B	100

รูปที่ 4.12 แสดงฟอร์มการเพิ่มสินค้าลงในฐานข้อมูล

1.3 แสดงฟอร์มแสดงรายละเอียดของลูกค้า โดยบอกวันที่ซื้อสินค้า พนักงานที่ขายสินค้า และราคา

**Order By Customer**

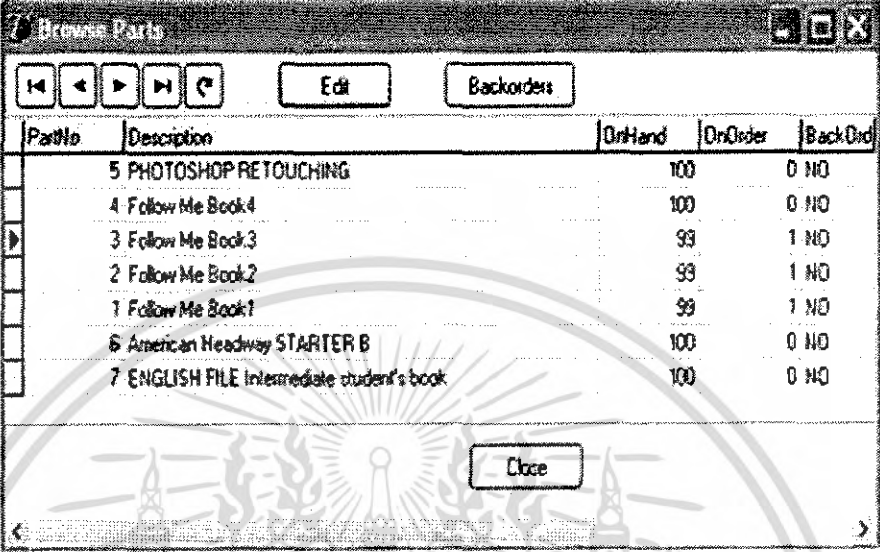
CustNo	Company	Phone
2	Pornip Aramrattana	03-5511-287
1	Kamalapon Kamthenthong	0-6328-8937
0	Tanaporn bortsud	0-3537-9029

OrderNo	SaleDate	FirstName	LastName	ItemsTotal
16	01/19/06	Jeabyu	Peysai	845

รูปที่ 4.13 แสดงฟอร์มรายละเอียดของลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

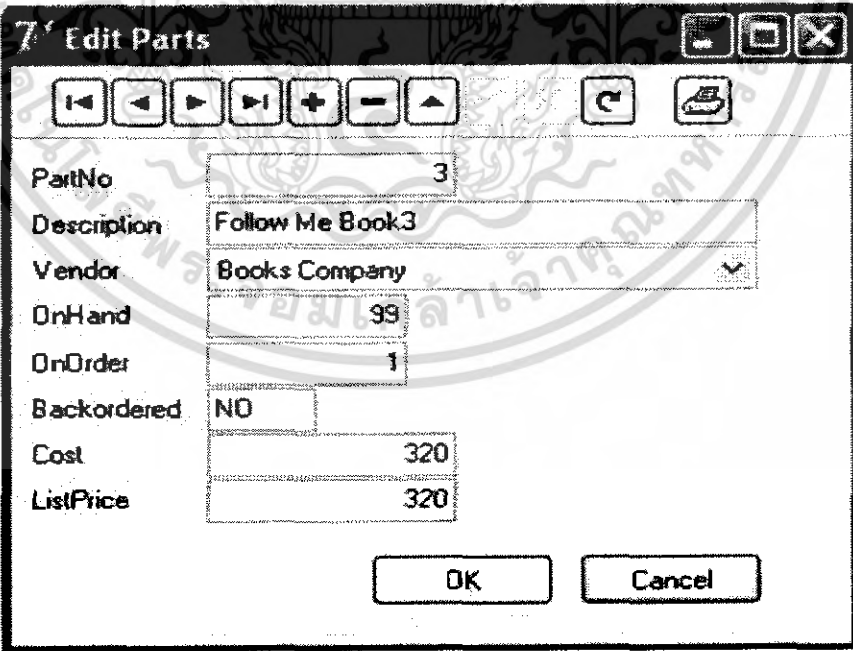
- 1.4 แสดงฟอร์มข้อมูลของสินค้าในฐานข้อมูล โดยแสดงถึงจำนวนสินค้าในคลัง (OnHand) และจำนวนสินค้าที่ขาย(OnOrder)



PartNo	Description	OnHand	OnOrder	BackOrd
5	PHOTOSHOP RETOUCHING	100	0	NO
4	Follow Me Book4	100	0	NO
3	Follow Me Book3	99	1	NO
2	Follow Me Book2	99	1	NO
1	Follow Me Book1	99	1	NO
6	American Headway STARTER B	100	0	NO
7	ENGLISH FILE Intermediate student's book	100	0	NO

รูปที่ 4.14 แสดงฟอร์มข้อมูลของสินค้า

- 1.5 แสดงฟอร์มการแก้ไขข้อมูลสินค้า



7 Edit Parts

PartNo: 3

Description: Follow Me Book3

Vendor: Books Company

OnHand: 99

OnOrder: 1

Backordered: NO

Cost: 320

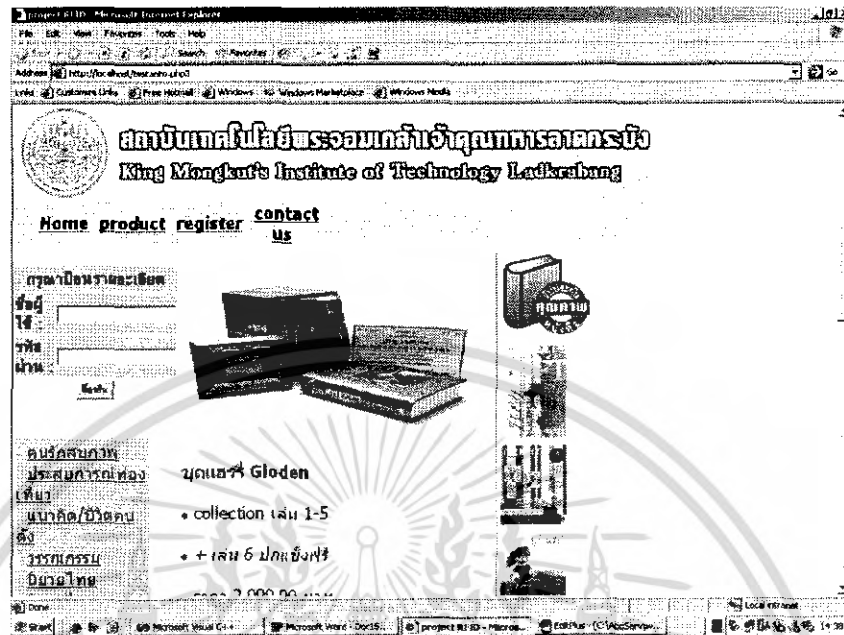
ListPrice: 320

OK Cancel

รูปที่ 4.15 แสดงฟอร์มการแก้ไขข้อมูลสินค้า

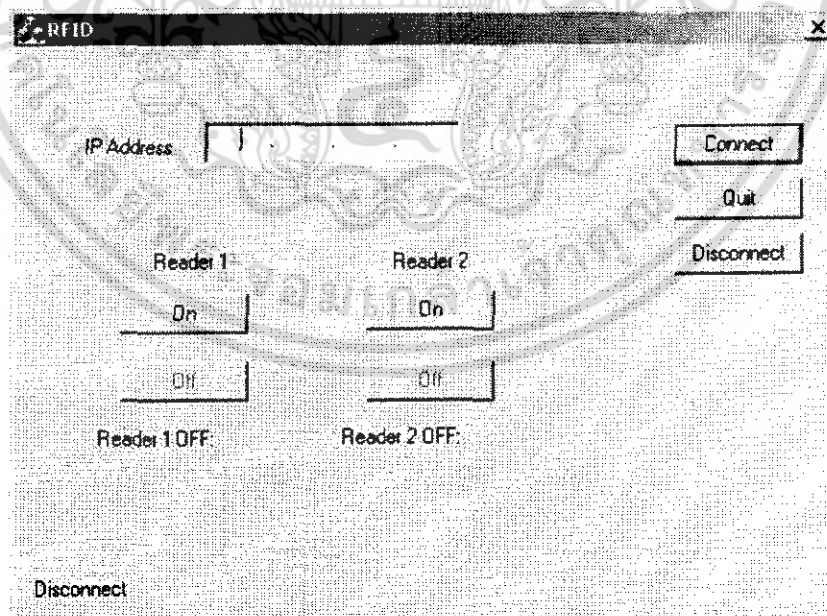
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.3 แสดงรายการสินค้าผ่านเว็บไซต์



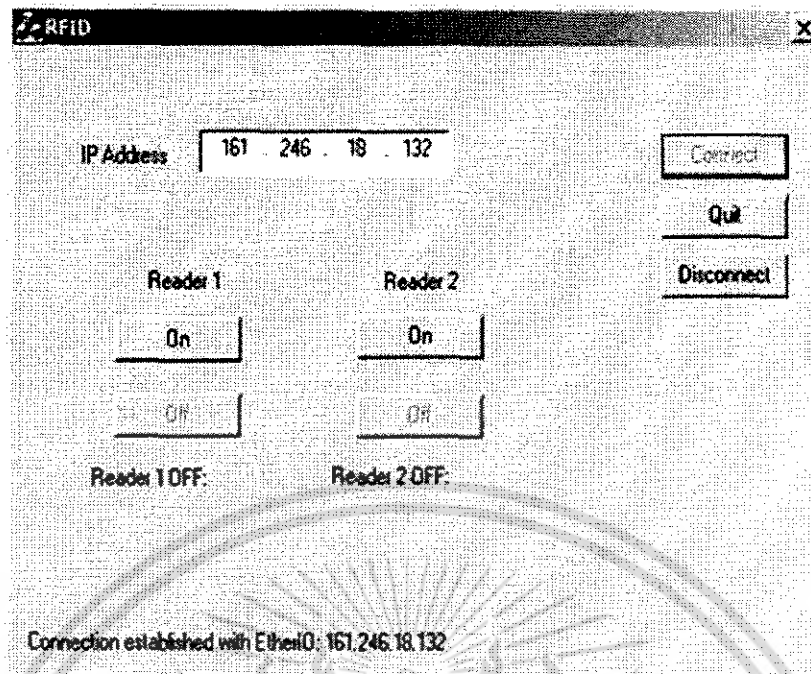
รูปที่ 4.16 แสดงรายการสินค้าผ่านเว็บไซต์

#### 4.4 ผลการทดลองส่วนควบคุมการปิดเปิดผ่านเครือข่าย

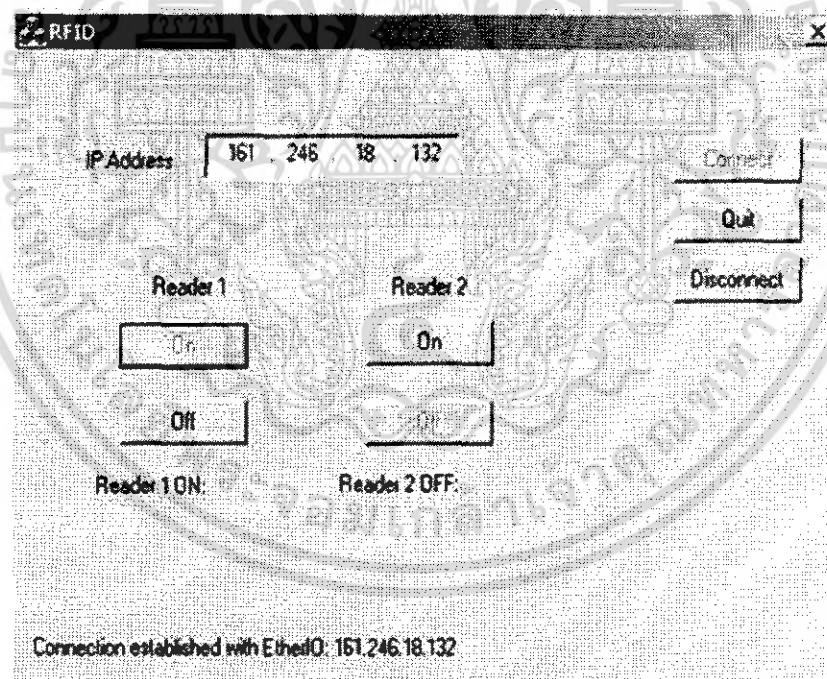


รูปที่ 4.17 หน้าต่างควบคุมอุปกรณ์เครื่องอ่านสัญญาณ เมื่อทำการรัน โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

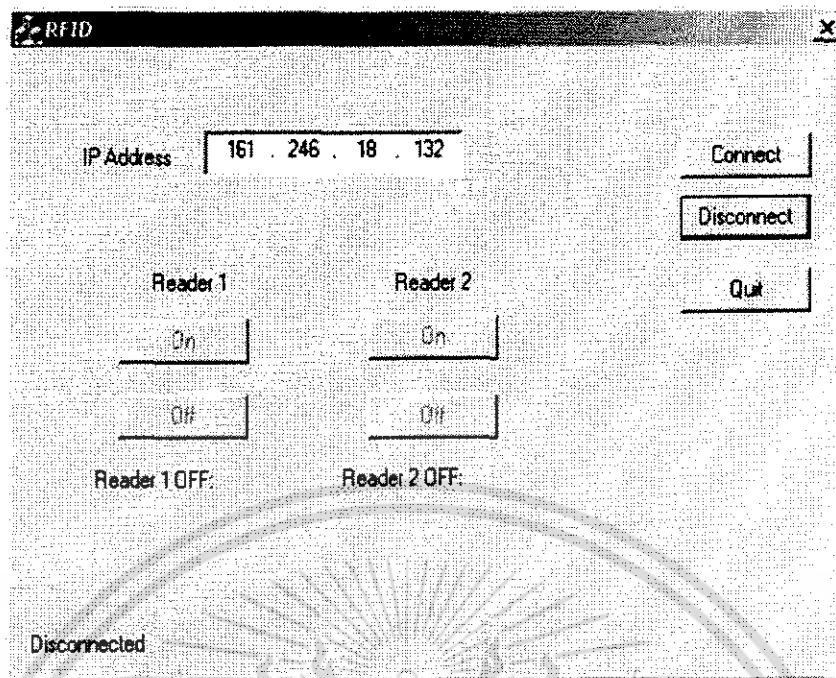


รูปที่ 4.18 หน้าต่างควบคุมอุปกรณ์เครื่องอ่านสัญญาณ เมื่อทำการเชื่อมต่อกับบอร์ด



รูปที่ 4.19 หน้าต่างโปรแกรมที่สามารถควบคุมเครื่องอ่านสัญญาณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 หน้าต่างโปรแกรมที่ถูกหยุดการเชื่อมต่อเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5 บทสรุปและวิจารณ์

### 5.1 คุณสมบัติของโครงการ

1. ระบบซื้อขายสินค้าสามารถใช้งาน เพื่อตรวจสอบข้อมูลสินค้า ข้อมูลลูกค้า และข้อมูลของบริษัทผู้ผลิต
2. ระบบซื้อขายสินค้าสามารถเพิ่มจำนวนสินค้าและลดจำนวนสินค้าเมื่อมีการซื้อขายสินค้า
3. สามารถดูรายการสินค้าผ่านทางเวปไซค์ได้
4. สามารถควบคุมการเปิดและการปิดเครื่องอ่านสัญญาณผ่านเครือข่ายได้

### 5.2 ปัญหาในการทำโครงการ

1. เครื่องอ่านสัญญาณยังอ่านได้ในระยะสั้นซึ่งและอ่านได้ทีละแทกต์
2. เวลาในการอ่านข้อมูลระหว่างเครื่องอ่านสัญญาณกับแทกต์ใช้เวลาานทำให้เวลาป้อนข้อมูลลงแทกต์และอ่านข้อมูลจากแทกต์ต้องใช้เวลาาน

### 5.3 แนวทางในการพัฒนาต่อไป

1. พัฒนาเครื่องอ่านสัญญาณให้อ่านได้ในระยะที่อ่านได้ไกลขึ้น ซึ่งจะต้องพัฒนาให้ใช้ในย่านความถี่สูงขึ้น
2. พัฒนาระบบซื้อขายสินค้าให้สามารถใช้งานได้อย่างสมบูรณ์แบบ

## หนังสืออ้างอิง

- [1] สัจจะ จรัสรุ่งรวีร จักรพงษ์ สุขประเสริฐ, “คู่มือการสร้างแอปพลิเคชันด้วย Delphi 5.0 ฉบับสมบูรณ์”, อินโฟเควส, พิมพ์ครั้งที่ 2, 2543
- [2] นิรุช อำนวยศิลป์, “การสร้างเว็บเพจอย่างไรจึงจำกัด PHP เพื่อการประยุกต์ใช้งาน”, ซีเคเอสเอ็มเคียบ, พิมพ์ครั้งที่ 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Delphi

### MAIN FORM

```
unit main;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Grids, DBGrids, Mask, DBCtrls, ExtCtrls,
  XPMAN, Menus, DB, Buttons, VaClasses, VaComm, Async32, MemDS, DBAccess,
  MyAccess, ImgList;
type
  TDateOrder = (doMDY, doDMY, doYMD);
  TMainForm = class(TForm)
    MainPanel: TPanel;
    BrowseBtn: TSpeedButton;
    PartsBtn: TSpeedButton;
    CloseBtn: TSpeedButton;
    ActiveSource: TDataSource;
    MainMenu: TMainMenu;
    File1: TMenuItem;
    NewOrder2: TMenuItem;
    N1: TMenuItem;
    Exit1: TMenuItem;
    View1: TMenuItem;
    Order1: TMenuItem;
    PartInventory1: TMenuItem;
    Help1: TMenuItem;
    About1: TMenuItem;
    XPManifest1: TXPManifest;
    Panel: TPanel;
    DBGrid2: TDBGrid;
    SearchEd: TEdit;
    AddBtn: TSpeedButton;
    ImageList1: TImageList;
    AddGoods1: TMenuItem;
    Label18: TLabel;
    SaleDateEdit: TDBEdit;
    PopupCalBtn: TSpeedButton;
    Label1: TLabel;
    Label3: TLabel;
    DescriptionEdit: TDBEdit;
    PartsSource1: TDataSource;
    Label2: TLabel;
    TotalEdit: TDBEdit;
    Label4: TLabel;
    SoldByCombo: TDBLookupComboBox;
    PaymentMethod: TLabel;
    PaymentCombo: TDBComboBox;
    DBEditBtns: TDBNavigator;
    DBNavBtns: TDBNavigator;
    Label5: TLabel;
    CancelBtn: TBitBtn;
    OKBtn: TBitBtn;
    Edit6: TEdit;
    Label6: TLabel;
    ChangeEd: TEdit;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Comm1: TComm;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ReadBtn: TButton;
DBGrid1: TDBGrid;
Datasource: TDataSource;
Label12: TLabel;
CompanyCombo: TDBLookupComboBox;
SaveBtn: TBitBtn;
Label28: TLabel;
CustNoEdit: TDBEdit;
procedure NewOrder(Sender: TObject);
procedure BrowseCustOrd(Sender: TObject);
procedure BrowseParts(Sender: TObject);
procedure CloseApp(Sender: TObject);
procedure Comm1RxChar(Sender: TObject; Count: Integer);
procedure ReadBtnClick(Sender: TObject);
procedure WriteBtnClick(Sender: TObject);
procedure CloseClick(Sender: TObject);
procedure DBGrid2Enter(Sender: TObject);
procedure DBGrid2Exit(Sender: TObject);
procedure SearchEdChange(Sender: TObject);
procedure SearchEdKeyPress(Sender: TObject; var Key: Char);
procedure AddBtnClick(Sender: TObject);
procedure Aboutclick(Sender: TObject);
procedure PickDate(Sender: TObject);
procedure SaleDateEditKeyPress(Sender: TObject; var Key: Char);
procedure CancelBtnClick(Sender: TObject);
procedure OKBtnClick(Sender: TObject);
procedure SaveBtnClick(Sender: TObject);
private
  SrchFld: TField;
  { Private declarations }
public
  { Public declarations }
end;
var
  MainForm: TMainForm;
implementation
uses
  DataMod, { Data Module }
  BrCstOrd, { The Browse Orders by Customer form }
  BrParts, { The Browse Parts form }
  EdOrders, { The Edit Orders form }
  EdAdd, { The Add Goods form }
  QryCust, { The Specify Date Range form }
  IniFiles, { Delphi Unit for INI file support }
  PickInvc, { The invoice number selection dialog }
  about, {The about form}
  Pickdate{The Select a Date form};
{$R *.dfm}
function GetDateOrder(const DateFormat: string): TDateOrder;
var
  I: Integer;
begin
  Result := doMDY;
  I := 1;
  while I <= Length(DateFormat) do
  begin
    case Chr(Ord(DateFormat[I]) and $DF) of
      'Y': Result := doYMD;
      'M': Result := doMDY;
      'D': Result := doDMY;
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Inc(I); Continue;
    end;
    Exit;
end;
Result := doMDY;
end;
procedure TMainForm.BrowseCustOrd(Sender: TObject);
begin
    case GetDateOrder(ShortDateFormat) of
        doYMD: ShortDateFormat := 'yy/mm/dd';
        doMDY: ShortDateFormat := 'mm/dd/yy';
        doDMY: ShortDateFormat := 'dd/mm/yy';
    end;
    BrCustOrdForm.Show;
end;
procedure TMainForm.BrowseParts(Sender: TObject);
begin
    BrPartsForm.Show;
end;
procedure TMainForm.AddBtnClick(Sender: TObject);
begin
    comm1.Close;
    EdAddForm.ShowModal;
end;
procedure TMainForm.AboutClick(Sender: TObject);
begin
    AboutBox.ShowModal;
end;
procedure TMainForm.CloseApp(Sender: TObject);
begin
    close;
end;
procedure TMainForm.CommlRxChar(Sender: TObject; Count: Integer);
var
    Buffer: array[0..1024] of char;
    RF: string;
    Bytes, P: Integer;
begin
    Fillchar(Buffer, Sizeof(Buffer), 0);
    Bytes := Comml.Read(Buffer, Count);
    for P := 0 to Bytes - 1 do
        begin
            a:=a+1;
            RF:=IntToStr((ord(Buffer[P])));
            if(a=7) then
                searchEd.text:= IntToStr((ord(Buffer[P])));
            if(a>32)then
                a:=0;
        end;
    end;
end;
procedure TMainForm.ReadBtnClick(Sender: TObject);
var
    S1: string;
    Count: Integer;
begin
    comm1.Open;
    S1:=chr($FB)+chr($02)+chr($FF)+chr($06)+chr($FC);
    Count := Comml.Write(S1[1], Length(st1));
    searchEd.Text:='';
end;
procedure TMainForm.WriteBtnClick(Sender: TObject);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

var
  S: string;
  Count: Integer;
begin
  comml.Open;
  S:=chr($FB)+chr($04)+chr($01)+chr($00)+chr($01)+
    chr($00)+chr($00)+chr($FF)+chr($FC);
  Count := Comml.Write(S[1], Length(st));
end;
procedure TMainForm.CloseClick(Sender: TObject);
begin
  MastData.Orders.Close;
  Mastdata.Parts.Close;
  MastData.Orders.Open;
  MastData.Orders.Insert;
  MastData.Parts.Open;
end;
procedure TMainForm.DBGrid2Enter(Sender: TObject);
begin
  ActiveSource.Dataset := MastData.Items;
end;
procedure TMainForm.DBGrid2Exit(Sender: TObject);
begin
  ActiveSource.Dataset := MastData.Orders;
end;
procedure TMainForm.SearchEdChange(Sender: TObject);
begin
  SrchFld := Datasource.Dataset.FieldName('PartNo');
  if Datasource.Dataset.Locate('PartNo', SearchEd.Text,
    [loCaseInsensitive]) then
  begin
    if DBGrid2.SelectedField = MastData.ItemsPartNo then
    begin
      if MastData.ItemsPartNo.Value <> 0 then
        DBGrid1.SelectedField.Value := MastData.ItemsPartNo.Value;
      { start with current PartNo }
      begin
        MastData.Items.Edit;
        MastData.ItemsPartNo.Value:=MastData.PartsPartNo.Value;
        MastData.Items.Append;
      end;
    end;
  end;
end;
procedure TMainForm.SearchEdKeyPress(Sender: TObject; var Key: Char);
begin
  if Assigned(SrchFld) and (Key > ' ') and not
    (SrchFld.IsValidChar(Key)) then
  begin
    MessageBeep(0);
    Key := #0;
  end;
end;
procedure TMainForm.PickDate(Sender: TObject);
begin
  BrDateForm.Date := MastData.OrdersSaleDate.Value;    { start with
current date }
  if BrDateForm.ShowModal = mrOk then
  begin
    MastData.Orders.Edit;
    MastData.OrdersSaleDate.Value := BrDateForm.Date;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        SaleDateEdit.SelectAll;
    end;
end;
procedure TMainForm.SaleDateEditKeyPress(Sender: TObject; var Key:
Char);
begin
    if Key = ^J then
    begin
        PickDate(Sender);
        Key := #0;
    end;
end;
procedure TMainForm.CancelBtnClick(Sender: TObject);
begin
    MastData.OrdersAfterCancel(MastData.Orders);
end;
procedure TMainForm.OKBtnClick(Sender: TObject);
var f:Real;
begin
    f:=strToFloat(Edit6.Text)-Mastdata.OrdersItemsTotal.Value;
    ChangeEd.Text:= FloatToStr(f);
    showmessage('Change='+ChangeEd.Text);
end;
procedure TMainForm.SaveBtnClick(Sender: TObject);
begin
    MastData.Items.Delete;
    MastData.Orders.Post;
    MastData.Orders.Close;
    Mastdata.Parts.Close;
    MastData.Orders.Open;
    MastData.Orders.Insert;
    MastData.Parts.Open;
end;
end.

```

---

### **BROWSE CUSTOMER FORM**

```

unit BrCstOrd;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms, Dialogs, StdCtrls, Grids, DBGrids, DBCtrls, ExtCtrls, Buttons,
    DB;
type
    TBrCustOrdForm = class(TForm)
        CtrlsPanel: TPanel;
        ActivateBtn: TSpeedButton;
        DefineBtn: TSpeedButton;
        Bevell: TBevel;
        Navigator: TDBNavigator;
        CustPanel: TPanel;
        CustGrid: TDBGrid;
        OrdersPanel: TPanel;
        OrdersGrid: TDBGrid;
        BtnPanel: TPanel;
        CloseBtn: TButton;
        procedure CloseBtnClick(Sender: TObject);
        procedure CustGridEnter(Sender: TObject);
        procedure OrdersGridEnter(Sender: TObject);
        procedure SetQuery(Sender: TObject);
        procedure ActivateQuery(Sender: TObject);
        procedure FormShow(Sender: TObject);
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private
    FActiveSource: TDataSource;
    function GetCustNo: Double;
    function GetOrderNo: Double;
    procedure SetActiveSource(DataSource: TDataSource);
    procedure SetCustNo(NewCustNo: Double);
    procedure SetOrderNo(NewOrderNo: Double);
    property ActiveSource: TDataSource read FActiveSource write
SetActiveSource;
    { Private declarations }
public
    property CustNo: Double read GetCustNo write SetCustNo;
    property OrderNo: Double read GetOrderNo write SetOrderNo;
    { Public declarations }
end;
var
    BrCustOrdForm: TBrCustOrdForm;
implementation
uses QryCust, EdCust, Main, DataMod;
{$R *.dfm}
function TBrCustOrdForm.GetCustNo: Double;
begin
    Result := MastData.CustMasterSrc.Dataset.Fields[0].AsFloat;
end;
procedure TBrCustOrdForm.SetCustNo(NewCustNo: Double);
begin
    MastData.CustMasterSrc.Dataset := MastData.Cust;
    MastData.Cust.Locate('CustNo', NewCustNo, []);
end;
function TBrCustOrdForm.GetOrderNo: Double;
begin
    Result := MastData.OrdByCustOrderNo.Value;
end;
procedure TBrCustOrdForm.SetOrderNo(NewOrderNo: Double);
begin
    MastData.OrdByCust.Locate('OrderNo', NewOrderNo, []);
end;
procedure TBrCustOrdForm.SetActiveSource(DataSource: TDataSource);
begin
    FActiveSource := DataSource;
    Navigator.DataSource := FActiveSource;
end;
procedure TBrCustOrdForm.CloseBtnClick(Sender: TObject);
begin
    Close;
end;
procedure TBrCustOrdForm.CustGridEnter(Sender: TObject);
begin
    ActiveSource := MastData.CustMasterSrc;
    CustGrid.Options := CustGrid.Options + [dgAlwaysShowSelection];
    OrdersGrid.Options := OrdersGrid.Options - [dgAlwaysShowSelection];
end;
procedure TBrCustOrdForm.OrdersGridEnter(Sender: TObject);
begin
    ActiveSource := MastData.OrdByCustSrc;
    OrdersGrid.Options := OrdersGrid.Options + [dgAlwaysShowSelection];
    CustGrid.Options := CustGrid.Options - [dgAlwaysShowSelection];
end;
procedure TBrCustOrdForm.SetQuery(Sender: TObject);
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ActivateQuery(Self);
    end;
    procedure TBrCustOrdForm.ActivateQuery(Sender: TObject);
    begin
    if not ActivateBtn.Down then
        MastData.CustMasterSrc.Dataset := MastData.Cust
    else
        with MastData.CustQuery do
        try
            Close;
            Params[0].AsDatetime := QueryCustDlg.FromDate;
            Params[1].AsDatetime := QueryCustDlg.ToDate;
            Open;
            { Any records in the result set? }
            if BOF and EOF then Abort;
            MastData.CustMasterSrc.Dataset := MastData.CustQuery;
        except
            MastData.CustMasterSrc.Dataset := MastData.Cust;
            ActivateBtn.Down := false;
            ShowMessage('No matching records in the specified date range.');
```

#### **BROWSE PARTS FORM**

```

    end;
    procedure TBrCustOrdForm.FormShow(Sender: TObject);
    begin
        MastData.Cust.Open;
        MastData.Cust.First;
    end;
    end.
    -----
    unit BrParts;
    interface
    uses
        Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
        Forms, Dialogs, Grids, DBGrids, StdCtrls, DBCtrls, ExtCtrls, Buttons;
    type
        TBrPartsForm = class(TForm)
            Panel1: TPanel;
            ActivateBtn: TSpeedButton;
            Bevel1: TBevel;
            Navigator: TDBNavigator;
            EditBtn: TButton;
            Panel3: TPanel;
            CloseBtn: TButton;
            DBGrid1: TDBGrid;
            procedure CloseBtnClick(Sender: TObject);
            procedure ActiveQuery(Sender: TObject);
            procedure FormShow(Sender: TObject);
            procedure EditBtnClick(Sender: TObject);
        private
            function GetPartNo: Double;
            procedure SetPartNo(NewPartNo: Double);
            { Private declarations }
        public
            property PartNo: Double read GetPartNo write SetPartNo;
            { Public declarations }
        end;
    var
        BrPartsForm: TBrPartsForm;
    implementation
    uses EdParts, DataMod;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{$R *.dfm}
function TBrPartsForm.GetPartNo: Double;
begin
    Result := MastData.PartsSource.Dataset.Fields[0].AsFloat;
end;
procedure TBrPartsForm.SetPartNo(NewPartNo: Double);
begin
    MastData.PartsSource.Dataset := MastData.Parts;
    MastData.Parts.Locate('PartNo', NewPartNo, []);
end;
procedure TBrPartsForm.CloseBtnClick(Sender: TObject);
begin
    Close;
end;
procedure TBrPartsForm.ActiveQuery(Sender: TObject);
begin
    if not ActivateBtn.Down then
        MastData.PartsSource.Dataset := MastData.Parts
    else
        try
            MastData.PartsQuery.Close;
            MastData.PartsQuery.Open;
            MastData.PartsSource.Dataset := MastData.PartsQuery;
        except
            MastData.PartsSource.Dataset := MastData.Parts;
            raise;
        end;
end;
procedure TBrPartsForm.FormShow(Sender: TObject);
begin
    MastData.Parts.Open;
end;
procedure TBrPartsForm.EditBtnClick(Sender: TObject);
begin
    if ActivateBtn.Down then
        begin
            EdPartsForm.Edit(MastData.PartsQueryPartNo.Value);
            MastData.PartsQuery.Close;
            MastData.PartsQuery.Open;
        end
    else
        EdPartsForm.Edit(MastData.PartsPartNo.Value);
end;
end.

```

---

#### DATA MODULE

```

unit DataMod;
interface
uses
    Windows, Messages, SysUtils, Classes, DB, DBAccess, MyAccess, MemDS,
    Dialogs, Controls, frxClass, frxDBSet, DBTables;
type
    TMastData = class(TDataModule)
        PartsSource: TDataSource;
        VendorSource: TDataSource;
        OrdersSource: TDataSource;
        CustByOrdSrc: TDataSource;
        ItemsSource: TDataSource;
        CustSource: TDataSource;
        OrdByCustSrc: TDataSource;
        CustMasterSrc: TDataSource;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CustByCompSrc: TDataSource;
Orders: TMyTable;
OrdersOrderNo: TIntegerField;
OrdersCustNo: TIntegerField;
OrdersSaleDate: TDateTimeField;
OrdersShipDate: TDateTimeField;
OrdersEmpNo: TIntegerField;
OrdersPaymentMethod: TStringField;
OrdersItemsTotal: TFloatField;
OrdersTaxRate: TIntegerField;
OrdersFreight: TFloatField;
OrdersAmountPaid: TFloatField;
OrdersAmountDue: TFloatField;
OrdersTaxTotal: TFloatField;
CustByOrd: TMyTable;
CustByOrdCustNo: TIntegerField;
CustByOrdCompany: TStringField;
CustByOrdAddr1: TStringField;
CustByOrdAddr2: TStringField;
CustByOrdZip: TStringField;
CustByOrdPhone: TStringField;
CustByOrdFax: TStringField;
CustByOrdTaxRate: TIntegerField;
CustByOrdLastInvoiceDate: TDateTimeField;
Parts: TMyTable;
PartsPartNo: TIntegerField;
PartsVendorNo: TIntegerField;
PartsDescription: TStringField;
PartsOnHand: TIntegerField;
PartsOnOrder: TIntegerField;
PartsCost: TFloatField;
PartsListPrice: TFloatField;
PartsBackOrd: TBooleanField;
Vendors: TMyTable;
NextOrd: TMyTable;
NextOrdNewKey: TIntegerField;
NextCust: TMyTable;
NextCustNewCust: TIntegerField;
LastItemQuery: TMyQuery;
PartsQuery: TMyQuery;
PartsQueryPartNo: TIntegerField;
PartsQueryVendorNo: TIntegerField;
PartsQueryDescription: TStringField;
PartsQueryOnHand: TIntegerField;
PartsQueryOnOrder: TIntegerField;
PartsQueryCost: TFloatField;
PartsQueryListPrice: TFloatField;
PartsQueryBackOrd: TBooleanField;
Cust: TMyTable;
CustCustNo: TIntegerField;
CustCompany: TStringField;
CustAddr1: TStringField;
CustAddr2: TStringField;
CustZip: TStringField;
CustPhone: TStringField;
CustFax: TStringField;
CustTaxRate: TIntegerField;
CustLastInvoiceDate: TDateTimeField;
CustQuery: TMyQuery;
CustQueryCustNo: TIntegerField;
CustQueryCompany: TStringField;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CustQueryLastInvoiceDate: TDateTimeField;
CustQueryPhone: TStringField;
OrdByCust: TMyTable;
OrdByCustOrderNo: TIntegerField;
OrdByCustCustNo: TIntegerField;
OrdByCustSaleDate: TDateTimeField;
OrdByCustShipDate: TDateTimeField;
OrdByCustItemsTotal: TFloatField;
OrdByCustTaxRate: TIntegerField;
OrdByCustFreight: TFloatField;
OrdByCustAmountPaid: TFloatField;
OrdByCustAmountDue: TIntegerField;
CustByComp: TMyTable;
CustByLastInvQuery: TMyQuery;
CustByLastInvQueryCustNo: TIntegerField;
CustByLastInvQueryCompany: TStringField;
CustByLastInvQueryAddr1: TStringField;
CustByLastInvQueryAddr2: TStringField;
CustByLastInvQueryZip: TStringField;
CustByLastInvQueryPhone: TStringField;
CustByLastInvQueryFax: TStringField;
CustByLastInvQueryTaxRate: TIntegerField;
CustByLastInvQueryLastInvoiceDate: TDateTimeField;
MyConnection1: TMyConnection;
Items: TMyTable;
ItemsOrderNo: TIntegerField;
ItemsItemNo: TFloatField;
ItemsPartNo: TIntegerField;
ItemsQty: TIntegerField;
ItemsDiscount: TIntegerField;
ItemsExtPrice: TFloatField;
ItemsDescription: TStringField;
ItemsSellPrice: TFloatField;
OrdersByDateQuery: TMyQuery;
Emps: TMyTable;
EmpsEmpNo: TIntegerField;
EmpsLastName: TStringField;
EmpsFirstName: TStringField;
EmpsPhoneExt: TStringField;
EmpsHireDate: TDateTimeField;
EmpsSalary: TIntegerField;
EmpsFullName: TStringField;
EmpsSource: TDataSource;
OrdByCustFirstName: TStringField;
OrdByCustLastName: TStringField;
OrdByCustEmpNo: TIntegerField;
procedure OrdersAfterCancel(DataSet: TDataSet);
procedure OrdersAfterDelete(DataSet: TDataSet);
procedure OrdersAfterPost(DataSet: TDataSet);
procedure OrdersBeforeOpen(DataSet: TDataSet);
procedure OrdersBeforeEdit(DataSet: TDataSet);
procedure OrdersCustNoChange(Sender: TField);
procedure ItemsAfterPost(DataSet: TDataSet);
procedure EnsureOrdersEdit(DataSet: TDataSet);
procedure ItemsBeforeEdit(DataSet: TDataSet);
procedure ItemsBeforeOpen(DataSet: TDataSet);
procedure ItemsBeforePost(DataSet: TDataSet);
procedure ItemsCalcFields(DataSet: TDataSet);
procedure ItemsNewRecord(DataSet: TDataSet);
procedure EmpsCalcFields(DataSet: TDataSet);
procedure PartsBeforeOpen(DataSet: TDataSet);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure PartsCalcFields(DataSet: TDataSet);
procedure PartsUpdateError(DataSet: TDataSet; E: EDatabaseError;
  UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
procedure CustBeforeOpen(DataSet: TDataSet);
procedure CustBeforePost(DataSet: TDataSet);
procedure OrdersNewRecord(DataSet: TDataSet);
procedure OrdByCustCalcFields(DataSet: TDataSet);
procedure OrdersBeforeCancel(DataSet: TDataSet);
procedure OrdersBeforeClose(DataSet: TDataSet);
procedure OrdersBeforeDelete(DataSet: TDataSet);
procedure OrdersBeforeInsert(DataSet: TDataSet);
procedure EditUpdateError(DataSet: TDataSet; E: EDatabaseError;
  UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
procedure PartsQueryCalcFields(DataSet: TDataSet);
procedure ItemsAfterDelete(DataSet: TDataSet);
private
  PrevPartNo: Double;      { remembers Item's previous part# }
  PrevQty: Longint;        { remembers Item's previous qty }
  DeletingItems: Boolean;  { suppress totals calc. if deleting
items }
  FItemNo: Integer;
  procedure DeleteItems;
  procedure UpdateTotals;
  { Private declarations }
public
  function DataSetApplyUpdates(DataSet: TDataSet; Apply: Boolean):
Boolean;
  { Public declarations }
end;
function Confirm(Msg: string): Boolean;
var
  MastData: TMastData;
implementation
{$R *.dfm}
uses Variants;
function Confirm(Msg: string): Boolean;
begin
  Result := MessageDlg(Msg, mtConfirmation, mbYesNoCancel, 0) = mrYes;
end;
procedure TMastData.UpdateTotals;
var
  TempTotal: Extended;
begin
  if DeletingItems then Exit; { don't calculate if deleting all
items }
  try
    Items.DisableControls;
    Items.First;
    TempTotal := 0;           { use temp for efficiency }
    while not Items.EOF do
    begin
      TempTotal := TempTotal + ItemsExtPrice.Value;
      Items.Next;
    end;
    OrdersItemsTotal.Value := TempTotal;
  finally
    Items.EnableControls;
  end;
end;
procedure TMastData.OrdersAfterCancel(DataSet: TDataSet);
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Cust.CancelUpdates;
    Parts.CancelUpdates;
    Items.CancelUpdates;
    Orders.CancelUpdates;
end;
procedure TMasterData.OrdersAfterDelete(DataSet: TDataSet);
begin
    MyConnection1.ApplyUpdates;
end;
procedure TMasterData.OrdersAfterPost(DataSet: TDataSet);
begin
    if Cust.Locate('CustNo', OrdersCustNo.Value, []) and
        (CustLastInvoiceDate.Value < OrdersShipDate.Value) then
    begin
        Cust.Edit;
        CustLastInvoiceDate.Value := OrdersShipDate.Value;
        Cust.Post;
    end;
    MyConnection1.ApplyUpdates;
end;

procedure TMasterData.OrdersBeforeCancel(DataSet: TDataSet);
begin
    if (Orders.State = dsInsert) and not (Items.BOF and Items.EOF) then
        if not Confirm('Cancel order being inserted and delete all line
items?') then
            Abort;
end;
procedure TMasterData.OrdersBeforeEdit(DataSet: TDataSet);
begin
    LastItemQuery.Close;
    LastItemQuery.Open;
    { SQL servers return Null for some aggregates if no items are
present }
    with LastItemQuery.Fields[0] do
        if IsNull then FItemNo := 1
        else FItemNo := AsInteger + 1;
end;
procedure TMasterData.OrdersBeforeOpen(DataSet: TDataSet);
begin
    CustByComp.Open;
    CustByOrd.Open;
    Cust.Open;
    Emps.Open;
    Items.Open;
end;
procedure TMasterData.ItemsAfterPost(DataSet: TDataSet);
{Reduce/increase Parts table's OnOrder field }
procedure UpdateParts(PartNo: Double; Qty : Longint);
begin
    if (PartNo > 0) and (Qty <> 0) then
    try
        if not Parts.Locate('PartNo', PartNo, []) then Abort;
        Parts.Edit;
        PartsOnOrder.Value := PartsOnOrder.Value + Qty;
        PartsOnHand.Value:=PartsOnHand.Value-Qty;
        Parts.Post;
        {if PartsOnHand.Value < -1 then
        begin
            MessageDlg('No Goods.', mtInformation, [mbOK], 0);
            Parts.Delete;
        }
    }
end;

```

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end;}
    except
        on E: Exception do
            ShowMessage(Format('Error updating parts table for PartNo:
%d', [PartNo]));
        end;
    end;
begin
    { Maintain next available item number }
    Inc(FItemNo);
    UpdateTotals;
    if not ((PrevPartNo = ItemsPartNo.Value) and (PrevQty =
ItemsQty.Value)) then
        begin
            { Reduce previous Part#'s OnOrder field by previous Qty }
            UpdateParts(PrevPartNo, -PrevQty);
            { Increase new Part#'s OnOrder field by previous Qty }
            UpdateParts(ItemsPartNo.Value, ItemsQty.Value);
        end;
    end;
procedure TMasterData.EnsureOrdersEdit(DataSet: TDataSet);
begin
    Orders.Edit;
end;
procedure TMasterData.ItemsBeforeEdit(DataSet: TDataSet);
begin
    Orders.Edit;
    PrevPartNo := ItemsPartNo.Value;
    PrevQty := ItemsQty.Value;
end;
    procedure TMasterData.ItemsBeforeOpen(DataSet: TDataSet);
begin
    Parts.Open;
end;
procedure TMasterData.ItemsBeforePost(DataSet: TDataSet);
begin
    ItemsItemNo.Value := FItemNo;
end;
    { Lookup PartNo info for the item; calculate its extended price }
procedure TMasterData.ItemsCalcFields(DataSet: TDataSet);
begin
    ItemsExtPrice.Value := ItemsQty.Value *
        ItemsSellPrice.Value * (100 - ItemsDiscount.Value) / 100;
end;
procedure TMasterData.ItemsNewRecord(DataSet: TDataSet);
begin
    PrevPartNo := 0;
    PrevQty := 0;
    ItemsOrderNo.Value := OrdersOrderNo.Value;
    ItemsQty.Value := 1;
    ItemsDiscount.Value := 0;
end;
    { Concatenate last name + first name for the order's SoldBy
DBLookupCombo }
procedure TMasterData.EmpsCalcFields(DataSet: TDataSet);
begin
    EmpsFullName.Value := Format('%s, %s', [EmpsLastName.Value,
EmpsFirstName.Value]);
end;
procedure TMasterData.DeleteItems;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    DeletingItems := True;    { suppress recalc of totals during
delete }
    Items.DisableControls;   { for faster table traversal }
    try
        Items.First;
        while not Items.EOF do Items.Delete;
    finally
        DeletingItems := False;
        Items.EnableControls; { always re-enable controls after
disabling }
    end;
end;
procedure TMastData.PartsBeforeOpen(DataSet: TDataSet);
begin
    Vendors.Open;
end;
procedure TMastData.PartsCalcFields(DataSet: TDataSet);
begin
    PartsBackOrd.Value := PartsOnOrder.Value > PartsOnHand.Value;
end;
procedure TMastData.PartsUpdateError(DataSet: TDataSet; E:
EDatabaseError;
    UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
var
    Key: Variant;
const
    UpdErrMsg = '%s.#13#10'Discard the edits to %S %S and continue
updating?';
begin
    if UpdateKind = ukDelete then
        Key := Dataset.Fields[0].OldValue else
        Key := Dataset.Fields[0].NewValue;
    if MessageDlg(Format(UpdErrMsg, [E.Message,
DataSet.Fields[0].DisplayLabel, Key]),
mtConfirmation, [mbYes, mbCancel], 0) = mrYes then
        UpdateAction := uaSkip else
        UpdateAction := uaAbort;
end;
procedure TMastData.CustBeforeOpen(DataSet: TDataSet);
begin
    OrdByCust.Open;
end;
procedure TMastData.CustBeforePost(DataSet: TDataSet);
begin
    if Cust.State = dsInsert then
        with NextCust do
            begin
                Open;
                try
                    Edit;
                    CustCustNo.Value := NextCustNewCust.Value;
                    NextCustNewCust.Value := NextCustNewCust.Value + 1;
                    Post;
                finally
                    Close;
                end;
            end;
end;
end;
procedure TMastData.OrdersCustNoChange(Sender: TField);
var
    TaxRate: Variant;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  TaxRate := Cust.Lookup('CustNo', OrdersCustNo.Value, 'TaxRate');
  if not VarIsNull(TaxRate) then
    OrdersTaxRate.Value := TaxRate;
end;
procedure TMasterData.OrdersNewRecord(DataSet: TDataSet);
begin
  { Get the Next Order Value from the NextOrd Table }
  with NextOrd do
  begin
    Open;
    try
      Edit;
      OrdersOrderNo.Value := NextOrdNewKey.Value;
      NextOrdNewKey.Value := NextOrdNewKey.Value + 1;
      Post;
    finally
      Close;
    end;
  end;
  OrdersSaleDate.Value := Date;
  OrdersPaymentMethod.Value := 'Cash';
  OrdersItemsTotal.Value := 0;
  OrdersTaxRate.Value := 0;
end;
function TMasterData.DataSetApplyUpdates(DataSet: TDataSet; Apply:
Boolean): Boolean;
begin
  Result := True;
  with TDataSet(DataSet) do
  begin
    if (State in dsEditModes) then
    begin
      if Apply then
      begin
        Myconnection1.ApplyUpdates;
        { Always call CancelUpdates to remove any discard changes }
      end
      else
      begin
        if (MessageDlg('Unsaved changes, exit anyway?',
mtConfirmation,
          [mbYes, mbCancel], 0) = mrYes) then
          else
            Result := False;
        end;
      end;
    end;
  end;
end;

{ Determine the next available ItemNo for this order }
procedure TMasterData.OrdersBeforeClose(DataSet: TDataSet);
begin
  Items.Close;
  Emps.Close;
  CustByOrd.Close;
end;
procedure TMasterData.OrdersBeforeDelete(DataSet: TDataSet);
begin
  if not Confirm('Delete order and line items?') then
    Abort

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
  DeleteItems;
end;
procedure TMasterData.OrdersBeforeInsert(DataSet: TDataSet);
begin
  if Orders.State in dsEditMode then
  begin
    if Confirm('An order is being processed. Save changes and start
a new one?') then
      Orders.Post
    else
      Abort;
    end;
    FItemNo := 1;
  end;
procedure TMasterData.EditUpdateError(DataSet: TDataSet; E:
EDatabaseError;
  UpdateKind: TUpdateKind; var UpdateAction: TUpdateAction);
var
  Key: Variant;
const
  UpdErrMsg = '%s.'#13#10'Discard the edits to %S %S and continue
updating?';
begin
  if UpdateKind = ukDelete then
    Key := DataSet.Fields[0].OldValue else
    Key := DataSet.Fields[0].NewValue;
  if MessageDlg(Format(UpdErrMsg, [E.Message,
DataSet.Fields[0].DisplayLabel, Key]),
  mtConfirmation, [mbYes, mbCancel], 0) = mrYes then
    UpdateAction := uaSkip else
    UpdateAction := uaAbort;
end;
procedure TMasterData.PartsQueryCalcFields(DataSet: TDataSet);
begin
  PartsQueryBackOrd.Value := PartsOnOrder.Value > PartsOnHand.Value;
end;
procedure TMasterData.ItemsAfterDelete(DataSet: TDataSet);
begin
  UpdateTotals;
end;
end.

```

---

#### DATE FORM

```

unit Pickdate;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, XPMan, StdCtrls, Grids, Calendar, Buttons, ExtCtrls;
type
  TBrDateForm = class(TForm)
    Bevel1: TBevel;
    TitleLabel: TLabel;
    PrevMonthBtn: TSpeedButton;
    NextMonthBtn: TSpeedButton;
    Calendar1: TCalendar;
    OkBtn: TButton;
    CancelBtn: TButton;
    XPManifest1: TXPManifest;
    procedure PrevMonthClick(Sender: TObject);
    procedure NextMothClick(Sender: TObject);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    procedure Calendar1Click(Sender: TObject);
private
    procedure SetDate(Date: TDateTime);
    function GetDate: TDateTime;
    { Private declarations }
public
    property Date: TDateTime read GetDate write SetDate;
    { Public declarations }
end;
var
    BrDateForm: TBrDateForm;
implementation
{$R *.dfm}
procedure TBrDateForm.SetDate(Date: TDateTime);
begin
    Calendar1.CalendarDate := Date;
end;
function TBrDateForm.GetDate: TDateTime;
begin
    Result := Calendar1.CalendarDate;
end;
procedure TBrDateForm.PrevMonthClick(Sender: TObject);
begin
    Calendar1.PrevMonth;
end;
procedure TBrDateForm.NextMoithClick(Sender: TObject);
begin
    Calendar1.NextMonth;
end;
procedure TBrDateForm.Calendar1Click(Sender: TObject);
begin
    TitleLabel.Caption := FormatDateTime('MMMM, YYYY',
    Calendar1.CalendarDate);
end;
end.

```

#### **EDIT PARTS FORM**

```

unit EdParts;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms, Dialogs, DB, StdCtrls, DBCtrls, Mask, ExtCtrls, Buttons;
type
    TEdPartsForm = class(TForm)
        Panel1: TPanel;
        PrintBtn: TSpeedButton;
        Bevel1: TBevel;
        Navigator: TDBNavigator;
        Panel2: TPanel;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        Label7: TLabel;
        Label8: TLabel;
        Label6: TLabel;
        DBEdit2: TDBEdit;
        DBEdit4: TDBEdit;
        DBEdit5: TDBEdit;
        DBEdit7: TDBEdit;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DBEdit8: TDBEdit;
DBEdPartNo: TDBEdit;
DBEdit3: TDBEdit;
DataComboBox1: TDBLookupComboBox;
OKButton: TButton;
CancelButton: TButton;
PartsSource1: TDataSource;
procedure PrintBtnClick(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
private
  { Private declarations }
public
  procedure Edit(PartNo: Double);
  { Public declarations }
end;
var
  EdPartsForm: TEdPartsForm;
implementation
uses DataMod;
{$R *.dfm}
procedure TEdPartsForm.Edit(PartNo: Double);
begin
  MastData.Parts.Open;
  MastData.Parts.Locate('PartNo', PartNo, []);
  ShowModal;
end;
procedure TEdPartsForm.FormCloseQuery(Sender: TObject;
var CanClose: Boolean);
begin
  CanClose := MastData.DataSetApplyUpdates(MastData.Parts,
ModalResult = mrOK);
end;
end.

```

---

#### **QUERY CUSTOMER DIALOG**

```

unit QryCust;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Buttons, ExtCtrls;
type
  TQueryCustDlg = class(TForm)
    Bevel1: TBevel;
    Label1: TLabel;
    Label2: TLabel;
    MsgLab: TLabel;
    PopupCalBtnFrom: TSpeedButton;
    PopupCalToBtn: TSpeedButton;
    FromEdit: TEdit;
    ToEdit: TEdit;
    CancelBtn: TButton;
    OkBtn: TButton;
    procedure PopupCalBtnFromClick(Sender: TObject);
    procedure PopupCalToBtnClick(Sender: TObject);
    procedure OkBtnClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    function GetFromDate: TDateTime;
    function GetToDate: TDateTime;
    procedure SetFromDate(NewDate: TDateTime);
    procedure SetToDate(NewDate: TDateTime);
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { Private declarations }
public
    property FromDate: TDateTime read GetFromDate write SetFromDate;
    property ToDate: TDateTime read GetToDate write SetToDate;
    { Public declarations }
end;
var
    QueryCustDlg: TQueryCustDlg;
implementation
uses Pickdate;
{$R *.dfm}
procedure TQueryCustDlg.SetFromDate(NewDate: TDateTime);
begin
    FromEdit.Text := DateToStr(NewDate);
end;
procedure TQueryCustDlg.SetToDate(NewDate: TDateTime);
begin
    ToEdit.Text := DateToStr(NewDate);
end;
function TQueryCustDlg.GetFromDate: TDateTime;
begin
    if FromEdit.Text = '' then Result := 0
    else Result := StrToDate(FromEdit.Text);
end;
function TQueryCustDlg.GetToDate: TDateTime;
begin
    if ToEdit.Text = '' then Result := 0
    else Result := StrToDate(ToEdit.Text);
end;
procedure TQueryCustDlg.PopupCalBtnFromClick(Sender: TObject);
begin
    BrDateForm.Date := StrToDate(FromEdit.Text); { start with current
date }
    if BrDateForm.ShowModal = mrOk then
        FromEdit.Text := DateToStr(BrDateForm.Date);
end;
procedure TQueryCustDlg.PopupCalToBtnClick(Sender: TObject);
begin
    BrDateForm.Date := StrToDate(ToEdit.Text); { start with current
date }
    if BrDateForm.ShowModal = mrOk then
        ToEdit.Text := DateToStr(BrDateForm.Date);
end;
procedure TQueryCustDlg.OkBtnClick(Sender: TObject);
begin
    try
        if (ToDate <> 0) and (ToDate < FromDate) then
            begin
                ShowMessage('"TO" date cannot be less than "FROM" date');
                ModalResult := mrNone;
            end
        else ModalResult := mrOk;
    except
        ShowMessage(' Invalid date specified');
        ModalResult := mrNone;
    end;
end;
end;
procedure TQueryCustDlg.FormCreate(Sender: TObject);
begin
    MsgLab.Caption := 'Customers with LastInvoiceDate ranging: ';
    FromDate := EncodeDate(2005, 01, 01);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ToDate := Now;
end;
end.
```

---

#### ADD INFORMATION FORM

```
unit EdAdd;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, Async32, Grids, DBGrids, DB, Mask, DBCtrls,
  ExtCtrls,
  Buttons, XPMan;
type
  TEdAddForm = class(TForm)
    Comm1: TComm;
    Label1: TLabel;
    GroupBox1: TGroupBox;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    CheckBox4: TCheckBox;
    CheckBox5: TCheckBox;
    CheckBox6: TCheckBox;
    CheckBox7: TCheckBox;
    CheckBox8: TCheckBox;
    CheckBox9: TCheckBox;
    CheckBox10: TCheckBox;
    Datasource: TDataSource;
    DBGrid1: TDBGrid;
    XPManifest1: TXPManifest;
    GroupBox2: TGroupBox;
    Label2: TLabel;
    PartNoEd: TEdit;
    Label3: TLabel;
    QtyEd: TEdit;
    Label4: TLabel;
    DBEdit1: TDBEdit;
    AddBtn: TBitBtn;
    ReadBtn: TButton;
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure CheckBox3Click(Sender: TObject);
    procedure CheckBox4Click(Sender: TObject);
    procedure CheckBox5Click(Sender: TObject);
    procedure CheckBox6Click(Sender: TObject);
    procedure CheckBox7Click(Sender: TObject);
    procedure CheckBox8Click(Sender: TObject);
    procedure CheckBox9Click(Sender: TObject);
    procedure ReadBtnClick(Sender: TObject);
    procedure Comm1RxChar(Sender: TObject; Count: Integer);
    procedure CheckBox10Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure AddBtnClick(Sender: TObject);
    procedure QtyEdKeyPress(Sender: TObject; var Key: Char);
  private
    SrchFld: TField;
    { Private declarations }
  public
    { Public declarations }
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

var
  EdAddForm: TEdAddForm;
implementation
uses DataMod;
{$R *.dfm}
procedure TEdAddForm.CheckBox1Click(Sender: TObject);
var
  st: string;
  Count: Integer;
begin
  edit4.Text:='';
  comm1.Open;
  st:=chr($FB)+chr($04)+chr($01)+chr($00)+char($00)+
    chr($00)+chr($01)+chr($FF)+chr($FC);
  Count := Comm1.Write(st[1], Length(st));
  if CheckBox1.Checked=false then
    comm1.Close;

end;
procedure TEdAddForm.CheckBox2Click(Sender: TObject);
var
  st: string;
  Count: Integer;
begin
  edit4.Text:='';
  comm1.Open;
  st:=chr($FB)+chr($04)+chr($01)+chr($00)+char($00)+
    chr($00)+chr($02)+chr($FC)+chr($FC);
  Count := Comm1.Write(st[1], Length(st));
  if CheckBox2.Checked=false then
    comm1.Close;

end;
procedure TEdAddForm.CheckBox3Click(Sender: TObject);
var
  st: string;
  Count: Integer;
begin
  edit4.Text:='';
  comm1.Open;
  st:=chr($FB)+chr($04)+chr($01)+chr($00)+char($00)+
    chr($00)+chr($03)+chr($FD)+chr($FC);
  Count := Comm1.Write(st[1], Length(st));
  if CheckBox3.Checked=false then
    comm1.Close;

end;
procedure TEdAddForm.CheckBox4Click(Sender: TObject);
var
  st: string;
  Count: Integer;
begin
  edit4.Text:='';
  comm1.Open;
  st:=chr($FB)+chr($04)+chr($01)+chr($00)+char($00)+
    chr($00)+chr($04)+chr($FA)+chr($FC);
  Count := Comm1.Write(st[1], Length(st));
  if CheckBox4.Checked=false then
    comm1.Close;

end;
procedure TEdAddForm.CheckBox5Click(Sender: TObject);
var
  st: string;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Count: Integer;
begin
    edit4.Text:='';
    comm1.Open;
    st:=chr($FB)+chr($04)+chr($01)+chr($00)+chr($00)+
        chr($00)+chr($05)+chr($FB)+chr($FC);
    Count := Comm1.Write(st[1], Length(st));
    if CheckBox5.Checked=false then
        comm1.Close;
end;
procedure TEdAddForm.CheckBox6Click(Sender: TObject);
var
    st: string;
    Count: Integer;
begin
    edit4.Text:='';
    comm1.Open;
    st:=chr($FB)+chr($04)+chr($01)+chr($00)+chr($00)+
        chr($00)+chr($06)+chr($FB)+chr($FC);
    Count := Comm1.Write(st[1], Length(st));
    if CheckBox6.Checked=false then
        comm1.Close;
end;
procedure TEdAddForm.CheckBox7Click(Sender: TObject);
var
    st: string;
    Count: Integer;
begin
    comm1.Open;
    st:=chr($FB)+chr($04)+chr($01)+chr($00)+chr($00)+
        chr($00)+chr($07)+chr($FB)+chr($FC);
    Count := Comm1.Write(st[1], Length(st));
    if CheckBox7.Checked=false then
        comm1.Close;
end;
procedure TEdAddForm.CheckBox8Click(Sender: TObject);
var
    st: string;
    Count: Integer;
begin
    edit4.Text:='';
    comm1.Open;
    st:=chr($FB)+chr($04)+chr($01)+chr($00)+chr($00)+
        chr($00)+chr($08)+chr($FB)+chr($FC);
    Count := Comm1.Write(st[1], Length(st));
    if CheckBox8.Checked=false then
        comm1.Close;
end;
procedure TEdAddForm.CheckBox9Click(Sender: TObject);
var
    st,st1: string;
    Count: Integer;
begin
    edit4.Text:='';
    comm1.Open;
    st:=chr($FB)+chr($04)+chr($01)+chr($00)+chr($00)+
        chr($00)+chr($09)+chr($FB)+chr($FC);
    Count := Comm1.Write(st[1], Length(st));
    st1:=chr($FB)+chr($02)+chr($FF)+chr($06)+chr($FC);
    Count := Comm1.Write(st1[1], Length(st1));
    if CheckBox9.Checked=false then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    comm1.Close;
end;
procedure TEdAddForm.ReadBtnClick(Sender: TObject);
var
st1: string;
Count: Integer;
begin
    Edit4.Text:='';
    comm1.Open;
    st1:=chr($FB)+chr($02)+chr($FF)+chr($06)+chr($FC);
    Count := Comm1.Write(st1[1], Length(st1));

end;
procedure TEdAddForm.Comm1RxChar(Sender: TObject; Count: Integer);
var
    Buffer: array[0..1024] of char;
    RF : string;
    Bytes, P: Integer;
begin
    Fillchar(Buffer, Sizeof(Buffer), 0);
    Bytes := Comm1.Read(Buffer, Count);
    for P := 0 to Bytes -1 do
        begin
            x:=x+1;
            RF:=IntToStr((ord(Buffer[P])));
            Edit4.text := Edit4.Text + ' ' + inttostr((ord(Buffer[P])));
            if(x=7) then
                PartNoEd.text:= IntToStr((ord(Buffer[P])));
            if(x>32)then
                x:=0;
        end;
    end;
end;
procedure TEdAddForm.CheckBox10Click(Sender: TObject);
var
st: string;
Count: Integer;
begin
    edit4.Text:='';
    comm1.Open;
    st:=chr($FB)+chr($04)+chr($01)+chr($00)+chr($00)+
        chr($00)+chr($0A)+chr($F4)+chr($FC);
    Count := Comm1.Write(st[1], Length(st));
    if CheckBox10.Checked=false then
        comm1.Close;
end;
procedure TEdAddForm.Button1Click(Sender: TObject);
begin
    close;
    comm1.Close;
end;
procedure TEdAddForm.AddBtnClick(Sender: TObject);
var Qty:integer;
begin
    Qty:=StrToInt(QtyEd.Text);
    SrchFld := Datasource.Dataset.FieldName('PartNo');
    if Datasource.Dataset.Locate('PartNo', PartNoEd.Text,
        [loCaseInsensitive,loCaseInsensitive]) then
        begin
            MastData.Parts.Edit;
            MastData.PartsOnHand.Value := MastData.PartsOnHand.Value + Qty;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    MastData.Parts.Post;
end;
    MastData.Orders.Post;
end;
procedure TEdAddForm.QtyEdKeyPress(Sender: TObject; var Key: Char);
begin
    if Assigned(SrchFld) and (Key > ' ') and not
    (SrchFld.IsValidChar(Key)) then
    begin
        MessageBeep(0);
        Key := #0;
    end;
end;
end.

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CODE เว็บไซต์

```
<html>
<head>
<title>project RFID</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-
874">
<link rel="stylesheet" href="style/my_style.css" type="text/css">
</head>
<body bgcolor="#E8FFF3" leftmargin="0" topmargin="0">

</head>
<!--<body bgcolor="#FFFFFF" leftmargin="0" topmargin="0"
onload="PopupNews();"-->
<body bgcolor="#FFFFFF" leftmargin="0" topmargin="0">

<Script Language="JavaScript">
function CheckSearch(){
Valid = true;
if (document.Search.WORD.value == ""){
Valid = false;
document.Search.WORD.focus();
}
else{
document.Search.submit();
//location.href = "search.asp?WORD=" +
escape(document.Search.WORD.value) + "&PAGE=1";
}
return Valid;
}
function ChkKey(keypressed){
var key;
key = window.event.keyCode;
if (key == 13) {
window.event.keyCode = 0;
CheckSearch();
}
}
</Script>
<table width="780" border="0" cellspacing="0" cellpadding="0">

<td valign="top"></td>

<table width="780" border="0" cellpadding="0" cellspacing="0"
class="Search">
<form name="Search" method="post" action="prel.php3">
<tr bgcolor="#FFCCFF">
<td width="10">
<td width="410">
<table width="400" border="0" cellpadding="0" cellspacing="0"
class="body_e">
<tr>
<td width="2" align="center"><font color="white"><b>|
</b></font></td>
<td width="50" align="center">
<b><a href="test.php3" class="linkwhite">Home</a></b>
</td>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<td width="2" align="center"><font color="white"><b>|
</b></font></td>
<td width="55" align="center">
<a href="antozz.php3" class="linkwhite"><b>product</b></a>
</td>
<td width="2" align="center"><font color="white"><b>|
</b></font></td>
<td width="70" align="center">
<a href="register.php3
" class="linkwhite"><b>register</b></a>
</td>
<td width="2" align="center"><font color="white"><b>|
</b></font></td>

```

```

<td width="70" align="center">
<a href="ppp.php3
" class="linkwhite"><b>contact us</b></a>
</td>
<td width="2" align="center"><font color="white"><b>|
</b></font></td>
<td width="217"></td>
</tr>
</table>
</td>

```

```

<table width="780" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="5"></td>
<td width="170" valign="top" align="center" bgcolor="#FFE7CE">
<form action="prel.php3" method="post">
<br>
<table align="center">
<tr>
<td bgcolor="#FFC78C" colspan="2" align="center" height="30">
<font face="Ms Sans serif">
<b>กรุณาป้อนรายละเอียด</b>
</font>
</td>
</tr>
<tr>
<td bgcolor="#FFC78C">
<font face="Ms Sans serif">
<b>ชื่อผู้ใช้:</b>
</font>
</td>
<td bgcolor="#FFC78C">
<input name="username" type="text">
</td>
</tr>
<tr>
<td bgcolor="#FFC78C">
<font face="Ms Sans serif">
<b>รหัสผ่าน:</b>
</font>
</td>
<td bgcolor="#FFC78C">
<input name="password" type="password">
</td>
</tr>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

<td width="384" valign="top">
<table width="380" border="0" cellspacing="15" cellpadding="0" >
<form name="form" method="post" action="product.php3" target="output"
align="center">
<p>&nbsp;   </p>
<p>

```

```

<select name="produtname">
<option value="">productname
<option value="01">HARRY POTER
<option value="02">เครื่องพิมพ์
<option value="03">ROOM
<option value="04">ชุดกีฬา
<option value="05">บ้านอยู่ๆกับดา
<option value="07">ของขวัญ(Gift)
<option value="08">ตลกต่างๆ
<option value="09">GREEN FOOD
<option value="10">แพรว
<option value="11">บ้านและสวน
<option value="12">pen

```

```

</select>
<input type="submit" name="Submit" value="ดูข้อมูล">

```

```

</p><p>

```

```

</form>
<?php

```

```

$product=addslashes($productname);
$Product="$product";
echo "$product ";

```

```

?>
</table>

```

```

<table width="380" border="0" cellspacing="0" cellpadding="0">
<tr>
<td>

<map name="Head">
<b><font color="green">ชุดแฮรี่ Gloden</font></b><br><br>
<li> collection เล่ม 1-5 <br><br>
<li> + เล่ม 6 ปกแข็งฟรี<br><br>
<li> ราคา 2,999.00 บาท<br><br>
<area shape="rect" coords="25,221,138,241" href="MB89F202P.asp">
</map>
</td>
</tr>
</table>
<br>

```

```

<table width="380" border="0" cellspacing="0" cellpadding="0">
<tr height="20">
<td width="190" bgcolor="#A40829" class="body_e">
&nbsp;  <font color="white">:: NEW PRODUCT ::</font>
</td>
<td width="190"></td>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

</tr>
<tr>
<td colspan="2">
<table width="380" border="0" cellspacing="0" cellpadding="0"
class="body_ts">
<tr height="5"><td></td></tr>
<tr>
<td colspan="2">

<b><font color="green">แจ้งทีมจอบนางแห่งวังหลวงเล่ม 4

<br><br>
</td>
</tr>
<tr>
<td width="200">

<b><font color="green">GREENFOOD</font></b><br><br>

<br><br>

</td>
</tr>
<tr height="5"><td colspan="2"></td></tr>
<tr>
<td colspan="2" align="center">
<hr color="#CACACA" width="80%">
</td>
</tr>

</table>
<!--<table width="380" border="0" cellspacing="0" cellpadding="0"
class="body_ts">
<tr height="5"><td></td></tr>
<tr>
<td>

</td>
</tr>
<tr height="5"><td></td></tr>
<tr>
<td>
<b><font size="3">Flash MCU</font></b>
</td>
</tr>
<tr height="5"><td></td></tr>
<tr>
<td width="136" valign="top">
<a href="mb89-mb90.asp"></a><br><br>
<b>MB89N202P</b><br>
Low cost 8 bit<br>
32 pin shrink dip<br>
</td>
<td width="3"></td>
<td width="1"></td>
<td width="3"></td>
<td width="137" valign="top">
<a href="mb89-mb90.asp"></a><br><br>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

<a href="antozz.php3" class="linkred">More...</a>
</td>
</tr>
</table>
</td>
<td width="1" bgcolor="#777777"></td>
<td width="5"></td>
<td width="180" valign="top">
<table width="180" border="0" cellspacing="0" cellpadding="0"
class="body_e">
<tr>
<td>
<a href="workshop.asp"></a>
</td>
</tr>
<tr height="5"><td></td></tr>
<tr>
<td>

</td>
</tr>
<tr height="5"><td></td></tr>
<tr>
<td>
<a href="prodtype.asp?Ctn=008&Group=001"></a>
</td>
</tr>
<!--<tr height="5"><td></td></tr>
<tr>
<td>
<a href="ethernet_io.asp"></a>
</td>
</tr-->
<tr height="5"><td></td></tr>
<tr>
<td>
<a href="pic10f.asp"></a>
</td>
</tr>
<tr height="5"><td></td></tr>
<tr>
<td>
<a href="kia278rxx.asp"></a>
</td>
</tr>
<tr height="5"><td></td></tr>
<tr>
<td>
<a href="skit-dspic.asp"></a>
</td>
</tr>
</table>
<!--<table width="180" border="0" cellspacing="0" cellpadding="0"
class="body_e">
<tr height="20">

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<td align="center" bgcolor="#A40829">
<font color="white">:: FUTURE PRODUCT ::</font>
</td>
</tr>
<tr>
<td>
<table width="180" border="0" cellspacing="0" cellpadding="0"
class="body_ts">
<tr height="5"><td></td></tr>
<tr>
<td>

</td>
</tr>
<tr height="10"><td></td></tr>
<tr height="20">
<td>
dsPIC30F2010-20I/SP<br>
dsPIC30F6014-30I/PF<br>
Digital Signal Controller 16 bits<br>
MCU with Internal DSP
</td>
</tr>
</table>
<br>
</td>
</tr>
</table>-->
<br>
<!--<table width="180" border="0" cellspacing="0" cellpadding="0"
class="body_e">
<tr height="20">
<td align="center" bgcolor="#A40829">
<font color="white">:: NEWS ::</font>
</td>
</tr>
<tr>
<td>
</td>
</tr>
</table>-->
<br>
<!--include file="banner.html">-->
</td>
<td width="5"></td>
</tr>
</table>
<table width="575" border="0" cellspacing="0" cellpadding="0">
<tr height="5"><td></td></tr>
</table>
</td>
</tr>
</table>
<br><br><br>
```

```
<table width="780" border="0" cellspacing="0" cellpadding="0"
class="footer">
<tr>
<td align="center">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<table width="90%" border="0" cellspacing="0" cellpadding="0"
class="footer">

<tr><td height="1" bgcolor="coral"></td></tr>
</table>
</td>
</tr>
<tr>
<td valign="top" align="center">
<br>
| <a href="register.php3" class="footer">สมัครสมาชิก</a> | <a
href="buy.php3" class="footer">วิธีสั่งซื้อสินค้า</a> | <a href="cart.asp"
class="footer">แผนกคุณลูกค้า</a> |
</td>
</tr>
<tr>
<td valign="top" align="center">
<br>
<font color="navy" size="3"><b></b></font><br>
<font color="brown">
KMITL<br>

</font>
</td>
</tr>
</table>
<br>
</body>
</html>

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Source code program C

```
#include <AT89X52.H>

sbit relay1=P1^1;
sbit relay2=P1^2;

/*****

    FN DELAY

*****/
void delay1(int time)
{
int i,j;
for(i=0;i<=255;i++)
for(j=0;j<=time;j++);
}

/*****

    FN Serial_Port

*****/
void init_interr_serial(void)
{
// XTAL 11.0592 MHZ
// 12Machines/Cycle
    TMOD = 0x20;
    SCON = 0x52;
    PCON = 0x00;
    TH1 = 0xFD; //BaudRate 9600
    TR1 = 1;
    REN = 1;
    TI = 1;
    RI = 0;
// ES = 0; //Disable Interrupt Coz,Not Work
}

/*****

    FN RECIEVE_CHAR

*****/
char recv_char()
{
    char c;
    while (!RI)
    delay1(1);
    c=SBUF;
    RI=0;
    return(c);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****
      FN MAIN
*****/
void main()
{
int ms,type;
char x;
ms=0;
init_interr_serial();
while(1)
{
    P0=0;
    x = recv_char();
    delay1(10);
    switch(x)
    {
        case 'A' : type = 0;break;
        case 'B' : type = 1;break;
        case 'D' : type = 2;break;
        case 'C' : type = 3;break;
        case 'E' : type = 4;break;
        default : break;
    }
    RI=0;
    if(type==0 )relay1=0;
    if(type==1 )relay1=1;
    if(type==2 )relay2=0;
    if(type==3 )relay2=1;
    if(type==4 )
    {
        relay1=1;
        relay2=1;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Source code program Visual C++

### Header File

```
// RFIDDlg.h : header file
//

#if !defined(AFX_RFIDDLG_H__7AC6E0CE_8770_4377_B9C3_5151DF5A1ED1__INCLUDE
D_)
#define
AFX_RFIDDLG_H__7AC6E0CE_8770_4377_B9C3_5151DF5A1ED1__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "netcom.h"

////////////////////////////////////
////////////////////////////////////
// CRFIDDlg dialog

class CRFIDDlg : public CDialog
{
// Construction
public:
    CRFIDDlg(CWnd* pParent = NULL); // standard constructor

    Cnetcom        m_EthernetIO;
    CString        m_StatusStr;
    CString        m_StatusStr1;
    CString        m_StatusStr2;
// Dialog Data
    //{{AFX_DATA(CRFIDDlg)
    enum { IDD = IDD_RFID_DIALOG };
    CIPAddressCtrl m_IPAddress;
        // NOTE: the ClassWizard will add data members here
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CRFIDDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX);    //
// DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
    //{{AFX_MSG(CRFIDDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    afx_msg void OnReaderlon();
    afx_msg void OnReaderloff();
    afx_msg void OnButton2();
    afx_msg void OnReader2on();
    afx_msg void OnReader2off();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        virtual void OnOK();
        virtual void OnCancel();
        afx_msg void OnButton5();
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations
immediately before the previous line.

#endif //
!defined(AFX_RFIDDLG_H__7AC6E0CE_8770_4377_B9C3_5151DF5A1ED1__INCLUDE
D_)
.....
RFID Form
// RFIDDlg.cpp : implementation file
//

#include "stdafx.h"
#include "RFID.h"
#include "RFIDDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV
support
//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
//}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{{AFX_DATA_INIT(CAboutDlg)
    //}}}AFX_DATA_INIT

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
    // No message handlers
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
////////////////////////////////////
// CRFIDDlg dialog

CRFIDDlg::CRFIDDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CRFIDDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CRFIDDlg)
    // NOTE: the ClassWizard will add member initialization
here
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon
in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CRFIDDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CRFIDDlg)
    // NOTE: the ClassWizard will add DDX and DDV calls here
    //}}AFX_DATA_MAP
    DDX_Text(pDX, IDC_STATIC6, m_StatusStr);
    DDX_Text(pDX, IDC_STATIC5, m_StatusStr1);
    DDX_Text(pDX, IDC_STATIC4, m_StatusStr2);
    DDX_Control(pDX, IDC_IPADDRESS1, m_IPaddress);
}

BEGIN_MESSAGE_MAP(CRFIDDlg, CDialog)
    //{{AFX_MSG_MAP(CRFIDDlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, OnReader1on)
    ON_BN_CLICKED(IDC_BUTTON3, OnReader1off)
    ON_BN_CLICKED(IDC_BUTTON2, OnReader2on)
    ON_BN_CLICKED(IDC_BUTTON4, OnReader2off)
    ON_BN_CLICKED(IDC_BUTTON5, OnButton5)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
////////////////////////////////////
// CRFIDDlg message handlers

BOOL CRFIDDlg::OnInitDialog()

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this
    // automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon

    // TODO: Add extra initialization here
    GetDlgItem(IDC_BUTTON1)->EnableWindow( FALSE );
    GetDlgItem(IDC_BUTTON2)->EnableWindow( FALSE );
    GetDlgItem(IDC_BUTTON3)->EnableWindow( FALSE );
    GetDlgItem(IDC_BUTTON4)->EnableWindow( FALSE );
    m_EthernetIO.SetStatusMsg( &m_StatusStr);
    m_EthernetIO.SetStatusMsg( &m_StatusStr1);
    m_EthernetIO.SetStatusMsg( &m_StatusStr2);
    m_EthernetIO.SetServerState( false ); // run as client
    m_EthernetIO.SetSmartAddressing( false ); // always send to
server

    return TRUE; // return TRUE unless you set the focus to a
control
}

void CRFIDDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the
code below

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// to draw the icon. For MFC applications using the document/view
model,
// this is automatically done for you by the framework.

void CRFIDDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(),
0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the
user drags
// the minimized window.
HCURSOR CRFIDDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

void CRFIDDlg::OnReaderlon()
{
    char message[]="A";

    if(m_EthernetIO.IsOpen())
    {
        if(m_EthernetIO.WriteComm((unsigned char*) &message[0],
sizeof(message), INFINITE) <= 0L)
        {
            m_StatusStr.Format("Send fail");
            UpdateData(FALSE);
        }
        m_StatusStr2 = _T("Reader 1 ON: ");
        UpdateData(FALSE);
        GetDlgItem(IDC_BUTTON1)->EnableWindow( FALSE );
        GetDlgItem(IDC_BUTTON3)->EnableWindow( TRUE );
    }
}

void CRFIDDlg::OnReaderloff()
{
    char message[]="B";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(m_EthernetIO.IsOpen()){

    if(m_EthernetIO.WriteComm((unsigned char*) &message[0],
sizeof(message), INFINITE) <= 0L)
    {
        m_StatusStr.Format("Send fail");
        UpdateData(FALSE);
    }
    m_StatusStr2 = _T("Reader 1 OFF: ");
    UpdateData(FALSE);
    GetDlgItem(IDC_BUTTON3)->EnableWindow( FALSE );
    GetDlgItem(IDC_BUTTON1)->EnableWindow( TRUE );
}

}

void CRFIDDlg::OnReader2on()
{
    char message[]="D";

    if(m_EthernetIO.IsOpen())
    {
        if(m_EthernetIO.WriteComm((unsigned char*) &message[0],
sizeof(message), INFINITE) <= 0L)
        {
            m_StatusStr.Format("Send fail");
            UpdateData(FALSE);
        }
        m_StatusStr1 = _T("Reader 2 ON: ");
        UpdateData(FALSE);
        GetDlgItem(IDC_BUTTON2)->EnableWindow( FALSE );
        GetDlgItem(IDC_BUTTON4)->EnableWindow( TRUE );
    }
}

void CRFIDDlg::OnReader2off()
{
    char message[]="C";

    if(m_EthernetIO.IsOpen()){

        if(m_EthernetIO.WriteComm((unsigned char*) &message[0],
sizeof(message), INFINITE) <= 0L){
            m_StatusStr.Format("Send fail");
            UpdateData(FALSE);
        }
        m_StatusStr1 = _T("Reader 2 OFF: ");
        UpdateData(FALSE);

        GetDlgItem(IDC_BUTTON4)->EnableWindow( FALSE );
        GetDlgItem(IDC_BUTTON2)->EnableWindow( TRUE );
    }

}

void CRFIDDlg::OnOK()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bool bSuccess;
m_IPAddress.GetWindowText( IPServerStr );

//connecting
bSuccess = m_EthernetIO.ConnectTo( IPServerStr ); // TCP

//if connecting suces Show message
if (bSuccess && m_EthernetIO.WatchComm())
{
    m_StatusStr = _T("Connection established with EtherIO: ")
+ IPServerStr;
    GetDlgItem(IDOK)->EnableWindow( FALSE );
    GetDlgItem(IDC_BUTTON1)->EnableWindow( TRUE );
    GetDlgItem(IDC_BUTTON2)->EnableWindow( TRUE );
    GetDlgItem(IDC_BUTTON3)->EnableWindow( FALSE );
    GetDlgItem(IDC_BUTTON4)->EnableWindow( FALSE );
    UpdateData( FALSE );
    m_StatusStr1 = _T("Reader 2 OFF: ");
    UpdateData( FALSE );
    m_StatusStr2 = _T("Reader 1 OFF: ");
    UpdateData( FALSE );
}
else
{
    int error = WSAGetLastError();
    TRACE("socket error = 0x%X", error);
    m_StatusStr.Format("Connection fail");
    UpdateData( FALSE );
}
//CDialog::OnOK();
}

void CRFIDDlg::OnCancel()
{
    m_EthernetIO.StopComm();
    CDialog::OnCancel();
}

void CRFIDDlg::OnButton5()
{
    char message[]="E";

    if(m_EthernetIO.IsOpen()){

        if(m_EthernetIO.WriteComm((unsigned char*) &message[0],
sizeof(message), INFINITE) <= 0L){
            m_StatusStr.Format("Send fail");
            UpdateData( FALSE );
        }

    }

    m_StatusStr = _T("Disconnected");
    GetDlgItem(IDOK)->EnableWindow( TRUE );
    m_EthernetIO.StopComm();
    UpdateData( FALSE );
    m_StatusStr1 = _T("Reader 2 OFF: ");
    UpdateData( FALSE );
    m_StatusStr2 = _T("Reader 1 OFF: ");
    UpdateData( FALSE );
    GetDlgItem(IDC_BUTTON1)->EnableWindow( FALSE );

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GetDlgItem(IDC_BUTTON2)->EnableWindow( FALSE );  
GetDlgItem(IDC_BUTTON3)->EnableWindow( FALSE );  
GetDlgItem(IDC_BUTTON4)->EnableWindow( FALSE );  
  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

- Contactless Read/Write Data Transmission
- Radio Frequency  $f_{RF}$  from 100 kHz to 150 kHz
- e5550 Binary Compatible or T5557 Extended Mode
- Small Size, Configurable for ISO/IEC 11784/785 Compatibility
- 75 pF On-chip Resonant Capacitor (Mask Option)
- 7 × 32-bit EEPROM Data Memory Including 32-bit Password
- Separate 64-bit memory for Traceability Data
- 32-bit Configuration Register in EEPROM to Setup:
  - Data Rate
    - RF/2 to RF/128, Binary Selectable or
    - Fixed e5550 Data Rates
  - Modulation/Coding
    - FSK, PSK, Manchester, Biphasic, NRZ
  - Other Options
    - Password Mode
    - Max Block Feature
    - Answer-On-Request (AOR) Mode
    - Inverse Data Output
    - Direct Access Mode
    - Sequence Terminator(s)
    - Write Protection (Through Lock-bit per Block)
    - Fast Write Method (5 kbps versus 2 kbps)
    - OTP Functionality
    - POR Delay up to 67 ms

## Description

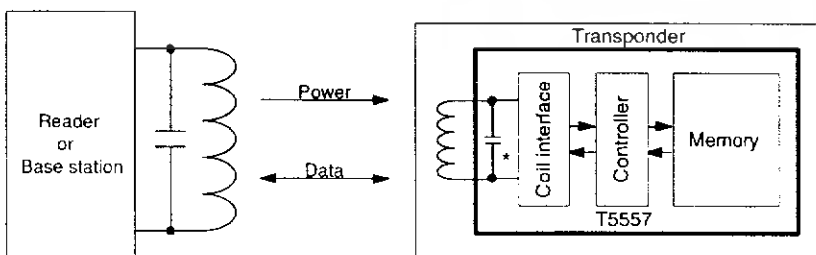
The T5557 is a contactless R/W IDentification IC (IDIC<sup>®</sup>) for applications in the 125 kHz frequency range. A single coil, connected to the chip, serves as the IC's power supply and bi-directional communication interface. The antenna and chip together form a transponder or tag.

The on-chip 330-bit EEPROM (10 blocks, 33 bits each) can be read and written block-wise from a reader. Block 0 is reserved for setting the operation modes of the T5557 tag. Block 7 may contain a password to prevent unauthorized writing.

Data is transmitted from the IDIC using load modulation. This is achieved by damping the RF field with a resistive load between the two terminals Coil 1 and Coil 2. The IC receives and decodes 100% amplitude modulated (OOK) pulse interval encoded bit streams from the base station or reader.

## System Block Diagram

Figure 1. RFID System Using T5557 Tag



\* Mask option



Multifunctional  
330-bit  
Read/Write  
RF-Identification  
IC

T5557

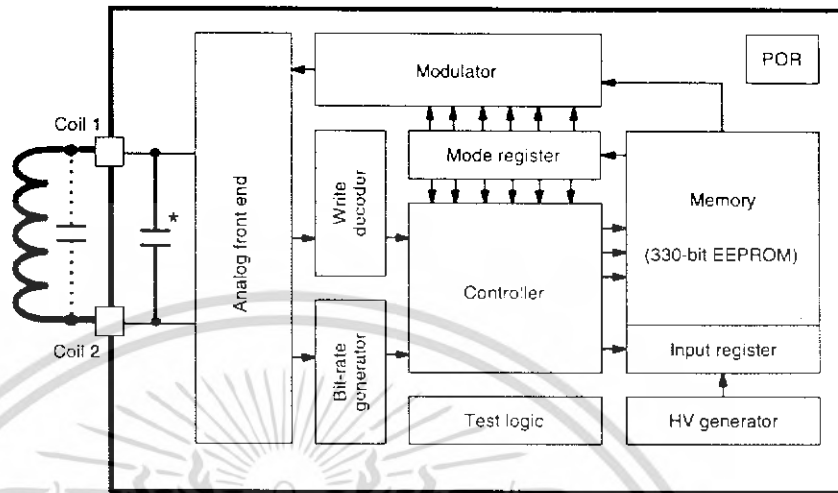
Rev. 4517F-RFID-11/03



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## T5557 – Building Blocks

Figure 2. Block Diagram



\* Mask option

### Analog Front End (AFE)

The AFE includes all circuits which are directly connected to the coil. It generates the IC's power supply and handles the bi-directional data communication with the reader. It consists of the following blocks:

- Rectifier to generate a DC supply voltage from the AC coil voltage
- Clock extractor
- Switchable load between Coil 1/Coil 2 for data transmission from tag to the reader
- Field gap detector for data transmission from the base station to the tag
- ESD protection circuitry

### Data-rate Generator

The data rate is binary programmable to operate at any data rate between  $RF/2$  and  $RF/128$  or equal to any of the fixed e5550/e5551 and T5554 bitrates ( $RF/8$ ,  $RF/16$ ,  $RF/32$ ,  $RF/40$ ,  $RF/50$ ,  $RF/64$ ,  $RF/100$  and  $RF/128$ ).

### Write Decoder

This function decodes the write gaps and verifies the validity of the data stream according to the Atmel e555x write method (pulse interval encoding).

### HV Generator

This on-chip charge pump circuit generates the high voltage required for programming of the EEPROM.

### DC Supply

Power is externally supplied to the IDIC via the two coil connections. The IC rectifies and regulates this RF source and uses it to generate its supply voltage.

**Power-On Reset (POR)**

This circuit delays the IDIC functionality until an acceptable voltage threshold has been reached.

**Clock Extraction**

The clock extraction circuit uses the external RF signal as its internal clock source.

**Controller**

The control-logic module executes the following functions:

- Load-mode register with configuration data from EEPROM block 0 after power-on and also during reading
- Control memory access (read, write)
- Handle write data transmission and write error modes
- The first two bits of the reader to tag data stream are the opcode, e.g., write, direct access or reset
- In password mode, the 32 bits received after the opcode are compared with the password stored in memory block 7

**Mode Register**

The mode register stores the configuration data from the EEPROM block 0. It is continually refreshed at the start of every block read and (re-)loaded after any POR event or reset command. On delivery the mode register is preprogrammed with the value '0014 3000'h which corresponds to continuous read of block 0, Manchester coded, RF/64.

Figure 3. Block 0 Configuration Mapping – e5550 Compatibility Mode

L	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
	0	1	1	0	0	0	0	0	0	0	0				0								0							0	0	
Lock Bit	Safer Key Note 1), 2)				Data Bit Rate				Modulation								PSK- CF		AOR	MAX- BLOCK			PWD	ST-Sequence Terminator		POR delay						
	0 Unlocked				RF/8 0 0 0												0 0 RF/2															
	1 Locked				RF/16 0 0 1												0 1 RF/4															
					RF/32 0 1 0												1 0 RF/8															
					RF/40 0 1 1												1 1 Res.															
					RF/50 1 0 0				0 0 0 0 0 Direct																							
					RF/64 1 0 1				0 0 0 0 1 PSK1																							
					RF/100 1 1 0				0 0 0 1 0 PSK2																							
					RF/128 1 1 1				0 0 0 1 1 PSK3																							
									0 0 1 0 0 FSK1																							
								0 0 1 0 1 FSK2																								
								0 0 1 1 0 FSK1a																								
								0 0 1 1 1 FSK2a																								
								0 1 0 0 0 Manchester																								
								1 0 0 0 0 Biphase('50)																								
								1 1 0 0 0 Reserved																								
1) If Master Key = 6 then test mode write commands are ignored 2) If Master Key <> 6 or 9 then extended function mode is disabled																																





## Modulator

The modulator consists of data encoders for the following basic types of modulation:

**Table 1.** Types of e5550-compatible Modulation Modes

Mode	Direct Data Output
FSK1a <sup>(1)</sup>	FSK/8-/5 '0' = r/8; '1' = r/5
FSK2a <sup>(1)</sup>	FSK/8-/10 '0' = r/8; '1' = r/10
FSK1 <sup>(1)</sup>	FSK/5-/8 '0' = r/5; '1' = r/8
FSK2 <sup>(1)</sup>	FSK/10-/8 '0' = r/10; '1' = r/8
PSK1 <sup>(2)</sup>	Phase change when input changes
PSK2 <sup>(2)</sup>	Phase change on bit clock if input high
PSK3 <sup>(2)</sup>	Phase change on rising edge of input
Manchester	'0' = falling edge, '1' = rising edge
Biphase	'1' creates an additional mid-bit change
NRZ	'1' = damping on, '0' = damping off

- Notes:
1. A common multiple of bitrate and FSK frequencies is recommended.
  2. In PSK mode the selected data rate has to be an integer multiple of the PSK sub-carrier frequency.

## Memory

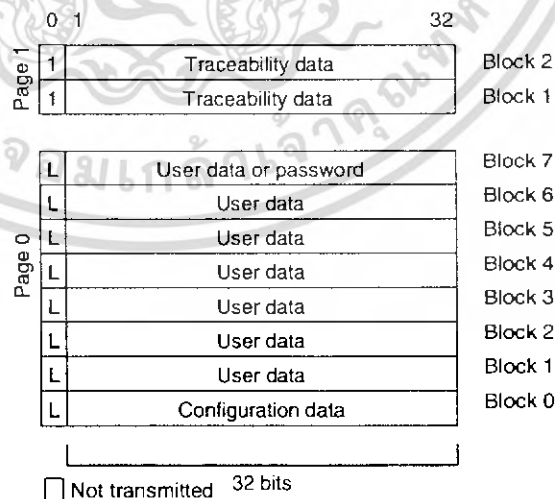
The memory is a 330-bit EEPROM, which is arranged in 10 blocks of 33 bits each. All 33 bits of a block, including the lock bit, are programmed simultaneously.

Block 0 of page 0 contains the mode/configuration data, which is not transmitted during regular-read operations. Block 7 of page 0 may be used as a write protection password.

Bit 0 of every block is the lock bit for that block. Once locked, the block (including the lock bit itself) is not re-programmable through the RF field again.

Blocks 1 and 2 of page 1 contain traceability data and are transmitted with the modulation parameters defined in the configuration register after the opcode '11' is issued by the reader (see Figure 11 on page 9). These traceability data blocks are programmed and locked by Atmel.

**Figure 4.** Memory Map

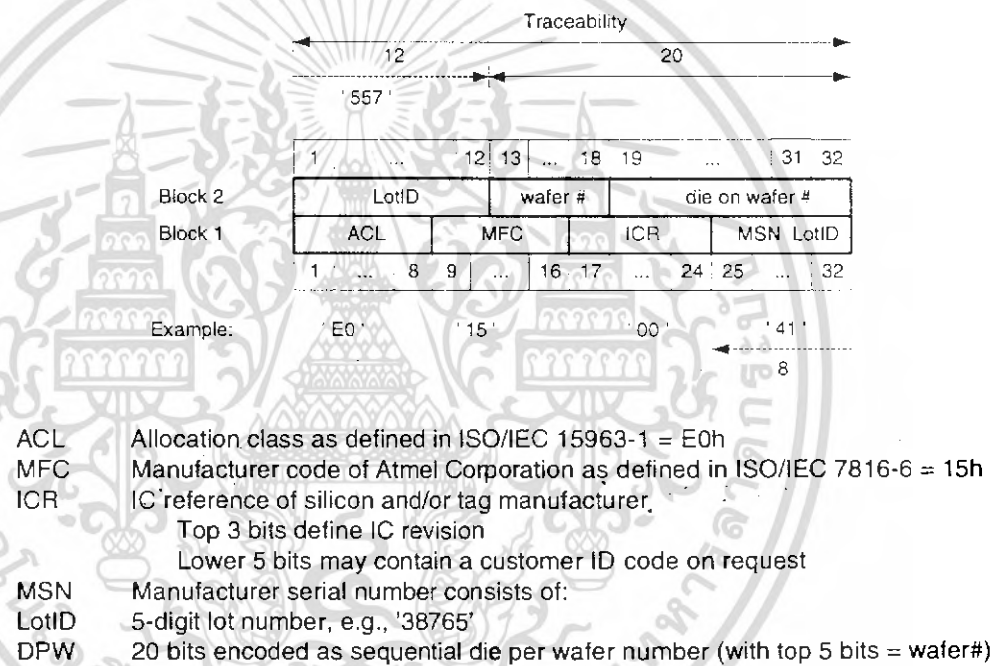


**Traceability Data Structure**

Blocks 1 and 2 of page 1 contain the traceability data and are programmed and locked by Atmel during production testing. The most significant byte of block 1 is fixed to 'E0'hex, the allocation class (ACL) as defined in ISO/IEC 15963-1. The second byte is therefore defined as the manufacturer's ID of Atmel (= '15'hex). The following 8 bits are used as IC reference byte (ICR - Bits 47 to 40). The 3 most significant bits define the IC and/or foundry version of the T5557. The lower 5 bits are by default reset (=00) as the Atmel standard value. Other values may be assigned on request to high volume customers as tag issuer identification.

The lower 40 bits of the data encode the traceability information of Atmel and conform to a unique numbering system. These 40 data bits are divided in two sub-groups, a 5-digit lot ID number, the binary wafer number (5 bit) concatenated with the sequential die number per wafer.

**Figure 5. T5557 Traceability Data Structure**



**Operating the T5557**

**Initialization and POR Delay**

The Power-On-Reset (POR) circuit remains active until an adequate voltage threshold has been reached. This in turn triggers the default start-up delay sequence. During this configuration period of about 192 field clocks, the T5557 is initialized with the configuration data stored in EEPROM block 0. During initialization of the configuration block 0, all T55570x variants the load damping is active permanently (see Figure 10 on page 9). The T55571x types (without damping option) achieve a longer read range based on the lower activation field strength.

If the POR-delay bit is reset, no additional delay is observed after the configuration period. Tag modulation in regular-read mode will be observed about 3 ms after entering the RF field. If the POR delay bit is set, the T5557 remains in a permanent damping state until 8190 internal field clocks have elapsed.

$$T_{INIT} = (192 + 8190 \times \text{POR delay}) \times T_C \approx 67 \text{ ms} ; T_C = 8 \mu\text{s at } 125 \text{ kHz}$$





Any field gap occurring during this initialization phase will restart the complete sequence. After this initialization time the T5557 enters regular-read mode and modulation starts automatically using the parameters defined in the configuration register.

### Tag to Reader Communication

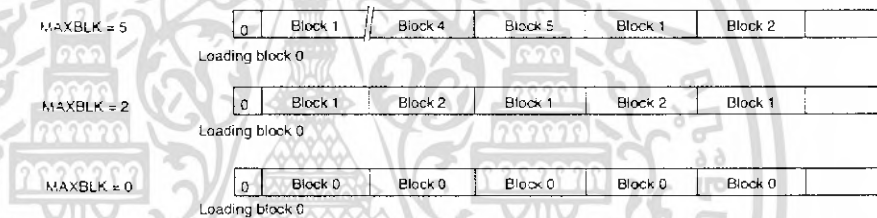
During normal operation, the data stored within the EEPROM is cycled and the Coil 1, Coil 2 terminals are load modulated. This resistive load modulation can be detected at the reader module.

### Regular-read Mode

In regular-read mode data from the memory is transmitted serially, starting with block 1, bit 1, up to the last block (e.g., 7), bit 32. The last block which will be read is defined by the mode parameter field MAXBLK in EEPROM block 0. When the data block addressed by MAXBLK has been read, data transmission restarts with block 1, bit 1.

The user may limit the cyclic datastream in regular-read mode by setting the MAXBLK between 0 and 7 (representing each of the 8 data blocks). If set to 7, blocks 1 through 7 can be read. If set to 1, only block 1 is transmitted continuously. If set to 0, the contents of the configuration block (normally not transmitted) can be read. In the case of MAXBLK = 0 or 1, regular-read mode can not be distinguished from block-read mode.

Figure 6. Examples for Different MAXBLK Settings



Every time the T5557 enters regular- or block-read mode, the first bit transmitted is a logical '0'. The data stream starts with block 1, bit 1, continues through MAXBLK, bit 32, and cycles continuously if in regular-read mode.

Note: This behavior is different from the original e555x and helps to decode PSK-modulated data.

### Block-read Mode

With the direct access command, the addressed block is repetitively read only. This mode is called block-read mode. Direct access is entered by transmitting the page access opcode ('10' or '11'), a single '0' bit and the requested 3-bit block address when the tag is in normal mode.

In password mode (PWD bit set), the direct access to a single block needs the valid 32-bit password to be transmitted after the page access opcode whereas a '0' bit and the 3-bit block address follow afterwards. In case the transmitted password does not match with the contents of block 7, the T5557 tag returns to the regular-read mode.

Note: A direct access to block 0 of page 1 will read the configuration data of block 0, page 0. A direct access to block 3 .. 7 of page 1 reads all data bits as zero.

### e5550 Sequence Terminator

The sequence terminator ST is a special damping pattern which is inserted before the first block and may be used to synchronize the reader. This e5550-compatible sequence terminator consists of 4 bit periods with underlying data values of '1'. During the second and the fourth bit period, modulation is switched off (Manchester encoding – switched on). Biphase modulated data blocks need fixed leading and trailing bits in combination with the sequence terminator to be identified reliably.

The sequence terminator may be individually enabled by setting of mode bit 29 (ST = '1') in the e5550-compatibility mode (X-mode = '0').

In the regular-read mode, the sequence terminator is inserted at the start of each MAXBLK-limited read data stream.

In block-read mode – after any block-write or direct access command – or if MAXBLK was set to 0 or 1, the sequence terminator is inserted before the transmission of the selected block.

Especially this behavior is different to former e5550 – compatible ICs (T5551, T5554).

Figure 7. Read Data Stream with Sequence Terminator

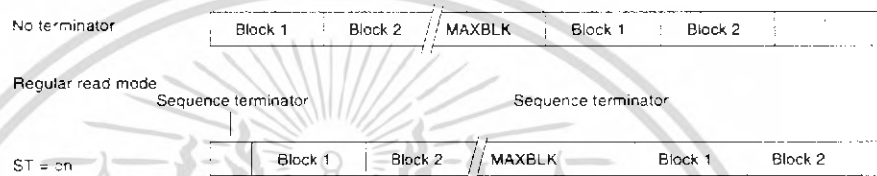
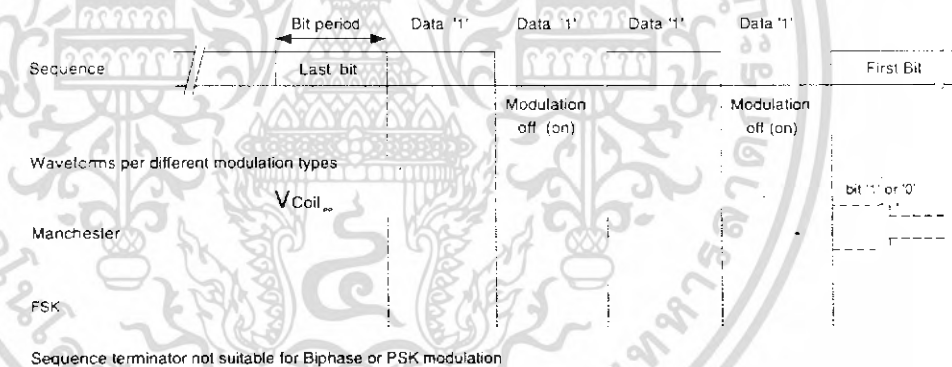


Figure 8. e5550-compatible Sequence Terminator Waveforms



**Reader to Tag Communication**

Data is written to the tag by interrupting the RF field with short field gaps (on-off keying) in accordance with the e5550 write method. The time between two gaps encodes the '0/1' information to be transmitted (pulse interval encoding). The duration of the gaps is usually 50 μs to 150 μs. The time between two gaps is nominally 24 field clocks for a '0' and 54 field clocks for a '1'. When there is no gap for more than 64 field clocks after a previous gap, the T5557 exits the write mode. The tag starts with the command execution if the correct number of bits were received. If there is a failure detected the T5557 does not continue and will enter regular-read mode.

**Start Gap**

The initial gap is referred to as the start gap. This triggers the reader to tag communication. During this mode of operation, the receive damping is permanently enabled to ease gap detection. The start gap may need to be longer than subsequent gaps in order to be detected reliably.

A start gap will be accepted at any time after the mode register has been loaded ( $\geq 3$  ms). A single gap will not change the previously selected page (by former opcode '10' or '11').

Figure 9. Start of Reader to Tag Communication

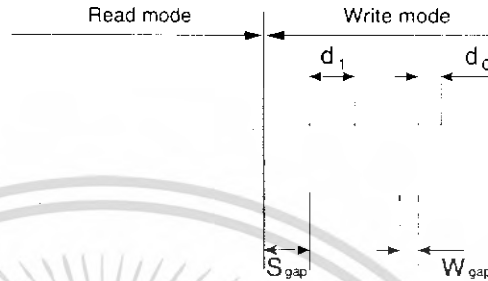


Table 2. Write Data Decoding Scheme

Parameters	Remark	Symbol	Min.	Max.	Unit
Start gap		$S_{gap}$	10	50	FC
Write gap	Normal write mode	$W_{gap}$	8	30	FC
Write data in normal mode	'0' data	$d_0$	16	31	FC
	'1' data	$d_1$	48	63	FC

## Write Data Protocol

The T5557 expects to receive a dual bit opcode as the first two bits of a reader command sequence. There are three valid opcodes:

- The opcodes '10' and '11' precede all block write and direct access operations for page 0 and page 1
- The RESET opcode '00' initiates a POR cycle
- The opcode '01' precedes all test mode write operations. Any test mode access is ignored after master key (bits 1..4) in block 0 has been set to '6'. Any further modifications of the master key are prohibited by setting the lock bit of block 0 or the OTP bit.

Writing has to follow these rules:

- Standard write needs the opcode, the lock bit, 32 data bits and the 3-bit address (38 bits total)
- Protected write (PWD bit set) requires a valid 32-bit password between opcode and data, address bits
- For the AOR wake-up command an opcode and a valid password are necessary to select and activate a specific tag

Note: The data bits are read in the same order as written.

If the transmitted command sequence is invalid, the T5557 enters regular-read mode with the previously selected page (by former opcode '10' or '11').

Figure 10. Complete Writing Sequence

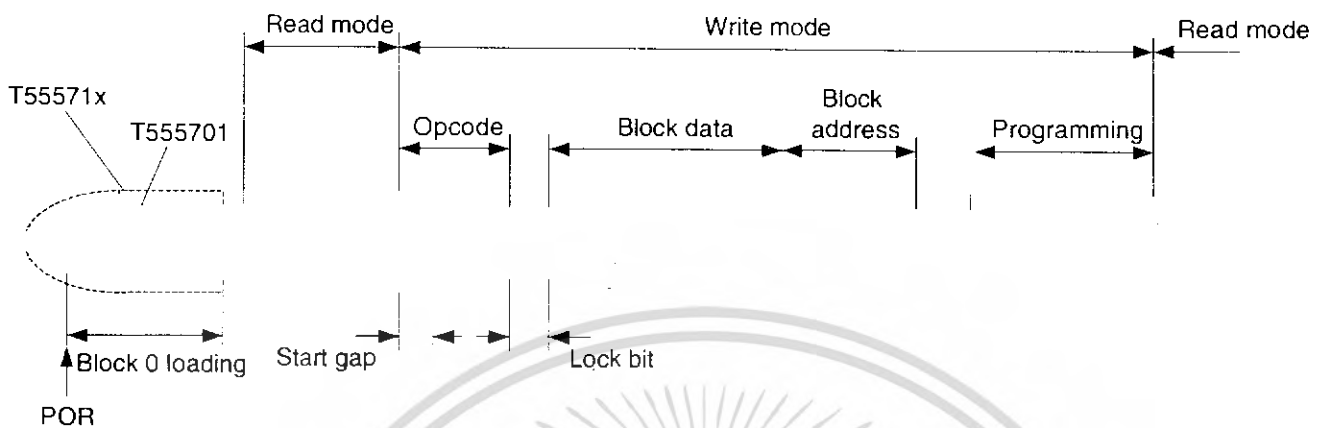


Figure 11. T5557 Command Formats

Command	Opcode	1	Data	32	2	Addr	0
Standard write	1p*	L	1	Data	32	2	Addr 0
Protected write	1p*	1	Password	32	L	1	Data 32 2 Addr 0
AOR (wake-up command)	10	1	Password	32			
Direct access (PWD = 1)	1p*	1	Password	32	0	2	Addr 0
Direct access (PWD = 0)	1p*	0	2	Addr	0		
Page 0/1 regular read	1p*						
Reset command	00						

\* p = page selector

**Password**

When password mode is active (PWD = 1), the first 32 bits after the opcode are regarded as the password. They are compared bit by bit with the contents of block 7, starting at bit 1. If the comparison fails, the T5557 will not program the memory, instead it will restart in regular-read mode once the command transmission is finished.

Note: In password mode, MAXBLK should be set to a value below 7 to prevent the password from being transmitted by the T5557.

Each transmission of the direct access command (two opcode bits, 32 bits password, '0' bit plus 3 address bits = 38 bits) needs about 18 ms. Testing all possible combinations (about 4.3 billion) takes about two years.

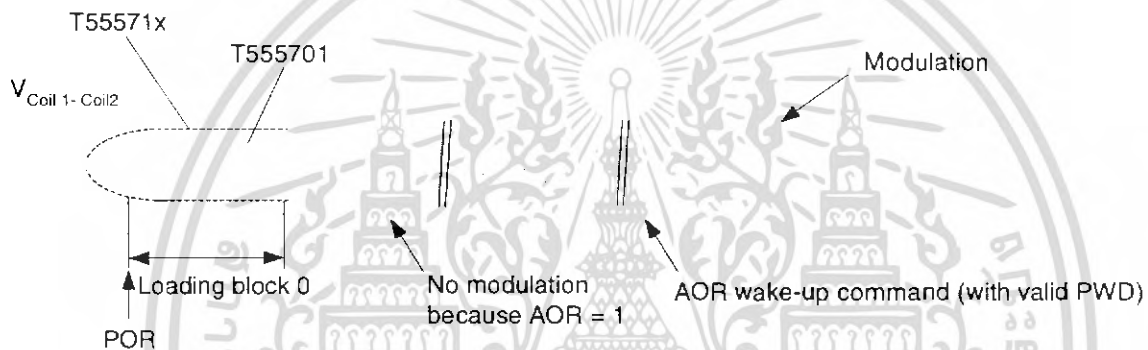
**Answer-On-Request (AOR) Mode**

When the AOR bit is set, the T5557 does not start modulation in the regular-read mode after loading configuration block 0. The tag waits for a valid AOR data stream ("wake-up command") from the reader before modulation is enabled. The wake-up command consists of the opcode ('10') followed by a valid password. The selected tag will remain active until the RF field is turned off or a new command with a different password is transmitted which may address another tag in the RF field.

**Table 3. T5557 — Modes of Operation**

PWD	AOR	Behavior of Tag after Reset Command or POR	De-activate Function
1	1	Answer-On-Request (AOR) mode: <ul style="list-style-type: none"> <li>• Modulation starts after wake-up with a matching password</li> <li>• Programming needs valid password</li> </ul>	Command with non-matching password deactivates the selected tag
1	0	Password mode: <ul style="list-style-type: none"> <li>• Modulation in regular-read mode starts after reset</li> <li>• Programming and direct access needs valid password</li> </ul>	
0	--	Normal mode: <ul style="list-style-type: none"> <li>• Modulation in regular-read mode starts after reset</li> <li>• Programming and direct access without password</li> </ul>	

**Figure 12. Answer-On-Request (AOR) Mode**



**Figure 13. Coil Voltage after Programming of a Memory Block**

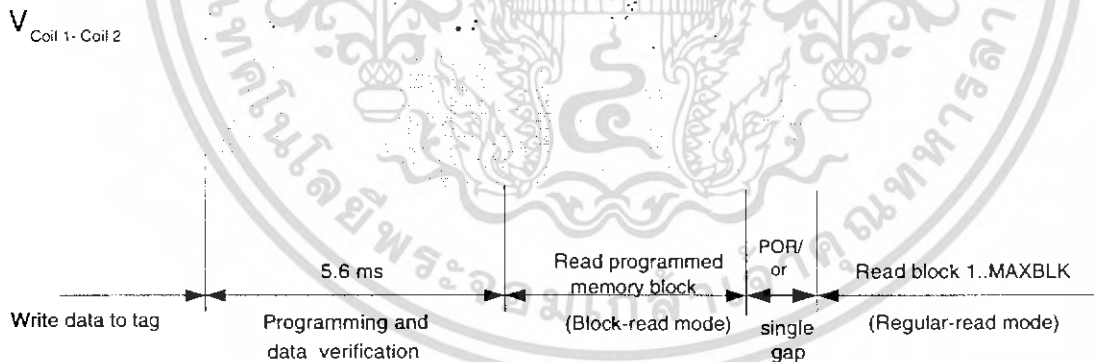
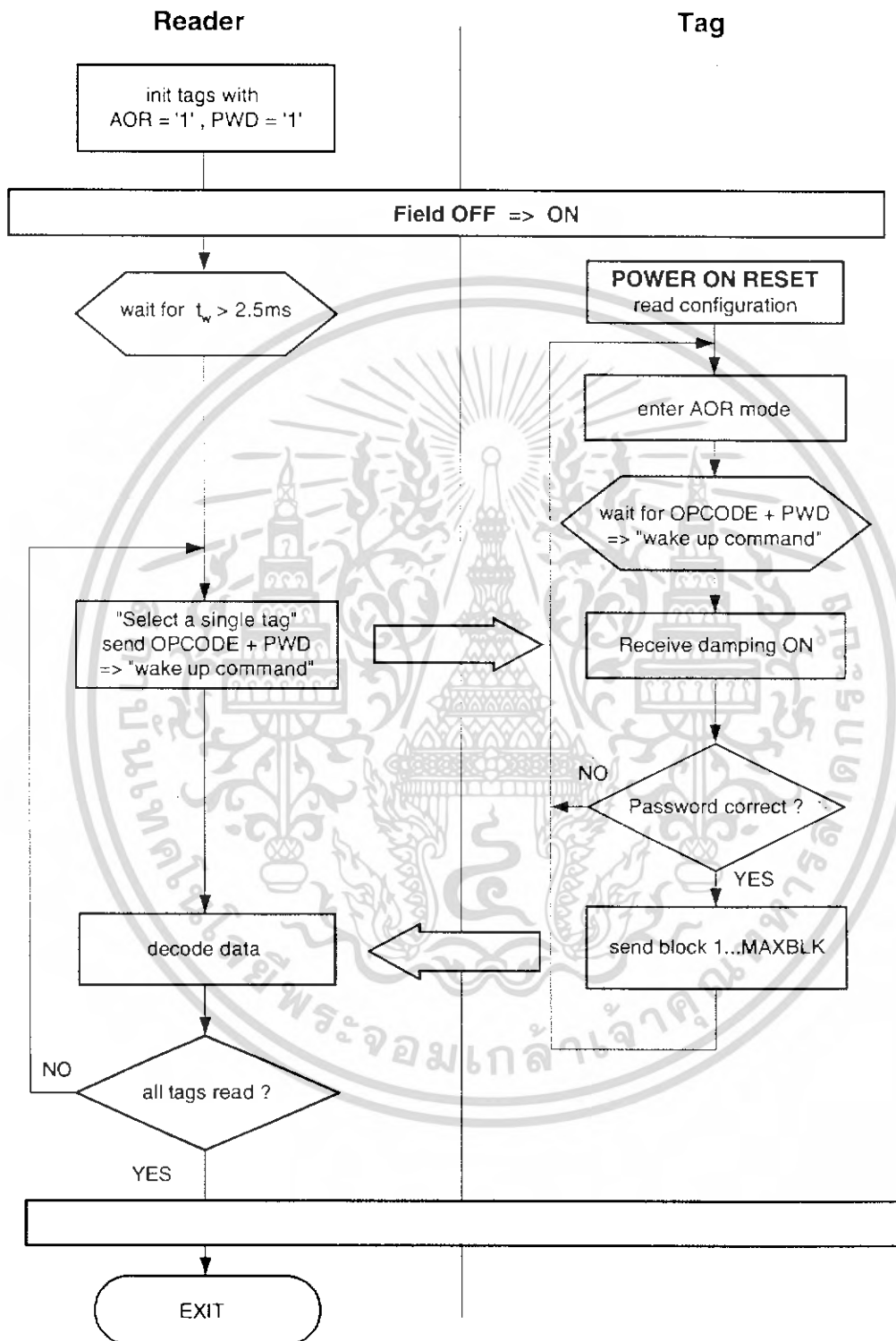


Figure 14. Anticollision Procedure Using AOR Mode





## Programming

When all necessary information has been received by the T5557, programming may proceed. There is a clock delay between the end of the writing sequence and the start of programming.

Typical programming time is 5.6 ms. This cycle includes a data verification read to grant secure and correct programming. After programming was executed successfully, the T5557 enters block-read mode transmitting the block just programmed (see Figure 13 on page 10).

Note: This timing and behavior is different from the e555x-family predecessors.

## Error Handling

Several error conditions can be detected to ensure that only valid bits are programmed into the EEPROM. There are two error types, which lead to two different actions.

### Errors During Writing

The following detectable errors could occur during writing data into the T5557:

- Wrong number of field clocks between two gaps (i.e., not a valid '1' or '0' pulse stream)
- Password mode is activated and the password does not match the contents of block 7
- The number of bits received in the command sequence is incorrect

Valid bit counts accepted by the T5557 are:

Password write	70 bits	(PWD = 1)
Standard write	38 bits	(PWD = 0)
AOR wake up	34 bits	(PWD = 1)
Direct access with PWD	38 bits	(PWD = 1)
Direct access	6 bits	(PWD = 0)
Reset command	2 bits	
Page 0/1 regular-read	2 bits	

If any of these erroneous conditions were detected, the T5557 enters regular-read mode, starting with block 1 of the page defined in the command sequence.

### Errors Before/During Programming

If the command sequence was received successfully, the following error could still prevent programming:

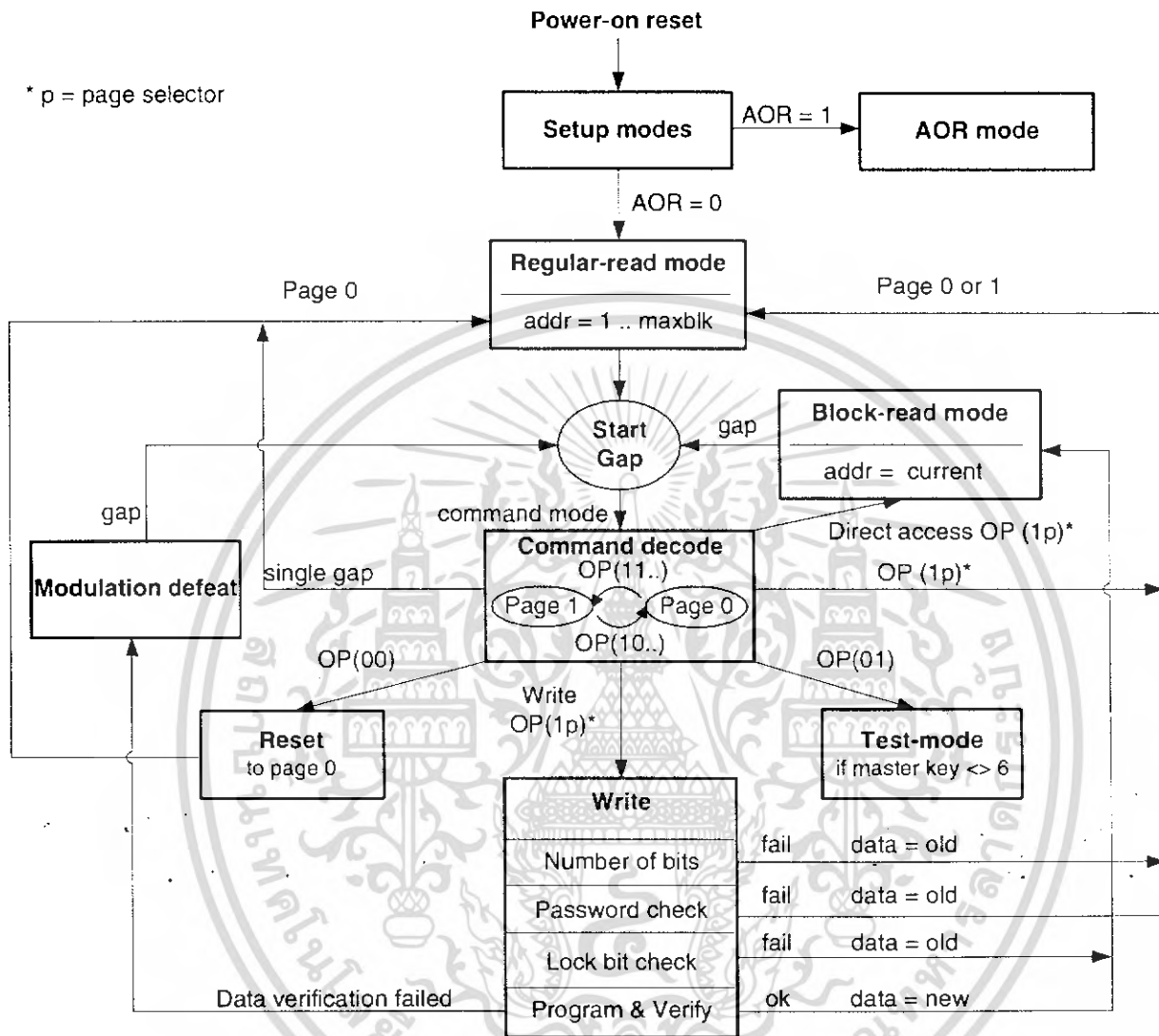
- The lock bit of the addressed block is set already
- In case of a locked block, programming mode will not be entered. The T5557 reverts to block-read mode continuously transmitting the currently addressed block.

If the command sequence is validated and the addressed block is not write protected, the new data will be programmed into the EEPROM memory. The new state of the block write protection bit (lock bit) will be programmed at the same time accordingly.

Each programming cycle consists of 4 consecutive steps: erase block, erase verification (data = '0'), programming, write verification (corresponding data bits = '1').

- If a data verification error is detected after an executed data block programming, the tag will stop modulation (modulation defeat) until a new command is transmitted.

Figure 15. T5557 Functional Diagram



**T5557 in Extended Mode (X-mode)**

In general, the block 0 setting of the master key (bits 1 to 4) to the value '6' or '9' together with the X-mode bit will enable the extended mode functions.

- Master key = '9': Test mode access and extended mode are both enabled.
- Master key = '6': Any test mode access will be denied but the extended mode is still enabled.

Any other master key setting will prevent the activation of the T5557 extended mode options, even when the X-mode bit is set.

**Binary Bit-rate Generator**

In extended mode the data rate is binary programmable to operate at any data rate between RF/2 and RF/128 as given in the formula below.

$$\text{Data rate} = \text{RF}/(2n+2)$$





### OTP Functionality

If the OTP bit is set to '1', all memory blocks are write protected and behave as if all lock bits are set to 1. If the master key is set to '6' additionally, the T5557 mode of operation is locked forever (= OTP functionality).

If the master key is set to '9', the test-mode access allows the re-configuration of the tag again.

Figure 16. Block 0 — Configuration Map in Extended Mode (X-mode)

L	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32							
	1	0	0	1	0	0	0	0							1																								
Lock Bit	Master Key Note 1), 2)				Data Bit Rate RF/(2n+2)						X-Mode	Modulation				PSK- CF	AOR	OTP	MAX- BLOCK	PWD	SST-Sequence Start Marker	Fast write	Inverse Data	POR-Delay															
	0	Unlocked				Direct	0	0	0	0	0	0	0	0	0	0	0	1	RF/4																				
	1	Locked				PSK1	0	0	0	0	0	1	1	0	RF/8																								
						PSK2	0	0	0	1	0	1	1	Res.																									
						PSK3	0	0	0	1	1																												
						FSK1	0	0	1	0	0																												
						FSK2	0	0	1	0	1																												
						Manchester	0	1	0	0	0																												
						Biphase ('50)	1	0	0	0	0																												
						Biphase ('57)	1	1	0	0	0																												

1) If Master Key = 6 and bit 15 set, then test-mode access is disabled and extended mode is active  
 2) If Master Key = 9 and bit 15 set, then extended mode is enabled

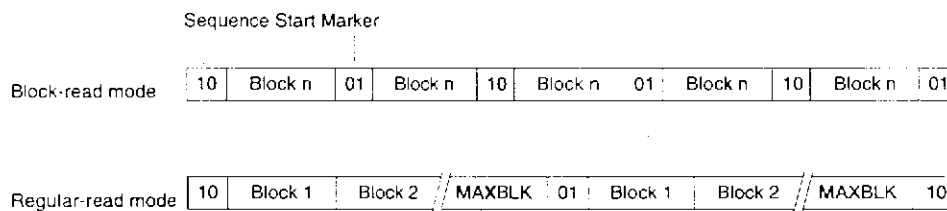
Table 4. T5557 Types of Modulation in Extended Mode

Mode	Direct Data Output Encoding	Inverse Data Output Encoding
FSK1 <sup>(1)</sup>	FSK/5-/8 '0' = RF/5; '1' = RF/8	FSK/8-/5 '0' = RF/8; '1' = RF/5 (= FSK1a)
FSK2 <sup>(1)</sup>	FSK/10-/8 '0' = RF/10; '1' = RF/8	FSK/8-/10 '0' = RF/8; '1' = RF/10 (= FSK2a)
PSK1 <sup>(2)</sup>	Phase change when input changes	Phase change when input changes
PSK2 <sup>(2)</sup>	Phase change on bit clock if input high	Phase change on bit clock if input low
PSK3 <sup>(2)</sup>	Phase change on rising edge of input	Phase change on falling edge of input
Manchester	'0' = falling edge, '1' = rising edge on mid-bit	'1' = falling edge, '1' = rising edge on mid-bit
Biphase 1 ('50)	'1' creates an additional mid-bit change	'0' creates an additional mid-bit change
Biphase 2 ('57)	'0' creates an additional mid-bit change	'1' creates an additional mid-bit change
NRZ	'1' = damping on, '0' = damping off	'0' = damping on, '1' = damping off

- Notes: 1. A common multiple of bitrate and FSK frequencies is recommended.  
 2. In PSK mode the selected data rate has to be an integer multiple of the PSK sub-carrier frequency.

Sequence Start Marker

Figure 17. T5557 Sequence Start Marker in Extended Mode

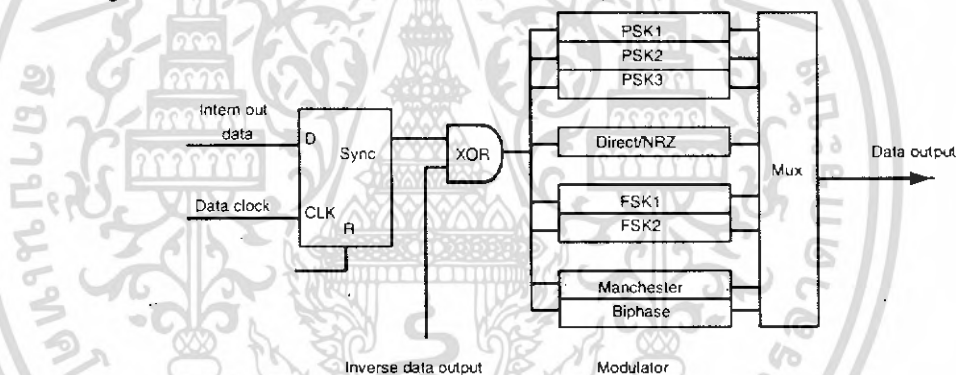


The T5557 sequence start marker is a special damping pattern, which may be used to synchronize the reader. The sequence start marker consists of two bits ('01' or '10') which are inserted as header before the first block to be transmitted if the bit 29 in extended mode is set. At the start of a new block sequence, the value of the two bits is inverted.

Inverse Data Output

The T5557 supports in its extended mode (X-mode) an inverse data output option. If inverse data is enabled, the modulator as shown in Figure 18 works on inverted data (see Table 4 on page 14). This function is supported for all basic types of encoding.

Figure 18. Data Encoder for Inverse Data Output



Fast Write

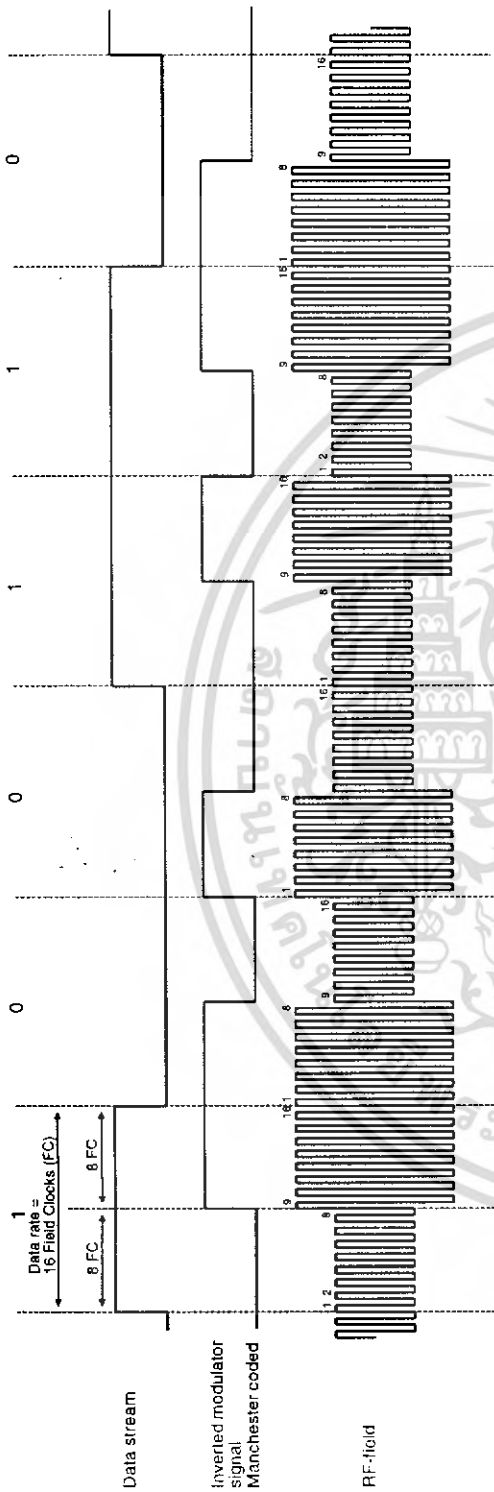
In the optional fast write mode the time between two gaps is nominally 12 field clocks for a '0' and 27 field clocks for a '1'. When there is no gap for more than 32 field clocks after a previous gap, the T5557 will exit the write mode. Please refer to Table 5 and Figure 8 on page 7.

Table 5. Fast Write Data Decoding Schemes

Parameters	Remark	Symbol	Min.	Max.	Unit
Start gap	—	$S_{gap}$	10	50	FC
Write gap	Normal write mode	$Wn_{gap}$	8	30	FC
	Fast write mode	$Wf_{gap}$	8	20	FC
Write data in normal mode	'0' data	$d_0$	16	31	FC
	'1' data	$d_1$	48	63	FC
Write data in fast mode	'0' data	$d_0$	8	15	FC
	'1' data	$d_1$	24	31	FC



**Figure 19.** Example of Manchester Coding with Data Rate RF/16



**Figure 20.** Example of Biphase Coding with Data Rate RF/16

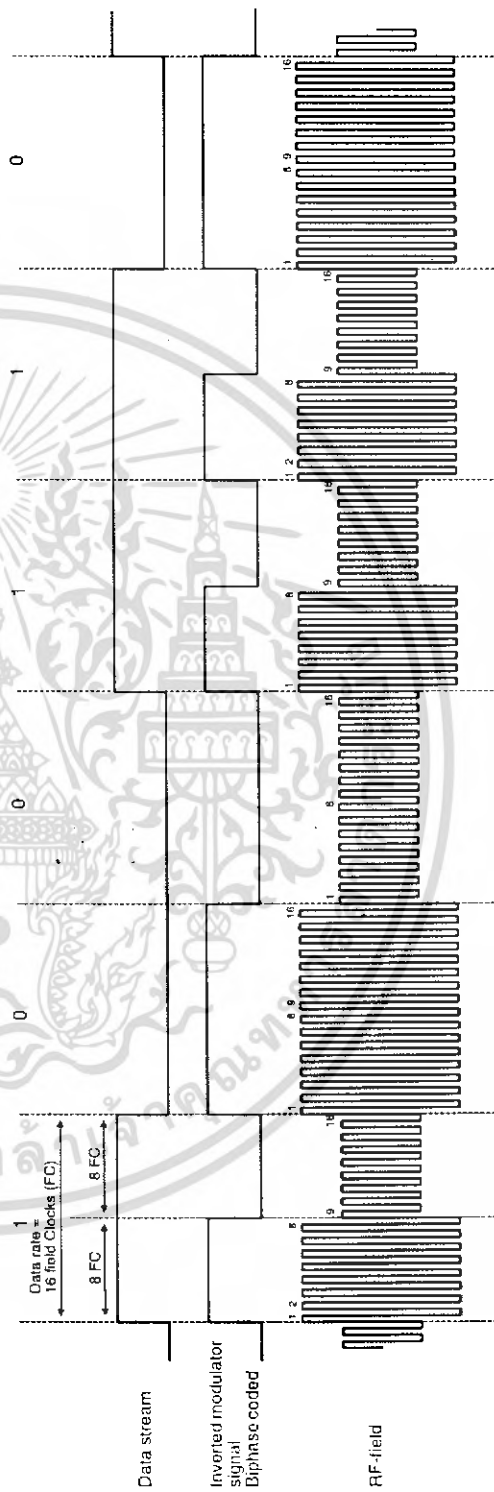


Figure 21. Example: FSK1a Coding with Data Rate RF/40, Subcarrier  $f_0 = RF/8$ ,  $f_1 = RF/5$

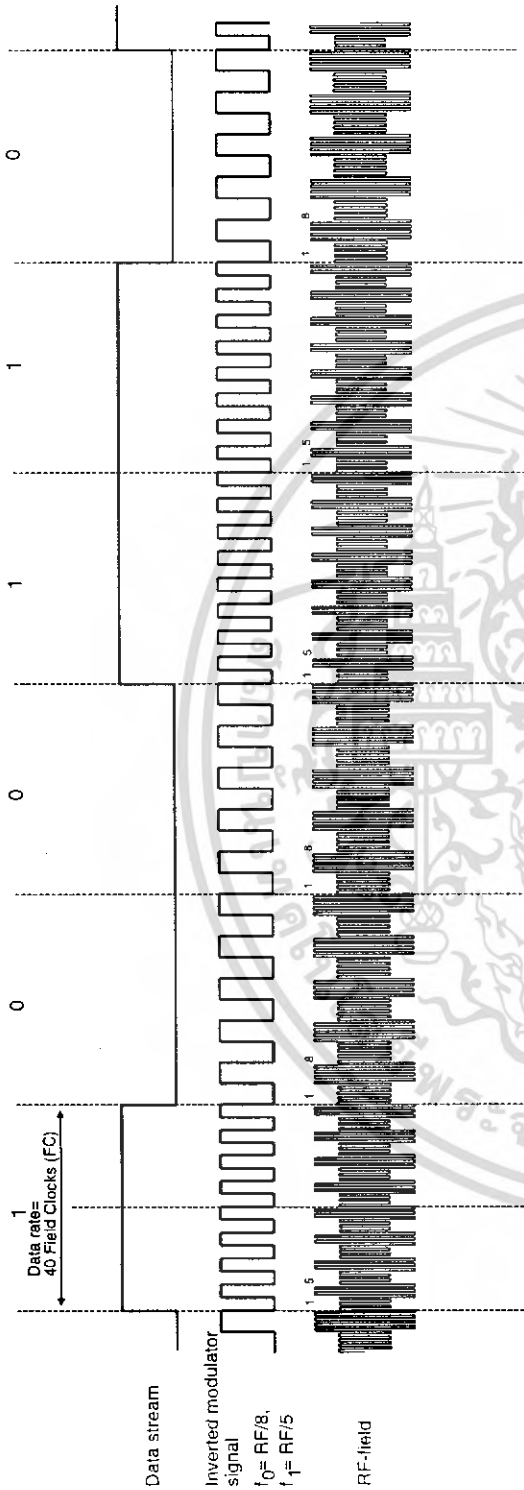
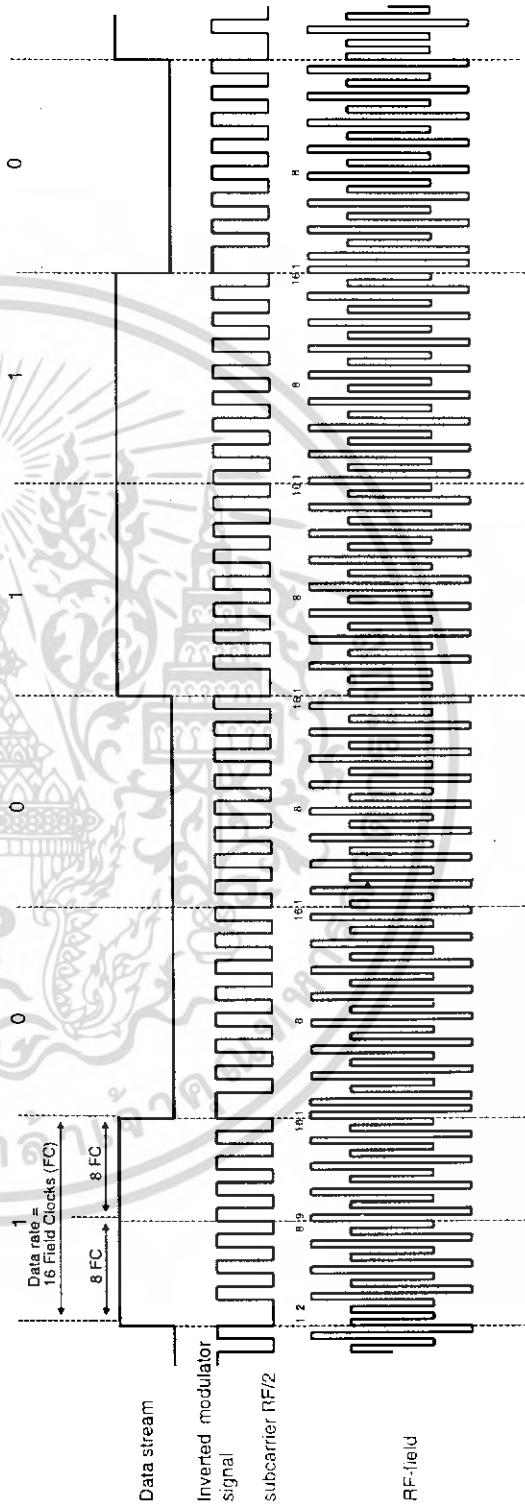
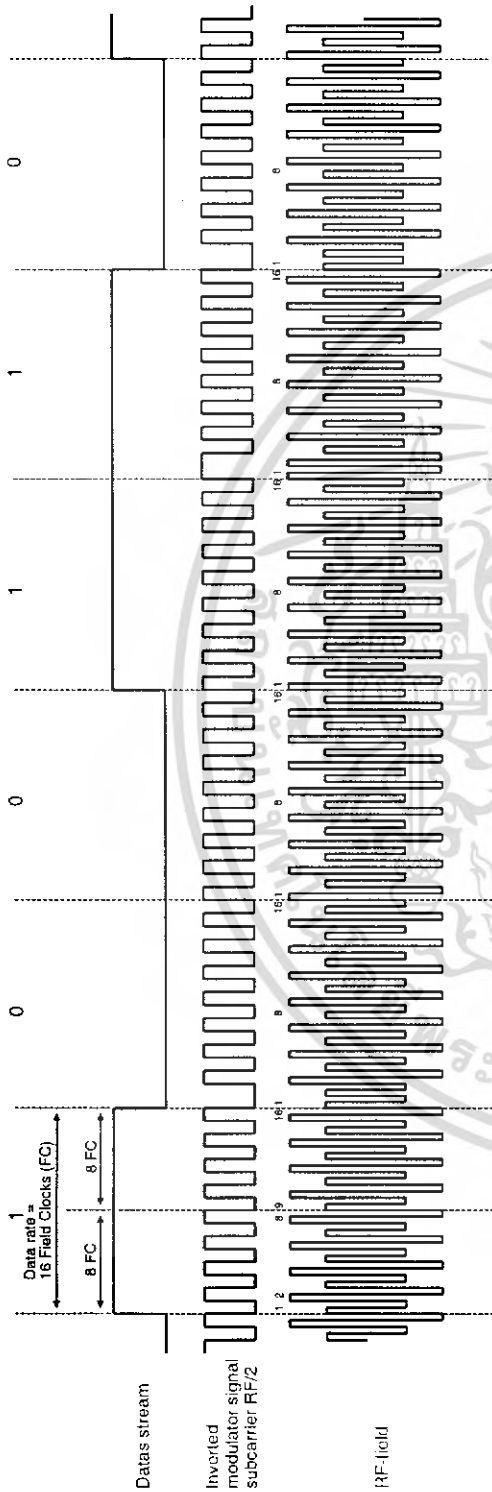


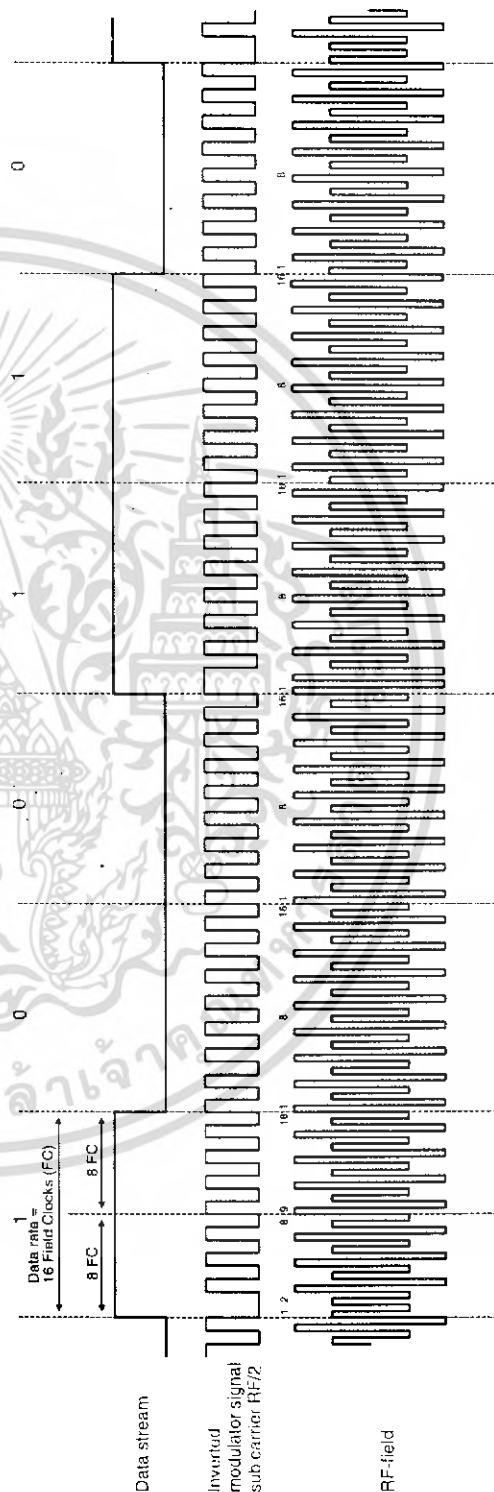
Figure 22. Example of PSK1 Coding with Data Rate RF/16



**Figure 23.** Example of PSK2 Coding with Data Rate RF/16



**Figure 24.** Example of PSK3 Coding with Data Rate RF/16



## Absolute Maximum Ratings

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Parameters	Symbol	Value	Unit
Maximum DC current into Coil 1/Coil 2	$I_{coil}$	20	mA
Maximum AC current into Coil 1/Coil 2 $f = 125$ kHz	$I_{coil p}$	20	mA
Power dissipation (dice) (free-air condition, time of application: 1 s)	$P_{tot}$	100	mW
Electrostatic discharge maximum to MIL-Standard 883 C method 3015	$V_{max}$	4000	V
Operating ambient temperature range	$T_{amb}$	-40 to +85	°C
Storage temperature range (data retention reduced)	$T_{stg}$	-40 to +150	°C

## Electrical Characteristics

$T_{amb} = +25^{\circ}\text{C}$ ;  $f_{coil} = 125$  kHz; unless otherwise specified

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type*
1	RF frequency range		$f_{RF}$	100	125	150	kHz	
2.1	Supply current (without current consumed by the external LC tank circuit)	$T_{amb} = 25^{\circ}\text{C}^{(1)}$ (see Figure 24 on page 18)	$I_{DD}$		1.5	3	$\mu\text{A}$	T
2.2		Read – full temperature range			2	4	$\mu\text{A}$	Q
2.3		Programming full temperature range			25	40	$\mu\text{A}$	Q
3.1	Coil voltage (AC supply)	POR threshold (50 mV hysteresis)	$V_{coil pp}$	3.2	3.6	4.0	V	Q
3.2		Read mode and write command <sup>(2)</sup>		6		$V_{clamp}$	V	Q
3.3		Program EEPROM <sup>(2)</sup>		8		$V_{clamp}$	V	Q
4	Start-up time	$V_{coil pp} = 6$ V	$t_{startup}$		2.5	3	ms	Q
5	Clamp voltage	10 mA current into Coil 1/2	$V_{clamp}$	17		23	V	T

\* ) Type means: T: directly or indirectly tested during production; Q: guaranteed based on initial product qualification data

- Notes:
- $I_{DD}$  measurement setup  $R = 100$  k;  $V_{CLK} = V_{coil} = 5$  V; EEPROM programmed to 00 ... 000 (erase all); chip in modulation defeat.  $I_{DD} = (V_{OUTmax} - V_{CLK})/R$
  - Current into Coil 1/Coil 2 is limited to 10 mA. The damping circuitry has the same structure as the e5550. The damping characteristics are defined by the internally limited supply voltage (= minimum AC coil voltage)
  - $V_{mod}$  measurement setup:  $R = 2.3$  k;  $V_{CLK} = 3$  V; setup with modulation enabled (see Figure 25 on page 20).
  - Since EEPROM performance is influenced by assembly processes, Atmel confirms the parameters for DOW (tested dice on uncutted wafer) delivery.
  - The tolerance of the on-chip resonance capacitor  $C_r$  is  $\pm 10\%$  at  $3\sigma$  over whole production. The capacitor tolerance is  $\pm 3\%$  at  $3\sigma$  on a wafer basis.
  - The tolerance of the microcodule resonance capacitor  $C_r$  is  $\pm 5\%$  at  $3\sigma$  over whole production.



## Electrical Characteristics

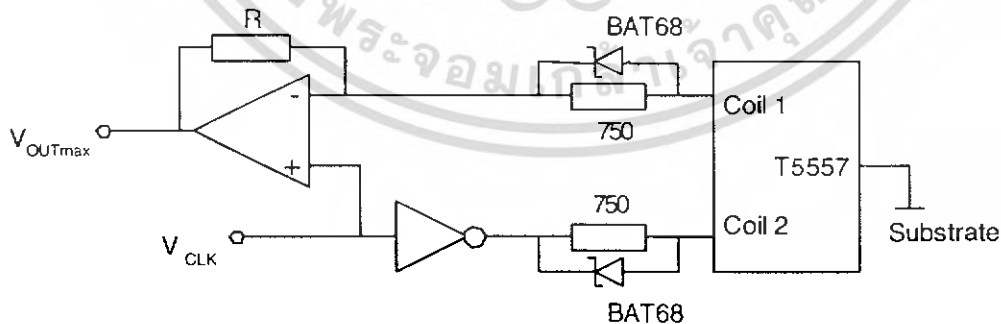
$T_{amb} = +25^{\circ}\text{C}$ ;  $f_{coil} = 125\text{ kHz}$ ; unless otherwise specified

No.	Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit	Type*
6.1	Modulation parameters	$V_{coilpp} = 6\text{ V}$ on test circuit generator and modulation ON <sup>(3)</sup>	$V_{mod\ pp}$		4.2	4.8	V	T
6.2			$I_{mod\ pp}$	400	600		$\mu\text{A}$	T
6.3			Thermal stability	$V_{mod}/T_{amb}$		-6		$\text{mV}/^{\circ}\text{C}$
7	Programming time	From last command gap to re-enter read mode (64 + 648 internal clocks)	$T_{prog}$	5	5.7	6	ms	T
8	Endurance	Erase all / Write all <sup>(4)</sup>	$n_{cycle}$	100000			Cycles	Q
9.1	Data retention	Top = $55^{\circ}\text{C}$ <sup>(4)</sup>	$t_{retention}$	10	20	50	Years	
9.2		Top = $150^{\circ}\text{C}$ <sup>(4)</sup>	$t_{retention}$	96			hrs	T
9.3		Top = $250^{\circ}\text{C}$ <sup>(4)</sup>	$t_{retention}$	24			hrs	Q
10	Resonance capacitor	Mask option <sup>(5)</sup>	$C_r$	70	78	86	pF	T
11.1	Microdual capacitor parameters	Capacitance tolerance	$C_r$	313.5	330	346.5	pF	T
11.2		Temperature coefficient	TBD	TBD	TBD	TBD	TBD	TBD
11.3			TBD	TBD	TBD	TBD	TBD	TBD

\* Type means: T: directly or indirectly tested during production; Q: guaranteed based on initial product qualification data

- Notes:
- $I_{DD}$  measurement setup  $R = 100\text{ k}$ ;  $V_{CLK} = V_{coil} = 5\text{ V}$ ; EEPROM programmed to 00 ... 000 (erase all); chip in modulation defeat.  $I_{DD} = (V_{OUTmax} - V_{CLK})/R$
  - Current into Coil 1/Coil 2 is limited to 10 mA. The damping circuitry has the same structure as the e5550. The damping characteristics are defined by the internally limited supply voltage (= minimum AC coil voltage)
  - $V_{mod}$  measurement setup:  $R = 2.3\text{ k}$ ;  $V_{CLK} = 3\text{ V}$ ; setup with modulation enabled (see Figure 25 on page 20).
  - Since EEPROM performance is influenced by assembly processes, Atmel confirms the parameters for DOW (tested dice on uncutted wafer) delivery.
  - The tolerance of the on-chip resonance capacitor  $C_r$  is  $\pm 10\%$  at  $3\sigma$  over whole production. The capacitor tolerance is  $\pm 3\%$  at  $3\sigma$  on a wafer basis.
  - The tolerance of the microdual resonance capacitor  $C_r$  is  $\pm 5\%$  at  $3\sigma$  over whole production.

Figure 25. Measurement Setup for  $I_{DD}$  and  $V_{mod}$



Ordering Information<sup>(2)</sup>

T 5 5 5 7	a b	M c c	- x x x	<b>Package</b> - DDW - Dice on wafer, 6" un-sawn wafer, thickness 300 µm - DDT - Dice in Tray (waffle pack), thickness 300 µm - DBW - Dice on solder bumped wafer, thickness 390 µm Sn63Pb37 on 5 µm Ni/Au, height 70 µm - TAS - SO8 Package - PAE - MOA2 Micro-Module	<b>Drawing</b>  see Figure 27 on page 23 see Figure 28 on page 23 see Figure 31 on page 26 see Figure 29 on page 24
				<b>Customer ID<sup>(1)</sup></b> - Atmel standard (corresponds to "00") M01 - Customer 'X' unique ID code <sup>(1)</sup>	
			11	- 2 Pads without on-chip C	see Figure 26 on page 22
			14	- 4 Pads with on-chip 75 pF	see Figure 27 on page 23
15	- Micro - Module with 330 pF	see Figure 29 on page 24			
01	- 2 Pads without C; Damping during initialisation	see Figure 26 on page 22			

- Notes: 1. Unique customer ID code programming according to Figure 5 is linked to a minimum order quantity of 1 Mio parts per year.  
 2. For available order codes refer to Atmel Sales/Marketing.

Ordering Examples  
(Recommended)

T555711-DDW Tested dice on unsawn 6" wafer, thickness 300 µm, no on-chip capacitor, no damping during POR initialisation; especially for ISO 11784/785 and access control applications

Available Order Codes

T555711-DDW, DDT, TAS  
 T555714-DDW, DBW, TAS  
 T555715-PAE



## Package Information

Figure 26. 2 Pad Layout for Wire Bonding

Dimensions in  $\mu\text{m}$

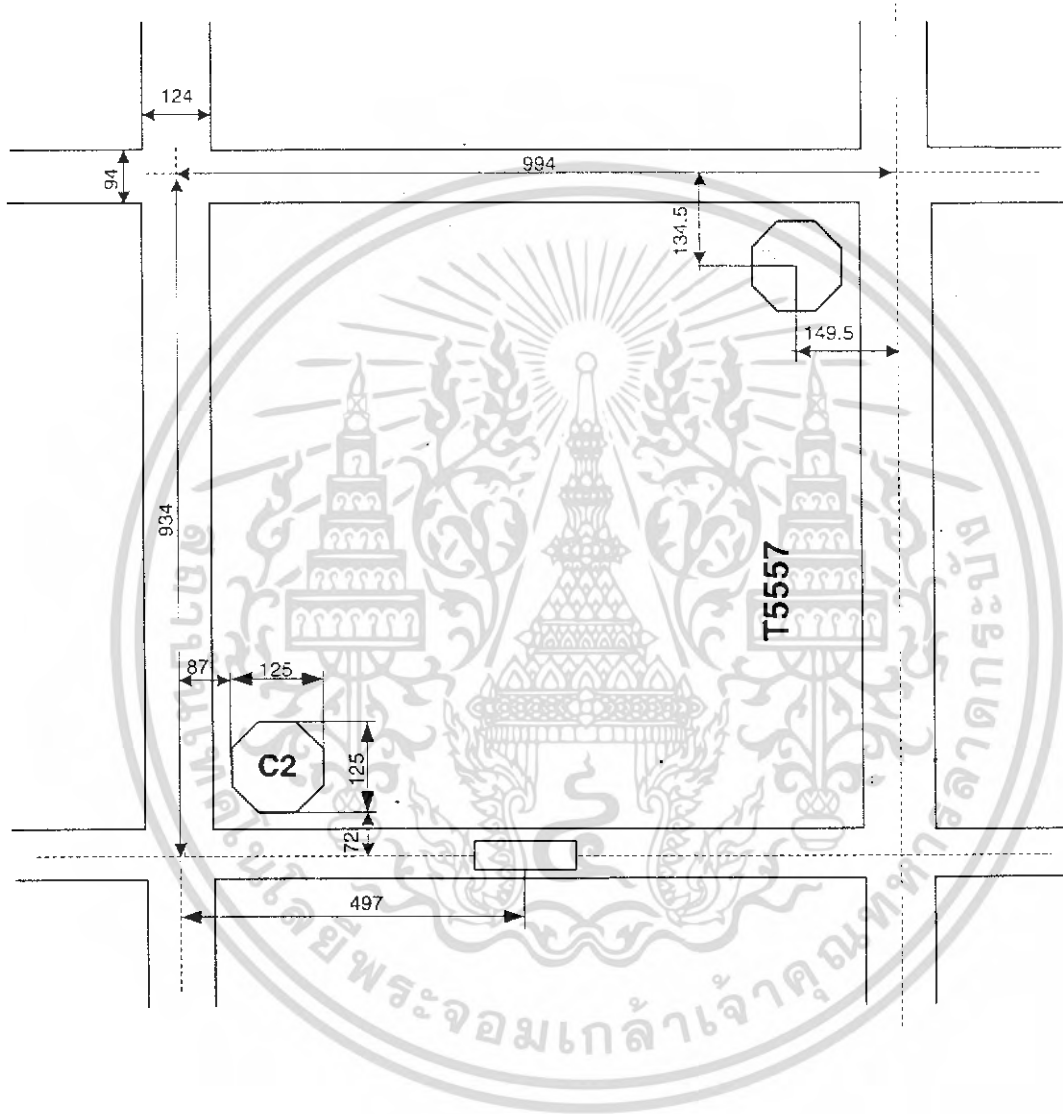


Figure 27. 4 Pad Flip-chip Version with 70  $\mu\text{m}$  Solder Bumps  
 Dimensions in  $\mu\text{m}$

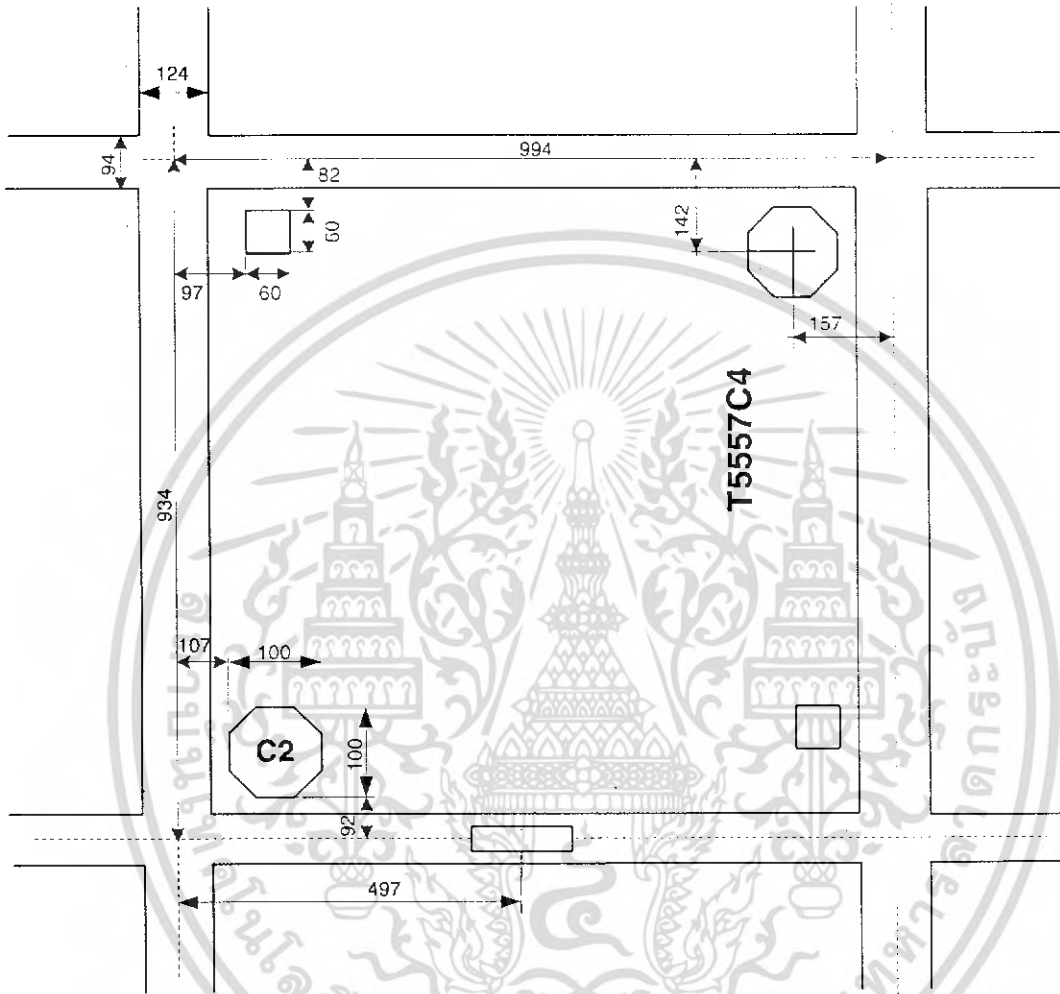


Figure 28. Solder bump on NiAu

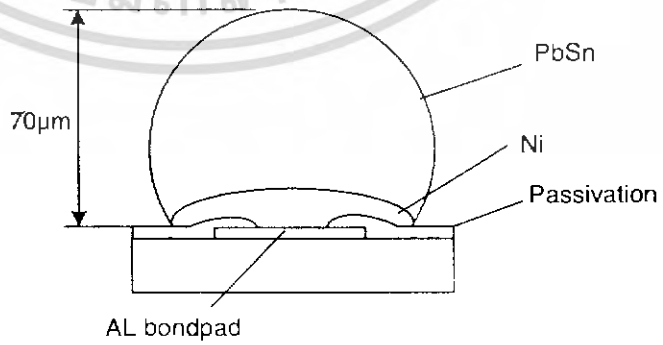




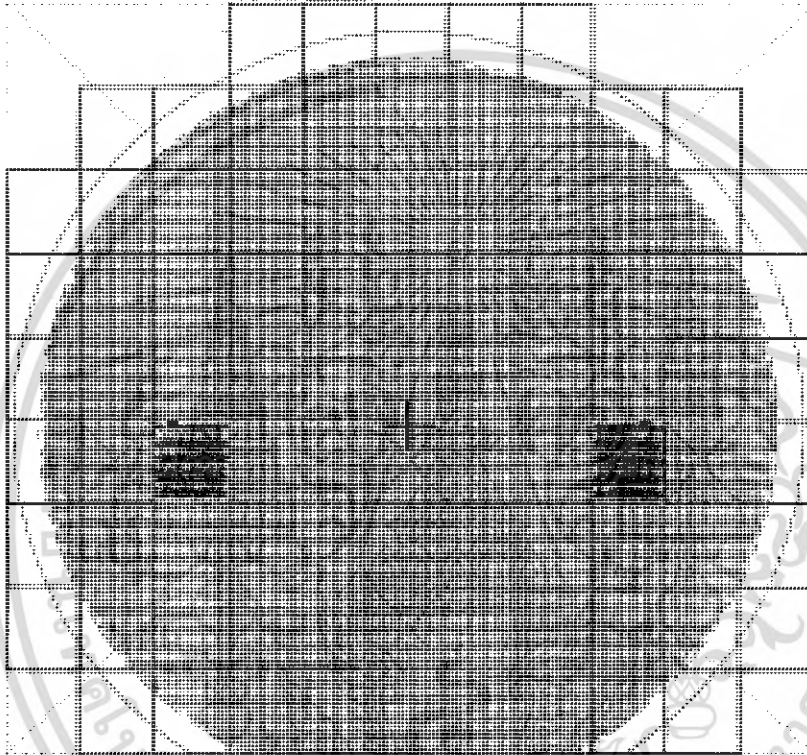
Figure 29. Wafer Map

6061 [z95]

# WAFER MAP

automap V2.7  
05-OCT-2001 15:41:29

CD MEASURE SHOTS



Flat Edge

Die: 0.894x0.864, Step: 0.994x0.934, N: 14x17, Frame Step: 13.916x15.878

> Shift-ASML=[0.3;-6.9] : 15539 dice, 87 shots (11cols x 9rows)

> Shift-CANON/ALARM/SEM=[0.3;-6.9] - W2=[-13.152;6.9] - W1=[-6.648;6.9]

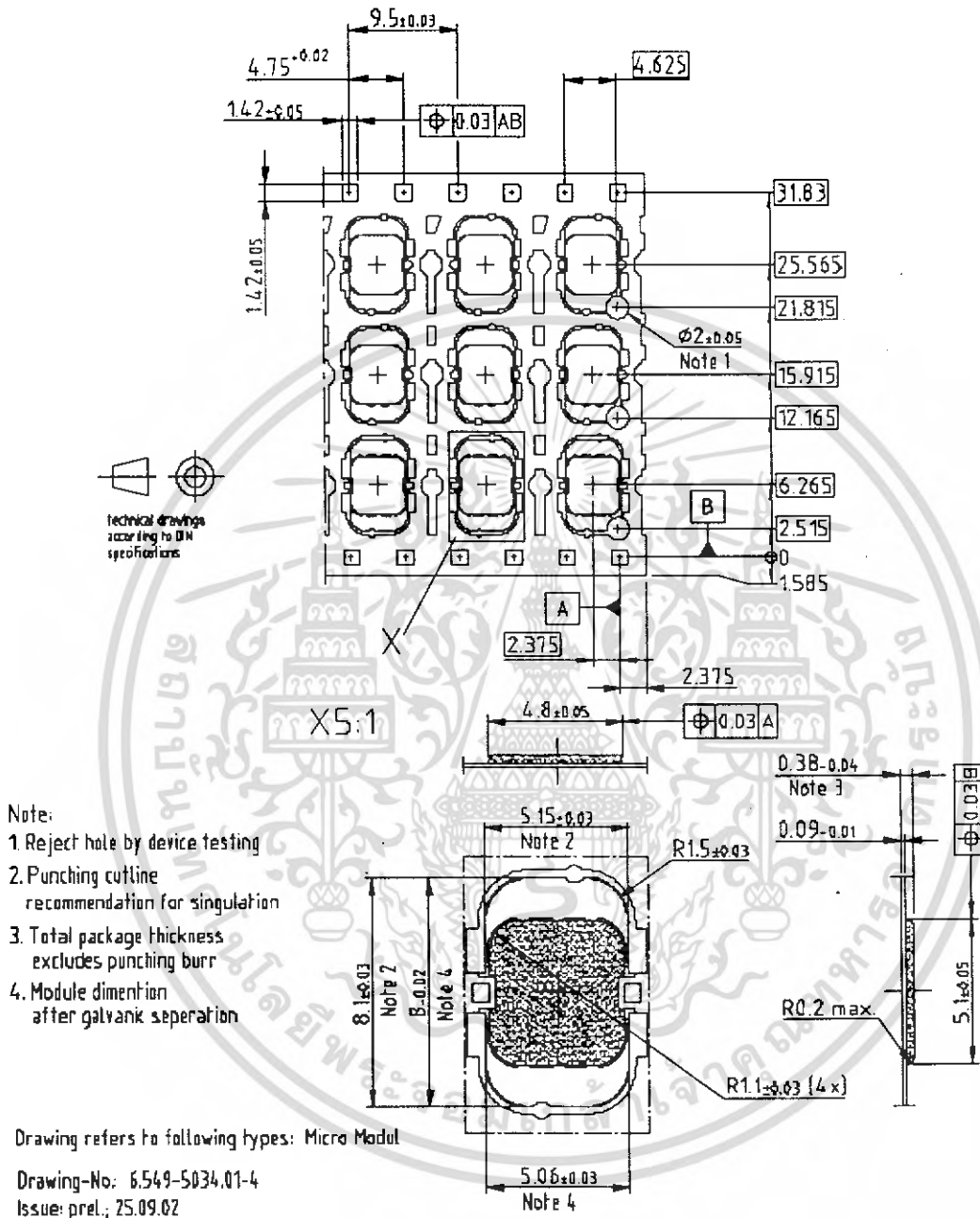
## Failed Die Identification

Every die on the wafer not passing Atmel test sequence is marked with inch. The inch dot specification:

- Dot size: 200  $\mu$ m
- Position: center of die
- Color: black



Figure 30. NOA2 Micromodule



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 31. Shipping Reel

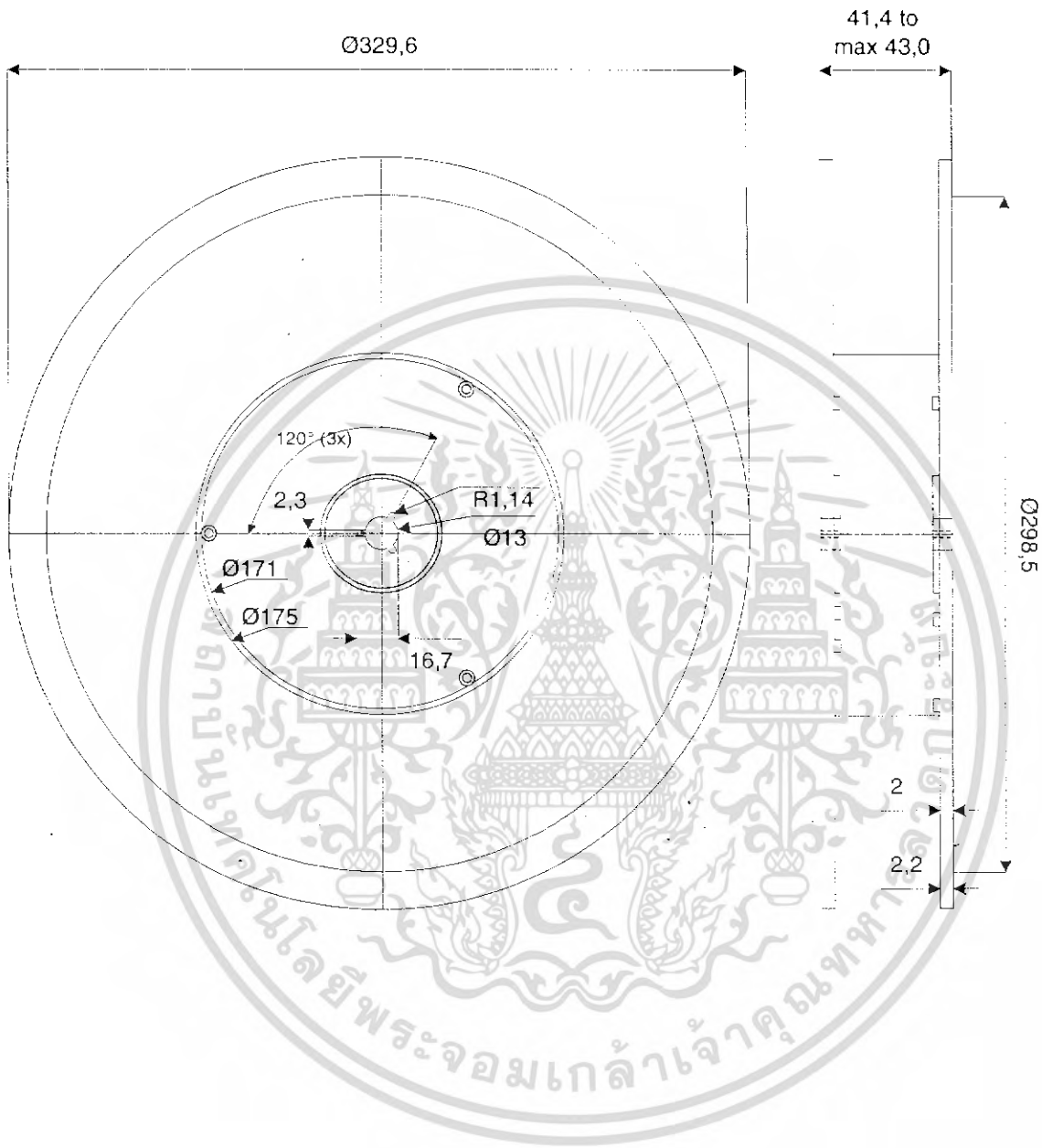


Figure 32. SO8 Package

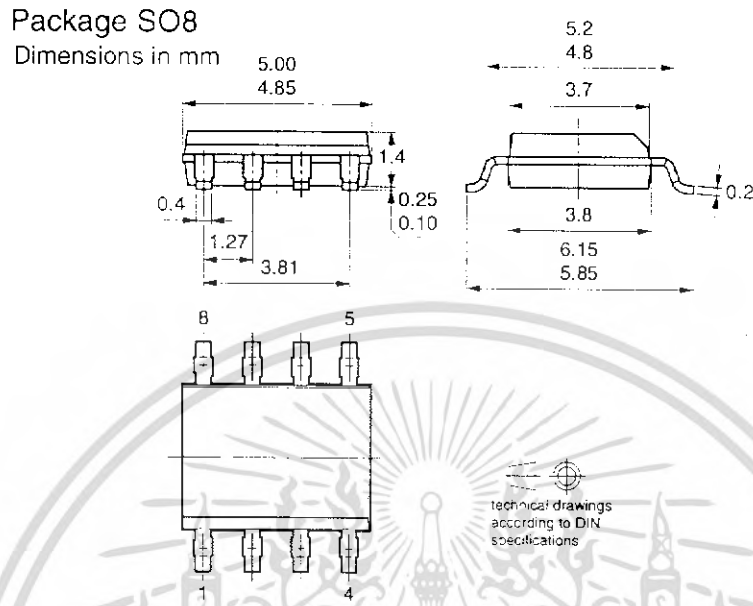
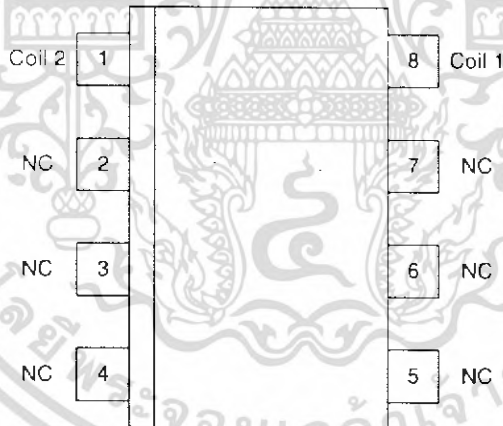


Figure 33. Pinning SO8







## Atmel Corporation

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 487-2600

## Regional Headquarters

### Europe

Atmel Sarl  
Route des Arsenaux 41  
Case Postale 80  
CH-1705 Fribourg  
Switzerland  
Tel: (41) 26-426-5555  
Fax: (41) 26-426-5500

### Asia

Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimshatsui  
East Kowloon  
Hong Kong  
Tel: (852) 2721-9778  
Fax: (852) 2722-1369

### Japan

9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
Tel: (81) 3-3523-3551  
Fax: (81) 3-3523-7581

## Atmel Operations

### Memory

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 436-4314

### Microcontrollers

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 436-4314

La Chantrerie  
BP 70602  
44306 Nantes Cedex 3, France  
Tel: (33) 2-40-18-18-18  
Fax: (33) 2-40-18-19-60

### ASIC/ASSP/Smart Cards

Zone Industrielle  
13106 Rousset Cedex, France  
Tel: (33) 4-42-53-60-00  
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906, USA  
Tel: 1(719) 576-3300  
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park  
Maxwell Building  
East Kilbride G75 0QR, Scotland  
Tel: (44) 1355-803-000  
Fax: (44) 1355-242-743

### RF/Automotive

Theresienstrasse 2  
Postfach 3535  
74025 Heilbronn, Germany  
Tel: (49) 71-31-67-0  
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906, USA  
Tel: 1(719) 576-3300  
Fax: 1(719) 540-1759

### Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
Tel: (33) 4-76-58-30-00  
Fax: (33) 4-76-58-34-80

### Literature Requests

[www.atmel.com/literature](http://www.atmel.com/literature)

**Disclaimer:** Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© Atmel Corporation 2003. All rights reserved.

Atmel® and combinations thereof are the registered trademarks of Atmel Corporation or its subsidiaries.

Other terms and product names may be the trademarks of others.

Printed on recycled paper.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4517F-RFID-11/03



# Read/Write analog front end for 125kHz RFID Basestation

### Description

The EM4095 (previously named P4095) chip is a CMOS integrated transceiver circuit intended for use in an RFID basestation to perform the following functions:

- antenna driving with carrier frequency
- AM modulation of the field for writable transponder
- AM demodulation of the antenna signal modulation induced by the transponder

communicate with a microprocessor via simple interface.

- Data transmission by Amplitude Modulation with externally adjustable modulation index using single ended driver
- Multiple transponder protocol compatibility (Ex: EM400X, EM4050, EM4150, EM4070, EM4170, EM4069 .)
- Sleep mode 1µA
- USB compatible power supply range
- 40 to +85°C temperature range
- Small outline plastic package SO16

### Features

- Integrated PLL system to achieve self adaptive carrier frequency to antenna resonant frequency
- No external quartz required
- 100 to 150 kHz carrier frequency range
- Direct antenna driving using bridge drivers
- Data transmission by OOK (100% Amplitude Modulation) using bridge driver

### Applications

- Car immobiliser
- Hand held reader
- Low cost reader

### Typical Operating Configuration

Read Only Mode

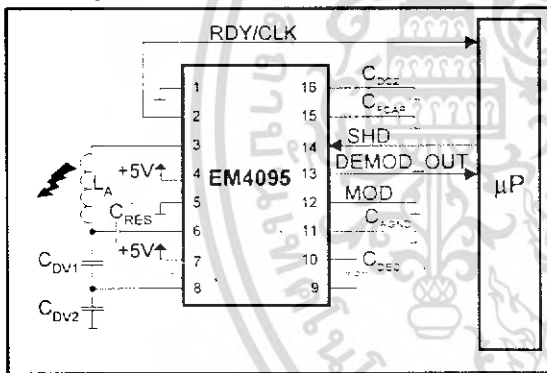


Fig. 1

### Pin Assignment

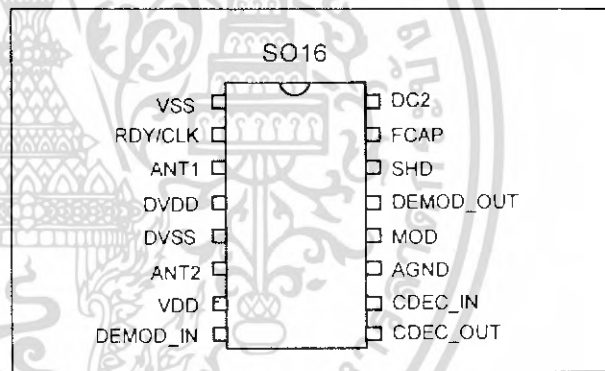


Fig. 3

### Read/Write Mode

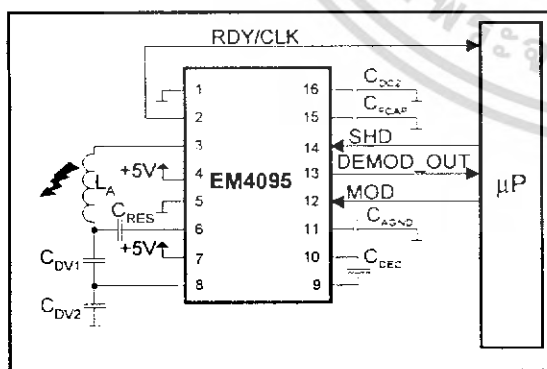
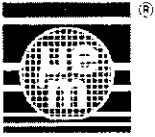


Fig. 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



# EM4095

## System principle

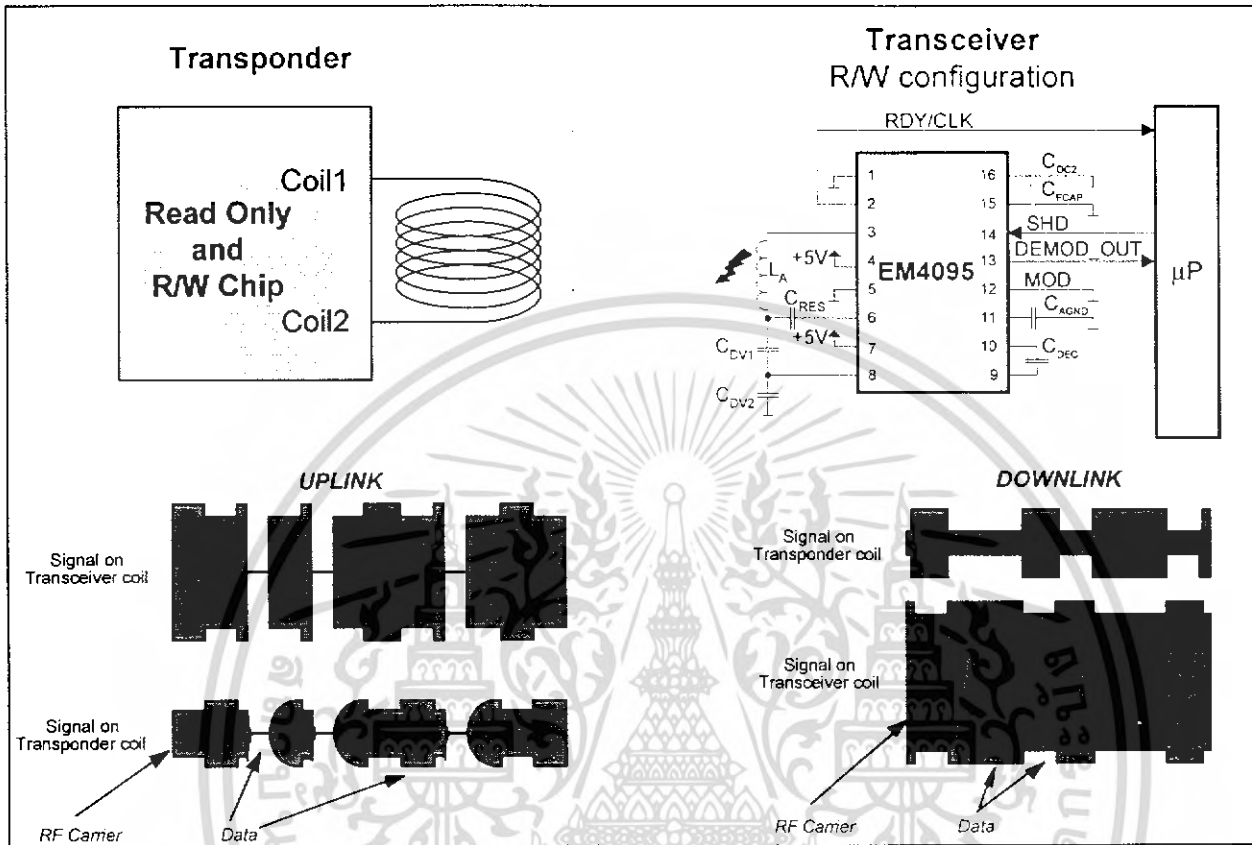
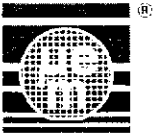


Fig. 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า



# EM4095

## Absolute Maximum Ratings

Parameter	Symbol	Conditions
Storage temperature	$T_{S-C}$	-55 to +150°C
Maximum voltage at V <sub>DD</sub>	$V_{DD-max}$	V <sub>SS</sub> +6V
Minimum voltage at V <sub>DD</sub>	$V_{DD-min}$	V <sub>SS</sub> -0.3V
Max. voltage other pads	$V_{VAX}$	V <sub>DD</sub> +0.3V
Min. voltage other pads	$V_{VIN}$	V <sub>SS</sub> -0.3V
Max. junction temperature	$T_{JMAX}$	+125°C
Electrostatic discharge max. to MIL-STD-883C method 3015 against V <sub>SS</sub>	$V_{ESD}$	4000V
Electrostatic discharge max. to MIL-STD-883C method 3015 (only for pins ANT1 and ANT2) against V <sub>SS</sub>	$V_{ESD\_ANT}$	10000V
Maximum Input/Output current on all pads except V <sub>DD</sub> , V <sub>SS</sub> , DV <sub>DD</sub> , DV <sub>SS</sub> , ANT1, ANT2, RDY/CLK	$I_{IMAX}$ $I_{OMAX}$	10mA
Maximum AC peak current on ANT1 and ANT2 pads 100 kHz duty cycle 50%	$I_{ANTmax}$	300mA

Stresses above these listed maximum ratings may cause permanent damages to the device. Exposure beyond specified operating conditions may affect device reliability or cause malfunction.

## Handling Procedures

This device has built-in protection against high static voltages or electric fields; however, anti-static precautions must be taken as for any other CMOS component. Unless otherwise specified, proper operation can only occur when all terminal voltages are kept within the voltage range. Unused inputs must always be tied to a defined logic voltage level.

## Operating Conditions

Parameter	Symb	Min	Typ	Max	Units
Operating junction temperature	$T_J$	-40		+110	°C
Supply voltage	$V_{DD}$	4.1	5	5.5	V
Antenna circuit resonant frequency	$F_{RES}$	100	125	150	kHz
AC peak current on ANT1 & ANT2 pads	$I_{ANT}$			250	mA
$C_{FCAP}$		*	10	*	nF
$C_{DEC}$		*	100	*	nF
$C_{DC2}$		*	6.80	*	nF
$C_{AGND}$		100		220	nF
Package thermal resistor SO16	$R_{th-ja}$	69	70	71	°C/W

\* ±10% tolerance capacitors should be used

\*\* According to 1S2P JEDEC test board

Due to antenna driver current the internal junction temperature is higher than ambient temperature. Please calculate ambient temperature range from max. antenna current and package Thermal Resistor. It is the user's responsibility to guarantee that  $T_J$  remains below 110°C.

Supply voltage (V<sub>DD</sub> and DV<sub>DD</sub> pads) must be blocked by a 100nF capacitor (to V<sub>SS</sub>) as close as possible to the chip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



# EM4095

## Electrical and Switching Characteristics:

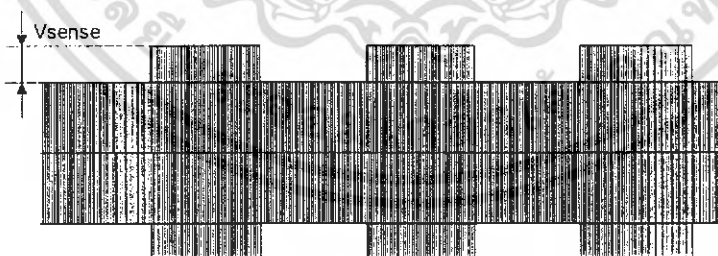
Parameters specified below are valid only in case the device is used according to Operating Conditions defined on previous page.

$V_{SS}=D_{VSS}=0V$ ,  $V_{DD}=D_{VDD}=5V$ ,  $T_j = -40$  to  $110^\circ C$ , unless otherwise specified

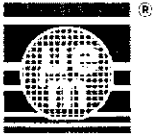
Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Supply current in sleep mode	$I_{DDsleep}$			1	2	$\mu A$
Supply current excluding drivers current	$I_{DDcn}$			5	7	mA
AGND level	$V_{AGND}$	Note 1	2.35	2.5	2.65	V
<i>Logic signals SHD, MOD, DEMOD_OUT</i>						
Input logic high	$V_{IH}$		$0.8V_{DD}$			V
Input logic low	$V_{IL}$				$0.2V_{DD}$	V
Output logic high	$V_{OH}$	$I_{SOURCE}=1mA$	$0.9V_{DD}$			V
Output logic low	$V_{OL}$	$I_{SINK}=1mA$			$0.1V_{DD}$	V
MOD pull down resistor	$R_{PD}$	$0.2V_{DD}$	20	50	90	k $\Omega$
SHD pull up resistor	$R_{PU}$	$0.8V_{DD}$	20	50	90	k $\Omega$
<i>PLL</i>						
Antenna capture frequency range	$F_{ANT\_C}$		100		150	kHz
Antenna locking frequency range	$F_{ANT\_L}$		100		150	kHz
<i>Drivers</i>						
ANT drivers output resistance	$R_{AD}$	$I_{ANT}=100mA$		3	9	$\Omega$
RDY/CLK driver output resistance	$R_{CL}$	$I_{RDY/CLK}=10mA$		12	36	$\Omega$
<i>AM demodulation</i>						
DEMOD_IN common mode range	$V_{CM}$		$V_{SS} + 0.5$		$V_{DD} - 0.5$	V
DEMOD_IN input sensitivity	$V_{sense}$	Note 2		0.85	2	mVpp

Note 1:  $A_{GND}$  is a EM4095 internal reference point. Any external connection except specified capacitor to  $V_{SS}$  may lead to device malfunction.

Note 2: Modulating signal 2Khz square wave on 125 kHz carrier, total signal inside  $V_{CM}$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



# EM4095

## Timing Characteristics:

Parameters specified below are valid only in case the device is used according to Operating Conditions defined on previous page.

$V_{SS}=D_{VSS}=0V$ ,  $V_{DD}=D_{VDD} = 5V$

Parameter	Symbol	Test Conditions	Typ	Max	Units
Set-up time after a sleep period	$T_{set}$		25	35	ms
Time from full power to modulation state	$T_{mdon}$	antenna circuit specifications: $Q=15, F_{RES}=125Khz$		50	$\mu s$
AM demodulation: Delay time from input to output	$T_{pd}$	modulation index: 100% Modulating signal 2Khz square wave 10mVpp	40	100	$\mu s$
Recovery time of reception after antenna modulation	$T_{rec}$	Note 1	400	500	$\mu s$

Note 1: RF period is time of one period transmitted on ANT outputs (at 125 kHz 8 $\mu s$ ).  $T_{rec}$  after antenna modulation receiver chain is ready to demodulate. The condition is of course that the amplitude on antenna has already reached its steady state by that time (this depends on Q of antenna). See also Application Notes.

## Block Diagram

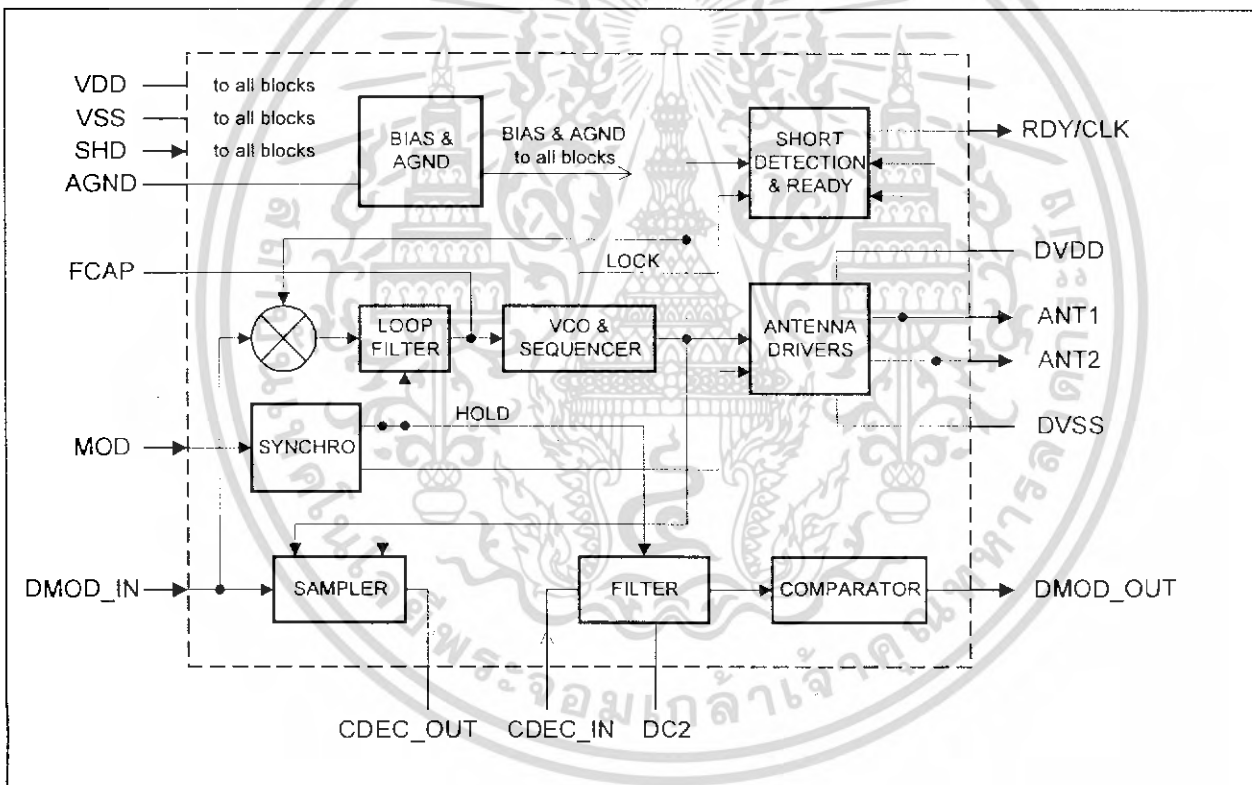


Fig. 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



# EM4095

## Functional Description

### General

The EM4095 is intended to be used with an attached antenna circuit and a microcontroller. Few external components are needed to achieve DC and RF filtering, current sensing and power supply decoupling.

A stabilised power supply has to be provided. Please refer to EM4095 Application Notes for advice.

Device operation is controlled by logic inputs SHD and MOD. When SHD is high EM4095 is in sleep mode, current consumption is minimised. At power up the input SHD has to be high to enable correct initialisation. When SHD is low the circuit is enabled to emit RF field, it starts to demodulate any amplitude modulation (AM) signal seen on the antenna. This digital signal coming from the AM demodulation block is provided through DEMOD\_OUT pin to the microcontroller for decoding and processing.

High level on MOD pin forces in tri-state the main antenna drivers synchronously with the RF carrier. While MOD is high the VCO and AM demodulation chain are kept in state before the MOD went high. This ensures fast recovery after MOD is released. The switching ON of VCO and AM demodulation is delayed by 41 RF clocks after falling edge on MOD. In this way the VCO and AM demodulation operating points are not perturbed by start-up of antenna resonant circuit.

### Analog Blocks

The circuit performs the two functions of an RFID basestation, namely: transmission and reception. Transmission involves antenna driving and AM modulation of the RF field. The antenna drivers deliver a current into the external antenna to generate the magnetic field.

Reception involves the AM demodulation of the antenna signal modulation induced by the transponder. This is achieved by sensing the absorption modulation applied by the tag (transponder).

### Transmission

Referring to the block diagram, transmission is achieved by a Phase Locked Loop (PLL) and the antenna drivers.

### Drivers

The antenna drivers supply the reader basestation antenna with the appropriate energy. They deliver current at the resonant frequency which is typically 125 kHz. Current delivered by drivers depends on Q of external resonant circuit.

It is strongly recommended that design of antenna circuit is done in a way that maximum peak current of 250 mA is never exceeded (see Typical Operating Configuration for antenna current calculation). Another limiting factor for antenna current is Thermal Convection of package. Maximum peak current should be designed in a way that internal junction temperature does not exceed maximum junction temperature at maximum application ambient temperature. 100% modulation (field stop) is done by switching OFF the drivers. The ANT drivers are protected against antenna DC short circuit to the power supplies. When a short circuit has been detected the RDY/CLK pin is pulled low while the main driver is forced in tri-state. The circuit can be restarted by activating the SHD pin.

### Phase locked loop

The PLL is composed of the loop filter, the Voltage Controlled Oscillator (VCO), and the phase comparator blocks. By using an external capacitive divider, pin DEMOD\_IN gets information about the actual high voltage signal on antenna.

Phase of this signal is compared with the signal driving antenna drivers. Therefore the PLL is able to lock the carrier frequency to the resonant frequency of the antenna. Depending on the antenna type the resonant frequency of the system can be anywhere in the range from 100 kHz to 150 kHz. Wherever the resonant frequency is in this range it will be maintained by the Phase Lock Loop.

### Reception

The demodulation input signal for the reception block is the voltage sensed on the antenna. DEMOD\_IN pin is also used as input to Reception chain. The signal level on the DEMOD\_IN input must be lower than VDD-0.5V and higher than VSS+0.5V. The input level is adjusted by the use of an external capacitive divider. Additional capacitance of divider must be compensated by accordingly smaller resonant capacitor. The AM demodulation scheme is based on the "AM Synchronous Demodulation" technique.

The reception chain is composed of sample and hold, DC offset cancellation, bandpass filter and comparator. DC voltage of signal on DEMOD\_IN is set to AGND by internal resistor. The AM signal is sampled, the sampling is synchronised by a clock from VCO. Any DC component is removed from this signal by the CDEC capacitor. Further filtering to remove the remaining carrier signal, high and low frequency noise is made by second order highpass filter and CDC2. The amplified and filtered receive signal is fed to asynchronous comparator. Comparator output is buffered on output pin DEMOD\_OUT.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



# EM4095

## Signal RDY/CLK

This signal provides the external microprocessor with clock signal which is synchronous with the signal on ANT1 and with information about EM4095 internal state. Clock signal synchronous with ANT1 indicates that PLL is in lock and that Reception chain operation point is set. When SHD is high RDY/CLK pin is forced low. After high to low transition on SHD the PLL starts-up, and the reception chain is switched on. After time  $T_{SET}$  the PLL is locked and reception chain operation point has been established. At this moment the same signal which is being transmitted to ANT1 is also put to RDY/CLK pin indicating to microprocessor that it can start observing signal on DEMOD\_OUT and giving at the same time reference clock signal. Clock on RDY/CLK pin is continuous, it is also present during time the ANT drivers are OFF due to high level on MOD pin. During the time  $T_{SET}$  from high to low transition on SHD pin RDY/CLK pin is pulled down by 100 k $\Omega$  pull down resistor. The reason for this is in additional functionality of RDY/CLK pin in case of AM modulation with index which is lower than 100%. In that case it is used as auxiliary driver which maintains lower amplitude on coil during modulation. (see also Typical Operating Configuration)

Remark: Please refer to AN4095 for external components calculation and limits.

## Typical Operating Configuration

### Read Only Mode

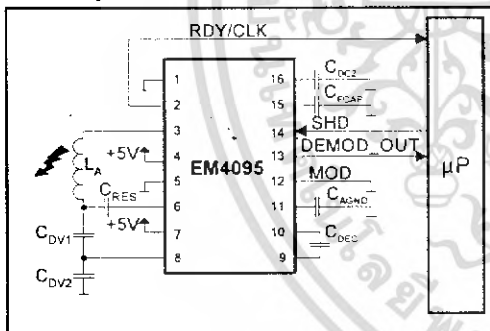


Fig. 6

### Read/Write mode (Low Q factor antenna)

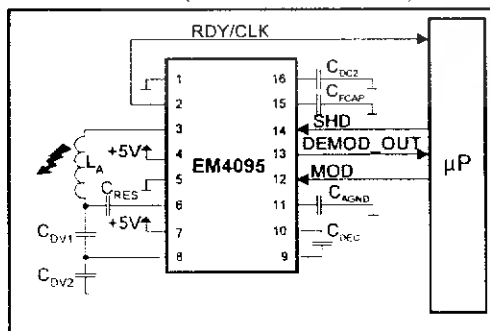


Fig. 7

### Read/Write mode (High Q factor antenna)

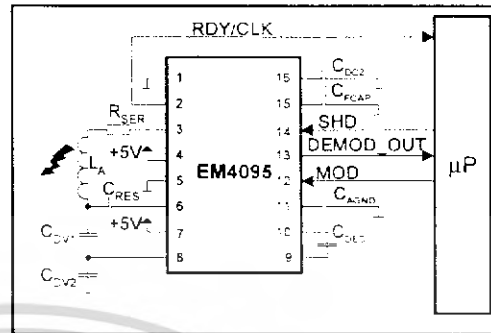


Fig. 8

### Read/Write mode (AM modulation)

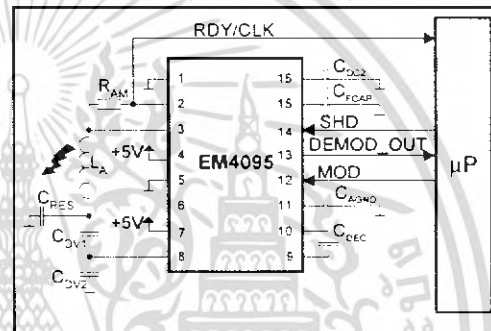


Fig. 9

Figure 6 presents EM4095 used in Read Only mode. Pin MOD is not used. It is recommended to connect it to VSS.

Figure 7 presents typical R/W configuration for OOK communication protocol reader to transponder (eg. EM4150). It is recommended to be used with low Q factor antennas (up to 15).

When the antenna quality is high using configuration of figure 6 or 7 the voltage on antenna can arrive in the range of few hundred volts and antenna peak current may exceed its maximum value. In such a case the capacitive divider ratio has to be high thus limiting the sensitivity. For such case it is better to reduce antenna circuit quality by adding serial resistor. In this way the antenna current is lower and thus power dissipation of IC is reduced with practically the same performance (Fig. 8).

In the case AM modulation communication protocol reader to transponder (eg. EM4069) is needed a single ended configuration has to be used (figure 9). When pin MOD is pulled high driver on ANT1 is put in three state, driver RDY/CLK continues driving thus maintaining lower antenna current. Modulation index is adjusted by resistor  $R_{AM}$ . As mentioned above RDY/CLK signal becomes active only after the demodulation chain operating point is set.

Before it is pulled down by high impedance pull down resistor (100 k $\Omega$ ) in order not to load ANT1 output. In the case of AM modulation configuration the total antenna current change at the moment RDY/CLK pin becomes active, so external microprocessor has to wait another  $T_{SET}$  before it can start observing DEMOD\_OUT.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการค้า



# EM4095

Read Only mode with external peak detector

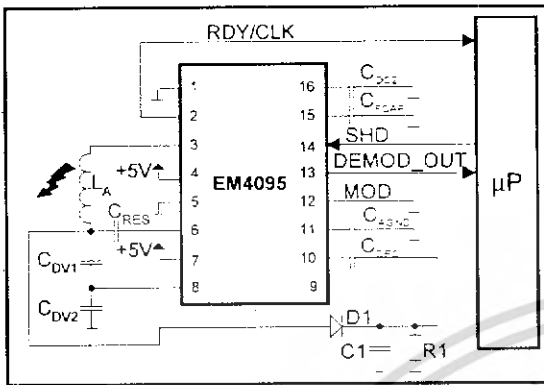


Fig. 10

Read/Write mode with external peak detector

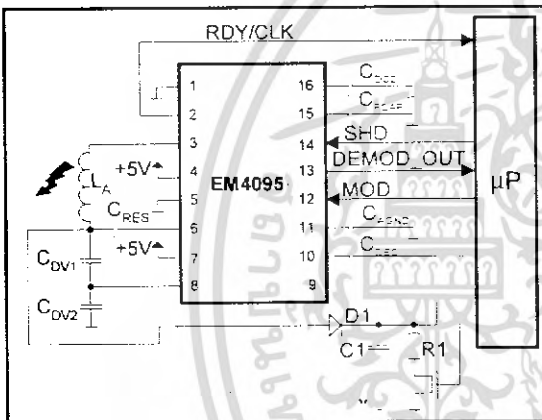


Fig. 11

As mentioned above for high Q antennas the voltage on antenna is high and read sensitivity is limited by demodulator sensitivity due to capacitive divider. Read sensitivity (and thus reading range) can be increased by using external envelope detector circuit. Input is taken on antenna high voltage side output is directly fed to CDEC\_IN pin. However, the capacitor divider is still needed for PLL locking. Such configuration is shown in figure 5, the envelope detector is formed by three components: D1, R1 and C1.

The configuration presented in figure 9 may also be used for read write applications but it has a drawback in the case fast recovery of reading is needed after communication reader to transponder is finished. The reason is in fact that DC voltage after diode D1 is lost during modulation and it takes very long time before it is established again.

Figure 10 presents a solution to that problem. A high voltage NMOS transistor blocks the discharge path during modulation, so operating point is preserved. The signal controlling NMOS gate has to be put low synchronously with signal MOD, but it can be put high only after the amplitude on antenna has recovered after modulation.

### PCB Layout

Refer to "EM4095 Application Note" (App. Note 404)



# EM4095

## Pin Description

### SOIC 16 package

Pin	Name	Description	Type
1	V <sub>SS</sub>	Negative power supply (substrate)	GND
2	RDY/CLK	Ready flag and clock output, driver for AM modulation	O
3	ANT1	Antenna driver	O
4	D <sub>VDD</sub>	Positive power supply for antenna drivers	PWR
5	D <sub>VSS</sub>	Negative power supply for antenna drivers	GND
6	ANT2	Antenna driver	O
7	V <sub>DD</sub>	Positive power supply	PWR
8	DEM0D_IN	Antenna sensing voltage	ANA
9	CDEC_OUT	DC blocking capacitor connection « out »	ANA
10	CDEC_IN	DC blocking capacitor connection « in »	ANA
11	A <sub>GND</sub>	Analog ground	ANA
12	MOD	A High level voltage modulates the antenna	IPD
13	DEM0D_OUT	Digital signal representing the AM seen on the antenna	O
14	SHD	A High level voltage forces the circuit into sleep mode	IPU
15	FCAP	PLL Loop filter capacitor	ANA
16	DC2	DC decoupling capacitor	ANA

GND: reference ground

IPD: input with internal pull down

PWR: power supply

IPU: input with internal pull up

ANA:

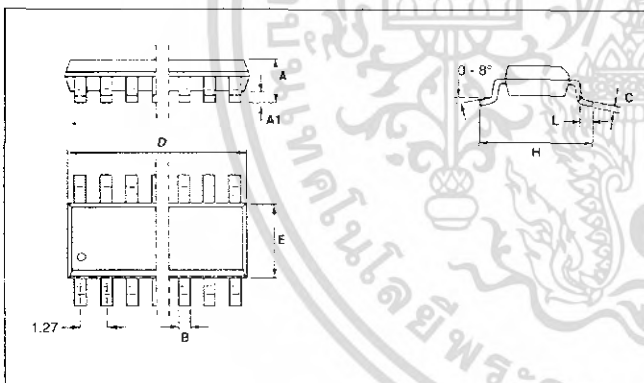
analog signal output

O:

output

## Package and Ordering Information

### Dimensions of SOIC 16 Package (table in millimeters)



Symbol	Common Dimensions (mm)		
	Min	Nom	Max
A	1.55	1.63	1.73
A1	0.127	0.15	0.25
B	0.35	0.41	0.49
C	0.19	0.20	0.25
D	9.80	9.93	9.98
E	3.81	3.94	3.99
H	5.84	5.99	6.20
L	0.41	0.64	0.89

Fig. 12

## Ordering Information

Please make sure to give the complete part number when ordering.

The EM4095 is available in the following package:

Part Number	Package	Delivery Form
EM4095HMSO16A	SOIC 16 package	stick

## Product Support

Check our Web Site under Products/RF Identification section.

Questions can be sent to [cid@emmicroelectronic.com](mailto:cid@emmicroelectronic.com)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า