สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบเพิ่มความปลอดภัยในบริการที่จอดรถยนต์ผ่าน RFID

A SECURITY IMPROVEMENT FOR PARKING SERVICE SYSTEM BY RFID

โดย

นายณกรณ์ นากอร่าม 48010234

นายณัฐพล มณีวงศ์ 48010272

นายดนุสรณ์ กุลนรากร 48010287

วท.
ณ111ร
2551

เลขหมู่.....
เลขทะเบียน 104362
วัน,เดือน,ปี - 2 พ.ย. 2552

b 12102146
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

ภาควิชา
วิศวกรรมโทรคมนาคม
ฝ่ายการตรวจรูปเล่มแล้ว
(ลงชื่อ).....ผู้ตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2551

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเพิ่มความปลอดภัยในบริการที่จอดรถยนต์ผ่าน RFID

A SECURITY IMPROVEMENT FOR PARKING SERVICE SYSTEM BY RFID

ผู้จัดทำ

1. นายณกรณ์ นากอร่าม 48010234
2. นายณัฐพล มณีวงศ์ 48010272
3. นายดนุสรณ์ กุลนรากร 48010287

อาจารย์ที่ปรึกษา

(รศ. เกรียงไกร วงศ์โรจนภรณ์)

อาจารย์ที่ปรึกษา

(รศ.ดร. ไกรสิน ส่งวัฒนา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเพิ่มความปลอดภัยในบริการที่จอดรถยนต์ผ่าน RFID

A SECURITY IMPROVEMENT FOR PARKING SERVICE SYSTEM BY RFID

ผู้จัดทำ 1 นายณกรณ์ นากอร่าม 48010234
2 นายณัฐพล มณีวงศ์ 48010272
3 นายดนุสรณ์ กุลนรากร 48010287

อาจารย์ที่ปรึกษา รศ. เกรียงไกร วงศ์โรจนภรณ์
รศ.ดร. ไกรสิน ส่งวัฒนา

บทคัดย่อ

ผู้ใช้บริการนำรถยนต์ เข้ามาบริเวณทางเข้าที่จอดรถพร้อม RFID Tag ของผู้ใช้บริการ ระบบจะทำการบันทึกคู่ RFID Tag กับ หมายเลขทะเบียนรถยนต์ เมื่อผู้ใช้บริการต้องการนำรถยนต์ออกจากที่จอดรถ ระบบจะทำการตรวจเช็ค RFID Tag กับ หมายเลขทะเบียนของรถยนต์ ว่าตรงคู่กันตามที่บันทึกหรือไม่ หากตรงกันจะอนุญาตให้ผ่านออกได้ พร้อมทั้งทำการหักค่าบริการจอดรถผ่าน RFID Tag โดยอัตโนมัติ

Abstract

The thesis presents a security improvement for parking service system by RFID. The car is provided with car license and register together with customer's RFID Tag. A system pairs of car license and RFID Tag on the database. When customer wants to exit, the system checks RFID Tag and car license pair. If they match the system permits to exit and automatically deduct the parking fee.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์ในการทำโครงงาน	1
1.3 ขอบเขตของโครงงาน	2
1.4 ขั้นตอนในการทำโครงงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 RFID	3
2.2 Carrier ในระบบ RFID	11
2.3 รูปแบบการทำงาน (หลักการ) ของ Reader	22
2.4 ภาษา SQL	30
2.5 หลักการสร้าง SQL เพื่อกำหนดออบเจ็กต์ฐานข้อมูล	39
2.6 ไมโครคอนโทรเลอร์	48
2.7 สเต็ปมอเตอร์ (STEPPING MOTOR)	49
บทที่ 3 การคำนวณและการสร้าง	53
3.1 การออกแบบการควบคุม TAG	53
3.2 การออกแบบการควบคุมตัวอ่าน	54
3.3 โปรแกรมข้อมูลผู้ถือบัตร	55
3.4 แนวคิดการออกแบบจอภาพในโปรแกรม	57
3.5 แนวทางการเขียนโปรแกรม	58
3.6 หลักการออกแบบฮาร์ดแวร์	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลองและผลการทดลอง 65

4.1 การทดลองอุปกรณ์ Tag และตัวอ่าน 65

4.2 ทดลองโปรแกรมข้อมูลผู้ถือบัตร 68

4.3 ทดลองสเต็ปมอเตอร์ 69

4.4 ผลการทดลอง 73

บทที่ 5 วิจารณ์และสรุป 74

5.1 สรุปผลการทดลอง 74

5.1 ปัญหาและอุปสรรค 74

5.2 ข้อเสนอแนะและแนวทางแก้ปัญหา 75

ภาคผนวก ก

กิตติกรรมประกาศ 76

บรรณานุกรม 77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่ **หน้า**

รูปที่ 2.1 ภาพการทำงานของระบบ RFID 4
รูปที่ 2.2 ระบบการอ่าน / เขียนข้อมูลอย่างง่ายของ RFID 5
รูปที่ 2.3 แสดงตัวอย่างการใช้งานแท็กและตัวอ่านข้อมูล (Reader) 6
รูปที่ 2.4 แสดง RFID tag ในรูปแบบต่างๆ 7
รูปที่ 2.5 แสดงบล็อกไดอะแกรมของ Passive tag 8
รูปที่ 2.6 แสดงรูปตัวอย่าง Active tag ที่มีแบตเตอร์รี่ lithium, 2 ก้อนอยู่ภายนอก 8
รูปที่ 2.7 แสดงโครงสร้างภายในตัวอ่าน 9
รูปที่ 2.8 แสดงส่วนประกอบของเครื่องอ่าน RFID 10
รูปที่ 2.10 แสดงความถี่ที่ระบบ RFID ถูกใช้งาน 11
รูปที่ 2.11 แสดงแผนผังการทำงานของระบบ RFID 12
รูปที่ 2.12 แสดงการการส่งข้อมูลระหว่างตัวอ่านและ Tag รูปแบบต่างๆ 15
รูปที่ 2.13 การสื่อสารระหว่าง Reader กับ Tag 15
รูปที่ 2.14 แสดงการสร้าง Power supply ให้กับ tag โดย reader 16
รูปที่ 2.15 การ Generation of Load Modulation 17
รูปที่ 2.16 วงจร Generation of Load Modulation with Sub Carrier 17
รูปที่ 2.17 การเข้ารหัสข้อมูลในแบบต่างๆ 18
รูปที่ 2.18 การผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงของยอดคลื่นพาหะ 19
รูปที่ 2.19 การผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงความถี่ 20
รูปที่ 2.20 การผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงเฟส 20
รูปที่ 2.21 การ Modulation แบบ ASK Sub carrier 21
รูปที่ 2.22 ระบบ RFID 22
รูปที่ 2.23 Master – Slave ระหว่าง Application Software, เครื่องอ่านและ Tag 23
รูปที่ 2.24 Block Diagram ของเครื่องอ่านที่ประกอบด้วยส่วนควบคุมและ HF Interface 24
รูปที่ 2.25 Block Diagram ของ HF Interface สำหรับ Inductive Couple 25
รูปที่ 2.26 Block Diagram ของ Control Unit ของ Reader และสื่อสารกับ Application Software 26
รูปที่ 2.27 สัญญาณ Coding และ Decoding ถูกกระทำโดย Control Unit ของ Reader 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.28 Block Diagram ของ RFID Data Carrier ของ Memory Function 28
รูปที่ 2.29 Block Diagram ของ HF Interface ของ Inductive Coupled ของ Tag with Load Modulation 28
รูปที่ 2.30 การสร้าง Load Modulation กับ Sub carrier Modulated 30
รูปที่ 2.31 แสดงขาต่างๆของ PIC16F877 48
รูปที่ 2.32 ผังไดอะแกรมของ PIC16F877 49
รูป 2.33 สเต็ปมอเตอร์แบบมีสาย 5 เส้น 50
รูป 2.34 มอเตอร์แบบมีสาย 6 เส้น 50
รูป 2.35 สเต็ปมอเตอร์หลายแบบไบโพลาร์ 51
รูป2.36 แสดงภาพถ่ายโครงสร้างสเต็ปมอเตอร์ 51
รูป 2.36 แสดง (ก) โครงสร้าง (ข) วงจรเทียบเท่า (equivalent circuit) ของมอเตอร์ ชนิด 4 ขดแสดงมุมของโรเตอร์ เทียบกับกระแสไฟฟ้าที่จ่ายแก่เฟสต่าง ๆ8 ตำแหน่ง 52
รูปที่ 3.1 แสดงรูป TAG ที่นำมาใช้งาน 53
รูปที่ 3.2 บิตที่ใช้ในการระบุ User 53
รูปที่ 3.3 แสดงรูป RFID Reader ที่นำมาใช้งาน 54
รูปที่ 3.4 ภาพการเชื่อมต่ออุปกรณ์ 55
รูปที่ 3.5 ตารางฐานข้อมูลรายละเอียดของผู้ใช้งาน 56
รูปที่ 3.6 หน้าต่างหลักของโปรแกรม 57
รูปที่ 3.7 ขั้นตอนเกี่ยวกับการเข้า – ออกรถยนต์ในห้างสรรพสินค้า 58
รูปที่ 3.8 โฟล์วชาร์ทโปรแกรมส่วนผู้ใช้บริการขาเข้า 59
รูปที่ 3.9 โฟล์วชาร์ทโปรแกรมส่วนผู้ใช้บริการขาออก 60
รูปที่ 3.10 โฟล์วชาร์ทการทำงานในส่วนควบคุมฮาร์ดแวร์ 62
รูปที่ 3.11 แสดงการควบคุมของระบบทั้งหมด 63
รูปที่ 3.12 บอร์ด PIC Microcontroller 63
รูปที่ 3.13 แสดงการทำงานของระบบทั้งหมด 64
รูปที่ 4.1 เชื่อมต่ออุปกรณ์ RFID เข้ากับคอมพิวเตอร์ 65
รูปที่ 4.2 โปรแกรม ReadWriteDataBlockSample 65
รูปที่ 4.3 แสดงสัญญาณที่ออกมาจาก RFID Reader ขณะที่ยังไม่ได้แตะ TAG 66
รูปที่ 4.4 แสดงสัญญาณที่ออกมาจาก RFID Reader ขณะที่แตะ TAG 66
รูปที่ 4.5 เมื่ออ่าน TAG ที่ถูกต้อง 67
รูปที่ 4.6 ข้อมูลภายใน TAG เมื่อ TAG ถูกต้อง 67
รูปที่ 4.7 หน้าต่างหลักโปรแกรมที่จอดรถ 68
รูปที่ 4.8 หน้าต่างแสดงผลเมื่อแตะ TAG ที่ตัวอ่าน 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 แสดงแขนกั้นเมื่อปิดอยู่ 69
รูปที่ 4.10 แสดงข้อความพร้อมทำงาน 69
รูปที่ 4.11 เมื่อนำ TAG มาแตะ RFID Reader 70
รูปที่ 4.12 แสดงข้อความหลังจากแตะ RFID ที่ RFID Reader 70
รูปที่ 4.13 แขนกั้นจะเปิดออกเพื่อให้รถยนต์เข้าไปได้ 71
รูปที่ 4.14 เมื่อนำ RFID มาแตะที่ RFID Reader เพื่อนำรถออก 71
รูปที่ 4.15 แขนกั้นจะเปิดออกเพื่อให้รถยนต์ออกไปได้ 72
รูปที่ 4.16 แสดงข้อความยอดเงินคงเหลือเมื่อรถกำลังออกไป 72
รูปที่ 4.17 แสดงข้อความเมื่อรถออกไปแล้ว 72
รูปที่ 4.18 แสดงข้อความเมื่อจำนวนเงินในบัตร RFID มีไม่เพียงพอ 73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 2.1 กระบวนการส่งข้อมูลของ RFID 21
ตารางที่ 2.2 ตัวอย่างการดำเนินการอ่านข้อมูลด้วย Application Software เครื่องอ่านและ Tag 24
ตารางที่ 2.3 ความถี่ Clock ที่ต้องการใน HF Interface ถูกสร้างโดย Binary ของสัญญาณ Carrier 13.56 MHz 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

เทคโนโลยี RFID (RADIO FREQUENCY IDENTIFICATION) เป็นระบบชี้เฉพาะอัตโนมัติด้วยคลื่นความถี่วิทยุ ที่ใช้สัญญาณวิทยุพลังต่ำเพื่อแลกเปลี่ยนข้อมูลกันแบบไร้สาย (Wireless) ซึ่งในปัจจุบันนี้เราอาจจะกล่าวได้ว่าเทคโนโลยี RFID เป็นระบบที่ถูกนำมาใช้งานในด้านต่างๆมากมาย ไม่ว่าจะเป็นหน่วยงาน หรือองค์กรทั้งภาครัฐ ภาคเอกชน โรงงานอุตสาหกรรม ร้านค้า ห้างสรรพสินค้า ซึ่งเราจะพบเห็นการใช้งานของระบบ RFID ได้ อาทิเช่น ระบบรักษาความปลอดภัย ระบบคลังสินค้า ระบบเครือข่ายเทคโนโลยี Smart Card/RFID ระบบการตรวจสอบและติดตามตู้สินค้าผนึกติดด้วย RFID ระบบควบคุมยานพาหนะเข้าออกด้วย RFID บาร์โค้ดและการปศุสัตว์ เป็นต้น

ด้วยความสามารถและการนำมาประยุกต์ใช้งานอย่างแพร่หลายของเทคโนโลยี RFID นี้เอง ทางผู้จัดทำโครงงานจึงมีแนวความคิดที่จะนำเทคโนโลยีมาใช้ในการแก้ไขปัญหาที่เกิดขึ้นกับระบบที่จอดรถที่มีใช้กันอยู่ในปัจจุบัน จึงได้จัดทำโครงงานระบบที่จอดรถโดยใช้เทคโนโลยี RFID ขึ้นมา โดยเป็นการออกแบบระบบที่จอดรถโดยใช้แท็ก (Tag) เป็นบัตรผ่านในการเข้าออกที่จอดรถ ซึ่งจะทำให้ผู้ใช้บริการที่จอดรถมีความสะดวกรวดเร็ว อีกทั้งยังทำให้ ผู้ให้บริการที่จอดรถสามารถควบคุมการเข้าออกอย่างมีระบบ ในการเก็บบันทึกข้อมูลการเข้าจอดรถ ทำให้สามารถตรวจสอบการเข้าใช้บริการย้อนหลังได้

ในส่วนของการติดต่อสื่อสารระหว่างแท็กกับเครื่องอ่าน (RFID Reader) ที่ติดตั้งอยู่บริเวณทางเข้าและทางออกที่จอดรถนั้น จะใช้คลื่นวิทยุในการติดต่อสื่อสารกัน ซึ่งทำให้ไม่ต้องมีการสัมผัสกันระหว่างเครื่องอ่านกับแท็ก เมื่อผู้ใช้บริการเขามาบริเวณทางเข้าที่จอดรถ เครื่องอ่านก็จะอ่านข้อมูลจากแท็ก และจะบันทึกหมายเลขทะเบียนรถยนต์ของผู้ใช้ด้วย แล้วนำข้อมูลที่ได้ไปประมวลผลแล้วบันทึกเวลาและข้อมูลของตัวรถและของข้อมูลผู้ใช้บริการลงในฐานข้อมูล และเมื่อผู้ใช้บริการ ผ่านเข้ามาในบริเวณทางออกที่จอดรถ เครื่องอ่านก็จะทำการอ่านข้อมูลจากแท็กและนำข้อมูลเวลาเข้าออกที่จอดรถมาทำการคำนวณค่าใช้บริการ และคิดค่าบริการจากแท็ก

1.2 วัตถุประสงค์ในการทำโครงงาน

1. เพื่อศึกษาหลักการทำงานของเทคโนโลยี RFID
2. เพื่อศึกษาหลักการทำงานของเทคโนโลยี PIC Microcontroller
3. สามารถประยุกต์การทำงานของ RFID มาใช้ในการให้บริการที่จอดรถยนต์ได้
4. เพื่อศึกษาการใช้โปรแกรม Microsoft SQL Server 2005
5. เพื่อศึกษาการใช้โปรแกรม Microsoft Visual C# 2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

1. สามารถรับและแสดงข้อมูลออกมาจากเครื่องอ่านได้
2. สามารถส่งข้อมูลระหว่างเครื่องอ่าน (RFID Reader) ไปยังเครื่องคอมพิวเตอร์ได้
3. สามารถสั่งเปิด-ปิดแขนกั้นโดยใช้ PIC Microcontroller
4. ออกแบบที่ระบบจอดรถโดยใช้เทคโนโลยี RFID

1.4 ขั้นตอนในการทำโครงงาน

1. ศึกษาข้อมูล RFID
2. ศึกษาข้อมูล PIC Microcontroller
3. เขียนโปรแกรมควบคุมการทำงาน
4. ใช้งานโปรแกรมกับอุปกรณ์
5. ทำอุปกรณ์ให้สมบูรณ์
6. ทำปริญญานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้ ความเข้าใจเกี่ยวกับเทคโนโลยี RFID
2. เพิ่มประสิทธิภาพและความสะดวกในการในบริการที่จอดรถยนต์
3. สามารถประยุกต์ใช้ในการทำธุรกิจที่ใช้ RFID ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 RFID

คือระบบชี้เฉพาะอัตโนมัติ (Automatic Identification) แบบไร้สาย (Wireless) เช่นเดียวกับระบบ Barcode แต่ใช้ความถี่วิทยุในการติดต่อสื่อสาร ระบบนี้จะประกอบด้วยอุปกรณ์สองส่วน คือ ส่วนเครื่องอ่าน (Reader) และส่วนป้ายชื่อ (Tag) โดยการทำงานนั้นเครื่องอ่านจะทำหน้าที่จ่ายกำลังงานในรูปคลื่นความถี่วิทยุให้กับ Tag ยังผลให้วงจรอิเล็กทรอนิกส์ภายในสามารถส่งข้อมูลจำเพาะที่แสดงถึง "Identity" กลับมาประมวลผลที่ตัวอ่านได้ ข้อดีของระบบ RFID คือเป็นระบบไร้สายจึงไม่จำเป็นต้องนำ Tag ไปสัมผัสกับเครื่องอ่านซึ่งช่วยเพิ่มความสะดวกและเป็นอัตโนมัติให้กับระบบชี้เฉพาะ โดยระบบ RFID สามารถนำไปประยุกต์ใช้งานได้อย่างกว้างขวาง เช่น การใช้บอกรหัสสินค้า การใช้บอกรหัสประจำตัวพนักงาน หรือ การใช้งานควบคุมการเข้าออก เป็นต้น

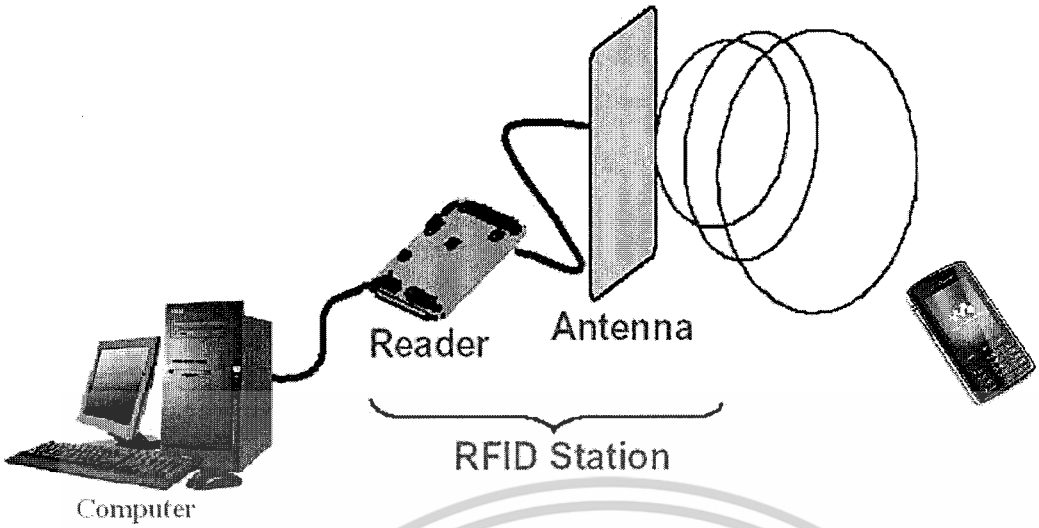
2.1.1 ความหมายของ RFID (Radio frequency Identification System)

RFID จะมีลักษณะใกล้เคียงกับสมาร์ตการ์ดที่สุดคือข้อมูลจะถูกเก็บไว้ในการ์ดหรือ Tags เหมือนกัน แต่ข้อแตกต่างที่เห็นได้ชัดคือการอ่าน/เขียนข้อมูลสามารถทำได้โดยไม่ต้อง สัมผัสเหมือนกับสมาร์ตการ์ด ซึ่งจะอาศัยคลื่นแม่เหล็กไฟฟ้าในช่วงความถี่วิทยุ (Radio Frequency) ในการติดต่อสื่อสารจึงเป็นสาเหตุให้ RFID ได้เปรียบระบบ Auto ID ทุกชนิดที่กล่าวมาแล้ว และมีแนวโน้มจะถูกใช้มากขึ้นเรื่อยๆ Radio frequency Identification (RFID) RFID ย่อมาจาก Radio Frequency Identification เป็นระบบระบุเอกลักษณ์ของวัตถุ ด้วยคลื่นความถี่วิทยุที่ได้ถูกพัฒนามาตั้งแต่ปี ค.ศ. 1980 เพื่อวัตถุประสงค์หลักเพื่อนำไปใช้งานแทนระบบบาร์โค้ด (Barcode) โดยจุดเด่นของ RFID ก็อยู่ตรงการอ่านข้อมูลจาก Tag ได้หลาย ๆ Tag แบบไร้สัมผัส และสามารถอ่านค่าได้แม้ในสภาพที่ทัศนะวิสัย ทนต่อความเปียกชื้นแรงสั่นสะเทือน การกระทบกระแทกและสามารถจะอ่านข้อมูลได้ด้วยความเร็วสูง โดยข้อมูลจะถูกเก็บไว้ในไมโครชิพที่อยู่ใน Tag

Radio Frequency Identification (RFID) ประกอบด้วย

1. Label หรือ RFID Tag คือแผงวงจรวิทยุขนาดเล็กบรรจุข้อมูลความจำ (Memory chip) บนแผ่นกระดาษขนาด 2 ตารางนิ้ว
2. Reader เครื่องอ่านสัญญาณจะสร้างความถี่สัญญาณวิทยุขึ้นแล้วส่งไปยังสายอากาศ
3. Antenna จะเชื่อมต่อกับ Reader เพื่อส่งคลื่นวิทยุที่กำเนิดจากตัว Reader ไปยัง Tag เพื่อกระตุ้นให้ Tag ส่งข้อมูลกลับมาให้ตัว Reader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ภาพการทำงานของระบบ RFID

2.1.2 ส่วนประกอบของระบบ RFID

โดยทั่วไประบบ RFID จะประกอบไปด้วย 2 ส่วนสำคัญคือ

ฉลากอิเล็กทรอนิกส์ (Transponder) หรือ Data Carrier หรือเรียกง่ายๆ ว่า Tags จะเป็นส่วนของการเก็บข้อมูลและถูกติดตั้งอยู่กับวัตถุที่เราต้องการบ่งชี้ (Identify)

Tags จะประกอบไปด้วยเสาอากาศทำหน้าที่คล้องสัญญาณที่ส่งมาจากเครื่องอ่านและส่วนของไมโครชิพ ในกรณีที่ Tags ไม่มีแบตเตอรี่ในตัวอยู่นอกพื้นที่ที่มีสัญญาณจะไม่มีการทำงานเกิดขึ้น Tags จะทำงานก็ต่อเมื่อ Tags เข้ามาในพื้นที่ที่มีสัญญาณซึ่ง Tags จะได้รับพลังงานจากการคล้องของสัญญาณสนามแม่เหล็กไฟฟ้าและสร้างแรงดันไฟฟ้าขึ้นจำนวนหนึ่ง ปริมาณเพียงพอที่จะใช้ในการทำงานของ Tags

เครื่องอ่าน (Reader) จะประกอบไปด้วยภาครับ/ส่งสัญญาณวิทยุ ส่วนควบคุมและเสาอากาศ (Antenna) ทำหน้าที่คล้องสัญญาณกับ Tags และส่วนของการติดต่อสื่อสารกับอุปกรณ์ควบคุมภายนอก (RS232C, RS485 RS422 หรือ USB) ขึ้นอยู่กับลักษณะการสื่อสารที่เราต้องการ

Tag หรือ Transponder

แท็ก (Tag) นั้นเรียกอีกชื่อหนึ่งว่าทรานสปอนเดอร์ (Transponder) มาจากคำว่าทรานสมิตเตอร์ (Transmitter) ผสมกับคำว่าเรสปอนเดอร์ (Responder) ถ้าจะแปลให้ตรงตามศัพท์ แท็กก็จะทำหน้าที่ส่งสัญญาณหรือข้อมูลที่บันทึกอยู่ในแท็กตอบสนองไปที่ตัวอ่านข้อมูล การสื่อสารระหว่างแท็กและตัวอ่านข้อมูลจะเป็นการสื่อสารกันโดยอาศัยช่องความถี่วิทยุผ่านอากาศ โครงสร้างภายในแท็กจะประกอบไปด้วย 2 ส่วนใหญ่ๆ ได้แก่ ส่วนของไอซีซึ่งเป็นชิพสารกึ่งตัวนำ (Semiconductor Chip) และส่วนของขดลวดซึ่งทำหน้าที่เป็นเสาอากาศสำหรับรับส่งข้อมูลโดยทั้งสองส่วนนี้จะเชื่อมต่ออยู่ด้วยกัน ไอซีของแท็กที่มีการผลิตออกมาจะมีทั้งขนาดและรูปร่างเป็นได้ตั้งแต่แท่งหรือแผ่นขนาดเล็กจนแทบไม่สามารถมองเห็น หรือไปจนถึงขนาดใหญ่จะสะดุดตา ซึ่งต่างก็มี

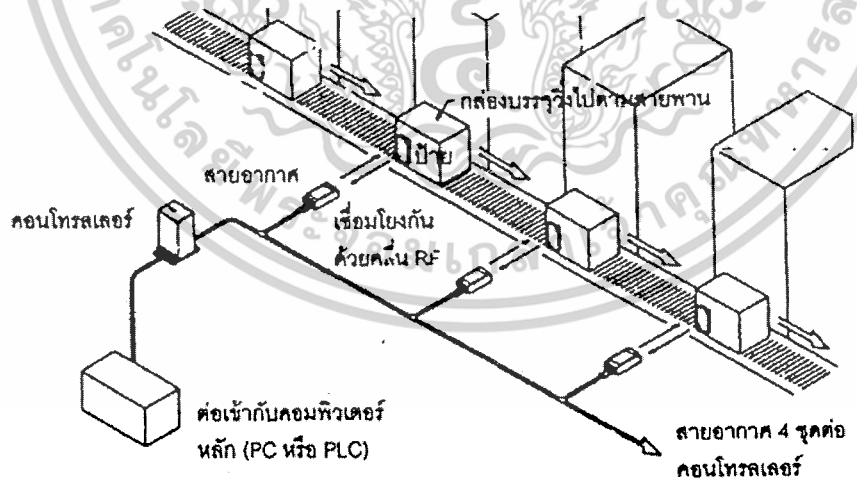
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเหมาะสมกับชนิดงานที่แตกต่างกัน โดยทั่วไปโครงสร้างภายในส่วนที่เป็นไอซีของแท็กนั้นก็จะประกอบด้วย 2 ส่วนหลัก ๆ ได้แก่

1. ส่วนของการควบคุมภาครับส่งสัญญาณวิทยุ สำหรับโครงสร้างของส่วนนี้ประกอบด้วยภาคดีมอดูเลต และภาคมอดูเลต (สำหรับรับส่งข้อมูลระหว่างแท็กกับตัวเครื่องอ่าน) และวงจรกำเนิดไฟฟ้าขนาดเล็ก

2. ส่วนของการควบคุมภาคดิจิตอล ซึ่งรับหน้าที่จัดการเกี่ยวกับกระบวนการทางดิจิตอลทั้งหมด โครงสร้างหลักๆ ของส่วนการทำงานนี้ประกอบด้วย ส่วนบันทึกข้อมูล (ประกอบด้วยหน่วยความจำแรม (RAM), รอม (ROM), อีอีพรอม (EEPROM) ส่วนของการเข้ารหัส (Crypts Unit) ส่วนตอบรับสัญญาณร้องขอ (Answer to Request) ส่วนควบคุมและประมวลผลทางคณิตศาสตร์ (Control & Arthritic Unit) อย่างไรก็ตามโครงสร้างภายในของแท็กที่ต่างผู้ผลิตหรือต่างรุ่นกัน บางครั้งก็อาจมีไม่ครบถ้วนทุกส่วนอย่างที่ได้ยกมา ซึ่งรายละเอียดโครงสร้างตลอดจนรายละเอียดในการทำงานของแท็กเบอร์ใดๆ ก็สามารถดูได้จาก Data Sheet ของบริษัทผู้ผลิตแท็กเบอร์นั้นๆ

Transponder หรือ Tags มีลักษณะเป็นไมโครชิพ (microchip) ที่ยอมให้ผู้ใช้ติดเข้าระหว่างชั้นของกระดาษหรือพลาสติกที่ใช้ทำป้ายฉลาก ชิพหรือแท็กอาจมีรูปร่างได้หลายแบบขึ้นอยู่กับการนำไปใช้งานโดยอาจมีรูปร่างเหมือนบัตรเครดิตในการใช้งานทั่วไป หรือเล็กขนาดไส้ดินสอยากเพียง 10 มิลลิเมตร เพื่อฝังเข้าไปใต้ผิวหนังสัตว์ในกรณีนำไปใช้ในงานปศุสัตว์ หรืออาจมีขนาดใหญ่มากสำหรับแท็กที่ใช้ติดกับเครื่องจักรขณะทำการขนส่ง แท็กอาจนำไปติดไว้กับสินค้าในร้านปลีกทั่วไปเพื่อป้องกันขโมย โดยจะมีการติดตั้งสายอากาศของตัวอ่านข้อมูลขนาดใหญ่ไว้ตรงประตูทางออกเพื่อทำการตรวจจับขโมย โดยแท็กจะรับพลังงานจากสัญญาณ RF เพื่อติดต่อสื่อสารกับเครื่องอ่าน หรือใช้พลังงานจากแบตเตอร์รี่ที่บรรจุภายในป้าย ซึ่งเป็นแบตเตอร์รี่ Lithium-Ion มีอายุการใช้งานที่ยาวนาน จึงมักนำมาใช้กับแผ่นป้ายนี้

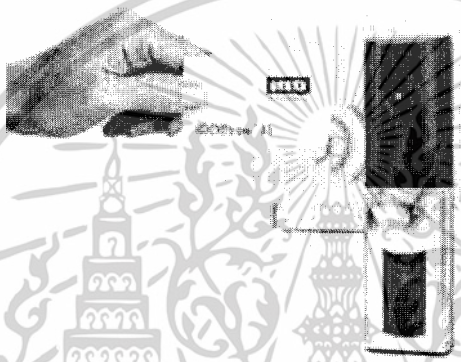


รูปที่ 2.2 ระบบการอ่าน / เขียนข้อมูลอย่างง่ายของ RFID

แท็กประกอบไปด้วยสายอากาศที่มีขนาดเล็กที่จะช่วยให้แท็กตอบสนองกับเครื่องอ่าน โดยสายอากาศจะแผ่สัญญาณวิทยุจำนวนหนึ่งออกมา เพื่อกระตุ้นให้แท็กอ่านหรือเขียนข้อมูลลงไปสายสายอากาศสามารถทำได้ทุกขนาดและรูปร่าง เพื่อที่จะสามารถออกแบบให้ติดตั้งได้ทุกที่ และเพื่อให้เกิดความครอบคลุมได้ดีที่สุดในหลายๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสายอากาศจะถูกติดไปโดยตรงกับ Transceiver เหมือนกับเป็นอุปกรณ์ติดกัน ชิพที่อยู่ในแท็กจะมีหน่วยความจำซึ่งอาจเป็นแบบอ่านได้อย่างเดียว (ROM) หรือทั้งอ่านทั้งเขียน (RAM) ทั้งนี้ขึ้นอยู่กับความต้องการในการใช้งาน โดยปกติหน่วยความจำแบบ ROM จะใช้เก็บข้อมูลเกี่ยวกับการรักษาความปลอดภัย เช่นข้อมูลของบุคคลที่มีสิทธิผ่านเข้าออกในบริเวณที่มีการควบคุมหรือระบบปฏิบัติการ ในขณะที่ RAM จะใช้เก็บข้อมูลชั่วคราวในระหว่างแท็กและตัวอ่านข้อมูลทำการติดต่อสื่อสารกัน นอกจากนี้อาจมีการนำหน่วยความจำแบบ EEPROM มาใช้ในกรณีต้องการเก็บข้อมูลในระหว่างที่แท็กและตัวอ่านข้อมูลทำการสื่อสาร และข้อมูลยังคงอยู่ถึงแม้จะไม่มีพลังงานไฟฟ้าป้อนให้แก่แท็ก



รูปที่ 2.3 แสดงตัวอย่างการใช้งานแท็กและตัวอ่านข้อมูล (Reader)

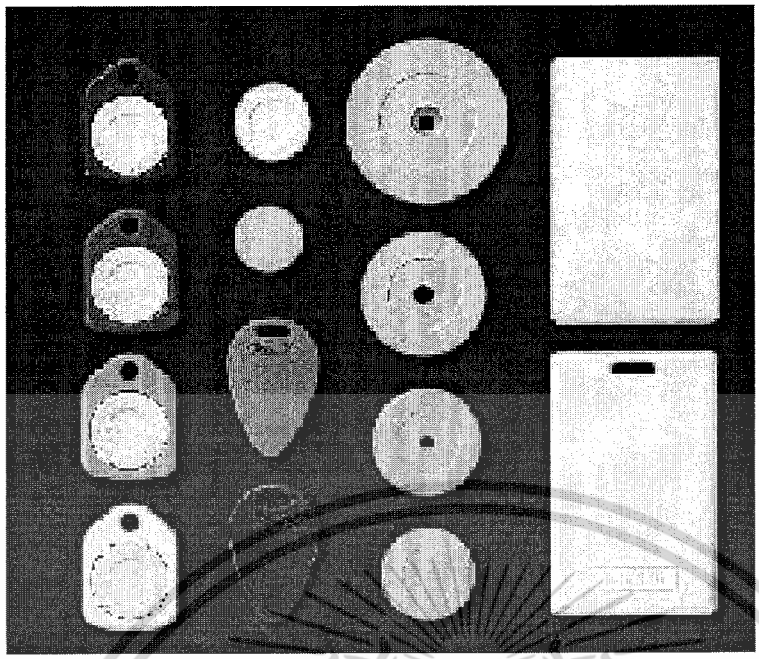
นอกจากการแท็กแบ่งจากชนิดที่ว่ามาแล้วแท็กก็ยังถูกแบ่งประเภทจากรูปแบบในการใช้งานได้เป็น 3 แบบ คือ

1. แบบที่สามารถถูกอ่านและเขียนข้อมูลได้อย่างอิสระ (Read – write)
2. แบบเขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้อย่างอิสระ (Write-One, Read-Many หรือ WORM)
3. แบบอ่านได้เพียงอย่างเดียว (Read-Only)

2.1.3 รูปแบบของ Tag

โครงสร้างภายในของแท็กจะประกอบด้วย 2 ส่วนใหญ่ๆ ได้แก่ ขดลวดขนาดเล็กซึ่งทำหน้าที่เป็นสายอากาศ (Antenna) สำหรับรับส่งสัญญาณคลื่นความถี่วิทยุและสร้างพลังงานป้อนให้ส่วนของไมโครชิพ (Microchip) ที่ทำหน้าที่เก็บข้อมูลของวัตถุ เช่น รหัสสินค้า โดยทั่วไปตัวแท็กอาจอยู่ในชนิดทั้งเป็นกระดาษ แผ่นฟิล์ม พลาสติก มีขนาดและรูปร่างต่างๆ กันไป ทั้งนี้ ขึ้นอยู่กับวัสดุที่จะนำเอาไปติด และมีหลายรูปแบบ เช่น ขนาดเท่ากับบัตรเครดิต เหรียญ กระดุม ฉลากสินค้า แคปซูล เป็นต้น (พิจารณารูปที่ 2.4) แต่โดยหลักการอาจแบ่ง Tag ที่มีใช้งานกันอยู่นั้นจะมีอยู่ 2 ชนิดใหญ่ๆ โดยแต่ละชนิดก็จะแตกต่างกันในแง่ของการใช้งาน ราคาโครงสร้าง และหลักการทำงานอยู่ ซึ่งจะกล่าวถึงและอธิบายเป็นหัวข้อดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดง RFID tag ในรูปแบบต่างๆ

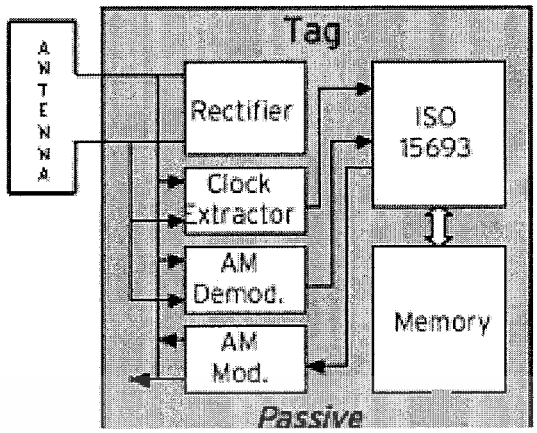
2.1.4 ชนิดของ Tag

- Passive tag
- Active tag

Passive RFID Tags

Tag ชนิดนี้ไม่ต้องอาศัยแหล่งจ่ายไฟภายนอกใดๆ เพราะภายใน Tag จะมีวงจรกำเนิดไฟฟ้าเหนี่ยวนำขนาดเล็กเป็นแหล่งจ่ายไฟในตัวทำให้การอ่านข้อมูลทำได้ไม่ไกลมากนัก ระยะอ่านสูงสุดประมาณ 1 เมตรขึ้นอยู่กับความแรงของเครื่องส่งและคลื่นความถี่วิทยุที่ใช้ โดยปกติ tag ชนิดนี้มักมีหน่วยความจำขนาดเล็ก โดยทั่วไปประมาณ 16 ถึง 1,024 ไบต์ มีขนาดเล็กและน้ำหนักเบา ราคาต่อหน่วยต่ำไอซีของ tag ชนิด passive ที่มีการผลิตออกมาจะมีทั้งขนาดและรูปร่างเป็นได้ตั้งแต่แท่งหรือแผ่นขนาดเล็กจนแทบไม่สามารถมองเห็นได้ไปจนถึงขนาดใหญ่สะดุดตา ซึ่งต่างก็มีความเหมาะสมกับชนิดงานที่แตกต่างกันโดยทั่วไปโครงสร้างภายในส่วนที่เป็นไอซีของ tag นั้นก็จะประกอบด้วย 3 ส่วนหลักๆ ได้แก่ส่วนของควบคุมการทำงานของภาครับส่งสัญญาณวิทยุ (Analog Front-End) ส่วนควบคุมภาคลอจิก (Digital Control Unit) ส่วนของหน่วยความจำ (Memory) ซึ่งอาจจะเป็นแบบ ROM หรือ EEPROM

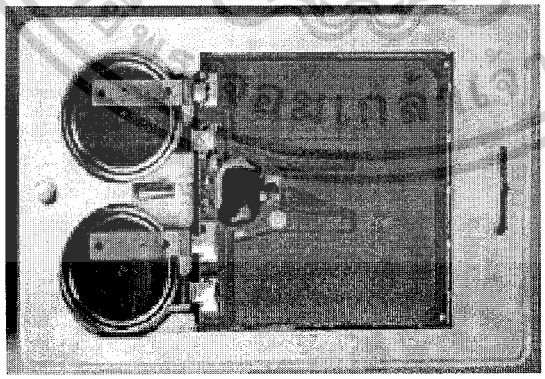
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงบล็อกไดอะแกรมของ Passive tag

Active RFID Tags

Tag ชนิดนี้จะต้องอาศัยแหล่งจ่ายไฟจากแบตเตอรี่ภายนอก เพื่อจ่ายพลังงานให้กับวงจรภายในทำงานโดย tag แบบนี้สามารถมีหน่วยความจำภายในขนาดใหญ่ๆได้ถึง 1 เมกกะไบต์ และสามารถอ่านได้ในระยะไกลสูงสุดประมาณ 10 เมตรแม้ว่า tag ชนิดนี้จะมีข้อดีอยู่หลายข้อแต่ก็มีข้อเสียอยู่ด้วยเหมือนกัน เช่น มีราคาต่อ หน่วยมีขนาดค่อนข้างใหญ่ และมีระยะเวลาในการทำงานที่จำกัดนอกจากการแบ่งจากชนิดที่ว่ามาแล้ว tag ก็ยังถูกแบ่งประเภทจากรูปแบบในการใช้งานได้เป็น 3 แบบ คือ แบบที่สามารถถูกอ่านและเขียนข้อมูลได้อย่างอิสระ (Red-Write), แบบเขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้อย่างอิสระ (Write-Once Read-Many หรือ WORM) และแบบอ่านได้เพียงอย่างเดียว (Read-Only) ด้วย อย่างไรก็ตามเนื่องจากในการใช้งานจะนิยมใช้ Tag ชนิด passive มากกว่า และในที่นี้จะกล่าวถึงเฉพาะ tag ชนิดนี้เป็นหลัก

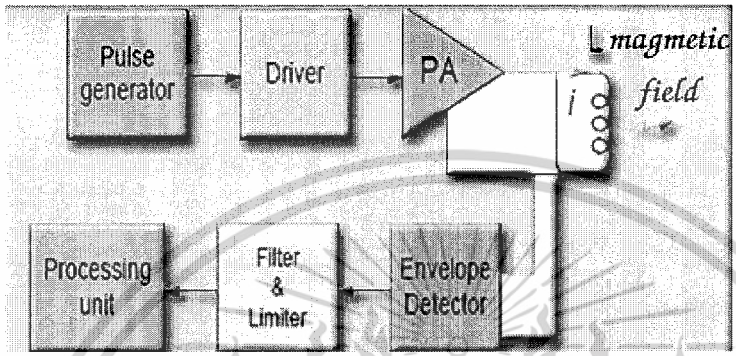


รูปที่ 2.6 แสดงรูปตัวอย่าง Active tag ที่มีแบตเตอร์รี่ lithium, 2 ก้อนอยู่ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 เครื่องอ่าน (eader)

โดยหน้าที่ของเครื่องอ่านก็คือ การเชื่อมต่อเพื่อเขียนหรือข้อมูลลง tag ด้วยสัญญาณความถี่วิทยุภายในเครื่องอ่านจะประกอบด้วย เสาอากาศที่ทำจากขดลวดทองแดง เพื่อใช้รับส่งสัญญาณ ภาครับและภาคส่งสัญญาณวิทยุ และวงจรควบคุมการอ่าน-เขียนข้อมูลจำพวกไมโครคอนโทรเลอร์ และส่วนของการติดต่อกับคอมพิวเตอร์



รูปที่ 2.7 แสดงโครงสร้างภายในตัวอ่าน

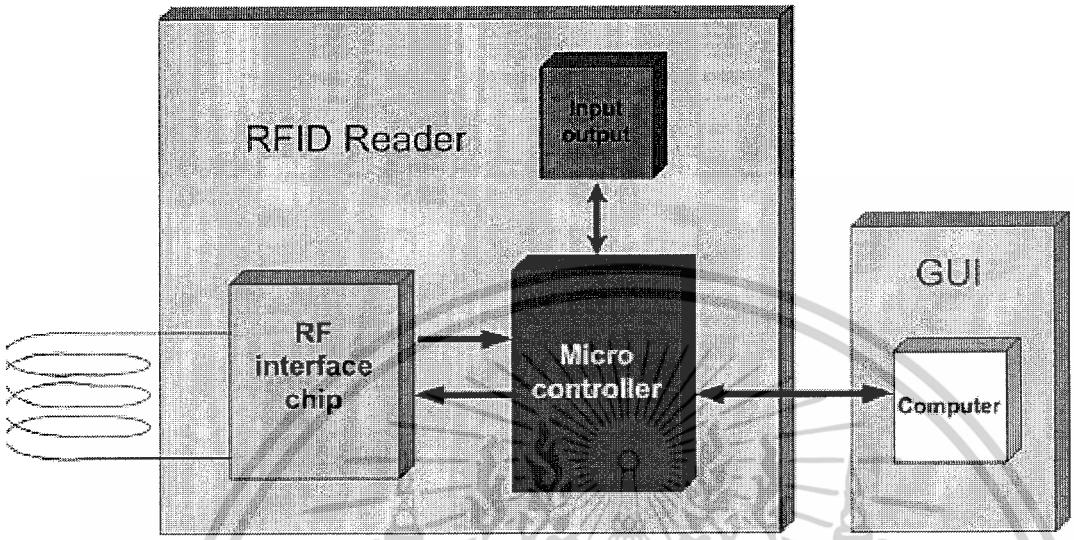
โดยทั่วไปเครื่องอ่านจะประกอบด้วยส่วนประกอบหลักดังนี้

- ภาครับและส่งสัญญาณวิทยุ
- ภาคสร้างสัญญาณพาหะ
- ขดลวดที่ทำหน้าที่เป็นสายอากาศ
- วงจรจูนสัญญาณ
- หน่วยประมวลผลข้อมูล และภาคติดต่อกับคอมพิวเตอร์

โดยทั่วไปหน่วยประมวลข้อมูลที่อยู่ภายในเครื่องอ่านมักใช้เป็นไมโครคอนโทรเลอร์ ซึ่งอัลกอริทึมที่อยู่ภายในโปรแกรม จะทำหน้าที่ถอดรหัสข้อมูล (Decoding) ที่ได้รับและทำหน้าที่ติดต่อกับคอมพิวเตอร์โดยลักษณะขนาด และรูปร่างของเครื่องอ่านจะแตกต่างกันไปตามประเภทของการใช้งาน เช่น แบบมือถือขนาดเล็กหรือ ติดผนัง จนไปถึงขนาดใหญ่เท่าประตู (Gate size) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 ส่วนประกอบของเครื่องอ่าน RFID



รูปที่ 2.8 แสดงส่วนประกอบของเครื่องอ่าน RFID

RF Interface Chip

- แปลงสัญญาณ Analog เป็น digital ในกรณีที่ส่งข้อมูลจาก Coil ไปยัง MCU
- แปลงสัญญาณ digital เป็น analog ในกรณีที่ส่งข้อมูลจาก MCU ไปยัง Coil

MCU (Micro Controller Unit) เป็นหน่วยประมวลผลของ Controller

Microcontroller

- ควบคุมการทำงานของ RFID Reader
- ถอดรหัสสัญญาณตามรูปแบบที่เข้ารหัสสัญญาณมา

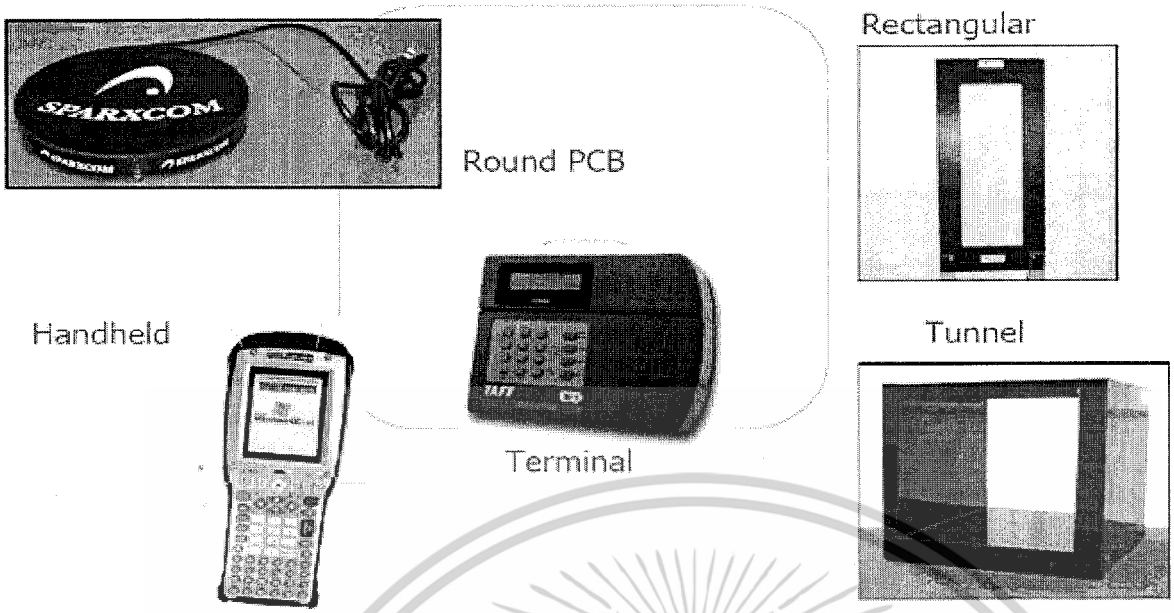
Input/Output

- เป็นตัวรับข้อมูลจากภายนอกเช่น Keypad เป็นต้น
- เป็นอุปกรณ์ที่แสดงหรือควบคุมอุปกรณ์ที่อยู่ภายนอกเช่น 7-Segment, LCD เป็นต้น

Graphic Interface User

- แสดงผลของข้อมูลให้อยู่ในรูปแบบที่ผู้ใช้งานสามารถเข้าใจได้ง่าย
- ควบคุมการทำงานของ RFID Reader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

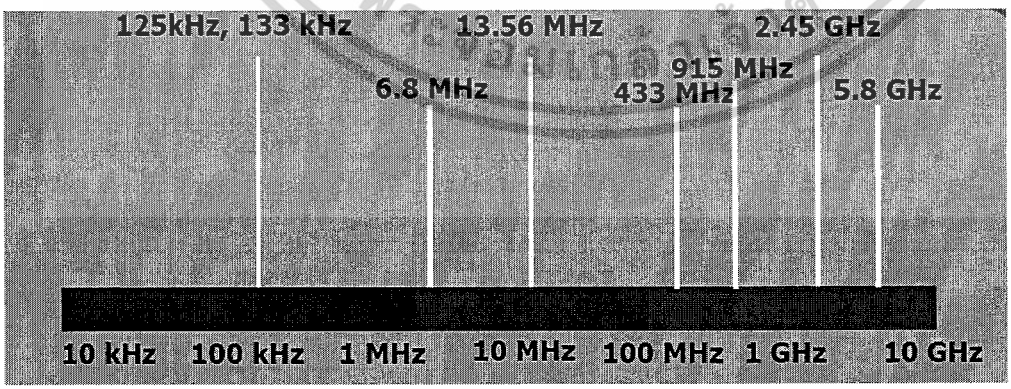


รูปที่ 2.9 ตัวอย่างรูปร่างของ Reader ต่างๆ

2.2 Carrier ในระบบ RFID

ในปัจจุบันคลื่นพาหะที่ใช้งานกันในระบบ RFID จะอยู่ในย่านความถี่ ISM (Industrial-Scientific-Medical) ซึ่งเป็นย่านความถี่ที่กำหนดในการใช้งานในเชิงอุตสาหกรรม วิทยาศาสตร์ และการแพทย์ สามารถใช้งานได้โดยไม่ตรงกับย่านความถี่ที่ใช้งานในการสื่อสารโดยทั่วไป โดยมี 3 ย่านความถี่ที่ใช้งาน คือสำหรับคลื่นพาหะที่ใช้กันในระบบ FRID อาจ แบ่งออกได้เป็น 3 ย่านหลัก ได้แก่

- ย่านความถี่ต่ำ (Low Frequency: LH) ต่ำกว่า 150 kHz
- ย่านความถี่สูง (High Frequency: HF) 13.56 MHz
- ย่านความถี่สูงยิ่ง (Ultra High Frequency: UHF) 433/868/915 MHz



รูปที่ 2.10 แสดงความถี่ที่ระบบ RFID ถูกใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

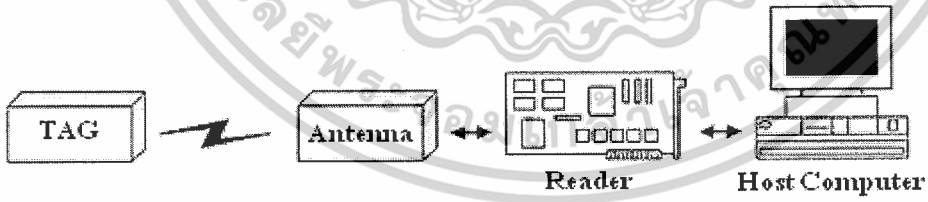
ในแง่การใช้งาน 2 ย่านความถี่แรกจะเหมาะสำหรับใช้ กับงานที่มีระยะการสื่อสารข้อมูลในระยะใกล้ (LH ระยะอ่านประมาณ 10-20 เซนติเมตร และ HF ระยะอ่านประมาณ 1 เมตร) เช่น การตรวจสอบการผ่านเข้าออกพื้นที่ การตรวจหาและเก็บประวัติในสัตว์ ส่วนย่านความถี่สูงยิ่งจะถูกใช้กับงานที่มีระยะการสื่อสารข้อมูลในระยะไกล (UHFระยะอ่าน ประมาณ 1-10 เมตร) เช่น ระบบเก็บค่าบริการทางด่วน และในปัจจุบันระบบ RFID กำลังถูกวิจัยและพัฒนาในย่านความถี่ไมโครเวฟที่ความถี่ 2.4 GHz และความถี่ 5.8 GHz เพื่อใช้งานที่ต้องการระยะอ่านที่ไกลกว่า 10 เมตร เป็นต้น

ในแง่ของราคาและความเร็วในการสื่อสารข้อมูล เมื่อเทียบกันแล้ว RFID ซึ่งใช้คลื่นพาหะย่านความถี่สูงเป็นระยะที่มีความเร็วในการส่งข้อมูลสูงสุดและมีราคาแพงที่สุดด้วยเช่นกัน ส่วน RFID ที่ใช้คลื่นพาหะในอีก 2 ย่านความถี่จะมีระดับราคาและความเร็วลดหลั่นกันไป

2.2.1 หลักการทำงานและเทคนิคการรับและการส่งข้อมูลของ RFID

ลักษณะการทำงานของระบบ RFID

หัวใจของเทคโนโลยี RFID ได้แก่ "Inlay" ที่บรรจุอุปกรณ์และวงจรอิเล็กทรอนิกส์กับโลหะที่ยืดหยุ่นได้สำหรับการติดตามหรือทำหน้าที่เป็นเสาอากาศนั่นเอง Inlay มีความหนาสูงสุดอยู่ที่ 0.375 มิลลิเมตร สามารถทำเป็นแผ่นบางอัดเป็นชั้นๆ ระหว่างกระดาษ, แผ่นฟิล์ม หรือพลาสติกก็ได้ ซึ่งเป็นการผลิตเครื่องหมายหรือฉลาก จากวัสดุที่มีราคาไม่แพงมากนัก ซึ่งจะเห็นว่า Inlay มีลักษณะรูปร่างที่บางมากจึงทำให้ง่ายต่อการติดเป็นป้ายชื่อหรือฉลากของชิ้นงานหรือวัตถุนั้นๆ ได้สะดวก RFID เป็นระบบที่นำเอาคลื่นวิทยุมาเป็นคลื่นพาหะเพื่อใช้ในการสื่อสารข้อมูลระหว่างอุปกรณ์สองชนิดที่เรียกว่า แท็ก (Tag) และตัวอ่านข้อมูล (Reader หรือ Interrogator) ซึ่งเป็นการสื่อสารแบบไร้สาย (Wireless) โดยการนำข้อมูลที่ต้องการส่ง มาทำการมอดูเลต (Modulation) กับคลื่นวิทยุแล้วส่งออกผ่านทางสายอากาศที่อยู่ในตัวรับข้อมูล ดังแผนผังการทำงานของระบบ RFID ในรูปที่ 2.11



รูปที่ 2.11 แสดงแผนผังการทำงานของระบบ RFID

การประยุกต์ใช้งาน RFID จะมีลักษณะการใช้งานที่คล้ายกับบาร์โค้ด (Barcode) และยังสามารถรองรับความต้องการอีกหลายอย่างที่บาร์โค้ดไม่สามารถตอบสนองได้ เนื่องจากบาร์โค้ดจะเป็นระบบที่อ่านได้อย่างเดียว (Reader only) ไม่สามารถทำการเปลี่ยนแปลงข้อมูลที่อยู่บนบาร์โค้ดได้ แต่แท็กของระบบ RFID จะสามารถทั้งอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และบันทึกข้อมูลได้ ดังนั้นเราจึงสามารถเปลี่ยนแปลง หรือทำการบันทึกข้อมูลที่อยู่ในแท็กได้ตามความต้องการของผู้ใช้งานนอกจากนี้ระบบ RFID ยังสามารถใช้งานได้แม้ในขณะที่วัตถุที่กำลังเคลื่อนที่ เช่นในขณะสินค้ากำลังเคลื่อนที่อยู่บนสายพานการผลิต (Conveyor) หรือในบางประเทศก็มีการใช้ระบบ RFID ในการเก็บค่าผ่านทางด่วนที่ผู้ใช้บริการทางด่วนไม่ต้องหยุดรถเพื่อจ่ายค่าบริการผู้ใช้บริการทางด่วนจะมีแท็กติดอยู่กับรถ และแท็กจะทำการสื่อสารกับตัวอ่านข้อมูล ผ่านสายอากาศขนาดใหญ่ที่ติดตั้งอยู่ตรงบริเวณทางขึ้นทางด่วน ในขณะที่รถแล่นผ่านสายอากาศ ตัวอ่านข้อมูลก็จะคิดค่าบริการและบันทึกจำนวนเงินที่เหลือลงในแท็กโดยอัตโนมัติ หรือแม้กระทั่งการใช้งานในปศุสัตว์เพื่อบันทึกประวัติ หรือระบุความแตกต่างของสัตว์แต่ละตัวที่อยู่ในฟาร์มข้อดีของระบบ RFID อีกอย่างก็คือแท็กและตัวอ่านข้อมูลสามารถสื่อสารผ่านตัวกลางได้หลายอย่างเช่น น้ำ, พลาสติก, กระจก, หรือวัสดุทึบแสงอื่นๆ ในขณะที่บาร์โค้ดทำไม่ได้

หลักการทำงานเบื้องต้นของระบบ RFID

1. ตัวอ่านข้อมูลจะปล่อยคลื่นแม่เหล็กไฟฟ้าออกมาตลอดเวลา และคอยตรวจจับว่ามีแท็กเข้ามาอยู่ในบริเวณสนามแม่เหล็กไฟฟ้าหรือไม่ หรืออีกนัยหนึ่งก็คือการคอยตรวจจับว่ามีการมอดูเลตสัญญาณเกิดขึ้นหรือไม่

2. เมื่อมีแท็กเข้ามาอยู่ในบริเวณสนามแม่เหล็กไฟฟ้า แท็กจะได้รับพลังงานไฟฟ้าที่เกิดจากการเหนี่ยวนำของคลื่นแม่เหล็กไฟฟ้าเพื่อให้แท็กเริ่มทำงาน และจะส่งข้อมูลในหน่วยความจำที่ผ่านการมอดูเลตกับคลื่นพาหะแล้วออกมาทางสายอากาศที่อยู่ภายในแท็ก

3. คลื่นพาหะที่ถูกส่งออกมาจากแท็กจะเกิดการเปลี่ยนแปลงแอมปลิจูด, ความถี่ หรือเฟส ขึ้นอยู่กับวิธีการมอดูเลต

4. ตัวอ่านข้อมูลจะตรวจจับความเปลี่ยนแปลงของคลื่นพาหะแปลงออกมาเป็นข้อมูลแล้วทำการถอดรหัสเพื่อนำข้อมูลไปใช้งานต่อไป

2.2.2 วิธีการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่าน

โดยมากมักจะใช้วิธีการมอดูเลตทางแอมปลิจูดหรือใช้การมอดูเลตทางแอมปลิจูดบวกกับการเข้ารหัสแมนเชสเตอร์ (Manchester encoded AM) แต่ในปัจจุบันก็มีแท็กที่ใช้การมอดูเลตแบบอื่นๆ ด้วย เช่น การมอดูเลชั่นแบบเฟสชีฟคีย์อิ้ง (Phase Shift Keying: PSK) ฟรีเควนซีชีฟคีย์อิ้ง (Frequency Shift Keying: FSK) หรือการใช้การมอดูเลตทางความถี่ (Frequency Modulation: FM)

ในการรับส่งข้อมูลหรือสัญญาณวิทยุระหว่างแท็กกับเครื่องอ่าน จะได้อย่างมีประสิทธิภาพต่อเมื่อสายอากาศมีความยาวที่เหมาะสมกับความถี่พาหะที่ใช้งาน เช่น เมื่อความถี่ใช้งานเป็น 13.56 เมกะเฮิรตซ์ ความยาวของเสาอากาศ (เป็นเส้นตรง) ที่เหมาะสมก็คือ 22.12 แน่นอนว่าในทางปฏิบัติคงไม่สามารถนำเสาอากาศที่ใหญ่ขนาดนั้นมาใช้งานกับแท็กขนาดเล็กได้ สายอากาศที่ดูจะเหมาะจะใช้ร่วมกับแท็กมากที่สุดก็คือ สายอากาศที่เป็นขดลวดขนาดเล็กหรือที่มีชื่ออย่างเป็นทางการว่าสายอากาศแบบแมกเนติกไดโพล (Magnetic dipole Antenna) รูปแบบของสายอากาศแบบนี้ก็จะมีอยู่หลากหลายทั้งแบบที่เป็นขดลวดพันแกนอากาศหรือแกนเฟอร์ไรต์ แบบที่เป็นวงลูปที่ทำขึ้นจากลายทองแดงบนแผ่นวงจรพิมพ์ ทั้งที่เป็นลูปแบบวงกลมและสี่เหลี่ยม ทั้งนี้ความเหมาะสมในการใช้งานก็แตกต่างกันไปตามถี่พาหะและประเภทของงานด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากการรับส่งข้อมูลแล้วสายอากาศก็ยังทำหน้าที่เป็นแหล่งจ่ายไฟให้กับแท็กด้วย โดยอาศัยหลักการทำงานตามแนวคิดของไมเคิล ฟาราเดย์ เรื่องแรงดันเหนี่ยวนำในขดลวดที่เกิดขึ้นจากเส้นแรงแม่เหล็ก (จากเครื่องอ่าน) ที่มีค่าเปลี่ยนแปลงไปตามเวลา (Time-varying magnetic field) พุ่งผ่านสายอากาศของแท็ก เมื่อแท็กและเครื่องอ่านตั้งอยู่ห่างกันในระยะ 0.16 เท่าของความยาวของคลื่นพาหะที่ใช้ เรียกปรากฏการณ์ที่เกิดขึ้นว่า Transformer-type Coupling ซึ่งเป็นปรากฏการณ์แบบเดียวกับการเกิดแรงดันไฟฟ้าเหนี่ยวนำขึ้นระหว่างขดลวดปฐมภูมิ (Primary) และขดลวดทุติยภูมิ (Secondary) ในทรานสฟอร์เมอร์ (Transformer) จะเป็นวงจรพื้นฐานสำหรับอธิบายกลไกที่เกิดขึ้นในการส่งข้อมูลของแท็ก

การป้องกันการชนกันของสัญญาณข้อมูล (Anti-Collision)

ในการที่จะรับข้อมูลจากแท็กหลายๆ อัน ทั้งแท็กและตัวเครื่องอ่านต้องได้รับการออกแบบให้รองรับสภาวะที่มีแท็กมากกว่า 1 อันทำงาน (ส่งสัญญาณ) มิเช่นนั้นแล้วสัญญาณพาหะก็จะมีการส่งออกในเวลาเดียวกันทำให้เกิดการชนของสัญญาณ (Collision) จะทำให้ไม่มีข้อมูลใดๆ ส่งถึงตัวเครื่องอ่านเลยการติดต่อระหว่างแท็กกับตัวเครื่องอ่านเปรียบเสมือน บัสแบบอนุกรม แต่บัสชนิดนี้จะใช้อากาศเป็นตัวกลางในการส่งสัญญาณ ในระบบบัสที่ใช้เคเบิ้ลเป็นตัวกลางก็ต้องมีการควบคุมไม่ให้เกิดการชนกันของสัญญาณ RFID ก็จำเป็นที่จะต้องมีการป้องกันให้มีการส่งสัญญาณจากแท็กอันเดียวต่อช่วงเวลานั้นเช่นกัน

อัตราการรับส่งข้อมูลและแบนด์วิดธ์

อัตราการรับส่งข้อมูล (Data Transfer Rate) จะขึ้นอยู่กับความถี่ของคลื่นพาหะ โดยปกติถ้าความถี่ของคลื่นพาหะยิ่งสูง อัตราการรับส่งข้อมูลก็จะยิ่งสูงตามไปด้วย ส่วนการเลือกแบนด์วิดธ์ หรือย่านความถี่นั้นก็จะมีผลต่ออัตราการรับส่งข้อมูลเช่นกันโดยมีหลักว่า แบนด์วิดธ์ควรจะมีค่ามากกว่าอัตราการรับส่งข้อมูลที่ต้องการอย่างน้อยสองเท่า ยกตัวอย่างเช่น ถ้าใช้แบนด์วิดธ์ในช่วง 2.4-2.5 GHz ก็จะสามารถรองรับอัตราการรับส่งข้อมูลได้ถึงประมาณ 2 Megabits ต่อวินาที เป็นต้น แต่การใช้แบนด์วิดธ์ที่กว้างเกินไปก็อาจทำให้เกิดปัญหาเกี่ยวกับสัญญาณรบกวนมาก หรือทำให้ S/N Ratio ต่ำลงนั่นเอง ดังนั้นการเลือกใช้แบนด์วิดธ์ให้ถูกต้องก็เป็นส่วนสำคัญในการพิจารณา

ระยะการรับส่งข้อมูลและกำลังส่ง

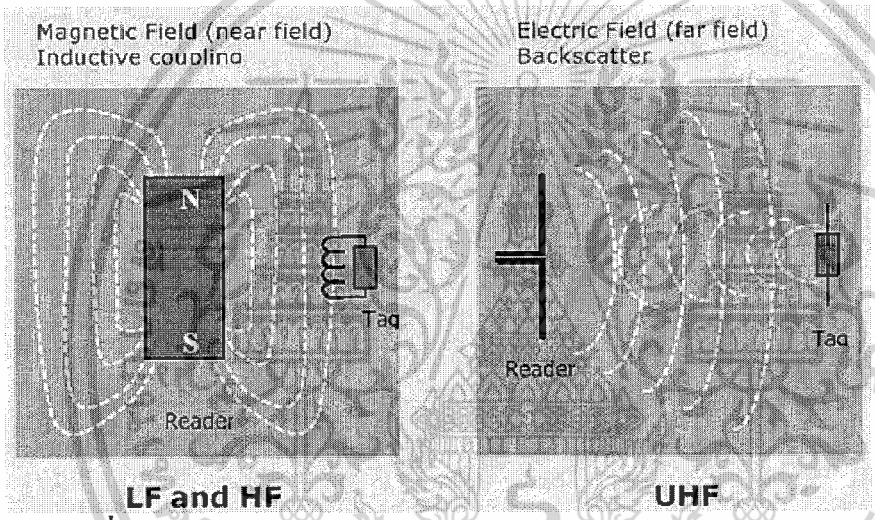
ระยะการรับส่งข้อมูลในระบบ RFID ขึ้นอยู่กับปัจจัยสำคัญต่างๆ คือ กำลังส่งของตัวอ่านข้อมูล (Reader/Interrogator Power) กำลังส่งของแท็ก (Tag Power) และสภาพแวดล้อม ส่วนการออกแบบสายอากาศของตัวอ่านข้อมูล จะเป็นตัวกำหนดลักษณะรูปร่างของคลื่นแม่เหล็กไฟฟ้าที่แผ่กระจายออกมาจากสายอากาศ ดังนั้นระยะการรับส่งข้อมูล บางทีอาจขึ้นอยู่กับมุมของการรับส่งระหว่างแท็กและตัวอ่านข้อมูลด้วยเช่นกัน ทั้งนี้ขึ้นอยู่กับรูปร่างของคลื่นแม่เหล็กไฟฟ้าเป็นสำคัญความเข้มของคลื่นแม่เหล็กไฟฟ้าโดยทั่วไปจะลดลงตามระยะทางโดยแปรผกผันกับระยะทางยกกำลังสอง แต่ในบางสภาพแวดล้อมซึ่งอาจมีการสะท้อนกลับของคลื่นแม่เหล็กไฟฟ้าจากสิ่งต่างๆรอบตัว เช่น โลหะ ก็อาจทำให้ความเข้มของคลื่นแม่เหล็กไฟฟ้าลดลงอย่างรวดเร็ว โดยอาจแปรผกผันกับระยะทางยกกำลังสี่ ปรากฏการณ์เช่นนี้เราเรียกว่า “Multi-path Altenuation” ซึ่งจะส่งผลให้ระยะการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สั้นลง หรือแม้กระทั่งความชื้นในอากาศก็อาจมีผลในกรณีที่ความถี่สูงๆ ดังนั้นการนำระบบ RFID ไปใช้งานก็ควรมีการคำนึงถึงสภาพแวดล้อมเพราะจะมีผลกระทบกับระยะการรับส่งข้อมูล และพยายามติดตั้งระบบให้ห่างไกลจากโลหะ ซึ่งอาจทำให้เกิดการสะท้อนของคลื่นแม่เหล็กไฟฟ้าได้ กำลังส่งของแท็กที่จะส่งกลับมายังตัวอ่านข้อมูลนั้นโดยทั่วไปจะมีกำลังที่ต่ำมากเมื่อเทียบกับกำลังส่งของ ตัวอ่านข้อมูล ดังนั้นความไวในการตรวจจับสัญญาณของตัวอ่านข้อมูล ก็เป็นอีกจุดหนึ่งที่ต้องพิจารณา ถึงแม้ในทางเทคนิคเราจะสามารถทำให้ตัวอ่านความถี่ ดังนั้นในระบบ RFID โดยทั่วๆ ไปจะมีกำลังส่งเพียงระหว่าง 100-500 mW

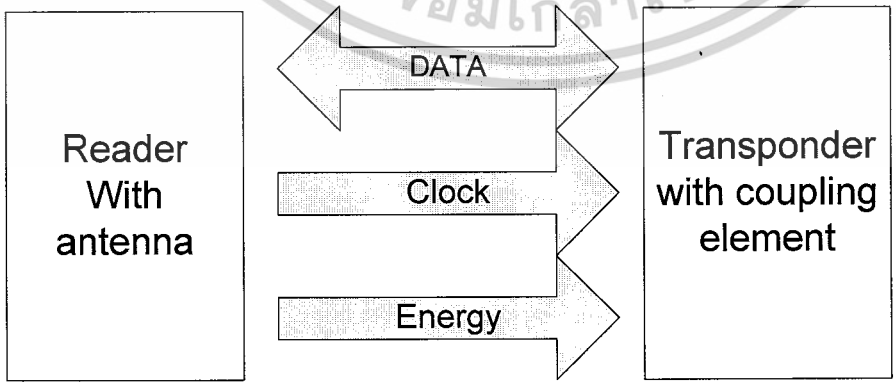
การส่งข้อมูลระหว่างตัวอ่านและ Tag มี 2 วิธีคือ

1. Inductive Coupling (LF and HF)
2. Backscatter Coupling (UHF and Microwave)



รูปที่ 2.12 แสดงการการส่งข้อมูลระหว่างตัวอ่านและ Tag รูปแบบต่างๆ

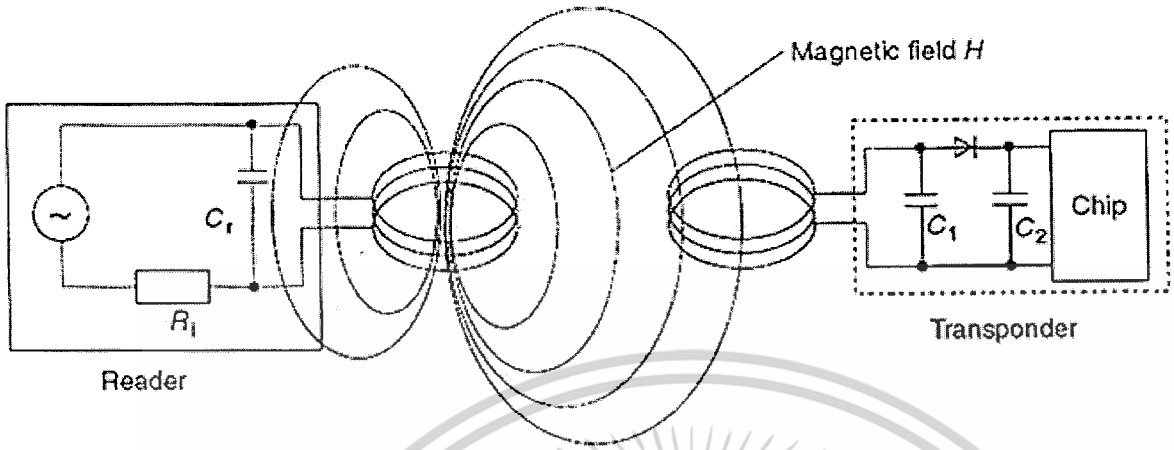
ที่นี้เราจะขอกล่าวถึงการส่งข้อมูลระหว่างตัวอ่านและ Tag แบบ inductive coupling



รูปที่ 2.13 การสื่อสารระหว่าง Reader กับ Tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Inductive Coupling



รูปที่ 2.14 แสดงการสร้าง Power supply ให้กับ tag โดย reader

จากรูปเราจะเห็นว่ารูปแบบของ Inductive Coupling จะอาศัยการเชื่อมโยงของสนามแม่เหล็ก โดยจะถูก generate ด้วยเครื่องอ่านเพื่อเป็นแหล่งจ่ายให้กับ Tag

2.2.3 Power supply to passive transponder

การเหนี่ยวนำของสนามแม่เหล็กที่ถูก เชื่อมโยงจากเครื่องอ่านไปยัง Tag จะประกอบด้วย ข้อมูลของสัญญาณไมโครชิพและส่วนของพื้นที่ของขดลวดซึ่งอยู่ในส่วนของสายอากาศ หมายถึงว่ากำลังงานทั้งหมดที่ต้องการสำหรับการดำเนินการของไมโครชิพที่อยู่ส่วนของ tag จะถูกจัดการหรือจ่ายให้โดยเครื่องอ่าน (Reader) ตามรูปที่ 2.14 สำหรับในส่วนนี้สายอากาศของเครื่องอ่านจะทำการสร้างสนามแม่เหล็กไฟฟ้าความถี่สูงที่ทะลุผ่านและตัดตามขวางของขดลวดและรอบๆขดลวด เพราะว่าความยาวคลื่นของช่วงความถี่ที่ถูกใช้ (<135 kHz : 2400 m, 13.56 MHz : 22.1 m) และระยะทางมันไกลกว่าระหว่างระยะทางสายอากาศของเครื่องอ่านและสายอากาศของ Tag ดังนั้นสนามแม่เหล็กไฟฟ้าอาจจะถูกพิจารณาหรือจัดการการเปลี่ยนแปลงของสนามที่ทะลุผ่านเข้าไปในสายอากาศของ Tag ที่ระยะทางจากเครื่องอ่านและแรงดันจะถูกสร้างที่ขดลวดสายอากาศของ Tag โดยการเหนี่ยวนำและแรงดันนี้จะถูก rectifier และก็เป็นแหล่งจ่ายของไมโครชิพที่อยู่ใน Tag

2.2.4 การ Interface ข้อมูลของ RFID

การ Interface ข้อมูลของ RFID เราจะแยกเป็น 2 แบบคือ

- Data transfer reader – tag
- Data transfer tag – reader

Data transfer reader-tag

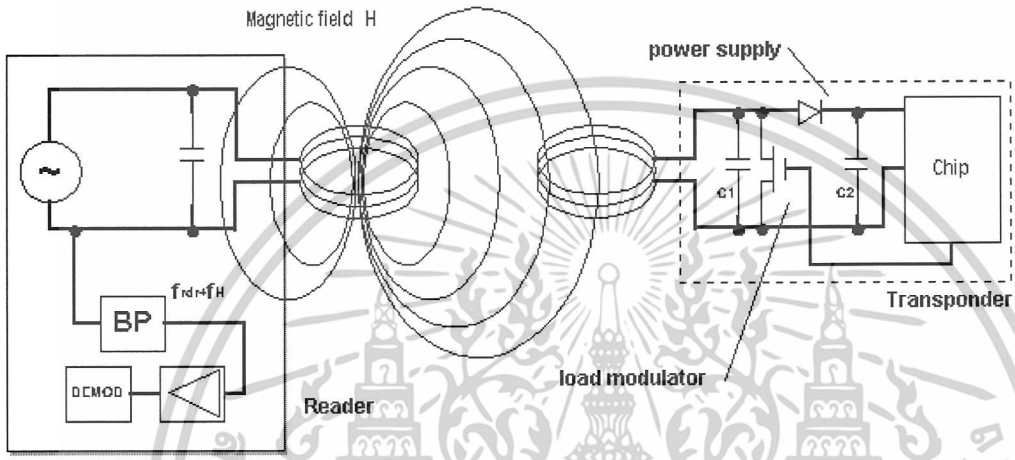
การส่งจาก Reader – Tag จะใช้การมอดูเลชั่นแบบดิจิตอลคือ Reader จะแปลงสัญญาณดิจิตอลเป็นอนาล็อกเพื่อส่งไปให้ Tag การมอดูเลตแบบ ASK จะใช้ในการส่งผ่านข้อมูลจาก Reader ไปยัง Tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

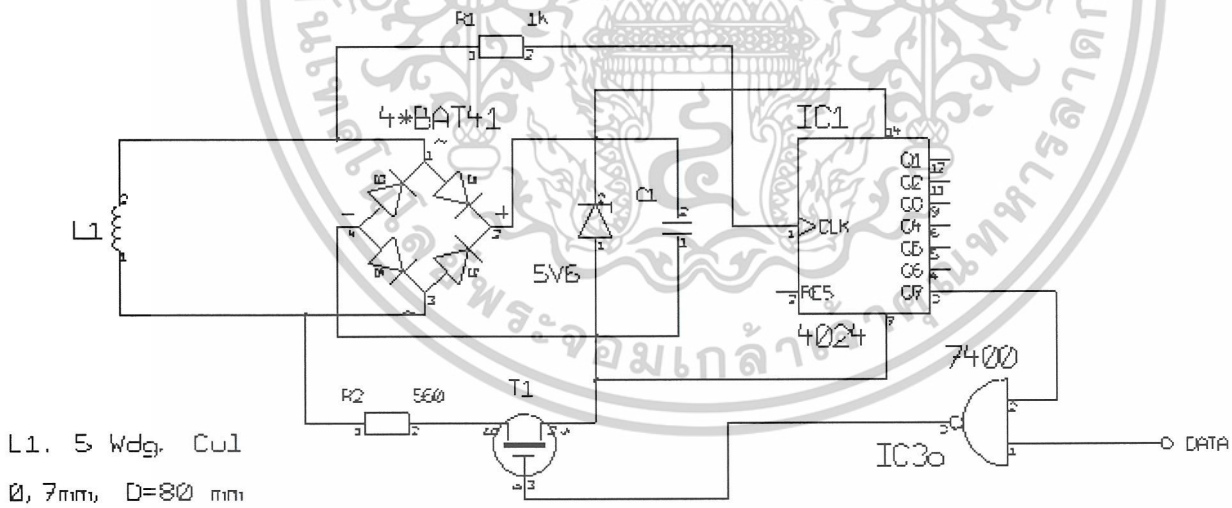
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

Data transfer tag – reader

การส่งข้อมูลจาก Tag ไปยัง Reader คือ Tag มันจะใช้พลังงานของการทำงานของสนามแม่เหล็กเพื่อที่จะส่งกลับ (Return) หรือตอบสนอง (Response) ไปยังเครื่องอ่าน (Reader) วิธีการนี้ถูกเรียกว่า Load Modulation และ Load Modulation นี้จะอาศัยวิธีการของความต้านทานโหลด (Load Resistance) ที่รวมอยู่ในวงจรที่ถูก On – Off "1" หรือ "0" ขึ้นอยู่กับข้อมูลและถูกควบคุมโดยข้อมูลและจาก รูปที่ 2.15 จะเป็นกระบวนการการสร้าง Load Modulation ใน Tag ด้วยสวิตช์ Drain – Source Resistance ของ FET บนตัว Chip



รูปที่ 2.15 การ Generation of Load Modulation



รูปที่ 2.16 วงจร Generation of Load Modulation with Sub Carrier

จากรูปที่ 2.16 เป็นตัวอย่างวงจร Generation of Load Modulation with Sub Carrier สำหรับ Tag ในการใช้ Load Modulation With Sub Carrier วงจรนี้ถูกออกแบบสำหรับ การดำเนินการของความถี่ 13.56 MHz และสร้าง Sub Carrier 212 kHz จากวงจรจะเห็นว่าแรงดันที่ถูกเหนี่ยวที่ขดลวดสายอากาศ L1 โดยการเปลี่ยนแปลงของสนามแม่เหล็กของเครื่องอ่าน (Reader) และถูก Rectified โดยใช้บริดจ์ไดโอด (D1-D4) และทำให้เรียบด้วย C1

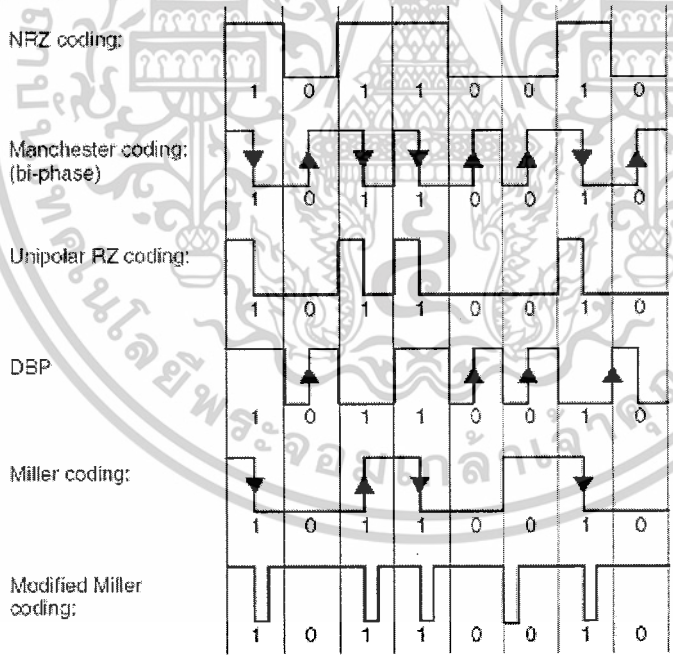
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
104362

และมีวงจรที่เป็น Supply Voltage (ZD 5V6) ป้องกันแรงดันที่ไม่สามารถควบคุมได้เมื่อแรงดันมันเพิ่มขึ้นเมื่อ Tag เข้ามาใกล้เสาอากาศของเครื่องอ่านและในส่วนของแรงดันความถี่สูงของสายอากาศ (13.56 MHz) เดินทางไปที่ความถี่อินพุท (CLK) และถูกป้องโดย R1 และใช้ Tag เป็นส่วนสำคัญสำหรับการสร้างสัญญาณคล็อกภายในหลังจากที่หารด้วย 64 สัญญาณคล็อกของ Sub Carrier คือ 212 kHz และออกมาที่เอาท์พุทของ Q7 สัญญาณคล็อกของ Sub Carrier ถูกควบคุมด้วยการไหลของ Serial Data ที่ Data Input (DATA) ถูกผ่านด้วยสวิตช์ (T1) ถ้ามีสัญญาณลอจิก High ที่ Input (DATA) แล้วสัญญาณคล็อกของ Sub Carrier จะถูกส่งผ่านเข้าไปในสวิตช์ (T1) และ Load – Resistor (R2) แล้วถูกสวิตช์ให้ ON และ OFF ในเวลาเดียวกันกับ Sub Carrier Frequency

2.2.5 การเข้ารหัสข้อมูล (Data Coding)

การรับส่งข้อมูลแบบตรงไปตรงมาจะทำให้ข้อมูลที่ส่งและรับนั้นมีความยาวเกินไปจึงมีการคิดค้นวิธีเข้ารหัสซึ่งการใช้รหัสจะช่วยการรับส่งข้อมูลสิ้นสุดและไม่ถูกรบกวนจากภายนอก การเข้ารหัสมักเป็นที่นิยมอย่างกว้างขวางในการสื่อสารแบบดิจิตอล

ในการสื่อสารแบบดิจิตอลนั้นใช้สัญลักษณ์ 0 กับ 1 แทนข้อมูลโดยความแตกต่างของข้อมูลจะเป็นจะเป็นตัวกำหนดค่า 0 กับ 1 ของมาตรฐานของแต่ละบิดของข้อมูล ข้อมูลจะถูกจัดเรียงแบบแนวนอนหรือแบบเส้นตรง (Line Code) ซึ่งมาตรฐานของ Line Code จะมีหลายมาตรฐานดังนี้



รูปที่ 2.17 การเข้ารหัสข้อมูลในแบบต่างๆ

จากรูปที่ 2.17 เราจะพบว่ามาตรฐานของ Line Code มีอยู่หลายชนิด ความแตกต่างของมันก็คือการแสดงค่า 0 กับ 1 ที่ให้สัญลักษณ์แตกต่างกัน Line Code ที่พบบ่อยและที่นิยมใช้คือ NRZ และ Manchester Coding

NRZ: สัญลักษณ์แทนค่า “1” คือช่วงสัญญาณเป็น High สัญลักษณ์แทนค่า “0” คือช่วงสัญญาณเป็น Low

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Manchester Coding: สัญลักษณ์ “1” จะแทนด้วยช่วงขอบขาลงของสัญญาณ สัญลักษณ์ “0” จะแทนด้วยช่วงขอบขาขึ้นของสัญญาณ

2.2.6 Digital modulation

เนื่องจากหลักการพื้นฐานของ RFID คือการใช้หลักคลื่นแม่เหล็กไฟฟ้าในการสื่อสารรับส่งข้อมูล ดังนั้นก่อนการสื่อสารจะต้องมีกระบวนการผสมข้อมูลที่เข้ารหัสไว้แล้วคลื่นพาหะ (Modulation) แต่เนื่องจากลักษณะข้อมูลเป็นแบบดิจิตอลจึงต้องใช้วิธีผสมแบบดิจิตอลจะแบ่งเป็น 3 แบบ

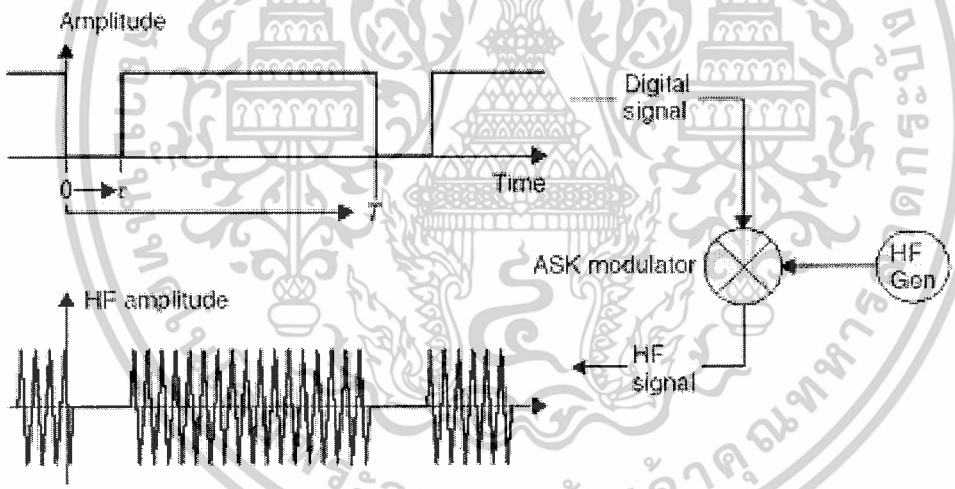
ASK: amplitude shift keying

PSK: phase shift keying

FSK: frequency shift keying

- Amplitude Shift Key

เป็นการผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงของยอดคลื่นพาหะ (Amplitude) เป็นตัวแสดงลักษณะข้อมูลโดยความถี่ของคลื่นพาหะไม่เปลี่ยนดังรูป

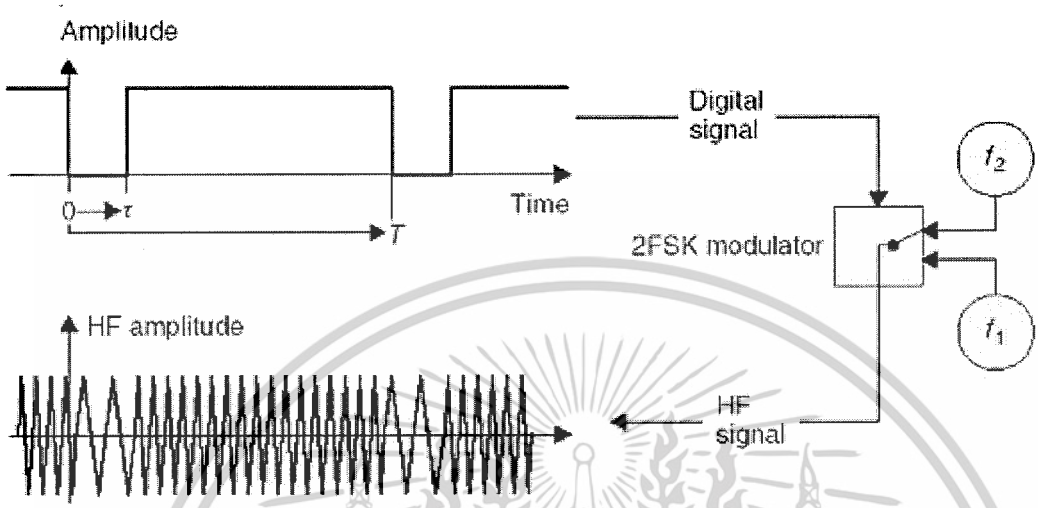


รูปที่ 2.18 การผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงของยอดคลื่นพาหะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Frequency Shift Keying (FSK)

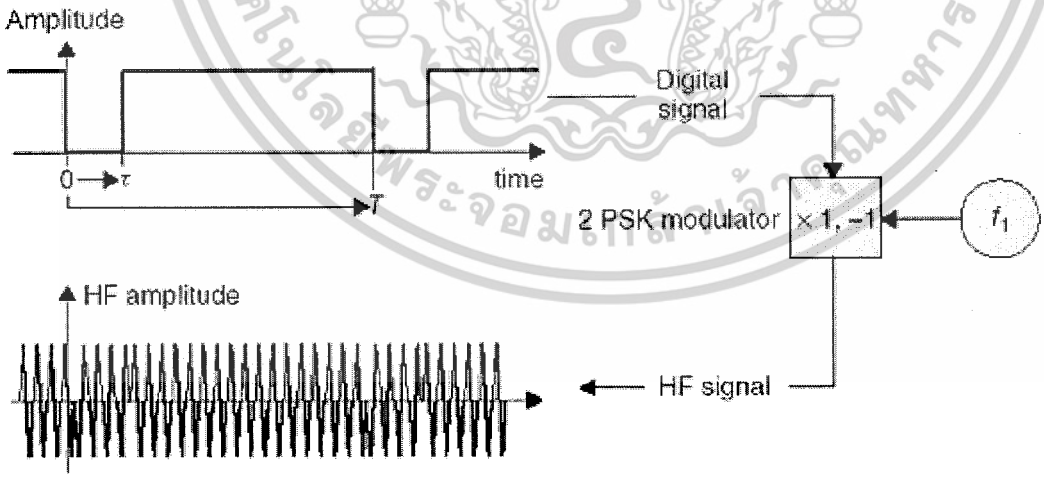
วิธีนี้อาศัยการเปลี่ยนแปลงความถี่ของคลื่นพาหะระหว่าง 2 ความถี่ขึ้นอยู่กับสถานะของข้อมูล 0 กับ 1 โดยความสูงของยอดคลื่น (Amplitude) ไม่เปลี่ยนแปลง



รูปที่ 2.19 การผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงความถี่

- Phase Shift Keying (PSK)

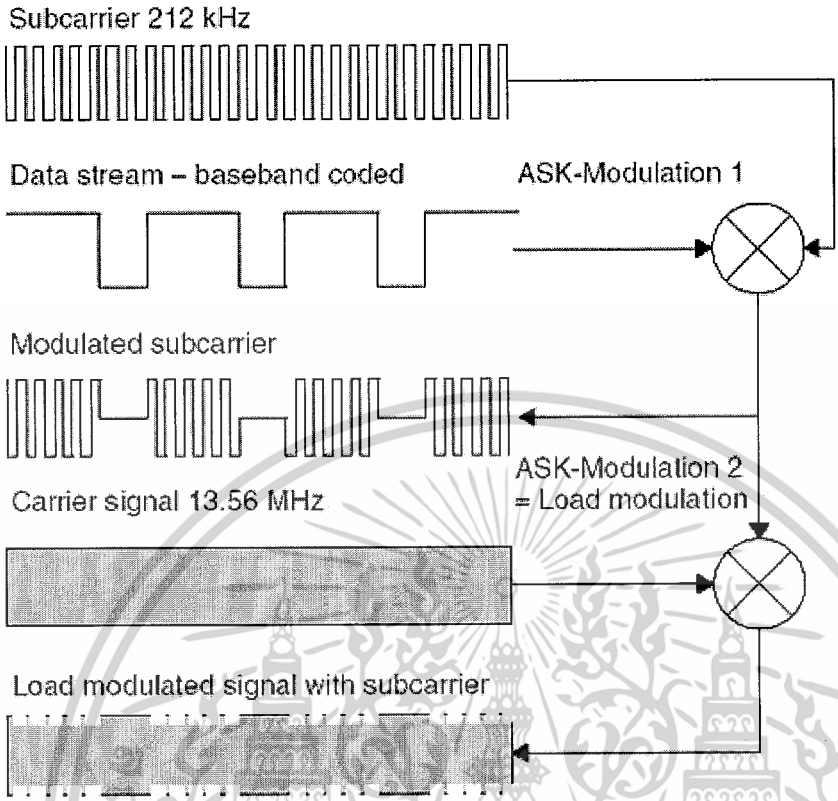
วิธีนี้จะใช้หลักการเปลี่ยนแปลงของเฟสของรูปคลื่นเป็นตรงกันข้าม (0 องศากับ 180 องศา) เมื่อมีการเปลี่ยนแปลงสถานะข้อมูลดังรู)ที่ 2.20



รูปที่ 2.20 การผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ Modulation แบบ ASK Sub carrier

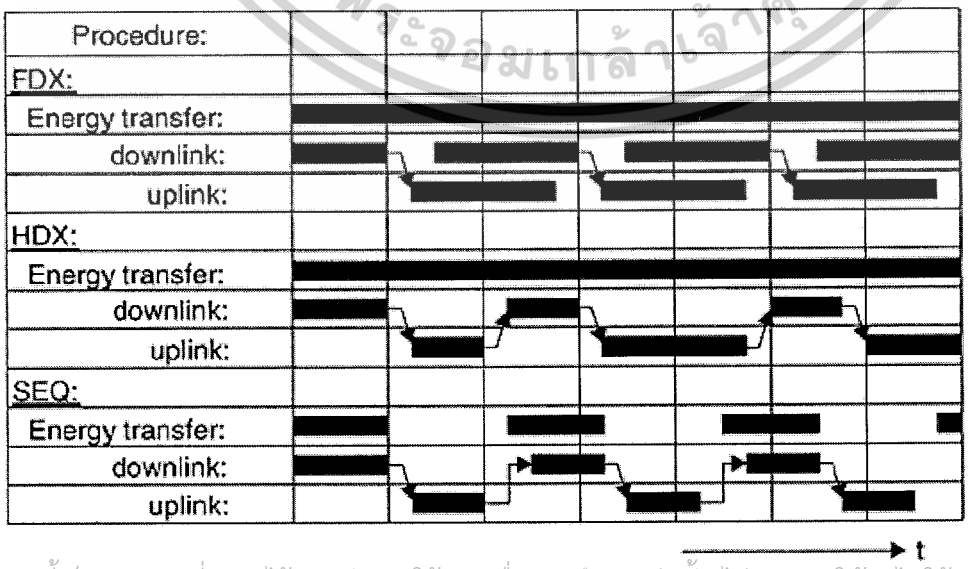


รูปที่ 2.21 การ Modulation แบบ ASK Sub carrier

2.2.7 การสื่อสารระหว่างตัวอ่านและ Tag

- FDX (Full duplex)
- HDX (Half duplex)
- SEQ (Sequential)

ตารางที่ 2.1 กระบวนการส่งข้อมูลของ RFID

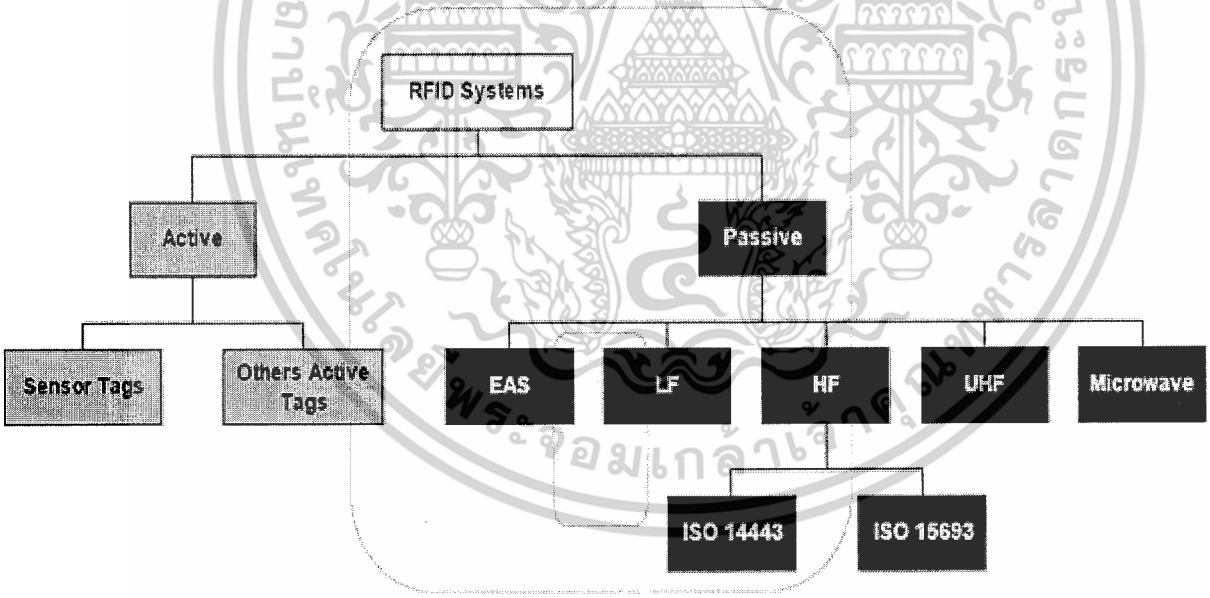


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ Tag จะถูกพิจารณาในส่วนที่ใช้อิเล็กทรอนิกส์ไมโครชิพแสดงเป็น Data carrying และมันจะมีความจุของข้อมูลและสะสมไว้ประมาณ 2-3 กิโลไบต์ขึ้นไป และการอ่านและเขียน Data Carrying ต้องมีความเป็นไปได้ที่จะส่งข้อมูล ระหว่าง Tag กับเครื่องอ่าน การส่งผ่านข้อมูลนี้จะนำเอากระบวนการอย่างใดอย่างหนึ่งมาพิจารณาคือกระบวนการ Full – Half Duplex ที่จะถูกพิจารณาในส่วนนี้และจะพิจารณาในส่วนของ Sequential ด้วย

ในส่วนของกระบวนการของ Half Duplex (HDX) ข้อมูลที่จะส่งจาก Tag ไปยังเครื่องอ่านหรือเครื่องอ่านไปยังตัว Tag ที่ความถี่ที่ต่ำกว่า 30 MHz ในส่วนนี้มักจะถูกใช้กับกระบวนการ Load Modulation ทั้งมีและไม่มี Sub Carrier รวมถึงวงจรที่ค่อนข้างง่าย รวมทั้งเทคโนโลยีของ เรดาร์ ที่ใช้ความถี่มากกว่า 100 MHz กระบวนการ Load Modulation และสัญญาณที่ถูก Modulate กระบวนการ การสะท้อน ที่มีอิทธิพลโดยตรงของสนามแม่เหล็กหรือสนามแม่เหล็กไฟฟ้าจะถูกสร้างโดยเครื่องอ่านและนั่นคือเป็นกระบวนการ ฮาโมนิค

ในส่วนของ Full Duplex (FDX) ข้อมูลที่จะส่งจาก Tag ไปยังเครื่องอ่านและในเวลาเดียวกันก็เป็นการส่งจากเครื่องอ่านไปยัง Tag และนี่เป็นการรวมกระบวนการที่ข้อมูลจะถูกส่งผ่านจาก Tag ที่เป็นความถี่ส่วนน้อยของเครื่องอ่านเป็นซับฮาโมนิค ความถี่ไม่ขึ้นกับใคร อย่างไรก็ตามทั้งสองกระบวนการจะมีสิ่งที่เหมือนกันตรงที่ว่าการส่งพลังงานจากเครื่องอ่านไปยัง Tag จะส่งไปแบบต่อเนื่องทิศทางการไหลของข้อมูลเป็นอิสระและในระบบ Sequential (SEQ) การส่งพลังงานจาก Tag ไปยังเครื่องอ่านมันจะถูกจำกัดด้วยเวลาเท่านั้น (การดำเนินการของ พัลส์และระบบของ พัลส์) ข้อมูลที่จะส่งจาก Tag ไปยังเครื่องอ่าน เกิดขึ้นระหว่างแหล่งจ่ายกำลังงานกับ Tag



รูปที่ 2.22 ระบบ RFID

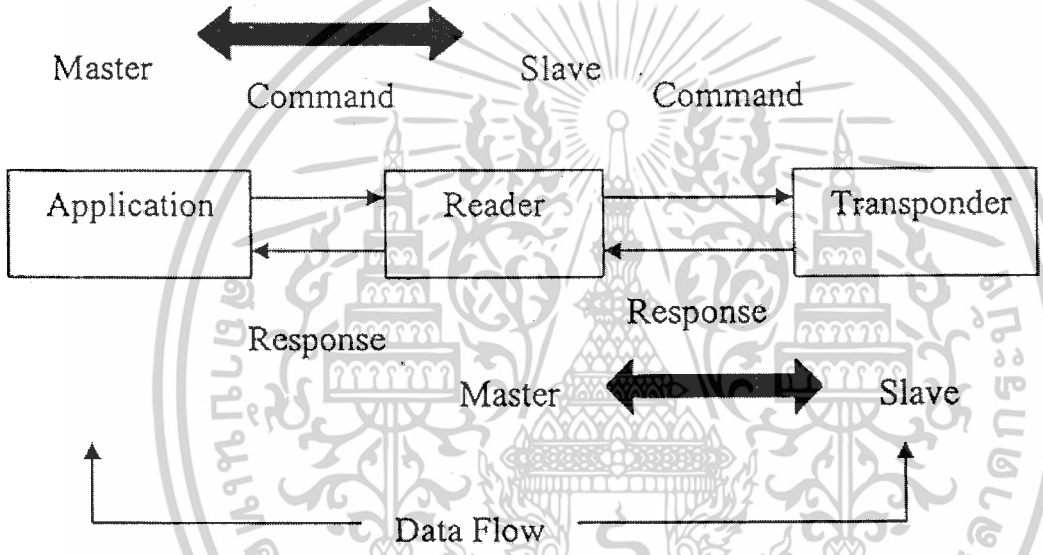
2.3 รูปแบบการทำงาน (หลักการ) ของ Reader

2.3.1 รูปแบบของ Reader

การประยุกต์ใช้งานของซอฟแวร์ที่ถูกกำหนดกับการอ่านข้อมูลจาก Tag หรือเขียนข้อมูลลงไปใน Tag ซึ่งกระบวนการนี้จะเรียกเครื่องอ่าน (Reader) เป็นตัวเชื่อมต่อ (Interface) จากจุดของการ Application Software และกระบวนการข้อมูลควรจะเข้าใจง่ายและเป็นไปได้การเขียนและอ่านจะรวมกับข้อมูล Carrier ที่ถูกกระทำบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นฐานของ Master-Slave ตามรูป 2.23 หมายถึงว่าเครื่องอ่านทั้งหมดและ Tag จะถูกกระตุ้นและถูกรวมเข้าไปใน Application Software และ Application Software จะแสดงเป็น Master ในขณะที่เครื่องอ่าน (Reader) จะเป็น Slave และถูกกระตุ้นเมื่อมีคำสั่งอ่านและเขียนข้อมูลที่รับจาก Application Software เท่านั้น การดำเนินคำสั่งจาก Application Software เริ่มแรกเครื่องอ่านเข้าไปสื่อสารกับ Tag ก่อนและตอนนี้เครื่องอ่านก็จะมีสถานะเป็น Master ที่มีความสัมพันธ์กับ Tag และ Tag ก็จะตอนสนอง (Response) กับคำสั่งจากเครื่องอ่านเท่านั้นตัวอย่างของคำสั่งอ่านจาก Application Software กับเครื่องสามารถรวมเข้ากับการติดต่อสื่อสารกันระหว่างเครื่องอ่านและ Tag ในตัวอย่างตารางที่ 2.2 ตอนแรกคำสั่งอ่านก็จะคำสั่งนี้ไปกระตุ้นกับ Tag และตามด้วยการดำเนินการของขั้นตอนการส่งข้อมูลที่ถูกร้องขอ (Request)



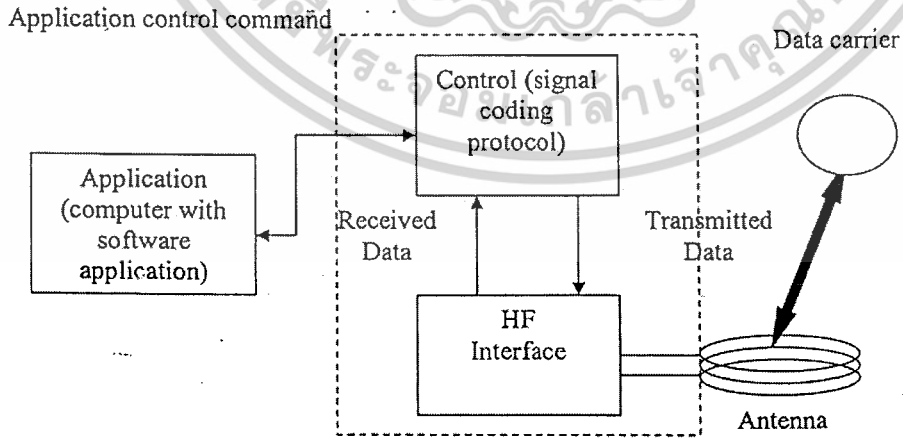
รูปที่ 2.23 Master – Slave ระหว่าง Application Software, เครื่องอ่านและ Tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ตัวอย่างการดำเนินการอ่านข้อมูลด้วย Application Software เครื่องอ่านและ Tag

Application Reader	**Reader Tag**	**Comment**
Block read_Address[00]		Read Tag memory [address[
	Request	Tag operate with serial number Initiate authentication
	ART_SNR[4712] GET Random	Tag operate with serial number Initiate authentication
	Random[081514]	
	SEND_Token1	
	GET_Token2	Authentication successfully completed
	Read_@[00]	Read command [address]
Data[987654321]	Data[987654321]	Data from Tag Data to application

และการเชื่อมโยง (Inductive-electromagnetic) ในขั้นตอนการสื่อสาร (FDX, HDX, SEQ) ขั้นตอนการส่งข้อมูลจาก Tag ไปยังเครื่องอ่าน Reader (Load modulation, backscatter, sub harmonic) ช่วงความถี่และเครื่องอ่านทุกๆ แบบก็มีการกระทำแบบนี้เป็นพื้นฐานในกาออกแบบเครื่องอ่านในระบบทุกระบบสามารถยุบลงหรือลดลงเหลือ 2 ส่วนที่เป็นส่วนสำคัญมีสอง Block คือ ส่วนควบคุมระบบและ HF Interface จะประกอบด้วยในส่วนของ Transmitter และ Receiver ดังแสดงในรูปที่ 2.27



รูปที่ 2.24 Block Diagram ของเครื่องอ่านที่ประกอบด้วยส่วนควบคุมและ HF Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 HF Interface of Reader

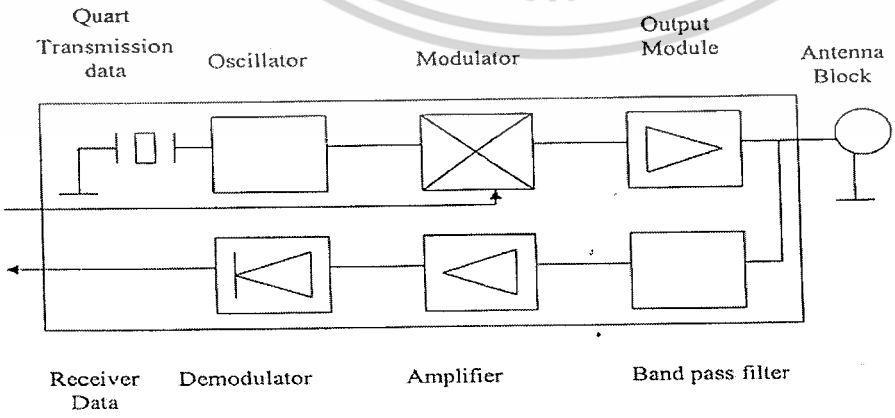
HF Interface ของเครื่องอ่านจะกระทำตามฟังก์ชั่นตามนี้

- ทำการสร้างกำลังงานความถี่สูงที่จะส่งไปและไปกระตุ้น Tag เพื่อเป็นกำลังงานและเป็นแหล่งจ่ายให้ Tag
- ทำการ Modulation ในส่วนของสัญญาณข้อมูลที่จะส่งข้อมูลไปให้ Tag
- ทำการรับและ Demodulation ของสัญญาณ HF ที่จะถูกส่งมาโดย Tag

HF Interface จะแยกสัญญาณเป็น 2 ส่วนโดยการพิจารณาของสองทิศทางการส่งและรับข้อมูลจาก Tag ตามรูปที่ 2.25 ข้อมูลจะถูกส่งผ่านไปยัง Tag และเดินทางผ่านในส่วน Transmission Data ในทางกลับกันข้อมูลที่ถูกรับจาก Tag จะเป็นขบวนการใน Receiver Data

2.3.3 Inductively Coupled System, FDX / HDX

สัญญาณที่ต้องการจากดำเนินการของความถี่ 135 KHz หรือ 13.56 MHz จะถูกสร้างใน Transmitter arm ด้วยความถี่ Quartz Oscillator และเพื่อป้องกันสัญญาณ Noise Radio กับสัญญาณที่รับจาก Tag Oscillator จะเป็นส่วนที่ร้องขอการพิจารณาเฟสที่แน่นอนสัญญาณ Oscillator ที่เข้าไปในส่วนของ Modulation จะถูกควบคุมด้วยสัญญาณ Base Band ของระบบสัญญาณการ Coding และ Base Band Signal นี้จะเป็นตัวถอดรหัสสัญญาณแรงดันใน Binary Data ที่ถูกแสดงเป็น Serial Code (Manchester, Miller, NRZ) ขึ้นอยู่กับชนิดของการ Modulator และ ASK หรือ PSK Modulator จะถูกกระทำบนสัญญาณ Oscillator และสัญญาณที่ถูก Modulated แล้วก็ถูกส่งไปโดย Power output Module และจะถูก Decoupled ไปยัง Antenna Block และในส่วนของ Receiver arm เริ่มต้นที่ Antenna Block และองค์ประกอบแรก Band Pass Filter หรือ Notch Filter ในระบบ FDX/HDX Filter นี้มันก็จะทำการบล็อกสัญญาณที่ส่งมาจาก Out put Module และตอบสนองสัญญาณจาก Tag จะเห็นว่าการส่งและการรับความถี่จะเป็นส่วนที่แยกต่างหากและในระบบ Load Modulation ในการใช้ Sub Carrier ตัว Filter ควรจะพิจารณาให้ดีเพราะว่าในส่วนนี้สัญญาณที่ถูกส่งและถูกรับจะถูกแยกคนละส่วนด้วย Sub Carrier Frequency เท่านั้น ชนิด Sub Carrier ในระบบ 13.56 MHz, 847 kHz หรือ 212 kHz



รูปที่ 2.25 Block Diagram ของ HF Interface สำหรับ Inductive Couple

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

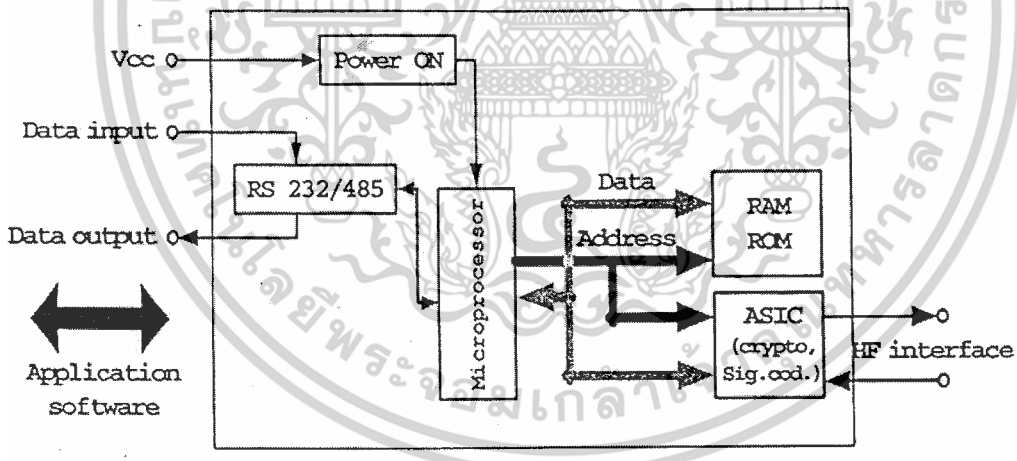
2.3.4 Control Unit

หน่วยควบคุมเครื่องอ่าน Control Unit) ตามรูปที่ 2.26 จะถูกกระทำตามฟังก์ชั้นดังต่อไปนี้

- การสื่อสารกับ Application Software และดำเนินการกับคำสั่งจาก Application Software
- ควบคุมการติดต่อสื่อสารกับ Tag (Master-Slave)
- สัญญาณ Coding และ Decoding
- ดำเนินการตามกระบวนการ Anti-Collision
- การเข้ารหัสและถอดรหัสของข้อมูลที่ถูกส่งผ่านระหว่างเครื่องอ่าน (Reader) และ Tag
- ตรวจสอบความถูกต้องของข้อมูลที่สื่อสารกันระหว่าง Tag และเครื่องอ่าน (Reader)

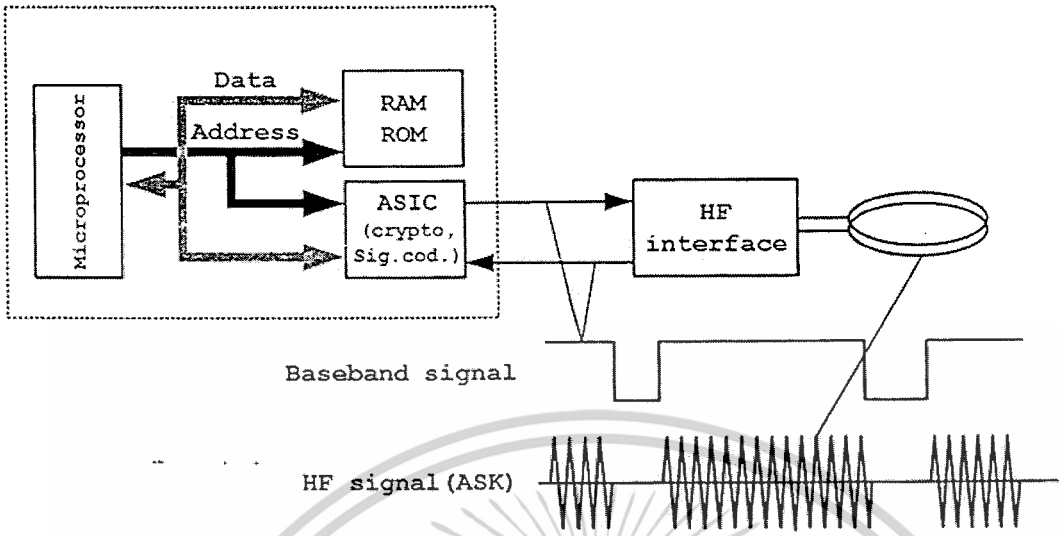
Control Unit ปกติมักจะขึ้นอยู่กับการกระทำของฟังก์ชั่นของไมโครโปรเซสเซอร์และกระบวนการนี้เป็นกระบวนการเข้ารหัสที่ส่งผ่านระหว่าง Tag และเครื่องอ่าน (Reader) ดังนั้นสัญญาณ Coding มักจะถูกกระทำเข้าไปใน ASI Module กับขบวนการคำนวณของไมโครโปรเซสเซอร์สำหรับเหตุผลของ ASI ที่ถูกส่งเข้าไปในไมโครโปรเซสเซอร์ข้อมูลที่ถูกแลกเปลี่ยนระหว่าง Application Software และส่วนหน่วยประมวลผลของเครื่องอ่าน (Reader) ที่ถูกกระทำโดย RS232 หรือ RS485 Interface ซึ่งเป็นพอร์ตที่ใช้กันอยู่ทั่วไป

การ Interface กันระหว่าง HF Interface และ Control Unit จะแสดงสถานะ HF Interface เป็นเลข Binary ในระบบ ASK Modulated ลอจิก "1" ที่มอดูเลตอินพุทของ HF Interface จะแสดงสถานะของ HF Interface เป็น 'HF Signal On' และลอจิก '0' จะแสดงสถานะเป็น 'HF Signal Off'



รูปที่ 2.26 Block Diagram ของ Control Unit ของ Reader และสื่อสารกับ Application Software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



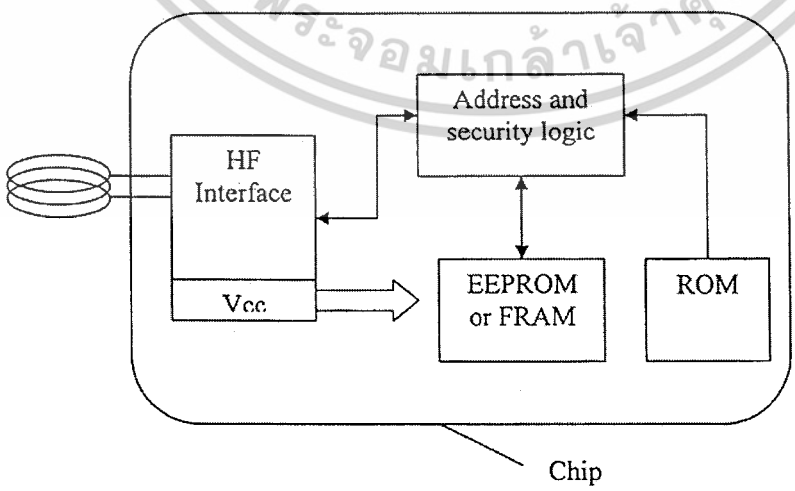
รูปที่ 2.27 สัญญาณ Coding และ Decoding ถูกกระทำโดย Control Unit ของ Reader

2.3.5 สถาปัตยกรรมของ Electronic Data Carrier

ก่อนที่เราจะพิจารณาฟังก์ชั่นของ Data Carrier ที่ถูกใช้ในระบบ RFID อันดับแรกเราต้องรู้ความแตกต่างระหว่างสองฟังก์ชั่น คือ มี Electronic Data Carrier ขึ้นอยู่กับวงจรที่ถูกอินทริเกรต (Microchip) และ Data Carrier คือข้อมูลที่ใช้สำหรับบันทึกข้อมูล

Tag with Function Memory

Tag กับฟังก์ชั่นหน่วยความจำจาก Tag ที่อ่านอย่างเดียวจนถึง Tag ที่สามารถอ่านและเขียนได้จากฟังก์ชั่นต่างๆ ตามรูปที่ 2.28 ที่ Tag กับฟังก์ชั่นหน่วยความจำจะบรรจุ RAM, ROM, EEPROM หรือ FRAM และ HF Interface ที่ใช้เป็นแหล่งจ่ายและยอมให้สื่อสารกับเครื่องอ่าน (Reader)



รูปที่ 2.28 Block Diagram ของ RFID Data Carrier ของ Memory Function

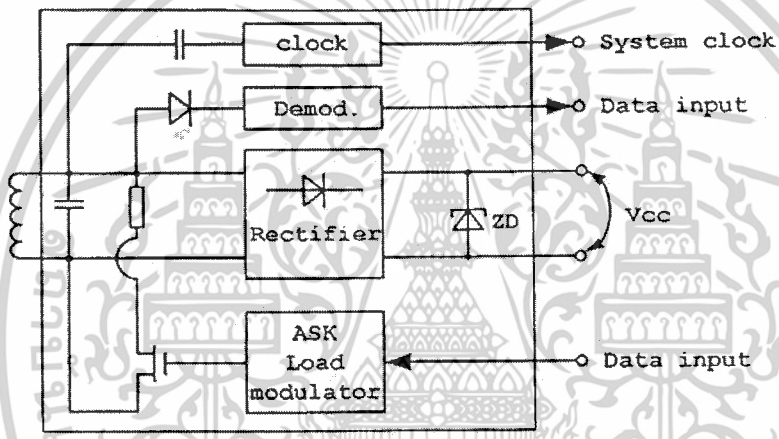
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 HF Interface of Tag

HF Interface เป็นรูปแบบของการ Interface ระหว่าง สัญญาณ Analog ที่ส่งในรูปแบบความถี่สูงจากเครื่องอ่าน (Reader) ไปยัง Tag และวงจร Digital ของ Tag HF Interface ก็กระทำตามฟังก์ชั่นทั่วไปของ โมเด็มเหมือนโมเด็ม (Modulator – Demodulator) ที่ถูกใช้ในการส่งข้อมูล Analog ในสายโทรศัพท์

สัญญาณ HF ที่ถูกมอดูเลตจากเครื่องอ่าน (Reader) จะถูกสร้างใหม่ใน HF Interface โดยการ Demodulation กับการสร้างสัญญาณข้อมูล Digital ข้อมูลจะผ่านไปสำหรับกระทำการใหม่อีกครั้งใน Address Security Logic การสร้าง Clock – Pulse ระบบวงจรจะสร้าง Clock สำหรับ Data Carrier จากความถี่ Carrier ของ HF field

HF Interface จะถูกรวมเข้าไปใน Load Modulator และถูกควบคุมด้วยข้อมูล Digital ที่ถูกส่งออกไปกับข้อมูลที่ส่งกลับไปให้เครื่องอ่าน (Reader) ตามรูปที่ 2.28



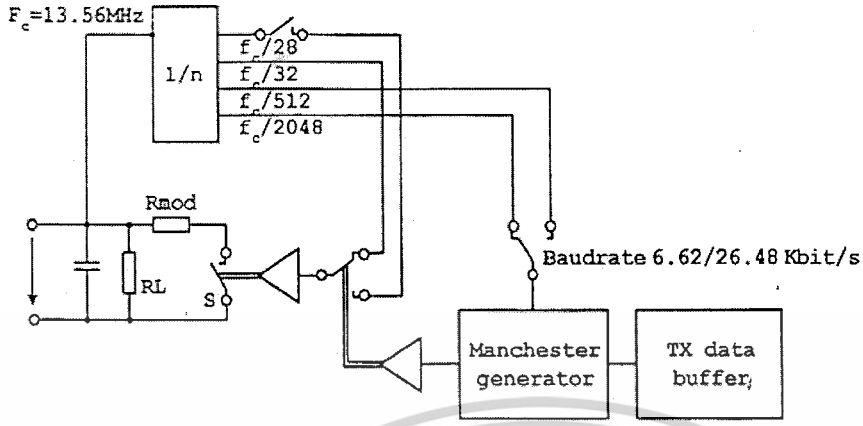
รูปที่ 2.29 Block Diagram ของ HF Interface ของ Inductive Coupled ของ Tag with Load Modulation

Example Circuit – load Modulation with Sub Carrier

วงจรพื้นฐานของ Load Modulation แสดงในรูปที่ 2.29 ส่วนนี้จะเป็นการสร้าง Load Modulation โดยการใช้ ASK หรือ FSK Modulated Sub Carrier ความถี่ของ Sub Carrier และ baud rate จะถูกพิจารณาตามมาตรฐานของ ISO 15693 (Vicinity Coupling Smart Cards) คุณสมบัติความถี่ในมาตรฐานสำหรับ Sub Carrier และ baud rate ที่ได้มาจาก Binary ในส่วนของสัญญาณอินพุต 13.56 MHz ตามตารางที่ 2.2

ข้อมูลจะถูกส่งไปและส่งผ่านไปยัง Manchester Generator การพิจารณานี้ baud rate ของสัญญาณ Base Band Signal ที่จะถูกเปลี่ยนระหว่างค่าสองค่า Manchester Coded ของ Base Band Signal ตอนนี้จะถูกสวิตช์ระหว่าง ความถี่ Carrier สองค่าคือ f1 และ f2 ในการใช้ '1' และ '0' ตามระดับของสัญญาณในส่วนของการสร้าง FSK ที่ถูกมอดูเลตของสัญญาณ Sub Carrier ถ้า Clock ของสัญญาณ f2 ถูก Interface ผลของในส่วนนี้สัญญาณ Sub Carrier ที่ถูกมอดูเลตใน ASK หมายถึงว่าจะมีการสวิตช์ระหว่าง ASK และ FSK Modulation สัญญาณ Sub Carrier ที่ถูกมอดูเลตและตอนนี้ถูกส่งผ่านไปที่สวิตช์ S ดังนั้น Modulation Resistor ของ Load Modulator สามารถที่จะสวิตช์ 'On' และ 'Off' ในเวลาเดียวกันกับความถี่ Sub Carrier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 การสร้าง Load Modulation กับ Sub carrier Modulated

ตารางที่ 2.3 ความถี่ Clock ที่ต้องการใน HF Interface ถูกสร้างโดย Binary ของสัญญาณ Carrier 13.56 MHz

Splitter N	**Frequency**	**Use**
1/28	485 kHz	Q2 of the FSK Sub carrier
1/32	423 kHz	Q1 of the FSK Sub carrier, plus ASK Sub carrier
1/512	26.48 kHz	Bit clock Signal For high baud rate
1/2048	6.62 kHz	Bit clock Signal For high baud rate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ภาษา SQL

ภาษา SQL ย่อมาจาก Structured Query Language เป็นภาษาที่ใช้สำหรับการจัดการและการดึงข้อมูลจากฐานข้อมูล ซึ่งฐานข้อมูลที่จะใช้ภาษา SQL ได้ต้องเป็นฐานข้อมูลเชิงสัมพันธ์ ซึ่งมีมุมมองของข้อมูลในรูปแบบตารางที่สัมพันธ์กัน ซึ่งในการดึงข้อมูลเราจะสั่งงานคำสั่งในภาษาSQL ไปยัง DBMS เพื่อให้มันทำหน้าที่ดึงข้อมูลที่ต้องการแล้วส่งกลับมาซึ่งผลลัพธ์ที่ได้เราเรียกว่า การคิวรีฐานข้อมูล Query Database

SQL ไม่เพียงแต่เป็นภาษาที่ใช้สำหรับการคิวรีเท่านั้น แต่ยังนำมาใช้ควบคุมการทำงานของ DBMS เช่น การใช้คำสั่ง SQL เพื่อสร้างตารางในฐานข้อมูลหรือแม้แต่การอนุญาตให้ผู้ใช้แต่ละคนมีสิทธิในการใช้ฐานข้อมูล เป็นต้น

2.4.1 ลักษณะเด่นของภาษาSQL

SQL ก็คือ การใช้รูปแบบการสร้างประโยคเหมือนภาษาอังกฤษซึ่งมีประโยคคำสั่งคล้ายกับภาษามนุษย์มากกว่าภาษาที่สำหรับเขียนโปรแกรมภาษาอื่นๆ ทำให้ภาษาSQL เข้าใจได้ง่าย และง่ายต่อการเรียนรู้ การเขียนโค้ดและการดูแลบำรุงรักษา ซึ่งข้อดีเหล่านี้ทำให้เหล่าบริษัทผู้ผลิตฐานข้อมูลนิยมใช้ภาษา SQL เป็นภาษาหลักในการจัดการข้อมูล ซึ่งภาษาSQL ได้รับรองโดยสถาบันมาตรฐานแห่งชาติอเมริกาหรือ ANSI

2.4.2 การแบ่งกลุ่ม SQL

ภาษาSQL แบ่งได้ทั้งหมด 3 กลุ่ม คือ

1. Data Manipulation Language DML เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับการเข้าถึงข้อมูลและการแก้ไขข้อมูล เช่น SELECT ใช้เพื่อคิวรีข้อมูล INSERT ใช้เพื่อค้นหาข้อมูล DELETE ใช้เพื่อลบข้อมูล และ UPDATE ใช้เพื่ออัพเดตฐานข้อมูล
2. Data Definition Language DDL เป็นกลุ้มคำสั่งในภาษา SQLที่ใช้กำหนดออบเจ็กต์ฐานข้อมูล เช่น CREATE ใช้เพื่อเพิ่มตารางลงในฐานข้อมูล DROP ใช้เพื่อลบตารางออก และ ALTER ใช้เพื่อเปลี่ยนแปลงโครงสร้างในฐานข้อมูล
3. Data Control Language DCL เป็นกลุ่มคำสั่งในภาษา SQL ที่ควบคุมความปลอดภัยข้อมูล เช่น GRANT ใช้เพื่อสิทธิในการใช้ฐานข้อมูล REVOKE ใช้เพื่อยกเลิกสิทธิในการใช้ฐานข้อมูล

2.4.3 การคิวรีอย่างง่าย

1.การคิวรีโดยคำสั่ง SELECT คำสั่งSELECT ใช้เพื่อดึงข้อมูลจากฐานข้อมูลพร้อมกับแสดงผลบนหน้าจอ รูปแบบคำสั่งมีดังนี้

```
SELECT [DISTINCT] <columns>
 FROM <tables>
[WHERE <condition>]
[GROUP BY <columns>][HAVING<condition>]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[ORDER BY <columns>[ASC | DESC]]

***หมายเหตุ [ ] เป็นตัวเลือกที่จะมีหรือไม่มีก็ได้

< > บังคับต้องมีข้อมูล

ตัวพิมพ์ใหญ่ หมายถึงคำสงวน

| หมายถึงเลือกตัวใดตัวหนึ่ง

SELECT จะดึงข้อมูลจากฐานข้อมูลจากคอลัมน์ต่างๆ ในฐานข้อมูลหรือคอลัมน์อื่นๆ ที่อาจจะเกิดจากการคำนวณโดย SQL เดียวหรือหลายๆคอลัมน์แล้วส่งค่ากลับในรูปแบบผลลัพธ์การคิวรี ถ้ามีหลายๆคิวรีจะใช้เครื่องหมายจุลภาคคั่นเอาไว้

FROM จะตามด้วยตารางต่างๆ ถ้ามีหลายๆตาราง จะใช้เครื่องหมายจุลภาคคั่นไว้ คำสั่งSELECT จะใช้คู่กับคำสั่งFROM เสมอ และเป็นข้อกำหนดตามมาตรฐาน ANSI ส่วนประโยคที่เหลือทั้งหมดจะเป็นทางเลือกในการคิวรี ซึ่งอาจจะมีหรือไม่มีก็ได้

WHERE ใช้เพื่อเลือกเฉพาะเร็คคอร์ดที่มีข้อมูลอยู่เท่านั้น ซึ่งจะเงื่อนไขค้นหาเพื่อระบุข้อมูลที่ต้องการ นอกจากนี้แล้วเราจะใช้เงื่อนไขการค้นหาแบบคิวรีย่อย

GROUP BY เป็นการคิวรีสรุปแทนที่จะให้ผลลัพธ์การคิวรีหนึ่งเร็คคอร์ดสำหรับแต่ละเร็คคอร์ดในฐานข้อมูล การคิวรีสรุปจะทำเร็คคอร์ดที่เหมือนกันให้เป็นกลุ่ม แล้วสร้างผลลัพธ์การคิวรีหนึ่งเร็คคอร์ดให้กับแต่ละกลุ่ม การคิวรีสรุปจะใช้ฟังก์ชัน Aggregate เป็นหลัก

HAVING ใช้เพื่อรวมผลลัพธ์การคิวรีให้เป็นกรุ๊ปหรือกลุ่มที่สร้างโดยประโยค GROUP BY ประโยค HAVING จะเหมือนกับประโยคWHERE ตรงที่ใช้เงื่อนไขการค้นหาเพื่อระบุกลุ่มท่ต้องการ

ORDER BY ใช้เรียงผลลัพธ์การคิวรีจากน้อยไปหามาก และจากมากไปหาน้อย แล้วแต่ขึ้นอยู่กับว่าข้อมูลที่มีอยู่ในหนึ่งคอลัมน์หรือหลายคอลัมน์

คอลัมน์ที่เกิดจากการคำนวณ (Calculated Columns)

นอกจากคอลัมน์ที่มาจากตารางโดยตรงแล้ว การคิวรี SQL ยังสามารถรวมคอลัมน์ที่เกิดจากการคำนวณได้ ซึ่งค่าที่นำมาใช้ในการคำนวณจะเอาจากค่าที่เก็บไว้ในฐานข้อมูลในการคำนวณนี้จะเกี่ยวข้องกับการบวก การลบ การคูณ การหาร นอกจากนี้เราสามารถสร้างรูปแบบการคำนวณที่ซับซ้อนได้ด้วยการใช้วงเล็บ ซึ่งคอลัมน์ที่เป็นชนิดอักขระหรือสตริงมาทำการบวก ลบ คูณ หาร SQL ก็จะรายงานความผิดพลาดทันที

2.4.5 การเลือกทุกคอลัมน์

บางครั้งเพื่อความสะดวกในการแสดงในการแสดงข้อมูลของคอลัมน์ทั้งหมดของตารางหนึ่งในฐานข้อมูลใหม่เนื่องจากเราต้องการทราบข้อมูลที่เก็บไว้ในฐานข้อมูล แต่ถ้าเราใช้คำสั่งของSQL เราทราบเฉพาะโครงสร้างของตารางแต่เราจะไม่เป็นข้อมูล เราใช้เครื่องหมายดอกจัน (*) ตามหลังคำสั่งSELECTซึ่งเครื่องหมายดอกจันจะแทนความหมายเป็นการเลือกทุกคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.6 เร็คคอร์ดที่ซ้ำกัน

ถ้าเราทำการคิวรีที่มีคีย์หลักของตารางนั้นผลลัพธ์ที่ได้ทุกเร็คคอร์ดจะไม่มีการซ้ำกันเลยหรือเรียกว่า Unique เพราะว่าคีย์หลักจะมีค่าที่เหมือนกันในแต่ละเร็คคอร์ด แต่ถ้าคีย์หลักไม่ได้รวมอยู่ในผลลัพธ์การคิวรี เร็คคอร์ดก็จะมีเร็คคอร์ดที่ซ้ำกันอยู่ เราสามารถกำจัดเร็คคอร์ดที่ซ้ำกันออกด้วยการใช้คีย์เวิร์ดคำว่า DISTINCT ต่อจากคำสั่ง SELECT คำว่า DISTINCT ในที่นี้หมายถึงไม่มีสิ่งที่ซ้ำกัน

การกลั่นกรองเร็คคอร์ดที่ต้องการด้วยประโยค WHERE

โดยปกติแล้วเราต้องการเลือกบางเร็คคอร์ดในตารางมาแสดงผล เราจะใช้ประโยคWHERE เพื่อกำหนดเงื่อนไขในการกลั่นกรองผลลัพธ์ที่จะได้ ในประโยคWHERE จะต้องมีเงื่อนไขการค้นหา ซึ่งจะระบุเร็คคอร์ดที่เราต้องการดึงมา ซึ่งสรุปได้ 5 รูปแบบดังนี้

- การเปรียบเทียบค่าในนิพจน์เป็นการเปรียบเทียบค่าของข้อมูลในคอลัมน์ซึ่งจะเป็นชื่อคอลัมน์กับข้อมูลเฉพาะที่เป็นค่าคงที่
- การทดสอบค่าในนิพจน์กับขอบเขตที่กำหนด เป็นการทดสอบค่าต้องอยู่ในระหว่างค่า2ค่า
- การทดสอบค่าในนิพจน์กับสมาชิกเซตเป็นการตรวจสอบค่าต้องตรงกับค่าที่อยู่ในกลุ่มสมาชิกของข้อมูล
- การทดสอบสตริงเพื่อจับคู่ในรูปแบบที่กำหนด เป็นการตรวจสอบค่าในคอลัมน์ที่เป็นอักขระหรือสตริงซึ่งจะต้องจับคู่กับรูปแบบที่กำหนดไว้
- การทดสอบค่าที่ไม่ทราบค่า ในคอลัมน์เป็นการตรวจสอบค่าในคอลัมน์มีค่าNULL หรือไม่ทราบค่าการเปรียบเทียบค่าในนิพจน์

รูปแบบนี้เป็นลักษณะที่พบในการคิวรีมากที่สุด ในการทดสอบการเปรียบเทียบแล้วSQL จะคำนวณและเปรียบเทียบระหว่างค่าในชื่อคอลัมน์กับข้อมูลเฉพาะที่เป็นค่าคงที่ภาษาSQLมีถึง 6 วิธีในการเปรียบเทียบรูปแบบดังต่อไปนี้

WHERE<Expression> 1 Operator Expression 2 >

โดยที่ Expression 1 คือนิพจน์แรกหลังประโยค WHERE

Operator คือตัวกระทำเปรียบเทียบ 6 วิธี

Expression 2 คือนิพจน์ที่สองที่อยู่ด้านขวาของประโยค WHERE

Operator 6 วิธีมีดังนี้ เท่ากับ(=) น้อยกว่า(<) มากกว่า(>) น้อยกว่าหรือเท่ากับ(<=) มากกว่าหรือเท่ากับ(>=) ไม่เท่ากับ(<>)

2.4.7 การทดสอบค่าในนิพจน์กับขอบเขตที่กำหนด

ภาษาSQLมีเงื่อนไขการค้นหาอีกรูปแบบหนึ่ง ซึ่งก็คือ การใช้ BETWEEN เพื่อทดสอบขอบเขตของข้อมูล มีรูปแบบดังนี้

WHERE <Expression 1[NOT] BETWEEN Low Expression 2 AND High Expression 2>

โดยที่ Expression 1 คือ นิพจน์แรกหลังประโยค WHERE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low Expression 2 คือ นิพจน์ที่สองด้านล่าง

High Expression 2 คือ นิพจน์ที่สองด้านบน

ประโยค WHERE จะเป็นจริงก็ต่อเมื่อค่าในนิพจน์แรกต้องอยู่ในระหว่างค่าในนิพจน์ที่สองด้านล่างกับด้านบน

2.4.8 การทดสอบค่าในนิพจน์กับสมาชิกเซต

เงื่อนไขการทดสอบจะใช้ IN ซึ่งมีรูปแบบดังนี้

WHERE <Expression [NOT] IN (Constant1,2,…,n)>

โดยที่ Expression คือนิพจน์แรกหลังประโยค WHERE

Constant1, 2… n คือ ค่าคงที่ ถ้าทีค่าคงที่มากกว่าหนึ่งค่า จะใช้เครื่องหมายจุลภาคคั่นไว้

[NOT] เป็นตัวเลือกที่จะมีหรือไม่มีก็ได้

2.4.9 การทดสอบสตริงเพื่อจับคู่ในรูปแบบที่กำหนด

เงื่อนไขการทดสอบนี้จะใช้ LIKE ซึ่งมีรูปแบบดังนี้

WHERE <Column name [NOT] LIKE pattern (ESCAPE)>

โดยที่ Column name คือชื่อคอลัมน์ที่ต้องการจับคู่กับรูปแบบที่กำหนด

Pattern คืออักขระที่รวมอักขระ Wildcard ซึ่งจะมีอักขระ%และ_

ESCAPE ใช้กันอักขระพิเศษออกไป

[NOT] เป็นตัวเลือกที่จะมีหรือไม่มีก็ได้

ประโยค WHERE จะเป็นจริงก็ต่อเมื่อค่าในคอลัมน์แรกจับคู่ได้กับรูปแบบที่กำหนด

2.4.10 การทดสอบค่าที่ไม่ทราบค่า

เงื่อนไขการทดสอบจะใช้ IS NULL ซึ่งมีรูปแบบดังนี้

WHERE Expression IS [NOT]

2.4.11 การใช้หลายๆเงื่อนไขรวมกัน

ประโยค WHERE สามารถมีเงื่อนไขหลายๆแบบรวมกันได้ ได้แก่ AND OR NOT มีรูปแบบเงื่อนไขดังนี้

WHERE [(Condition 1 [AND | OR] Condition 2)] |[NOT Condition] |[Condition]

โดยที่ [(Condition 1 [AND | OR] Condition 2)] คือเงื่อนไขการทดสอบระหว่าง Condition 1 กับ condition 2 และจะมีตัวดำเนินการอย่างใดอย่างหนึ่ง ระหว่าง AND, OR

NOT Condition คือเงื่อนไขซึ่งประโยค WHERE จะเป็นจริงก็ต่อเมื่อผลลัพธ์จาก Condition เป็นเท็จ และจะเป็นเท็จก็ต่อเมื่อผลลัพธ์จาก Condition เป็นจริง

Condition เป็นเงื่อนไขใดๆก็ได้ที่ต้องการทดสอบที่ได้กล่าวมาแล้วในข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.12 การเรียงลำดับผลลัพธ์โดยประโยค ORDER BY

ประโยค ORDER BY ใช้เรียงผลลัพธ์การคิวรีที่ขึ้นอยู่กับข้อมูลในหนึ่งคอลัมน์หรือหลายคอลัมน์ ถ้าเราละไว้ ผลลัพธ์การคิวรีก็จะไม่มีการเรียง รูปแบบคำสั่ง ORDER BY

ORDER BY <column name | column number > [ASC|DESC]

โดยที่ Column name คือชื่อคอลัมน์ที่ต้องการเรียง ถ้ามีมากกว่าหนึ่งคอลัมน์ให้ใช้เครื่องหมาย,คั่นไว้

Column number คือหมายเลขคอลัมน์ที่เรียงตามลำดับหลังคำสั่ง SELECT

ASC คือ เรียงจากมากไปน้อย

DESC คือเรียงจากน้อยไปมากและเป็นค่าดีฟอลต์

2.4.13 สรุปข้อมูลด้วย Aggregate Function

ภาษา SQLมีกลุ่มฟังก์ชันที่เราเรียกว่า Aggregate คอยทำหน้าที่สรุปข้อมูลจากฐานข้อมูลในรูปแบบต่างๆ โดยที่ฟังชันก์เหล่านี้ จะเอาข้อมูลทั้งคอลัมน์มาเป็นอาร์กิวเมนต์ แล้วสร้างรายการข้อมูลอันเดียวซึ่งสรุปจากคอลัมน์ เช่น

SUM คำนวณผลรวมของคำในคอลัมน์ ข้อมูลในคอลัมน์ต้องเป็นตัวเลขเท่านั้น

MIN คำนวณหาค่าต่ำที่สุดในคอลัมน์ ข้อมูลในคอลัมน์จะเป็นตัวเลข สตริง วันที่

MAX คำนวณหาค่ามากที่สุดในคอลัมน์ ข้อมูลในคอลัมน์จะเป็นตัวเลข สตริง วันที่

AVG คำนวณค่าเฉลี่ยของคำในคอลัมน์ ข้อมูลในคอลัมน์ต้องเป็นตัวเลขเท่านั้น

COUNT นับจำนวนข้อมูลในคอลัมน์ หรือ จำนวนแถว ข้อมูลในคอลัมน์นี้เป็นชนิดใดๆก็ได้ และฟังก์ชันนี้จะส่งค่ากลับเป็นข้อมูลชนิดจำนวนเต็มเสมอ

ฟังก์ชันทั้งหมดที่กล่าวมาสามารถเพิ่มคีย์ DISTINCT เพื่อขจัดเร็คคอร์ดที่ซ้ำกันยกเว้นฟังชันก์ MAX, MIN เท่านั้นที่อาจจะใช่หรือไม่ใช่แต่จะไม่มีผลต่อการคำนวณ

2.4.14 การจักกลุ่มข้อมูลด้วย GROUP BY

หลังจากได้ใช้ฟังก์ชัน Aggregate ต่างๆซึ่งจะอยู่ที่ท้ายรายการของผลลัพธ์การคิวรี อย่างไรก็ตามบางครั้งเราจำเป็นต้องมีการแบ่งกลุ่มออกมาเป็นกลุ่มย่อยๆหรือรายการย่อยๆในผลลัพธ์ด้วย เราสามารถใช้ประโยค GROUP BY เพื่อทำการแบ่งออกเป็นรายการย่อยๆ การคิวรีที่รวมประโยค GROUP BY จะเรียกว่า การคิวรีแบบกลุ่ม เพราะว่ามันจะรวมกลุ่มข้อมูลจากคำสั่ง SELECT แล้วสร้างเป็นเร็คคอร์ดสรุปเพียงเร็คคอร์ดเดียวให้กับแต่ละกลุ่ม ส่วนคอลัมน์ต่างๆที่อยู่ในประโยค GROUP BY เรียกการทำกลุ่มคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปประโยคของ GROUP BY มีดังนี้

GROUP BY < Expression>

เมื่อใช้ GROUP BY คอลัมน์ต่างๆในคำสั่ง SELECT จะต้องเป็นค่าเดียวต่อกลุ่มนอกจากนั้นคำสั่ง SELECT จะต้องประกอบด้วยชื่อคอลัมน์ ฟังก์ชัน Aggregate ค่าคงที่และนิพจน์ที่เกิดจากการรวมกันของชื่อคอลัมน์ฟังก์ชัน Aggregate และค่าคงที่เท่านั้น

ชื่อคอลัมน์ทั้งหมดในคำสั่ง SELECT จะต้องปรากฏในประโยค GROUP BY เสมอเว้นเสียแต่ว่าชื่อคอลัมน์ถูกใช้ในฟังก์ชัน Aggregate แต่ก็ไม่เป็นจริงเสมอไป ถ้าชื่อคอลัมน์ในประโยค GROUP BY ไม่ได้ปรากฏอยู่ในประโยค SELECT เมื่อประโยค WHERE ใช้กับGROUP BY ประโยค WHERE ก็จะถูกทำก่อนจากนั้นจึงจะทำ GROUP BY แล้วทำให้กลุ่มซึ่งจะสอดคล้องกับเงื่อนไข

2.4.15 การทำหลายๆกลุ่มคอลัมน์

ภาษา SQL สามารถทำกลุ่มผลลัพธ์การคิวรีที่ขึ้นอยู่กับหลายๆคอลัมน์ได้โดยใช้ฟังก์ชัน COMPUTE ซึ่งคล้ายกับฟังก์ชัน Aggregate ที่ใช้ร่วมกับประโยค GROUP BY และผลลัพธ์การคิวรีจะคล้ายกับการสร้างรายงานสรุป

2.4.16 การจำกัดกลุ่มโดยประโยค HAVING

ประโยค HAVING ถูกออกแบบมาใช้กับประโยค GROUP BY เพื่อจำกัดกลุ่มที่มีอยู่ในผลลัพธ์การคิวรี ถึงแม้ว่าประโยค HAVING และ WHERE จะมีรูปแบบคล้ายกัน แต่ก็มีวัตถุประสงค์การใช้ที่แตกต่างกัน โดยที่ WHERE จะเลือกเร็คคอร์ดที่ตรงกับเงื่อนไขแล้วแสดงผลออกในผลลัพธ์การคิวรี ขณะที่ประโยค HAVING จะกรองข้อมูลเป็นกลุ่มหรือกรุ๊ปแล้วแสดงผลในผลลัพธ์การคิวรี รูปแบบของประโยค HAVING จะคล้ายกับประโยค WHERE ที่ประกอบด้วยคีย์ HAVING แล้วตามด้วยเงื่อนไขการค้นหา ดังนี้

HAVING <condition>

ตามาตรฐานของ ANSI มีข้อกำหนดว่าชื่อคอลัมน์ที่ใช้ในประโยคHAVING จะต้องปรากฏอยู่ในประโยค GROUP BY ด้วยหรือต้องอยู่ในอาร์กิวเมนต์ของฟังก์ชัน Aggregate ในทางปฏิบัติแล้วเงื่อนไขการค้นหาในประโยค HAVING อย่างน้อยที่สุดจะมีหนึ่งฟังก์ชัน Aggregateเสมอ มิฉะนั้นแล้วเงื่อนไขการค้นหาควรจะถูกนำมาใช้ในประโยค WHERE

การใช้ฟังก์ชัน Scalar

ฟังก์ชัน Scalar ใช้เพื่อคำนวณค่าต่างๆฟังก์ชัน Scalarไม่จำเป็นต้องมีพารามิเตอร์ก็ได้ แต่ถ้ามีก็สามารถมีได้ตั้งแต่พารามิเตอร์เดียวจนถึงหลายพารามิเตอร์ ผลลัพธ์ของค่าฟังก์ชันScalarจะขึ้นอยู่กับพารามอเตอร์ต่างๆ ฟังก์ชันScalar สามารถแบ่งออกได้หลายประเภทโดยขึ้นอยู่กับชนิดข้อมูลในพารามิเตอร์

ข้อแตกต่างระหว่างฟังก์ชันScalarและฟังก์ชันAggregate มีดังต่อไปนี้ ฟังก์ชันScalar จะใช้กับค่าที่ไม่มีการเกิดขึ้นซ้ำและส่งค่ากลับค่าเดียวซึ่งขึ้นอยู่กับค่าที่รับมารวมทั้งบางฟังก์ชันไม่จำเป็นต้องมีพารามิเตอร์ขณะที่ฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Aggregate จะใช้กับค่าที่เป็นกลุ่มต้องมีพารามิเตอร์เสมอและส่งค่ากลับค่าเดียว ซึ่งบางครั้งเรียกว่าการสรุปค่า ฟังก์ชันScalar เช่น

ฟังก์ชัน ABS

เมื่อใช้ฟังก์ชันนี้ด้วยพารามิเตอร์ที่เป็นตัวเลขจำนวน จะส่งค่าสมบูรณ์หรือค่า Absolute

ฟังก์ชัน ADO_MONTHS

เราใช้งานฟังก์ชันนี้เพื่อเพิ่มจำนวนเดือน ฟังก์ชันนี้จะมี 2 พารามิเตอร์ พารามิเตอร์ตัวแรกเป็นตัว แปรชนิด DATE ที่ต้องการเพิ่มจำนวนเดือน พารามิเตอร์ตัวที่สองเป็นตัวแปรชนิดจำนวนเต็มซึ่งจะเป็นจำนวนเดือน

ฟังก์ชัน CEIL

CEIL ใช้เพื่อคืนค่าจำนวนเต็มน้อยที่สุดซึ่งมากกว่าหรือเท่ากับค่าในพารามิเตอร์

ฟังก์ชัน CHR

CHR ใช้เพื่อแปลงโค้ด ASCII ซึ่งอยู่ระหว่าง 0 ถึง 255 ให้เป็นอักขระ

ฟังก์ชัน ASCII

ASCII ใช้เพื่อแปลงอักขระให้เป็นโค้ด ASCII ผลลัพธ์ที่ได้จะอยู่ในจำนวนเต็มบวกตั้ง 0 ถึง 255

ฟังก์ชัน DECODE

DECODE ใช้สำหรับถอดรหัสจากการค้นหาโดยการเปรียบเทียบ

DECODE < ตัวแปรที่จะทำการถอดรหัส, เงื่อนไขที่ต้องการ1, ผลลัพธ์, เงื่อนไขที่ต้องการ2, ผลลัพธ์2,>

```
If SEX = M Then
        Result: = MAN
Else If SEX = F Then
        Result: = FEMALE
```

ฟังก์ชัน FLOOR

FLOOR ใช้เพื่อคืนค่าจำนวนเต็มที่มีค่ามากที่สุดซึ่งน้อยกว่าหรือเท่ากับในพารามิเตอร์

FLOOR (12.34) ผลลัพธ์ 12

FLOOR (-12.34) ผลลัพธ์ -13

ฟังก์ชัน GREATEST

GREATEST ใช้เพื่อหาค่าสูงสุดในลิสต์พารามิเตอร์ พารามิเตอร์มีได้ตั้งแต่สองพารามิเตอร์ขึ้นไป

GREATEST (20, 50, 100, 150) ผลลัพธ์จะเป็น 150

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GREATEST (‘A’, ‘B’, ‘E’, ‘D’) ผลลัพธ์จะเป็น E

ฟังก์ชัน LAST_DAY

LAST_DAY ใช้เพื่อหาวันสุดท้ายของเดือนจากพารามิเตอร์ที่เป็นข้อมูลชนิด DATE

LAST_DAY (‘27/2/1999’) ผลลัพธ์จะเป็น27/2/1999

ฟังก์ชัน LEAST

LEAST ใช้เพื่อหาค่าต่ำสุดในลิสต์พารามิเตอร์ พารามิเตอร์มีได้ตั้งแต่สองพารามิเตอร์ขึ้นไป

ฟังก์ชัน LENGTH

LENGTH ใช้เพื่อคืนค่าจำนวนอักขระหน่วยเป็นจำนวนเต็ม

ฟังก์ชัน INITCAP

INITCAP ใช้เพื่อคืนค่าอักขระตัวแปรของสตริงหรือข้อความในพารามิเตอร์ ให้เป็นตัวพิมพ์ใหญ่ ถ้าในพารามิเตอร์มีหลายๆคำ จะคืนค่าอักขระตัวแรกของคำๆนั้น

INITCAP (‘sql’)

INITCAP (‘sql Language’)

INITCAP (‘s q l ‘)

ฟังก์ชัน INSTR

INSTR ใช้เพื่อหาตำแหน่งในพารามิเตอร์แรกสุด เมื่อกำหนดค่าในพารามิเตอร์

ฟังก์ชัน LPAD

LPAD ใช้เพื่อเติมค่าทางด้านซ้ายของคำหรือคอลัมน์ในพารามิเตอร์ตัวแรก ด้วยค่าในพารามิเตอร์ที่สาม และจำนวนที่เติมจะขึ้นอยู่กับพารามิเตอร์ที่สอง ซึ่งเมื่อเติมแล้วจะไม่มากกว่าค่าในพารามิเตอร์

ฟังก์ชัน RPAD

RPAD ใช้เพื่อเติมค่าทางด้านขวามือของคำหรือคอลัมน์ในพารามิเตอร์ตัวแรก ด้วยค่าในพารามิเตอร์ที่สาม และจำนวนที่เติมจะขึ้นอยู่กับพารามิเตอร์ที่สอง ซึ่งเมื่อเติมแล้วจะไม่มากกว่าค่าในพารามิเตอร์

ฟังก์ชัน LTRIM

LTRIM ใช้เพื่อตัดอักขระด้านซ้ายจากพารามิเตอร์แรก โดยที่จำนวนอักขระที่จะถูกตัดออกจะขึ้นอยู่กับพารามิเตอร์ที่สอง และถ้าพารามิเตอร์ที่สองไม่ได้ระบุจะเปรียบเทียบเป็นการเว้นคำหรือช่องว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน RTRIM

RTRIM ใช้เพื่อตัดอักขระด้านขวาพารามิเตอร์แรก โดยที่จำนวนอักขระที่จะถูกตัดออกจะขึ้นอยู่กับพารามิเตอร์ที่สองและถ้าพารามิเตอร์ไม่ได้ระบุจะเปรียบเสมือนเป็นการเว้นคำไว้

ฟังก์ชัน MOD

MOD คืนค่าจากการหารที่เป็นเศษ โดยพารามิเตอร์แรกเป็นตัวตั้งและพารามิเตอร์ที่สองเป็นตัวหาร

MOD (17, 5)

ฟังก์ชัน MONTHS_BETWEEN

MONTHS_BETWEEN ใช้เพื่อคำนวณจำนวนเดือนที่อยู่ในระหว่างเดือนแรกกับเดือนที่สอง ค่าที่คำนวณจะเป็นจำนวนจริง ในการใช้งานจริงๆเราต้องคูณด้วยค่าเฉลี่ยของวันในหนึ่งเดือนคือ 30.4 แล้วใช้ฟังก์ชัน TRUNC เพื่อปัดเศษทิ้ง

ฟังก์ชัน NVL

NVL เป็นฟังก์ชันที่ใช้แทนค่า NULL ด้วยค่าอื่นๆ ถ้าค่าของพารามิเตอร์แรกเป็นค่า NULL ผลลัพธ์ของฟังก์ชันนี้จะเท่ากับค่าในพารามิเตอร์ที่สอง ถ้าค่าไม่เป็น NULL ค่าผลลัพธ์ก็ยังคงเป็นค่าในพารามิเตอร์แรก ฟังก์ชันนี้จะคล้ายกับฟังก์ชัน DECODE

ฟังก์ชัน POWER

POWER ใช้เพื่อคำนวณเลขยกกำลังโดยพารามิเตอร์แรกจะเป็นเลขฐานและพารามิเตอร์ที่สองเป็นเลขยกกำลัง

ฟังก์ชัน ROUND

ROUND ใช้เพื่อปัดเศษขึ้นหรือลงในพารามิเตอร์แรก โดยขึ้นอยู่พารามิเตอร์ที่สอง ถ้าไม่ได้ระบุจะปัดเศษเป็นศูนย์ ถ้าระบุเป็นจำนวนเต็มบวกจะปัดเศษขึ้นหรือลงตามที่ระบุและถ้าระบุเป็นจำนวนเต็มลบจะปัดจำนวนที่อยู่ด้านซ้ายทศนิยมให้เป็นศูนย์ตามจำนวนที่ระบุ

ฟังก์ชัน SIGH

SIGH ใช้เพื่อคืนค่าเครื่องหมายคณิตศาสตร์ ถ้าเป็นบวกในพารามิเตอร์จะคืนค่าเป็น 1 ถ้าเป็น 0 จะคืนค่าเป็น 0 และถ้าเป็นลบจะคืนค่า -1

ฟังก์ชัน SQRT

SQRT ใช้เพื่อคำนวณการถอดรากกำลังสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน SUBSTR

SUBSTR ใช้เพื่อตัดข้อความในพารามิเตอร์แรกโดยจะเริ่มตัดข้อความให้เหลือ ณ ตำแหน่งแรกตามพารามิเตอร์ที่สองและสิ้นสุด ณ ตำแหน่งแรกบวกด้วยจำนวนที่ระบุในพารามิเตอร์ที่สาม ถ้าพารามิเตอร์ที่สามมีการละไว้จะตัดข้อความให้เหลือ ณ ตำแหน่งแรกในพารามิเตอร์ที่สองจนถึงอักขระสุดท้าย

ฟังก์ชัน TO_CHAR

TO_CHAR ใช้เพื่อจัดรูปแบบการแสดงวันที่ของตัวแปร DATE ที่อยู่ในพารามิเตอร์แรก ส่วนรูปแบบที่ต้องการแสดงจะอยู่ในพารามิเตอร์

ฟังก์ชัน TO_DATE

TO_DATE ใช้เพื่อแปลงข้อมูลในพารามิเตอร์แรกมาอยู่ในรูปแบบที่กำหนดในพารามิเตอร์ที่สองและใช้รูปแบบเดียวกับฟังก์ชัน TO_CHAR

ฟังก์ชัน UPPER

UPPER ใช้เพื่อแปลงอักขระหรือสตริงให้เป็นตัวพิมพ์ใหญ่หมด

2.5 หลักการสร้าง SQL เพื่อกำหนดออบเจ็กต์ฐานข้อมูล

ในการสร้างตารางใหม่จะใช้คำสั่ง CREATE TABLE ซึ่งมีรูปแบบดังนี้

CREATE TABLE <table name>

< column name 1 ><data type>[CONSTRAINT<constraint name><integrity constraint>]

…

< column name N ><data type>[CONSTRAINT<constraint name><integrity constraint>]

โดยที่

table name คือชื่อตารางที่เราจะสร้างใหม่และจะต้องไม่มีชื่อซ้ำกับตารางที่มีอยู่แล้ว

column name คือ ชื่อ คอลัมน์หรือฟิลด์

data type คือชนิดของข้อมูล

constraint name คือเงื่อนไขข้อบังคับที่อยู่ภายใต้ระบบฐานข้อมูลซึ่งใช้เพื่อระบุเงื่อนไข และชื่อเงื่อนไข ข้อบังคับนี้จะต้องไม่ซ้ำกับในตารางที่เราสร้างรูปแบบทั่วไปในการตั้งเงื่อนไขข้อบังคับขึ้นอยู่กับ integrity constraint

รูปแบบการตั้งชื่อเงื่อนไขข้อบังคับ

<table name>_<column name>_<Abb.Constraint>

CONSTRAINT customer_Username_ID_PRIMARY KEY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

***ในกรณีที่ชื่อตารางยาวและทำให้เงื่อนไขข้อบังคับเกิน 30 อักขระ ซึ่งเป็นข้อห้ามของระบบฐานข้อมูลเราอาจจะใช้ตัวย่อแทนก็ได้

Pk = Primary Key

Fk = Foreign Key

Un = Unique

Cc = Check condition

Nn = Not Null

2.5.1 เงื่อนไขบังคับความถูกต้องและความสมบูรณ์

1.เงื่อนไขข้อบังคับคีย์หลัก

เมื่อเราใช้เงื่อนไขข้อบังคับกับคอลัมน์ระบบฐานข้อมูลจะบังคับตามกฎที่ว่า

- ค่าในคอลัมน์จะต้องไม่เป็นค่าซ้ำกัน
- ค่าในคอลัมน์จะต้องมีค่าอยู่และไม่เป็นค่าที่ไม่ทราบ

[CONSTRAINT <constraint name>] Primary Key

2.เงื่อนไขข้อบังคับคีย์นอก

เงื่อนไขข้อบังคับกับความถูกต้องและความสมบูรณ์ในการอ้างอิงกับคอลัมน์อื่นๆ ซึ่งอาจจะเป็นตารางเดียวกัน หรือ ตารางอื่นก็ได้ บ่อยครั้งค่าต่างๆในคอลัมน์คีย์นอกจะอ้างกับค่าในคอลัมน์ของตารางหลัก ซึ่งจะทำให้เกิดความสัมพันธ์แบบตารางลูกกับตารางหลัก คำสั่งเงื่อนไขข้อบังคับคีย์นอกคือ

CONSTRAINT<constraint name> REFERENCE<table name>[column][ON DELETE CASCADE]

***โดยที่ ON DELETE CASCADE เป็นตัวเลือกที่จะมีหรือไม่มีก็ได้ ตัวเลือกนี้สามารถเป็นได้ 4 กรณีด้วยกัน

1. ON DELETE CASCADE จะมีความหมายเมื่อเราลบเร็คคอร์ดในตารางหลัก จะลบเร็คคอร์ดในตารางลูกที่มีคีย์นอกด้วย โดยที่คีย์นอกในตารางลูกต้องจับคู่กับคีย์หลักในตารางหลักจึงจะลบเร็คคอร์ดโดยอัตโนมัติ
2. ON DELETE SET NULL จะมีความหมายเมื่อเราลบเร็คคอร์ดในตารางหลัก จะลบเร็คคอร์ดในตารางลูกโดยอัตโนมัติก็ต่อเมื่อคีย์นอกจับคู่กับคีย์หลักได้และเมื่อลบแล้วจะเซตค่าในคอลัมน์ที่มีคีย์นอกเป็นค่า NULL นั้นหมายความว่าค่าในคอลัมน์ที่มีคีย์นอกตารางลูกต้องไม่เป็น NOT NULL
3. ON DELETE DEFAULT เมื่อลบเร็คคอร์ดในตารางหลัก จะลบเร็คคอร์ดในตารางลูกโดยอัตโนมัติก็ต่อเมื่อคีย์นอกจับคู่กับคีย์หลักได้ และเมื่อลบแล้วจะเซตค่าในคอลัมน์ที่มีคีย์นอกเป็นค่าดีฟอลต์
4. NO Action ซึ่งจะเป็นค่าดีฟอลต์เมื่อเราไม่ได้ระบุในตัวเลือกนี้ ระบบฐานข้อมูลจะปฏิเสธการลบในตารางลูกทันทีที่เราลบเร็คคอร์ดในตารางหลัก

3.เงื่อนไขข้อบังคับไม่เป็นค่า NULL

เมื่อเราใช้เงื่อนไขข้อบังคับไม่ให้เป็นค่า NULL จะป้องกันการป้อนข้อมูลหรือ ค่า NULL ลงในคอลัมน์โดยที่ปกติแล้วเมื่อใช้เงื่อนไขข้อบังคับคีย์หลัก มันจะมีเงื่อนไขข้อบังคับไม่ให้เป็นค่า NULL โดยอัตโนมัติอยู่แล้ว แต่คีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกสามารถเป็นค่า NULL ได้ซึ่งบางกรณีคีย์นอกจะต้องไม่เป็นค่า NULL [CONSTRAINT <constraint>] NOT NULL

4.เงื่อนไขข้อบังคับค่าไม่ซ้ำกัน

เงื่อนไขข้อบังคับนี้จะคล้ายกับเงื่อนไขข้อบังคับคีย์หลัก ยกเว้นตรงที่ว่าเงื่อนไขนี้สามารถมีค่าที่ไม่ทราบได้ รูปแบบเงื่อนไขข้อบังคับค่าไม่ซ้ำกัน มีรูปแบบคล้ายกับเงื่อนไขข้อบังคับคีย์หลักและไม่เป็นค่า NULL

[CONSTRAINT <constraint name>] Unique

5.เงื่อนไขข้อบังคับการตรวจสอบค่า

เงื่อนไขนี้ใช้เพื่อจำกัดค่าที่จะนำมาอัพเดตหรือเพิ่มข้อมูลในเร็คคอร์ดใหม่โดยการใช้เงื่อนไขการตรวจสอบข้อมูลก่อน

[CONSTRAINT <constraint name>] CHECK (<condition>)

ซึ่ง Condition คือเงื่อนไขการตรวจสอบข้อมูลเช่นเดียวกับประโยค WHERE เพื่อกรองข้อมูล

2.5.2 การแก้ไขตารางฐานข้อมูล

เราจะใช้คำสั่ง ALTER เพื่อเปลี่ยนแปลงโครงสร้างตารางสามารถทำได้ 3 รูปแบบดังนี้

แบบที่ 1 การเพิ่มคอลัมน์ลงในตารางฐานข้อมูล

ALTER TABLE <table name >

ADD(<column name><data type>[CONSTRAINT])

เราจะใช้คำสั่ง ALTER TABLE ... ADD ก็ต่อเมื่อต้องการเพิ่มคอลัมน์ใหม่หรือเพิ่มคีย์นอกหรือเพิ่มเงื่อนไขการตรวจสอบ

แบบที่ 2 การแก้ไขคอลัมน์

ALTER TABLE <table name >

MODIFY<column name ><data type>

เราจะคำสั่ง ALTER TABLE ... MODIFY ก็ต่อเมื่อต้องการแก้ไขชนิดของข้อมูลและขนาดของคอลัมน์เงื่อนไข NULL หรือเพิ่มค่า ดีฟอลต์

แบบที่ 3 การลบเงื่อนไขข้อบังคับต่างๆทิ้ง

ALTER TABLE <table name >

DROP CONSTRAINT < constraint name>

ข้อจำกัดของคำสั่ง ALTER TABLE

สำหรับข้อจำกัดในการใช้คำสั่ง ALTER TABLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.การเปลี่ยนแปลงชนิดข้อมูล ขนาด และค่าดีฟอลต์ของคอลัมน์ จะทำได้ก็ต่อเมื่อค่าในคอลัมน์นั้นไม่มีค่าอยู่

2.การเพิ่มคีย์หลักจะทำได้ก็ต่อเมื่อค่าในคอลัมน์ปัจจุบันไม่มีค่าซ้ำกันเลย

3. การเพิ่มเงื่อนไขข้อบังคับการตรวจสอบและค่าที่ไม่ซ้ำกันแก่คอลัมน์นั้นจะทำได้ก็ต่อเมื่อค่าในคอลัมน์ปัจจุบันสอดคล้องกับเงื่อนไขข้อบังคับที่ถูกเพิ่มเข้ามา

4.การลบคอลัมน์ออกจากตารางไม่สามารถทำได้

5.การเพิ่มคีย์นอกจะทำได้ก็ต่อเมื่อค่าในคอลัมน์ปัจจุบันเป็นค่า NULL และต้องมีอยู่ในตารางที่ถูกอ้างอิงด้วย

6.การเพิ่มเงื่อนไขข้อบังคับไม่เป็นค่า NULL 1 จะทำได้ก็ต่อเมื่อค่าในคอลัมน์ไม่เป็น NULL เลย

2.5.3 การลบตารางออกจากฐานข้อมูล

เมื่อไม่ต้องการใช้ข้อมูลในตารางนั้นอีกต่อไป เราจะใช้คำสั่ง DROP เพื่อลบตารางออก

DROP TABLE <table name>

Ex.Code DROP TABLE Customer ID;

DROP TABLE TIME OUT;

DROP TABLE Provinces;

2.5.4 การเปลี่ยนชื่อตาราง

ถ้าต้องการเปลี่ยนชื่อตารางให้ใช้คำสั่ง RENAME <old table name> TO<newtable name>

2.5.5 การสร้างและลบ SYNONYM และ INDEX

การใช้ SYNONYM ไม่ได้เปลี่ยนความหมายของการคิวรีคือ เรายังสามารถทำการคิวรีได้ ถ้าเราได้รับการอนุญาติให้เข้าถึงตารางของผู้ใช้คนอื่นๆอย่างไรก็ตาม SYNONYM จะเหมือนกับเป็นตารางที่เราสร้างขึ้นในอีกชื่อหนึ่งของตาราง

CREATE [PUBLIC] SYNONYM <symnonym>

FOR <table name>

2.5.6 วัตถุประสงค์หลักการใช้ SYNONYM

1.ผู้ใช้สามารถสร้าง SYNONYM ให้เป็นชื่อตารางหรือวิวที่สองที่ผู้ใช้ต้องการใช้อีกชื่อ ซึ่งอาจจะมีเหตุผลว่าตารางหรือวิวนั้นไม่เหมาะสมต่อการทำบางสิ่งด้วยเหตุผลบางอย่าง

2.ผู้ใช้สามารถสร้าง SYNONYM จากตารางหรือวิวที่มีผู้ใช้คนอื่นเป็นเจ้าของอยู่ถ้าเราต้องการใช้ SYNONYM นี้เราก็ระบุชื่อเจ้าของแล้ววางไว้ข้างหน้าชื่อตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7 การใช้INDEXกับตารางข้อมูล

การใช้INDEX เพื่อเข้าถึงข้อมูลโดยตรง ซึ่งทำให้การเข้าถึงข้อมูลมีความรวดเร็วและมีประสิทธิภาพในการจัดเรียงดีขึ้นเมื่อใช้กับตารางที่มีจำนวนเร็คคอร์ดมากๆซึ่งในตารางหนึ่งสามารถมี INDEX ได้หลายๆคอลัมน์ ทำให้เราสามารถเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพ

ถ้าเราไม่ใช้ INDEX ระบบการจัดการฐานข้อมูลจะใช้วิธีการเข้าถึงข้อมูลแบบลำดับซึ่งจะทำการอ่านทีละเร็คคอร์ดแล้วตรวจสอบข้อมูลในเร็คคอร์ดนั้นๆตรงกับข้อมูลที่เราต้องการหรือไม่ซึ่งทำให้การหาผลลัพธ์ช้าหากฐานข้อมูลมีขนาดใหญ่โดยมีรูปแบบคำสั่งดังนี้

CREATE [UNIQUE] INDEX <index name>

ON<table name>(<column name 1>),(<column name2>),…)

โดย UNIQUE เป็นตัวเลือกที่จะมีหรือไม่มีก็ได้ ถ้าระบุUNIQUE อินเด็กซ์จะรับประกันว่าแต่ละดัชนีแถวจะไม่มีค่าซ้ำกันในคอลัมน์ดัชนี ถ้าเราสร้างดัชนีในคอลัมน์หนึ่งแบบUNIQUE แล้วและพยายามป้อนข้อมูลหรืออัพเดทข้อมูลในคอลัมน์นั้นด้วยค่าที่ซ้ำกัน

2.5.8 การใช้งานซีเควนซ์

ซีเควนซ์เป็นลิสต์ที่มีลำดับของหมายเลข ซึ่งหมายเลขนี้ถูกสร้างโดยอัตโนมัติจากฐานข้อมูลและสามารถนำมาใช้กับคำสั่ง INSERT UPDATE DELETE

CREATE SEQUENCE <sequence name>

[START WITH <integer>]

[INCREMENT BY <integer>]

[MAXVALUE<integer>] | NOMAXVALUE

[MINVALUE<integer>] |NOMINVALUE

[CYCLE | NOCYCLE]

[ORDER|NOORDER]

[CACHE<integer>]

[NOCACHE]

คำสั่ง CREATE SEQUENCE เป็นคำสั่งที่ต้องการใช้สร้างซีเควนซ์เท่านั้น ส่วนคำสั่งที่อยู่ใน […] เป็นคำสั่งทางเลือกทั้งหมด ซึ่งสามารถอธิบายได้ดังต่อไปนี้

START WITH ใช้ระบุค่าเริ่มต้นซีเควนซ์ ถ้าไม่มีการระบุค่านี้ ค่าดีฟอลต์จะเป็น 1 ถ้าเรามีข้อมูลในคอลัมน์หนึ่งซึ่งเป็นคีย์หลักตั้งแต่ 1001 – 1010เราควรจะเริ่มต้นด้วยค่า 1011

INCREMENT BY ใช้ระบุค่าที่ซีเควนซ์จะถูกเพิ่มขึ้น ถ้าไม่มีการระบุค่านี้ มันจะเพิ่มซีเควนซ์เป็น 1 โดยอัตโนมัติ ถ้าค่านี้เป็นจำนวนเต็มบวก จะเป็นการสร้างซีเควนซ์เพิ่มขึ้น แต่ถ้าค่าเป็นจำนวนเต็มลบจะเป็นการสร้างซีเควนซ์ลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAXVALUE ใช้ระบุค่าสูงสุดของหมายเลขซึ่งซีเควนซ์สามารถสร้างได้ ถ้าเราไม่ได้ระบุค่านี้ค่าสูงสุดจะเป็น NOMAXVALUE ซึ่งก็คือ $1*10^{27}$

MINVALUE ใช้ระบุค่าต่ำสุดของหมายเลขซึ่งซีเควนซ์สามารถสร้างได้ ถ้าเราไม่ได้ระบุค่านี้ค่าต่ำสุดจะเป็น NOMINVALUEแทน

CYCLE ใช้ระบุการสร้างซีเควนซ์แบบวนลูปก็ต่อเมื่อซีเควนซ์ไปถึงค่าสูงสุดแล้วจะกลับมาเริ่มต้นที่ค่าต่ำสุดอีกครั้ง ถ้าไม่ระบุค่านี้จะใช้ค่าดีฟอลต์ NOCYCLE แทนที่จะสร้างค่าซีเควนซ์ไปจนถึงค่า MAXVALUE และจะไม่กลับมาค่าเริ่มต้นอีก

ORDER ใช้เพื่อให้แน่ใจว่าหมายเลขที่ถูกสร้างขึ้นมีลำดับที่ถูกต้องเสมอ ถ้าไม่มีการระบุจะใช้ค่าดีฟอลต์ NOORDER ซึ่งจะไม่รับประกันว่าหมายเลขที่สร้างขึ้นจะมีลำดับ

CACHE ใช้ระบุจำนวนหมายเลขซีเควนซ์ที่จะถูกเก็บไว้ในหน่วยความจำ CACHE ถ้าเราไม่ได้ระบุจะกำหนดค่าที่เป็น 20 เพื่อเก็บหมายเลขซีเควนซ์ไว้ใน CACHE

NOCACHE ไม่มีการกำหนดหน่วยความจำ CACHE สำหรับเก็บหมายเลขซีเควนซ์

2.5.9 การสร้างวิว (VIEW)

วิว (VIEW) เปรียบเสมือนกับตารางในฐานข้อมูลซึ่งข้อมูลในวิวถูกกำหนดโดยการคิวรี

CREATE [OR REPLACE] VIEW <view name> | <column name>]AS

<Query [WITH CHECK OPTION [constraint <name>]]>

โดยที่

OR REPLACE จะสร้างวิวแทนที่วิวเดิม

view name คือชื่อวิวใหม่ที่เราต้องการสร้าง

column name คือชื่อคอลัมน์

Query คำสั่ง SELECET

เราสามารถเลือกที่จะกำหนดชื่อคอลัมน์หรือไม่มีก็ได้ ถ้ามีการเลือกเพื่อกำหนดชื่อคอลัมน์แล้วเราต้องใช้จำนวนคอลัมน์ตรงกับจำนวนคอลัมน์ใน Query ด้วย แต่ถ้าไม่มีการกำหนดชื่อคอลัมน์แต่ละชื่อคอลัมน์นี้จะตรงกับชื่อคอลัมน์ใน Query บางครั้งเราจำเป็นต้องกำหนดชื่อคอลัมน์อันเนื่องมากจาก Query ได้รวมคอลัมน์ที่มีการคำนวณ ประโยค Query ก็คือ การกำหนดคิวรีถ้ามีการระบุด้วย WITH CHECK OPTION ภาษา SQL จะตรวจสอบการทำ INSERT และ UPDATE แต่ละครั้ง เพื่อจะให้แน่ใจว่าผลลัพธ์ของเร็คคอร์ดจะต้องสอดคล้องกับเงื่อนไขในประโยค WHERE ที่เป็นการกำหนดการคิวรีให้แก่วิว ถ้าเร็คคอร์ดที่เพิ่มเข้าไปหรือเร็คคอร์ดที่แก้ไขไม่สอดคล้องกับเงื่อนไขคำสั่ง INSERT และ UPDATE ก็จะไม่สามารถทำต่อไปได้ ดัง Code

```
CREATE VIEW project AS
SELECT a. Username, a. Password, b. Telephone, b. Car
FROM Login a, Test b
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WHERE (a. Username = b. Username);


USE [TEST]
GO
/****** Object:  Table [dbo].[UserName]    Script Date: 09/03/2008 01:33:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[UserName](
        [UserName] [nvarchar](250) NULL,
        [Password] [nvarchar](250) NULL,
        [Auth] [nvarchar](250) NULL,
        [IsNormal] [nvarchar](1) NULL
) ON [PRIMARY]

USE [TEST]
GO
/****** Object:  Table [dbo].[Human]    Script Date: 09/03/2008 01:33:54 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Human](
        [HID] [nvarchar](10) NOT NULL,
        [InitialID] [nvarchar](2) NULL,
        [FullName] [nvarchar](100) NULL,
        [Sex] [nvarchar](1) NULL,
        [IDCard] [nvarchar](17) NULL,
        [Address] [nvarchar](250) NULL,
        [ProvinceID] [nvarchar](2) NULL,
        [Telephone] [nvarchar](100) NULL,
        [BirthDate] [datetime] NULL,
        [PID] [nvarchar](4) NULL,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

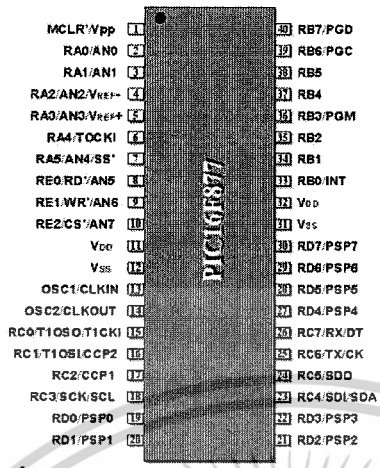
```
	[FirstDateToWork] [datetime] NULL,
	[Salary] [int] NULL,
	[PositionFee] [int] NULL,
	[DailyWage] [int] NULL,
	[HourWage] [int] NULL,
	[OTPerHour] [int] NULL,
	[Education] [nvarchar](500) NULL,
	[IsLeave] [nvarchar](1) NULL,
	[LastAccessBy] [nvarchar](16) NULL,
 CONSTRAINT [PK_Customer] PRIMARY KEY CLUSTERED
(
	[HID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'ÃËÑÊ¾'Ñ¡§Ò'' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'HID'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'ª×èÍ-Ê¡ØÅÅÙ¡¤éÒ' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'FullName'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'0 = ªÒÂ, 1 = Ë-Ô§' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'Sex'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'ÃËÑÊºÑµÃ»ÃÐªÒª'' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'IDCard'
GO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'·ÕèÍÂÙè' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'Address'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'ÃËÑÊ¨Ñ§ËÇÑ´' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'ProvinceID'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'à°ÍÃìâ·ÃÈÑ¾·ì' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'Telephone'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'ÇÑ¹à¡×´' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'BirthDate'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'ÇÑ¹·Ó§Ò¹ÇÑ¹áÃ¡' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'FirstDateToWork'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'0=»¡µÔ, 1=ÅÒÍÍ¡áÅéÇ' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Human',
@level2type=N'COLUMN',@level2name=N'IsLeave'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ไมโครคอนโทรเลอร์



รูปที่ 2.31 แสดงขาต่างๆของ PIC16F877

พอร์ต A จำนวน 6 ขาพอร์ต (RA0 ถึง RA5)

-อินพุตอะนาลอก หรือ อินพุตเอาท์พุตดิจิตอล

-อินพุตสัญญาณนาฬิกาจากภายนอกของไทเมอร์ 0 ผ่านขา RA4/T0CKI

-เอาท์พุตเปรียบเทียบสัญญาณอะนาลอก

พอร์ต B จำนวน 8 ขาพอร์ต (RB0 ถึง RB7)

-อินพุตเอาท์พุตดิจิตอล

-สามารถพูลอัพ (Pull-up) ขาพอร์ตด้วยกระบวนการทางซอฟท์แวร์ได้

-อินเตอร์รัปต์เนื่องจากสัญญาณภายนอก (External Interrupt) ที่ขา RB0/INt และขาอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงสัญญาณที่ขา

พอร์ต C จำนวน 8 ขาพอร์ต (RC0 ถึง RC7)

-อินพุตเอาท์พุตดิจิตอล

-ขาโมดูล CCP อินพุตตรงจับสัญญาณ เอาท์พุตเปรียบเทียบข้อมูล และสร้างสัญญาณ PWM

-ขาโมดูลUART สื่อสารข้อมูลอนุกรมแบบ SPI, I2C และRS-232

พอร์ต D จำนวน 8 ขาพอร์ต (RD0 ถึง RD7)

-อินพุตเอาท์พุตดิจิตอล

-ขาพอร์ตขนานเสริม (Parallel slave port)

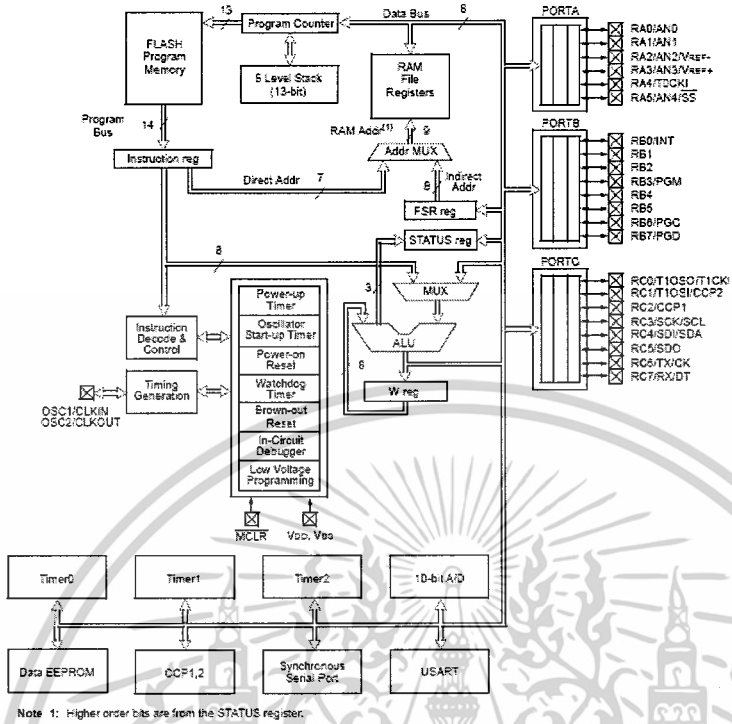
พอร์ต E จำนวน 3 ขาพอร์ต (RE0 ถึง RE2)

-อินพุตอะนาลอก หรือ อินพุตเอาท์พุตดิจิตอล

-ขาอ่านเขียนพอร์ตขนาน ใช้งานร่วมกับขาพอร์ตD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 โครงสร้างภายในชิพ



รูปที่ 2.32 ผังไดอะแกรมของ PIC16F877

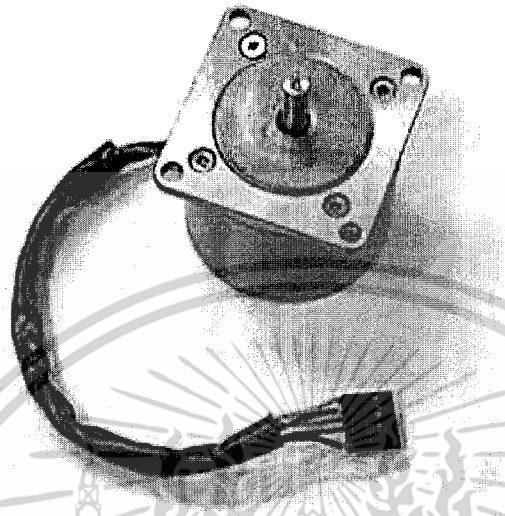
2.7 สเต็ปมอเตอร์ (STEPPING MOTOR)

สเต็ปมอเตอร์ เป็นมอเตอร์ที่ขับเครื่องด้วยพัลส์ ลักษณะการขับเคลื่อน จะหมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่อง แต่มีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะขับเคลื่อนได้ 1, 1.5, 1.8 หรือ 2 องศา แล้วแต่ละโครงสร้างของมอเตอร์ลักษณะที่ นำมอเตอร์ไปใช้ จะเป็นงานที่ต้องการตำแหน่งแม่นยำ เช่น ระบบขับเคลื่อนหัวแม่พิมพ์ในเครื่องพิมพ์ (PRINTER) ระบบขับเคลื่อนหัวอ่านในเครื่องอ่านบันทึกเหล็ก ระบบขับเคลื่อนตำแหน่งของปากกาใน X-Y PLOTER เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 ชนิดและโครงสร้างของสเต็ปมอเตอร์

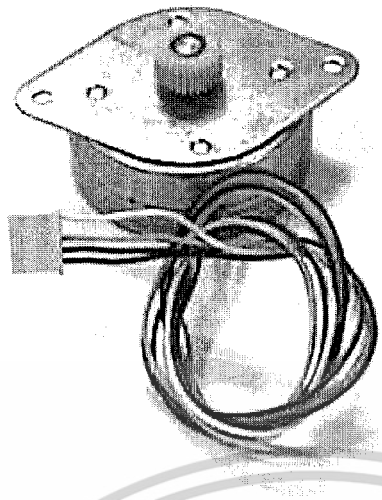
สเต็ปมอเตอร์มีลักษณะดังรูป



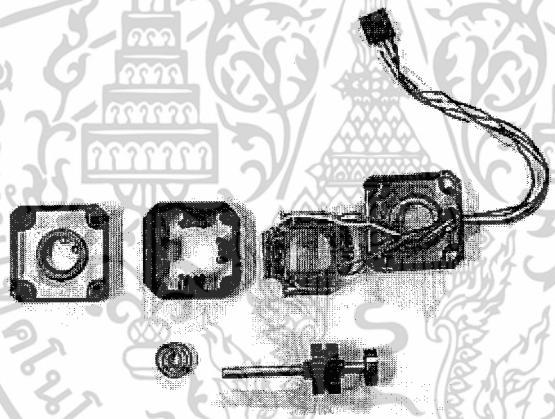
รูป 2.33 สเต็ปมอเตอร์แบบมีสาย 5 เส้น

รูป 2.34 มอเตอร์แบบมีสาย 6 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.35 สเต็ปมอเตอร์หลายแบบไบโพลาร์



รูป2.36 แสดงภาพถ่ายโครงสร้างสเต็ปมอเตอร์

สเต็ปมอเตอร์ที่พบในปัจจุบันมี 3 ลักษณะดังนี้

1. แบบแม่เหล็กถาวร (PERMANENT MAGNET_PM)

สเต็ปมอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆ โพล โดยมีโรเตอร์ (ROTOR) เป็นรูปทรง กระบอกฟันเลื่อย และโรเตอร์ทำด้วยแม่เหล็กถาวร เพื่อป้อนไฟกระแสตรง ให้กับขดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าผลักต่อโรเตอร์ ทำให้มอเตอร์หมุนมอเตอร์แบบ PM จะเกิดแรงฉุดยึดให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด

2. แบบแปรค่ารีลักแตนซ์ (VARIABLE RELUCTANCE- VR)

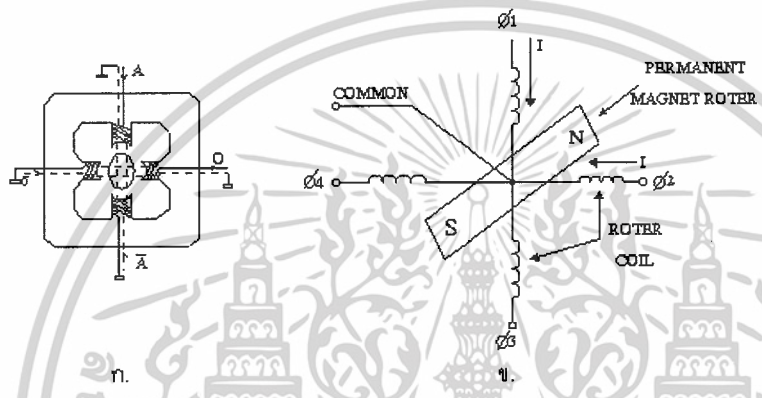
สเต็ปมอเตอร์แบบVR จะมีการหมุนโรเตอร์ได้อย่างอิสระ แม้จะไม่ได้จ่ายไฟให้โรเตอร์ทำจากสารเฟอร์โรแมกเนติก กำลังอ่อน มีลักษณะเป็นฟันเลื่อย รูปทรงกระบอกโดยจะมีความสัมพันธ์ โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะไปหมุนโรเตอร์ ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนท์ต่ำที่สุด ตำแหน่งที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุดดังนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างขดกันไป ก็ทำให้มอเตอร์ หมุนไปตำแหน่งต่างๆ กันโรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

3. แบบผสม (HYBRID-H)

สเต็ปมอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างดีทำให้ได้มุม การหมุนและครั้งน้อยและแม่นยำ ข้อดีก็คือ ให้แรงบิดสูงและมีขนาดกระทัดรัด และให้แรงฉุดยึดโรเตอร์นิ่งกับที่ตอนไม่จ่ายไฟ



รูป 2.36 แสดง (ก) โครงสร้าง (ข) วงจรเทียบเท่า (equivalent circuit) ของมอเตอร์ ชนิด 4 ขด แสดงมุมของโรเตอร์เทียบกับกระแสไฟฟ้าที่จ่ายแก่เฟสต่าง ๆ8 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

ระบบรักษาความปลอดภัย โดยใช้เทคโนโลยี RFID สามารถแบ่งออกเป็น 3 ส่วนที่สำคัญในการออกแบบคือ

1.ส่วนในการควบคุม TAG

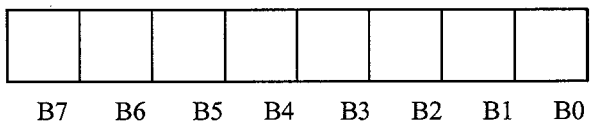
2.ส่วนของโปรแกรมข้อมูลผู้ถือบัตร

3.ส่วน PIC Microcontroller ควบคุมแขนกั้น

3.1 การออกแบบการควบคุม TAG

รูปที่ 3.1 แสดงรูป TAG ที่นำมาใช้งาน

การใช้งานการติดต่อ การแทน Password โดยการใช้ TAG แทน Password และเงินสดได้ทำการออกแบบให้ข้อมูลใน TAG สามารถใช้งานได้ในอัตราที่ทำการออกแบบในการทดลองโดยได้กำหนดให้บิตที่ใช้ในการระบุ User ให้มีจำนวน 8 บิต ซึ่งสามารถทำให้เราสามารถมีจำนวนผู้ใช้บริการได้ทั้งหมด 256 User โดยเราสามารถที่จะกำหนดจำนวนผู้ใช้บริการเพิ่มขึ้นได้โดยทำการเพิ่มจำนวนบิตที่ใช้ในการระบุ User

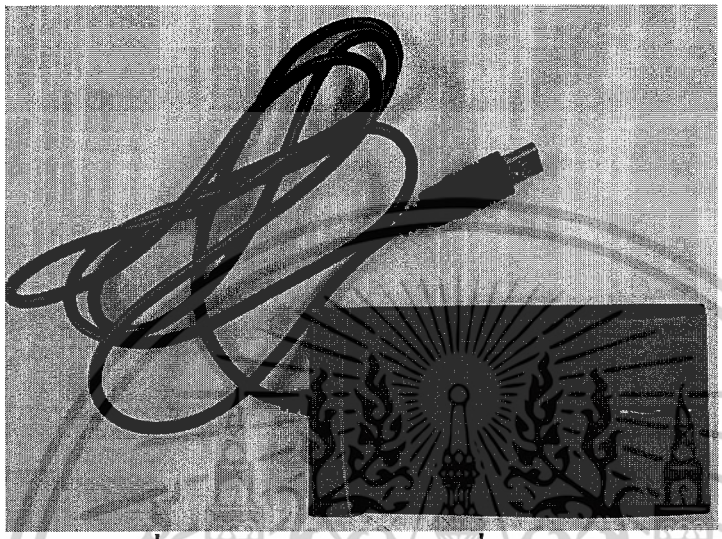


รูปที่ 3.2 บิตที่ใช้ในการระบุ User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเรากำหนดให้แต่บิตมีค่าเป็น 1 หรือ 0 ก็ได้ โดยจะมี User ตั้งแต่ 00000000 ถึง 11111111 โดยมีจำนวนผู้ใช้บริการ 256 User

3.2 การออกแบบการควบคุมตัวอ่าน



รูปที่ 3.3 แสดงรูป RFID Reader ที่นำมาใช้งาน

จะทำการออกแบบเป็น 2 ส่วน คือ

1. ส่วนของตัวอ่านเมื่อผู้ใช้บริการนำรถยนต์มาจอดในพื้นที่ให้บริการ

จะทำการติดตั้งตัวอ่านที่ส่วนของป้อมรักษาความปลอดภัย โดยเมื่อทำการแสดง TAG ให้กับตัวอ่าน ตัวอ่านจะทำการอ่านข้อมูลเฟรมที่ระบุผู้ใช้บริการให้กับคอมพิวเตอร์ หลังจากนั้นคอมพิวเตอร์ก็จะทำการเก็บข้อมูลที่ผู้ใช้บริการไว้ในส่วนของ Database และตัวอ่านก็จะเช็คข้อมูลในเฟรมที่เหลือเพื่อดูว่า TAG ที่ผู้ใช้บริการนั้นเป็น TAG ชนิดเดียวกันกับ TAG ที่ให้บริการหรือไม่ ถ้าถูกต้องแขนกั้นรถก็จะยกตัวขึ้นผู้ใช้บริการก็จะสามารถนำรถเข้าไปจอดในพื้นที่ให้บริการได้ แต่หาก TAG นั้นไม่ถูกต้องผู้ใช้บริการจะไม่สามารถผ่านแขนกั้นไปได้

2. ส่วนของตัวอ่านเมื่อผู้ใช้บริการนำรถออกจากพื้นที่ให้บริการ

จะใช้การติดตั้งตัวอ่านตัวเดิม โดยเมื่อเราทำการแสดง TAG หน้าตัวอ่าน ตัวอ่านจะทำการอ่านข้อมูลเฟรมที่ระบุผู้ใช้บริการให้กับคอมพิวเตอร์ ถัดจากนั้นคอมพิวเตอร์ก็จะทำการเก็บข้อมูลเวลาออกอีกครั้ง แล้วนำข้อมูลมาทำการเปรียบเทียบเวลาที่ใช้บริการ แล้วหักเงินสดที่อยู่ใน TAG ที่เหลืออยู่ จากนั้นที่กั้นก็จะเปิดออกรถผู้ใช้บริการก็สามารถออกจากที่จอดรถไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

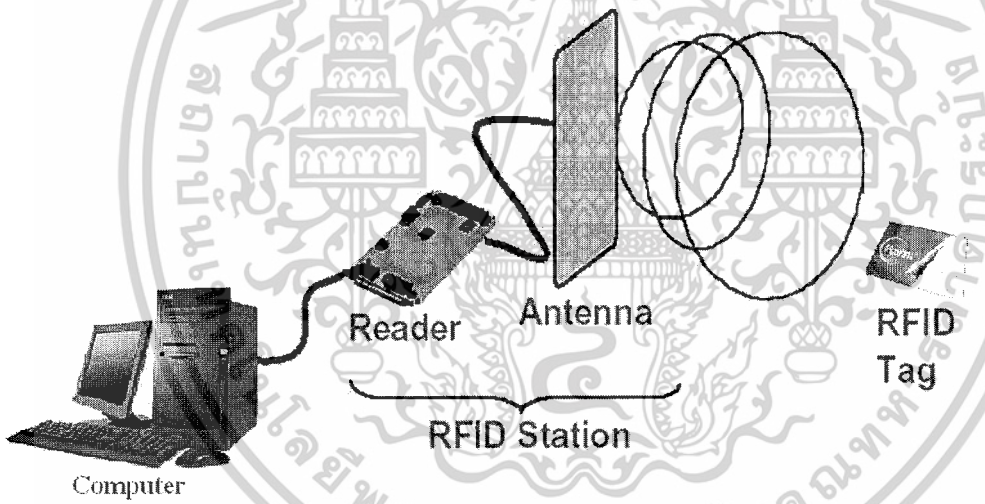
3.3 โปรแกรมข้อมูลผู้ถือบัตร

ระบบรักษาความปลอดภัยห้างสรรพสินค้าโดยใช้เทคโนโลยี RFID นี้มีหลักการออกแบบดังนี้

3.3.1 เครื่องอ่าน RFID จะทำการอ่านบัตรหากบัตรนั้นถูกต้องจึงจะยอมให้ใช้บริการ และถ้าบัตรไม่ถูกต้องก็จะไม่ยอมให้ใช้บริการ

3.3.2 การสร้างโปรแกรมรักษาความปลอดภัยห้างสรรพสินค้า ซึ่งโปรแกรมนั้นใช้โปรแกรม Visual C# .net โดยจะมีหน้าต่างของโปรแกรมเป็นแบบ ส่วนที่รับข้อมูลว่าผู้ใช้บริการนำรถเข้ามาจอดในห้างสรรพสินค้า จะเป็นส่วนที่ให้ผู้ใช้บริการเป็นผู้ป้อนข้อมูลเข้าฐานข้อมูล ซึ่งผู้ใช้บริการจะนำบัตรมาวางบริเวณเครื่องอ่านและระบบจะทำการเก็บ Username และ Password หลังจากนั้นจะเก็บเวลาขาเข้า เพื่อใช้ในการมาคำนวณคิดยอดเงินค่าบริการทั้งหมดและผู้ให้บริการก็จะสามารถดูรายละเอียดที่อยู่ เบอร์โทรได้เผื่อในกรณีเกิดเหตุฉุกเฉิน

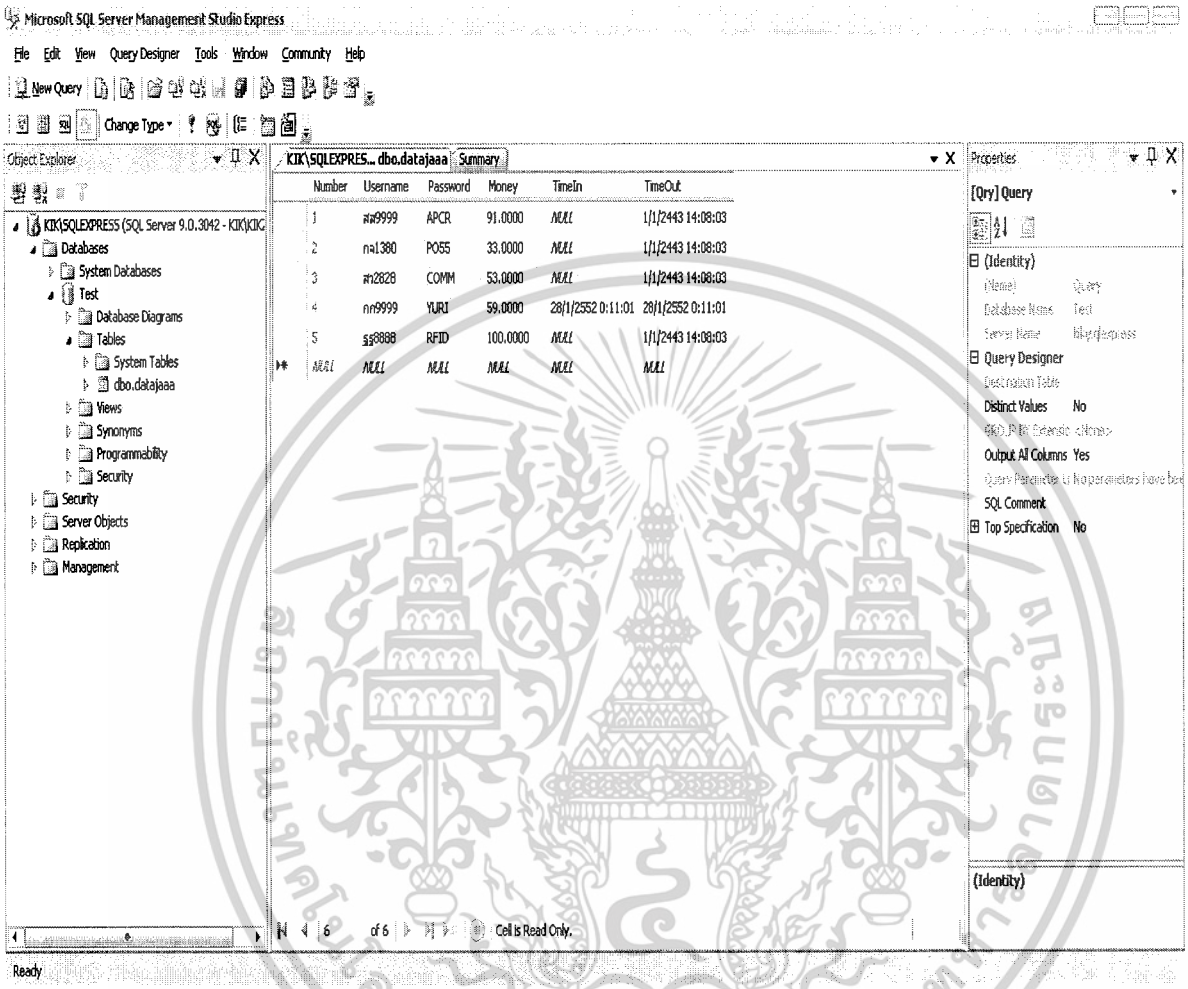
บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์ของระบบนี้



รูปที่ 3.4 ภาพการเชื่อมต่ออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ทำการแบ่งรายละเอียดของแต่ละส่วนประกอบใหญ่ๆแล้วเราจะนำรายละเอียดเหล่านั้นมาสร้างตารางฐานข้อมูล สำหรับหรับชนิดของฐานข้อมูลที่จะนำมาใช้ในที่นี้จะเลือกใช้ SQL ซึ่งมีความเหมาะสมในการสร้างฐานข้อมูลที่เหมาะกับเครื่อง Server

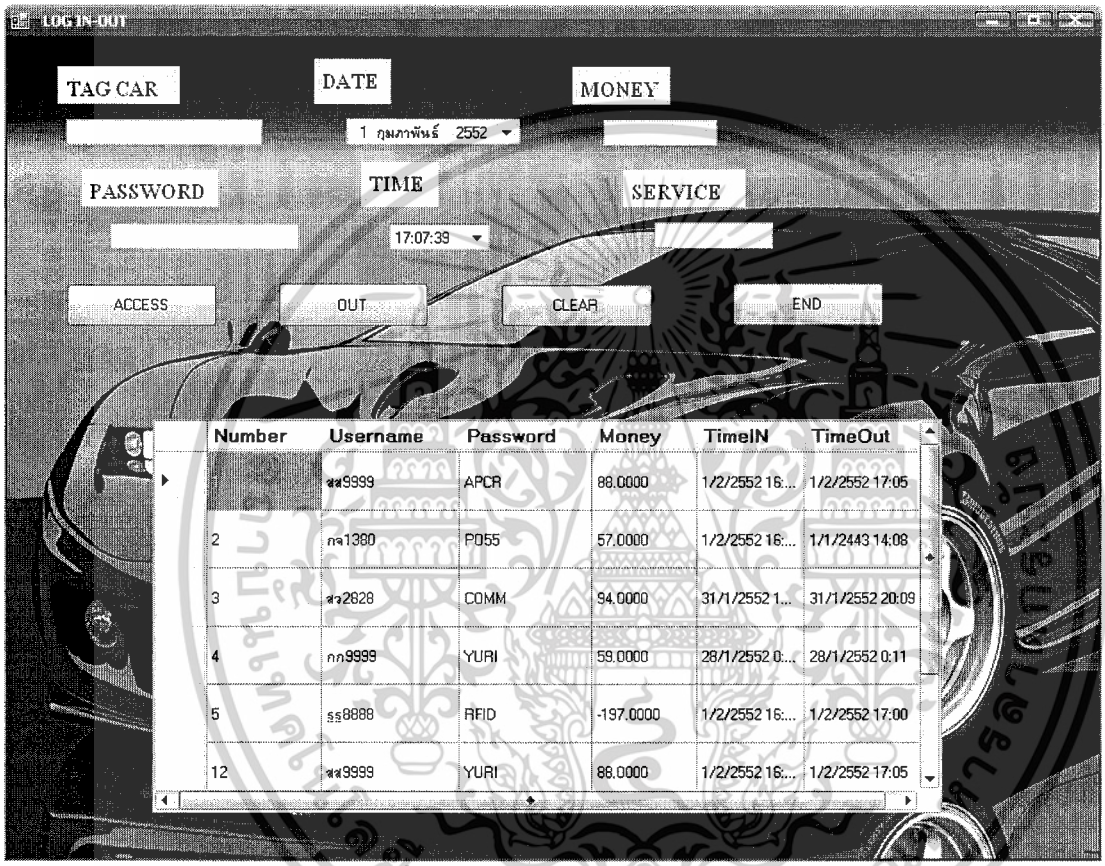


รูปที่ 3.5 ตารางฐานข้อมูลรายละเอียดของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 แนวคิดการออกแบบจอภาพในโปรแกรม

ส่วนประกอบของจอภาพที่จะนำมาใช้กับโปรแกรมถือว่าเป็นส่วนสำคัญเราจำเป็นต้องออกแบบให้ใช้งานได้ง่ายและสวยงามซึ่งตัวโปรแกรมส่วนหนึ่งนั้น ผู้ใช้งานจะเป็นผู้ใช้โดยตรงเช่น หน้าต่างเช็คยอดเงินและประวัติส่วนตัว สำหรับให้เจ้าหน้าที่ใช้งานคือ Login เข้าลานจอดรถในพื้นที่บริการ ซึ่งส่วนนี้จะต้องทำการใส่รหัสผ่านเพื่อใช้งาน ชื่อหน้าต่างต่างๆของโปรแกรมได้ออกแบบไว้ดังนี้



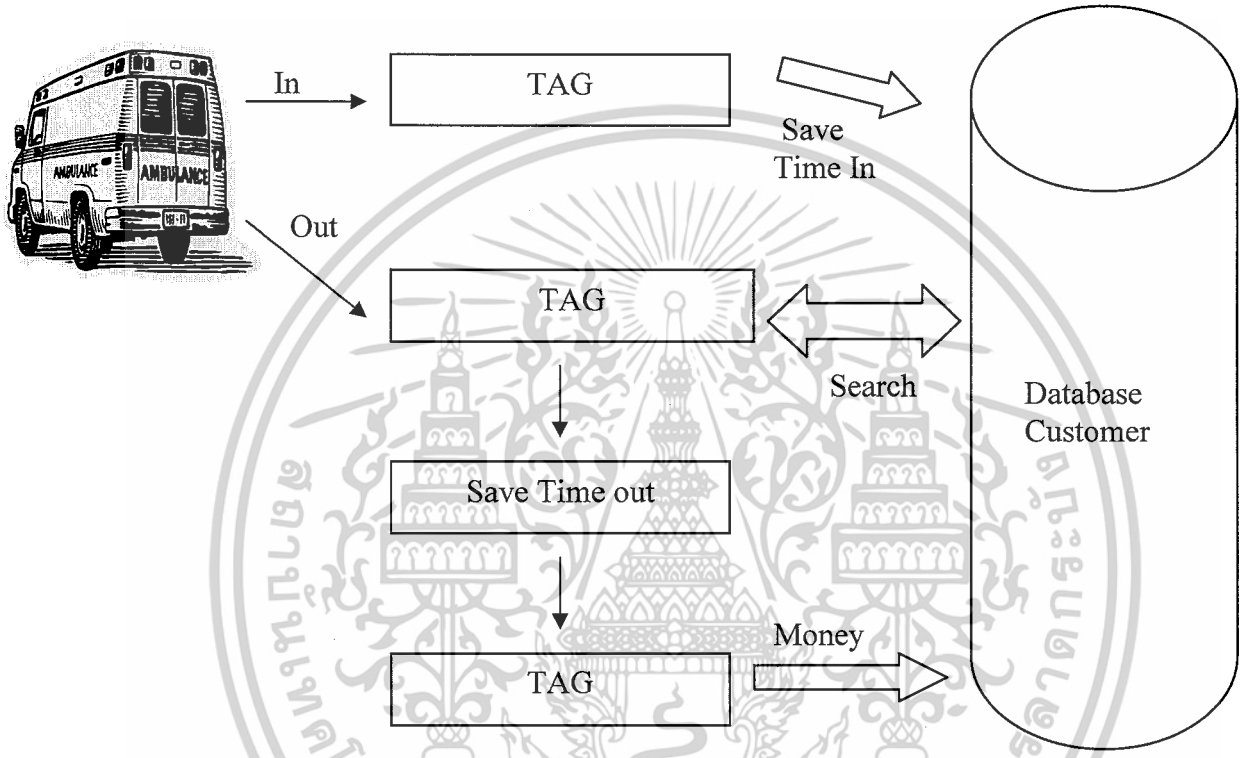
รูปที่ 3.6 หน้าต่างหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 แนวทางการเขียนโปรแกรม

หลังจากที่ได้สร้างหน้าต่างมาเป็นที่เรียบร้อยแล้วต่อมาจึงร่างขั้นตอนการเขียนโปรแกรมออกมาโดยใช้โฟล์วชาร์ทได้ดังนี้

การเก็บข้อมูลลูกค้าที่เข้าใช้บริการ

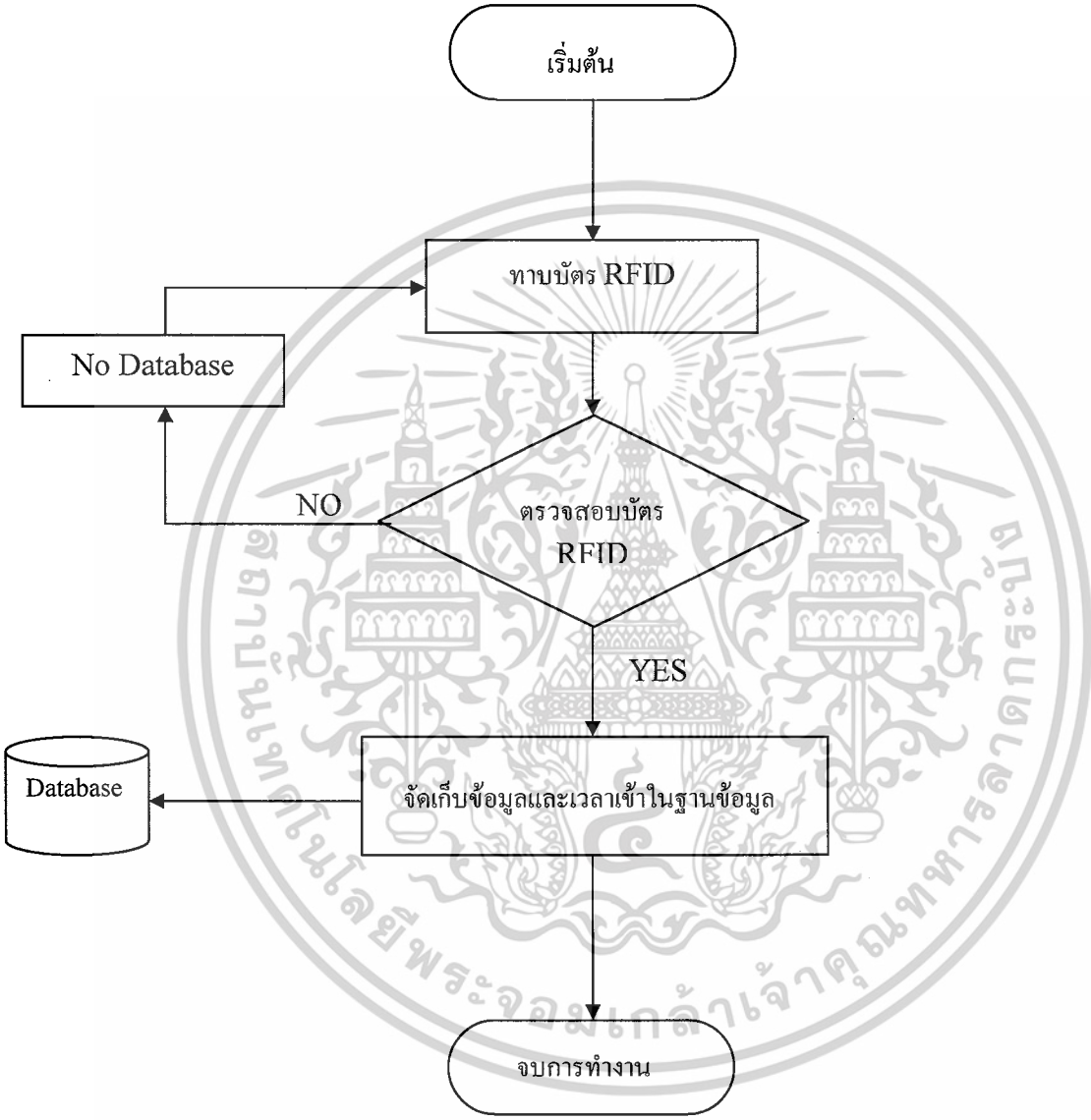


รูปที่ 3.7 ขั้นตอนเกี่ยวกับการเข้า – ออกรถยนต์ในห้างสรรพสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

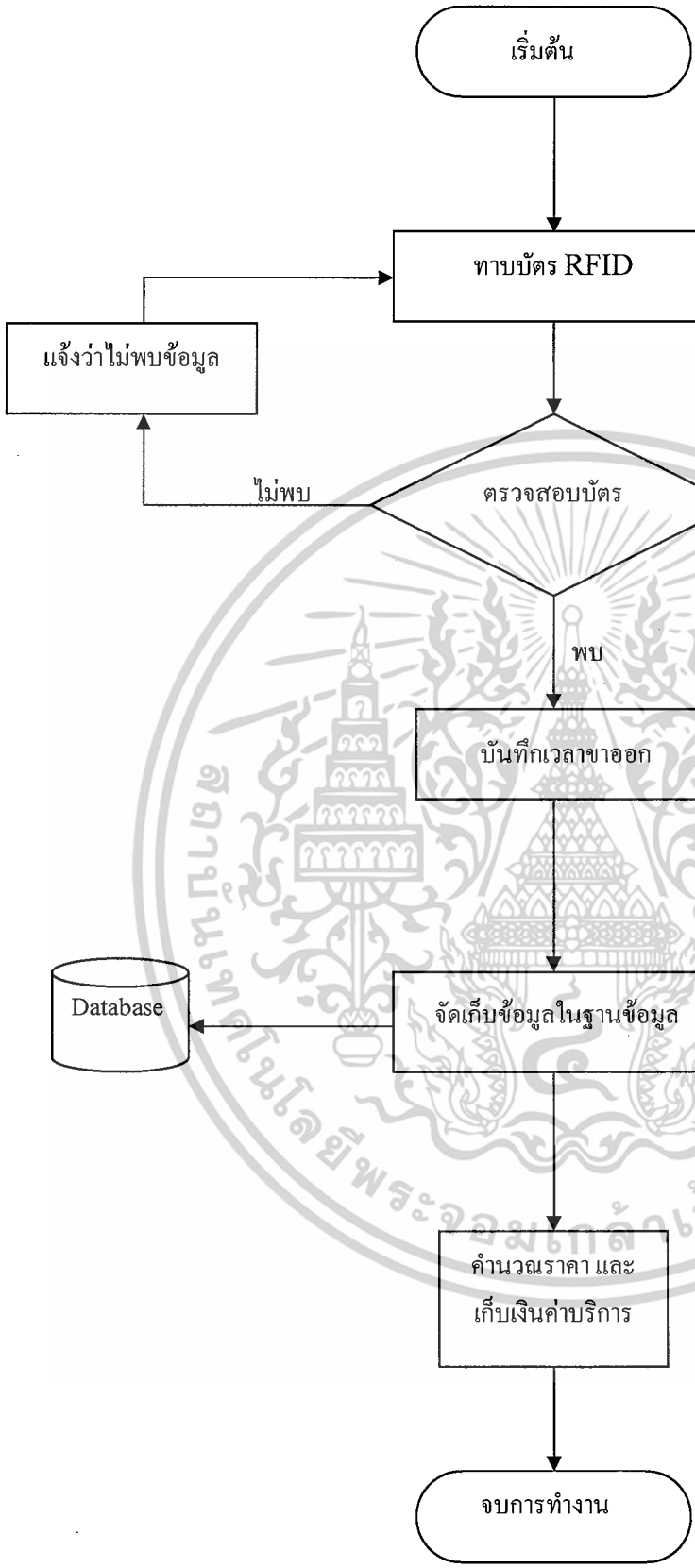
การใช้งานบริการระบบเพิ่มความปลอดภัยที่จอดรถยนต์

ก่อนที่จะใช้งานส่วนนี้จะต้องมีการทาบบัตร RFID หลังจากบัตรนั้นถูกต้องจึงสามารถเข้าใช้งานได้ ซึ่งมีขั้นตอนดังนี้



รูปที่ 3.8 โฟล์วชาร์ทโปรแกรมส่วนผู้ใช้บริการขาเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 โฟล์วชาร์ทโปรแกรมส่วนผู้ใช้บริการขาออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 หลักการออกแบบฮาร์ดแวร์

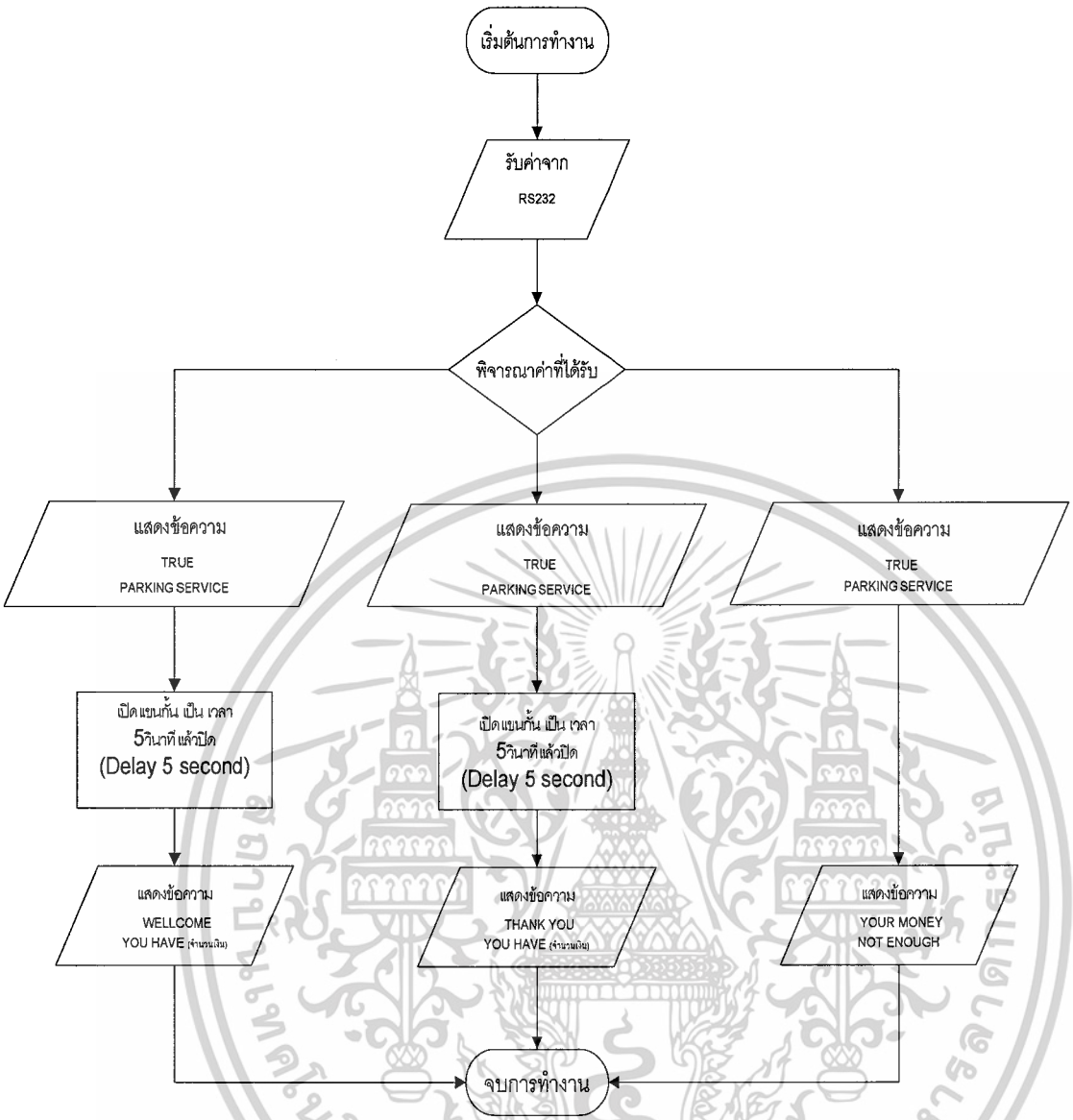
ในการออกแบบระบบที่จอดรถนี้เพื่อให้มีการแสดงการทดลอง โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุม โดยชุดควบคุมนี้ประกอบด้วย ส่วนแสดงผลแอลซีดี และส่วนแสดงการเปิดแขนกั้น โดยแทนแขนกั้นด้วยหลอดไฟแอลอีดี โดยอธิบายการทำงานได้ดังนี้

-หลอดไฟแอลอีดีสีแดง สว่าง แสดงว่า แขนกั้นยังไม่ยกขึ้นหรือแขนกั้นได้ยกลงที่เดิม

-หลอดไฟแอลอีดีสีเขียว สว่าง แสดงว่าแขนกั้นได้ยกขึ้น

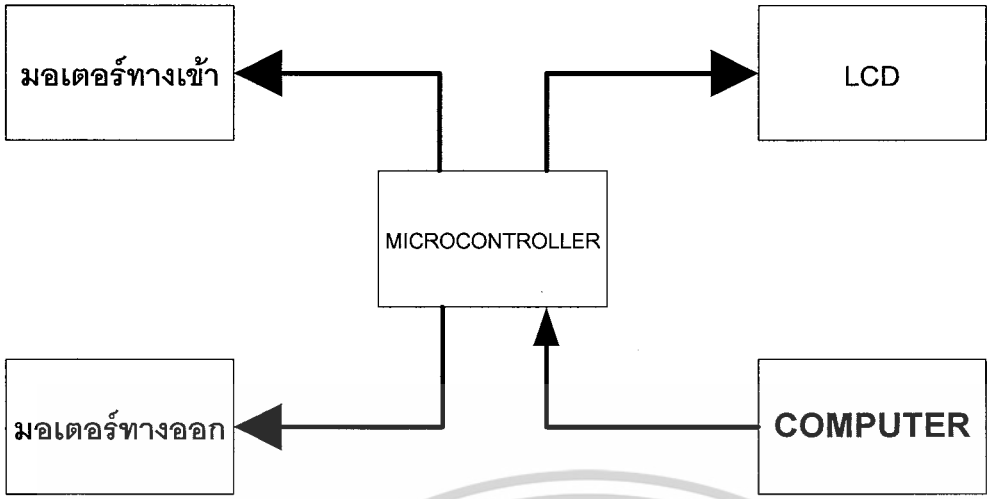
-เมื่อมีการทาบบัตรอาร์เอฟไอดีที่เป็นสมาชิก ทุกครั้งจะมีข้อความแสดงผลจากแอลซีดีทั้งขาเข้าและขาออกของรถที่เข้าใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

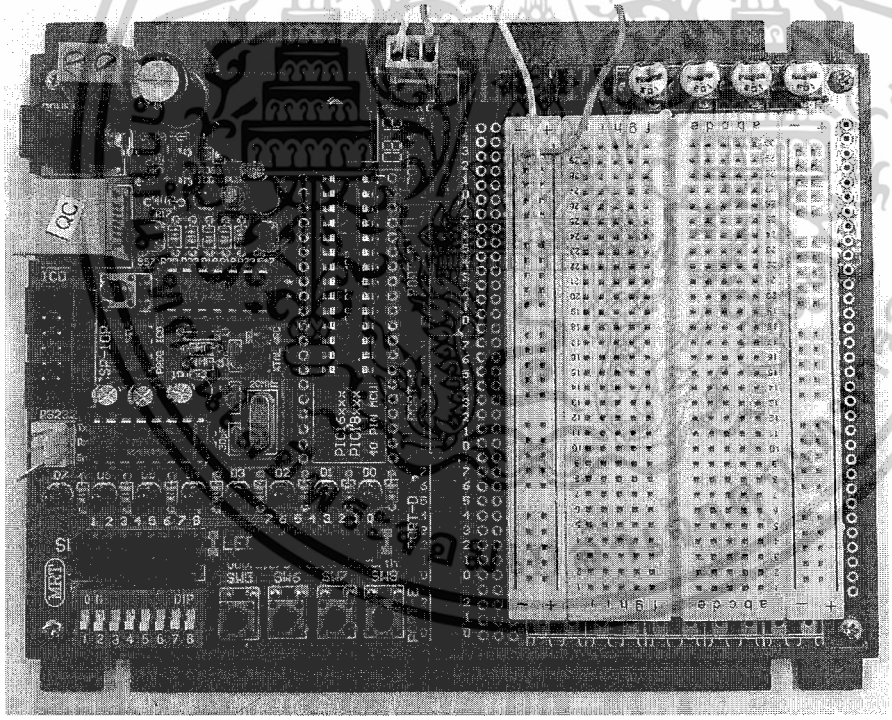


รูปที่ 3.10 โฟล์วชาร์ทการทำงานในส่วนควบคุมฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

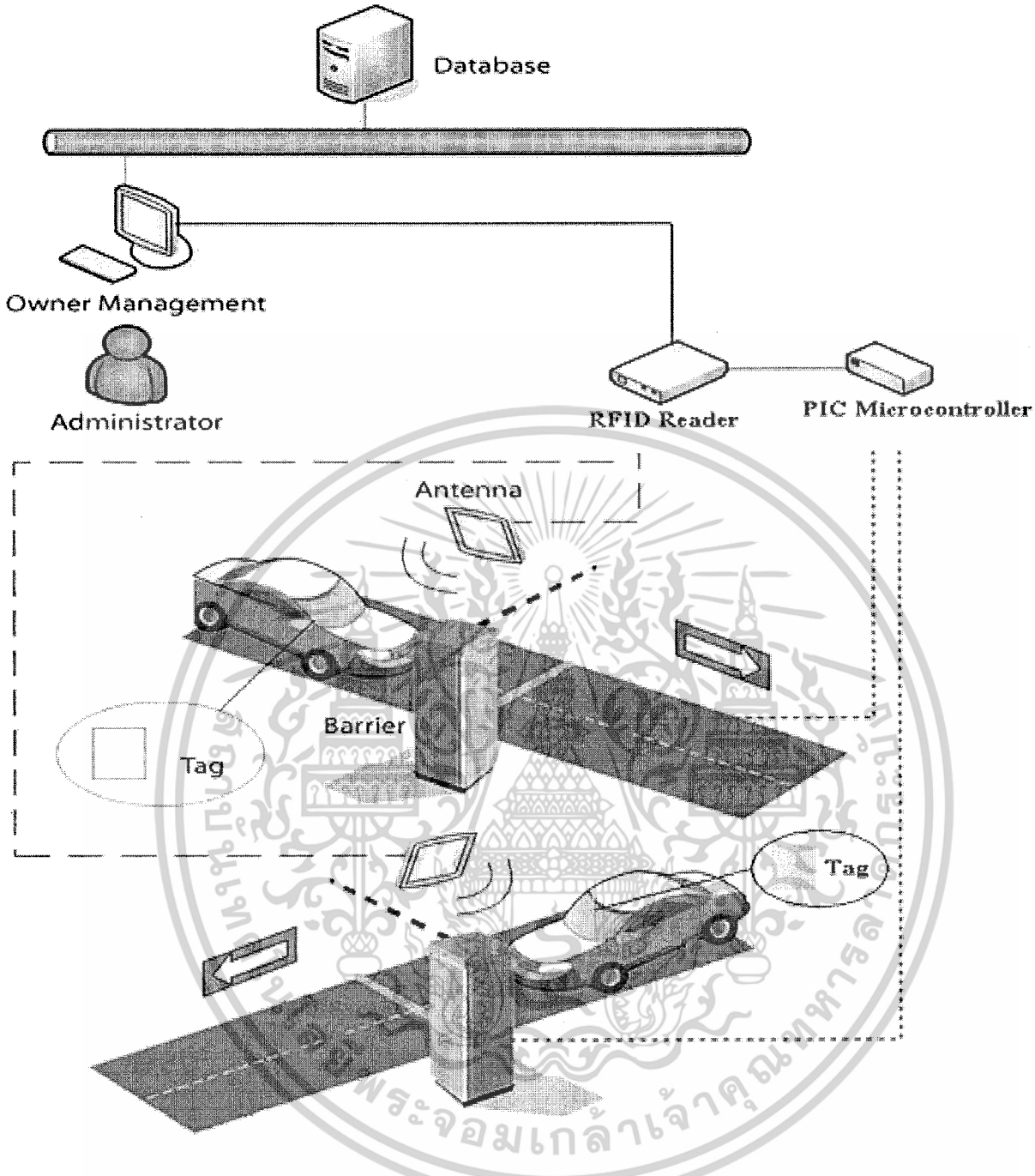


รูปที่ 3.11 แสดงการควบคุมของระบบทั้งหมด



รูปที่ 3.12 บอร์ด PIC Microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงการทำงานของระบบทั้งหมด

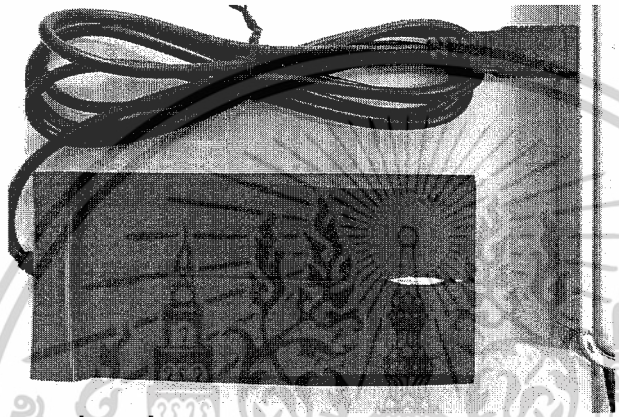
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

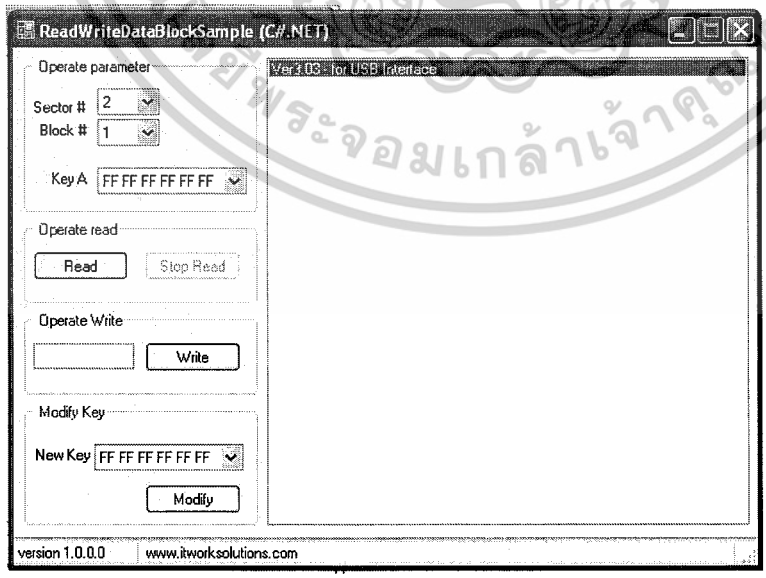
4.1 การทดลองอุปกรณ์ Tag และตัวอ่าน

เมื่อทำการเชื่อมต่อตัวอ่านเข้ากับคอมพิวเตอร์โดยเชื่อมต่อเข้าทาง Port USB ทางด้านหลังคอมพิวเตอร์ดังรูปที่ 4.1 และจะมีไฟสว่างขึ้นมาเพื่อแสดงว่าพร้อมที่จะทำงานแล้ว



รูปที่ 4.1 เชื่อมต่ออุปกรณ์ RFID เข้ากับคอมพิวเตอร์

ทำการเปิดโปรแกรม ReadWriteDataBlockSample ซึ่งเป็นโปรแกรมที่ใช้ในการทดสอบระบบ RFID โดยเราสามารถทำการเขียนและอ่าน TAG ได้โดยผ่าน RFID Reader และเรายังสามารถรู้ว่ามีข้อมูลอะไรอยู่ใน TAG บ้างและอยู่ในเฟรมใด บิตที่เท่าไหร่

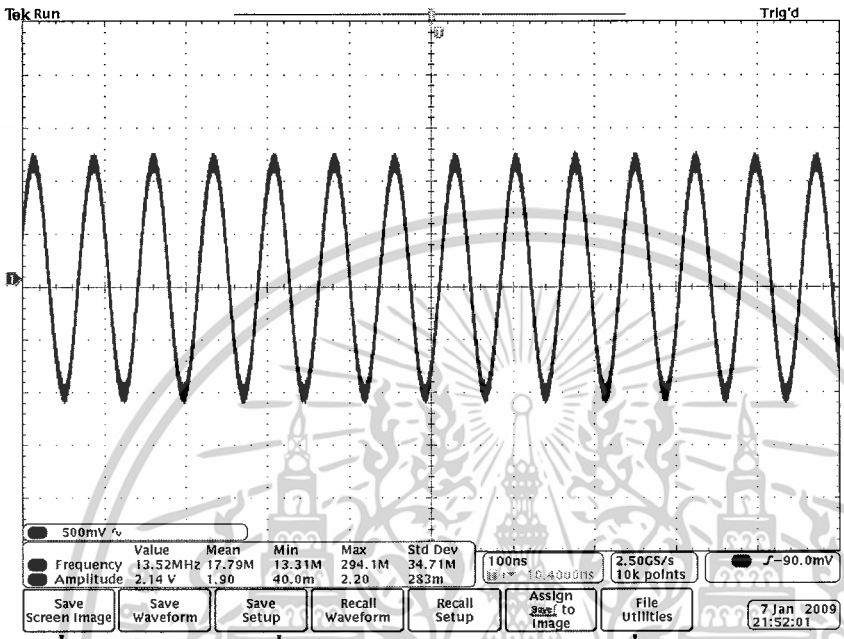


รูปที่ 4.2 โปรแกรม ReadWriteDataBlockSample

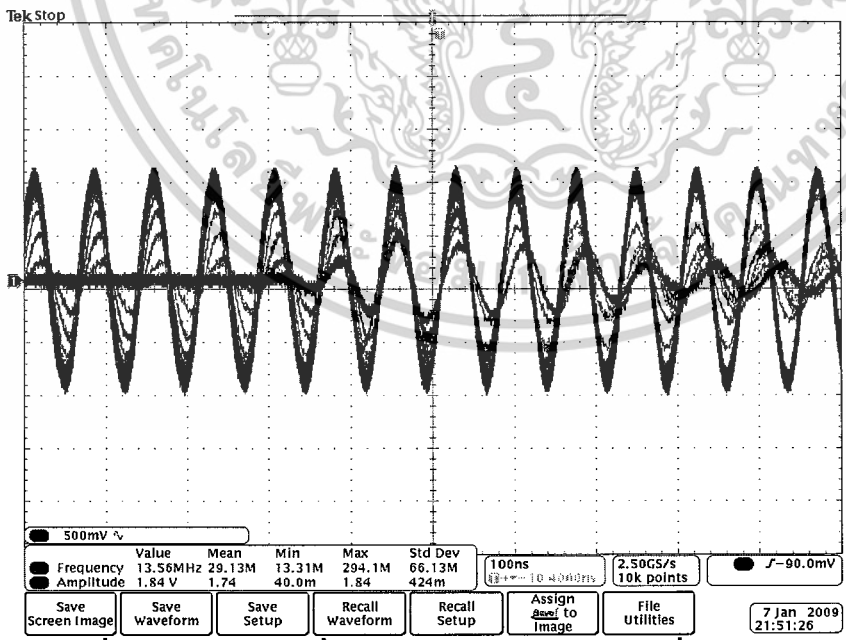
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้อน TAG ที่ถูกต้อง

เมื่อเราทำการนำ TAG มา Touch หน้าตัวอ่าน โดย TAG ที่เรานำมาใช้นั้นได้ใส่ค่า ณ ตำแหน่งบิตต่างๆเรียบร้อยแล้วโดยที่ Sector #2 & Block# 1 เรากำหนดให้เป็น Password ของผู้ใช้บริการโดยใช้จำนวน 8 บิต ใช้เป็นบัตรผ่านเข้าออกได้ ดังรูปที่ 4.3

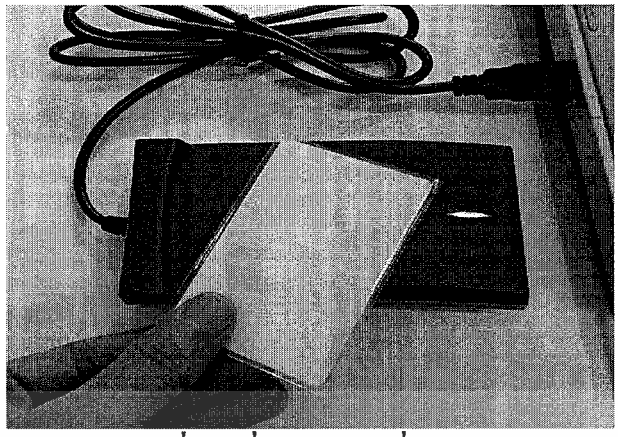


รูปที่ 4.3 แสดงสัญญาณที่ออกมาจาก RFID Reader ขณะที่ยังไม่ได้แตะ TAG

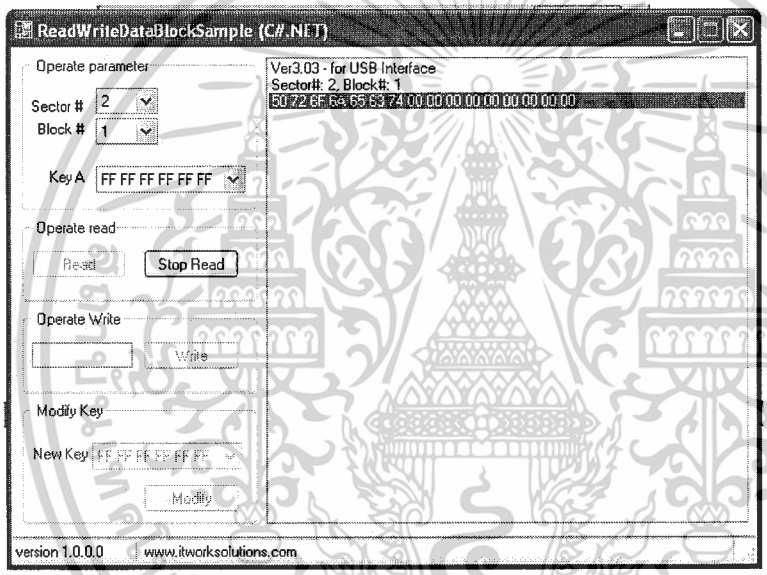


รูปที่ 4.4 แสดงสัญญาณที่ออกมาจาก RFID Reader ขณะที่แตะ TAG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 เมื่ออ่าน TAG ที่ถูกต้อง

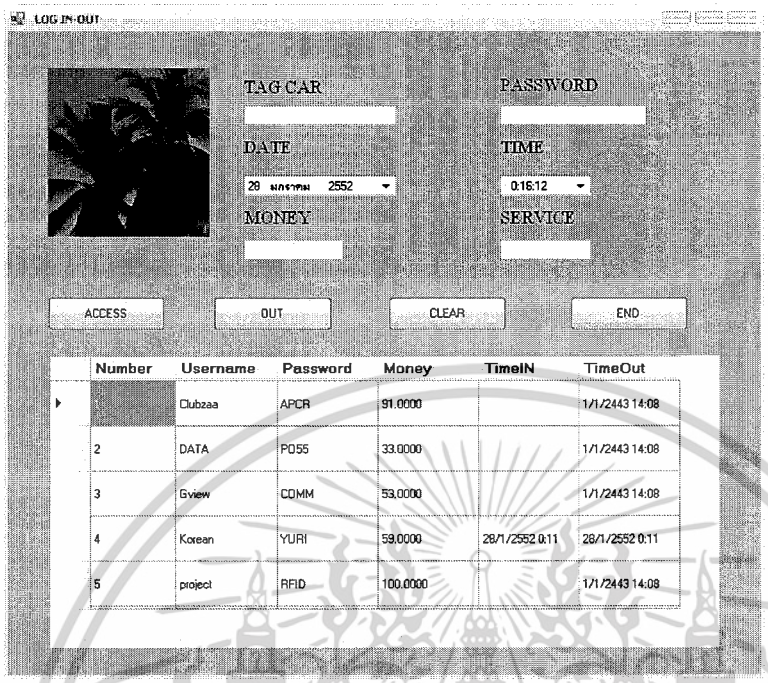


รูปที่ 4.6 ข้อมูลภายใน TAG เมื่อ TAG ถูกต้อง

จากรูปที่ 4.4 พบว่าใน Sector# 2 Block #1 จะมีค่า 50726F6A656374000000000000000000 ซึ่งเป็นค่าแสดง Password ของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ทดลองโปรแกรมข้อมูลผู้ถือบัตร



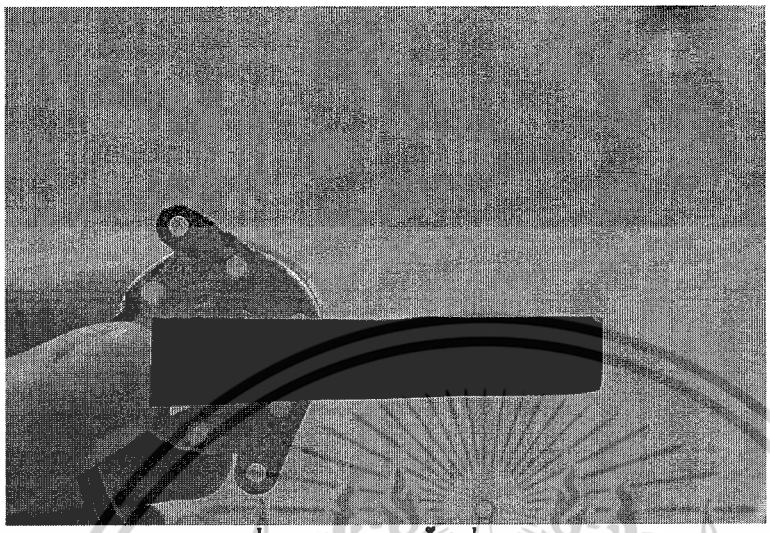
รูปที่ 4.7 หน้าต่างหลักโปรแกรมที่จอดรถ

รูปที่ 4.8 หน้าต่างแสดงผลเมื่อแตะ TAG ที่ตัวอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดลองสเต็ปมอเตอร์

เราจะใช้สเต็ปมอเตอร์ในการทำแขนกั้นโดยจะใช้ PIC Microcontroller เป็นตัวควบคุม

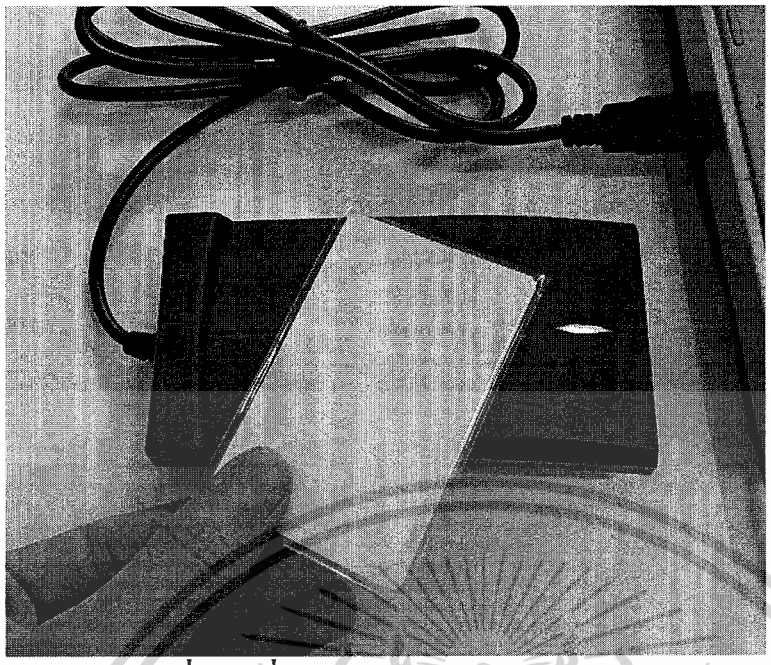


รูปที่ 4.9 แสดงแขนกั้นเมื่อปิดอยู่

จอ LCD จะแสดงข้อความดังนี้ เพื่อแสดงว่าอุปกรณ์พร้อมที่จะทำงาน

รูปที่ 4.10 แสดงข้อความพร้อมทำงาน

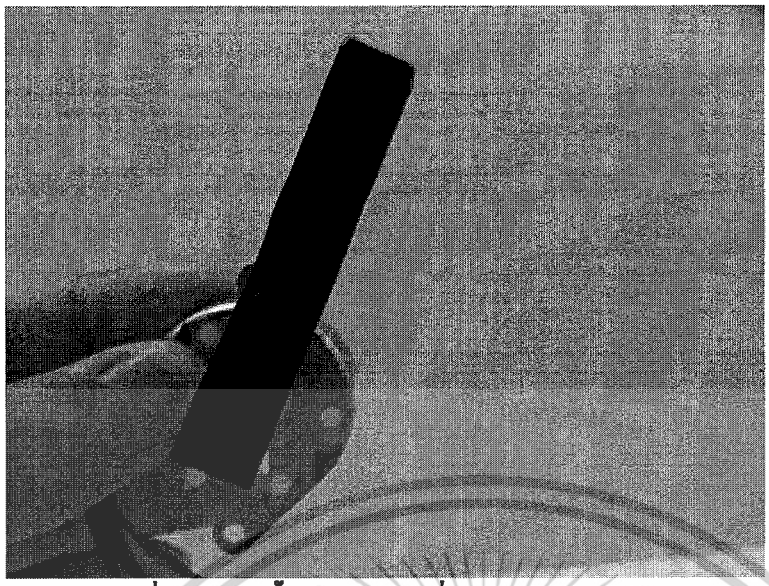
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 เมื่อนำ TAG มาแตะ RFID Reader

รูปที่ 4.12 แสดงข้อความหลังจากแตะบัตร RFID ที่ RFID Reader

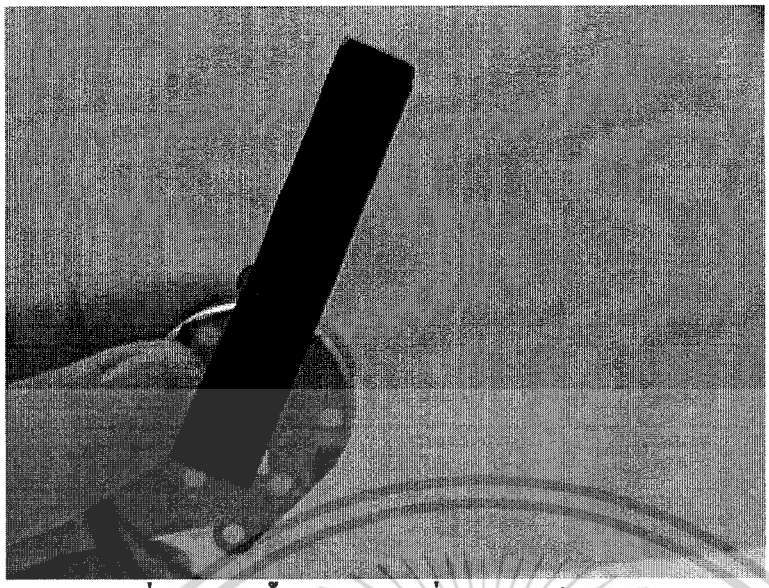
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 แขนกั้นจะเปิดออกเพื่อให้รถยนต์เข้าไปได้

รูปที่ 4.14 เมื่อนำ RFID มาแตะที่ RFID Reader เพื่อนำรถออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

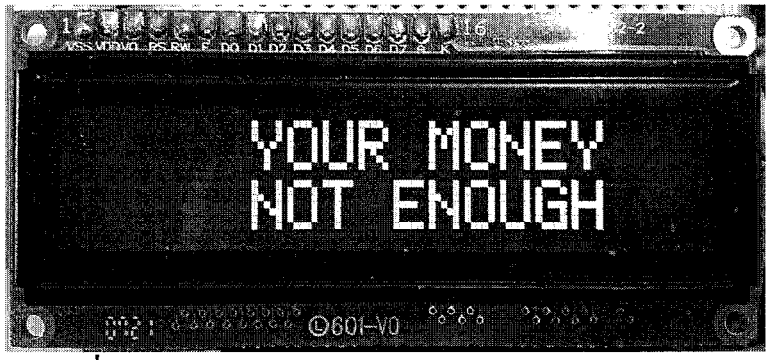


รูปที่ 4.15 แขนกั้นจะเปิดออกเพื่อให้รถยนต์ออกไปได้

รูปที่ 4.16 แสดงข้อความยอดเงินคงเหลือเมื่อรถกำลังออกไป

รูปที่ 4.17 แสดงข้อความเมื่อรถออกไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงข้อความจำนวนเงินในบัตร RFID มีไม่เพียงพอ

4.4 ผลการทดลอง

จากที่ได้ทำการทดลองการใช้โปรแกรมของระบบที่จอดรถโดยใช้ RFID แล้ว ซึ่งการทดลองใช้โปรแกรมนี้ประกอบไปด้วยขั้นตอนการทำงานของโปรแกรมระบบที่จอดรถและเครื่องอ่านข้อมูลที่ติดต่อกับเครื่องคอมพิวเตอร์ผ่านพอร์ต USB และขั้นตอนของการอ่านข้อมูลจาก TAG และการบันทึกข้อมูลโดยที่บัตรนั้นจะต้องมีหมายเลขประจำของแต่ละใบ รวมถึงข้อมูลต่างๆที่เป็นการยืนยันตัวตนของสมาชิกเพื่อเป็นเจ้าของบัตรเพื่อการใช้บริการของระบบที่จอดรถโดยใช้ RFID

ผลที่ได้จากการทดลองนี้ถือเป็นไปตามจุดประสงค์และขอบเขตของโครงงานที่ออกแบบไว้แล้วในขั้นตอนของการวางแผน และเพื่อให้โปรแกรมนี้มีความสมบูรณ์มากยิ่งขึ้นในการที่จะนำไปพัฒนาใช้งานจริงกับระบบที่จอดรถตามสถานที่ต่างๆ ยังจะต้องมีบางส่วนที่ต้องปรับปรุงให้การทำงานมีความเหมาะสมและมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
วิจารณ์และสรุป

5.1 สรุปผลการทดลอง

การทดลองเรื่องการรักษาความปลอดภัยที่จอดรถยนต์ด้วยเทคโนโลยี RFID เป็นเทคโนโลยีที่ใช้การติดต่อสื่อสารโดยใช้คลื่นความถี่วิทยุ (Radio frequency) โดยจะมีอุปกรณ์ที่สำคัญ 2 ชนิดคือ TAG และ เครื่องอ่าน ซึ่งเทคนิคในการรับส่งข้อมูลระหว่างเครื่องอ่านและ TAG จะใช้หลักการมอดูเลตทาง Amplitude และ Frequency ซึ่งนอกจากสายอากาศจะมีหน้าที่ในการรับส่งข้อมูลแล้วยังมีหน้าที่ในการเป็นแหล่งพลังงานให้กับ TAG ด้วย โดยตัวอ่านจะทำการส่งสัญญาณวิทยุออกเป็นจังหวะและรอคอยสัญญาณตอบจากตัว TAG ซึ่งเมื่อ TAG ได้รับสัญญาณคลื่นวิทยุที่ส่งมาจากเครื่องอ่าน ก็จะทำการเหนี่ยวนำเพื่อสร้างพลังงานให้กับ TAG โดยจะอ่านข้อมูลจากหน่วยความจำและเข้ารหัสข้อมูลแล้ว โดยข้อมูลที่ถูกมอดูเลตแล้วจะถูกส่งไปยังเครื่องอ่านผ่านสายอากาศโดยเครื่องอ่านจะถอดรหัสข้อมูลและส่งไปยังคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม

การทดลองในส่วนของโปรแกรมผู้ให้บริการนั้นจะทำการใช้โปรแกรม Microsoft Visual C# 2005 โดยทำการออกแบบรูปแบบตาม Flowchart ส่วนหน้าต่างหลักของโปรแกรมจะมีส่วนของ Access จะเป็นส่วนของผู้ให้บริการ โดยจะต้องมีการใส่รหัสผ่านเสียก่อนและตัวโปรแกรมจะเชื่อมโยงในส่วนของ Database ที่จัดเก็บโดยใช้โปรแกรม Microsoft SQL Server ดังนั้นจึงต้องทำการสร้างฐานข้อมูลของผู้ใช้ไว้ในส่วนของโปรแกรมนี้

5.2 ปัญหาและอุปสรรค

1. ศึกษาประเภทและการทำงานของเทคโนโลยี RFID ไม่เพียงพอทำให้การจัดหายืมอุปกรณ์ไม่ตรงตามวัตถุประสงค์ของการทำโครงงานครั้งนี้

2. เนื่องจากเครื่อง RFID Reader มีราคาที่ค่อนข้างสูงจึงทำให้งบประมาณในส่วนอุปกรณ์อื่นต้องลดคุณภาพลงไป

3. เนื่องจากมาตรฐานการเชื่อมต่อเครื่อง RFID Reader แต่ละรุ่นไม่เหมือนกันเวลานำไปใช้งานจริงถ้าต้องการเปลี่ยนเครื่องรุ่นใหม่ จำเป็นต้องเขียนโปรแกรมใหม่ซึ่งเสียเวลามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะและแนวทางแก้ปัญหา

1.ศึกษาและค้นคว้าข้อมูลเพิ่มเติมเกี่ยวกับเทคโนโลยีอาร์เอฟไอดี และการประยุกต์ใช้งานในด้านต่างๆให้มากขึ้น

2.ควรมีการทำระบบรักษาความปลอดภัยให้กับตัว TAG เพื่อป้องกันปัญหาที่อาจเกิดขึ้นเช่น การปลอมแปลง TAG

3.ศึกษาและค้นคว้าเกี่ยวกับการออกกแบบฐานข้อมูลให้มากขึ้นหรือปรึกษากับผู้ที่มีความเชี่ยวชาญด้านฐานข้อมูล เพื่อให้มีความถูกต้องสมบูรณ์ยิ่งขึ้น

4.ศึกษาระบบอาร์เอฟไอดี และการทำงานของอุปกรณ์แต่ละชิ้นเป็นอย่างดี เพื่อสามารถนำไปประยุกต์ใช้งานได้อย่างเต็มประสิทธิภาพ

5.ทำการทดลองใช้โปรแกรมตัวอย่างของระบบที่จอดรถโดยใช้เทคโนโลยีอาร์เอฟไอดีแล้วทำการปรับปรุงแก้ไขส่วนที่บกพร่อง อย่างเช่น แก้ไขปรับปรุงหน้าตาของโปรแกรมบางส่วนที่ใช้แสดงข้อมูลไม่ให้มากเกินความจำเป็นเวลาใช้งานจริง เป็นต้น เพื่อให้สามารถใช้งานโปรแกรมได้อย่างเต็มประสิทธิภาพ และช่วงของการเข้าใช้บริการให้มีการบันทึกข้อมูลอัตโนมัติโดยมีเจ้าหน้าที่เฉพาะในส่วนของทางออกเท่านั้น ซึ่งจะช่วยให้ลดเวลาในการเข้าใช้บริการได้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โค้ดที่ใช้ในการทดลอง

ส่วนของโปรแกรมการใช้งาน

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Data.SqlClient;
using System.Drawing;
using System.Diagnostics;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Runtime;
using System.IO;
using System.IO.Ports;
using System.Globalization;
using System.Security.Cryptography;
using System.Threading;
using Microsoft.VisualBasic;
using Itworks.RfidM1NET;

namespace Project_Final
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      SqlConnection Conn = new SqlConnection();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    SqlCommand comAdd;
    DataSet ds = new DataSet();
    StringBuilder sb = new StringBuilder();
    SqlDataAdapter da;
    SqlTransaction tr;

    bool IsFind = false;
    bool IsFindImage = false;

    string Username = "";
    string Password = "";

    int Price = 0;
    int read_value = 0;
    private bool _ReadFlag = false;
   SerialPort SRport = new SerialPort("COM5",9600,Parity.None,8, StopBits.One);
//ส่วนของ Form และ การเชื่อมต่อ database
    private void Form1_Load(object sender, EventArgs e)
    {
       Getconnect();
    }
    private void Getconnect()
    {
       string strConn = "Data Source=KIK\\SQLEXPRESS;Initial Catalog=Test;Integrated Security=True";
       Conn = new SqlConnection();
       if (Conn.State == ConnectionState.Open)
       {
          Conn.Close();
       }
       Conn.ConnectionString = strConn;
       Conn.Open();
       ClearAllData();
       ShowAllCustomer();
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//ส่วนของdavcustomer
    private void ClearAllData()
    {
       txtUsername.Text = "";
       txtPassword.Text = "";
    }


 private void ShowAllCustomer()
    {
       string sqlCustomer = "SELECT * FROM datajaaa";
       if (IsFind == true)
       {
          ds.Tables["Customer"].Clear();
       }
       SqlDataAdapter da = new SqlDataAdapter(sqlCustomer, Conn);
       da.Fill(ds, "Customer");
       if (ds.Tables["Customer"].Rows.Count != 0)
       {
          IsFind = true;
          dgvCustomer.ReadOnly = true;
          dgvCustomer.DataSource = ds.Tables["Customer"];
          FormatDataGridView();
          FormatAllRowsHeight();
       }
       else
       {
          IsFind = false;
       }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void FormatDataGridView()
    {
        DataGridViewCellStyle cs = new DataGridViewCellStyle();
        cs.Font = new Font("Ms Sans Serif", 10, FontStyle.Bold);

        dgvCustomer.ColumnHeadersDefaultCellStyle = cs;
        dgvCustomer.Columns[0].HeaderText = "Number";
        dgvCustomer.Columns[1].HeaderText = "Username";
        dgvCustomer.Columns[2].HeaderText = "Password";
        dgvCustomer.Columns[3].HeaderText = "Money";
        dgvCustomer.Columns[4].HeaderText = "TimeIN";
        dgvCustomer.Columns[5].HeaderText = "TimeOut";

        dgvCustomer.Columns[0].Width = 50;
        dgvCustomer.Columns[1].Width = 100;
        dgvCustomer.Columns[2].Width = 100;
        dgvCustomer.Columns[3].Width = 100;
        dgvCustomer.Columns[4].Width = 100;
        dgvCustomer.Columns[5].Width = 100;
        dgvCustomer.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.ColumnHeader;
        dgvCustomer.Rows[dgvCustomer.Rows.Count - 1].Height = 0;
    }


private void FormatAllRowsHeight()
    {
        for (int i = 0; i <= dgvCustomer.Rows.Count - 2; i++)
        {
            dgvCustomer.Rows[i].Height = 45;
        }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//ส่วนของlogin
    public void cmdAccess_Click(object sender, EventArgs e)
    {
      ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;
      _ReadFlag = true;
      while (_ReadFlag)
      {
        Application.DoEvents();
        int device_address = 0;
        byte[] atq = new byte[65];
        rfidm1_result = ISO14443A.MF_Request(device_address, RequestModeTypes.RequestIdle, atq);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
          continue;
        };
        Thread.Sleep(3);
        byte[] uid = new byte[2];
        byte collision = 0;
        rfidm1_result = ISO14443A.MF_Anticoll(device_address, uid, ref collision);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
          continue;
        };
        Thread.Sleep(5);
        rfidm1_result = ISO14443A.MF_Select(device_address, uid);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
          continue;
        };
        byte[] key = new byte[6] { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF }; //// key A = FF FF FF FF FF FF
        rfidm1_result = Mifare.MF_LoadKey(device_address, key);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
          continue;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        };
        byte sector_number = 0;
        byte block_number = 1;
        rfidm1_result = Mifare.MF_Auth(device_address, KeyModeTypes.KeyA, uid, (sector_number * 4) +
block_number);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
          this.Beep2(device_address);
          Thread.Sleep(3);
          ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
          Thread.Sleep(3);
          _ReadFlag = false;
          continue;
        };
        Thread.Sleep(2);
        byte[] buffer1 = new byte[16] { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00 };
        if (ResultCodeTypes.CommandOK == Mifare.MF_Read(device_address, (byte)((sector_number * 4) +
block_number), 1, buffer1))
        {
          txtPassword.Text = READTAG(buffer1);
          _ReadFlag = false;
          this.Beep1(device_address);
          byte sector_number2 = 0;
          byte block_number2 = 2;
          rfidm1_result = Mifare.MF_Auth(device_address, KeyModeTypes.KeyA, uid, (sector_number2 * 4)
+ block_number2);
          if (ResultCodeTypes.CommandOK != rfidm1_result)
          {
            this.Beep2(device_address);
            Thread.Sleep(3);
            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
            Thread.Sleep(3);
            return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            };
            Thread.Sleep(2);
            if (ResultCodeTypes.CommandOK == Mifare.MF_ReadValue(device_address,
(byte)((sector_number2 * 4) + block_number2), ref read_value))
            {
               txtmoney.Text = read_value.ToString();
               this.Beep1(device_address);
               Application.DoEvents();
               Thread.Sleep(3);
               ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
               Thread.Sleep(3);
            }
          };
        }
        if (txtUsername.Text.Trim() == "")
        {
           MessageBox.Show("กรุณากรอก Username ให้เรียบร้อยก่อน", "ข้อผิดพลาด",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
           txtUsername.Focus();
           return;
        }

        if (txtPassword.Text.Trim() == "")
        {
           MessageBox.Show("กรุณากรอก Password ให้เรียบร้อยก่อน", "ข้อผิดพลาด",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
           txtPassword.Focus();
           return;
        }
        CheckPassword();
        if (txtUsername.Text.Trim() != Username && txtPassword.Text.Trim() != Password)
        {
           long result;
           tr = Conn.BeginTransaction();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        try
        {
            string sqlpass;
            SqlCommand comAdd = new SqlCommand();
            sqlpass = "INSERT INTO datajaaa(Username,Password,TimeIn,Money)";
            sqlpass += " VALUES (@Username, @Password, @TimeIn, @Money)";
            comAdd.CommandText = sqlpass;
            comAdd.CommandType = CommandType.Text;
            comAdd.Connection = Conn;
            comAdd.Transaction = tr;
            comAdd.Parameters.Clear();
            comAdd.Parameters.Add("@Username", SqlDbType.NVarChar).Value = txtUsername.Text.Trim();
            comAdd.Parameters.Add("@Password", SqlDbType.NVarChar).Value = txtPassword.Text.Trim();
            comAdd.Parameters.Add("@TimeIn", SqlDbType.DateTime).Value = dtpTime2.Text;
            comAdd.Parameters.Add("@Money", SqlDbType.Money).Value = txtmoney.Text;
             result = comAdd.ExecuteNonQuery();
             tr.Commit();
             MessageBox.Show(" เรียบร้อยแล้ว ผ่านได้ !!!", "ผลการทำงาน", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            ShowAllCustomer();
        }
        catch
        {
            MessageBox.Show("ไม่ผ่านนะ !!!", "ผลการทำงาน", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
            tr.Rollback();
            return;
        }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (txtUsername.Text.Trim() == Username && txtPassword.Text.Trim() == Password)
    {
        long result;
        tr = Conn.BeginTransaction();
        try
        {
            string sqlpass;
            SqlCommand comAdd = new SqlCommand();
            sqlpass = "UPDATE datajaaa";
            sqlpass += " SET TimeIn=@TimeIn,";
            sqlpass += "Money=@Money";
            sqlpass += " WHERE (Username=@Username)";
            comAdd.CommandText = sqlpass;
            comAdd.CommandType = CommandType.Text;
            comAdd.Connection = Conn;
            comAdd.Transaction = tr;
            comAdd.Parameters.Clear();
            comAdd.Parameters.Add("@Username", SqlDbType.VarChar).Value = txtUsername.Text.Trim();
            comAdd.Parameters.Add("@TimeIn", SqlDbType.DateTime).Value = dtpTime2.Text;
            comAdd.Parameters.Add("@Money", SqlDbType.NVarChar).Value = txtmoney.Text;
            result = comAdd.ExecuteNonQuery();
            tr.Commit();
            MessageBox.Show(" เรียบร้อยแล้ว ผ่านได้ !!!", "ผลการทำงาน", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            ShowAllCustomer();
        }
        catch
        {
            MessageBox.Show("ไม่ผ่านนะ !!!", "ผลการทำงาน", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
            tr.Rollback();
            return;
        }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        string newnumber;
        int value;
        value = 1000 + read_value;
        newnumber = value.ToString();
        SentCommandP(newnumber);
    }


//ส่วนของlogout
    public void cmdLogout_Click(object sender, EventArgs e)
    {
        ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;
        _ReadFlag = true;
        while (_ReadFlag)
        {
            Application.DoEvents();
            int device_address = 0;
            byte[] atq = new byte[65];
            rfidm1_result = ISO14443A.MF_Request(device_address, RequestModeTypes.RequestIdle, atq);
            if (ResultCodeTypes.CommandOK != rfidm1_result)
            {
                continue;
            };
            Thread.Sleep(3);
            byte[] uid = new byte[2];
            byte collision = 0;
            rfidm1_result = ISO14443A.MF_Anticoll(device_address, uid, ref collision);
            if (ResultCodeTypes.CommandOK != rfidm1_result)
            {
                continue;
            };
            Thread.Sleep(5);
            rfidm1_result = ISO14443A.MF_Select(device_address, uid);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
            continue;
        };
        byte[] key = new byte[6] { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF }; //// key A = FF FF FF FF FF FF
        rfidm1_result = Mifare.MF_LoadKey(device_address, key);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
            continue;
        };
        byte sector_number = 0;
        byte block_number = 1;
        rfidm1_result = Mifare.MF_Auth(device_address, KeyModeTypes.KeyA, uid, (sector_number * 4) +
block_number);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
            this.Beep2(device_address);
            Thread.Sleep(3);
            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
            Thread.Sleep(3);
            _ReadFlag = false;
            continue;
        };
        Thread.Sleep(2);
        byte[] buffer1 = new byte[16] { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00 };
        if (ResultCodeTypes.CommandOK == Mifare.MF_Read(device_address, (byte)((sector_number * 4) +
block_number), 1, buffer1))
        {
            txtPassword.Text = READTAG(buffer1);
            _ReadFlag = false;
            this.Beep1(device_address);
            Application.DoEvents();
            Thread.Sleep(3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
            Thread.Sleep(3);
        };
        Application.DoEvents();
    };


if (txtUsername.Text.Trim() == "" && txtPassword.Text.Trim() == "")
    {
        MessageBox.Show("กรุณากรอก Username และ Password ให้เรียบร้อย", "ข้อผิดพลาด",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtUsername.Focus();
    }
    else
    {
        CheckPassword();
        if (txtUsername.Text.Trim() == Username && txtPassword.Text.Trim() == Password)
        {
            long result;
            try
            {
                string sqlpass;
                SqlCommand comAdd = new SqlCommand();
                sqlpass = "UPDATE datajaaa";
                sqlpass += " SET TimeOut=@TimeOut";
                sqlpass += " WHERE (Username=@Username)";
                comAdd.CommandText = sqlpass;
                comAdd.CommandType = CommandType.Text;
                comAdd.Connection = Conn;
                comAdd.Parameters.Add("@Username", SqlDbType.VarChar).Value = txtUsername.Text;
                comAdd.Parameters.Add("@TimeOut", SqlDbType.DateTime).Value = dtpTime2.Text;
                result = comAdd.ExecuteNonQuery();
                ShowAllCustomer();
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            catch
            {
               MessageBox.Show("Username หรือ Password ที่คุณป้อน ไม่ถูกต้อง !!!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
               txtUsername.Text = "";
               txtUsername.Focus();
               txtPassword.Text = "";
               return;
            }
         }
         else
         {
            MessageBox.Show("กรุณาใส่ Username และ Password ให้ถูกต้อง !!!", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            return;
         }
         Status_Money();
         string newnumber;
         int value;
         value = 5000 + read_value;
         newnumber = value.ToString();
        SentCommandP(newnumber) ;
      }
   }


   private void cmdclear_Click(object sender, EventArgs e)
   {
      txtUsername.Clear();
      txtPassword.Clear();
      txtmoney.Clear();
      txtPrice.Clear();
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void cmdEnd_Click(object sender, EventArgs e)
    {
      Application.Exit();
    }


private void CheckPassword()
    {
      try
      {
        StringBuilder sb = new StringBuilder();
        sb.Remove(0, sb.Length);
        sb.Append("SELECT Username,Password");
        sb.Append(" FROM datajaaa");
        sb.Append(" WHERE (Username=@Username)");
        string sqlpass = sb.ToString();
        SqlCommand comAdd = new SqlCommand();
        SqlDataReader dr;
        comAdd.CommandType = CommandType.Text;
        comAdd.CommandText = sqlpass;
        comAdd.Connection = Conn;
        comAdd.Parameters.Clear();
        comAdd.Parameters.Add("@Username", SqlDbType.NVarChar).Value = txtUsername.Text.Trim();
        dr = comAdd.ExecuteReader();
        if (dr.HasRows)
        {
          dr.Read();
          Username = dr.GetString(dr.GetOrdinal("Username"));
          Password = dr.GetString(dr.GetOrdinal("Password"));
        }
        else
        {
          MessageBox.Show("ไม่พบข้อมูลที่ท่านต้องการ", "ผลการค้นหา", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            txtUsername.Focus();
            txtUsername.SelectAll();
        }
        dr.Close();
      }
      catch
      {
        MessageBox.Show("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูล", "ผลการค้นหา",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
      }
    }
    public void Culculate_Time()
    {
     DateTime TimeIn2 = new DateTime() ;
     DateTime TimeIn3 = new DateTime();
     try
      {
        StringBuilder sb = new StringBuilder();
        sb.Remove(0, sb.Length);
        sb.Append(" SELECT TimeIn,TimeOut");
        sb.Append(" FROM datajaaa");
        sb.Append(" WHERE (Password=@Password)");
        string sqlpass = sb.ToString();
        SqlCommand comAdd = new SqlCommand();
        SqlDataReader dr;
        comAdd.CommandType = CommandType.Text;
        comAdd.CommandText = sqlpass;
        comAdd.Connection = Conn;
        comAdd.Parameters.Clear();
        comAdd.Parameters.Add("@Password", SqlDbType.VarChar).Value = txtPassword.Text.Trim();
        dr = comAdd.ExecuteReader();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (dr.HasRows)
            {
                dr.Read();
                TimeIn2 = dr.GetDateTime(dr.GetOrdinal("TimeIn"));
                TimeIn3 = dr.GetDateTime(dr.GetOrdinal("TimeOut"));
                dr.Close();
            }
            else
            {
                MessageBox.Show("ไม่พบในฐานข้อมูล !!!", "ผลการค้นหา", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                return;
            }
        }
            catch
            {
                MessageBox.Show("เกิดข้อผิดพลาดระหว่างอัพข้อมูล", "ผลการทำงาน", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
            }
            TimeSpan ts = TimeIn3 - TimeIn2;
            string cost = ts.ToString();
            string[] sp = cost.Split(':');
            if (sp[0].ToString() == "00")
            {
                txtPrice.Text = "20";
                Price = 20;
            }
            else if (sp[0].ToString() == "01")
            {
                txtPrice.Text = "25";
                Price = 25;
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else if (sp[0].ToString() == "02")
        {
            txtPrice.Text = "30";
            Price = 30;
        }
        else if (sp[0].ToString() == "03")
        {
            txtPrice.Text = "35";
            Price = 35;
        }
        else if (sp[0].ToString() == "04")
        {
            txtPrice.Text = "40";
            Price = 40;
        }
        else
        {
            txtPrice.Text = "10";
            Price = 10;
        }
        Down_money();
    }

private void dgvCustomer_CellMouseUp(object sender, DataGridViewCellMouseEventArgs e)
{
    if (e.RowIndex == dgvCustomer.Rows.Count - 1)
    {
        Username = "";
        return;
    }
    txtUsername.Text = dgvCustomer.Rows[e.RowIndex].Cells[0].Value.ToString();
    txtPassword.Text = dgvCustomer.Rows[e.RowIndex].Cells[1].Value.ToString();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//ส่วนของเสียง
private void Beep1(int deviceAddress)
{
   byte[] sound = new byte[5] { 0x01, 0x01, 0x00, 0x00, 0x01 };
   ConfigurationAndIO.ActiveBuzzer(deviceAddress, BuzzerModeTypes.SoundPattern, sound);
}

private void Beep2(int deviceAddress)
{
   byte[] sound = new byte[5] { 0x01, 0x01, 0x01, 0x01, 0x01 };
   ConfigurationAndIO.ActiveBuzzer(deviceAddress, BuzzerModeTypes.SoundPattern, sound);
}

//ส่วนของอ่านTAG
private string READTAG(byte[] buffer)
{
   byte[] B = new byte[8];
   for (int i = 0; i < 8; i++)
   {
      B[i] = buffer[i];
   }
   string s = System.Text.ASCIIEncoding.ASCII.GetString(B);
   return s;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//ส่วนของส่งคำสั่งไปไมโครคอนโทรลเลอร์
    public void SentCommandP(string number)
    {
        SRport.Open();
        SRport.ReadTimeout = 500;
        byte [] tt = new byte[4] {0x00,0x00,0x00,0x00};
        System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();
        tt = encoding.GetBytes(number);
        SRport.Write(tt, 0, 4);
        SRport.Close();

    }


//ส่วนของเช็คยอดเงินและการหักเงิน
    public void Status_Money()
    {
        ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;
        int device_address = 0;
        byte[] atq = new byte[65];
        rfidm1_result = ISO14443A.MF_Request(device_address, RequestModeTypes.RequestAll, atq);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
            return;
        };
        Thread.Sleep(3);
        byte[] uid = new byte[2];
        byte collision = 0;
        rfidm1_result = ISO14443A.MF_Anticoll(device_address, uid, ref collision);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
            return;
        };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Thread.Sleep(5);
        rfidm1_result = ISO14443A.MF_Select(device_address, uid);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
           return;
        };
        byte[] key = new byte[6] { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF }; //// key A = FF FF FF FF FF FF
        rfidm1_result = Mifare.MF_LoadKey(device_address, key);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
           return;
        };
        byte sector_number = 0;
        byte block_number = 2;
        if (block_number == 3)
        {
           this.Beep2(device_address);
           Thread.Sleep(3);
           ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
           Thread.Sleep(3);
           return;
        }
        rfidm1_result = Mifare.MF_Auth(device_address, KeyModeTypes.KeyA, uid, (sector_number * 4) +
block_number);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
           this.Beep2(device_address);
           Thread.Sleep(3);
           ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
           Thread.Sleep(3);
           return;
        };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Thread.Sleep(2);
        int read_value = 0;
        if (ResultCodeTypes.CommandOK == Mifare.MF_ReadValue(device_address, (byte)((sector_number *
4) + block_number), ref read_value))
        {
           txtmoney.Text = read_value.ToString();
           Application.DoEvents();
           Thread.Sleep(3);
           ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
           Thread.Sleep(3);
        }
       int Bath = read_value;
        if (Bath >= 10)
        {
           MessageBox.Show("ขอบคุณที่มาใช้บริการค่ะ ", "Service", MessageBoxButtons.OK,
MessageBoxIcon.None);
           Culculate_Time();
        }
        else
        {
           MessageBox.Show("ยอดเงินของคุณหมดไม่สามารถให้บริการได้ กรุณาเติมเงิน", "Service",
MessageBoxButtons.OKCancel, MessageBoxIcon.None);
           SentCommandP("9999");
           Up_money();
        }
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void Up_money()
    {
      ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;
      int device_address = 0;
      byte[] atq = new byte[65];
      rfidm1_result = ISO14443A.MF_Request(device_address, RequestModeTypes.RequestAll, atq);
      if (ResultCodeTypes.CommandOK != rfidm1_result)
      {
        return;
      };
      Thread.Sleep(3);
      byte[] uid = new byte[2];
      byte collision = 0;
      rfidm1_result = ISO14443A.MF_Anticoll(device_address, uid, ref collision);
      if (ResultCodeTypes.CommandOK != rfidm1_result)
      {
        return;
      };
      Thread.Sleep(5);
      rfidm1_result = ISO14443A.MF_Select(device_address, uid);
      if (ResultCodeTypes.CommandOK != rfidm1_result)
      {
        return;
      };
      byte[] key = new byte[6] { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };  //// key A = FF FF FF FF FF FF
      rfidm1_result = Mifare.MF_LoadKey(device_address, key);
      if (ResultCodeTypes.CommandOK != rfidm1_result)
      {
        return;
      };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        byte sector_number = 0;
        byte block_number = 2;
        if (block_number == 3)
        {
           this.Beep2(device_address);
           Thread.Sleep(3);
           ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
           Thread.Sleep(3);
           return;
        }
        rfidm1_result = Mifare.MF_Auth(device_address, KeyModeTypes.KeyA, uid, (sector_number * 4) +
block_number);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
           this.Beep2(device_address);
           Thread.Sleep(3);
           ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
           Thread.Sleep(3);
           return;
        };
        Thread.Sleep(2);
        int increment_value;
        try
        {
           increment_value = Convert.ToInt32(
              Interaction.InputBox(
              "Enter value",
              "Refill",
              "99",
              this.Location.X + 105,
              this.Location.Y + 215));
        }
        catch
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            increment_value = 0;
        }
        if (ResultCodeTypes.CommandOK == Mifare.MF_Increment(device_address, (byte)((sector_number * 4)
+ block_number), increment_value))
        {
            Mifare.MF_Transfer(device_address, (byte)((sector_number * 4) + block_number));
            txtmoney.Text = increment_value.ToString();
            this.Beep1(device_address);
            DialogResult respone = MessageBox.Show("ท่านต้องการดำเนินรายการต่อไปหรือไม่", "",
MessageBoxButtons.YesNo);
            if (respone == DialogResult.Yes)
            {
               Culculate_Time();
            }
            else
            {
               MessageBox.Show("ขอบคุณที่ใช้บริการ เดินทางโดยสวัสดิภาพค่ะ");
            }
            Application.DoEvents();
            Thread.Sleep(3);
            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
            Thread.Sleep(3);
        }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void Down_money()
    {
        ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;
        int device_address = 0;
        byte[] atq = new byte[65];
        rfidm1_result = ISO14443A.MF_Request(device_address, RequestModeTypes.RequestAll, atq);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
           return;
        };
        Thread.Sleep(3);
        byte[] uid = new byte[2];
        byte collision = 0;
        rfidm1_result = ISO14443A.MF_Anticoll(device_address, uid, ref collision);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
           return;
        };
        Thread.Sleep(5);
        rfidm1_result = ISO14443A.MF_Select(device_address, uid);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
           return;
        };
        byte[] key = new byte[6] { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };  //// key A = FF FF FF FF FF FF
        rfidm1_result = Mifare.MF_LoadKey(device_address, key);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
           return;
        };
        byte sector_number = 0;
        byte block_number = 2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (block_number == 3)
        {
            this.Beep2(device_address);
            Thread.Sleep(3);
            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
            Thread.Sleep(3);
            return;
        }
        rfidm1_result = Mifare.MF_Auth(device_address, KeyModeTypes.KeyA, uid, (sector_number * 4) +
block_number);
        if (ResultCodeTypes.CommandOK != rfidm1_result)
        {
            this.Beep2(device_address);
            Thread.Sleep(3);
            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
            Thread.Sleep(3);
            return;
        };
        Thread.Sleep(2);
        int decrement_value;
        try
        {
            decrement_value = Price;
        }
        catch
        {
            decrement_value = 0;
        }
        if (ResultCodeTypes.CommandOK == Mifare.MF_Decrement(device_address, (byte)((sector_number *
4) + block_number), decrement_value))
        {
            Mifare.MF_Transfer(device_address, (byte)((sector_number * 4) + block_number));
            this.Beep1(device_address);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            byte sector_number2 = 0;
            byte block_number2 = 2;
            rfidm1_result = Mifare.MF_Auth(device_address, KeyModeTypes.KeyA, uid, (sector_number2 * 4) +
block_number2);
            if (ResultCodeTypes.CommandOK != rfidm1_result)
            {
               this.Beep2(device_address);
               Thread.Sleep(3);
               ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
               Thread.Sleep(3);
               return;
            };
            Thread.Sleep(2);
            if (ResultCodeTypes.CommandOK == Mifare.MF_ReadValue(device_address,
(byte)((sector_number2 * 4) + block_number2), ref read_value))
            {
               txtmoney.Text = read_value.ToString();
               this.Beep1(device_address);
               Application.DoEvents();
               Thread.Sleep(3);
               ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
               Thread.Sleep(3);
            }
         }
      }
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของไมโครคอนโทรลเลอร์

```
#include <16f877a.h>
#fuses HS,nowdt,noprotect,nolvp
#use delay(clock=20000000)
#use rs232(baud=9600,xmit=PIN_C6, rcv=PIN_C7)
#include <lcd.c>          //for driver LCD modules
#include <stdlib.h>
#define step_output(x) output_c(x)
int fslp[4]={0x01,0x02,0x04,0x08};
void delay_second(int i);
void delay_1ms(int16 i);
void fw_step(int16 step,int dl);
void rw_step(int16 step,int dl);

void delay_second(int i)
{
for(;i>0;i--){
delay_ms(1000);
}
}
void delay_1ms(int16 i)
{
if(i<50)
i=50;
for(;i>0;i--){
delay_ms(1);
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void fw_step(int16 step,int dl)
{
int n =0;
                    //driver motor forwad
while(step--) {
step_output(fslp[n]);
if(n++ == 3)
n=0;
delay_1ms(dl);
}
}

void rw_step(int16 step,int dl)
{
int n = 3;              //drive motor reverse
while(step--) {
step_output(fslp[n]);
if(n-- == 0)
n = 3;
delay_1ms(dl);
}
}

void main()
{
  int8 i;
  char buffer[4];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(i=0;i<5;i++)
{
  output_low(PIN_C0);
  delay_ms(200);
  output_high(PIN_C0);
  delay_ms(200);
}
lcd_init();


lcd_putc("\f     TRUE   ");
lcd_putc("\n PARKING SERVICE ");
printf("telecom");
output_high(PIN_B6);

delay_ms(1500);
while(TRUE)
{
  for(i=0;i<4;i++)
    buffer[i]=getc();


  if(buffer[0]=='1')
  {
    output_low(PIN_E0);
    lcd_putc("\f    WELLCOME    ");
    lcd_putc("\n YOUR HAVE ");
    for(i=1;i<4;i++)
      lcd_putc(buffer[i]);
    output_high(PIN_E1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fw_step(10,50);
    delay_ms(3000);
    rw_step(10,50);
   lcd_putc("\f     TRUE   ");
    lcd_putc("\n PARKING SERVICE ");
    output_high(PIN_E0);
    output_low(PIN_E1);
}
else if(buffer[0]=='5')
{
    output_low(PIN_B6);
    lcd_putc("\f   THANK YOU    ");
    lcd_putc("\nYOUR HAVE ");
    for(i=1;i<4;i++)
      lcd_putc(buffer[i]);
    output_high(PIN_A3);
    fw_step(10,50);
    delay_ms(3000);
    rw_step(10,50);
    lcd_putc("\f     TRUE   ");
    lcd_putc("\n PARKING SERVICE ");
    output_high(PIN_B6);
    output_low(PIN_B7);
}
else if(buffer[0]=='9')
{
     output_high(PIN_B6);
    lcd_putc("\f    YOUR MONEY  ");
    lcd_putc("\n    NOT ENOUGH   ");
    output_high(PIN_B3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. สัจจะ จรัสรุ่งรวีวร, **"คู่มือ Visual C# 2005 ฉบับสมบูรณ์"** ไอดีซีฯ, กรุงเทพฯ 2550

2. ศุภชัย สมพานิช, **"Database Programming ด้วย VB2005 & VC# 2005 ฉบับสมบูรณ์"** ไอดีซีฯ, กรุงเทพฯ 2550

3.ศุภชัย สมพานิช, **"พัฒนาระบบฐานขอมูลด้วย VB 2005&VC# 2005"** ไอดีซีฯ, กรุงเทพฯ 2550

4. มณีโชติ สมานไทย, **"คู่มือการออกแบบฐานข้อมูลและภาษา SQL ฉบับผู้เริ่มต้น"** ไอดีซีฯ, กรุงเทพฯ 2550

5. ประจิน พลังสันติกุล, **"PIC Works"** แอพซอฟต์เทค, กรุงเทพ ฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้