สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมประตูเข้าออกของบริการรถไฟฟ้าผ่านอาร์เอฟไอดี

THE ENTRANCE SYSTEM OF BTS SERVICE VIA RFID

T103011

โดย

นางสาวภัททิรา อ่ำใหญ่

นางสาวรัชดาวรรณ สักกพันธุ์นิกร

นายอรรถพร สิงห์คีรี

เลขหมู่.....
เลขทะเบียน 103011
วัน,เดือน,ปี 24 ส.ค. 2552

b 12091108
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมประตูเข้าออกของบริการรถไฟฟ้าผ่านอาร์เอฟไอดี

THE ENTRANCE SYSTEM OF BTS SERVICE VIA RFID

โดย

นางสาวภัททิรา	อ่ำใหญ่	รหัส 48010655
นางสาวรัชดาวรรณ	สักกพันธุ์นิกร	รหัส 48010729
นายอรรถพร	สิงห์คีรี	รหัส 48011083

อาจารย์ที่ปรึกษา

รศ.ดร. ไกรสิน ส่งวัฒนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

ภาควิชา
วิศวกรรมโทรคมนาคม
ผ่านการตรวจรูปเล่มแล้ว
(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2551

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมประตูเข้าออกของบริการรถไฟฟ้าผ่านอาร์เอฟไอดี

The Entrance System of BTS Service via RFID

ผู้จัดทำ

1. นางสาวภัททิรา อ่ำใหญ่ 48010655
2. นางสาวรัชดาวรรณ สักกพัธุ์นิกร 48010729
3. นายอรรถพร สิงห์คีรี 48011083

อาจารย์ที่ปรึกษา

(รศ.ดร.ไกรสิน ส่งวัฒนา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ประโยชน์ที่ได้รับจากโครงงาน	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ส่วนประกอบของระบบ RFID	3
2.1.1 ความถี่ของคลื่นพาหะ	4
2.1.2 ลักษณะของอุปกรณ์ RFID	5
2.1.3 การรับส่งข้อมูลแบบ FDX / HDX	5
2.1.4 การรับส่งข้อมูลแบบ SEQ	6
2.1.5 Tag	6
2.1.5.1 เครื่องลูกข่ายชนิดแอกทีฟ	7
2.1.5.2 เครื่องลูกข่ายชนิดพาสซีฟ	7
2.1.6 Reader	10
2.1.7 การสื่อสารแบบไร้สาย	11
2.1.8 รูปแบบการรับส่งข้อมูล	12
2.1.9 เทคโนโลยี RFID ที่ใช้การสื่อสารแบบ Duplexing	13
2.1.9.1 เทคโนโลยี Inductive Coupling	13
2.1.9.2 เทคโนโลยี Load Modulation	15
2.1.9.3 เทคโนโลยี Load Modulation with Sub carrier	16
2.1.9.4 เทคโนโลยี Sub Harmonic Procedure	17
2.1.9.5 เทคโนโลยี Close Coupling	18
2.1.9.6 เทคโนโลยี Electrical Coupling	20
2.1.10 การนำระบบ RFID ไปใช้งาน	23
2.2 สเต็ปปิ้งมอเตอร์	23
2.2.1 สเต็ปปิ้งมอร์เตอร์แบบยูนิโพลาร์	24
2.2.1.1 แบบ 1 เฟส	25
2.2.1.2 แบบ 2 เฟส	25
2.2.2 วงจรขับสเต็ปปิ้งมอเตอร์	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ไอซีเบอร์ ULN2003 28
2.3 การใช้งานพอร์ตอนุกรมของๆไมโครคอนโทรลเลอร์ 29
2.3.1 การสื่อสารแบบอนุกรม 29
2.3.2 ประโยชน์ของพอร์ตอนุกรม 29
2.3.3 การเพิ่มระยะทางของพอร์ตอนุกรม 30
2.3.4 การใช้งานไมโครคอนโทรลเลอร์ 30
2.3.5 การเขียนโปรแกรมติดต่อและควบคุม Serial Port 30
2.3.6 โครงสร้างภายในชิพของ PIC18F2525 34
2.4 กลุ่มออบเจกต์หลักของสถาปัตยกรรม ADO.NET 35
2.4.1 ภาษา SQL 35
2.4.2 โครงสร้างพื้นฐานของภาษา SQL 35
2.4.3 พื้นฐานการใช้งานของการใช้คำสั่ง DML 36

บทที่ 3 การออกแบบและการสร้าง 37
3.1 หลักการใช้งานชุดอ่านบัตร 37
3.1.1 ฟังก์ชั่นการทำงานของ RFID 37
3.2 การออกแบบและการสร้างชุดควบคุมการเปิดปิดประตูทางเข้าออก 40
3.2.1 การทำงานของชุดควบคุมเปิดปิดประตูทางเข้าออก 40
3.2.2 ผังการทำงานชุดควบคุม 41
3.2.3 อธิบายโปรแกรมการควบคุมเปิดปิดประตูทางเข้าออก 42
3.2.4 หลักการสร้างชุดควบคุมการเปิดปิดประตู 42
3.2.5 การออกแบบวงจร 44
3.3 หลักการสร้างฐานข้อมูลของบริการรถไฟฟ้า 45
3.3.1 ตารางฐานข้อมูลสำหรับรายละเอียดของผู้ใช้บริการ 45
3.3.2 ตารางฐานข้อมูลสำหรับเก็บรหัสของผู้ดูและระบบ 45
3.4 โปรแกรมการจัดการระบบบัตรรถไฟฟ้า 46
3.4.1 ผังการทำงานของโปรแกรมหลักของชุดอ่านบัตร RFID 46
3.4.2 ผังการทำงานส่วนของโปรแกรม EDIT 47
3.4.3 ผังการทำงานส่วนของโปรแกรม REGISTER 48
3.4.4 ผังการทำงานส่วนของโปรแกรม TOP UP 49
3.4.5 ผังการทำงานส่วนของโปรแกรม OBSERVE 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลองและผลการทดลอง 51

4.1 การทดลองในส่วนของโปรแกรมและฐานข้อมูล 51

4.1.1 โปรแกรมการลงทะเบียน 51

4.1.2 โปรแกรมการแก้ไขข้อมูล 53

4.1.3 โปรแกรมการเติมเงินเพื่อเพิ่มยอดการใช้งาน 56

4.1.4 โปรแกรมสังเกตการณ์ 58

4.2 ชุดควบคุมการเปิด – ปิด ประตูเข้าออกและจอแสดงผล LCD 61

4.2.1 วงจรสเต็ปปิ้งมอเตอร์ 61

4.2.2 วงจรจ่ายพัลส์ 64

4.2.3 วงจรแสดงผล LCD 66

บทที่ 5 สรุปและวิจารณ์ผลการทดลอง 72

เอกสารอ้างอิง

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปภาพ		หน้า
รูปที่ 1	บล็อคไดอะแกรมภาพรวมของการทำงาน	2
รูปที่ 2.1	ระบบ RFID	3
รูปที่ 2.2	การสื่อสารระหว่างแท็กและตัวรับข้อมูล	4
รูปที่ 2.3	RFID และ Tag	6
รูปที่ 2.4	แท็กชนิด ACTIVE	7
รูปที่ 2.5	แท็กชนิดพาสซีฟ	8
รูปที่ 2.6	เครื่องลูกข่ายแบบพวงกุญแจ (Key and Key fob)	9
รูปที่ 2.7	เครื่องลูกข่ายแบบบัตรเครดิต	9
รูปที่ 2.8	โครงสร้างภายในเครื่องอ่าน RFID	10
รูปที่ 2.9	การสื่อสารระหว่างเครื่องลูกข่ายและตัวรับข้อมูล	12
รูปที่ 2.10	ความแตกต่างของการสื่อสารแบบ Full Duplex / Half Duplex และ Sequential	13
รูปที่ 2.11	เทคโนโลยี Inductive Coupling กับการรับพลังงานไฟฟ้าจากเครื่องอ่าน	13
รูปที่ 2.12	องค์ประกอบพื้นฐานของเครื่องอ่านและเครื่องลูกข่าย แบบ Inductive Coupling	14
รูปที่ 2.13	ตัวอย่างผลิตภัณฑ์เครื่องลูกกข่าย RFID ประเภท Inductive Coupling	15
รูปที่ 2.14	สัญญาณข้อมูลที่ถูกรบกวนที่ถูกบรรจุอยู่ในไซด์แบนด์	16
รูปที่ 2.15	วงจรที่ใช้สำหรับเทคนิคการส่งแบบ Load Modulation with Sub carrier	17
รูปที่ 2.16	โครงสร้างวงจรภายในของเครื่องลูกข่าย RFID ที่ทำงานแบบ Sub harmonic	18
รูปที่ 2.17	เทคนิคการถ่ายโอนข้อมูลจากเครื่องลูกข่ายสู่เครื่องอ่านผ่านการคัปปลิ้ง	19
รูปที่ 2.18	แสดงโครงสร้างและการทำงานของระบบ RFID แบบ Electrical Coupling	20
รูปที่ 2.19	แสดงความสัมพันธ์ระหว่างระดับแรงดันไฟฟ้าที่ผลิตบนแผ่นอิเล็กโทรด	20
รูปที่ 2.20	วงจรสมมูลระหว่างเครื่องลูกข่ายและเครื่องอ่าน RFID แบบ Electrical Coupling21	22
รูปที่ 2.21	บล็อกไดอะแกรมแสดงการควบคุมสเต็ปปิ้งมอเตอร์	23
รูปที่ 2.22	แสดงลักษณะการพันขดลวดของมอเตอร์แบบยูนิโพลาร์	24
รูปที่ 2.23	แสดงการวางของขดลวดแต่ละเฟสของสเต็ปปิ้งมอเตอร์	24
รูปที่ 2.24	แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบหนึ่งเฟส	25
รูปที่ 2.25	แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบสองเฟส	26
รูปที่ 2.26	แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบครึ่งสเต็ป	27
รูปที่ 2.27	แสดงวงจรขับ Stepping Motor โดยใช้ไอซี Driver เบอร์ ULN2003	28
รูปที่ 2.28	แสดงวงจรขับ Stepping Motor โดยใช้ Power Transistor เบอร์ BD139	28
รูปที่ 2.29	แสดงวงจรขับ Stepping Motor โดยใช้ Power MOSFET เบอร์ IRF510	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.30	ไมโครคอนโทรเลอร์PIC18F2525	31
รูปที่ 2.31	รูปของสัญญาณนาฬิกา	32
รูปที่ 2.32	ออสซิลเลเตอร์แบบต่าง ๆ และการกำหนดโปรแกรมให้ไมโครคอนโทรลเลอร์	33
รูปที่ 2.33	ผังไดอะแกรมของ PIC 18F252	34
รูปที่ 2.34	โครงสร้างกลุ่มออบเจคสถาปัตยกรรม ADO.NET	36
รูปที่ 3.1	บล็อกไดอะแกรมแสดงขั้นตอนการทำงานของ Reader	38
รูปที่ 3.2	ผังการทำงานโปรแกรมส่วนของการควบคุมการเปิด-ปิดประตูทางเข้าออก	41
รูปที่ 3.3	รูปแสดงแบบวงจรของสเต็ปปิ้งมอร์เตอร์	42
รูปที่ 3.4	แสดงวงจรภายในของสเต็ปปิ้งมอเตอร์	43
รูปที่ 3.5	รูปแสดงวงจรรวมทั้งหมดของวงจรชุดควบคุมการเปิด-ปิดประตู	44
รูปที่ 3.6	ผังการทำงานโปรแกรมส่วนของการควบคุมการเปิดปิดประตูทางเข้าออก	46
รูปที่ 3.7	หลักการทำงานของโปรแกรมส่วน EDIT	47
รูปที่ 3.8	หลักการทำงานของโปรแกรมส่วน REGISTER	48
รูปที่ 3.9	หลักการทำงานของโปรแกรมส่วน TOP UP	49
รูปที่ 3.10	หลักการทำงานของโปรแกรมส่วน OBSERVE	50
รูปที่ 4.1	หน้าจอเตรียมพร้อมส่วนของการลงทะเบียนลูกค้าใหม่	51
รูปที่ 4.2	หน้าจอลงทะเบียนลูกค้าใหม่เมื่อมีการ Tag ของ RFID	52
รูปที่ 4.3	การยืนยันในการขอลงทะเบียนเพื่อรับรหัส ID ใหม่	52
รูปที่ 4.4	การเพิ่มรหัสใหม่ที่ได้ทำการลงทะเบียนลงในฐานข้อมูล	53
รูปที่ 4.5	หน้าจอในส่วนของการแก้ไขข้อมูล	54
รูปที่ 4.6	หน้าจอการเปลี่ยนแปลงข้อมูลใหม่ในช่องที่สามารถแก้ไขได้	54
รูปที่ 4.7	โปรแกรมทำการแสดงคำยืนยันในการแก้ไขหรือเปลี่ยนแปลงข้อมูล	55
รูปที่ 4.8	การเปลี่ยนแปลงข้อมูลใหม่พร้อมทั้งทำการ Update ในฐานข้อมูล	55
รูปที่ 4.9	หน้าจอเตรียมพร้อมสำหรับการเติมยอดการใช้งาน	56
รูปที่ 4.10	หน้าจอเมื่อมีการ Tag ของ RFID	57
รูปที่ 4.11	การกรอกจำนวนครั้งการใช้ที่ต้องการเพิ่ม	57
รูปที่ 4.12	ยอดเครดิตจำนวนครั้งการใช้สุทธิใน RFID	58
รูปที่ 4.13	หน้าจอเตรียมพร้อมสำหรับการใช้งานส่วนของระบบเฝ้าดูการใช้งาน	59
รูปที่ 4.14	หน้าจอเดิมก่อนการเข้าใช้บริการ (ยอดการใช้งาน 40 ครั้ง)	59
รูปที่ 4.15	สถานะของผู้ใช้งานเมื่อมีการ Tag ของ RFID เพื่อเข้าใช้บริการ	60
รูปที่ 4.16	สถานะของผู้ใช้งานเมื่อมีการ Tag ของ RFID เพื่อออกจากการใช้บริการ	60
รูปที่ 4.17	สถานะยอดเงินไม่เพียงพอ เมื่อมีการ Tag ของ RFID เพื่อเข้าใช้บริการ	61
รูปที่ 4.18	วงจรสำหรับใช้ควบคุม Stepper Motor แบบ Unipolar	62
รูปที่ 4.19	วงจรภายในของสเต็ปปิ้งมอเตอร์	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.20	รูปการต่อวงจรสำหรับใช้ควบคุม Stepping Motor แบบ Unipolar Step 1	63
รูปที่ 4.21	รูปการต่อวงจรสำหรับใช้ควบคุม Stepping Motor แบบ Unipolar Step 2	63
รูปที่ 4.22	การเปรียบเทียบสัญญาณพัลส์จากจุดที่ 1 กับ สัญญาณพัลส์จากจุดที่ 2	64
รูปที่ 4.23	การเปรียบเทียบสัญญาณพัลส์จากจุดที่ 2 กับ สัญญาณพัลส์จากจุดที่ 3	65
รูปที่ 4.24	การเปรียบเทียบสัญญาณพัลส์จากจุดที่ 3 กับ สัญญาณพัลส์จากจุดที่ 4	65
รูปที่ 4.25	การเปรียบเทียบสัญญาณพัลส์จากจุดที่ 1 กับ สัญญาณพัลส์จากจุดที่ 4	66
รูปที่ 4.26	วงจรสำหรับการแสดงผลที่จอ LCD	67
รูปที่ 4.27	การต่อวงจรสำหรับแสดงผลที่จอ LCD	67
รูปที่ 4.28	การเข้าใช้การเข้าใช้บริการที่ประตูทางเข้า	68
รูปที่ 4.29	ผลการเข้าใช้การเข้าใช้บริการที่ประตูทางเข้า	69
รูปที่ 4.30	ผลการออกจากการใช้บริการที่ประตูทางออก	69
รูปที่ 4.31	สถานะเตรียมพร้อมของหน้าจอแอลซีดี ก่อนการ Tag ของ RFID	70
รูปที่ 4.32	สถานะการใช้งานเมื่อมีการ Tag ของ RFID	70
รูปที่ 4.33	วงจรการทำงานในสถานะพร้อมใช้งาน	71
รูปที่ 4.34	ภาพรวมแบบจำลองการทำงานในสถานะพร้อมใช้งาน	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

ตารางที่ 2.1 ย่านความถี่ต่างๆ ของระบบ RFID และการใช้งาน 5

ตารางที่ 2.2 ลำดับการทำงานของขดลวดเมื่อได้รับการกระตุ้นแบบหนึ่งเฟส 25

ตารางที่ 2.3 ลำดับการทำงานของขดลวดเมื่อได้รับการกระตุ้นแบบสองเฟส 26

ตารางที่ 2.4 การทำงานของขดลวดเมื่อได้รับการกระตุ้นแบบครึ่งเสต็ป 27

ตารางที่ 4.1 ผลการทดลองการทำงานระหว่างของขดลวดในสเต็ปปิ้งมอเตอร์กับมุมที่ได้ 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมประตูเข้าออกของบริการรถไฟฟ้าผ่านอาร์เอฟไอดี

THE ENTRANCE SYSTEM OF BTS SERVICE VIA RFID

โดย	นางสาวภัททิรา	อ่ำใหญ่	48010655
	นาวสาวรัชดาวรรณ	สักกพันธุ์นิกร	48010729
	นายอรรถพร	สิงห์คีรี	48011083

อาจารย์ที่ปรึกษา รศ.ดร.ไกรสิน ส่งวัฒนา

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการประยุกต์ใช้เทคโนโลยีอาร์เอฟไอดี (Radio Frequency Identification) สำหรับควบคุมประตูเข้าออกของบริการรถไฟฟ้าบีทีเอส จากระบบเดิมที่เคยใช้บัตรสมาร์ตการ์ด แต่จะใช้เทคโนโลยี อาร์เอฟไอดี ในซิมการ์ดของโทรศัพท์เคลื่อนที่แทน ทำให้สะดวกเนื่องจากไม่ต้องพกบัตรหลายใบ

ส่วนประกอบของระบบ ประกอบไปด้วย สามส่วนคือ ส่วนของการเก็บข้อมูลผ่านอาร์เอฟไอดี ส่วนที่สองคือ ส่วนของไมโครคอนโทรลเลอร์ที่ควบคุมการเปิดปิดประตู การเปลี่ยนแปลงข้อมูลการใช้บริการพร้อมทั้ง แสดงผลบนจอแอลซีดี และส่วนสุดท้ายคือ ส่วนของ ฐานข้อมูลที่จะเก็บข้อมูลของผู้ใช้และจำนวนครั้งของผู้ใช้

ABSTRACT

This project presents the application of RFID in mobile SIM module which applied for entrancing system in BTS train service . It is more convenient than bring many cards.

This system is composed of 3 parts. First, Process for getting data from RFID tag in the SIM module. Second part is microcontroller. It's used for controlling entrance system , management data and show user status on LCD monitor . The last part is the database for information management.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

ปัจจุบันโลกมีการพัฒนาไปอย่างรวดเร็ว เกิดเทคโนโลยีใหม่ ๆ มากมาย ซึ่งเทคโนโลยีที่ล้วนสร้างขึ้นมาก็เพื่อตอบสนองความต้องการของมนุษย์และเพื่ออำนวยความสะดวก การจราจรก็เป็นอีกส่วนหนึ่งที่มนุษย์ต้องการเทคโน - โลยีเข้ามาช่วยให้ชีวิตประจำวันที่ต้องการความเร่งด่วน จึงได้เกิดเทคโนโลยีรถไฟฟ้าขึ้นมาอำนวยความสะดวก แต่ยังพบว่ามีปัญหาความล่าช้าในการเข้าแถวซื้อตั๋วจากพนักงานหรือเครื่องจำหน่ายตั๋วอัตโนมัติ

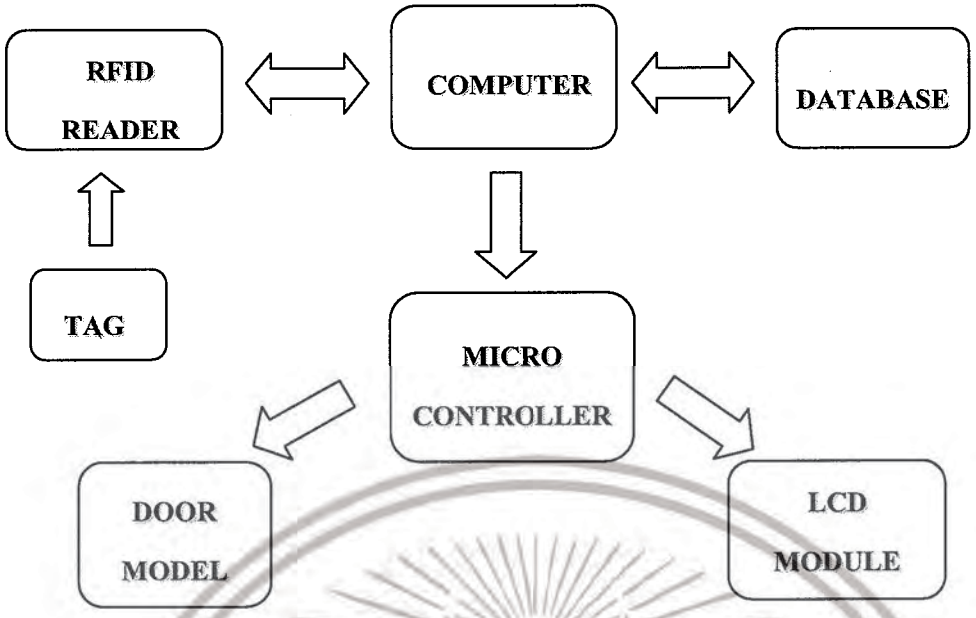
เพื่อแก้ไขปัญหาดังกล่าวเราจึงได้นำเอาเทคโนโลยี RFID(Radio Frequency Indentification) ที่อยู่ในซิมการ์ดของโทรศัพท์เคลื่อนที่แทน ซึ่งในปริญญานิพนธ์ฉบับนี้เป็นการนำแอพพลิเคชั่นของซิมการ์ดโทรศัพท์ที่มี RFID ในตัวมาประยุกต์ใช้ทำให้สะดวกเนื่องจากไม่ต้องพกบัตรหลายใบ เทคโนโลยี RFID เป็นเทคโนโลยีใหม่ที่กำลังมีบทบาทและความสำคัญเพิ่มขึ้นอย่างรวดเร็ว และการประยุกต์ใช้ RFID มีหลายรูปแบบด้วยจุดประสงค์ที่แตกต่างกัน แต่อยู่บนหลักการพื้นฐานเดียวกัน นั่นคือใช้คลื่นความถี่วิทยุเพื่อการระบุเอกลักษณ์ของวัตถุหรือเจ้าของวัตถุที่ติดป้าย RFID แทนการระบุด้วยวิธีการอื่น

ในอนาคตการใช้เทคโนโลยี RFID ที่อยู่ในซิมการ์ดของโทรศัพท์เคลื่อนที่แทน สามารถพัฒนาให้ใช้ในระบบการเข้าออกของระบบของรถไฟฟ้าใต้ดิน ซึ่งเหมาะกับยุคที่รวมรวมบริการต่าง ๆ เข้าไว้ด้วยกัน และยังสามารถจองตั๋วหรือเช็คยอดการใช้งานได้ด้วย

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษากระบวนการทำงานของระบบ RFID เพื่อนำไปประยุกต์ใช้กับบริการรถไฟฟ้า รวมถึงการนำไปประยุกต์ใช้กับอุตสาหกรรมอื่น ๆ ต่อไปในอนาคต
2. เพื่อศึกษาโครงสร้างการเชื่อมต่อในการทำงานร่วมกันของระบบ RFID กับฐานข้อมูลและไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 บล็อกไดอะแกรมภาพรวมของการทำงาน

1.3 ประโยชน์ที่ได้รับจากโครงงาน

1. มีความเข้าใจในเรื่อง RFID มากขึ้น
2. สามารถเขียนโปรแกรมจัดการฐานข้อมูลด้วยภาษา SQL
3. สามารถเขียนโปรแกรมภาษา c# เพื่อใช้ในการจัดการฐานข้อมูล
4. สามารถโปรแกรมไมโครคอนโทรลเลอร์เพื่อควบคุมอุปกรณ์ต่างๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

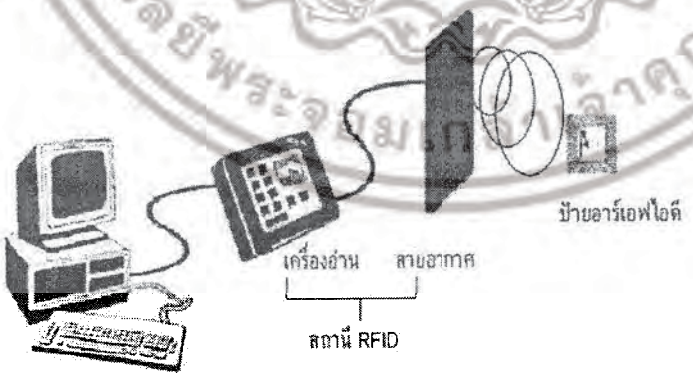
บทที่ 2
ทฤษฎีและหลักการ

ในบทที่จะกล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องกับปริญญานิพนธ์ฉบับนี้ ซึ่งประกอบไปด้วย

1. ชุดอ่านบัตร RFID
2.สเต็ปปิ้งมอร์เตอร์
3.ไมโครคอนโทรลเลอร์
4.SQL Server

2.1 ส่วนประกอบของระบบ RFID

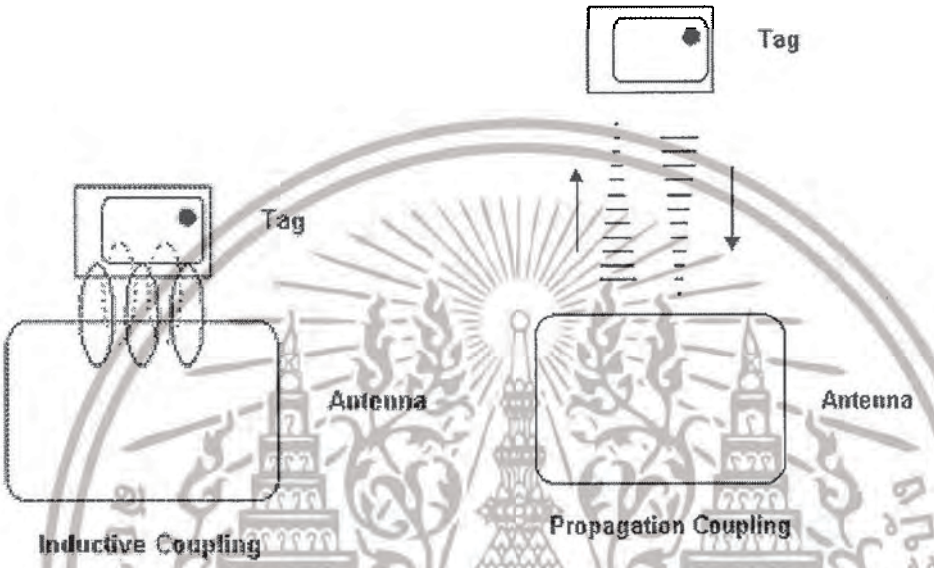
ในระบบ RFID จะมีองค์ประกอบหลัก ๆ อยู่ 3 ส่วนด้วยกัน คือ ส่วนแรกคือฉลากหรือทราน สพอนเดอร์ (Transponder , Transmitter & Responder) หรือที่เรียกกันโดยทั่วๆ ไปว่า " แท็ก " (Tag) เป็นป้ายขนาดเล็กที่จะถูกผนึกอยู่กับวัตถุที่เราสนใจ โดยฉลากนี้จะทำการบันทึกข้อมูลเกี่ยวกับวัตถุชิ้นนั้น เอาไว้ ส่วนที่สอง ก็คืออุปกรณ์สำหรับอ่านหรือเขียนข้อมูลภายในแทกส์ มีชื่อเรียกอย่างรวม ๆ ว่า ทรานซิฟเวอร์ (Transceiver , Transmitter & Responder) หรือที่เรียกกันโดยทั่ว ๆ ไปว่า " เครื่องอ่าน " (Reader) ทั้งสองส่วนจะสื่อสารกันโดยใช้ช่องความถี่วิทยุ สัญญาณนี้ผ่านได้ทั้งโลหะและอโลหะ แต่จะไม่สามารถติดต่อกับเครื่องอ่านได้โดยตรง เมื่อเครื่องอ่านส่งข้อมูลผ่านความถี่วิทยุ แสดงถึงความต้องการข้อมูลที่ถูกระบุไว้จากป้าย ป้ายจะตอบข้อมูลกลับและเครื่องอ่านจะส่งข้อมูลต่อไปยังส่วนประมวลผลหลักของคอมพิวเตอร์โดยเครื่องอ่านจะติดต่อสื่อสารกับคอมพิวเตอร์โดยผ่านทางสายโครง-ข่าย LAN (Local Area Network) หรือส่งผ่านทางความถี่วิทยุจากทั้งอุปกรณ์มีสายและอุปกรณ์ไร้สาย ส่วนที่ 3 ได้แก่ ระบบการประยุกต์ใช้งาน ทั้งนี้รวมถึงฮาร์ดแวร์และซอฟแวร์ประยุกต์ใช้งาน หรือระบบฐานข้อมูล ทั้งนี้ขึ้นกับระบบการใช้งานที่เกี่ยวข้อง เช่น ระบบข้อมูลสินค้า



รูปที่ 2.1 ระบบ RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลของระบบ RFID คือระหว่างแท็กและตัวอ่านข้อมูล (Reader หรือ Interrogator) จะสื่อสารแบบไร้สายผ่านอากาศ โดยจะนำข้อมูลมาทำการมอดูเลต (Modulation) กับคลื่นพาหะที่เป็นคลื่นความถี่วิทยุโดยมีสายอากาศ (Antenna) ที่อยู่ในตัวอ่านข้อมูลเป็นตัวรับและส่งคลื่นซึ่งแบ่งออกเป็น 2 วิธีด้วยกันคือ วิธีเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้า (Inductive Coupling หรือ Proximity Electromagnetic) กับ วิธีการแผ่คลื่นแม่เหล็กไฟฟ้า (Electromegnatic Propogation Coupling) ดังรูปที่ 2.2



รูปที่ 2.2 การสื่อสารระหว่างแท็กและตัวรับข้อมูล

เทคนิคการมอดูเลตข้อมูลเข้ากับคลื่นพาหะก็มีด้วยกันหลายวิธี เช่น ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying) หรือ PSK (Phase Shift Keying) ซึ่งขึ้นอยู่กับผู้ออกแบบจะเลือกให้มีความเหมาะสมกับการใช้งานแต่ละประเภท

2.1.1 ความถี่ของคลื่นพาหะ

ในปัจจุบันได้มีการรวมกลุ่มระหว่างแต่ละประเทศ เพื่อทำการกำหนดมาตรฐานความถี่คลื่นพาหะของระบบ RFID โดยมีสามกลุ่มใหญ่ๆ คือ กลุ่มประเทศในยุโรปและอาฟริกา (Region 1), กลุ่มประเทศอเมริกาเหนือและอเมริกาใต้ (Region 2) และสุดท้ายคือกลุ่มประเทศตะวันออกไกลและออสเตรเลีย (Region 3) ซึ่งแต่ละกลุ่มประเทศจะกำหนดแนวทางในการเลือกใช้ความถี่ต่างๆให้แก่บรรดาประเทศสมาชิก

อย่างไรก็ตาม ความถี่ของคลื่นพาหะที่นิยมใช้งานในย่านความถี่ต่ำ ย่านความถี่ปานกลาง และย่านความถี่สูงก็คือ 125 kHz, 13.56 MHz และ 2.45 GHz ตามลำดับดังที่แสดงไว้ในตารางที่ 2.1 นอกจากนี้รัฐบาลของแต่ละประเทศโดยทั่วไปจะมีการออกกฎหมายเกี่ยวกับระเบียบการใช้งานย่านความถี่ต่างๆ รวมถึงกำลังส่งของระบบ RFID ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ย่านความถี่ต่างๆ ของระบบ RFID และการใช้งาน

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

2.1.2 ลักษณะของอุปกรณ์ RFID

อุปกรณ์ RFID ทั้งเครื่องลูกข่าย (Transponder) หรือเครื่องอ่าน (Reader) ไม่ว่าจะเป็นรุ่นใดที่มีจะเป็นรุ่นใดที่มีจำหน่ายตามท้องตลาด จะมีรูปแบบการทำงานอย่างใดอย่างหนึ่ง ระหว่างการรับส่งข้อมูลแบบ Full Duplex (FDX) / Half Duplex (HDX) กับแบบ Sequential (SEQ) ซึ่งทั้งสองแบบมีหลักการทำงานที่แตกต่างกันดังนี้

2.1.3 การรับส่งข้อมูลแบบ FDX / HDX

เครื่องอ่านจะทำการส่งสัญญาณข้อมูลออกตลอดเวลา ซึ่งเท่ากับว่าเป็นการส่งพลังงานไฟฟ้าผ่านคลื่นวิทยุ ไปจ่ายให้กับบรรดาเครื่องลูกข่ายที่อยู่ในอาณาบริเวณการใช้งานด้วย และเนื่องจากความแรงของคลื่นสัญญาณความถี่วิทยุที่เครื่องลูกข่ายส่งออกมามีกำลังส่งต่ำ ทำให้มีแนวโน้มว่าสัญญาณอาจส่งไปไม่ถึงเครื่องอ่าน หรืออาจส่งไปถึงแต่มีระดับสัญญาณรบกวนปะปนอยู่ด้วย เมื่อเทียบกับสัญญาณที่ถูกส่งออกมาจากเครื่องอ่าน ซึ่งมีแหล่งจ่ายไฟฟ้าเป็นของตนเอง ทำให้ส่งสัญญาณออกมาได้สูงกว่า ดังนั้นจึงมีมาตรการในการสร้างความแตกต่างของสัญญาณคลื่นวิทยุ ที่มีการส่งออกมาจากเครื่องลูกข่ายแต่ละเครื่อง ให้มีเอกลักษณ์ (Identity) ที่แตกต่างกัน เพื่อให้เครื่องอ่านสามารถแยกแยะแหล่งที่มาได้อย่างถูกต้องโดยใช้มอดูเลตสัญญาณแบบ Load Modulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การรับส่งข้อมูลแบบ SEQ

เป็นการสื่อสารระหว่างเครื่องอ่านและเครื่องลูกข่าย RFID ที่กำหนดให้เครื่องอ่านมีการหยุดส่งสัญญาณคลื่นความถี่วิทยุเป็นช่วง ๆ ต่อเนื่อง หรือเป็นรายคาบเวลาสม่ำเสมอ ช่วงเวลาที่ไม่มีการส่งสัญญาณออกจากเครื่องอ่านนี้เองที่เป็นจังหวะ ให้เครื่องลูกข่ายทำการส่งสัญญาณกลับมายังเครื่องอ่านเป็นการสร้างฐานเวลา (Timing) สำหรับใช้อ้างอิงภายในระบบสื่อสารแบบ RFID ที่ประกอบไปด้วยเครื่องอ่าน และกลุ่มของเครื่องลูกข่ายที่มีความเที่ยงตรงมาก แต่การรับส่งข้อมูลแบบนี้มีข้อจำกัดอยู่ที่ประเด็นการจ่ายพลังงานให้กับเครื่องลูกข่าย เนื่องจากในช่วงเวลาที่เครื่องอ่านหยุดส่งสัญญาณนั้น บรรดาเครื่อลูกข่ายทั้งหลายก็จะไม่ได้รับพลังงานไฟฟ้าที่ส่งผ่านทางคลื่นวิทยุไปด้วย ทำให้ต้องมีการติดตั้งอุกกรณ์เก็บประจุ ไฟฟ้าหรือแบตเตอรรี่ขนาดเล็กไว้ในเครื่องลูกข่ายที่นำมาใช้ กับการรับส่งข้อมูลแบบนี้

2.1.5 Tags หรือ Transponders

แท็กส์ (Tag) หรือเรียกอีกชื่อหนึ่งว่า ทรานสปอนเดอร์ (Transponder) มาจากคำว่าทรานสมิตเตอร์ (Transmitter) ผสมกับคำว่า เรสปอนเดอร์ (Responder) นั่นเอง ถ้าจะแปลให้ตรงตามศัพท์ แท็กส์ ก็จะทำหน้าที่ส่งสัญญาณหรือข้อมูลที่บันทึกอยู่ในแท็กส์ตอบสนองไปที่ตัวอ่านข้อมูล การสื่อสารระหว่างแท็ก และตัวอ่านข้อมูลจะเป็นแบบไร้สายผ่านอากาศ ภายในแท็กจะประกอบไปด้วยชิปสารกึ่งตัวนำ (Semiconductor Chip) ซึ่งเชื่อมต่ออยู่กับสายอากาศ แท็กอาจมีรูปร่างได้หลายแบบขึ้นอยู่กับการนำไปใช้งาน โดยอาจมีรูปร่างเหมือนบัตรเครดิตในการใช้งานทั่วไป หรือเล็กขนาดไส้ดินสอยาวเพียง 10 มิลลิเมตร เพื่อฝังเข้าไปใต้ผิวหนังสัตว์ในกรณีนำไปใช้ในงานปศุสัตว์ หรือ อาจมีขนาดใหญ่มากสำหรับแท็กที่ใช้ติดกับเครื่องจักร ขณะทำการขนส่งแท็กอาจนำไปติดไว้กับสินค้าในร้านค้าปลีกทั่วไป เพื่อป้องกันขโมย โดยจะมีการติดตั้งสายอากาศของตัวอ่านข้อมูลขนาดใหญ่ไว้ตรงประตูทางออก เพื่อทำการตรวจจับขโมย

ชิปที่อยู่ในแท็กจะมีหน่วยความจำซึ่งอาจเป็นแบบอ่านได้อย่างเดียว (ROM) หรือทั้งอ่านทั้งเขียน (RAM) ทั้งนี้ขึ้นอยู่กับความต้องการในการใช้งาน โดยปกติหน่วยความจำแบบ ROM จะใช้เก็บข้อมูลเกี่ยวกับการรักษาความปลอดภัย เช่น ข้อมูลของบุคคลที่มีสิทธิ์ผ่านเข้าออกในบริเวณที่มีการควบคุมหรือระบบปฏิบัติการ ในขณะที่ RAM จะใช้เก็บข้อมูลชั่วคราวในระหว่างที่ แท็ก และ ตัวอ่านข้อมูล ทำการติดต่อสื่อสารกัน

รูปที่ 2.3 RFID และ Tag

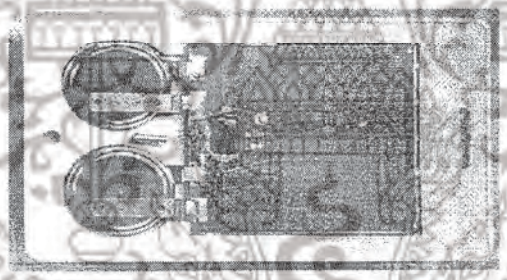
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้อาจมีการนำหน่วยความจำแบบ EEPROM มาใช้ในกรณีการเก็บข้อมูลในระหว่างที่แท็กและตัวอ่านข้อมูลทำการสื่อสาร และข้อมูลยังคงอยู่ ถึงแม้จะไม่มีพลังงานไฟฟ้าป้อนให้แก่แท็ก เราสามารถแบ่งชนิดของแท็กได้เป็น 2 ชนิด คือ

1. เครื่องลูกข่ายชนิดแอ็กตีฟ (Active Tag)
2. เครื่องลูกข่ายชนิดพาสซีฟ (Passive Tag)

2.1.5.1 เครื่องลูกข่ายชนิดแอกตีฟ (Active Tag)

เครื่องลูกข่ายชนิดนี้จะมีแบตเตอรี่อยู่ภายในซึ่งใช้เป็นแหล่งจ่ายไฟขนาดเล็ก เพื่อป้อนพลังงานไฟฟ้าให้เครื่องลูกข่ายทำงานโดยปกติ โดยเครื่องลูกข่ายชนิดนี้มีฟังก์ชันการทำงานทั่วไปทั้งอ่านและเขียนข้อมูลลงในเครื่องลูกข่ายได้ และการที่ต้องใช้แบตเตอรี่จึงทำให้เครื่องลูกข่ายชนิดแอกทีฟมีอายุการใช้งานจำกัดตามอายุของแบตเตอรี่ เมื่อแบตเตอรี่หมดก็ต้องนำแท็กไปทิ้ง ไม่สามารถนำกลับมาใช้ใหม่ได้ เนื่องจากมีการซีล (Seal) ที่ตัวเครื่องลูกข่ายจึงไม่สามารถเปลี่ยนแบตเตอรี่ได้ อย่างไรก็ตามถ้าสามารถออกแบบวงจรของเครื่องลูกข่ายให้กินกระแสไฟฟ้าน้อย ๆ ก็อาจจะมีอายุการใช้งานนานนับสิบปี เครื่องลูกข่ายชนิดแอกตีฟนี้จะมีหน่วยความจำภายในขนาดใหญ่ได้ถึง 1 เมกะไบต์ มีกำลังส่งสูงและระยะการรับส่งข้อมูลไกลสูงสุดถึง 6 เมตร ซึ่งไกลกว่าเครื่องลูกข่ายชนิดพาสซีฟ นอกจากนี้ยังทำงานในบริเวณที่มีสัญญาณรบกวนได้ดี แม้เครื่องลูกข่ายชนิดนี้จะมีข้อดีอยู่หลายข้อ แต่ก็มีข้อด้อยเช่นกัน ในเรื่องของราคาต่อหน่วยที่ค่อนข้างแพง มีขนาดค่อนข้างใหญ่ และมีการใช้งานในระยะเวลาที่จำกัด



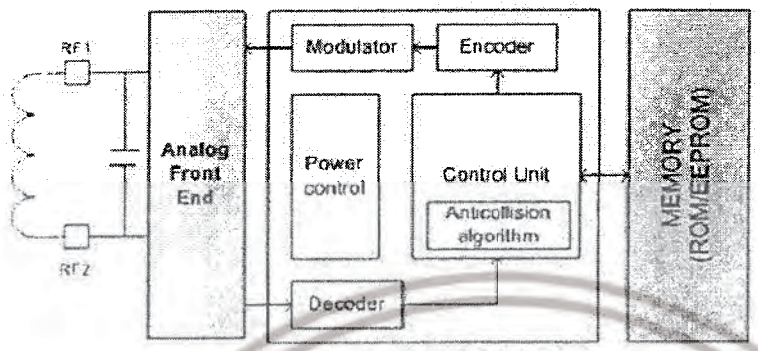
รูปที่ 2.4 แท็กชนิดแอกตีฟ

2.1.5.2 เครื่องลูกข่ายชนิดพาสซีฟ

จะไม่มีแบตเตอรี่อยู่ภายในหรือไม่จำเป็นต้องรับแหล่งจ่ายไฟใด ๆ เพราะจำทำงานโดยอาศัยพลังงานไฟฟ้าที่เกิดจากการเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้าจากตัวอ่านข้อมูล (มีวงจรกำเนิดไฟฟ้าขนาดเล็กอยู่ในตัว) หรือที่เรียกว่าอุปกรณ์ Transceiver จึงทำให้เครื่องลูกข่ายชนิดพาสซีฟ มีน้ำหนักเบาและเล็กกว่าเครื่องลูกข่ายชนิดแอ็กตีฟ ราคาถูกกว่า และมีอายุการใช้งานไม่จำกัด แต่ข้อเสียก็คือ ระยะการรับส่งข้อมูลใกล้ซึ่งสามารถส่งข้อมูลได้ไกลสุดเพียง 1.5 เมตร ซึ่งเป็นระยะการอ่านที่สั้น มีหน่วยความจำขนาดเล็กซึ่งโดยทั่ว ๆ ไป ประมาณ 32 ถึง 128 บิต และตัวเครื่องอ่านข้อมูลจะต้องมีความไวและกำลังที่สูงนอกจากนี้เครื่องลูกข่ายชนิดพาสซีฟมักจะมีปัญหาเมื่อนำไปใช้งานในสิ่งแวดล้อมที่มีสัญญาณแม่เหล็กไฟฟ้ารบกวนสูงอีกด้วย แต่ข้อได้เปรียบในเรื่องราคาต่อหน่วยที่ต่ำกว่าเครื่องลูกข่ายชนิดแอกตีฟ และอายุการใช้งานที่ยาวนานกว่าทำให้เครื่องลูกข่ายชนิดพาสซีฟนี้เป็นที่นิยมมากกว่า ไอซีของเครื่องลูกข่ายชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พาสซีฟที่มีการผลิตออกมาจะมีขนาดและรูปร่างเป็นได้ตั้งแต่แท่งหรือแผ่นขนาดเลกจนแทบไม่สามารถมองเห็นได้ ไปจนถึงขนาดใหญ่จนสะดุดตา ซึ่งต่างก็มีความเหมาะสมกับชนิดงานที่แตกต่างกัน



รูปที่ 2.5 แท็กชนิดพาสซีฟ

นอกจากการแบ่งชนิดที่ว่ามาแล้ว แท็ก ยังถูกแบ่งประเภทจากรูปแบบในการใช้งานเป็น 3 แบบ คือ แบบที่สามารถถูกอ่านและเขียนข้อมูลได้อย่างอิสระ , แบบเขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้อย่างอิสระ และ แบบอ่านได้เพียงอย่างเดียวด้วย อย่างไรก็ตามเนื่องจากในการใช้งานเพื่อเป็นฉลาก นิยมใช้แท็กชนิดพาสซีฟมากกว่า

2.1.5.3 ประเภทของเครื่องลูกข่าย RFID (Tag)

อุปกรณ์เครื่องลูกข่าย RFID ประเภทต่าง ๆ ที่มีการผลิตใช้งานในกิจการต่าง ๆ ในปัจจุบัน โดยแยกตามความแตกต่างของโครงสร้างและการออกแบบบ เป็นสำคัญตามด้วยตัวอย่างประเภทของการประยุกต์ใช้งาน

2.1.5.3.1 เครื่องลูกข่ายแบบจานและเหรียญ (Disk and coin)

เครื่องลูกข่ายจะถูกบรรจุอยู่ในโครงสร้างทรงกลมคล้ายจานหรือเหรียญ และมีการเคลือบป้องกันไว้เป็นอย่างดี ขยาดเส้นผ่านศูนย์กลางของเครื่องลูกข่ายชนิดนี้มีตั้งแต่ไม่กี่มิลลิเมตร ไปจนถึง 10 เซนติเมตร

2.1.5.3.2 เครื่องลูกข่ายแบบกระเปาะแก้ว (Glass housing)

เป็นเครื่องลูกข่ายที่ได้รับการพัฒนาและผลิตขึ้นสำหรับใช้ฝังไว้ใต้ผิวหนังของสัตว์ เพื่อใช้ในการติดตามข้อมูลโดยเฉพาะอย่างยิ่งในแวดวงสัตวศาสตร์และการสาธารณสุข ตัวกระเปาะแก้วมีขนาดความยาวในช่วง 12 – 32 มิลลิเมตร

2.1.5.3.3 เครื่องลูกข่ายแบบพลาสติก (Plastic housing)

นิยมเรียกกันในชื่อย่อว่า PP (Plastic Package) ได้รับการพัฒนาขึ้นสำหรับใช้ในกิจการอุตสาหกรรมบางประเภทที่มีความต้องการเฉพาะรูปแบบ เครื่องลูกข่ายแบบนี้มีขนาดบางและสามารถนำไปติดตั้งประกอบผลิตภัณฑ์หลาย ๆ ประเภท ที่พบเห็นทั่วไปก็คือ การนำเครื่องลูกข่ายแบบ PP ติดตั้งฝังอยู่กับพวงกุญแจรถยนต์ ในลักษณะของพวงกุญแจอัจฉริยะ (Electronic immobilization system)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.3.4 เครื่องลูกข่ายยสำหรับใช้เฉพาะกิจ

เป็นเครื่งลูกข่าย RFID อีกประเภท ที่ได้รับการออกแบบมาเป็นพิเศษ สำหรับใช้งานในกรณีเฉพาะกิจ โดยมีการคิดค้นและพัฒนาเพื่อบรรจุเครื่องลูกข่ายลงในโครงสร้างและหน้าสัมผัสที่เป็นโลหะ

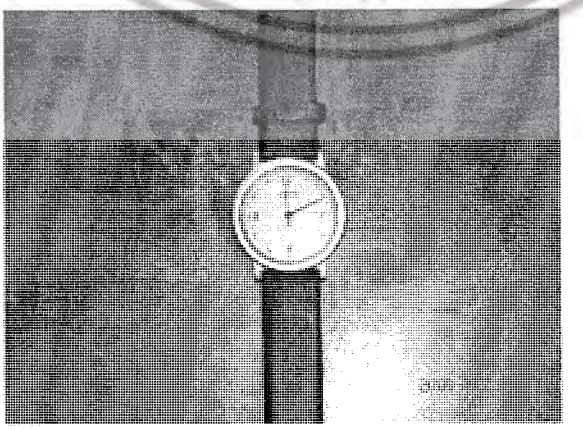
2.1.5.3.5 เครื่องลูกข่ายแบบพวงกุญแจ (Key and Key fob)

เครื่องลูกข่าย RFID สามารถได้รับการพัฒนาให้อยู่ในรูปของกุญแจพิเศษที่ใช้ในการเปิดปิดประตู หรือใช้กับระบบรักษาความปลอดภัยต่าง ๆ ซึ่งพวงกุญแจอิเล็กทรอนิกส์ชนิดนี้ไม่มีลูกกุญแจ แต่กลับทำหน้าที่ในการส่งสัญญาณข้อมูลต่าง ๆ ซึ่งอาจจะมีการนำเข้ารหัสเพื่อใช้ในการยืนตัวตนบุคคล

รูปที่ 2.6 เครื่องลูกข่ายแบบพวงกุญแจ (Key and Key fob)

2.1.5.3.6 เครื่องลูกข่ายแบบนาฬิกา

เครื่องลูกข่ายแบบนาฬิกาได้รับการคิดค้นและพัฒนาตั้งแต่ พ.ศ. 2533 – 2535 โดยบริษัทสัญชาติออสเตรเรียที่มีชื่อว่า Sky – Data โดยผลิตขึ้นเพื่อใช้เป็นนาฬิกาสวมใส่ข้อมือและใช้เป็นอุปกรณ์แสดงตัวตนในลักษณะช่องเซ็นเซอร์แบบไร้การสัมผัส (Contact less sensor) เพื่อผ่านเข้าสู่ลู่เล่นสกีสำหรับนักสกีทั้งหลาย



รูปที่ 2.7 เครื่องลูกข่ายแบบนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.3.7 เครื่องลูกข่ายมาตรฐาน ID – 1

มีรูปลักษณ์ภายนอกไม่ต่างจากบัตรเดตดิตหรือบัตรโทรศัพท์ทั่วไป โดยมีขนาดประมาณ 85.72 x 54.03 มิลลิเมตร ความหนา 0.76 มิลลิเมตร เป็นเครื่องลูกข่ายที่ได้รับการนำไปใช้งานอย่างแพร่หลาย ด้วยคุณสมบัติพิเศษก็คือระยะทำการรับส่งสัญญาณที่เหนือกว่าเครื่องลูกข่ายแบบอื่น ๆ ทั้งนี้ก็เนื่องมาจากพื้นที่หน้าตัดที่กว้างมากของตัวบัตร ทำให้สามารถจัดวางระบบสายอากาศที่มีแกนขนาดใหญ่ช่วยเพิ่มรัศมีทำการขึ้นได้อีกมาก

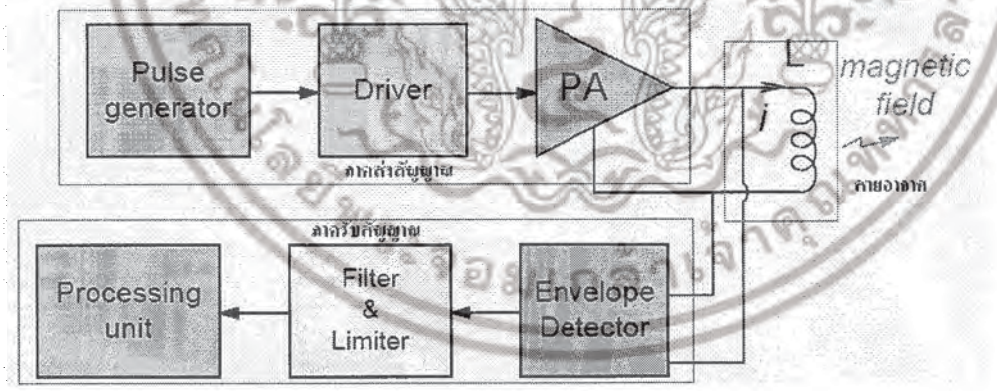
2.1.5.3.8 เครื่องลูกข่ายแบบ Smart label

เป็นเครื่องลูกข่าย RFID บางที่สุดที่ได้รับการออกแบบเป็นกระดาษบาง ๆ สามารถงอหรือทนได้ โดยขดลวดสำหรับส่งสัญญาณจะถูกออกแบบแถบฟอยล์พลาสติกความหนาเพียง 0.1 มิลลิเมตร โดยใช้ขบวนการผลิตแบบพิมพ์สกรีนโดยทั่วไปแถบฟอยล์เหล่านี้จะถูกเคลือบชั้นหนึ่ง โดยเยื่อกระดาษและทับอีกชั้นหนึ่งด้วยสารยึดเกาะ เพื่อป้องกันการฉีกขาด

2.1.5.3.9 เครื่องลูกข่ายแบบขดลวดบนชิป (Coil – on Chip)

เครื่องลูกข่าย RFID ชนิดนี้มีความแตกต่างจากเครื่องลูกข่ายแบบอื่น ๆ ซึ่งมีการแยกส่วนระหว่างเครื่องลูกข่ายกับขดลวดหรือสายอากาศอย่างชัดเจน แต่สำหรับเครื่องลูกข่ายแบบขดลวดบนชิปนั้น จะรวมส่วนที่เป็นขดลวดสายอากาศและตัววงจรประมวลผลเข้าด้วยกัน

2.1.6 Reader หรือ Interrogator



รูป 2.8 โครงสร้างภายในเครื่องอ่าน RFID

หน้าที่สำคัญของตัวอ่านข้อมูล (Reader หรือ Interrogator) ก็คือการรับข้อมูลที่ส่งมาจากแท็ก แล้วทำการตรวจสอบความผิดพลาดของข้อมูล ถอดรหัสข้อมูล และนำข้อมูลผ่านเข้าสู่กระบวนการต่อไป นอกจากนี้ตัวอ่านข้อมูลที่ดีต้องมีความสามารถในการป้องกันการอ่านข้อมูลซ้ำ เช่นในกรณีที่แท็กถูกวางทิ้งอยู่ในบริเวณสนามแม่เหล็กไฟฟ้าที่ตัวอ่านข้อมูลสร้างขึ้น หรืออยู่ในระยะการรับส่ง ก็อาจทำให้ตัวอ่านข้อมูลทำการรับหรืออ่านข้อมูลจากแท็กซ้ำอยู่เรื่อย ๆ ไม่สิ้นสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นตัวอ่านข้อมูลที่ดีต้องมีระบบป้องกันเหตุการณ์เช่นนี้ที่เรียกว่าระบบ " Hands Down Polling " โดยตัวอ่านข้อมูล จะสั่งให้แท็กหยุดการส่งข้อมูลในกรณีเกิดเหตุการณ์ดังกล่าว หรืออาจมีบางกรณีที่มีแท็กหลายแท็กอยู่ในบริเวณสนามแม่เหล็กไฟฟ้าพร้อมกัน หรือที่เรียกว่า " Batch Reading " ตัวอ่านข้อมูลควรมีความสามารถที่จะจัดลำดับการอ่านแท็กทีละตัวได้

เครื่องอ่านจะประกอบด้วยส่วนประกอบหลักดังนี้

- ภาครับและส่งสัญญาณวิทยุ
- ภาคสร้างสัญญาณพาหะ
- ขดลวดที่ทำหน้าที่เป็นสายอากาศ
- วงจรจูนสัญญาณ
- หน่วยประมวลผลข้อมูล และ ภาคติดต่อกับคอมพิวเตอร์

2.1.6.1 หลักการทำงานเบื้องต้นของระบบ

- ตัวอ่านข้อมูลจะปล่อยคลื่นแม่เหล็กไฟฟ้าออกมาตลอดเวลา และคอยตรวจจับว่ามีแท็กเข้ามาอยู่ในบริเวณสนามไฟฟ้าหรือไม่ หรือ อีกนัยหนึ่งก็คือการคอยตรวจจับว่ามีการมอดูเลตสัญญาณเกิดขึ้นหรือไม่
- เมื่อมีแท็กเข้ามาอยู่ในบริเวณที่มีสนามไฟฟ้า แท็ก จะได้รับพลังงานไฟฟ้าที่เกิดจากการเหนี่ยวนำของคลื่นแม่เหล็กไฟฟ้าเพื่อให้แท็กเริ่มทำงาน และจะส่งข้อมูลในหน่วยความจำที่ผ่านการมอดูเลตกับคลื่นพาหะ แล้วออกมาทางสายอากาศที่อยู่ภายในแท็ก
- คลื่นพาหะที่ถูกส่งออกมาจากแท็ก จะเกิดการเปลี่ยนแปลงแอมพลิจูด , ความถี่ หรือ เฟส ขึ้นอยู่กับวิธีมอดูเลต
- ตัวอ่านข้อมูลจะตรวจจับความเปลี่ยนแปลงของคลื่นพาหะ แปลงออกมาเป็นข้อมูล แล้วทำการถอดรหัสเพื่อนำข้อมูลไปใช้งานต่อไป

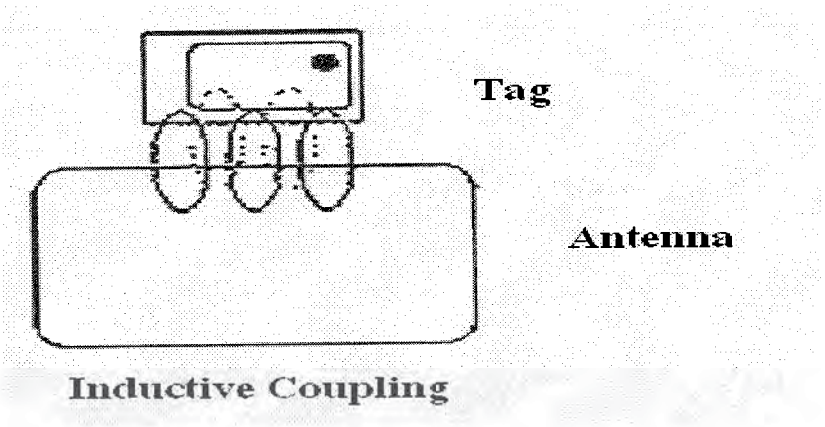
2.1.7 การสื่อสารแบบไร้สาย

การสื่อสารข้อมูลของระบบการชี้เฉพาะโดยคลื่นความถี่วิทยุ คือ ระหว่างคลื่นลูกข่ายและตัวอ่านข้อมูล จะสื่อสารแบบไร้สายผ่านอากาศ โดยจะนำข้อมูลมาทำการมอดูเลตกับคลื่นพาหะที่เป็นคลื่นความถี่วิทยุโดยมีสายอากาศ (Antenna) ที่อยู่ในตัวอ่านข้อมูลเป็นตัวรับและส่งคลื่น ซึ่งแบ่งออกเป็น 2 วิธี คือ

- วิธีเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้า (Inductive Coupling หรือ Proximity Electromagnetic)
- วิธีการแผ่คลื่นแม่เหล็กไฟฟ้า (Electromagnetic Propagation Coupling)

ดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



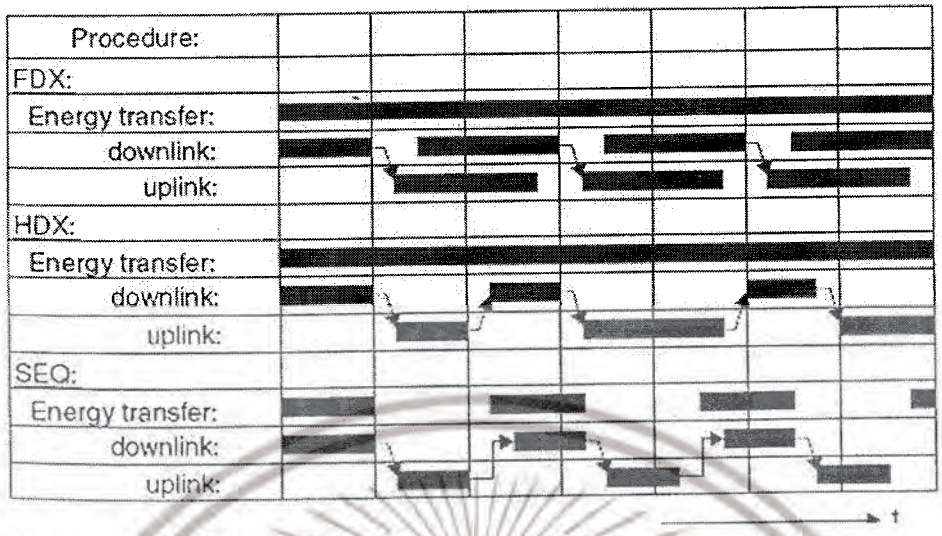
รูปที่ 2.9 การสื่อสารระหว่างเครื่องลูกข่ายและตัวรับข้อมูล

2.1.8 รูปแบบการรับส่งข้อมูล

มาตรฐาน RFID แบบที่ใช้ในการสื่อสารแบบ Full Duplex / Half Duplex และแบบ Sequential ซึ่งรูปแบบการสื่อสารทั้ง สามประเภทนี้ มีความแตกต่างจากมาตรฐาน RFID ประเภท 1 บิต ที่ใช้เทคโนโลยีง่าย ๆ ในการตรวจจับ โดยการเปลี่ยนสถานะลอจิกของอุปกรณ์เครื่องลูกข่าย RFID จาก " 0 " เป็น " 1 "

ซึ่งการรับส่งข้อมูลในระบบ RFID นี้จะมีรูปแบบการทำงานซับซ้อนกว่ารูปแบบการส่งข้อมูลประเภท 1 บิต นอกจากนั้นรูปแบบในการรับส่งข้อมูลระหว่างเครื่องลูกข่ายและเครื่องอ่านก็มีความแตกต่างกันออกไปตามกลวิธี ซึ่งสามารถแบ่งการทำงานออกได้ตามรูปที่ 2.10 อันประกอบไปด้วยการสื่สารแบบ Full Duplex ซึ่งเครื่องอ่านจะทำการส่งสัญญาณที่สามารถตรวจจับ และแปลงเป็นกำลังงานไฟฟ้าได้ตลอดเวลาการส่งข้อมูลจากเครื่องอ่านไปยังเครื่องลูกข่าย และการส่งข้อมูลจากเครื่องลูกข่ายไปยังเครื่องอ่าน สามารถกระทำขึ้นเมื่อใดก็ได้ ไม่จำเป็นต้องรอจังหวะในการส่งไปกลับแต่อย่างใด ต่างจากการสื่อสารแบบ Half Duplex ซึ่งแม่เครื่องอาจจะทำการส่งพลังงานไฟฟ้าไปให้กับเครื่องลูกข่ายตลอดเวลา แต่ก็ต้องมีการกำหนดจังหวะผลัดกันระหว่างเครื่องอ่านและเครื่องลูกข่าย สำหรับการสื่อสารแบบสุดท้าย คือ Sequential นั้น จะมีการจำกัดเวลาในการส่งพลังงานไฟฟ้าจากเครื่องอ่านไฟ ให้เครื่องลูกข่ายในลักษณะพัลซ์ (Pulse) หรือก้อนพลังงานที่ส่งออกในแต่ละช่วงเวลา และใช้การปรากฏขึ้นของพลังงาน ไฟฟ้า หรือพังซ์เป็นสัญญาณกำหนดให้มีการส่งข้อมูลจากเครื่องอ่านไปยังลูกข่าย ส่วนช่วงที่ไม่มีพลังงานปรากฏอยู่จะเป็นการส่งสัญญาณจากลูกข่ายไปยังเครื่องอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

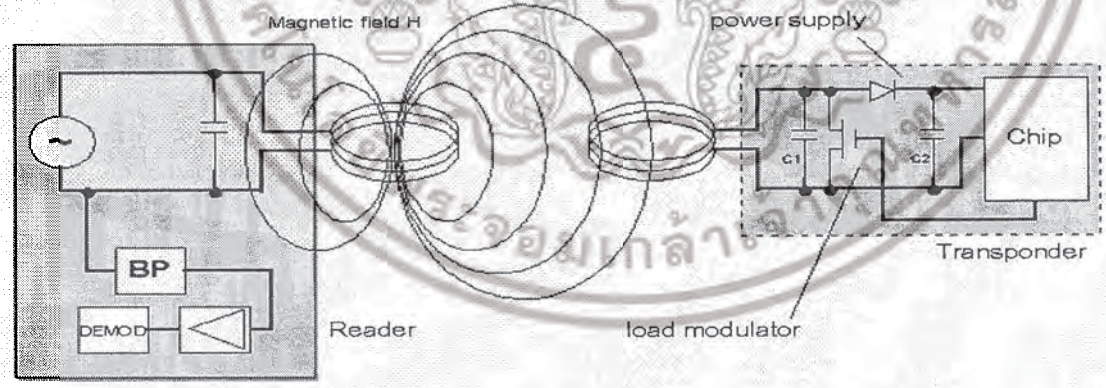


รูปที่ 2.10 ความแตกต่างของการสื่อสารแบบ Full Duplex / Half Duplex และ Sequential

2.1.9 เทคโนโลยี RFID ที่ใช้การสื่อสารแบบ Duplexing

2.1.9.1 เทคโนโลยี Inductive Coupling

เป็นมาตรฐานที่ใช้หลักการหนี่ยวนำทางคลื่นแม่เหล็ก โครงสร้างของเครื่องลูกข่ายประกอบไปด้วย แผลวงจรหรือไม่โครชิพที่ใช้เก็บบันทึกข้อมูลต่าง ๆ โดยมีขดลวดพื้นที่กว้างทำหน้าที่เป็นสายอากาศ สำหรับรับและส่งสัญญาณ ทั้งนี้จะกล่าวถึงเทคนิคในการจ่ายส่งพลังงานจากเครื่องลูกข่าย เพื่อเลี้ยงให้ไมโครชิพสามารถทำงานได้ก่อน จะกล่าวถึงเทคนิคที่ใช้ในการรับส่งข้อมูล ดังแสดงในรูป 2.11



รูปที่ 2.11 เทคโนโลยี Inductive Coupling กับการรับพลังงานไฟฟ้าจากเครื่องอ่าน

การส่งพลังงานจากเครื่องอ่าน

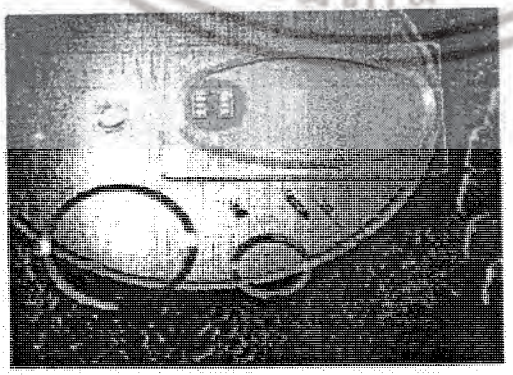
เนื่องจากการทำงานของเครื่องลูกข่ายนั้นเป็นแบบ passive กล่าวคือ ไม่มีแหล่งพลังงานภายในเป็นของตังเอง จึงจำเป็นต้องรับพลังงาน (Energy) มาจากเครื่องอ่านโดยตรง (ซึ่งทำหน้าที่เป็นทั้งเครื่องส่งข้อมูลและส่งพลังงานไปในตัว) ทำให้สามารถส่งพลังงานคลื่นแม่เหล็กไฟฟ้าที่กำลังสูง สามารถแพร่กระจายไปในพื้นที่ใช้งาน และสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตกกระทบขดลวดที่ทำหน้าที่เป็นสายอากาศของเครื่องลูกข่ายได้อย่างเหมาะสม ประกอบทั้งความยาวคลื่นของของสัญญาณคลื่นแม่เหล็กไฟฟ้าที่ส่งกระจายออกมา จากเครื่องอ่านนี้มีค่ามากกว่าระยะห่างระหว่างเครื่องลูกข่ายกับเครื่องอ่าน ซึ่งโดยทั่วไปมักวางห่างกันไม่มากนัก ในขณะที่เทคโนโลยี RFID แบบ Inductive Coupling ส่วนใหญ่ใช้ความถี่ต่ำเพียง 135 kHz หรืออาจต่ำกว่า ซึ่งคลื่อนแม่เหล็กไฟฟ้า จะมีความยาวคลื่นสูงมากถึง 2400 เมตร แม้ในบางระบบที่ใช้ความถี่สูงในย่าน 13.56 MHz ก็ยังกำเนิดคลื่นสัญญาณที่มีความยาวคลื่นถึง 22.1 เมตร จึงไม่มีปัญหา ในเรื่องของความยาวคลื่นที่สั้นเกินไป จึงเกิดปรากฏการณ์เลี้ยวเบน หรือ ถูกลดทอนจากผนังอาคาร หรือสิ่งก่อสร้างต่าง ๆ หากพิจารณาถึงพื้นที่ใช้งานโดยทั่วไป ที่ไม่ได้ไกลหรือกว้างใหญ่มากนัก

สัญญาณคลื่นแม่เหล็กไฟฟ้าส่วนหนึ่ง จะตกกระทบขดลวดสายอากาศของเครื่องลูกข่าย ก่อให้เกิดแรงดันไฟฟ้าขนาดอ่อน ๆ ขึ้นบนขดลวดดังกล่าว ซึ่งแรงดันไฟฟ้านี้ จะถูกนำเข้ากระบวนการ Rectifier โดยตัวเก็บประจุและไดโอด เพื่อกรองให้เป็นแรงดันไฟฟ้ากระแสตรงสำหรับป้อนให้กับไมโครชิพ ทั้งนี้หัวใจของการออกแบบอยู่ที่การเลือกค่าของตัวเก็บประจุ C1 ซึ่งจะต้องมีค่าสัมพันธ์กับค่าความเหนี่ยวนำของขดลวด (L) เพื่อให้สามารถกำหนดค่าความถี่กำทอน (Resonant Frequency) ที่ตรงกับความถี่ของคลื่นแม่เหล็กไฟฟ้า ที่ถูกส่งออกมาจากเครื่องอ่าน เพื่อให้เกิดความเหนี่ยวนำพลังงานไฟฟ้าสูงสุดที่เครื่องลูกข่าย สิ่งที่ผู้ออกแบบอุปกรณ์เครื่องอ่านและเครื่องลูกข่าย RFID แบบ Inductive Coupling ให้ความสำคัญมากที่สุด ก็คือการออกแบบขดลวดทั้งที่เครื่องอ่านและเครื่องลูกข่าย สามารถถ่ายทอดพลังงานไฟฟ้าให้แก่กันได้อย่างมีประสิทธิภาพมากที่สุด หากจินตนาการเปรียบเทียบการถ่ายทอดพลังงานไฟฟ้า ระหว่างขดลวดทั้งสองในรูปที่ 2.11 จะเทียบได้กับหม้อแปลงไฟฟ้าที่มีขดลวด 2 ชุด หากแต่ขดลวดทั้งสอง ไม่ได้พันอยู่บนแกนแม่เหล็กเดียวกัน กลับกลายเป็นใช้อากาศซึ่งมีคุณสมบัติในการนำพาสัญญาณที่แย่กว่าแกนแม่เหล็กมาก ๆ

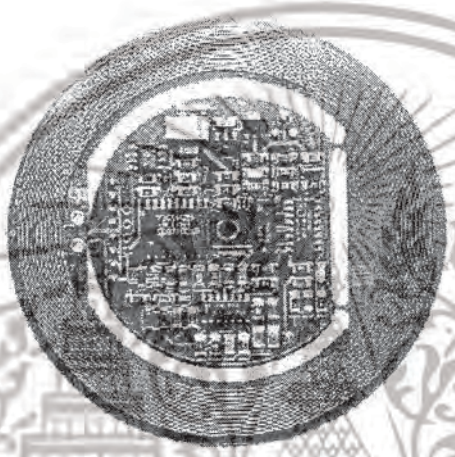
ดังนั้น ในการออกแบบระบบแบบ Inductive Coupling ให้ประสิทธิภาพในการถ่ายทอดพลังงานมากที่สุดจึงต้องเน้นไปที่ย่านความถี่ที่เหมาะสมต่อการส่งกระจายพลังงาน , จำนวนรอบของเครื่องอ่าน , พื้นที่หน้าตัดของขดลวดที่เครื่องลูกข่าย , มุมที่ติดตั้งและระยะห่างระหว่างเครื่องอ่านและเครื่องลูกข่าย ทั้งนี้ รูปที่ 2.12 แสดงตัวอย่างโครงสร้างพื้นฐานที่เป็นองค์ประกอบของอุปกรณ์ RFID แบบ Inductive Coupling โดยขดลวดวงกลมที่เห็นในภาพเป็นสายอากาศของเครื่องรับ ในขณะที่ขดลวดวงรีที่เป็นของเครื่องลูกข่าย จะติดตั้งอยู่บนแผ่นพลาสติกการ์ด พร้อมบรรจุวงจรอิเล็คทรอนิกส์ที่ทำหน้าที่เป็นวงจร Rectifier ร่วมกับแผงวงจรประมวลผลและเก็บบันทึกข้อมูลต่าง ๆ



รูปที่ 2.12 องค์ประกอบพื้นฐานของเครื่องอ่านและเครื่องลูกข่าย RFID ที่มีการทำงานแบบ Inductive Coupling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อความถี่ที่ใช้ในการสื่อสารมีค่าสูงมากขึ้น จะพบว่าค่าความเหนี่ยวนำที่ต้องใช้สำหรับขดลวดสายอากาศของเครื่องลูกข่ายมีค่าลดลง ซึ่งหมายถึงจำนวนรอบที่พันลดลง ซึ่งมีผลให้ สามารถลดขนาดของอุปกรณ์เครื่องลูกข่ายลงได้ แต่ขณะเดียวกันก็จะส่งผลให้แรงดันที่เกิดจากการเหนี่ยวนำของคลื่นแม่เหล็กไฟฟ้าในย่านความถี่สูงลดลงเมื่อเทียบกับการใช้งานในย่านความถี่ต่ำ ทำให้จำกัดระยะทางในการใช้งานมากขึ้น การเลือกใช้อุปกรณ์ Inductive Coupling จึงต้องพิจารณาทั้งขนาดของเครื่องลูกข่ายและระยะทางที่ต้องการใช้งานตรวจจับเป็นสำคัญ เพื่อใช้กำหนดเลือกย่านความถี่ที่เหมาะสมของอุปกรณ์ใช้งาน รูปที่ 2.13 เป็นตัวอย่างของเครื่องอ่านข้อมูล RFID แบบ Inductive Coupling ซึ่งจะเห็นว่ามีการผนึกขดลวดลงไปในบรรจุภัณฑ์ ทำให้มีขนาดเล็ก กะทัดรัด เหมาะสำหรับการติดตั้งใช้งานในพื้นที่จำกัด



รูปที่ 2.13 ตัวอย่างผลิตภัณฑ์เครื่องลูกกข่าย RFID ประเภท Inductive Coupling

การส่งข้อมูลจากเครื่องลูกข่ายมายังเครื่องอ่าน

มาตรฐาน RFID แบบ Inductive Coupling กำหนดรูปแบบการส่งข้อมูลจากเครื่องลูกข่ายมายังเครื่องอ่าน หรือ Uplink Communication โดยใช้เทคโนโลยีตัวเลือกที่เป็นการส่งกระจายคลื่นความถี่วิทยุออกเป็น 3 ประเภท คือ Load Modulation , Load Modulation with sub carrier และ Sub harmonic procedure ซึ่งอธิบายพอสังเขปได้ดังนี้

2.1.9.2 เทคโนโลยี Load Modulation

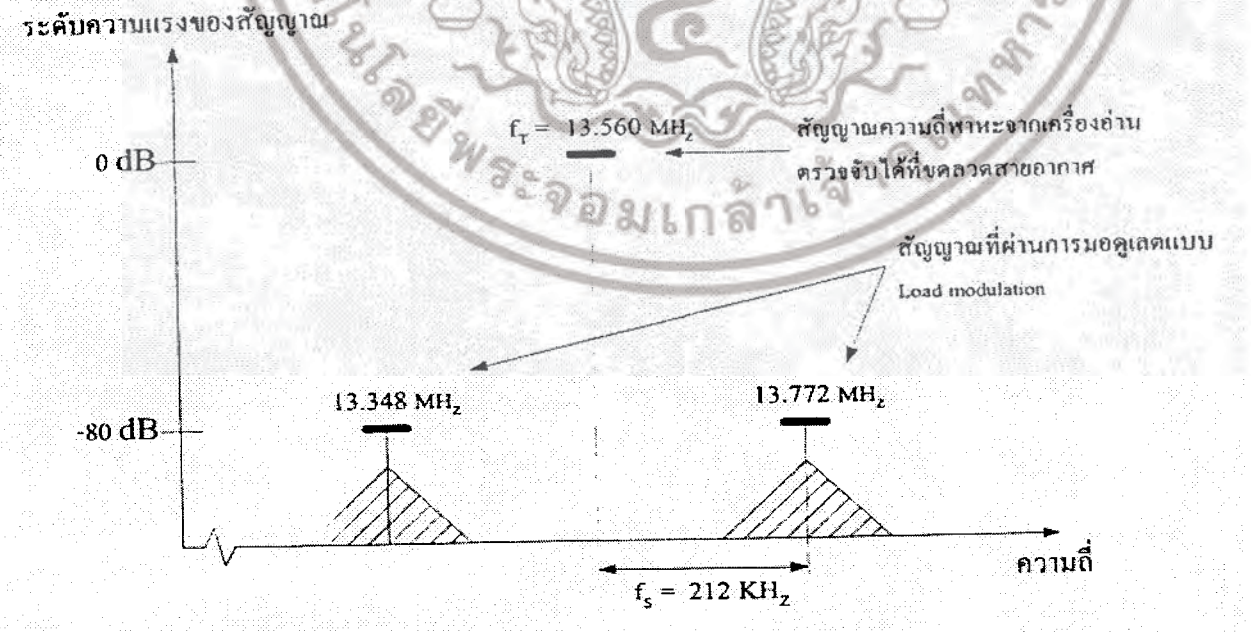
ดังได้กล่าวมาแล้วว่า พฤติกรรมในการส่งพลังงานและสัญญาณระหว่างขดลวดของเครื่องอ่านและขดลวดของเครื่องลูกข่าย มีแบบจำลองที่ไม่ต่างจากการทำงานของขดลวดหม้อแปลงไฟฟ้าแต่อย่างใด เพียงแต่ตัวกลางที่ใช้ในการถ่ายทอดพลังงานเป็นอากาศไม่ใช่โลหะหรือแท่งแม่เหล็กเหมือนในหม้อแปลงไฟฟ้าทั่วไป อย่างไรก็ตาม การใช้อากาศเป็นตัวกลางถ่ายทอดสัญญาณระหว่างขดลวดทั้งสองนั้น จะทำได้ก็ต่อเมื่อระยะห่างระหว่างขดลวดของเครื่องอ่านและเครื่องลูกข่ายห่างกันไม่เกิน 0.16 เท่าของค่าความยาวคลื่น (0.16λ) หากมีการนำเครื่องลูกข่ายซึ่งมีคุณสมบัติของขดลวดและตัวเก็บประจุ (LC) ตรงกับค่าความถี่กำทอนของเครื่องอ่านมาใช้ในพื้นที่ตรวจสอบที่มีการแพร่กระจายคลื่นแม่เหล็กไฟฟ้าจากเครื่องอ่าน เครื่องลูกข่ายย่อมดึงพลังงานไฟฟ้าผ่านกระบวนการเหนี่ยวนำเพื่อสร้างแรงดันไฟเลี้ยงให้กับอุปกรณ์อิเล็คทรอนิกส์ภายใน ซึ่งหากพิจารณาในแง่ของวงจรไฟฟ้าแล้ว เครื่องอ่านจะมองเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องลูกข่ายเป็นโหลดวงจรตัวหนึ่งที่มีค่าอิมพิแดนซ์เท่ากัน ZT ต่อพ่วงอยู่กัยขดลวดสายอากาศของเครื่องอ่านการเปลี่ยนแปลงใด ๆ ไม่ว่าจะเป็นการส่งหรือหยุดส่งพลังงานจากเครื่องอ่าน ย่อมมีผลต่อการเปลี่ยนแปลงค่าอิมพิแดนซ์ ZT ที่เครื่องอ่านมองเห็นอย่างแน่นอนเนื่องจากในวงจร Resonant ที่เกิดจากขดลวด (L) และตัวเก็บประจุ (C) บนเครื่องลูกข่ายย่อมมีค่าอิมพิแดนซ์เปลี่ยนแปลงไปตามความถี่ที่ตกกระทบ และไม่มีการจ่ายพลังงานก็เทียบเท่ากับความถี่เท่ากับศูนย์ ซึ่งการเปลี่ยนแปลงค่าอิมพิแดนซ์เสมือน ZT นี้ย่อมมีผลต่อการเปลี่ยนแปลงระดับแรงดันไฟฟ้าที่ตกตร่อมขดลวดสายอากาศของเครื่องอ่านโดยปริยาย ซึ่งนี่เองที่เป็นสัญญาณแจ้งให้เครื่องอ่านทราบว่ากำลังจะมีการส่งข้อมูลจากเครื่องลูกข่ายมา เรียกเทคโนโลยีนี้ว่า Half Duplex หรือ Full Duplex นั้นก็สุดแท้แต่การออกแบบวงจรตรวจจับและถอดรหัสสัญญาณซึ่งเป็นเรื่องที่ลงรายละเอียดไปในการออกแบบ

2.1.9.3 เทคโนโลยี Load Modulation with sub carrier

เนื่องจากระดับการความสามารถในการถ่ายทอด (Coupling) สัญญาณระหว่างขดลวดสายอากาศของเครื่องลูกข่าย และเครื่องอ่านค่อนข้างต่ำ การเปลี่ยนแปลงระดับสัญญาณที่เกิดขึ้นบนขดลวดของเครื่องอ่าน อันมีผลมาจากการส่งข้อมูลจากเครื่องลูกข่าย ย่อมอยู่ในระดับที่ต่ำมาก ๆ จนบางครั้งอาจไม่สามารถตรวจจับได้ เป็นผลให้เกิดปัญหาในการส่งข้อมูลจากเครื่องลูกข่ายมายังเครื่องอ่าน หากคิดในแง่ของการออกแบบ วงจรเพื่อตรวจจับสัญญาณการเปลี่ยนแปลงดังกล่าว ก็ต้องกล่าวว่าสัญญาณดังกล่าวมีระดับความแตกต่าง ระหว่างตัวสัญญาณจริงกับสัญญาณรบกวนต่ำมาก การใช้เทคโนโลยี Load Modulation โดยทั่วไปย่อมมีโอกาสตรวจจับข้อมูลได้ผิดพลาดง่ายมาก จึงเป็นที่มาของการออกแบบเทคนิคการตรวจจับข้อมูลแบบ Load Modulation with sub carrier ซึ่งมีหลักการคล้ายคลึงกับการรับส่งวิทยุ AM ในวงจรภายในเครื่องลูกข่าย จะทำการมอดูเลตสัญญาณทำให้ปรากฏคลื่นความถี่ไซด์แบนด์ หรือความถี่ข้างเคียงอันมีค่าเท่ากับ $f^T \pm f^s$ โดย f^T คือ ความถี่ในการติดต่อสื่อสาร ระหว่างเครื่องอ่านและเครื่องลูกข่าย ซึ่งมีรายละเอียดดังรูปที่ 2.14

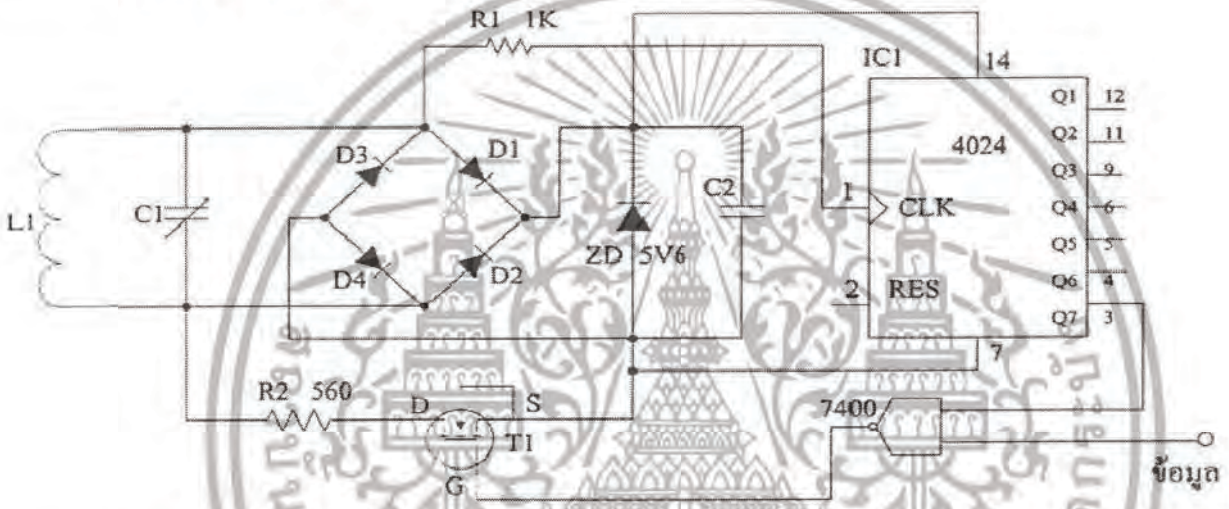


รูปที่ 2.14 สัญญาณข้อมูลที่ถูกรบกวนที่ถูกบรรจุอยู่ในไซด์แบนด์ของความถี่พื้นฐานที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องอ่านกับเครื่องลูกข่าย RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เมื่อเป็นเช่นนี้ การตรวจจับสัญญาณที่ปรากฏบนเครื่องอ่านก็จะง่ายขึ้น โดยเพียงแต่ทำการติดตั้งอุปกรณ์แยกกรองความถี่ (Band pass filter) ที่ขดลวดของเครื่องอ่านโดยสามารถเลือกเฉพาะสัญญาณในไซด์แบนด์ด้านใดด้านหนึ่งที่ต้องการ จากนั้นจึงนำสัญญาณที่กรองออกมาทำการขยาย แล้วทำการดีมอดูเลตก็จะได้สัญญาณข้อมูลที่ส่งออกมาจากเครื่องลูกข่าย อย่างไรก็ตามเนื่องจากความถี่ในการส่งข้อมูล จากเครื่องลูกข่ายมายังเครื่องอ่าน ทำให้ต้องมีการกันแบนด์วิดธ์สำหรับการรับส่งข้อมูลไว้ค่อนข้างสูง ดังตัวอย่างในรูปที่ 2.15 ซึ่งต้องกันแบนด์วิดธ์รอบความถี่พื้นฐานมากถึง 212 kHz ทำให้ไม่สามารถใช้เทคโนโลยีนี้ได้กับ ระบบ RFID ที่ใช้ความถี่ต่ำในการสื่อสารข้อมูลได้ ข้อกำหนดมาตรฐาน RFID ระบุว่าสามารถใช้เทคโนโลยีดังกล่าวได้เฉพาะกับมาตรฐาน RFID แบบ Inductive Coupling ที่ความถี่ 6.78 MHz , 13.56 MHz และ 27.125 MHz เท่านั้น รูปที่ 2.15 เป็นตัวอย่างวงจรจริงที่ใช้ในการสร้างสัญญาณไซด์แบนด์บนอุปกรณ์เครื่องลูกข่าย



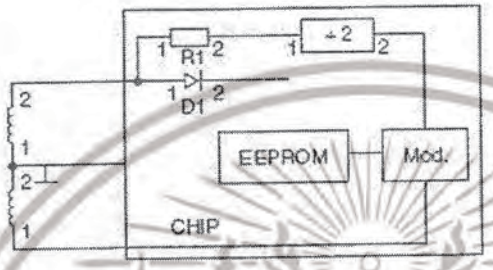
รูปที่ 2.15 ตัวอย่างวงจรที่ใช้สำหรับเทคนิคการส่งสัญญาณแบบ Load Modulation with Sub carrier ติดตั้งอยู่ในเครื่องลูกข่าย RFID แบบ Inductive Coupling

2.1.9.4 เทคโนโลยี Subharmonic Procedure

เป็นอีกทางเลือกหนึ่งที่ใช้งานในระบบ RFID ในย่านความถี่ต่ำและไม่สามารถนำเทคโนโลยี Load Modulation with sub carrier มาใช้งานได้ เทคโนโลยีนี้เป็นการนำสัญญาณข้อมูลที่เครื่องลูกข่ายต้องการส่งกลับไปยังเครื่องอ่าน มาทำการป้อนผ่านอุปกรณ์อิเล็กทรอนิกส์ที่มีการทำงานแบบไม่เป็นเชิงเส้นเพื่อทำให้เกิดการกระจายสัญญาณออกเป็นสัญญาณ ฮาร์โมนิกย่อย ๆ หรืออาจใช้การป้องกันสัญญาณที่ต้องการจะส่งออกไปยังวงจรหารความถี่ เพื่อให้เกิดสัญญาณข้อมูลที่บรรจุข้อมูลข่าวสารเหมือนกับสัญญาณต้นฉบับ แต่มีความถี่ลดลง 2 เท่า จากนั้นจึงนำสัญญาณใหม่ที่มีความถี่ต่ำลงนี้ไปผ่านเข้าสู่วงจรมอดูเลตสัญญาณ เพื่อป้อนเข้าสู่ขดลวดของเครื่องลูกข่ายได้มีการส่งผ่านอากาศกลับไปยังเครื่องอ่าน โดยทั่วไปมักนิยมใช้เทคโนโลยี Subharmonic Procedure กับระบบ RFID ที่ย่านความถี่ 128 kHz ย่อมหมายความว่าเครื่องลูกข่ายชนิดนี้ จะส่งสัญญาณข้อมูลกลับไปยังเครื่องอ่านที่ความถี่ครึ่งหนึ่ง คือ 64 kHz การแยกแยะสัญญาณจากเครื่องลูกข่ายกับแรงดันไฟฟ้าที่ล้วนปรากฏที่ขดลวดสายอากาศของเครื่องรับสัญญาณ สามารถทำได้ง่ายโดยการใช้วงจรแยกกรองความถี่ เช่นเดียวกับในกรณีของเทคโนโลยี Load Modulation with sub carrier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
103011

รูปที่ 2.16 เป็นตัวอย่างวงจรภายในอย่างง่ายของเครื่องลูกข่ายแบบ Subharmonic Procedure โดยมีการแยกขดลวดสายอากาศออกเป็นสองส่วน มีแท็ปกลาง (Center Tap) เป็นตัวแยกสัญญาณจากขดลวดด้านบนจะถูกดึงไปกรองเป็นแรงดันไฟเลี้ยงเพื่อป้อนให้กับไมโครชิปบนเครื่องลูกข่ายในขณะที่สัญญาณซึ่งผ่านการหารความถี่ครึ่งหนึ่งจากความถี่ที่ใช้งานที่ได้รับจากเครื่องอ่านนั้นจะถูกนำไปมอดูเลตกับข้อมูลที่มีการส่งออกมาจากไมโครชิป (ในที่นี้คือ EPROM) แล้วป้อนกลับเข้าสู่ขดลวดด้านล่างเพื่อส่งย้อนกลับไปยังเครื่องลูกข่าย



รูปที่ 2.16 โครงสร้างวงจรภายในอย่างง่ายของเครื่องลูกข่าย RFID ที่ทำงานแบบ Sub harmonic

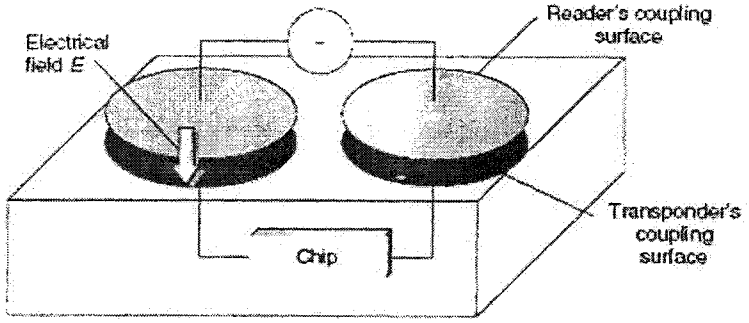
2.1.9.5 เทคโนโลยี close Coupling

เป็นเทคโนโลยีที่ใช้สำหรับการส่งข้อมูลจากเครื่องลูกข่ายไปยังเครื่องอ่าน RFID ในระยะทางสั้น ๆ ช่วง 0.1 ถึง 1 เซนติเมตร โดยในทางปฏิบัติจะพบมากในกรณีของระบบขนส่งมวลชน เช่น ระบบรถไฟฟ้า MRT ในประเทศไทย ซึ่งผู้โดยสารหยิบบัตรพลาสติก ซึ่งเป็นเครื่องลูกข่าย RFID ลักษณะนี้ไปแตะหรือจ่อกันเครื่องอ่าน เพื่อให้เครื่องอ่านตรวจสอบหมายเลขบัตร หรือบางระบบอาจมีการเขียนอ่านข้อมูลวงเงินคงเหลือ ก่อนจะพิจารณาว่าอนุญาตให้ผู้โดยสารผ่านเข้าสู่ระบบขนส่งมวลชนหรือไม่ ในทางเทคนิคนิยมเรียกรูปแบบการใช้งานนี้ว่า " Touch & Go "

การส่งพลังงานจากเครื่องอ่าน

เมื่อมีการสอดหรือวางบัตรพลาสติกที่เป็นเครื่องลูกข่าย RFID ชนิดนี้เข้าไปในหรืออยู่บนเครื่องอ่าน ในทางเทคนิคจะหมายถึงการวางตำแหน่งขดลวดของเครื่องลูกข่ายให้อยู่ในระยะทำการ โดยมีอากาศเป็นตัวกลางกั้นระหว่างเครื่องลูกข่ายและเครื่องอ่านในระยะสั้น ๆ เท่านั้น ตัวอย่างดังแสดงในรูป 2.17 ในกรณีนี้ ขดลวดสายอากาศของเครื่องอ่านจะทำหน้าที่เสมือนเป็นขดลวดปฐมภูมิ ในขณะที่ขดลวดสายอากาศของเครื่องลูกข่ายจะเป็นดังเช่นขดลวดทุติยภูมิ เมื่อเปรียบเทียบกับการทำงานของหม้อแปลงไฟฟ้าภายในเครื่องอ่านจะมีการกำเนิดสัญญาณไฟฟ้ากระแสสลับความถี่สูง เพื่อป้อนผ่านขดลวดปฐมภูมิ ก่อให้เกิดสนามแม่เหล็กความเข้มสูงเหนี่ยวนำผ่านช่องว่างของอากาศไปทำให้เกิดกระแสไฟฟ้าขึ้นบนขดลวดทุติยภูมิบนเครื่องลูกข่าย ก่อให้เกิดการจ่ายแรงดันไฟฟ้าเข้าสู่ไมโครชิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 เทคนิคการถ่ายโอนข้อมูลจากเครื่องลูกข่ายไปสู่เครื่องอ่านโดยผ่านการคัปปลิ้งทางไฟฟ้า

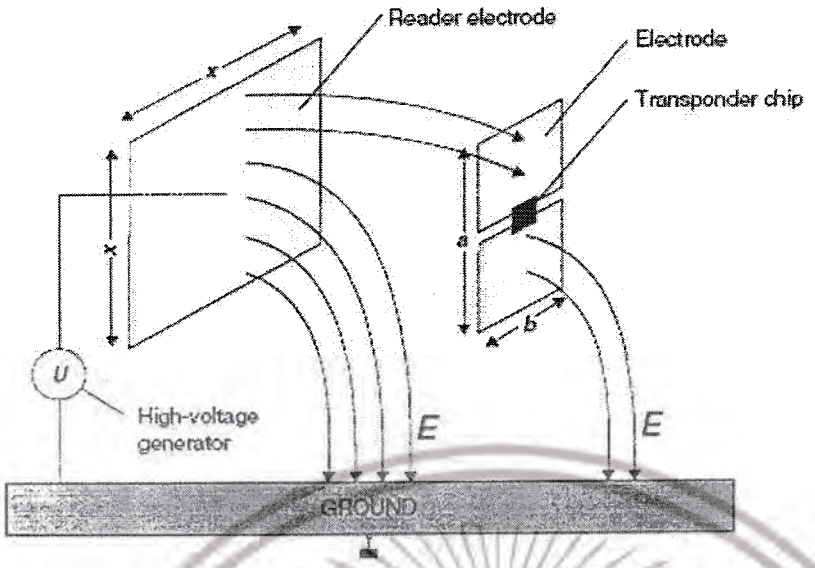
เนื่องจากระดับแรงดันไฟฟ้าที่จะถูกเหนี่ยวนำให้เกิดขึ้นบนขดลวดทุติยภูมิ มีค่าแปรผันโดยตรงกับความถี่ที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องอ่านกับเครื่องลูกข่าย ดังนั้น ในทางปฏิบัติจึงนิยมกำหนดความถี่ใช้งานให้สูงสุดเท่าที่เทคโนโลยีจะสามารถรับรองได้ ซึ่งในปัจจุบันก็ได้แก่ความถี่ในช่วง 1 ถึง 10 MHz นอกจากนั้นยังมีการเลือกใช้วัสดุประเภทฟลอไรท์ นำมาทำเป็นแกนสำหรับพันขดลวดของเครื่องอ่าน เนื่องจากเฟอร์ไรท์มีค่าความสูญเสียของการนำพาสนามแม่เหล็กต่ำกว่าวัสดุประเภทอื่น ๆ และเนื่องจากคุณสมบัติที่ดีที่สุดในการถ่ายทอดพลังงานเมื่อเปรียบเทียบกับเทคโนโลยีแบบ Inductive Coupling หรือเทคโนโลยี RFID ที่ใช้คลื่นไมโครเวฟเป็นตัวกลางในการติดต่อสื่อสาร เทคโนโลยีแบบ Close Coupling จึงได้รับความนิยมใช้งานมากที่สุดสำหรับเครื่องลูกข่ายที่ต้องการกำลังไฟฟ้าสูง ๆ เพื่อป้อนให้กับไมโครชิป ซึ่งในทางปฏิบัติสามารถป้อนกำลังไฟให้กับไมโครชิปที่กินไฟมากถึง 10 มิลลิวัตต์ได้อย่างไม่มีปัญหาใด ๆ ทั้งสิ้น ทั้งนี้มีการวางข้อกำหนดทางกลศาสตร์ และทางไฟฟ้าของแผ่นเครื่องลูกข่ายแบบ Close Coupling ไว้ตามมาตรฐาน ISO 10536

การส่งข้อมูลจากเครื่องลูกข่ายมายังเครื่องอ่าน

ในการส่งข้อมูลจากเครื่องลูกข่ายแบบ Close Coupling มายังเครื่องอ่านนั้น สามารถทำได้สองวิธีด้วยกัน คือ ใช้เทคโนโลยี Load Modulation with subcarrier ดังที่ได้กล่าวไปแล้วในกรณีมาตรฐานแบบ Inductive Coupling หรืออาจใช้เทคโนโลยี Capacitive Coupling ซึ่งอาศัยประโยชน์จากระยะห่างที่ค่อนข้างสั้นระหว่างขดลวดสายอากาศของเครื่องลูกข่ายและเครื่องอ่าน โดยมีการติดตั้งแผ่นตัวเก็บประจุ (Plate Capacitor)

ดังแสดงในรูปที่ 2.18 ทั้งบัตรพลาสติกบนเครื่องลูกข่ายและ บนเครื่องอ่านในลักษณะที่ให้หน้าสัมผัสคัปปลิ้ง (Coupling Surface) ของทั้งเครื่องลูกข่ายและเครื่องอ่านอยู่ในแนวเดียวกัน ในยามที่มีการสัมผัสหรือสอดเครื่องลูกข่ายเข้าไปในเครื่องอ่าน แล้วใช้การเหนี่ยวนำทางไฟฟ้าเป็นตัวส่งผ่านข้อมูลจากไมโครชิปบนเครื่องลูกข่ายผ่านสนามไฟฟ้าไปสู่เครื่องอ่าน

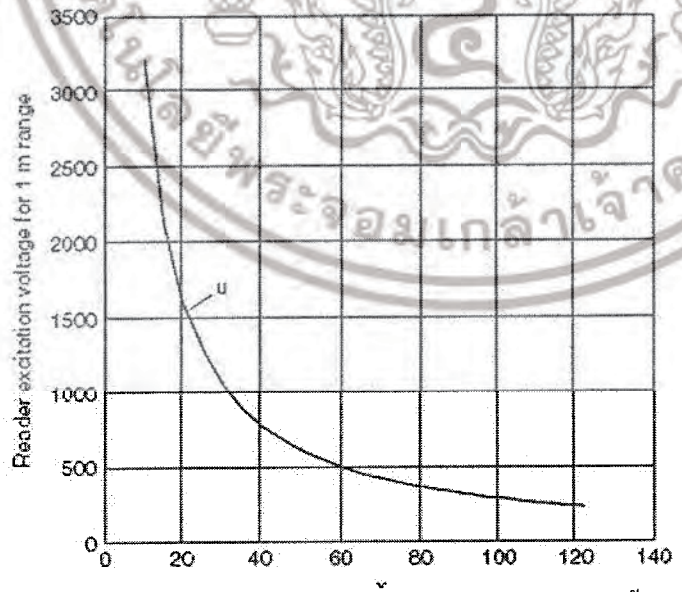
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 โครงสร้างและการทำงานของระบบ RFID แบบ Electrical Coupling

2.1.9.6 เทคโนโลยี Electrical Coupling

เป็นเทคโลยี RFID ชนิด Full Duplex / Half Duplex อีกประเภทหนึ่งที่ใช้หลักการสร้างสนามไฟฟ้า (Electric field) ขึ้นจากเครื่องอ่านแล้วใช้คุณสมบัติของสนามไฟฟ้าในระยะทางใกล้ ๆ สายอากาศของเครื่องอ่านซึ่งออกแบบให้มีลักษณะเป็นแผ่นอิเล็กทรอนิกขนาดใหญ่ ในการสร้างความต่างศักย์ไฟฟ้าให้เกิดขึ้นบนสายอากาศของเครื่องลูกข่ายที่มีการจัดสร้างเป็นแผ่นอิเล็กโทรดเช่นเดียวกัน



รูปที่ 2.19 ความสัมพันธ์ระหว่างระดับแรงดันไฟฟ้าที่จะต้องผลิตขึ้นบนแผ่นอิเล็กโทรด ของเครื่อง RFID โดยสัมพันธ์กับขนาดของแผ่นอิเล็กโทรด ในกรณีที่นำเครื่องลูกข่ายเข้ามาใช้งานในระยะห่าง 1 เมตร จากเครื่องอ่านใช้ความถี่ 125 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

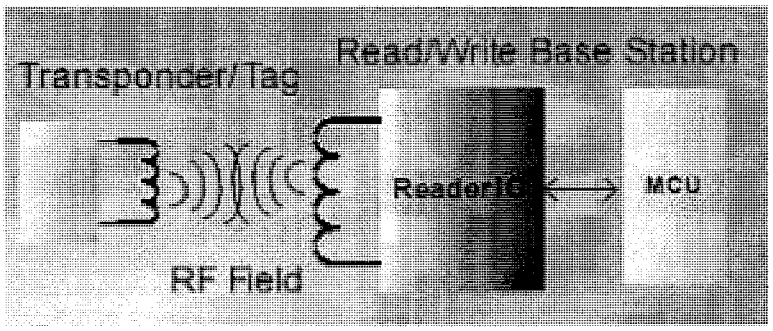
การส่งพลังงานจากเครื่องอ่าน

จากรูปที่ 2.19 เครื่องอ่านได้รับการออกแบบให้ต่อเชื่อมกับสายอากาศที่มีลักษณะเป็นอิเล็กโทรดขนาดใหญ่ โดยทั่วไปมักทำจากแผ่นโลหะหรือแผ่นฟอยล์โลหะ จากนั้นจึงทำการป้อนแรงดันไฟฟ้ากระแสสลับความถี่สูงเข้าแผ่นอิเล็กโทรด เนื่องจากลักษณะทางกานภาพของแผ่นอิเล็กโทรดที่มีค่าความเหนี่ยวนำภายใน (L) และค่าคาปาซิแตนท์ (เทียบได้กับตัวเก็บประจุ) ระหว่างแผ่นอิเล็กโทรดกับกราวน์หรือพื้นโลก จึงทำให้เกิดวงจรเสมือนของแผ่นอิเล็กโทรดที่เทียบได้กับวงจร LC ซึ่งมีความถี่กำทอน (Resonant Frequency) เป็นคุณสมบัติประจำตัว หากทำการป้อนสัญญาณแรงดันไฟฟ้าที่มีความถี่ตรงกันกับความถี่กำทอนของแผ่นอิเล็กโทรด ก็ทำให้แรงดันไฟฟ้าที่ปรากฏขึ้นบนแผ่นอิเล็กโทรดนั้นมีระดับสูงขึ้นมากอันเนื่องมาจากปรากฏการณ์เรโซแนนท์ ซึ่งเป้าหมายในการใช้งานจริงมักกำหนดให้แรงดันไฟฟ้าบนแผ่นอิเล็กโทรดเทียบกับกราวน์มีค่าสูงตั้งแต่หลักร้อยถึงหลักพันโวลต์ แรงดันดังกล่าวก่อให้เดิสนามไฟฟ้าระหว่างแผ่นอิเล็กโทรดกับกราวน์ที่วนอยู่รอบ ๆ

เมื่อนำเครื่องลูกข่ายที่ได้รับการออกแบบให้มีสายอากาศ 2 ชุด เป็นแผ่นอิเล็กโทรดขนาดเล็กกว่า เข้ามาอยู่ในพื้นที่ใช้งานอันมีสนามไฟฟ้าจากเครื่องอ่านปรากฏอยู่ จะทำให้เกิดการเหนี่ยวนำให้เกิดความต่างศักย์ระหว่าอิเล็กโทรดแผ่นบนและแผ่นล่างของเครื่องลูกข่ายดังแสดงในรูปที่ 2.20 ก่อให้เกิดกระแสไหลเวียนอยู่ภายในวงจรของเครื่องลูกข่ายเพื่อไปเลี้ยงให้กับไมโครชิปรูปที่ 2.20 แสดงให้เห็นถึงระดับแรงดันไฟฟ้าที่เครื่องอ่านจะต้องสร้างขึ้น เพื่อให้เพียงพอที่จะเหนี่ยวนำจนทำให้เกิดแรงดันไฟฟ้าเพียงพอแก่เครื่องลูกข่ายที่อยู่ห่างจากแผ่นอิเล็กโทรดของเครื่องอ่านประมาณ 1 เมตร โยกำหนดว่าแผ่นอิเล็กโทรดของเครื่องลูกข่ายมีขนาดประมาณ 4.5 x 7 เซนติเมตร ซึ่งเป็นขนาดมาตรฐานของแผ่น Smartcard ทั่วไป ใช้ความถี่ในการกำเนิดสัญญาณ 125 kHz จะเห็นได้ว่ายิ่งขนาดของแผ่นอิเล็กโทรดของเครื่องอ่านมีขนาดเล็กลงเท่าไร ก็ยิ่งต้องทำให้มีแรงดันไฟฟ้าปรากฏบนแผ่นอิเล็กโทรดสูงขึ้นเท่านั้น จึงเป็นสาเหตุที่นิยมออกแบบให้เครื่องอ่านมีอิเล็กโทรดขนาดใหญ่เพื่อเป็นการประหยัดพลังงานในการส่งออก

รูปที่ 2.20 แสดงวงจรสมมูล (Equivalent Circuit) ของการเชื่อมต่อระหว่างเครื่องอ่านและเครื่องลูกข่าย โดยเครื่องอ่านกำเนิดแรงดันไฟฟ้าด้วยค่าแรงดัน U ความถี่ f โดยแผ่นอิเล็กโทรดของเครื่องอ่านแสดงพฤติกรรมของวงจรเรโซแนนซ์ L^1 และ C^1 อากาศซึ่งทำหน้าที่เป็นตัวกลางของเครื่องอ่านและเครื่องลูกข่ายมีคุณสมบัติเป็นตัวเก็บประจุ ค่า C^{R-T} โดยมีค่าคาปาซิแตนท์ระหว่างแผ่นอิเล็กโทรดของเครื่องอ่านกับกราวน์ C^{R-GND} และ คาปาซิแตนท์ระหว่างแผ่นอิเล็กโทรดของเครื่องลูกข่ายกับกราวน์ C^{T-GND} ส่วน RL เป็นค่าอินพุตอิมพิแดนซ์ ของเครื่องลูกข่าย ซึ่งตามวงจรสมมูลนี้จะเห็นว่ามีลักษณะเป็นวงจรแบ่งแรงดัน (Voltage Divider Circuit) โดยแรงดันสุดท้ายที่จะเกิดขึ้นบนเครื่องลูกข่าย ซึ่งก็คือแรงดันตกคร่อม R^L จะมีค่าขึ้นอยู่กับสัดส่วนของแรงดันที่แบ่งระหว่างค่าอิมพิแดนซ์ของ C^{R-T} และ R^L นั่นเอง สมการดังกล่าวจึงเป็นจุดเริ่มต้นสำหรับใช้คำนวณหาระดับแรงดีนที่เหมาะสมในการกระตุ้นให้โครชิปบนเครื่องลูกข่ายทำงานได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 วงจรสมมูลระหว่างเครื่องลูกข่ายและเครื่องอ่าน RFID แบบ Electrical Coupling

การส่งข้อมูลจากเครื่องลูกข่ายมายังเครื่องอ่าน

เมื่อมีการนำเครื่องลูกข่ายมาไว้ในพื้นที่ใช้งาน อันหมายถึงบริเวณที่มีสนามไฟฟ้าจากแผ่นอิเล็กโทรดของเครื่องอ่านปรากฏอยู่ จะมีผลทำให้ค่าอินพุทอิมพิแดนซ์ของเครื่องลูกข่าย (R^{L}) มีค่าเปลี่ยนแปลงไปตามค่าความถี่กำทอน โดยมีการแบ่งแรงดันไฟฟ้าผ่านมาทางค่าคาปาซิแตนท์ C^{R-T} ดังแสดงในรูป 2.19 อย่างไรก็ตามเมื่อมีการเพิ่มค่าอิมพิแดนซ์ R^{Mod} ซึ่งเครื่องลูกข่ายใช้ในการส่งสัญญาณไบนารี่ที่เป็นข้อมูลป้อนกลับไปยังเครื่องอ่าน โดยผ่านทางการสวิทชิ่ง เพื่อเพิ่มค่าอิมพิแดนซ์ R^{Mod} ซึ่งจะทำให้พฤติกรรมการสะท้อนสัญญาณผ่านทางวงจรแบ่งแรงดันมีค่าเปลี่ยนแปลงไป เช่น เมื่อเครื่องลูกข่ายต้องการส่งข้อมูลไบนารี่บิตใดบิตหนึ่งที่มีค่าเป็น “0” ก็จะทำการเปิดวงจรสวิทชิ่งออก ทำให้ค่าอินพุทอิมพิแดนซ์ของเครื่องลูกข่ายเป็น R^{L} ตามปกติ ในขณะที่จะปิดวงจรเพื่อเพิ่มค่า R^{Mod} ในกรณีที่ต้องการส่งบิตข้อมูลที่มีค่า “1” ทำให้ค่าอินพุทอิมพิแดนซ์ของเครื่องลูกข่ายลดลง ผลที่เกิดขึ้นจะทำให้เกิดการมอดูเลตสัญญาณคลื่นความถี่วิทยุจากการเปลี่ยนแปลงอินพุทอิมพิแดนซ์นี้ โดยทำให้เกิดผลการเปลี่ยนแปลงระดับแรงดันและสามารถตรวจจับได้ที่แผ่นอิเล็กโทรดของเครื่องอ่าน ซึ่งก็คือสายอากาศภาคส่งซึ่งการทำงานในลักษณะนี้จัดได้ว่าเป็นการส่งข้อมูลแบบ Load Modulation ประเภทหนึ่ง

การส่งข้อมูลจากเครื่องอ่านไปสู่เครื่องลูกข่าย

นับตั้งแต่ที่ได้กล่าวมาในตอนต้น อธิบายถึงหลักการในการส่งพลังงานจากเครื่องอ่านไปยังเครื่องลูกข่าย และกล่าวรายละเอียดถึงการส่งข้อมูลจากเครื่องลูกข่ายกลับมายังเครื่องอ่าน ซึ่งถือเป็นการสื่อสารในทิศทาง Uplink สำหรับเทคโนโลยี RFID แบบ Full Duplex / Half Duplex แต่ละประเภท สำหรับการส่งข้อมูลในทิศทาง Downlink จากเครื่องอ่านไผสู่เครื่องลูกข่ายนั้นจะใช้วิธีการเดียวกันทั้งสิ้น คือการมอดูเลตสัญญาณที่เครื่องอ่านต้องการส่งไปยังเครื่องลูกข่ายโดยตรง โดยไม่สรใจว่ารูปแบบการสื่อสารนั้นจะเป็นแบบ Full Duplex หรือ Half Duplex และไม่มีประเด็นเกี่ยวข้องกับความถี่ที่ใช้งานทั้งสิ้น สำหรับเทคโนโลยีที่สามารถใช้ในการมอดูเลตสัญญาณในทิศทาง Downlink สามารถทำได้ทั้งแบบ ASK , FSK , PSK แต่ที่ได้รับความนิยมมากที่สุดอันเนื่องมาจากความง่ายและสะดวกในการออกแบบวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.10 การนำระบบ RFIDไปใช้งาน

เราสามารถนำระบบ RFID ไปใช้งานได้หลากหลาย ไม่ว่าจะเป็นในอุตสาหกรรมการผลิต การค้า หรือการบริการต่างๆ ซึ่งสามารถบันทึกข้อมูลที่ต้องการได้ เช่น บันทึกเวลาทำงานของพนักงาน เก็บเงินค่าใช้บริการทางด่วน หรือระบบกันขโมยรถยนต์ แต่การพิจารณานำระบบ RFID มาใช้งานยังคงต้องคำนึงถึงข้อจำกัดต่างๆ ในการใช้งานไม่ว่าจะเป็นเรื่องของสนามแม่เหล็กไฟฟ้าในสภาพแวดล้อม หรือกฎหมายที่เกี่ยวข้องกับระเบียบการใช้คลื่นความถี่วิทยุและกำลังส่งของแต่ละประเทศ

2.2 สเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล โดยพลังงานทางไฟฟ้าที่ป้อนให้จะมีลักษณะเป็นไบนารีโวลต์เตจและเอาต์พุตมีลักษณะเป็นการเคลื่อนที่เชิงมุม คือ เมื่อป้อนสัญญาณพัลซ์ด้วยความถี่และลำดับที่เหมาะสมให้กับขดลวดสเตเตอร์ มอเตอร์จะหมุนเป็นจังหวะตามพัลซ์ที่ป้อนเข้ามา ซึ่งต่างจากมอเตอร์ทั่วไปที่จะหมุนทันทีและตลอดเวลาเมื่อป้อนแรงดันไฟฟ้า ข้อดีของสเต็ปปิ้งมอเตอร์ คือ สามารถกำหนดตำแหน่งของการหมุนด้วยตัวเลข(องศาหรือระยะทาง)ได้อย่างละเอียด โดยใช้คอมพิวเตอร์เป็นเครื่องกำหนดและจัดเก็บตัวเลขโดยไม่มีค่าตำแหน่งผิดพลาด(Position Error) สะสม นอกจากนี้ยังไม่จำเป็นต้องมีการป้อนกลับของตำแหน่งหรือความเร็ว

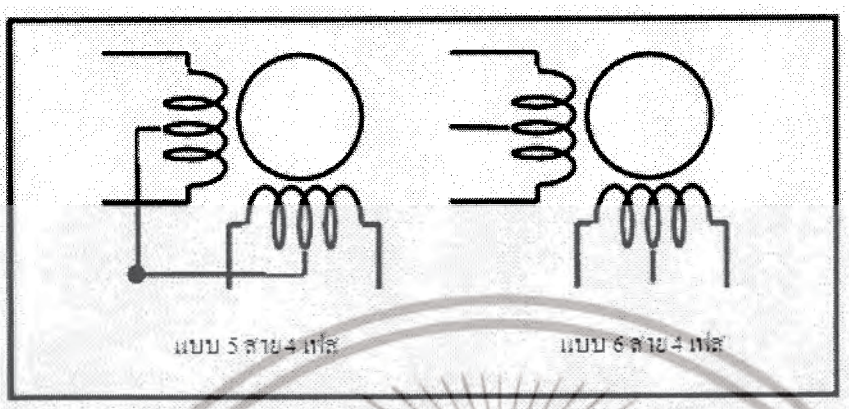
รูปที่ 2.21 บล็อกไดอะแกรมแสดงการควบคุมสเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์ได้รับการพัฒนามาอย่างต่อเนื่อง จนในปัจจุบันสเต็ปมอเตอร์ที่นิยมใช้อย่าง แพร่หลายมากที่สุด และหาได้ง่ายคือ สเต็ปปิ้งมอเตอร์แบบยูนิโพลาร์ (uni-polar stepper motor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 สเต็ปปิ้งมอเตอร์แบบยูนิโพลาร์ (uni-polar stepper motor)

มีลักษณะการพันขดลวดของมอเตอร์แสดงในรูปที่ 2.2



รูปที่ 2.22 ลักษณะการพันขดลวดของมอเตอร์แบบยูนิโพลาร์

รูปที่ 2.23 การวางของขดลวดแต่ละเฟสของสเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์แบบนี้มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ แต่ละขดแบ่งเป็น 2 เฟส รวมมอเตอร์ทั้งตัวจะมี 4 เฟสคือ เฟส 1,2,3 และ 4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่าย ไฟเลี้ยง ทำให้สเต็ปปิ้งมอเตอร์แบบนี้มีทั้งแบบ 5 สายและ 6 สาย ถ้าเป็นแบบ 5 สาย จะเป็นการนำ สายไฟเลี้ยงของขดลวดทั้งสองมาต่อรวมกันเป็นสายเดียว

การกระตุ้นและควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดทีละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นแบบซีเควนเชียลในรูปแบบที่ถูกต้องด้วย สามารถแบ่งได้เป็น 3 รูปแบบคือ แบบหนึ่งเฟส , แบบ 2 เฟสและแบบครึ่งสเต็ป

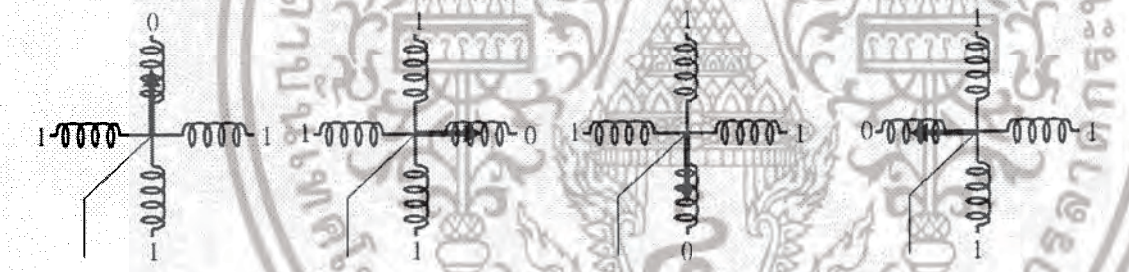
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1 แบบหนึ่งเฟสหรือแบบฟูลสเต็ป (full step)

เป็นการกระตุ้นที่มีรูปแบบง่ายที่สุด โดยทำการ กระตุ้นขดลวดทีละขดในเวลาหนึ่งไล่เรียงถัดกันไป เช่น เริ่มต้นที่ขดที่ 1,2,3,4 แล้ววนกลับมาขดที่ 1 วนไปเรื่อยๆ หรือเริ่มที่ขดที่ 1 แล้วย้อนไปยังขดที่ 4,3,2 แล้วกลับมายังขดที่ 1 อีกครั้ง ซึ่งทำให้ทิศทาง ของการหมุนสวนกันในการกระตุ้นรูปแบบนี้จึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น วงจรกระตุ้นแบบเวฟจึงมีราคาถูกและง่าย ขั้นตอนการทำงานต่างๆ แสดงดังในตาราง 2.2

ตารางที่ 2.2 ลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์ เมื่อได้รับการกระตุ้นแบบหนึ่งเฟส

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน



รูปที่ 2.24 การหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบหนึ่งเฟส

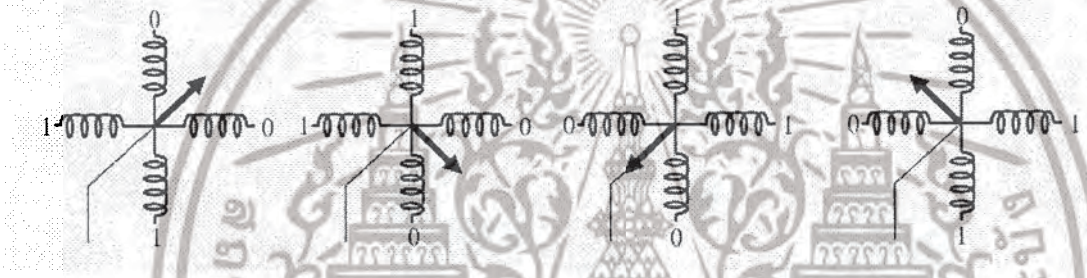
2.2.1.2 แบบ 2 เฟส

เป็นการกระตุ้นซึ่งคล้ายกับแบบหนึ่งเฟส แต่การกระตุ้นแบบนี้จะทำการกระตุ้น โดยจ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขด ที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไปเช่นเดียวกับแบบ เวฟ ดังตัวอย่าง ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 และ 4 ถัดไปเป็นขดที่ 4 และ 1 แล้วกลับมาที่ขดที่ 1 และ 2 วนไปตามลำดับเช่นนหรือ เริ่มที่ขด 1 และ 4 ตามด้วยขดที่ 4 และ 3 ถัดไปเป็นขดที่ 3 และ 2 ต่อไปเป็นขดที่ 2 และ 1 แล้ววน กลับมาที่ขดที่ 1 และ 4 ทิศทางการหมุนจะสวนทางกัน การกระตุ้นสเต็ปปิ้งมอเตอร์แบบนี้สามารถเพิ่ม แรงบิดได้มากกว่าแบบเวฟ โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อม กัน และต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือการกระตุ้นแบบนี้ต้องใช้กำลังไฟฟ้า มากขึ้น ขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์เมื่อได้รับการกระตุ้นแบบสองเฟส

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน



รูปที่ 2.25 การหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบสองเฟส

แบบ 2 เฟสเป็นการกระตุ้นซึ่งคล้ายกับแบบหนึ่งเฟส แต่การกระตุ้นแบบนี้จะทำการกระตุ้น โดยจ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขด ที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไปเช่นเดียวกับแบบ เวฟ ดังตัวอย่าง ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 และ 4 ถัดไปเป็นขดที่ 4 และ 1 แล้วกลับมาที่ขดที่ 1 และ 2 วนไปตามลำดับเช่นนหรือ เริ่มที่ขด 1 และ 4 ตามด้วยขดที่ 4 และ 3 ถัดไปเป็นขดที่ 3 และ 2 ต่อไปเป็นขดที่ 2 และ 1 แล้ววน กลับมาที่ขดที่ 1 และ 4 ทิศทางการหมุนจะสวนทางกัน การกระตุ้นสเต็ปปิ้งมอเตอร์แบบนี้สามารถเพิ่ม แรงบิดได้มากกว่าแบบเวฟ โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อม กัน และต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือการกระตุ้นแบบนี้ต้องใช้กำลังไฟฟ้า มากขึ้น ขั้นตอนการทำางานต่างๆ แสดงดังในตารางที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์เมื่อได้รับการกระตุ้นแบบครึ่งสเต็ป

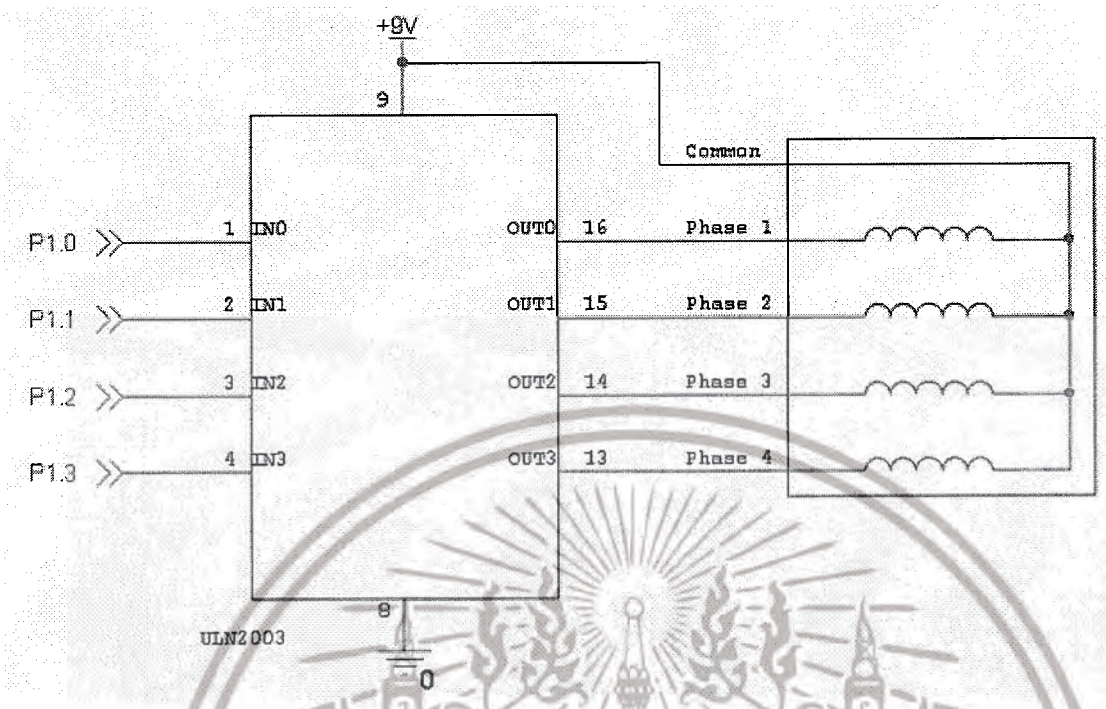
สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

รูปที่ 2.26 การหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบครึ่งสเต็ป

2.2.2 วงจรขับ Stepping Motor

จากที่กล่าวมาข้างต้นในการควบคุม Stepping Motor จะต้องมีการป้อนสัญญาณที่เป็นซีเควน เชียล ออกไป โดยที่หลังจากเราป้อนสัญญาณเราจะต้องคอยเวลา(Delay) จนกระทั่ง Stepping Motor หมุนไปเสร็จสิ้นแล้วจึงจะสามารถส่งสัญญาณต่อไปได้ ซึ่งจะต้องมีภาคขับ (Driver Circuit) ที่ สามารถจายกระแสได้เพียงพอทจะทาให้ Stepping Motor หมุนไปได้ ซึ่งในวงจรภาคขับนี้เราอาจใช้ ไอซีแบบ Open Collector เช่น ULN2003 , ULN2803 เป็นต้น หรืออาจใช้ Power Transistor หรือ Power MOSFET เช่น BD139 , IRF510 เป็นต้น

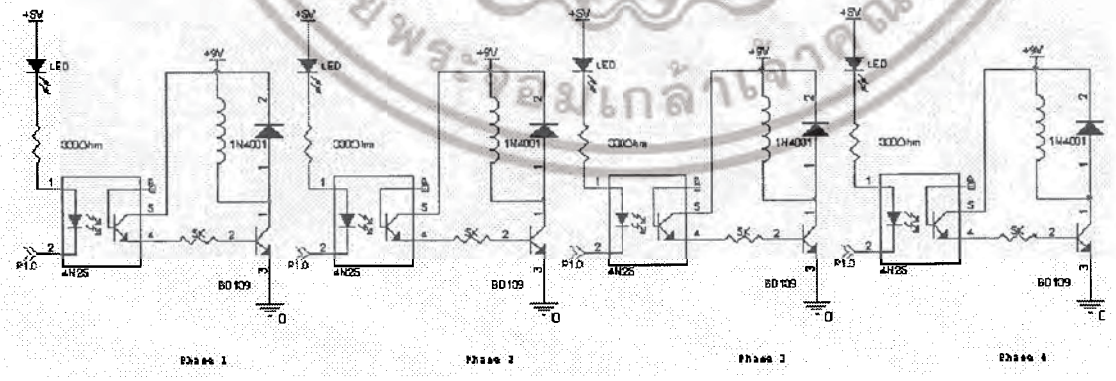
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 วงจรขับ Stepping Motor โดยใช้ไอซี Driver เบอร์ ULN2003

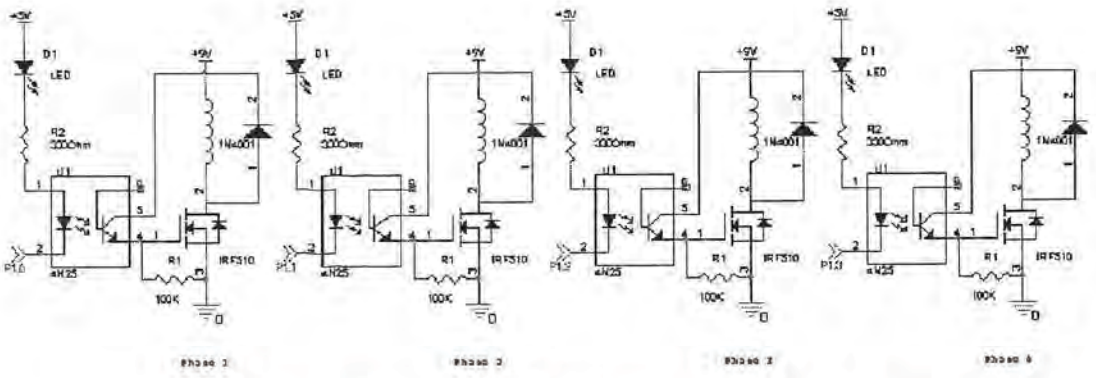
2.2.3 ไอซีเบอร์ ULN2003

วงจรที่ใช้ในการขับสเต็ปมิ้งมอเตอร์โดยใช้ไอซีสำเร็จรูปเบอร์ ULN2003 จะมีคุณสมบัติเป็นไอซีไดรเวอร์กระแสสูงแบบคอลเล็คเตอร์เปิด สามารถเลือกแรงดันได้กว้าง 5-30 โวลท์ จ่ายกระแสได้สูงถึง 500 mA ต่อขา และมีไดโอดที่ป้องกันกระแสย้อนกลับอยู่ภายในไอซี ส่วนแอลอีดีที่ต่อในวงจรเราจะต่อไว้เพื่อแสดงการกระตุ้นแต่ละเฟสของแต่ละแบบ



รูปที่ 2.28 วงจรขับ Stepping Motor โดยใช้ Power Transistor เบอร์ BD139

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 วงจรขับ Stepping Motor โดยใช้ Power MOSFET เบอร์ IRF510

2.3 การใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

สำหรับไมโครคอนโทรลเลอร์แล้วการมีพอร์ตที่เป็น Digital I/O อย่างเดียวนั้นยังไม่พอหรือยังไม่ตรงกับความต้องการของผู้ออกแบบระบบ Embedded system สักเท่าไหร่นัก ในไมโครคอนโทรลเลอร์ใหม่ๆนั้น มักจะมีพอร์ตอนุกรมเพิ่มเข้ามาอีกเพื่อทำให้การใช้งานในการติดต่อสื่อสารกับอุปกรณ์ภายนอกทำได้ง่ายขึ้น แต่การใช้งานก็มีข้อควรระวังอยู่บ้างเช่นกัน

2.3.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมก็คือการที่อุปกรณ์หรือ devices สองตัวทำการติดต่อกันโดยสัญญาณออกไปหรือรับเข้ามามีลักษณะของกระบวนสัญญาณที่เปลี่ยนแปลงสลับไปมาอย่างต่อเนื่อง ณ เวลาใดเวลาหนึ่งโดยมีรูปแบบการส่งที่แน่นอน เช่นการส่งรหัสมอร์ส การสื่อสารผ่าน RS232 หลายคนอาจคิดว่าเมื่อพูดถึงพอร์ตอนุกรมของไมโครคอนโทรลเลอร์มักจะเหมารวมว่ามันคือพอร์ต RS232 แต่จริงๆแล้ว RS232 คือข้อกำหนดของสัญญาณที่มีค่าบวกลบ 3Vถึงบวกลบ 15V เพราะฉะนั้นถ้าต้องการให้ไมโครคอนโทรลเลอร์สามารถสื่อสารแบบ RS232 ได้จะต้องมีตัวไอซีหรือวงจรที่จะทำให้สัญญาณอนุกรมที่เป็น TTL เปลี่ยนเป็นสัญญาณมาตรฐาน RS232 ก่อน เช่น MAX232 เป็นต้น แต่พอร์ตอนุกรมสามารถสื่อสารกับไมโครคอนโทรลเลอร์ได้โดยไม่ต้องมีไอซีหรือวงจรเข้ามาช่วยหรือกับอุปกรณ์อื่นๆ ที่มีการสื่อสารแบบเดียวกัน

2.3.2 ประโยชน์ของพอร์ตอนุกรม

พอร์ตอนุกรมนั้นนอกจากจะใช้สื่อสารกับอุปกรณ์ภายนอกด้วยกันแล้วยังใช้เป็นอุปกรณ์สำหรับการดีบักโปรแกรม เช่น ใช้ดูค่าตัวแปรของโปรแกรม ดูสถานะของพอร์ต ฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 การเพิ่มระยะทางของพอร์ตอนุกรม

ปกติแล้วสัญญาณอนุกรมที่ออกจากตัวไมโครคอนโทรลเลอร์จะเป็นสัญญาณ TTL ที่มีขนาด 0-5V ถ้าการสื่อสารไม่ไกลกันเช่นอยู่บนบอร์ดเดียวกันก็ไม่ต้องใช้วงจรใดๆช่วย แต่ถ้าตัวบอร์ดหรือวงจรอยู่ไกลกันก็ใช้ไอซีช่วย เช่น DS232 ซึ่งก็จะกลายเป็นการสื่อสารแบบ RS232 ทันทีแต่ RS232 ก็มีข้อจำกัดด้านระยะทางเหมือนกันคือประมาณ 15-20 เมตรที่บอร์ดเร็ต 9600 และบอร์ดเร็ตแปรผกผันกับระยะทางด้วยและยังขึ้นกับไอซีไดเวอร์และชนิดของสายด้วย แต่ถ้าใช้สายยาวๆแล้วใช้บอร์ดเร็ตสูงๆ ไม่แนะนำให้ใช้งานจริงถึงแม้จะทดลองว่าใช้ได้เพราะสถานที่ใช้งานจริงกับสถานที่ทดลองอาจมีสภาพแวดล้อมต่างกัน

2.3.4 การใช้งานไมโครคอนโทรลเลอร์ที่มีฮาร์ดแวร์ของพอร์ตอนุกรมในตัว

ในการใช้งานชุดโมดุลของ Hardware serial port นั้นตัวไมโครคอนโทรลเลอร์จะต้องมีส่วนที่เรียกว่า UART หรือ USART ซึ่งก็แล้วแต่จะเรียกกันในแต่ละยี่ห้อของเบอร์หรือตระกูลนั้นๆ การทำงานจะคล้ายกันแต่ความสามารถจะต่างกัน

2.3.5 การเขียนโปรแกรมติดต่อและควบคุม Serail Port

2.3.5.1 พื้นฐานการสื่อสารแบบอนุกรม

ถึงแม้ว่าการสื่อสารแบบอนุกรมในคอมพิวเตอร์นั้นจะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน ทั้งนี้เพราะว่าการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลทีละ 1 บิต แต่พอร์ตขนานนั้นสามารถส่งข้อมูลได้ทีละหลาย ๆ บิตพร้อมกันส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน

แต่ว่าการส่งข้อมูลแบบอนุกรมนั้นมีข้อที่เหนือกว่าการส่งข้อมูลแบบขนานคือการที่สามารถส่งข้อมูลได้ในระยะทางที่ไกลกว่าแบบขนาน อีกทั้งสายสัญญาณที่ใช้ยังมีน้อยกว่าการส่งข้อมูลแบบขนานอีกด้วย การสื่อสารแบบอนุกรมสามารถแบ่งออกได้เป็น 3 รูปแบบดังนี้

1. Simple สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว
2. Half-Duplex สามารถส่งข้อมูลไปยังปลายทางและสามารถรับข้อมูลจากปลายทางได้ แต่ไม่สามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกัน
3. Full-Duplex สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

นอกจากนี้แล้วยังสามารถแบ่งประเภทของการสื่อสารแบบอนุกรมตามลักษณะสัญญาณในการส่งได้อีก 2 แบบ คือ

1.การสื่อสารแบบซิงโครนัส (Synchronous) สำหรับการสื่อสารแบบซิงโครนัสนี้จะใช้สัญญาณนาฬิกาควบคุมการส่งสัญญาณ เช่น สายคีย์บอร์ดคอมพิวเตอร์ โดยมีสายสัญญาณเส้นหนึ่งเป็นสายนาฬิกา ส่วนอีกเส้นหนึ่งเป็นสายของข้อมูล (และจะมีสายกราวน์ด้วย)

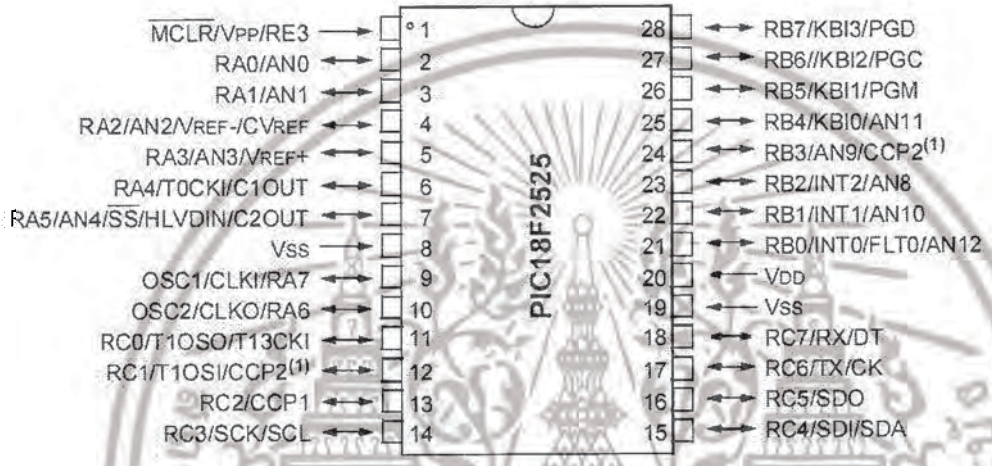
2.การสื่อสารแบบอะซิงโครนัส (Asynchronous) สำหรับการสื่อสารแบบอะซิงโครนัสนั้นจะใช้สายข้อมูลเพียงตัวเดียว แต่จะใช้รูปแบบการส่งข้อมูลหรือ Bit Pattern เป็นตัวกำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล ส่วนไหนเป็นตัวข้อมูล ส่วนไหนจะเป็นส่วนตรวจสอบความถูกต้องของข้อมูล และส่วนไหนเป็นส่วนปิดท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาคส่ง และภาครับ ซึ่งจะมีอุปกรณ์พิเศษที่ชื่อว่า UART หรือ Universal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UART ที่นิยมใช้งานในคอมพิวเตอร์ คือ ชิป 8250 ซึ่งให้ความเร็วสูงสุด 57.6 Kbps หรือ 56K ขณะที่ปัจจุบันเริ่มใช้งานกับชิป 16450 ซึ่งสนับสนุน 8250

2.3.5.2 ไมโครคอนโทรเลอร์PIC18F2525

พื้นฐานการทำงานของไมโครคอนโทรลเลอร์ก็ คือ ระบบดิจิตอลโดยค่าเอาท์พุตที่ได้จากไมโครคอนโทรลเลอร์จะเป็น 0 กับ 1 แต่ก็สามารถนำมาประยุกต์เชื่อมต่อกับอุปกรณ์ภายนอกต่างๆ มากมาย



รูปที่ 2.30 ไมโครคอนโทรเลอร์PIC18F2525

จากรูปขาของ PIC18F2525 แต่ละขาจะมีหน้าที่แตกต่างกันไปซึ่งแยกออกเป็น PORT A , PORT B , PORT C , โดยพื้นฐานแล้วพอร์ตแต่ละพอร์ตสามารถทำงานเป็นอินพุตและเอาท์พุตเป็นดิจิตอล ยกเว้น PORT A ที่สามารถทำงานเป็นตัวรับสัญญาณอนาลอกแปลงเป็นค่าดิจิตอลเพื่อนำมาวัดปริมาณทางฟิสิกส์ต่างๆ ที่เห็นได้อย่างชัดเจน คือ นำมาวัดความต่างศักย์

2.3.5.3 คุณสมบัติของ 18F2525

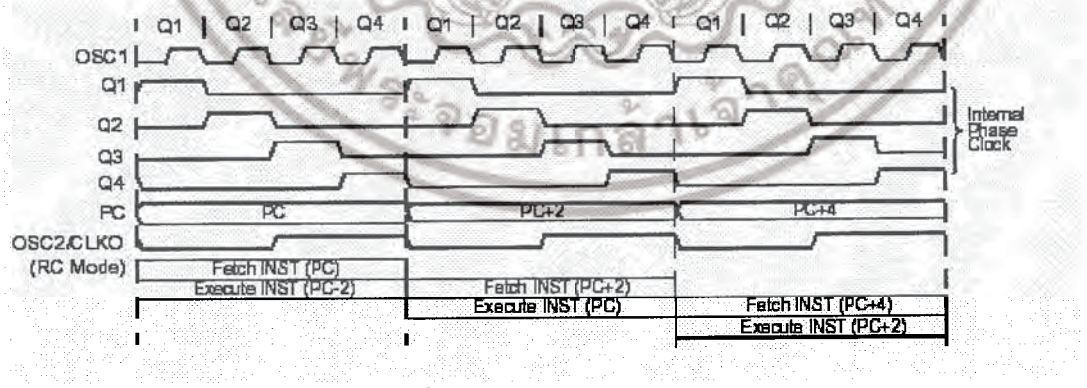
1. มีคำสั่งให้ใช้งาน 35 คำสั่ง
2. คำสั่งหนึ่งๆใช้เวลาทำงาน 1 ถึง 2 Cycle
3. ทำงานได้สูงสุดที่ 20MHz
4. ทำงานแบบ Pipe-line (มี 2 ท่อ) ทำให้ ณ เวลาหนึ่งทำงาน 2 อย่างพร้อมๆกันได้
5. หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8KWord (1 word=14 บิต)
6. มี RAM ขนาด 368 ไบต์ ให้เราใช้งาน
7. มี EEPROM ขนาด 256 ไบต์
8. ตอบสนองกับอินเตอร์รัพได้ทั้งหมด 14 แหล่ง
9. มี Stack ให้ใช้ได้สูงสุด 8 ระดับ
10. มีระบบ Power On Reset, Power Up Timer, Oscillator Start-up timer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. Watchdog timer
12. มีระบบ Code Protection
13. ประหยัดพลังงาน
14. สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งาน คือ อาจจะใช้ XTAL หรือ วงจร RC ก็ได้
15. สามารถโปรแกรมด้วยไฟ +5VDC ได้
16. ใช้การโปรแกรมแบบ In-Circuit Serial Programming
17. ทำงานที่ไฟเลี้ยง 2VDC ถึง 5.5VDC
18. Current Sink และ Current Source อยู่ที่ 25mA
19. มี Timer/Counter 3 ตัว
20. มีโมดูล Capture/Compare/PWM อีก 2 ชุด (มี PWM นี่คุม DC Motor ได้)
21. มี A-TO-D Converter แบบ 10 บิต จำนวน 8 ช่องนำเข้าในตัวเอง (ทำให้ไม่ต้องใช้ไอซี A-TO-D Converter)
22. มีระบบ USART สำหรับต่อกับ การสื่อสารแบบ RS232 หรือดีกว่า
23. มีระบบตรวจระดับไปเลี้ยง (Brown-out reset)
24. มี I/O พอร์ตทั้งหมด 3 พอร์ต (33 bit)

2.3.5.1 สัญญาณนาฬิกา

ไมโครคอนโทรลเลอร์จะทำงานได้ต้องมีสัญญาณนาฬิกาให้กับตัวมัน ซึ่งในหนึ่งไซเคิล (Clock Bus) ของซีพียูจะประกอบไปด้วยสัญญาณนาฬิกาภายนอกจำนวน 4 ไซเคิล คือ Q1 , Q2 , Q3 และ Q4 ตามรูปข้างล่าง ดังนั้นความความถี่ที่ซีพียูประมวลผลต่อหนึ่งคำสั่ง เท่ากับความถี่ของสัญญาณนาฬิกาภายนอกหารด้วย 4 หรือหากจะพิจารณาความเร็วของไมโครคอนโทรลเลอร์ตระกูล PIC สามารถประมวล ผลต่อหนึ่งคำสั่งเท่ากับ 1/4 เท่าของความถี่ออสซิเลเตอร์ภายนอก



รูปที่ 2.31 รูปของสัญญาณนาฬิกา

วงจรพื้นฐานที่ต้องต่อทุกครั้งไม่งั้นไมโครคอนโทรลเลอร์ของท่านจะไม่ทำงาน ส่วนขาที่เหลือจะนำมาเชื่อมต่อกับ อุปกรณ์ภายนอกเพื่อควบคุมตามจุดประสงค์ตามที่ผู้เขียนโปรแกรมต้องการ ซึ่งในที่นี้อุปกรณ์ภายนอกคือ stepping motor และ จอแสดงผล LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาจะเป็นการนำเสนอ ออสซิเลเตอร์แบบต่างๆ และการกำหนดในโปรแกรมเพื่อให้ไมโครคอนโทรลเลอร์ ของเราสามารถทำงานได้

- LP (Low Power Crystal) คริสตอลพลังงานต่ำ
- XT (Crystal/Resonator) คริสตอล หรือ เรโซเนเตอร์
- HS (High Speed Crystal/Resonator) คริสตอง หรือ เรโซเนเตอร์ความเร็วสูง
- RC (External Resistor/Capacitor) วงจร RC ภายนอก
- EC (External Clock) สัญญาณนาฬิกาจากภายนอก
- H4 (HS + PLL : High Speed Crystal/Resonator with PLL enabled)คูณ 4 PLL คือจะทำการคูณสัญญาณนาฬิกาที่เข้ามา ด้วย 4 เช่น OSC ความถี่ 10 MHz เมื่อผ่านกระบวน การนี้จะทำให้ได้ความถี่เท่ากับ 40 MHz(คุณสมบัตินี้มีเพียงบางเบอร์เท่านั้น เช่น 18F458)
- EC Clockout ใช้คริสตอลภายนอก และกำหนดให้ขา OSC2 เป็นเอาต์พุต
- INTRC (IN) วงจร RC ภายในไมโครคอนโทรลเลอร์
- EXTRC (ER) วงจร RC ภายในไมโครคอนโทรลเลอร์และกำหนดให้ขา OSC2 เป็นเอาต์พุต
- EXTRC Clockout วงจรตัวต้านทานภายนอกโดยกำหนดค่าความถี่จากค่าความต้านทานที่นำมาต่อ และ กำหนดให้ขา OSC2 เป็นเอาต์พุต

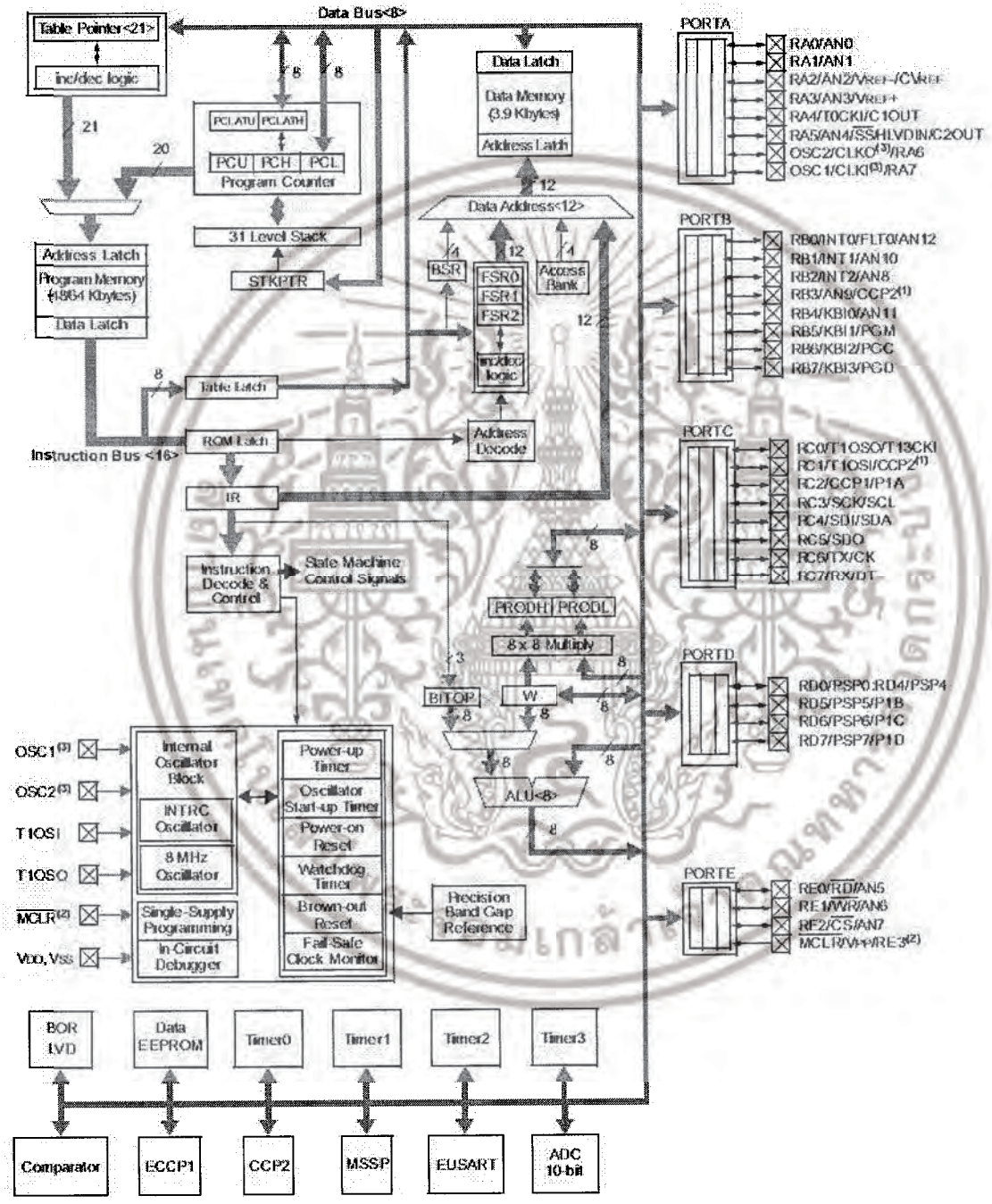
Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	C2
LP	32 kHz	30 pF	30 pF
XT	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	10 MHz	15 pF	15 pF
	20 MHz	15 pF	15 pF
	25 MHz	15 pF	15 pF

รูปที่ 2.32 ออสซิเลเตอร์แบบต่าง ๆ และการกำหนดในโปรแกรมให้ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 โครงสร้างภายในชิพ

ผังไดอะแกรมของ PIC 18F2525



รูปที่ 2.33 ผังไดอะแกรมของ PIC 18F2525

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 กลุ่มออบเจ็กต์หลักของสถาปัตยกรรม ADO.NET

กลุ่มออบเจ็กต์ที่อาศัย OLEDB Data Provider ทำหน้าที่เข้าถึงข้อมูลในฐานข้อมูล มี 3 ตัวคือ ออบเจ็กต์ OleDbConnection, OleDbDataAdapter และออบเจกต์ OleDbCommand ใช้งานร่วมกับฐานข้อมูลชนิด Access

กลุ่มออบเจ็กต์ที่อาศัย SQL Server Data Provider ทำหน้าที่เข้าถึงข้อมูลในฐาน มี 3 ตัวคือ ออบเจ็กต์ SqlConnection SqlDataAdapter และออบเจ็กต์ SqlCommand ใช้งานร่วมกับฐานข้อมูลชนิด SQL Server

2.4.1 ภาษา SQL

SQL ย่อมาจาก Structured Query Language หมายถึงภาษากลางที่ทำหน้าที่สำหรับจัดการข้อมูลในฐานข้อมูล โดยเฉพาะอย่างยิ่งฐานข้อมูลประเภท RDBMS (Relational Database Management System) จะรู้จัก SQL นี้เป็นอย่างดี

เราจะใช้ภาษา SQL ทำหน้าที่แสดงข้อมูล,เพิ่ม,แก้ไข หรือลบข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลประเภทต่างๆ ซึ่งจะเรียกว่าการทำคิวรี(Query)

เราจะใช้ภาษา SQL เพื่อทำคิวรีข้อมูลที่เก็บอยู่ในตารางต่างๆ โดยมีจุดประสงค์ต่างกัน เช่น การเรียกดูข้อมูลทั้งหมด การเรียกดูข้อมูลแบบมีเงื่อนไข อาจจะมาจากตารางเดียวกันหรือหลายตารางก็ได้ แล้วนำมาแสดงร่วมกันในเวลาเดียวกัน

ดังนั้นการทำคิวรีจึงเปรียบเสมือนกับเป็นการสร้างตารางเสมือนขึ้นมา เพื่อนำข้อมูลที่อยู่ในตารางต่างๆนำมาแสดงร่วมกัน เป็นตารางที่ไม่ได้มีอยู่ในฐานข้อมูลจริงๆ เป็นมุมมองของคุณที่ต้องการนำข้อมูลดังกล่าวมาใช้งานโดยที่ผู้เขียนใช้ฐานข้อมูล Northwind.mdb เป็นตัวอย่างประกอบคำอธิบายการใช้งานภาษา SQL

2.4.2 โครงสร้างพื้นฐานของภาษา SQL

เราสามารถแยกภาษา SQL ออกได้ 3 กลุ่มใหญ่ คือ

1.กลุ่มของ Data Definition Language เรียกสั้นๆ ว่า DDL เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดโครงสร้างของฐานข้อมูล เช่น การสร้างตารางในฐานข้อมูล ปรับปรุงโครงสร้างของฐานข้อมูล เพื่อเพิ่มหรือลดตารางในฐานข้อมูล เป็นต้น การสร้างฐานข้อมูลใน MS SQL Server ก็จะมีการใช้งานคำสั่งในกลุ่ม DDL เป็นหลัก

2.กลุ่มของ Data Manipulation Language เรียกสั้นๆ ว่า DML เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดโครงสร้างของฐานข้อมูล เช่น การแสดงข้อมูลแบบมีเงื่อนไข การเรียกดูข้อมูลจากหลายๆตาราง การเพิ่มข้อมูล การลบข้อมูล เป็นต้น เป็นกลุ่มคำสั่งที่ใช้งานมากที่สุด

3.กลุ่มฟังก์ชันภายใน Aggregate Function เป็นกลุ่มฟังก์ชันพิเศษของภาษา SQL ที่ทำหน้าที่เฉพาะอย่าง เช่น การหาผลรวมของเร็คคอร์ด การหาค่าสูงสุด ค่าต่ำสุด การกำหนดจำนวนเร็คคอร์ดที่ต้องการแสดง เป็นต้น เป็นกลุ่มของฟังก์ชันที่มีประโยชน์อย่างยิ่ง เพราะว่าจะช่วยลดภาระให้คุณไม่ต้องเขียนโค้ดจัดการเอง

สำหรับการใช้งานภาษา SQL ผู้เขียนจะอธิบายเฉพาะกลุ่มคำสั่ง DML และกลุ่มฟังก์ชัน Aggregate เท่านั้น ในส่วนของกลุ่มคำสั่ง DDL มักจะนำไปใช้ร่วมกับฐานข้อมูลประเภท Database Server เช่น SQL Server, Oracle เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 พื้นฐานการใช้งานกลุ่มคำสั่ง DML

คำสั่งในกลุ่มของ DML ประกอบไปด้วยคำสั่งพื้นฐานอยู่ 4 ด้วยกัน คือ

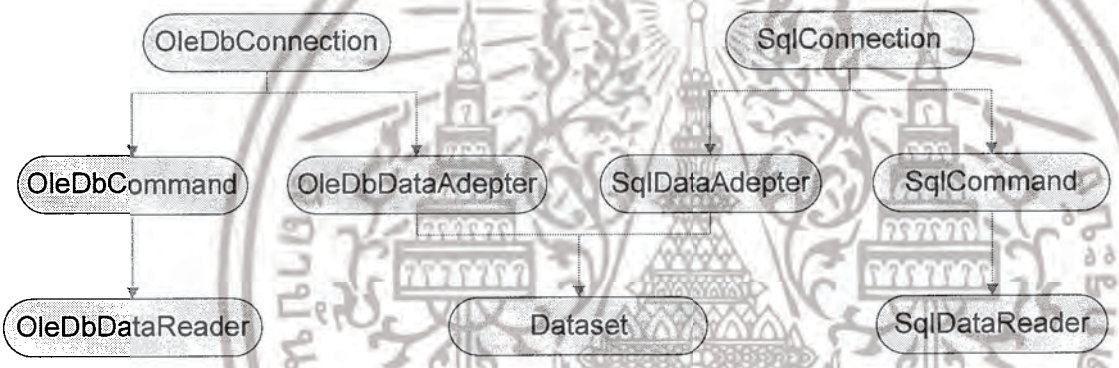
คำสั่ง SELECT เป็นคำสั่งที่ใช้สำหรับเรียกดูข้อมูล หรือแสดงเร็คคอร์ดจากตารางต่างๆ อาจจะมาจากตารางเดียวหรือหลายตารางก็ได้ แยกออกได้ 2 กรณีคือ

- การเรียกดูข้อมูลแบบไม่มีเงื่อนไข เป็นการแสดงข้อมูลทั้งหมดที่อยู่ในแต่ละตาราง
- การเรียกดูข้อมูลแบบมีเงื่อนไข เป็นการจำกัดข้อมูลที่จะแสดงออกมา เพื่อให้ได้เฉพาะข้อมูลที่อยู่ในเงื่อนไขที่คุณต้องการ โดยการใช้ภาษาร่วมกับคำสั่ง WHERE

คำสั่ง DELETE เป็นคำสั่งที่ใช้สำหรับลบข้อมูลหรือลบเร็คคอร์ดใดๆ ออกจากตาราง

คำสั่ง INSERT เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูลหรือเพิ่มเร็คคอร์ดเข้าไปในตาราง

คำสั่ง UPDATE เป็นคำสั่งที่ใช้สำหรับแก้ไขข้อมูลในเร็คคอร์ดใดๆ



รูปที่ 2.34 โครงสร้างกลุ่มออบเจคสถาปัตยกรรม ADO.NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและหลักการสร้างวงจร

การออกแบบและการสร้างระบบควบคุมการใช้บริการรถไฟฟ้าโดยใช้บัตร RFID มีส่วนประกอบดังนี้

1.ชุดอ่านบัตร RFID

2.ชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

3.ระบบฐานข้อมูลของระบบการใช้บริการรถไฟฟ้า

4.โปรแกรมการจัดการระบบบัตรรถไฟฟ้า

3.1 หลักการใช้งานชุดอ่านบัตร RFID

การเชื่อมต่อระหว่างชุดอ่านบัตร RFID กับคอมพิวเตอร์

การเชื่อมต่อระหว่างชุดอ่านบัตร RFID กับคอมพิวเตอร์ เชื่อมต่อกันด้วยมาตรฐาน RS-232 โดยที่ชุดอ่านบัตรจะมีการประมวลผลการทำงานโดยจะอ่านค่ารหัสประจำตัวบัตรที่ได้เพื่อส่งให้คอมพิวเตอร์ด้วยมาตราฐานการเชื่อมต่อแบบ RS-232 เพื่อนำไปใช้กับโปรแกรมจัดการระบบบัตรโดยรถไฟฟ้า เช่นการเพิ่มหรือลบข้อมูลผู้ใช้บริการ, การเข้าไปแก้ไขข้อมูล, การเก็บบันทึกข้อมูลของผู้ใช้บริการ หรือการเติมเงินของผู้ใช้บริการ

การเก็บข้อมูลของผู้ใช้บริการนั้น ฐานข้อมูลที่ใช้เขียนด้วยโปรแกรม SQL Server ซึ่งเป็นโปรแกรมฐานข้อมูล โดยมีการสร้างโปรแกรมที่เขียนด้วย C# เพื่อจัดการเกี่ยวกับข้อมูลของผู้ใช้บริการแล้วส่งข้อมูลไปเก็บไว้ที่ฐานข้อมูล โดยผ่าน ADO.NET

3.1.1 ฟังก์ชันการทำงานของ RFID

คำสั่งที่ใช้จะมาจาก Read Write Device (RWD) และถูกควบคุมโดย Digital Control Unit ของ MF1 IC S50 เพื่อให้มีเงื่อนไขที่สอดคล้องกัน สำหรับการติดต่อ Sector ซึ่งมีส่วนต่าง ๆ ที่ประกอบในบล็อคไดอะแกรมดังต่อไปนี้

3.1.1.1 Request Standard / All

หลังจาก Power On Reset (POR) ของการ์ดสามารถตอบสนองในคำสั่งรีเควส ก็จะมีการส่งจากสัญญาณ Read / Write เข้าสู่การ์ดทั้งหมดที่อยู่ในระยะแพร่กระจายคลื่นของสายอากาศ แล้วจัดการส่งคำตอบรับ

3.1.1.2 Anticollision Loop

สำหรับลูปนี้ จะเป็นการเลือก Serial Number โดยที่เมื่อ Serial Number ถูกอ่าน ถ้ามีการอ่านหรือเขียนค่ามากกว่าหนึ่งค่า จะทำการจำแนกค่าต่าง ๆ ออกมา แล้วทำการเลือกค่าสำหรับการติดต่อในการดำเนินการต่อไป หากไม่มีการเลือกค่าจะกลับเข้าสู่ Standby Mode และรอสำหรับการร้องขอครั้งใหม่

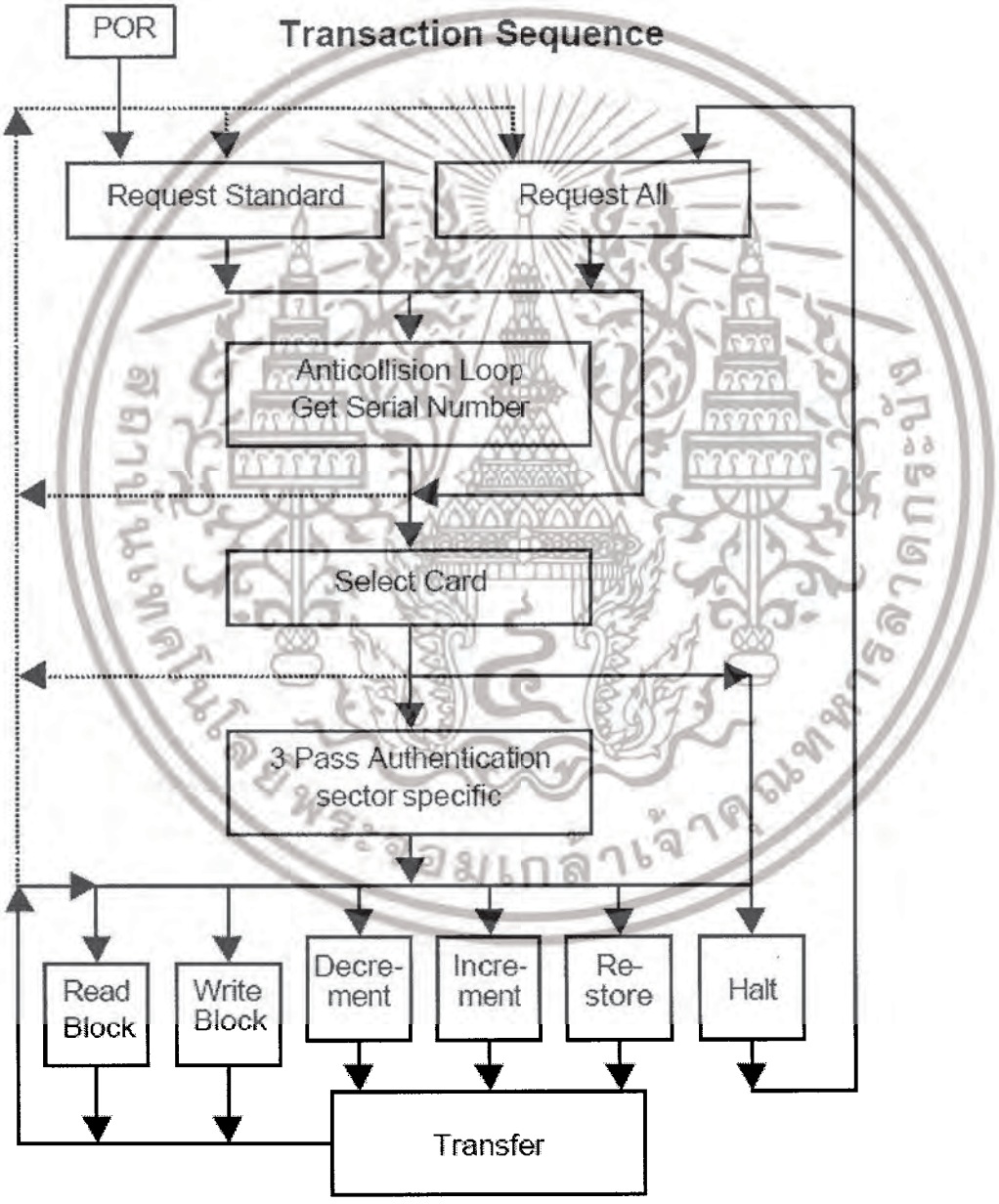
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.3 Select Card

ในการเลือกการ์ด คำสั่ง RWD จะทำการเลือกเพียงการ์ดเดียวสำหรับการยืนยันตัวตนและบันทึกในหน่วยความจำ ซึ่งการ์ดจะตอบกลับด้วย Select code ซึ่งในการกำหนดชนิดของ Select Card

3.1.1.4 Pass Authentication

หลังจากขั้นตอนของการ Select Card แล้ว RDW จะกำหนดที่จัดเก็บในหน่วยความจำในการเข้าถึงข้อมูล โดยมีกระบวนการ 3 ขั้นตอน ในการรับรอง ซึ่งหลังจากผ่านกระบวนการนี้แล้ว หน่วยความจำก็จะสร้างรหัสได้



รูปที่ 3.1 Block Diagram แสดงขั้นตอนการทำงานของ Reader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป Block Diagram ข้างต้น เมื่อ Power On Reset (POR) และมีสัญญาณจาก Tag เข้ามา จะเข้าสู่สถานะ Request Standard เพื่อเตรียมการที่จะส่งสัญญาณ Request หลังจากนั้นจะเข้าสู่กระบวนการ Anticollision Loop ซึ่งสำหรับลูปนี้ จะเป็นการเลือก Serial Number ในกรณีที่ Tag ถูกอ่าน ถ้ามีการอ่านหรือเขียนค่า มากกว่าหนึ่งค่า ส่วนนี้จะทำการจำแนกค่าต่าง ๆ ออกมา ค่าที่ไม่ได้ถูกเลือกก็จะกลับเข้าสู่ Stand by mode นั่นคือเตรียมพร้อมที่จะเข้าสู่สภาวะ Request อีกครั้ง สำหรับการติดต่อในการดำเนินการต่อไป ส่วนค่าจาก Tag ที่ถูกเลือก เมื่อมีการเลือกแล้วจะมีการผ่านกระบวนการ ในส่วนของ Pass Authentication sector specific ซึ่งเป็นกระบวนการในการตรวจสอบและยืนยันตัวตน

ทันทีที่เข้าสู่กระบวนการนี้ RWD จะเจาะจงsector ในการเข้าถึงข้อมูล และเลือก Key A หรือ Key B จะเริ่มกระบวนการหลัก 3 ขั้นตอนดังนี้

1.) การ์ดจะอ่านคีย์ลับและทำการเข้าถึงสถานะจากส่วนหางของ Sector จากนั่นการ์ดจะทำการสุ่มหมายเลขให้กับ RWD

2.) RWD จะคำนวณ Response โดยใช้คีย์ลับ และ additional input เป็นสัญญาณตอบสนองร่วมกันโดยทำการสุ่มจาก RWDหลังจากการส่งผ่านไปยังการ์ด

3.) การ์ดจะทำการพิสูจน์สัญญาณ Response ของ RWD จากการ Comparing และจากนั้นจะทำการคำนวณสัญญาณ Response เพื่อที่จะส่งต่อ

จากนั้น RWD ทำการพิสูจน์สัญญาณ Response ของการ์ดจากการ Comparing ซึ่งการส่งผ่านข้อมูลของการสุ่มครั้งแรกในการติดต่อสื่อสารระหว่างการ์ดและ RWD จะถูกสร้างรหัสลับขึ้นมาดังกล่าวที่ถูกเลือกนี้ก็จะผ่านเข้าสู่กระบวนการในการทำธุรกรรมต่าง ๆ ได้ แต่ในกรณีที่มีการทำงานนอกเหนือจากฟังก์ชันที่กำหนด เช่นมีการ Read / Write นอกเหนือ Sector ที่กำหนดไว้ กระบวนการดังกล่าวนี้จะเข้าสู่ Block Halt แล้วเข้าสู่กระบวนการ Request ใหม่อีกครั้ง

3.1.1.5 Memory Operations

หลังจากการ Authentication แล้วอาจมีกระบวนการต่าง ๆ ที่ตามมาดังนี้

- Read Block
- Write Block
- Decrement : การเพิ่มค่าของแต่ละบล็อคและเก็บค่าผลลัพท์ไว้ชั่วคราวภายในส่วนของ Data - register
- Increment : การลดค่าของแต่ละบล็อคและทำการเก็บค่าผลลัพท์ไว้ชั่วคราวภายในส่วนของ Data – register
- Restore : ย้ายค่าของบล็อคเข้าสู่ Data – register
- Transfer : เขียนค่าที่เก็บไว้ชั่วคราวของส่วนภายใน Data – register เข้าไปในบล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 Data Integrity

ตามโครงสร้างที่แสดงจะมีวิธีติดต่อกันระหว่าง RWD และการ์ด ในการยืนยันความถูกต้องของในการส่งผ่านข้อมูล

- ประกอบด้วย 16 bits CRC ต่อ หนึ่งบล็อค
- ประกอบด้วย Parity bit ในแต่ละ Byte
- Bit count Checking
- มี Bit coding ในการจำแนกระหว่าง "1" , "0" และไม่มีข้อมูล
- Channel monitoring

3.1.3 Security

ในการเตรียมการนี้มีระบบ Security ที่สูงมากซึ่งมีถึง 3 ขั้นตอนในกระบวนการ Authentication

ก.) RWD จะเจาะจง sector ในการเข้าถึงข้อมูล และเลือก Key A หรือ Key B

ข.) การ์ดจะอ่านคีย์ลับและทำการเข้าถึงสถานะจากส่วนหางของ Sector จากนั่นการ์ดจะทำการสุ่มหมายเลขให้กับ RWD (ขั้นตอนที่ 1)

ค.) RWD จะคำนวณ Response โดยใช้คีย์ลับ และ additional input เป็นสัญญาณตอบสนองร่วมกันโดยทำการสุ่มจาก RWDหลังจากการส่งผ่านไปยังการ์ด (ขั้นตอนที่ 2)

ง.) การ์ดจะทำการพิสูจน์สัญญาณ Response ของ RWD จากการ Comparing และจากนั้นจะทำการคำนวณสัญญาณ Response เพื่อที่จะส่งต่อ (ขั้นตอนที่ 3)

จ.) RWD ทำการพิสูจน์สัญญาณ Response ของการ์ดจากการ Comparing

หลังจากนั้น การส่งผ่านข้อมูลของการสุ่มครั้งแรกในการติดต่อสื่อสารระหว่างการ์ดและ RWD จะถูกสร้างรหัสลับขึ้นมา

3.2 การออกแบบและการสร้างชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

3.2.1 การทำงานของชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

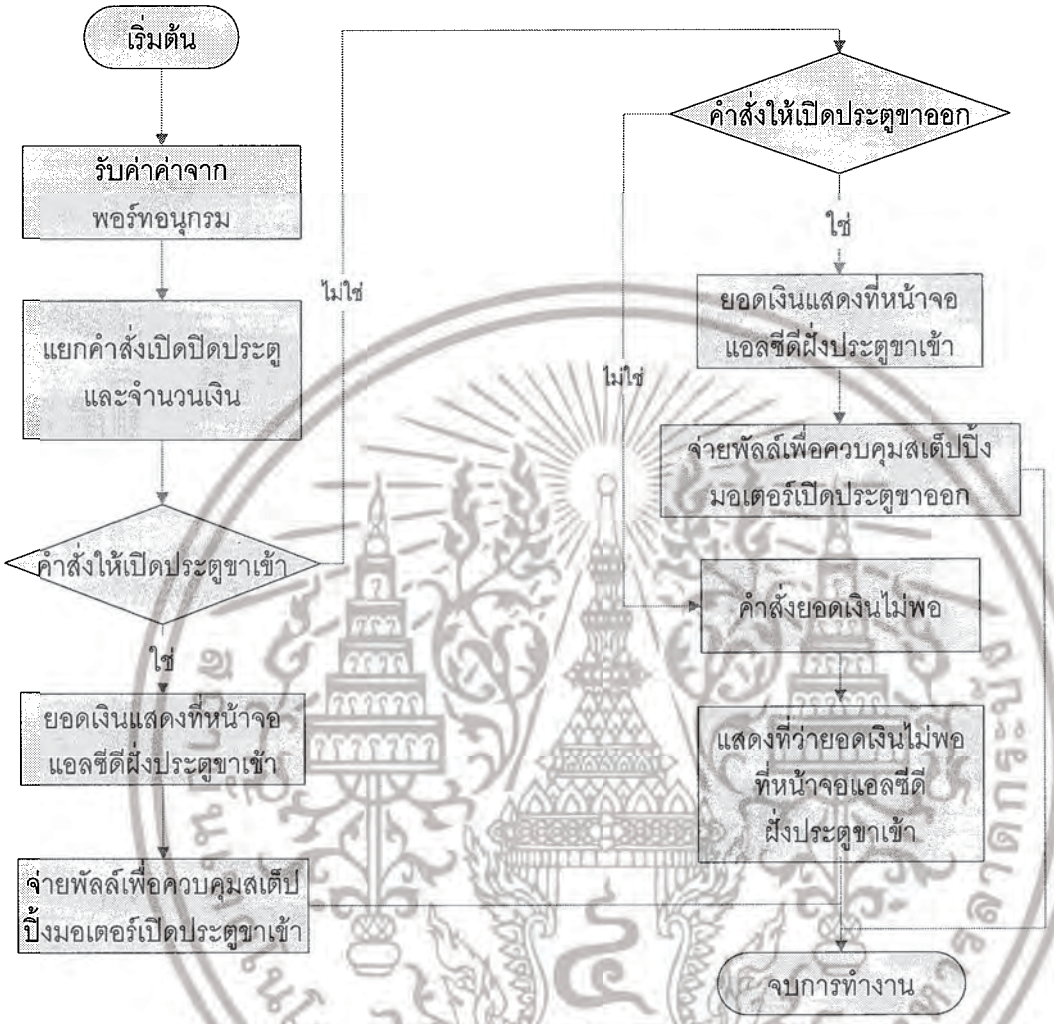
ชุดควบคุมประตูการเปิด-ปิดประตูทางเข้าออก ประกอบไปด้วยไมโครคอนโทรลเลอร์ที่ใช้ในการประมาณผลการทำงานต่างๆ ของโปรแกรมตามคำสั่งของคอมพิวเตอร์ที่ส่งผ่านการเชื่อมต่อตามมาตรฐาน RS-232 เพื่อควบคุมให้มีการเปิดหรือปิดประตู โดยใช้วงจรสเต็ปปิ้งมอเตอร์เป็นกลไกให้ประตูทำงานเปิด-ปิด

การแสดงผลจะแสดงผลด้วย LCD เพื่อแสดงผลการทำงานต่างๆ ที่ได้รับคำสั่งจากคอมพิวเตอร์และแสดงค่าจำนวนเงินของผู้ใช้บริการ ส่วนของประตูเข้า-ออก จะทำการเปิดประตูหลังการแสดงจำนวนครั้งของผู้ใช้งานที่จอ LCD แล้วตามคำสั่งไมโครคอนโทรลเลอร์

ชุดควบคุมการเปิด-ปิดประตูทางเข้าออก ถ้าเป็นชุดทางเข้า LCD จะแสดงข้อความจำนวนครั้งที่เหลือจากการใช้บริการ แต่ถ้าเป็นด้านออกจะมีการอ่านรหัสบัตรที่ถูกต้องและค้นหาข้อมูลปรับปรุงจำนวนเงินในฐานข้อมูลเสร็จแล้วก็จะแสดงออกมาที่ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ผังการทำงานชุดควบคุมการเปิด-ปิดประตูทางเข้าออก



รูปที่ 3.2 ผังการทำงานโปรแกรมส่วนของการควบคุมการเปิด-ปิดประตูทางเข้าออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

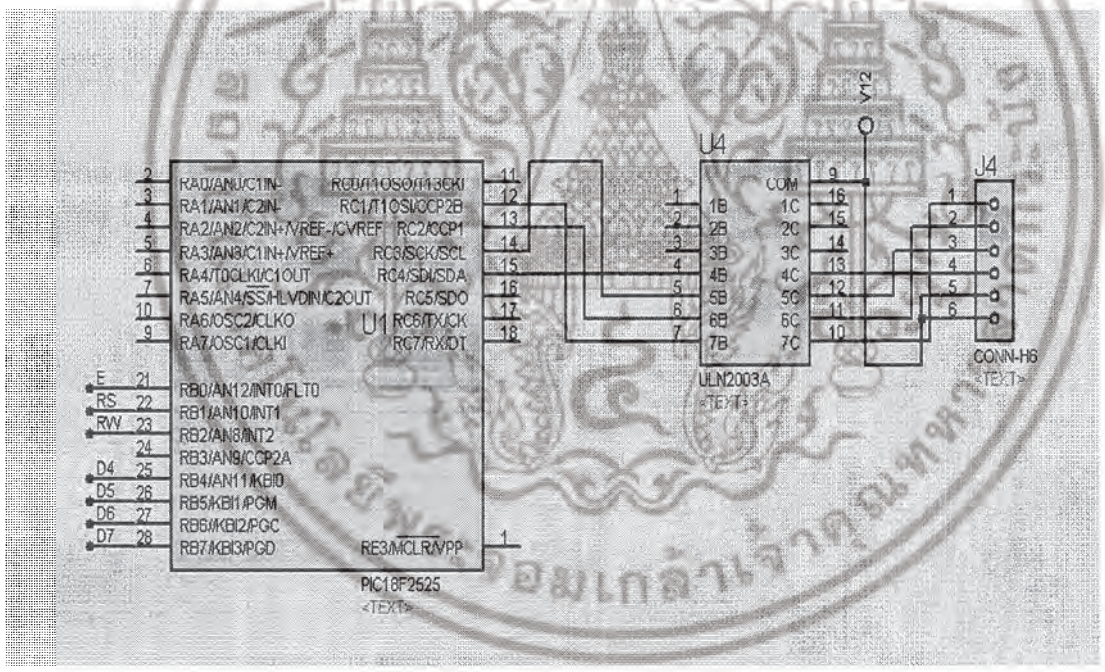
3.2.3 อธิบายโปรแกรมส่วนของการควบคุมการเปิด-ปิดประตูทางเข้าออก

เริ่มต้นโปรแกรมรอสัญญาณจาก serial port โปรแกรมจะตรวจสอบยอดการใช้งาน ถ้าเหลือยอดการใช้งาน โปรแกรมจะส่งคำสั่งการเปิดประตูไปยังไมโครคอนโทรลเลอร์ที่คุควบคุมการเปิดปิดประตู เพื่อเปิดประตู พร้อมทั้งแสดงผล ยอดการใช้ที่เหลือของลูกค้าบนหน้าจอ lcd

การสร้างชุดของการควบคุมการเปิด-ปิดประตูทางเข้าออกแบ่งเป็น 2 ส่วนหลัก คือส่วนการเปิด-ปิดประตูด้วยวงจรสเต็ปปิ้งมอเตอร์ และส่วนของการแสดงผลการเข้าออกที่จอแอลซีดี

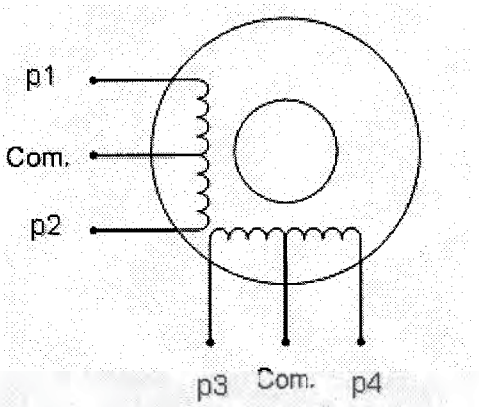
3.2.4 หลักการสร้างส่วนการเปิด-ปิดประตูด้วยวงจรสเต็ปปิ้งมอเตอร์

มอเตอร์เป็นอุปกรณ์ที่ต้องการกระแสมากเกินกว่าที่ไมโครคอนโทรลเลอร์จะให้ได้จึงต้องใช้วงจร (drive) ช่วยในการจ่ายกระแสแทน ในที่นี้ใช้ไอซีเบอร์ ULN 2003A ซึ่งมีวงจรขับ 7 ช่อง โดยแต่ละช่องสามารถจ่ายกระแสได้สูงสุด 0.5 A การควบคุมแบบ unipolar จะต้องใช้สัญญาณควบคุม 4 เส้น ซึ่งสามารถออกแบบวงจรได้ดังรูปที่ 3.3 โดยสาย common ของมอเตอร์ต่ออยู่กับแรงดัน 12V



รูปที่ 3.3 รูปแบบวงจรของสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจรภายในของสเต็ปปิ้งมอเตอร์

การเขียนโปรแกรม

เริ่มต้นด้วยการกำหนดให้พอร์ตที่ต้องการใช้ควบคุมมอร์เตอร์จำนวน 4 บิตเป็นพอร์ตเอาท์พุทของแต่ละประตู จากนั้นใน main loop เขียนคำสั่งควบคุมให้ on หรือ off สัญญาณของแต่ละพอร์ทตามลำดับของขดลวด จะทำให้มอร์หมุนไปตามจำนวน step ที่เราต้องการ

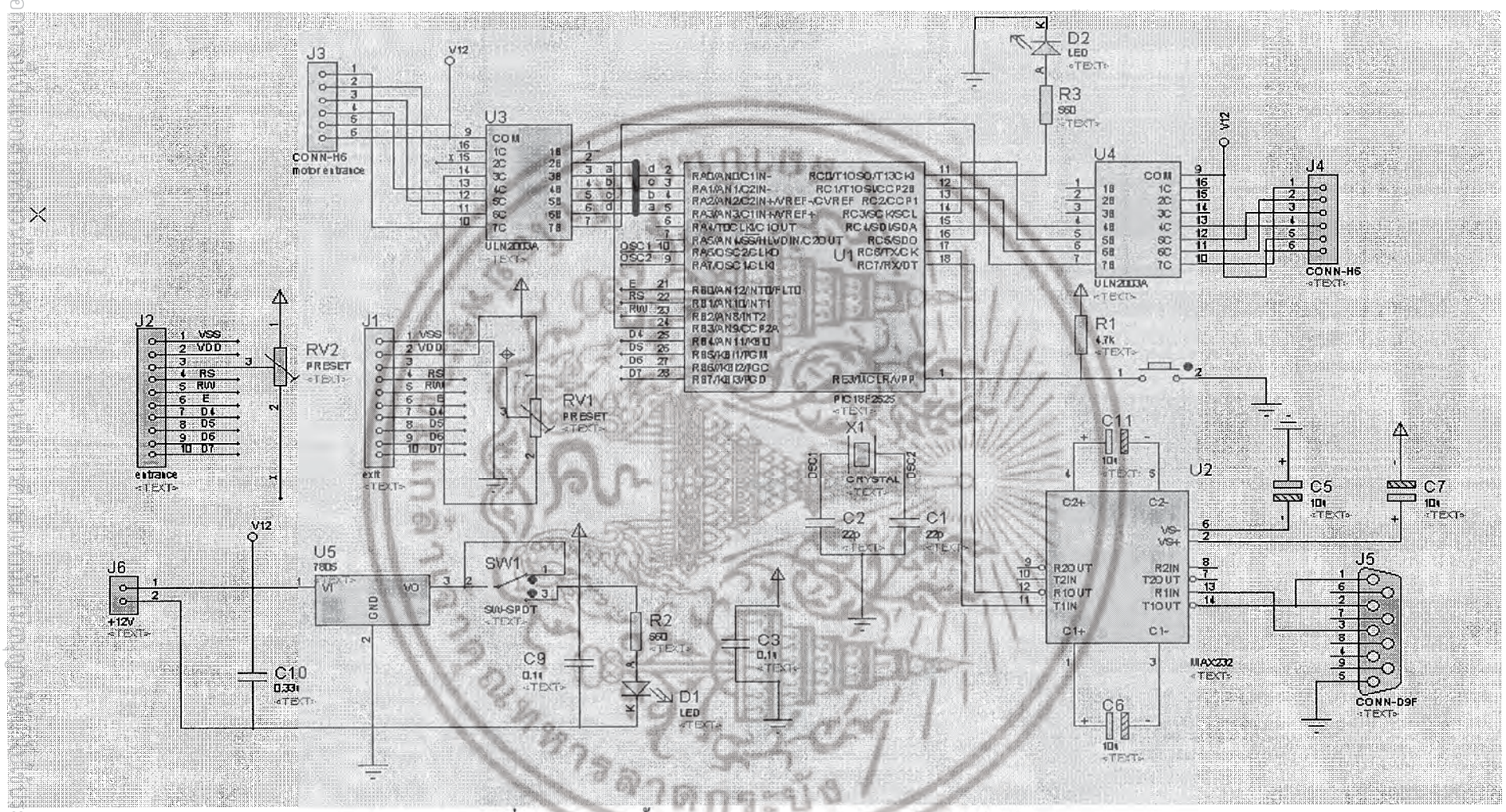
ในการออกแบบจะใช้พอร์ท RA0-RA3 เป็นพอร์ตสำหรับต่อวงจรของสเต็ปปิ้งมอเตอร์ของประตูทางเข้า และใช้พอร์ท RC1-RC5 เป็นพอร์ทสำหรับต่อวงจรของสเต็ปปิ้งมอเตอร์ของประตูทางออก ซึ่งเป็นสัญญาณควบคุมซึ่งต่อไปยัง P1 ,P2 ,P3 ,P4 ของสเต็ปปิ้งมอเตอร์

เนื่องจากการ on หรือ off สัญญาณแต่ละครั้งรวดเร็วมากจนกระแสไฟฟ้าไม่พอที่จะสร้างสนามแม่เหล็กเหนี่ยวนำแกนหมุน (rotor) ให้หมุนไปได้จึงจำเป็นจะต้องใส่ฟังก์ชันหรือกระบวนการอื่นๆ เพื่อทำการยืดเวลาระหว่างการ on หรือ off สัญญาณแต่ละครั้งออกไปให้ห่างกันพอสมควรซึ่งเป็นส่วนสำคัญในการกำหนดความเร็วการหมุนของมอร์เตอร์ด้วย หากระยะเวลาห่างกันเกินไปมอเตอร์ก็จะหมุนช้ามาก หากระยะเวลาสั้นเกินไปมอเตอร์ก็อาจไม่สามารถหมุนได้

ในกรณีที่เราต้องการให้มอเตอร์หมุนไปด้วยขนาดองศาที่เราต้องการ เราจะต้องรู้ว่ามอเตอร์มี stator ทั้งหมดกี่อัน โดยหมุนแกนของมอเตอร์จะรู้สึกว่าเป็นจังหวะ ให้นับจำนวนจังหวะในหนึ่งรอบก็จะได้เท่ากับจำนวน stator หรืออีกวิธีหนึ่งคือ เขียนโปรแกรมทดลองเพื่อให้มอเตอร์หมุนไปช้าๆ ก็สามารถเห็นจังหวะได้อย่างชัดเจนเช่นกัน ถ้ามี stator 20 อัน ใน 1 รอบการหมุนของมอเตอร์แสดงว่าเราสามารถกำหนดให้มอเตอร์หมุนไปด้วยความละเอียด = 360/48 = 7.5 องศา หรือคือ 1 สเต็ปการหมุนนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การออกแบบวงจร



รูปที่ 3.5 วงจรรวมทั้งหมดของวงจรชุดควบคุมการเปิด-ปิดประตู

3.3 หลักการสร้างฐานข้อมูลของบริการรถไฟฟ้า

การเก็บข้อมูลของผู้ใช้บริการนั้น ฐานข้อมูลที่ใช้เขียนด้วยโปรแกรม SQL Server ซึ่งเป็นโปรแกรมสร้างฐานข้อมูล ซึ่งในรายงานนี้จะสร้างตารางฐานข้อมูลไว้สองตาราง

3.3.1 ตารางฐานข้อมูลสำหรับรายละเอียดของผู้ใช้บริการ

การออกแบบในตารางประกอบไปด้วยข้อมูลดังนี้

1. เก็บรหัสบัตรของผู้ใช้บริการ
2. ชื่อและนามสกุลของผู้ใช้บริการ
3. เบอร์โทรศัพท์ของผู้ใช้บริการ เพื่อเป็นข้อมูลให้เจ้าของเครือข่ายโทรศัพท์มือถือ
4. ยอดการใช้งาน
5. สถานการณ์ใช้งาน

3.3.2 ตารางฐานข้อมูลสำหรับเก็บรหัสของผู้ดูแลระบบ

การกำหนดตารางฐานข้อมูลสำหรับเก็บรหัสของผู้ดูแลระบบเพื่อเป็นความปลอดภัยของข้อมูลลูกค้า โดยผู้ที่จะเข้าไปแก้ไขข้อมูลจะเป็นผู้ที่มีหน้าที่ดูแลระบบฐานข้อมูลนี้เท่านั้น

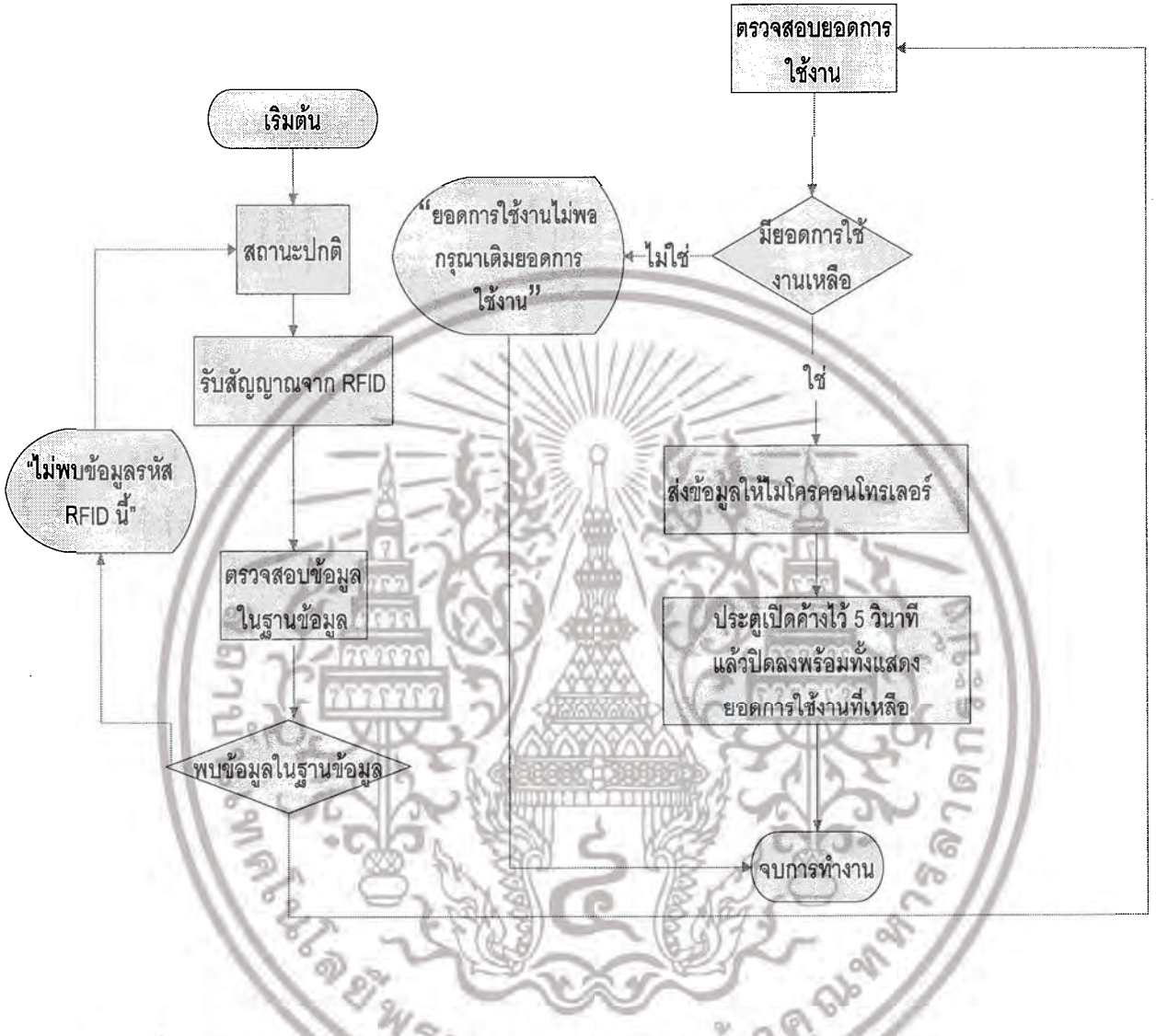
การออกแบบในตารางประกอบไปด้วยข้อมูลดังนี้

1. username
2. password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โปรแกรมการจัดการระบบบัตรรถไฟฟ้า

3.4.1 ผังการทำงานโปรแกรมส่วนหลักของชุดอ่านบัตร RFID



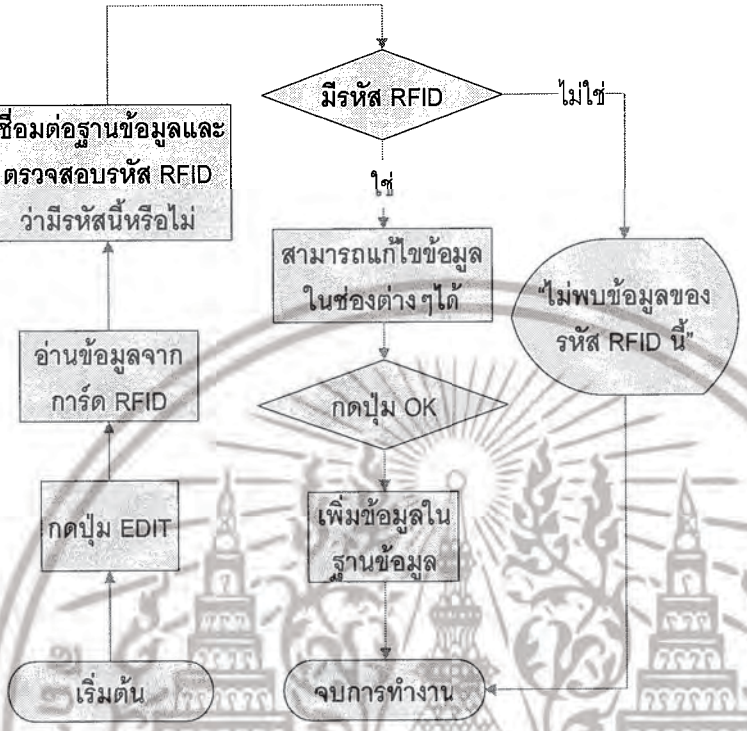
รูปที่ 3.6 ผังการทำงานโปรแกรมส่วนของการควบคุมการเปิดปิดประตูทางเข้าออก

โปรแกรมส่วนของการควบคุมการเปิดปิดประตูทางเข้าออก

เริ่มต้นโปรแกรมรอสัญญาณจากบัตร Tag RFID เมื่อตรวจพบ Tag RFID เครื่อง Reader จะส่งรหัส ของการ์ด RFID เพื่อนำมาใช้ในการตวจสอบรหัสและข้อมูลการใช้ของลูกค้า ถ้ารหัสของการ์ด RFID มีในฐานข้อมูล โปรแกรมจะตรวจสอบยอดการใช้งาน ถ้าเหลือยอดการใช้งาน โปรแกรมจะส่งคำสั่งการเปิดประตูไปยังไมโครคอนโทรลเลอร์ที่คุควบคุมการเปิดปิดประตู เพื่อเปิดประตู พร้อมทั้งแสดงผล ยอดการใช้ที่เหลือของลูกค้าบนหน้าจอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 ผังการทำงานส่วนหลักของโปรแกรมส่วน EDIT



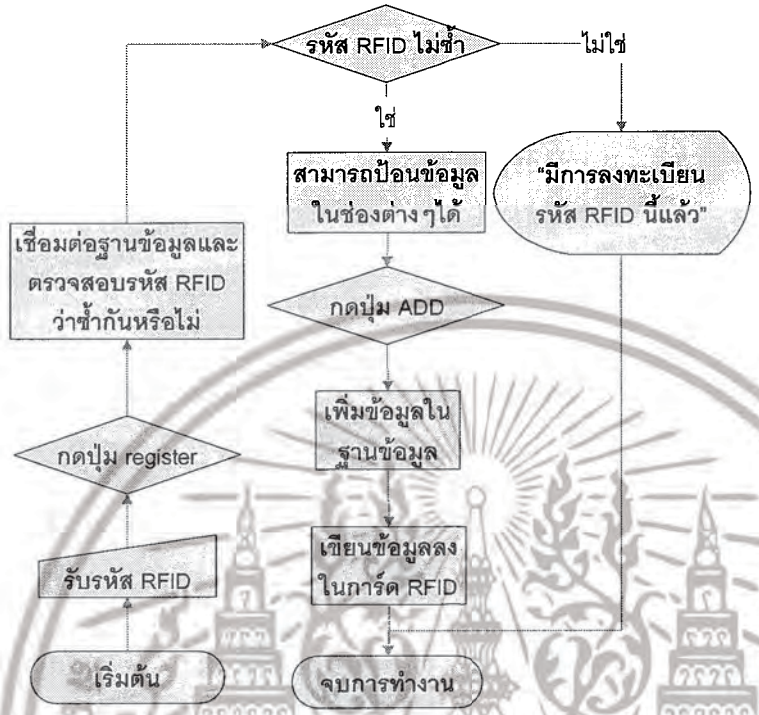
รูปที่ 3.7 หลักการทำงานของโปรแกรมส่วน EDIT

หลักการทำงานของโปรแกรมส่วน EDIT

ส่วนนี้จะทำหน้าที่ในการแก้ไขข้อมูลของเจ้าของรหัส RFID โดยเราต้องกรอกรหัส RFID ให้ตรงกันในฐานข้อมูล จึงจะสามารถแก้ไขข้อมูลของลูกค้าได้ การทำงานของส่วนนี้ เริ่มจากการป้อนรหัส RFID ของลูกค้า และกดปุ่ม EDIT โปรแกรมจะทำการเชื่อมต่อกับฐานข้อมูล เพื่อตรวจสอบว่ารหัส RFID นี้มีข้อมูลอยู่หรือไม่ ถ้ามีก็จะปลดล็อกกล่องข้อความเพื่อที่จะแก้ไขข้อมูลได้ แต่ถ้าไม่พบข้อมูล จะมีข้อความแสดงว่า ไม่พบผู้ใช้รหัสนี้ เมื่อแก้ไขข้อมูลเรียบร้อยให้กดปุ่ม OK ข้อมูลล่าสุดจะถูกแก้ไขและแสดงในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 ผังการทำงานส่วนหลักของโปรแกรมส่วน REGISTER



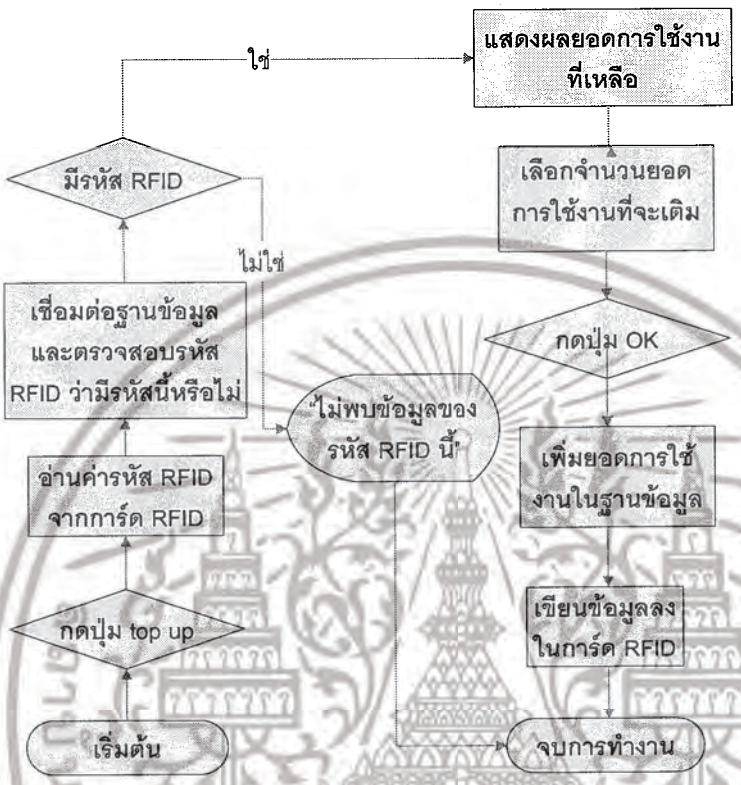
รูปที่ 3.8 หลักการทำงานของโปรแกรมส่วน REGISTER

หลักการทำงานของโปรแกรมส่วน REGISTER

โปรแกรมจะรอรับรหัสจากการ์ดที่จะทำการลงทะเบียน เมื่อได้รับรหัสแล้ว (โดยการกรอกรหัสลงในช่องRFID) ต้องกด ปุ่ม register เพื่อเพิ่มข้อมูลของเจ้าของรหัสการ์ด RFID ในช่องข้อความต่างๆ และเลือกจำนวนการใช้งานเริ่มต้นโดยการเลือกในกล่องข้อความ เมื่อกด ok ข้อมูลทั้งหมดจะส่งไปยังฐานข้อมูลเพื่อไว้เป็นประวัติในการเข้าออกของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 ผังการทำงานส่วนหลักของโปรแกรมส่วน TOP UP



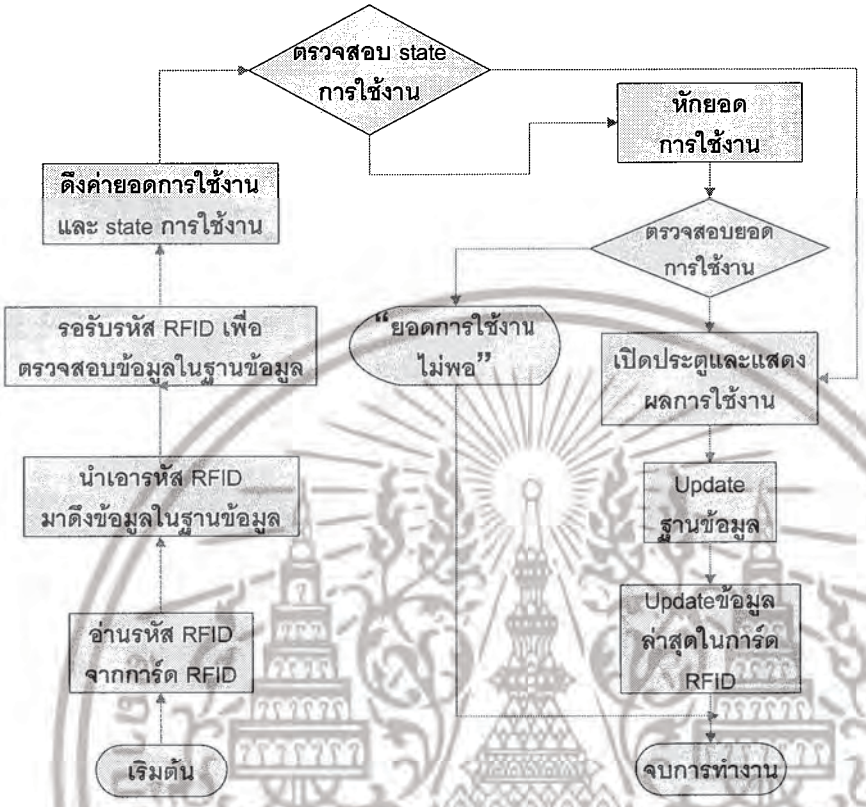
รูปที่ 3.9 หลักการทำงานของโปรแกรมส่วน TOP UP

หลักการทำงานของโปรแกรมส่วน TOP UP

โปรแกรมจะรอรับรหัส rfid เพื่อนำไปดึงข้อมูลยอดการใช้งานล่าสุดของเจ้าของรหัส เมื่อกดปุ่ม top up โปรแกรมจะเชื่อมต่อข้อมูล เพื่อตรวจสอบว่ามีข้อมูลของรหัส RFID นี้หรือไม่ ต่อจากนั้นจะแสดงยอดการใช้งานล่าสุดจะสามารถเลือกจำนวนการใช้งานเพื่อที่จะเติมยอดการใช้งานได้ เมื่อกด ok ยอดการใช้งานล่าสุดจะถูกอัพเดตบนฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 หลักการทำงานของโปรแกรมส่วน OBSERVE



รูปที่ 3.10 หลักการทำงานของโปรแกรมส่วน OBSERVE

หลักการทำงานของโปรแกรมส่วน OBSERVE

โปรแกรมจะรอรับค่ารหัส RFID แล้ว นำไปดึงข้อมูลยอดการใช้งานและสถานะการใช้งานจากฐานข้อมูล โปรแกรมจะเริ่มจากการตรวจสอบสถานะการใช้งานก่อนว่า เจ้าของรหัส RFID เคยได้เข้าระบบแล้วหรือยัง ถ้ายังไม่เคยเข้าระบบ สถานะจะเป็น "off" แล้ว จะเข้าขั้นตอนการตัดยอดการใช้งาน แต่ถ้ากำลังเข้าใช้งานระบบอยู่ สถานะจะเป็น "on" และจะไม่เข้าขั้นตอนการหักยอดการใช้งาน

ขั้นตอนต่อไปคือการหักยอดการใช้งาน ถ้าเจ้าของรหัสมียอดการใช้งานไม่เพียงพอ ประตูจะไม่เปิดและจะแสดงข้อความ ยอดการใช้งานคงเหลือไม่พอ ไม่สามารถเข้าใช้งานได้ แต่ถ้ามียอดเงินเหลือเพียงพอประตูจะเปิด และแสดงข้อความ ขอบคุณ ที่หน้าจอแอลซีดีที่ติดอยู่กับประตู หลังจากนั้นโปรแกรมจะทำการอัตเดตยอดการใชงานและสถานะไว้ในฐานข้อมูลเป็นขั้นตอนสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

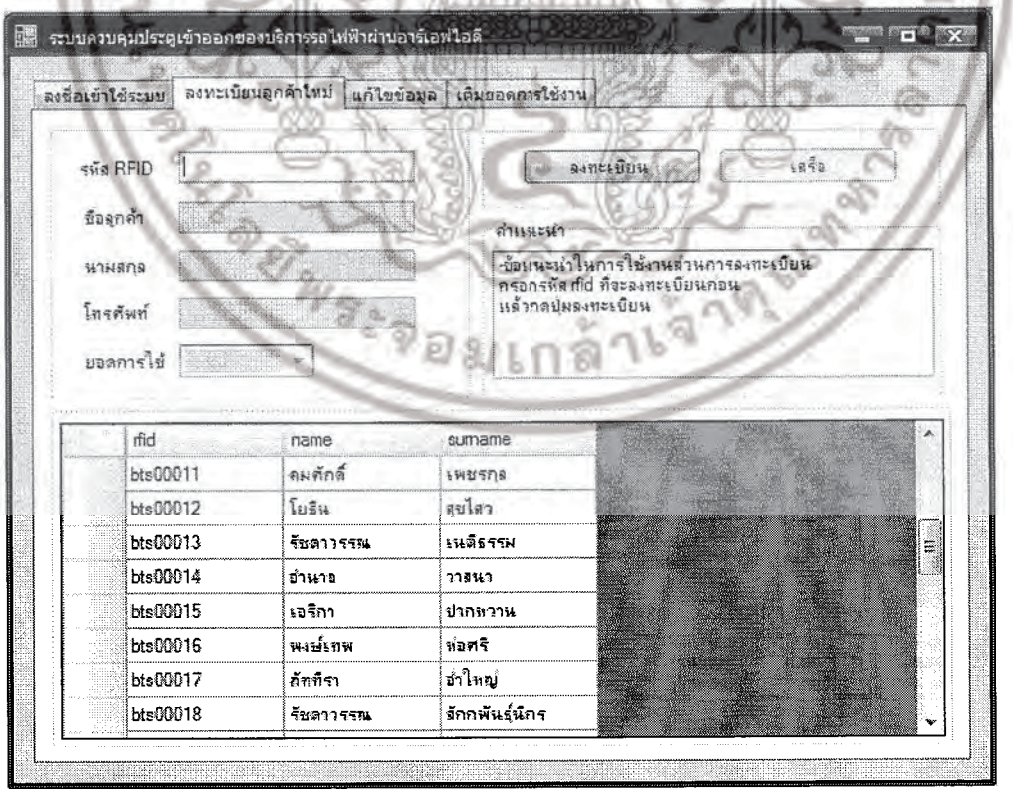
4.1 การทดลองในส่วนของโปรแกรมและฐานข้อมูล

4.1.1 โปรแกรมการลงทะเบียน (Register)

โปรแกรมการลงทะเบียนใช้งาน ใช้ในการลงทะเบียนเพื่อรับรหัส ID ใหม่ โดยจะทำหน้าที่กำหนดรหัสใหม่ใน Tag และทำการลงทะเบียนและเก็บข้อมูลส่วนตัวของผู้ใช้งานลงฐานข้อมูล

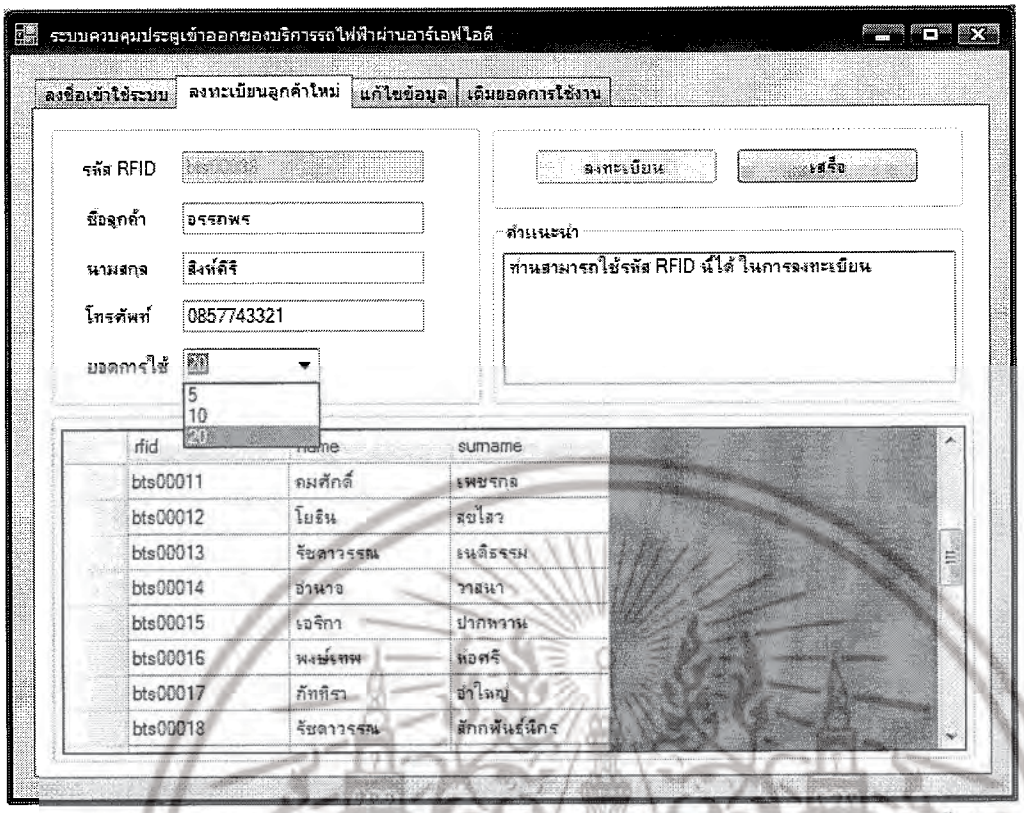
ลำดับขั้นตอนการทดลอง

1. ทำการกำหนดรหัส RFID ใหม่ให้กับผู้ใช้บริการ แล้วกดปุ่ม " ลงทะเบียน "พร้อมทั้งทำการสแกน Tag ที่เครื่องอ่าน คอมพิวเตอร์จะทำการตรวจสอบรหัส หากพบว่ารหัสไม่ซ้ำจะแสดงผลดังรูป 4.1
2. ทำการกรอกข้อมูลส่วนตัวและหมายเลขโทรศัพท์ของผู้ที่ต้องการลงทะเบียน พร้อมทั้งกำหนดเครดิตจำนวนครั้งของการใช้บริการในครั้งแรก แล้วกดปุ่ม " เสร็จ " โปรแกรมจะทำการประมวลผล แล้วแสดงคำถามเพื่อยืนยันบนหน้าจอคอมพิวเตอร์ดังรูป 4.2
3. ทำการยืนยันการลงทะเบียน โปรแกรมจะแสดงผลให้ทราบ และทำการลงทะเบียนด้วยรหัส RFID ใหม่ พร้อมทั้ง Update ข้อมูลลงในฐานข้อมูล ดังรูป 4.3 และ 4.4 เป็นการสิ้นสุดขั้นตอนการลงทะเบียน

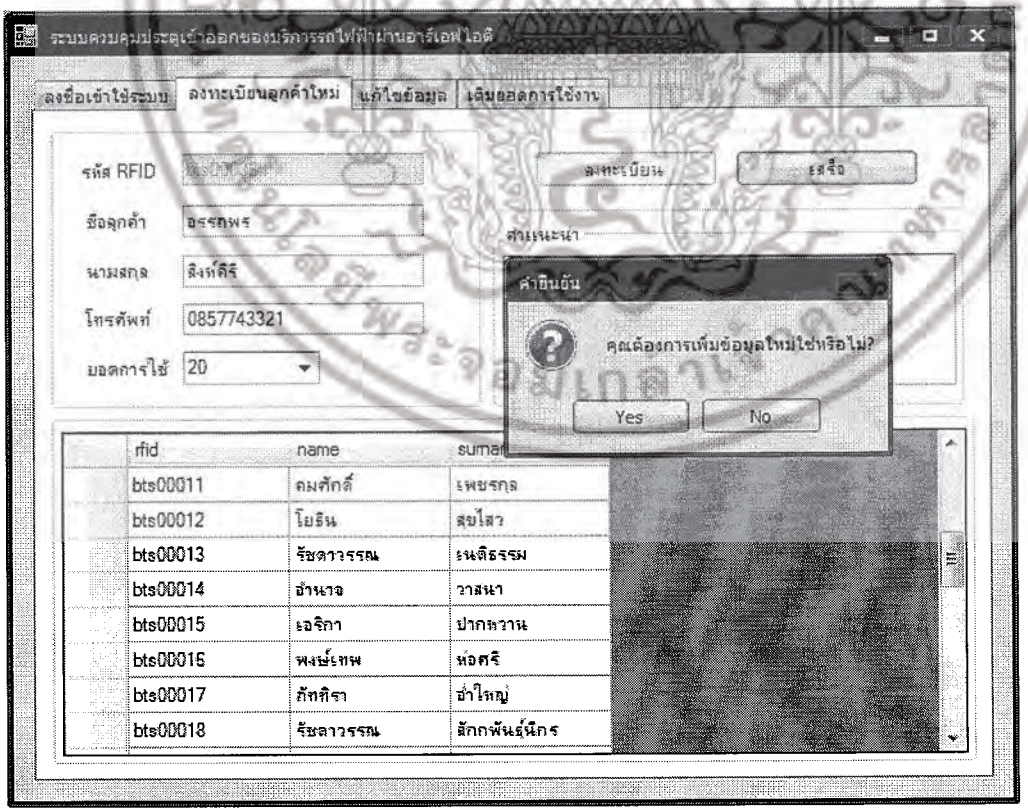


รูปที่ 4.1 หน้าจอเตรียมพร้อมส่วนของการลงทะเบียนลูกค้าใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

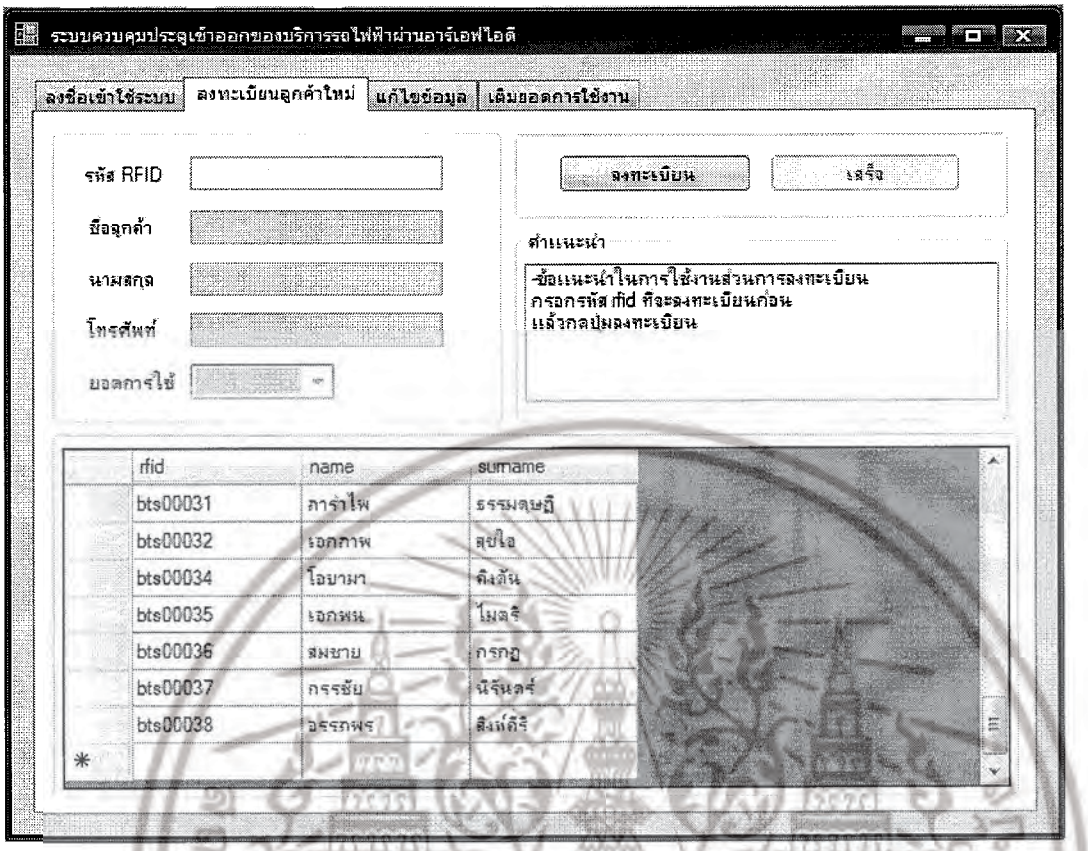


รูปที่ 4.2 หน้าจอลงทะเบียนลูกค้าใหม่เมื่อมีการ Tag ของ RFID และทำการเลือกจำนวนครั้งเริ่มต้น



รูปที่ 4.3 การยืนยันในการขอลงทะเบียนเพื่อรับรหัส ID ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การเพิ่มรหัสใหม่ที่ได้ทำการลงทะเบียนลงในฐานข้อมูล

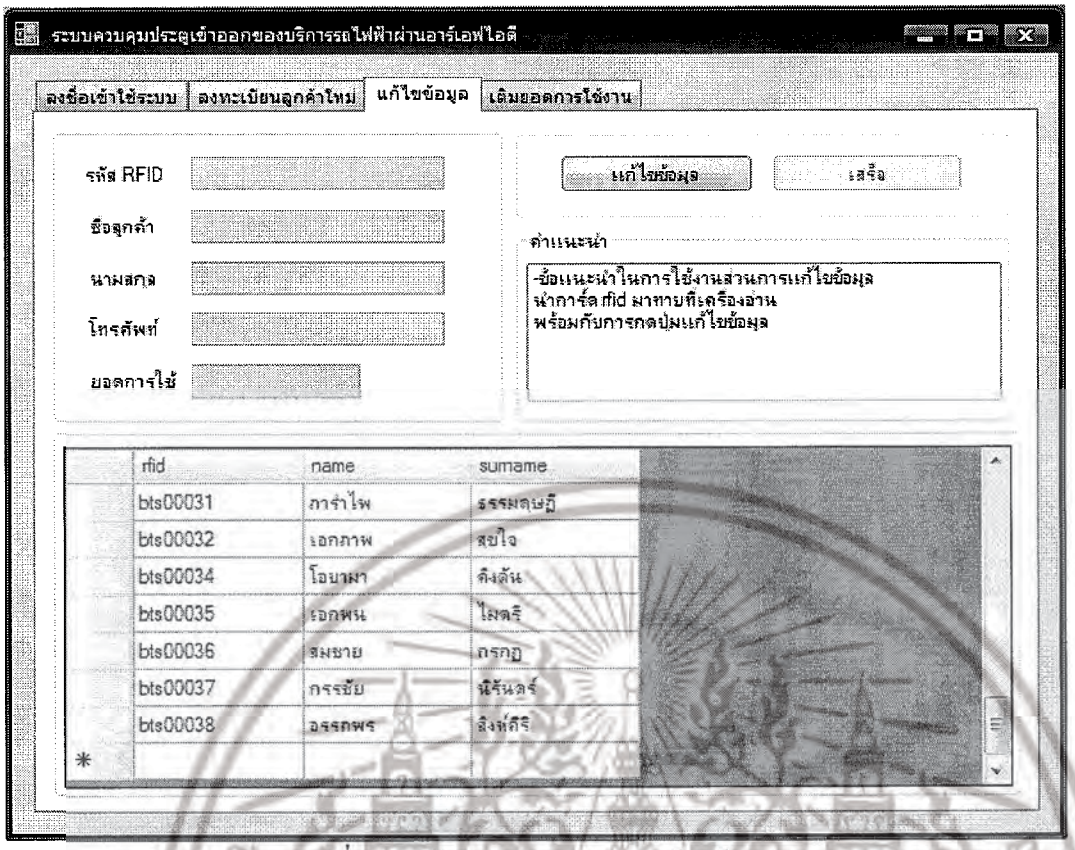
4.1.2 โปรแกรมการแก้ไขข้อมูล (Edit)

โปรแกรมการแก้ไขข้อมูล ใช้ในการแก้ไขหรือเปลี่ยนแปลงข้อมูลใหม่ของผู้ใช้บริการในกรณีที่ผู้ใช้มีการเปลี่ยนแปลงข้อมูลส่วนตัวต่าง ๆ ซึ่งโปรแกรมส่วนนี้จะทำหน้าที่แก้ไขข้อมูลพร้อมทั้ง Update ข้อมูลใหม่ลงในฐานข้อมูล

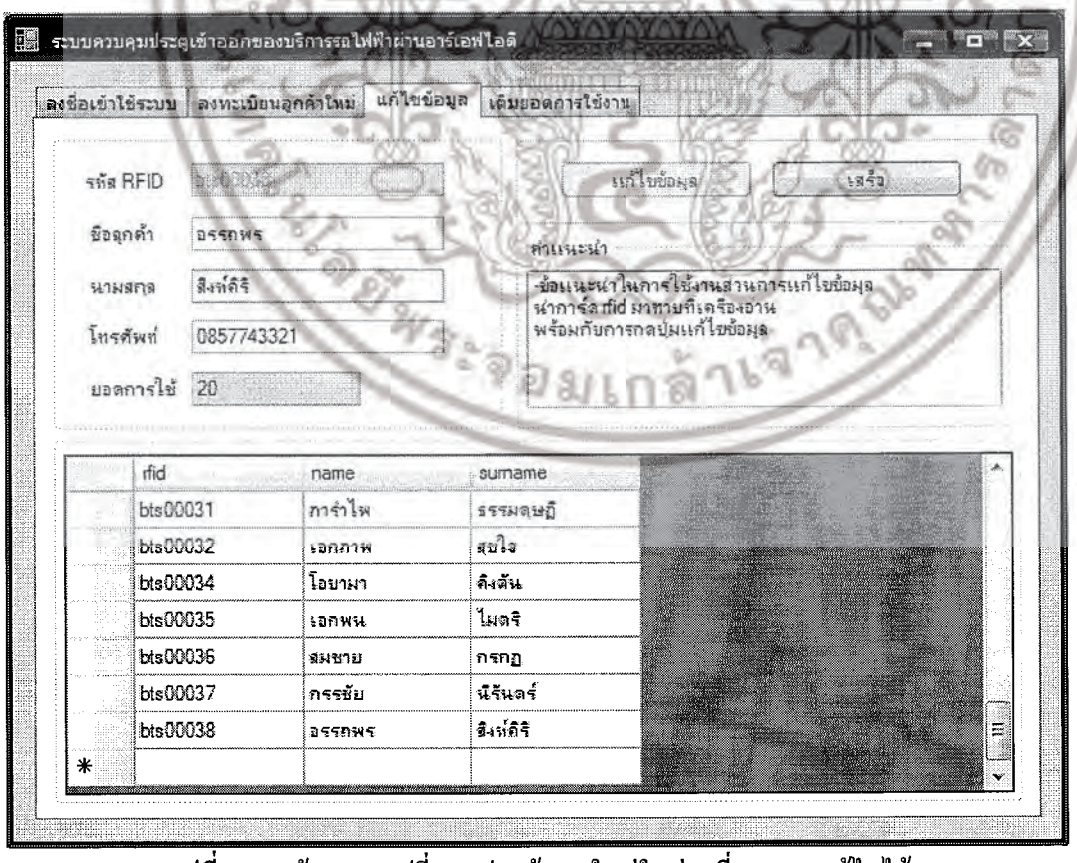
ลำดับขั้นตอนการทดลอง

1. เข้าสู่ส่วนของโปรแกรม Edit ทำการแสกน Tag พร้อมทั้งกดปุ่ม " แก้ไขข้อมูล " โปรแกรมจะทำการอ่าน Tag ผ่าน Reader แล้วแสดงข้อมูลส่วนตัวของผู้ใช้บนหน้าจอคอมพิวเตอร์ ดังรูป 4.5
2. ทำการเปลี่ยนแปลงแก้ไขข้อมูลลงในช่องที่ต้องการแก้ไข ดังรูป 4.6
3. เมื่อกดปุ่ม " เสร็จ " จะทำการแสดงคำถามเพื่อยืนยันการแก้ไขข้อมูลดังรูป 4.7
4. ตอบตกลงเพื่อยืนยันการแก้ไขข้อมูล โปรแกรมจะแสดงผล พร้อมทั้งทำการบันทึกข้อมูลใหม่และ Update ข้อมูลลงในฐานข้อมูล ดังรูป 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

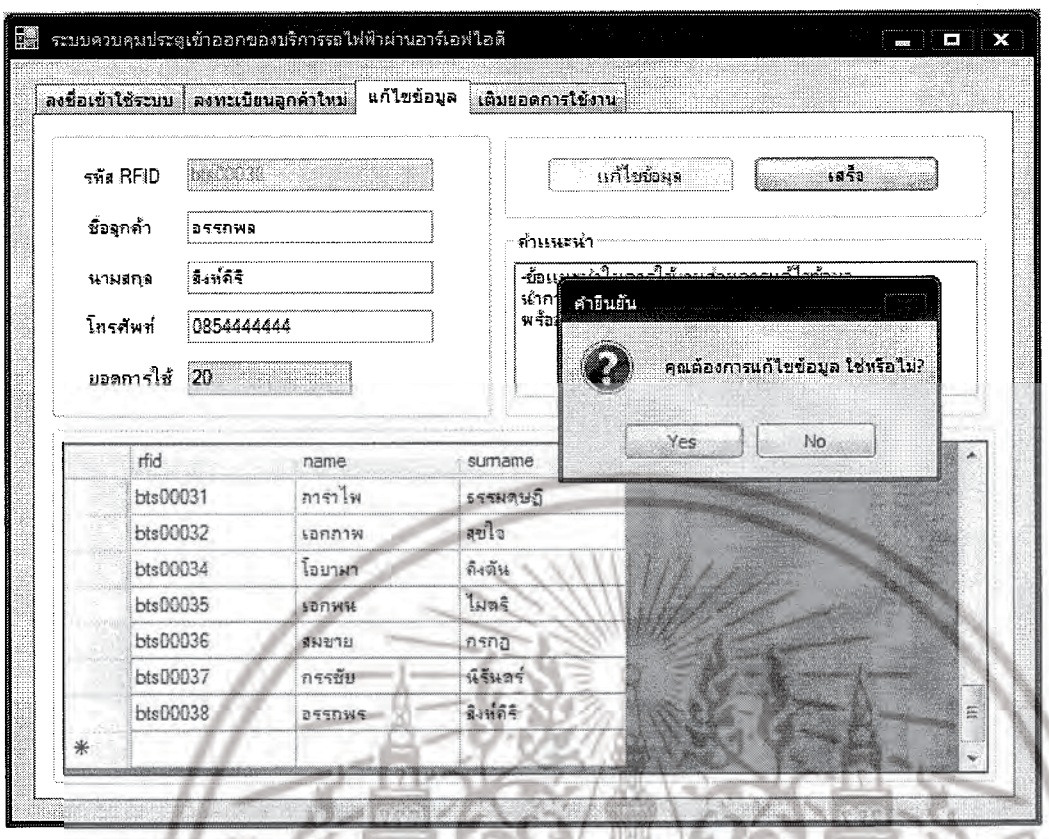


รูปที่ 4.5 หน้าจอในส่วนของการแก้ไขข้อมูล

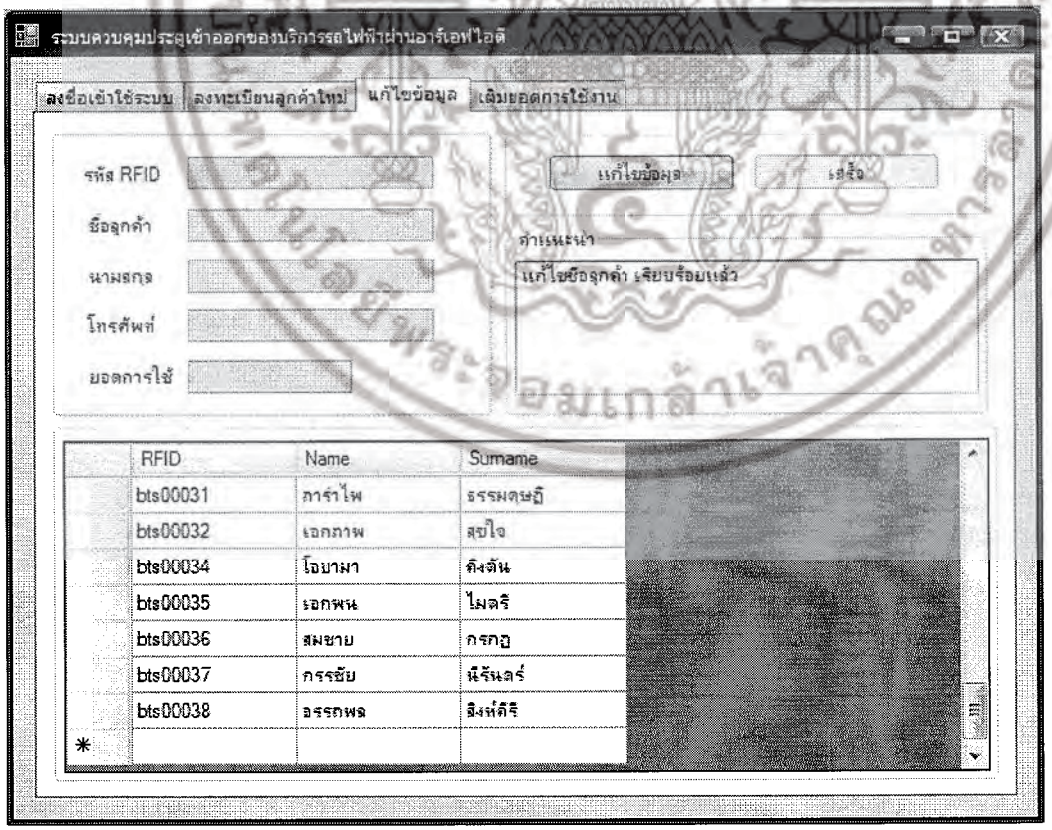


รูปที่ 4.6 หน้าจอการเปลี่ยนแปลงข้อมูลใหม่ในช่องที่สามารถแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 โปรแกรมทำการแสดงคำยืนยันในการแก้ไขหรือเปลี่ยนแปลงข้อมูล



รูปที่ 4.8 การเปลี่ยนแปลงข้อมูลใหม่พร้อมทั้งทำการ Update ในฐานข้อมูล

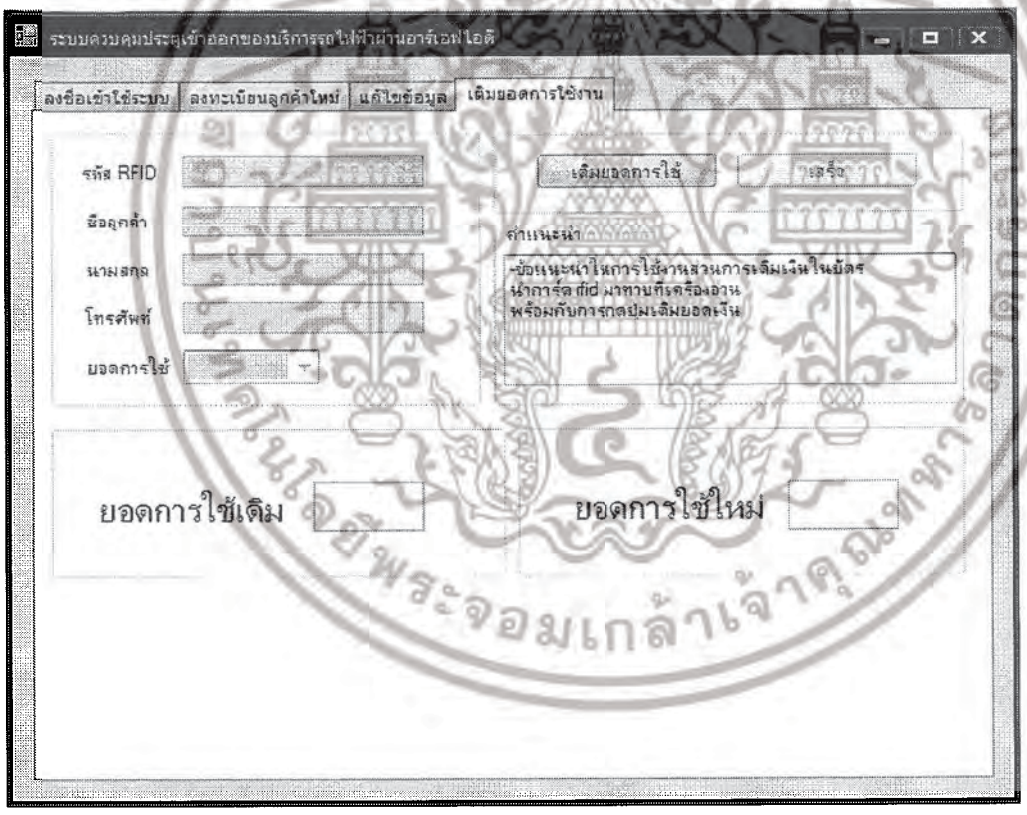
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 โปรแกรมการเติมเงินเพื่อเพิ่มยอดการใช้งาน (Top Up)

โปรแกรมการเติมเงินเพื่อเพิ่มยอดการใช้งาน ใช้สำหรับเติมยอด Credit ของจำนวนครั้งการใช้งาน ในกรณีที่ยอดการใช้งานหมด หรือต้องการเพิ่มยอดการใช้งาน ผู้ใช้บริการสามารถนำรหัส RFID มาทำการเติมยอดการใช้งานได้ใหม่ด้วยโปรแกรมส่วนนี้

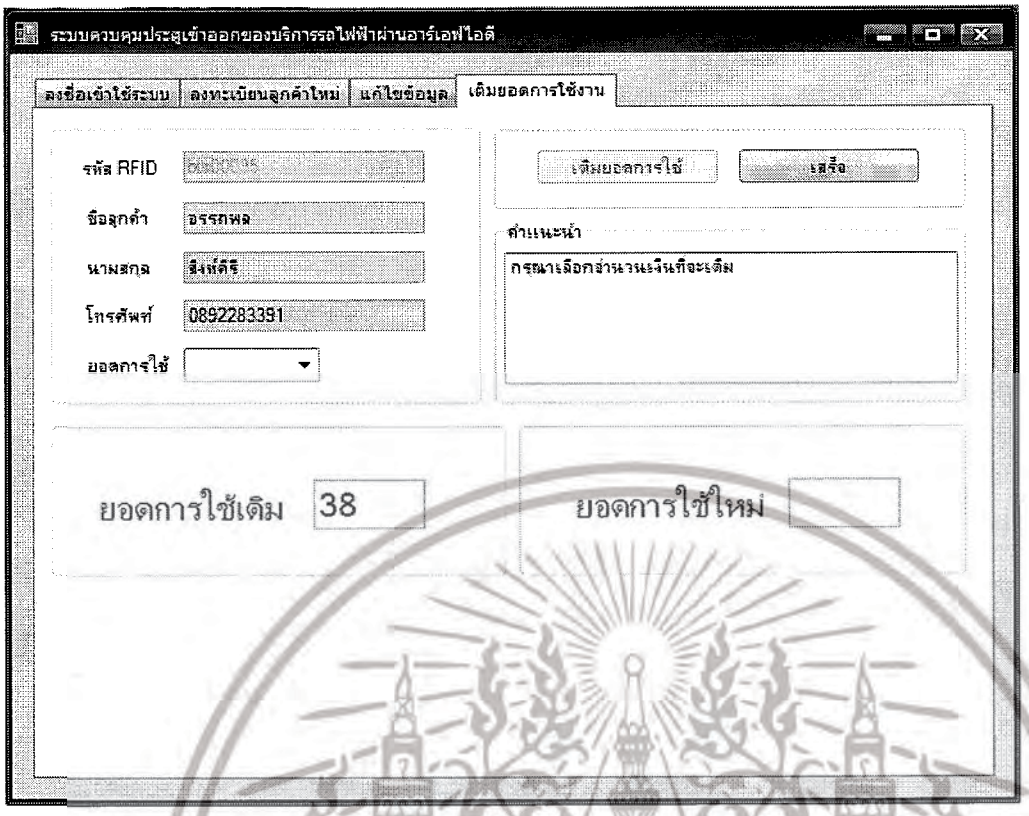
ลำดับขั้นตอนการทดลอง

1. เปิดโปรแกรมในส่วนของ Top Up ดังรูป 4.9
2. ทำการสแกน Tag พร้อมกับกดปุ่ม " เติมยอดการใช้ " โปรแกรมจะทำการประมวลผลแล้วแสดงยอดการใช้ที่คงเหลืออยู่ในบัตร ดังรูป 4.10
3. ทำการเลือกจำนวนเครดิตที่ต้องการใช้งานเพิ่มดังรูป 4.11 และกดปุ่ม " เสร็จ "
4. ยืนยันการเติมยอดเครดิต โปรแกรมจะทำการประมวลผลแล้วแจ้งให้ทราบว่าได้ทำการเติมยอดการใช้งานแล้ว พร้อมทั้งทำการ Update ยอดเงินใหม่ ลงในฐานข้อมูล ดังรูป 4.12

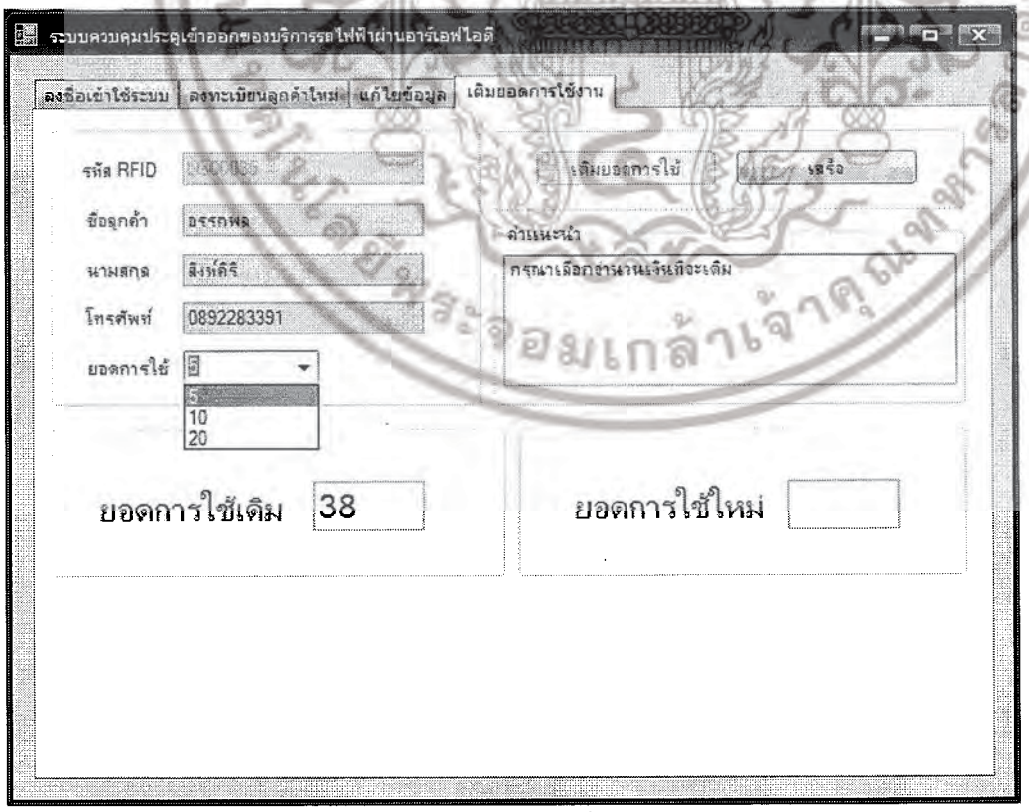


รูปที่ 4.9 หน้าจอเตรียมพร้อมสำหรับการเติมยอดการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

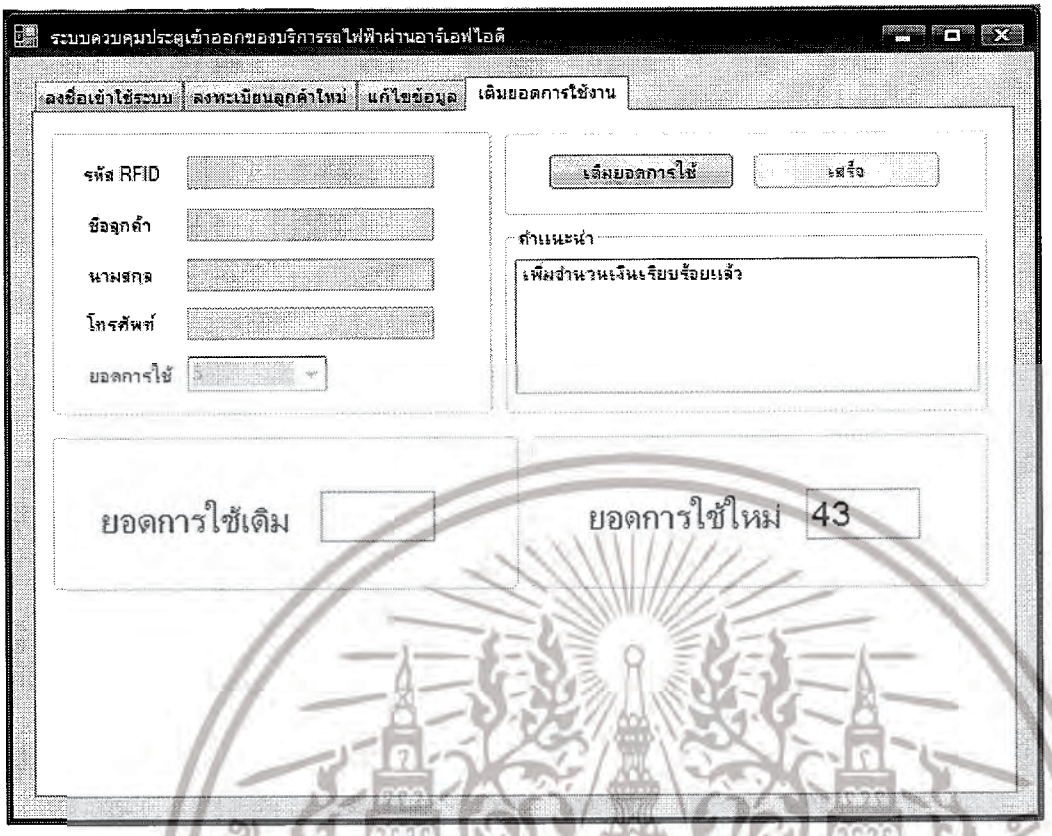


รูปที่ 4.10 หน้าจอเมื่อมีการ Tag ของ RFID



รูปที่ 4.11 การกรอกจำนวนครั้งการใช้ที่ต้องการเพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ยอดเครดิตจำนวนครั้งการใช้สุทธิใน RFID

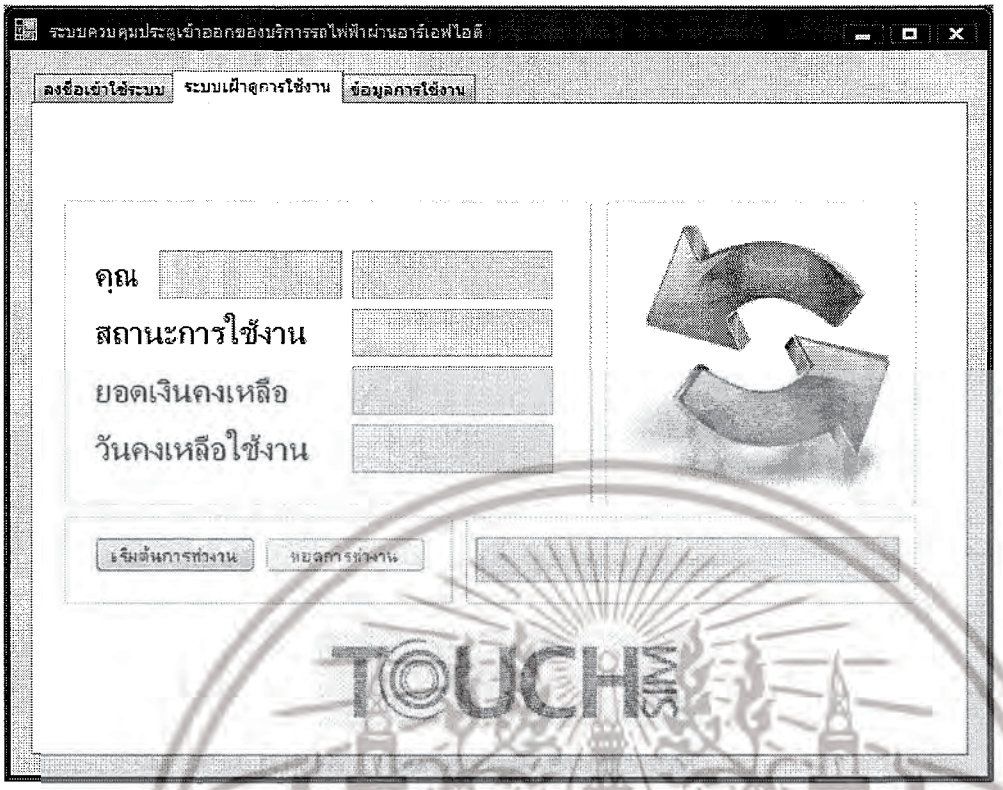
4.1.4 โปรแกรมสังเกตการณ์ (Observe)

โปรแกรมสังเกตการณ์ ใช้ในการหักยอดการใช้บริการในแต่ละครั้ง โดยจะหักลดยอดเครดิตใน RFID ทีละครั้ง เมื่อมีการแสกน Tag เพื่อเข้าใช้บริการ และทำการแสดงผลบนหน้าจอคอมพิวเตอร์ในทุก ๆ ครั้งที่มีการแสกน Tag ในการเข้า – ออก ผ่านประตู

ลำดับขั้นตอนการทดลอง

1. ทำการเข้าสู่ส่วนของโปรแกรมเฝ้าดูการใช้งานจะแสดงหน้าจอเตรียมพร้อมสำหรับการใช้งานดังรูป 4.13
2. เมื่อมีการสแกน Tag ในประตูขาเข้าเพื่อใช้บริการ Reader จะทำการอ่านค่าจาก Tag แล้วส่งให้โปรแกรมทำการประมวลผล เพื่อแสดงสถานะการเข้าใช้งานของผู้ใช้บริการ พร้อมทั้งยอดการใช้ที่คงเหลืออยู่ใน Tag ซึ่งทำการหักลดจากยอดการใช้ที่มีอยู่ก่อนหน้าแล้ว ดังแสดงในรูป 4.14 และ 4.15
3. เมื่อมีการสแกน Tag ที่ประตูขาออก Reader จะทำการอ่านค่าจาก Tag อีกครั้ง แล้วส่งให้โปรแกรมทำการประมวลผล และแสดงสถานะการออกจากการใช้บริการ ดังรูป 4.16 พร้อมทั้งแสดงยอดการใช้ที่คงเหลืออยู่ใน Tag อีกครั้ง เป็นการสิ้นสุดการใช้บริการ

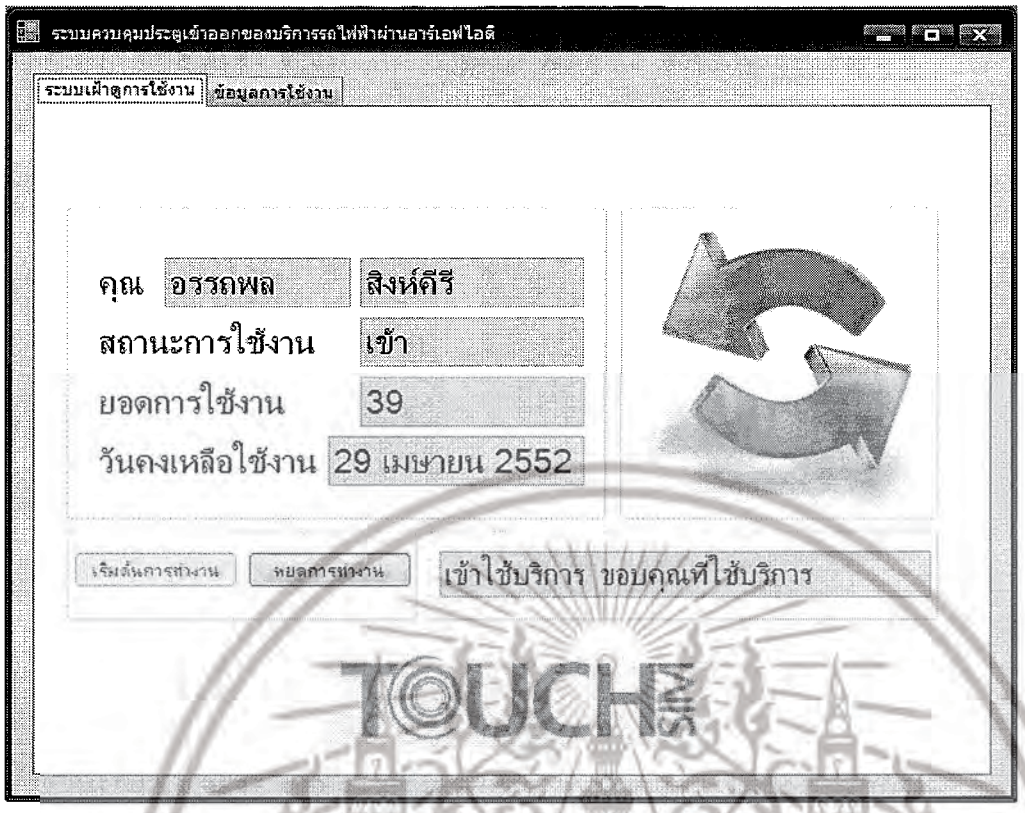
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



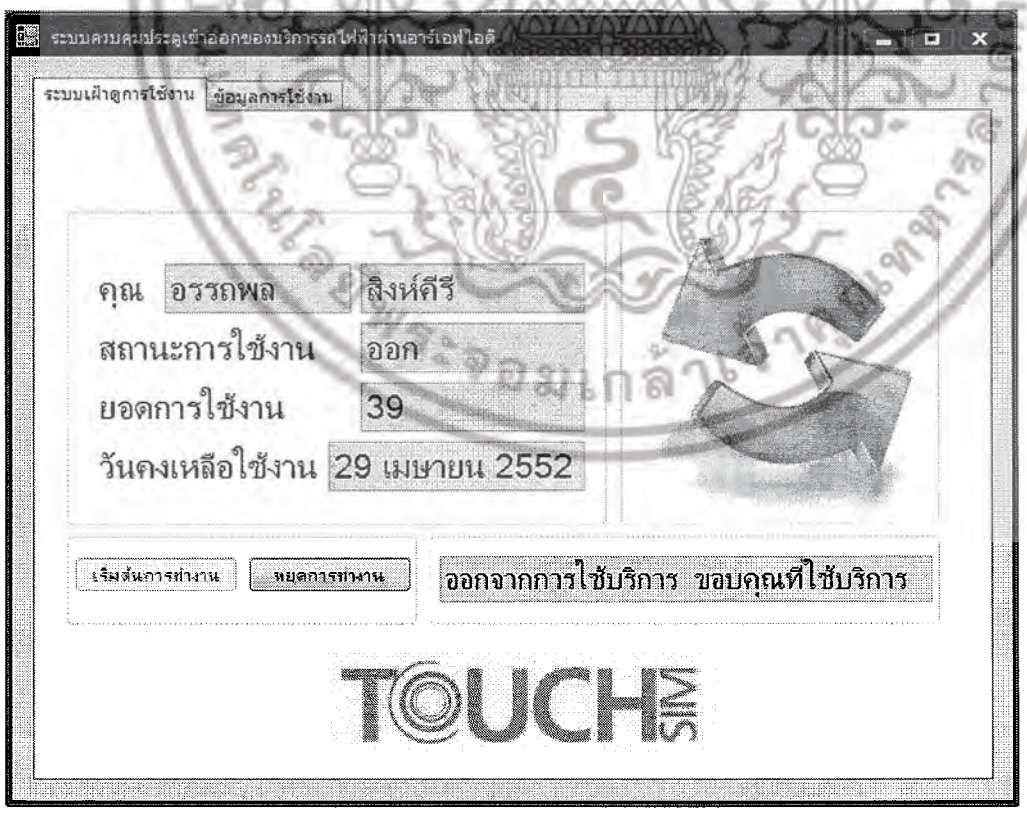
รูปที่ 4.13 หน้าจอเตรียมพร้อมสำหรับการใช้งานส่วนของระบบเฝ้าดูการใช้งาน

รูปที่ 4.14 หน้าจอเดิมก่อนการเข้าใช้บริการ (ยอดการใช้งาน 40 ครั้ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

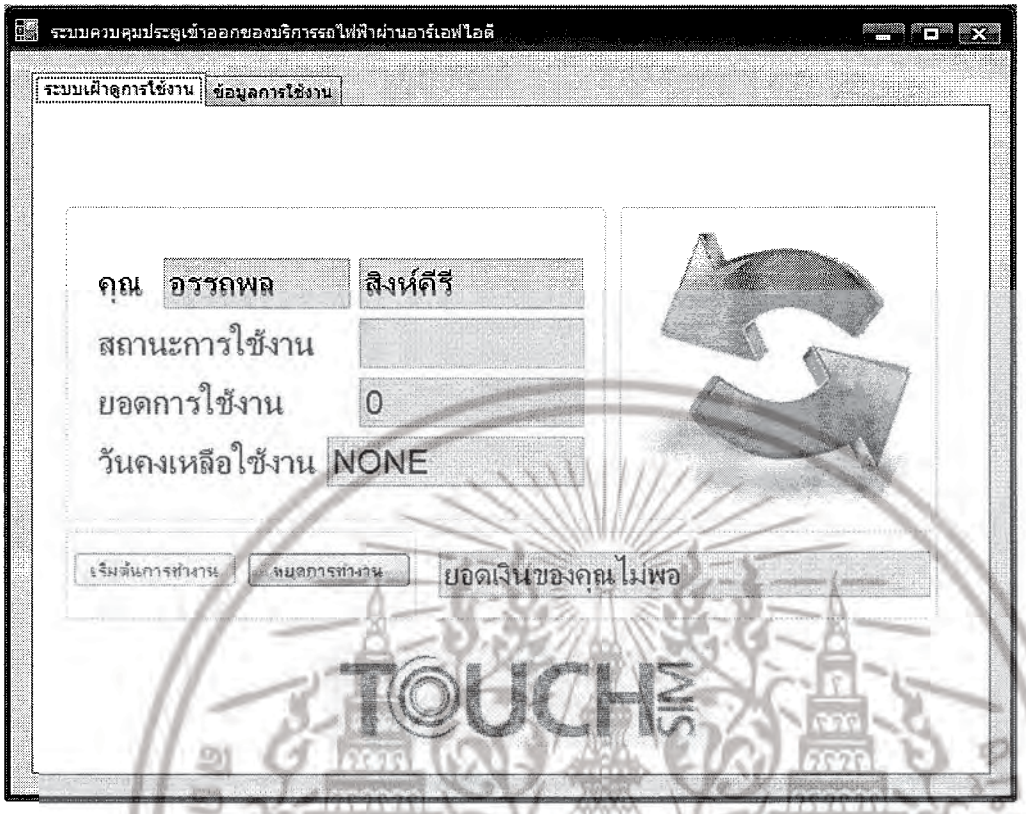


รูปที่ 4.15 สถานะของผู้ใช้งานเมื่อมีการ Tag ของ RFID เพื่อเข้าใช้บริการ (ยอดการใช้งาน 39 ครั้ง)



รูปที่ 4.16 สถานะของผู้ใช้งานเมื่อมีการ Tag ของ RFID เพื่อออกจากการใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 สถานะยอดเงินไม่เพียงพอ เมื่อมีการ Tag ของ RFID เพื่อเข้าใช้บริการ

4.2 ชุดควบคุมการเปิด-ปิดประตูทางเข้าออกและจอแสดงผลแอลซีดี

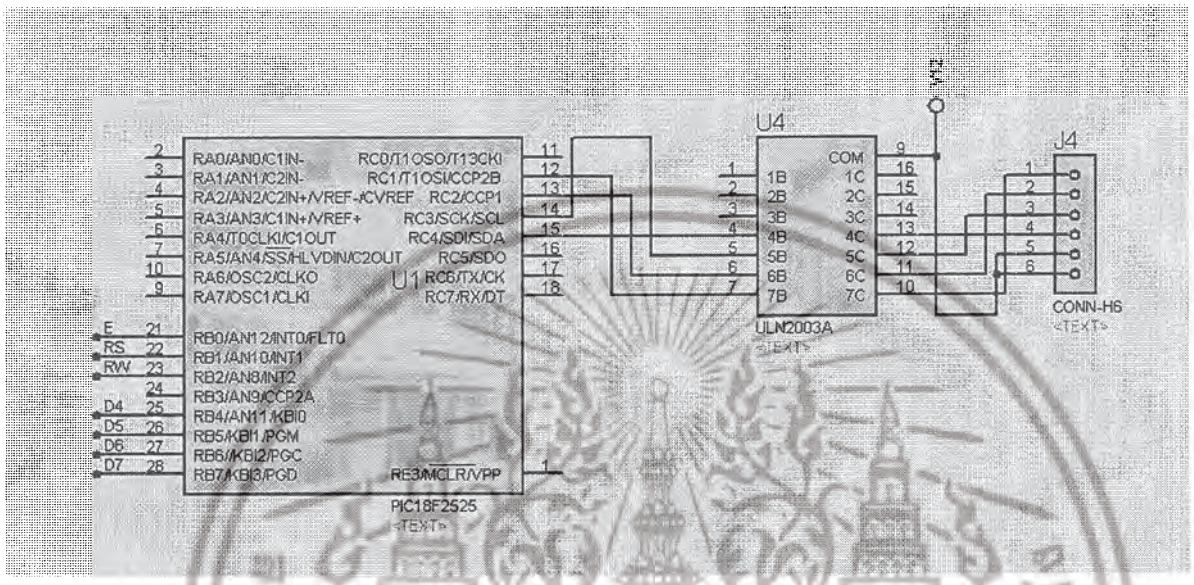
4.2.1 วงจร Stepping Moter

สเตปปิ้งมอเตอร์เป็นอุปกรณ์เอาต์พุตอย่างหนึ่ง ซึ่งสามารถนำไอซีไมโครคอนโทรลเลอร์ มาทำการควบคุมได้สะดวก และเป็นมอเตอร์ที่เหมาะสมสำหรับใช้ในงานควบคุมการหมุน ที่ต้องการตำแหน่งและทิศทางที่แน่นอน การทำงานของ สเตปปิ้งมอเตอร์จะขับเคลื่อนทีละขั้นๆ ละ (Step) ซึ่งขึ้นอยู่กับคุณสมบัติแต่ละชนิดของสเตปปิ้งมอเตอร์ตัวนั้นๆ สเต็ปปิ้งมอเตอร์จะแตกต่างจากมอเตอร์กระแสตรงทั่วไป (DC MOTOR) โดยการทำงานของมอเตอร์กระแสตรงจะหมุนไปแบบต่อเนื่อง ไม่สามารถหมุนเป็นแบบสเต็ปๆ ได้ดังนั้นในการนำไปกำหนดตำแหน่งจึงควบคุมได้ยากกว่า แต่ในส่วนใหญ่เราจะใช้สเตปปิ้งมอเตอร์มาทำการการควบคุมโดยใช้วิธีในระบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.1 ขั้นตอนการทดลอง

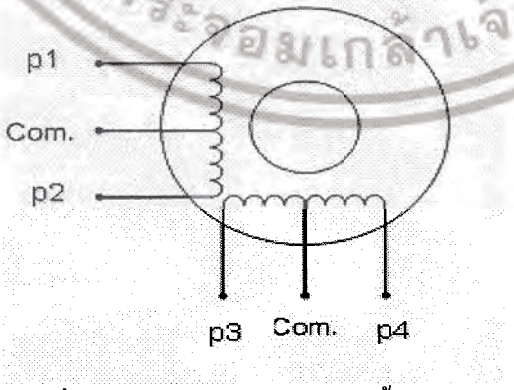
ในที่นี้ใช้ไอซีเบอร์ ULN 2003A ซึ่งมีวงจรขับ 7 ช่อง โดยแต่ละช่องสามารถจ่ายกระแสได้สูงสุด 0.5 A การควบคุมแบบ unipolar จะต้องใช้สัญญาณควบคุม 4 เส้น ซึ่งสามารถออกแบบวงจรได้ดังรูปที่ 4.19 โดยสาย common ของมอเตอร์ต่ออยู่กับแรงดัน 12V



รูปที่ 4.18 วงจรสำหรับใช้ควบคุม Stepper Motor แบบ Unipolar

4.2.1.2 การเขียนโปรแกรม

เริ่มต้นด้วยการกำหนดให้พอร์ตที่ต้องการใช้ควบคุมมอร์เตอร์จำนวน 4 บิตเป็นพอร์ตเอาท์พุท ของแต่ละประตู จากนั้นใน main loop เขียนคำสั่งควบคุมให้ on หรือ off สัญญาณของแต่ละพอร์ท ตามลำดับของขดลวด จะทำให้มอร์หมุนไปตามจำนวน step ที่เราต้องการ



รูปที่ 4.19 วงจรภายในของสเต็ปปิ้งมอเตอร์

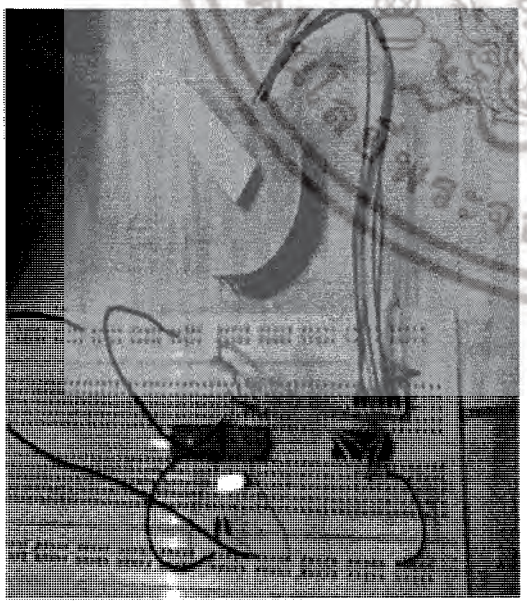
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.3 ผลการคำนวณ

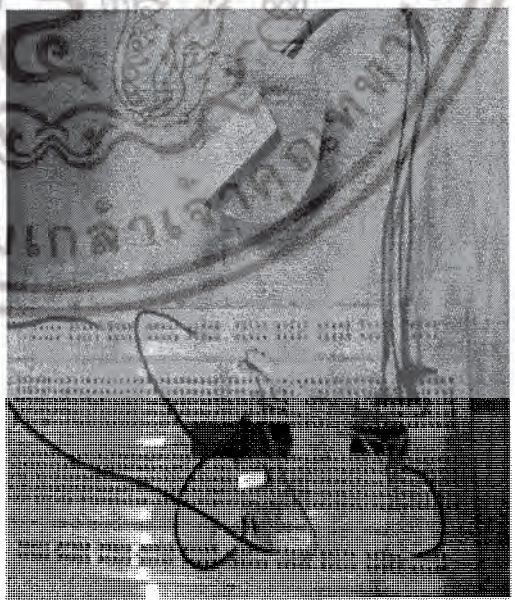
ตารางที่ 4.1 การคำนวณการทำงานระหว่างของขดลวดในสเต็ปปิ้งมอเตอร์กับมุมที่ได้

step	ขดลวดที่ทำงาน	องศา
1	P1	7.5
2	P2	15
3	P3	22.5
4	P4	30
5	P1	37.5
6	P2	45
7	P3	52.5
8	P4	60
9	P1	67.5
10	P2	75
11	P3	82.5
12	P4	90

ต่อสายสัญญาณจากมอเตอร์ให้ตรงตามคู่ สามารถต่อสลับกันระหว่างชุด P1 ,P2 ,P3 ,P4 แต่จะทำให้การหมุนกลับทิศทางกัน



รูปที่ 4.20 รูปการต่อวงจรสำหรับใช้ควบคุม Stepping Motor แบบ Unipolar Step 1



รูปที่ 4.21 รูปการต่อวงจรสำหรับใช้ควบคุม Stepping Motor แบบ Unipolar Step 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

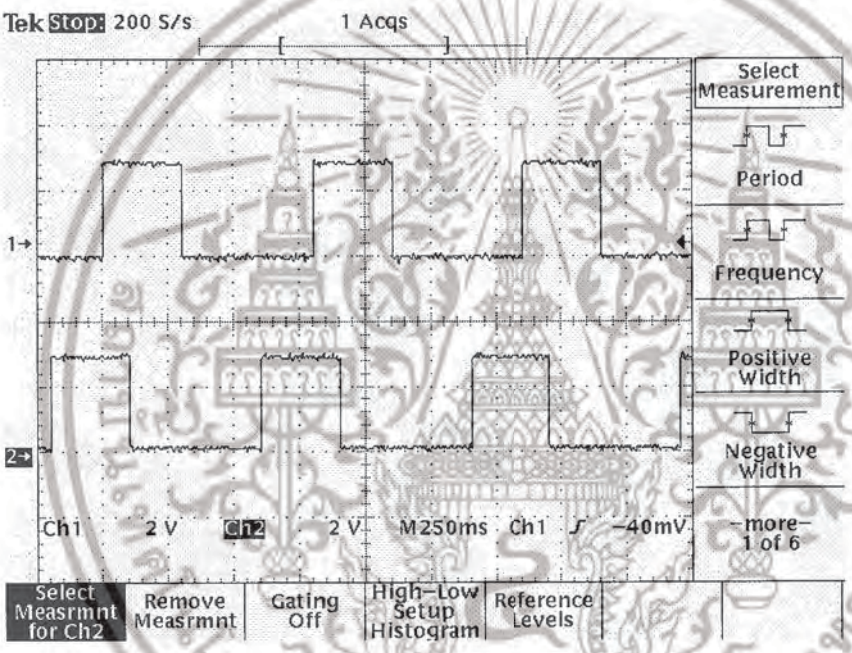
<u>ข้อควรระวัง</u> เนื่องจากมอเตอร์แต่ละตัวใช้แรงดันไฟฟ้าต่างกันขึ้นอยู่กับรุ่นมอเตอร์ที่ใช้ จึงควรตรวจสอบแรงดันไฟฟ้าให้ถูกต้องก่อนทำการจ่ายไฟ

4.2.2 วงจรจ่ายพัลส์

4.2.2.1 ขั้นตอนการทดลอง

ในการทดลองวัดสัญญาณพัลส์ จะทำการป้อนอินพุตด้วยสัญญาณไฟ DC 5 โวลต์ ให้วงจรแล้ววัดเอาท์พุตจากสัญญาณพัลส์ที่ได้จากแต่ละจุดเปรียบเทียบกัน

4.2.2.2 ผลการทดลอง



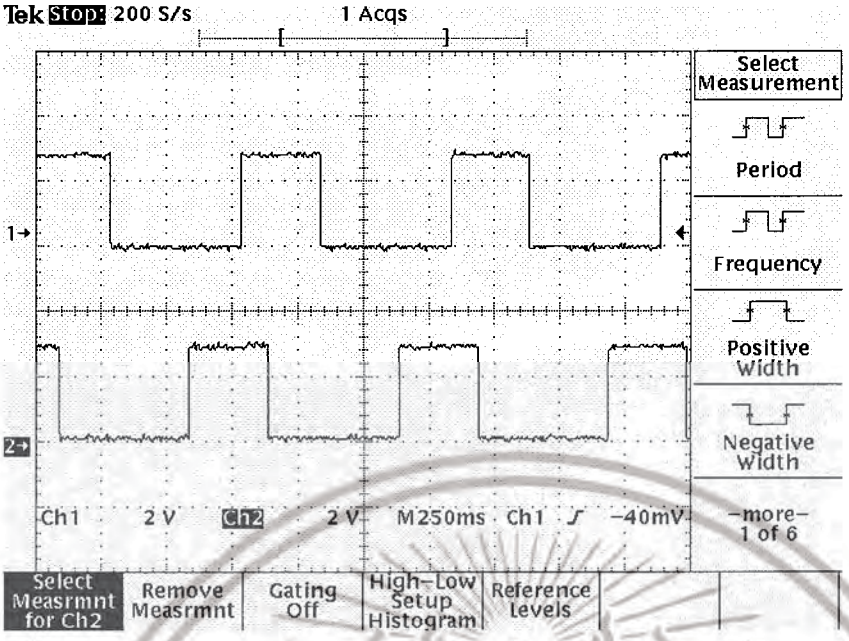
รูปที่ 4.22 การเปรียบเทียบสัญญาณพัลส์จากจุดที่ 1 กับ สัญญาณพัลส์จากจุดที่ 2

บน คือ สัญญาณพัลส์จากจุดที่ 1 voltage 2V , time/div 250 ms

ล่าง คือ สัญญาณพัลส์จากจุดที่ 2 voltage 2V , time/div 250 ms

สัญญาณพัลส์จากจุดที่ 1 กับ สัญญาณพัลส์จากจุดที่ 2 มีคาบเวลาต่างกัน เท่ากับ 200 ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

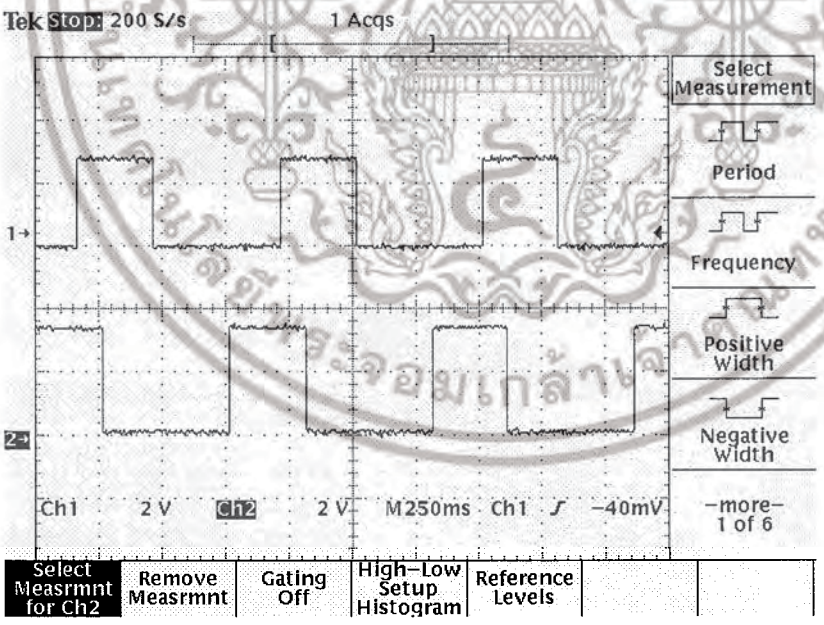


รูปที่ 4.23 การเปรียบเทียบสัญญาณพัลส์จากจุดที่ 2 กับ สัญญาณพัลส์จากจุดที่ 3

บน คือ สัญญาณพัลส์จากจุดที่ 2 voltage 2V , time/div 250 ms

ล่าง คือ สัญญาณพัลส์จากจุดที่ 3 voltage 2V , time/div 250 ms

สัญญาณพัลส์จากจุดที่ 2 กับ สัญญาณพัลส์จากจุดที่ 3 มีคาบเวลาต่างกัน เท่ากับ 200 ms



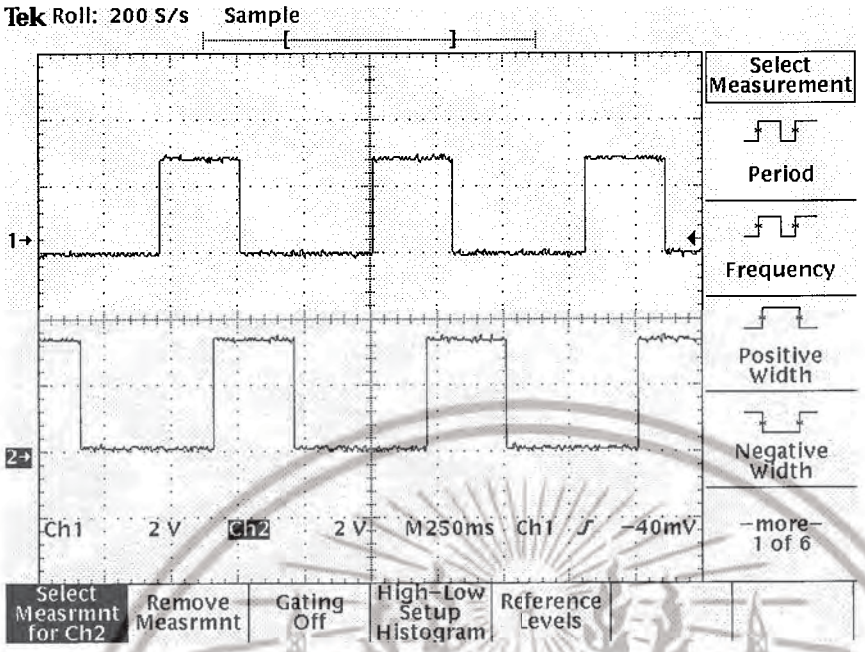
รูปที่ 4.24 การเปรียบเทียบสัญญาณพัลส์จากจุดที่ 3 กับ สัญญาณพัลส์จากจุดที่ 4

บน คือ สัญญาณพัลส์จากจุดที่ 3 voltage 2V , time/div 250 ms

ล่าง คือ สัญญาณพัลส์จากจุดที่ 4 voltage 2V , time/div 250 ms

สัญญาณพัลส์จากจุดที่ 3 กับ สัญญาณพัลส์จากจุดที่ 4 มีคาบเวลาต่างกัน เท่ากับ 200 ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



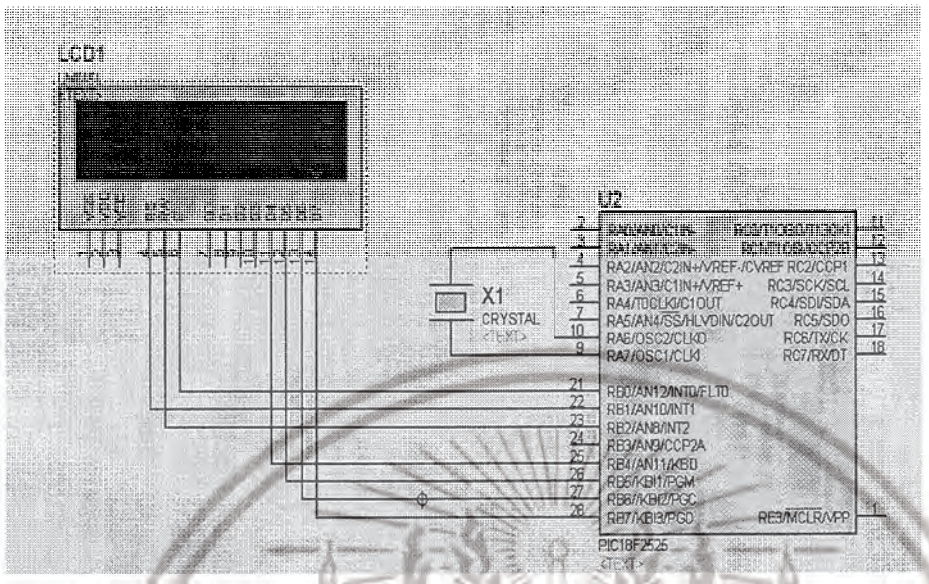
รูปที่ 4.25 การเปรียบเทียบสัญญาณพัลส์จากจุดที่ 1 กับ สัญญาณพัลส์จากจุดที่ 4
บน คือ สัญญาณพัลส์จากจุดที่ 1 voltage 2V , time/div 250 ms
ล่าง คือ สัญญาณพัลส์จากจุดที่ 4 voltage 2V , time/div 250 ms
สัญญาณพัลส์จากจุดที่ 1 กับ สัญญาณพัลส์จากจุดที่ 4 มีคาบเวลาต่างกัน เท่ากับ 200 ms

4.2.3 วงจรแสดงผลจอ LCD

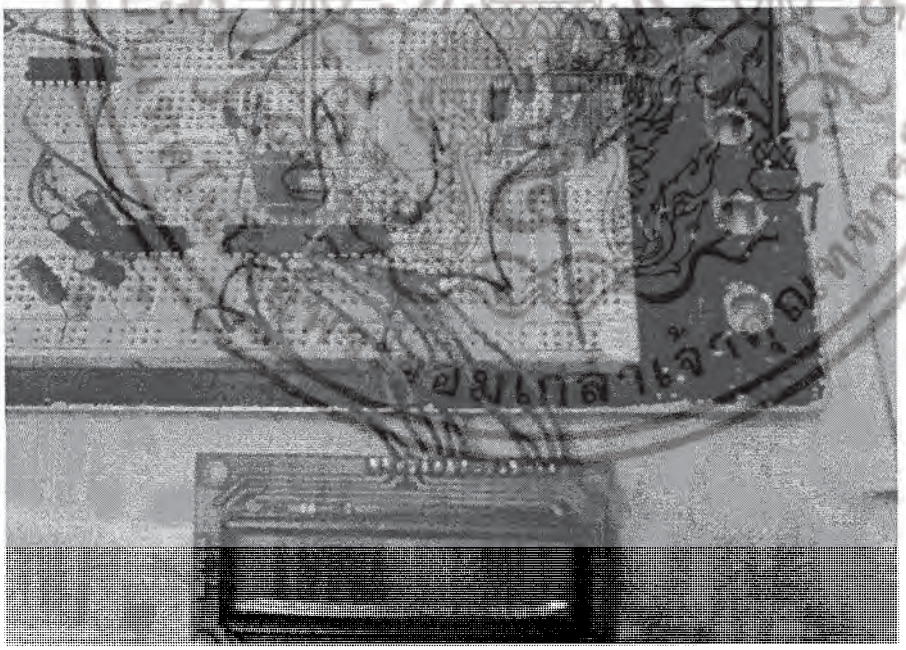
4.2.3 .1 ขั้นตอนการทดลอง

ในการส่งข้อความตัวอักษรเพื่อที่จะให้ไปแสดงผลบนมอดูล LCD นั้น ขั้นตอนแรก จะต้องเขียนชุดคำสั่งเพื่อส่งให้กับมอดูล LCD จากนั้นตามด้วยข้อมูลข้อความที่ต้องการแสดงผล ในปริญญานิพนธ์ฉบับนี้จะเขียนโปรแกรมควบคุมมอดูล LCD เพียงสี่ขาเท่านั้น คือ D4-D7 โดยโปรแกรมที่เขียนให้กับวงจรจะมีแต่การรับข้อมูลมาแสดงผลเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 วงจรสำหรับการแสดงผลที่จอ LCD



รูปที่ 4.27 การต่อวงจรสำหรับแสดงผลที่จอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3.2 ผลการทดลอง

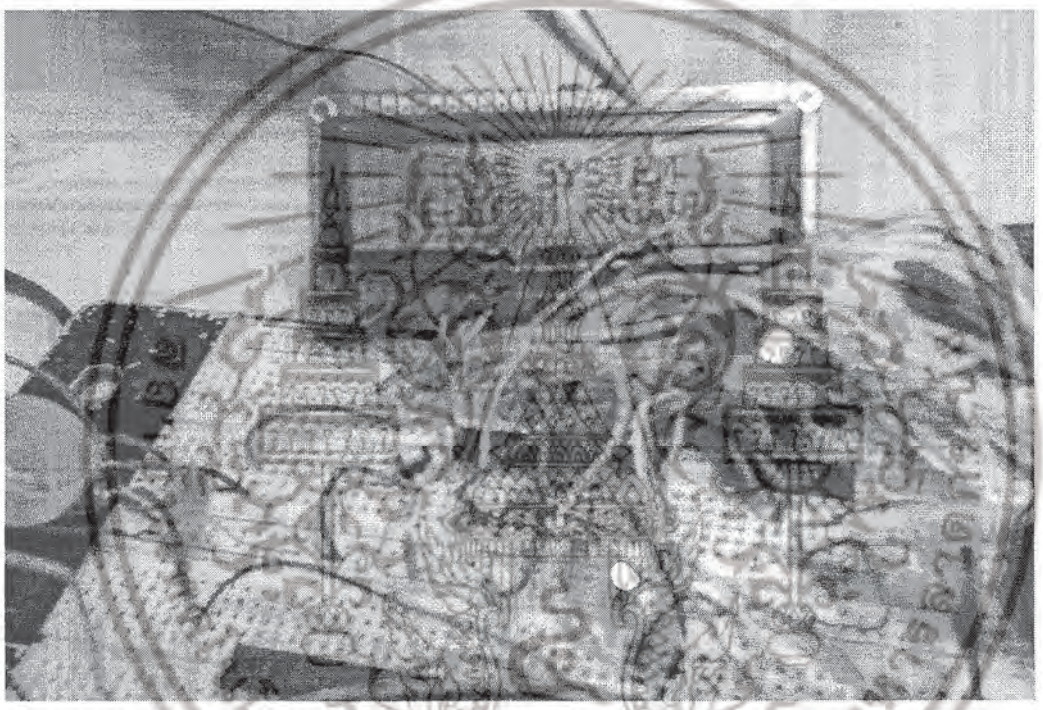
การเขียนโปรแกรมให้แสดงผลที่จอแอลซีดีมีการแสดงผลที่จอ 2 ส่วน คือ ที่ประตูทางเข้าและประตูทางออก

สำหรับประตูทางเข้า

ก่อนจะมีการเข้าใช้งาน ที่จอแอลซีดีจะมีการแสดงข้อความว่า

"BTS SYSTEM "

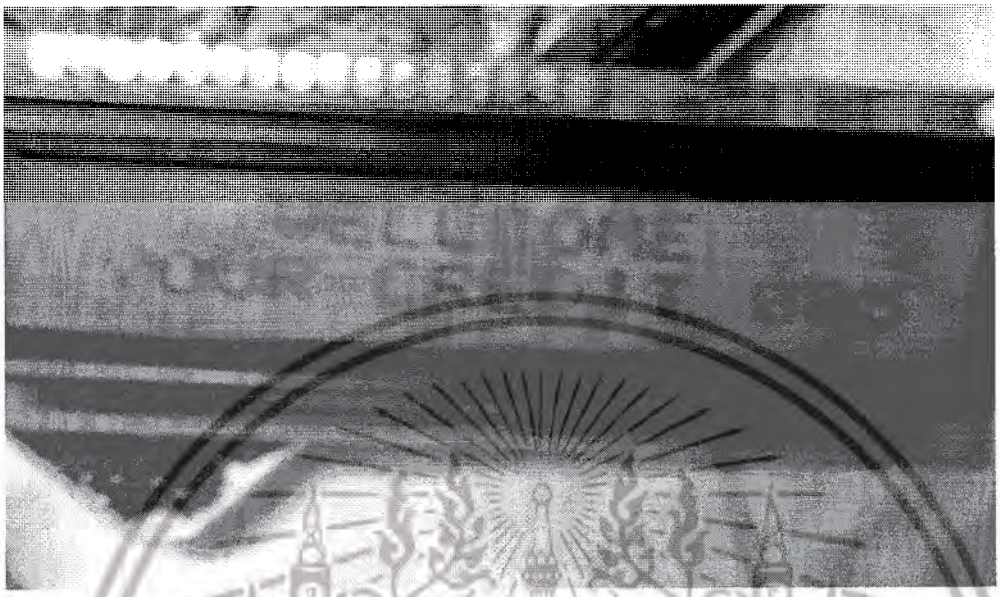
"TELECOM PROJECT"



รูปที่ 4.28 รูปการเข้าใช้บริการที่ประตูทางเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

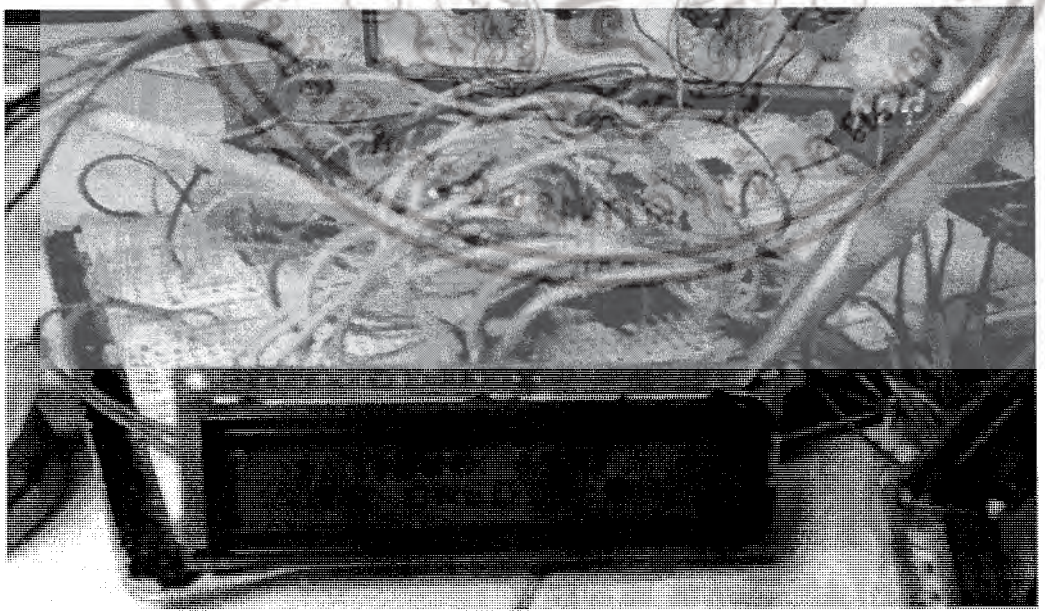
เมื่อมีการเข้าใช้บริการจะมีการแสดงผลข้อความว่า “WELLCOME ” และมีการแสดงยอดจำนวนการใช้งาน “YOUR CREDIT xx”



รูปที่ 4.29 ผลการเข้าใช้บริการที่ประตูทางเข้า

สำหรับประตูทางออก

สำหรับประตูทางออกจะมีการแสดงผลข้อความว่า “THANK YOU” และแสดงยอดคงเหลือการใช้บริการ “YOUR CREDIT xx”



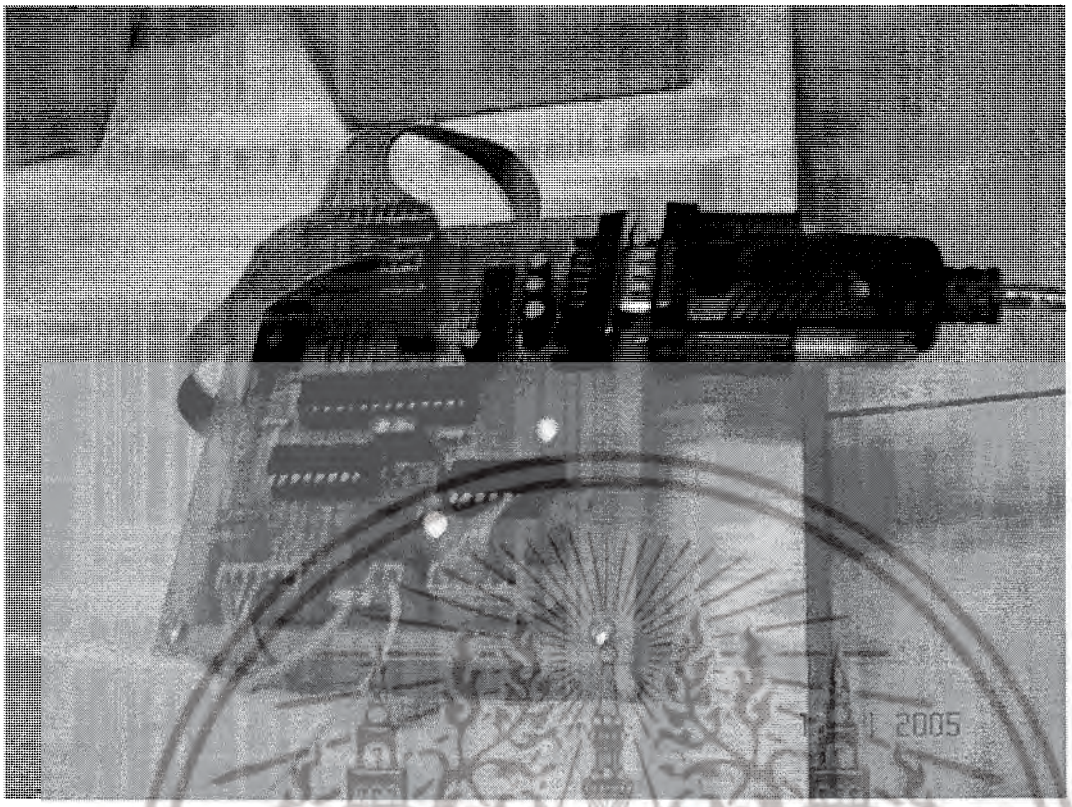
รูปที่ 4.30 ผลการออกจากการใช้บริการที่ประตูทางออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.31 สถานะเตรียมพร้อมของหน้าจอแอลซีดี ก่อนการ Tag ของ RFID

รูปที่ 4.32 สถานะการใช้งานเมื่อมีการ Tag ของ RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 วงจรการทำงานในสถานะพร้อมใช้งาน

รูปที่ 4.34 ภาพรวมแบบจำลองการทำงานในสถานะพร้อมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์

ระบบควบคุมการการผ่านเข้าออกของบริการรถไฟฟ้าผ่าน RFID มีการทำงานของระบบโดยรวมเริ่มจากการรับค่าจากบัตร RFID โดยตัวอ่านจะส่งค่าที่ได้ไปประมวลโดยคอมพิวเตอร์ได้ ทำการงานในฐานข้อมูลที่สร้างไว้ โดยใช้ C# เป็นตัวเชื่อมระหว่างคอมพิวเตอร์และส่วนของฮาร์ดแวร์ ในการส่งสัญญาณไป ที่ไมโครคอนโทรลเลอร์ซึ่งทำหน้าที่ควบคุมฮาร์ดแวร์ ซึ่งจะทำการเปิด – ปิด ประตูพร้อมทั้งแสดงจำนวนครั้งของการใช้งานผ่านหน้าจอ LCD โดยที่ในการทำงานทางทางด้านขาเข้าและขาออกจะมีการทำงานเหมือนกัน แต่แตกต่างตรงที่ทางด้านขาเข้าจะมีการหัดยอดเครดิตก่อน แล้วแสดงยอดแครดิตที่เหลือจากการหักบนจอ LCD ส่วนทางด้านขาออกจะแสดงยอดเครดิตที่เหมือนกับด้านขาเข้าอีกครั้งหนึ่ง และนอกจากนี้ผู้ใช้บริการสามารถนำรหัส RFID ไปใช้ในการทำธุรกรรมต่าง ๆ ได้ ในส่วนของโปรแกรมการแก้ไขข้อมูล และส่วนของโปรแกรมเติมยอดเครดิตได้โดยผ่านทางศูนย์บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

ศุภชัย สมพานิช. 2551. Database Programming ด้วย VB 2008 & VC# 2008. ไอดีซี. นนทบุรี

สัจจะ จรัสรุ่งรวีวร. 2550. คู่มือ Visual C# 2005 ฉบับสมบูรณ์. พิมพ์ครั้งที่ 1. นนทบุรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก code ที่ใช้ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดที่ใช้ในการทดลองส่วนของโปรแกรมกับฐานข้อมูล

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Timers;
using System.Threading;
using Itworks.RfidM1NET;
using System.Text;
using System.IO.Ports;
using System.Windows.Forms;
namespace BTStouchsim
{
  //ประกาศตัวแปร enum เพื่อกำหนดชื่อเดือนทั้ง 12 เดือน ให้ตรงกับหมายเลยตามที่กำหนด
  public enum Month
  {
    มกราคม = 1,
    กุมภาพันธ์ = 2,
    มีนาคม = 3,
    เมษายน = 4,
    พฤษภาคม = 5,
    มิถุนายน = 6,
    กรกฎาคม = 7,
    สิงหาคม = 8,
    กันยายน = 9,
    ตุลาคม = 10,
    พฤศจิกายน = 11,
    ธันวาคม = 12
  }
  public partial class Form1 : Form
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public Form1()
{
   InitializeComponent();
}

// สร้าง ตัวแปร เพื่อติดต่อกับฐานข้อมูล
SqlTransaction tr;
SqlConnection Conn;
SqlCommand com;
StringBuilder sb = new StringBuilder();
SerialPort SRport = new SerialPort("COM6", 9600, Parity.None, 8, StopBits.One);

string setNumber;

string username = "";
string password = "";
bool stateREGISTER = false;
private bool _ReadFlag = false;
private bool boolDB = false;


////////////////// ฟังก์ชัน แสดงผลข้อมูลที่ดึงจากฐานข้อมูลมาแสดงในตารางข้อมูล //////////////////
private void ShowDVG()
{
   sb.Remove(0, sb.Length);
   sb.Append("SELECT rfid,name,surname");
   sb.Append(" FROM mainDB");
   sb.Append(" ORDER BY rfid");
   string sqlshow = sb.ToString();

   SqlDataReader dr;
   DataTable dt;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    com = new SqlCommand();
    com.CommandType = CommandType.Text;
    com.CommandText = sqlshow;
    com.Connection = Conn;
    dr = com.ExecuteReader();
    if (dr.HasRows)
    {
        dt = new DataTable();
        dt.Load(dr);
        DGVreg.DataSource = dt;
        DGVedit.DataSource = dt;

    }
    else
    {
        DGVreg.DataSource = null;
        DGVedit.DataSource = null;

    }
    dr.Close();
}
//////////////////////// ตั้งค่ารูปแบบในการแสดงตารางข้อมูล ////////////////////////
private void FormatDGV()
{
    if (DGVreg.RowCount > 0)
    {
        DGVreg.Columns[0].HeaderText = "RFID";
        DGVreg.Columns[1].HeaderText = "Name";
        DGVreg.Columns[2].HeaderText = "Surname";



        DGVreg.Columns[0].Width = 100;
        DGVreg.Columns[1].Width = 100;
        DGVreg.Columns[2].Width = 100;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

    if (DGVedit.RowCount > 0)
    {
        DGVedit.Columns[0].HeaderText = "RFID";
        DGVedit.Columns[1].HeaderText = "Name";
        DGVedit.Columns[2].HeaderText = "Surname";


        DGVedit.Columns[0].Width = 100;
        DGVedit.Columns[1].Width = 100;
        DGVedit.Columns[2].Width = 100;

    }


}
////////////////////// ฟังก์ชั่นหลักที่กำหนดค่าต่างๆ เมื่อเริ่มต้นโปรแกรม //////////////////////
private void Form1_Load(object sender, EventArgs e)
{
    getconnect();
    functionlockTAB();
    suggestion();


    ShowDVG();
    FormatDGV();


    DataFunctionLock(true);
    functionLockOBS(true);
    functionLockMainpage(true);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

//////////////////////////////คำแนะนำการใช้งานในหน้าเติมเงิน ////////////////////////
private void suggestion()
{
   LBREG.Items.Add("-ข้อแนะนำในการใช้งานส่วนการลงทะเบียน");
   LBREG.Items.Add("กรอกรหัส rfid ที่จะลงทะเบียนก่อน");
   LBREG.Items.Add("แล้วกดปุ่มลงทะเบียน");

   LBEDIT.Items.Add("-ข้อแนะนำในการใช้งานส่วนการแก้ไขข้อมูล");
   LBEDIT.Items.Add("นำการ์ด rfid มาทาบที่เครื่องอ่าน ");
   LBEDIT.Items.Add("พร้อมกับการกดปุ่มแก้ไขข้อมูล");


   LBTOPUP.Items.Add("เมื่อเติมเงิน 250 บาท  ได้รับยอดการใช้งานเพิ่ม  15 ครั้ง ");
   LBTOPUP.Items.Add("\n");
   LBTOPUP.Items.Add("เมื่อเติมเงิน 400 บาท  ได้รับยอดการใช้งานเพิ่ม  30 ครั้ง ");
   LBTOPUP.Items.Add("\n");
   LBTOPUP.Items.Add("เมื่อเติมเงิน 500 บาท  ได้รับยอดการใช้งานเพิ่ม  40 ครั้ง ");

}

////////////////////// ฟังก์ชั่นล็อกปุ่ม ////////////////////////
private void DataFunctionLock(bool statusData)
{
   functionLockRegister(statusData);
   functionLockEDIT(statusData);
   functionLockTOPUP(statusData);
}
////////////////////// ฟังก์ชั่นล็อกปุ่ม Register//////////////////////
private void functionLockRegister(bool stateREG)
{
   TBrfidREG.Enabled = stateREG;
   TBnameREG.Enabled = !stateREG;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TBsurREG.Enabled = !stateREG;
    TBtelREG.Enabled = !stateREG;
    CBmoneyREG.Enabled = !stateREG;

    BTstartREG.Enabled = stateREG;
    BTendREG.Enabled = !stateREG;

}
////////////////// ฟังก์ชั่นล็อกปุ่ม EDIT //////////////////
private void functionLockEDIT(bool statusEDIT)
{
    TBrfidEDIT.Enabled = statusEDIT;
    TBnameEDIT.Enabled = !statusEDIT;
    TBsurEDIT.Enabled = !statusEDIT;
    TBtelEDIT.Enabled = !statusEDIT;
    TBmoneyEDIT.Enabled = !statusEDIT;

    BTstartEDIT.Enabled = statusEDIT;
    BTendEDIT.Enabled = !statusEDIT;

}
////////////////// ฟังก์ชั่นล็อกปุ่ม TOPUP //////////////////
private void functionLockTOPUP(bool statusTOPUP)
{
    TBrfidTOPUP.Enabled = statusTOPUP;
    TBnameTOPUP.ReadOnly = true;
    TBsurTOPUP.ReadOnly = true;
    TBtelTOPUP.ReadOnly = true;
    //CBmoneyTOPUP.Enabled = !statusTOPUP;

    //BTstartTOPUP.Enabled = statusTOPUP;
    //BTendTOPUP.Enabled = !statusTOPUP;
}
////////////////// ฟังก์ชั่นล็อกปุ่ม OBSERVE//////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void functionLockOBS(bool statusOBS)
{
   BTstartOBS.Enabled = statusOBS;
   BTendOBS.Enabled = !statusOBS;
}
////////////////// ฟังก์ชั่นล็อกปุ่ม หน้าแรก ////////////////////
private void functionLockMainpage(bool statusMAIN)
{
   BTloginData.Enabled = statusMAIN;
   BTloginObs.Enabled = statusMAIN;
   TBusername.Clear();
   TBpassword.Clear();

   BTlogout.Visible = !statusMAIN;
}
////////////////// ฟังก์ชั่นล็อกแทป หน้าแรก ////////////////////
private void functionlockTAB()
{
   MyTab.TabPages.Clear();
   MyTab.TabPages.Add(TABauth);
}
////////////////////////////////////////////////

//////////////////เหตุการณ์บนปุ่ม LOGOUT//////////////////
private void BTlogout_Click(object sender, EventArgs e)
{
   functionlockTAB();
   functionLockMainpage(true);
}
//////////////////เหตุการณ์บนปุ่ม LOGIN ไปหน้า OBSERVE ////////////////////
private void BTloginObs_Click(object sender, EventArgs e)
{
   if (TBusername.Text == "" && TBpassword.Text == "")
   {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MessageBox.Show("กรุณาใส่ username และ password ก่อน", "ข้อผิดพลาด",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            getpassword();
            if (TBusername.Text == username && TBpassword.Text == password)
            {


                MyTab.TabPages.Add(TABsystem);
                MyTab.TabPages.Add(LOG);

                functionLockMainpage(false);
            }
            else
            {
                MessageBox.Show("กรุณาใส่ username และ password ให้ถูกต้อง !!!", "",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
        }
    }
    ////////////////////เหตุการณ์บนปุ่ม LOGIN ไปหน้า จัดการข้อมูล ////////////////////
    private void BTloginData_Click(object sender, EventArgs e)
    {
        if (TBusername.Text == "" && TBpassword.Text == "")
        {
            MessageBox.Show("กรุณาใส่ username และ password ก่อน", "ข้อผิดพลาด",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        getpassword();
        if (TBusername.Text == username && TBpassword.Text == password)
        {
          MyTab.TabPages.Add(TABregister);
          MyTab.TabPages.Add(TABedit);
          MyTab.TabPages.Add(TABtopup);
          functionLockMainpage(false);
        }
        else
        {
          MessageBox.Show("กรุณาใส่ username และ password ให้ถูกต้อง !!!", "",
MessageBoxButtons.OK, MessageBoxIcon.Error);
          return;
        }
      }
    }
    ////////////////////เหตุการณ์บนปุ่ม ลงทะเบียน เพื่อเริ่มต้นการลงทะเบียน ////////////////
    private void BTstartREG_Click(object sender, EventArgs e)
    {
      functionStartREG();
    }
    ////////////////////เหตุการณ์บนปุ่ม ลงทะเบียน เพื่อเริ่มต้นการลงทะเบียน ////////////////
    private void BTendREG_Click(object sender, EventArgs e)
    {
      functionEndREG();
      if(!stateREGISTER)
      {
        functionLockRegister(true);
      }

    }
    ////////////////////เหตุการณ์บนปุ่ม แก้ไขข้อมูล เพื่อเริ่มต้นการแก้ไขข้อมูล ////////////////
    private void BTstartEDIT_Click(object sender, EventArgs e)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    functionStartEdit();
}
//////////////////////เหตุการณ์บนปุ่ม แก้ไขข้อมูล เพื่อจบการแก้ไขข้อมูล ////////////////////
private void BTendEDIT_Click(object sender, EventArgs e)
{
    functionEndEDIT();
}
//////////////////////เหตุการณ์บนปุ่ม แก้ไขข้อมูล เพื่อเริ่มต้นการเติมเงิน ////////////////////
private void BTstartTOPUP_Click(object sender, EventArgs e)
{
    TBmoneyNEW.Clear();
    functionStartTOPUP();

}
//////////////////////เหตุการณ์บนปุ่ม แก้ไขข้อมูล เพื่อเริ่มต้นการทำงานเปิดปิดประตู ////////////////////
private void BTstartOBS_Click(object sender, EventArgs e)
{
    BTendOBS.Enabled = true;
    BTstartOBS.Enabled = false;

    MyTab.TabPages.Clear();
    MyTab.TabPages.Add(TABsystem);
    MyTab.TabPages.Add(LOG);


    while (BTstartOBS.Enabled == false)
    {
        getTag();
    }
}
//////////////////////////////////////////////////////

//////////////////////////// เริ่มเชื่อมต่อฐานข้อมูล ////////////////////////////////
private void getconnect()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    string strConn = "";
    strConn = "Data Source=.\\SQLExpress;Initial Catalog=sqlTest;Integrated Security=True";
    Conn = new SqlConnection();

    if (Conn.State == ConnectionState.Open)
    {
        Conn.Close();
    }
    Conn.ConnectionString = strConn;
    Conn.Open();
}
//////////////////////////// ตรวจสอบ password ////////////////////////////
private void getpassword()
{
    try
    {
        StringBuilder sb = new StringBuilder();
        sb.Remove(0, sb.Length);
        sb.Append("SELECT username,password");
        sb.Append(" FROM getpass");
        string sqlpass = sb.ToString();

        SqlCommand com = new SqlCommand();
        SqlDataReader dr;
        com.CommandType = CommandType.Text;
        com.CommandText = sqlpass;
        com.Connection = Conn;
        com.Parameters.Clear();
        com.Parameters.Add("username", SqlDbType.NVarChar).Value = TBusername.Text.Trim();
        dr = com.ExecuteReader();
        if (dr.HasRows)
        {
            dr.Read();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            username = dr.GetString(dr.GetOrdinal("username"));
            password = dr.GetString(dr.GetOrdinal("password"));

        }
        else
        {
            MessageBox.Show("ไม่แสดงผล", "ผลการค้นหา", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            TBusername.Focus();
            TBusername.SelectAll();
        }
        dr.Close();
      }
      catch
      {
        MessageBox.Show("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูลได้", "ผลการค้นหา",
MessageBoxButtons.OK, MessageBoxIcon.Error);
      }

    }

    ////////////////////// ฟังก์ชั่นเริ่มต้นการทำการ ลงทะเบียนข้อมูล ////////////////////////////
    private void functionStartREG()
    {
      if (TBrfidREG.Text == "")
      {
        LBREG.Items.Clear();
        LBREG.Items.Add("ไม่สามารถอ่านรหัสบัตร rfid ได้");

      }
      else
      {
        LBREG.Items.Clear();
        CheckRfidRegister();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
}
////////// ฟังก์ชั่นตรวจสอบรหัส rfid ที่นำมาลงทะเบียน ว่าซ้ำกับข้อมูลเดิมหรือไม่ //////////////////
private void CheckRfidRegister()
{
  try
  {
    StringBuilder sb = new StringBuilder();
    sb.Remove(0, sb.Length);
    sb.Append("SELECT rfid");
    sb.Append(" FROM mainDB");
    sb.Append(" WHERE (rfid=@rfid)");
    string sqlpass = sb.ToString();

    SqlCommand com = new SqlCommand();
    SqlDataReader dr;
    com.CommandType = CommandType.Text;
    com.CommandText = sqlpass;
    com.Connection = Conn;
    com.Parameters.Clear();
    com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidREG.Text.Trim();
    dr = com.ExecuteReader();
    if (dr.HasRows)
    {
      LBREG.Items.Add("รหัส RFID ซ้ำ กรุณาใส่รหัส RFID ใหม่");
    }
    else
    {
      LBREG.Items.Add("ท่านสามารถใช้รหัส RFID นี้ได้ ในการลงทะเบียน");
      functionLockRegister(false);
      stateREGISTER = true;
    }
    dr.Close();
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    catch
    {
        LBREG.Items.Add("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูลได้");
    }

}
////////////////////// ฟังก์ชั่นลงทะเบียนข้อมูลในฐานข้อมูลและการ์ด rfid ////////////////////////////////
private void functionEndREG()
{
    if (TBnameREG.Text.Trim() == "")
    {
        LBREG.Items.Clear();
        LBREG.Items.Add("กรุณาป้อนชื่อ");
        TBnameREG.Focus();
        return;
    }

    if (TBsurREG.Text.Trim() == "")
    {
        LBREG.Items.Clear();
        LBREG.Items.Add("กรุณาป้อนนามสกุล");
        TBsurREG.Focus();
        return;
    }

    if (TBtelREG.Text.Trim() == "")
    {
        LBREG.Items.Clear();
        LBREG.Items.Add("กรุณาป้อนเบอร์โทรศัพท์");
        TBtelREG.Focus();
        return;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (CBmoneyREG.Text == "")
{
   LBREG.Items.Clear();
   LBREG.Items.Add("กรุณาเลือกการเติมเงินเริ่มต้นการใช้งาน");
   CBmoneyREG.Focus();
   return;
}

if (MessageBox.Show("คุณต้องการเพิ่มข้อมูลใหม่ใช่หรือไม่?", "คำยืนยัน",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
System.Windows.Forms.DialogResult.Yes)
{
   tr = Conn.BeginTransaction();
   try
   {
      sb.Remove(0, sb.Length);
      sb.Append("INSERT INTO mainDB ( rfid,name,surname,phone,number,status,date)");
      sb.Append(" VALUES (@rfid, @name, @surname, @phone ,@number,@status,@date)");
      string sqlAdd;
      sqlAdd = sb.ToString();

      com.CommandText = sqlAdd;
      com.CommandType = CommandType.Text;
      com.Connection = Conn;
      com.Transaction = tr;
      com.Parameters.Clear();
      com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidREG.Text.Trim();
      com.Parameters.Add("@name", SqlDbType.NVarChar).Value = TBnameREG.Text.Trim();
      com.Parameters.Add("@surname", SqlDbType.NVarChar).Value =
TBsurREG.Text.Trim();
      com.Parameters.Add("@phone", SqlDbType.NVarChar).Value = TBtelREG.Text.Trim();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            com.Parameters.Add("@number", SqlDbType.NVarChar).Value =
CBmoneyREG.SelectedItem.ToString();
            com.Parameters.Add("@status", SqlDbType.NChar).Value = "off";
            com.Parameters.Add("@date", SqlDbType.NChar).Value = ExpireDate() ;
            com.ExecuteNonQuery();

            tr.Commit();

            SetDataToRFID(TBrfidREG.Text.Trim(), CBmoneyREG.SelectedItem.ToString());
            LBREG.Items.Clear();
            LBREG.Items.Add("เพิ่มข้อมูลเรียบร้อยแล้ว เรียบร้อยแล้ว");

            TBrfidREG.Clear();
            TBnameREG.Clear();
            TBsurREG.Clear();
            TBtelREG.Clear();
            CBmoneyREG.Text = "";

            stateREGISTER = false;

            ShowDVG();

            FormatDGV();
        }
        catch
        {
            MessageBox.Show("เกิดข้อผิดพลาดระหว่างอัพเดตข้อมูล", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            tr.Rollback();
        }
    }
    TBnameREG.Focus();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
////////////////////////// ฟังก์ชั่น  เขียนข้อมูลลงในการ์ด rfid ////////////////////////////////
private void SetDataToRFID(string setName, string setNumber2)
{
   ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;
   int device_address = 0;
   byte[] atq = new byte[65];

   rfidm1_result = ISO14443A.MF_Request(device_address, RequestModeTypes.RequestIdle,
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;
   };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
byte[] key = new byte[6] { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };

rfidm1_result = Mifare.MF_LoadKey(device_address, key);
if (ResultCodeTypes.CommandOK != rfidm1_result)
{
  return;
};

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);

  return;
};

Thread.Sleep(2);

byte[] temp = System.Text.Encoding.Default.GetBytes(setName);
byte[] buffer = new byte[16];
for (int i = 0; i < 16; ++i)
{
  if (i < temp.Length)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          buffer[i] = temp[i];
        }
        else
        {
          buffer[i] = 0;
        }
      };

      if (ResultCodeTypes.CommandOK == Mifare.MF_Write(device_address, (byte)((sector_number
* 4) + block_number), 1, buffer))
      {

        Thread.Sleep(1);

      }

      block_number = 2;
      temp = System.Text.Encoding.Default.GetBytes(setNumber2);
      for (int i = 0; i < 16; ++i)
      {
        if (i < temp.Length)
        {
          buffer[i] = temp[i];
        }
        else
        {
          buffer[i] = 0;
        }
      };

      if (ResultCodeTypes.CommandOK == Mifare.MF_Write(device_address, (byte)((sector_number
* 4) + block_number), 1, buffer))
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        this.Beep1(device_address);

        Application.DoEvents();

        Thread.Sleep(3);

        ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);

        Thread.Sleep(1);

    }
}
////////////////////////// ฟังก์ชั่น เพิ่มเดือน ในขั้นตอนเติมเงิน //////////////////////////////
private string ExpireDate()
{
    string output, newmonth;

    int outputint;
    string[] currentdate = DateTime.Now.ToLongDateString().Split(' ');
    string d = currentdate[0];
    string m = currentdate[1];
    string y = currentdate[2];
    Month current_month_enum = (Month)Enum.Parse(typeof(Month), m, false);

    if ((int)current_month_enum == 12)
    {
        int newyear = int.Parse(y);
        newyear++;
        y = newyear.ToString();
    }

    outputint = (int)current_month_enum + 1;
    outputint = outputint % 12;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Month newm = (Month)outputint;
    newmonth = newm.ToString();
    output = string.Format("{0} {1} {2}", d, newmonth, y);


    return output;

}
////////////////////////////////////////////////

//////////////////////////// เสียง เครื่อง rfid ทำงานปกติ ////////////////////////
private void Beep1(int deviceAddress)
{
    byte[] sound = new byte[5] { 0x01, 0x01, 0x00, 0x00, 0x01 };
    ConfigurationAndIO.ActiveBuzzer(deviceAddress, BuzzerModeTypes.SoundPattern, sound);
}
//////////////////////////// เสียง เครื่อง rfid ทำงานไม่ปกติ ////////////////////////
private void Beep2(int deviceAddress)
{
    byte[] sound = new byte[5] { 0x01, 0x01, 0x01, 0x01, 0x01 };
    ConfigurationAndIO.ActiveBuzzer(deviceAddress, BuzzerModeTypes.SoundPattern, sound);
}
////////////////////////////////////

//////////////////////////// ฟังก์ชั่น แก้ไขข้อมูล ////////////////////////////
private void functionStartEdit()
{
    GetTagRFID("edit");
    if (TBrfidEDIT.Text == "")
    {
        LBEDIT.Items.Add("เกิดปัญหาในการอ่านข้อมูลในการ์ด rfid กรุณาลองใหม่อีกครั้ง");
    }
    else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        StartEditData();
    }
}
////////////////////  ฟังก์ชั่น รับค่าจากการ์ด rfid  ////////////////////////////
private void GetTagRFID(string gettag)
{

    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)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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;
        };

        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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);

            Thread.Sleep(3);

            _ReadFlag = false;

            continue;
        };

        Thread.Sleep(2);

        byte[] buffer = 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, buffer))
        {

          if (gettag == "edit")
          {
            TBrfidEDIT.Text = READTAG(buffer);
            _ReadFlag = false;
          }
          else if (gettag == "topup")
          {
            TBrfidTOPUP.Text = READTAG(buffer);
            _ReadFlag = false;
          }
          else if(gettag == "edittopup")
          {
            TBedittopupRFID.Text = READTAG(buffer);
            _ReadFlag = false;
          }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
        {
           LBEDIT.Items.Add(READTAG(buffer));
           LBTOPUP.Items.Add(READTAG(buffer));
           _ReadFlag = false;
        }


        this.Beep1(device_address);

        Application.DoEvents();

        Thread.Sleep(3);

        ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);

        Thread.Sleep(3);


        ////////////////////////

     };

     Application.DoEvents();
  };
}
////////////////////  ฟังก์ชั่น เปลี่ยนข้อมูล จาก byte เป็น string ////////////////////////
private string READTAG(byte[] buffer)
{
   byte[] B = new byte[8];
   for (int i = 0; i < 8; i++)
   {
      B[i] = buffer[i];
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    string s = System.Text.ASCIIEncoding.ASCII.GetString(B);
    return s;
}
////////////////// ฟังก์ชั่น เปลี่ยนข้อมูลที่ถูกแก้ไขใน ขั้นตอนการแก้ไขข้อมูล //////////////////////
private void StartEditData()
{

    try
    {
        StringBuilder sb = new StringBuilder();
        sb.Remove(0, sb.Length);
        sb.Append("SELECT rfid,name,surname,phone,number");
        sb.Append(" FROM mainDB");
        sb.Append(" WHERE (rfid=@rfid)");
        string sqlpass = sb.ToString();

        SqlCommand com = new SqlCommand();
        SqlDataReader dr;
        com.CommandType = CommandType.Text;
        com.CommandText = sqlpass;
        com.Connection = Conn;
        com.Parameters.Clear();
        com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidEDIT.Text.Trim();
        dr = com.ExecuteReader();
        if (dr.HasRows)
        {
            dr.Read();
            TBnameEDIT.Text = dr.GetString(dr.GetOrdinal("name"));
            TBsurEDIT.Text = dr.GetString(dr.GetOrdinal("surname"));
            TBtelEDIT.Text = dr.GetString(dr.GetOrdinal("phone"));
            TBmoneyEDIT.Text = dr.GetString(dr.GetOrdinal("number"));

            DataFunctionLock(false);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            else
            {
                LBEDIT.Items.Add("ไม่มี RFID รหัสนี้ในฐานข้อมูล");
                TBrfidEDIT.Clear();
                TBrfidEDIT.Focus();
            }
            dr.Close();
        }
        catch
        {
            MessageBox.Show("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูลได้", "ผลการค้นหา",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

    }
    ////////////////////////// ฟังก์ชั่น จบการทำงานขั้นตอนแก้ไขข้อมูล //////////////////////////
    private void functionEndEDIT()
    {
        if (TBrfidEDIT.Text.Trim() == "")
        {
            LBEDIT.Items.Clear();
            LBEDIT.Items.Add("กรุณาป้อน RFID");
            TBrfidEDIT.Focus();
            return;
        }

        if (TBnameEDIT.Text.Trim() == "")
        {
            LBEDIT.Items.Clear();
            LBEDIT.Items.Add("กรุณาป้อนชื่อ");
            TBnameEDIT.Focus();
            return;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (TBsurEDIT.Text.Trim() == "")
{
   LBEDIT.Items.Clear();
   LBEDIT.Items.Add("กรุณาป้อนนามสกุล");
   TBsurEDIT.Focus();
   return;
}

if (TBtelEDIT.Text.Trim() == "")
{
   LBEDIT.Items.Clear();
   LBEDIT.Items.Add("กรุณาป้อนเบอร์โทรศัพท์");
   TBtelEDIT.Focus();
   return;
}

if (MessageBox.Show("คุณต้องการแก้ไขข้อมูล ใช่หรือไม่?", "คำยืนยัน",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
System.Windows.Forms.DialogResult.Yes)
{
   tr = Conn.BeginTransaction();
   try
   {
      sb.Remove(0, sb.Length);
      sb.Append("UPDATE mainDB");
      sb.Append(" SET name=@name,");
      sb.Append("surname=@surname,");
      sb.Append("phone=@phone");
      sb.Append(" WHERE (rfid=@rfid)");
      string sqlEdit;
      sqlEdit = sb.ToString();

      com.CommandText = sqlEdit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            com.CommandType = CommandType.Text;
            com.Connection = Conn;
            com.Transaction = tr;
            com.Parameters.Clear();
            com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidEDIT.Text.Trim();
            com.Parameters.Add("@name", SqlDbType.NVarChar).Value =
TBnameEDIT.Text.Trim();
            com.Parameters.Add("@surname", SqlDbType.NVarChar).Value =
TBsurEDIT.Text.Trim();
            com.Parameters.Add("@phone", SqlDbType.NVarChar).Value = TBtelEDIT.Text.Trim();



            int result;
            result = com.ExecuteNonQuery();
            if (result == 0)
            {
               tr.Rollback();
               LBEDIT.Items.Clear();
               LBEDIT.Items.Add("การอัพเดตฐานข้อมูลผิดพลาด");
               TBrfidEDIT.SelectAll();
            }
            else
            {
               tr.Commit();
               LBEDIT.Items.Clear();
               LBEDIT.Items.Add("แก้ไขชื่อลูกค้า เรียบร้อยแล้ว");


               TBrfidEDIT.Clear();
               TBnameEDIT.Clear();
               TBsurEDIT.Clear();
               TBtelEDIT.Clear();
               TBmoneyEDIT.Clear();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ShowDVG();
            FormatDGV();

        }
    }
    catch (Exception ex)
    {
        tr.Rollback();
        LBEDIT.Items.Clear();
        LBEDIT.Items.Add("เกิดข้อผิดพลาด เนื่องจาก " + ex.Message);

    }
  }
  DataFunctionLock(true);


}
////////////////////////////////////////////////

bool clickTopup = false;
bool clickCb = true;
bool _WriteFlagNUMBER = false;
string datefortopup;

//////////////////////////////// ฟังก์ชั่นเริ่มต้นการเติมเงิน ////////////////////////////////
private void functionStartTOPUP()
{
  GetTagRFID("topup");
  if (TBrfidTOPUP.Text == "")
  {

    LBTOPUP.Items.Clear();
    LBTOPUP.Items.Add("กรุณาทาบบัตร rfid แล้วกดปุ่มเติมยอดเงินอีกครั้ง");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        TBrfidTOPUP.Focus();

    }
    else
    {

        GetDataTopUp();
        if (clickTopup)
        {
            DataFunctionLock(true);
        }
    }
}
////////////////// ฟังก์ชั่นเชื่อมต่อฐานข้อมูล เพื่อดึงข้อมูลมาใช้ในการเติมเงิน ////////////////////
private void GetDataTopUp()
{
    try
    {

        StringBuilder sb = new StringBuilder();
        sb.Remove(0, sb.Length);
        sb.Append("SELECT rfid,name,surname,phone,number,date");
        sb.Append(" FROM mainDB");
        sb.Append(" WHERE (rfid=@rfid)");
        string sqlpass = sb.ToString();

        SqlCommand com = new SqlCommand();
        SqlDataReader dr;
        com.CommandType = CommandType.Text;
        com.CommandText = sqlpass;
        com.Connection = Conn;
        com.Parameters.Clear();
        com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidTOPUP.Text.Trim();
        dr = com.ExecuteReader();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (dr.HasRows)
    {
      dr.Read();
      TBnameTOPUP.Text = dr.GetString(dr.GetOrdinal("name"));
      TBsurTOPUP.Text = dr.GetString(dr.GetOrdinal("surname"));
      TBtelTOPUP.Text = dr.GetString(dr.GetOrdinal("phone"));
      TBmoneyOLD.Text = dr.GetString(dr.GetOrdinal("number"));
      setNumber = dr.GetString(dr.GetOrdinal("number"));
      datefortopup = dr.GetString(dr.GetOrdinal("date"));

      functionLockTOPUP(false);
      clickTopup = false; /////
      LBTOPUP.Items.Clear();
      LBTOPUP.Items.Add("กรุณาเลือกจำนวนเงินที่จะเติม");

    }
    else
    {
      LBTOPUP.Items.Clear();
      LBTOPUP.Items.Add("เกิดปัญหาในการเชื่อมต่อฐานข้อมูล");
      TBrfidTOPUP.Clear();
      TBrfidTOPUP.Focus();
      clickTopup = false;
      return;
    }
    dr.Close();
  }
  catch
  {
    LBTOPUP.Items.Clear();
    LBTOPUP.Items.Add("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูลได้");
  }
}
```

////////////////////// ฟังชั่นตรวจสอบจำนวนเงินที่เติม(เก่า) //////////////////////

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void functionEndTOPUP()
{
  if (clickCb)
  {
    setRFID(setNumber);
    if (_WriteFlagNUMBER)
    {
      functionTopup();
      TBrfidTOPUP.Enabled = true;
      TBmoneyOLD.Clear();
    }
    else
    {
      LBTOPUP.Items.Clear();
      LBTOPUP.Items.Add("เกิดข้อผิดพลาดในการเติมเงิน ขั้นตอน เขียนจำนวนเงินในบัตร");
    }

  }
  else
    return;
}
////////////////////////// ฟังก์ชัน เพิ่มเดือน ในขั้นตอนการเติมเงิน //////////////////////////
private string UpdateExpireDate(string OldExpireDate)
{
  string[] dateSplit = OldExpireDate.Split(' ');
  string d = dateSplit[0];
  string m = dateSplit[1];
  string y = dateSplit[2];

  Month month_enum = (Month)Enum.Parse(typeof(Month), m, false);
  if ((int)month_enum == 12)
  {
    int newyear = int.Parse(y);
    newyear++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            y = newyear.ToString();
        }
        int outputint = ((int)month_enum + 1) % 12;
        Month newm = (Month)outputint;
        string output = string.Format("{0} {1} {2}", d, newm.ToString(), y);
        return output;
    }
    ////////////////////////// ฟังก์ชั่นการเติมเงิน //////////////////////////
    private void functionTopup()
    {
        if (MessageBox.Show("คุณต้องการเพิ่มจำนวนเงิน ใช่หรือไม่?", "คำยืนยัน",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
System.Windows.Forms.DialogResult.Yes)
        {
            tr = Conn.BeginTransaction();
            try
            {
                sb.Remove(0, sb.Length);
                sb.Append("UPDATE mainDB");
                sb.Append(" SET number=@number,date=@date");
                sb.Append(" WHERE (rfid=@rfid)");
                string sqltop;
                sqltop = sb.ToString();

                com.CommandText = sqltop;
                com.CommandType = CommandType.Text;
                com.Connection = Conn;
                com.Transaction = tr;
                com.Parameters.Clear();
                com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidTOPUP.Text.Trim();
                com.Parameters.Add("@number", SqlDbType.NVarChar).Value = setNumber;
                com.Parameters.Add("@date", SqlDbType.NChar).Value =
UpdateExpireDate(datefortopup);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            int result;
            result = com.ExecuteNonQuery();
            if (result == 0)
            {
               tr.Rollback();
               LBTOPUP.Items.Clear();
               LBTOPUP.Items.Add("ไม่สามารถเติมเงินได้ เพราะขั้นตอนการส่งข้อมูลไปที่ฐานข้อมูล
มีปัญหา ");
            }
            else
            {
               tr.Commit();
               LBTOPUP.Items.Clear();
               LBTOPUP.Items.Add("เพิ่มจำนวนเงินเรียบร้อยแล้ว ");

               TBrfidTOPUP.Clear();
               TBnameTOPUP.Clear();
               TBsurTOPUP.Clear();
               TBtelTOPUP.Clear();

               functionLockTOPUP(true);

               ShowDVG();
               FormatDGV();

            }
         }
         catch (Exception ex)
         {
            tr.Rollback();
            LBTOPUP.Items.Clear();
            LBTOPUP.Items.Add("เกิดข้อผิดพลาด เนื่องจาก " + ex.Message);
         }
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
//////////////////////////// ฟังชั่น เก็บข้อมูลใน การ์ด rfid ////////////////////////////
private void setRFID(string setNumber)
{
  ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;

  int device_address = 0;
  byte[] atq = new byte[65];

  rfidm1_result = ISO14443A.MF_Request(device_address, RequestModeTypes.RequestIdle,
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;
  };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
byte[] key = new byte[6] { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };

rfidm1_result = Mifare.MF_LoadKey(device_address, key);
if (ResultCodeTypes.CommandOK != rfidm1_result)
{
   return;
};

byte sector_number = 0;
byte block_number = 2; ////

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);

byte[] temp = System.Text.Encoding.Default.GetBytes(setNumber);
byte[] buffer = new byte[16];
for (int i = 0; i < 16; ++i)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            if (i < temp.Length)
            {
                buffer[i] = temp[i];
            }
            else
            {
                buffer[i] = 0;
            }
        };

        if (ResultCodeTypes.CommandOK == Mifare.MF_Write(device_address, (byte)((sector_number
* 4) + block_number), 1, buffer))
        {


            Application.DoEvents();

            Thread.Sleep(3);

            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);

            Thread.Sleep(1);

            _WriteFlagNUMBER = true;


        }
    }
    ////////////////////////// ฟังก์ชัน จบการทำงานขั้นตอน  เติมเงิน //////////////////////////
    private void BTendTOPUP_Click(object sender, EventArgs e)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    functionEndTOPUP();
}

//////////////////////////////////////////////////

string getTagData = "";
// อ่านข้อมูลจากการ์ด rfid
private void getTag()
    {

    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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
};

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)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
this.Beep2(device_address);

Thread.Sleep(3);

ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);

Thread.Sleep(3);

_ReadFlag = false;

continue;
};

Thread.Sleep(2);

byte[] buffer = 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, buffer))
{

getTagData = READTAG(buffer);
_ReadFlag = false;

this.Beep1(device_address);

Application.DoEvents();

Thread.Sleep(3);

ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Thread.Sleep(3);


        };
        getDATABASE();
        Application.DoEvents();
    };
}
// ตรวจสอบวัน เพื่อเช็คว่า มีวันคงเหลือพอที่จะใช้งานหรือไม่
private bool checkDate(string datenow, string dateold)
    {
        bool dateState = true;


        string[] dateSplit1 = datenow.Split(' ');
        string dnow = dateSplit1[0];
        string mnow = dateSplit1[1];
        string ynow = dateSplit1[2];
        Month now_month_enum = (Month)Enum.Parse(typeof(Month), mnow, false);


        string[] dateSplit2 = dateold.Split(' ');
        string dold = dateSplit2[0];
        string mold = dateSplit2[1];
        string yold = dateSplit2[2];
        Month old_month_enum = (Month)Enum.Parse(typeof(Month), mold, false);


        if (Convert.ToInt32(yold) < Convert.ToInt32(ynow)) //ตรวจสอบปีก่อน
        {
            dateState = false;
        }
        else if ((int)old_month_enum < (int)now_month_enum) //ตรวจสอบปีก่อน
        {
            dateState = false;
        }
        else if (Convert.ToInt32(dold) < Convert.ToInt32(dnow)) //ตรวจสอบปีก่อน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            dateState = false;
        }
        else
        {
            dateState = true;
        }

        return(dateState);
    }
// รับข้อมูลจาก ฐานข้อมูล
private void getDATABASE()
    {
        string nameS = "";
        string surnameS = "";
        string numberS = "";
        string status = "";
        string messageStatus = "";
        string dateObs = "";

        try
        {
            StringBuilder sb = new StringBuilder();
            sb.Remove(0, sb.Length);
            sb.Append("SELECT rfid,name,surname,number,status,date");
            sb.Append(" FROM mainDB");
            sb.Append(" WHERE (rfid=@rfid)");
            string sqlshow = sb.ToString();

            SqlCommand com = new SqlCommand();
            SqlDataReader dr;
            com.CommandType = CommandType.Text;
            com.CommandText = sqlshow;
            com.Connection = Conn;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            com.Parameters.Clear();
            com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = getTagData;
            dr = com.ExecuteReader();
            if (dr.HasRows)
            {
                dr.Read();
                nameS = dr.GetString(dr.GetOrdinal("name"));
                surnameS = dr.GetString(dr.GetOrdinal("surname"));
                numberS = dr.GetString(dr.GetOrdinal("number"));
                status = dr.GetString(dr.GetOrdinal("status"));
                dateObs = dr.GetString(dr.GetOrdinal("date"));

                dr.Close();
            }
            else
            {
                dr.Close();
                return;
            }

        }
        catch
        {
            MessageBox.Show("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูลได้", "ผลการค้นหา",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

        bool NowState = CheckStatus(status.Trim());
        int CCredit = CheckCredit(numberS);
        messageStatus = miniconvert(NowState);

        if (!checkDate(DateTime.Now.ToLongDateString(), dateObs))
        {
            TBsuggest.Clear();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TBsuggest.Text = "บัตรหมดอายุ ไม่สามารถใช้งานได้";
    TBnameOBS.Clear();
    TBnameOBS.Text = nameS;
    TBsurOBS.Clear();
    TBsurOBS.Text = surnameS;
    TBstateOBS.Clear();
    TBmoneyOBS.Clear();
    TBmoneyOBS.Text = "0";
    TBdayOBS.Clear();
    TBdayOBS.Text = "NONE";

    SentCommand("9000");
    return;
}

else if (CCredit <= 0)
{
    TBsuggest.Clear();
    TBsuggest.Text = "ยอดเงินของคุณไม่พอ";
    TBnameOBS.Clear();
    TBnameOBS.Text = nameS;
    TBsurOBS.Clear();
    TBsurOBS.Text = surnameS;
    TBstateOBS.Clear();

    TBmoneyOBS.Clear();
    TBmoneyOBS.Text = "0";
    TBdayOBS.Clear();
    TBdayOBS.Text = "NONE";

    SentCommand("9000");
    return;
}
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {

            OPERshow(ConvertCredit(NowState, numberS), nameS, surnameS,
NowState,messageStatus,dateObs);
            UPdate(ConvertCredit(NowState, numberS), getTagData, ChangeState(NowState));
            setLog(getTagData, numberS, nameS, surnameS, DateFunction(), TimeFunction());

        }
    }
// แสดง สถานะการเข้าออก การใช้งาน
private string miniconvert(bool nowStatus)
    {
        string messageStatus = "";
        if (nowStatus)
        {
            messageStatus = "ออก";
        }
        else
        {
            messageStatus = "เข้า";
        }
        return messageStatus;
    }
// ตรวจสอบ สถานะการเข้าออก การใช้งาน
private bool CheckStatus(string status)
    {
        bool nowstate;

        if (status.Trim() == "on")
        {
            nowstate = true;
        }
        else
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            nowstate = false;
        }
        return nowstate;

    }
    // ตรวจสอบยอดการใช้งาน
    private int CheckCredit(string credit)
    {
        int newvalue = int.Parse(credit);
        if (newvalue == 0)
        {
            return 0;
        }
        else
        {
            newvalue = newvalue - 1;
        }
        return newvalue;

    }
    // แสดงผล การใช้งาน เมื่อเวลาที่หน้าต่างโปรแกรมทำงาน
    private void OPERshow(string money, string name, string surname, bool nowstate,string
message,string date)
    {
        SentCommand(CommandOutput(nowstate, money));

        TBnameOBS.Clear();
        TBnameOBS.Text = name;
        TBsurOBS.Clear();
        TBsurOBS.Text = surname;

        TBstateOBS.Clear();

        TBstateOBS.Text = message;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        TBdayOBS.Clear();
        TBdayOBS.Text = date;

        if (!nowstate)
        {
          TBsuggest.Clear();
          TBsuggest.Text = "เข้าใช้บริการ  ขอบคุณที่ใช้บริการ";
        }
        else
        {
          TBsuggest.Clear();
          TBsuggest.Text = "ออกจากการใช้บริการ  ขอบคุณที่ใช้บริการ";
        }

        TBmoneyOBS.Clear();
        TBmoneyOBS.Text = money;


    }
// ส่งคำสั่งเปิดปิดประตู
private void SentCommand(string number)
    {
        if (SRport.IsOpen)
        {
          SRport.Close();
        }
        SRport.Open();
        SRport.Write(number);
        SRport.Close();
    }
// แปลงคำสั่งเปิดปิดประตูให้ ไมโครคอนโทรลเลอร์ เข้าใจ
private string CommandOutput(bool nowstate, string money)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        string newmoney;
        int newvalue;

        if (nowstate)
        {
           newvalue = int.Parse(money);
           newvalue = newvalue + 1000;
           newmoney = newvalue.ToString();
        }
        else
        {
           newvalue = int.Parse(money);
           newvalue = newvalue + 5000;
           newmoney = newvalue.ToString();

        }

        return newmoney;
    }
// อัพเดท สถานะการใช้งานปัจจุบันลงใน ฐานข้อมูล
private void UPdate(string Ncredit, string RFID, string nowstate)
    {
        tr = Conn.BeginTransaction();
        try
        {
           sb.Remove(0, sb.Length);
           sb.Append("UPDATE mainDB");
           sb.Append(" SET number=@number,status=@status");
           sb.Append(" WHERE (rfid=@rfid)");
           string sqltop;
           sqltop = sb.ToString();

           com.CommandText = sqltop;
           com.CommandType = CommandType.Text;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            com.Connection = Conn;
            com.Transaction = tr;
            com.Parameters.Clear();
            com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = RFID;
            com.Parameters.Add("@number", SqlDbType.NVarChar).Value = Ncredit;
            com.Parameters.Add("@status", SqlDbType.NVarChar).Value = nowstate;

            int result;
            result = com.ExecuteNonQuery();
            if (result == 0)
            {
                tr.Rollback();
            }
            else
            {
                tr.Commit();
            }
        }
        catch (Exception ex)
        {
            tr.Rollback();
            MessageBox.Show("เกิดข้อผิดพลาด เนื่องจาก " + ex.Message, "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }

    }
    // ฟังก์ชั่นจบการทำงาน OBSERVE
    private void BTendOBS_Click(object sender, EventArgs e)
    {
        _ReadFlag = false;
        boolDB = false;
        BTstartOBS.Enabled = true;
        BTendOBS.Enabled = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MyTab.TabPages.Clear();
        MyTab.TabPages.Add(TABsystem);
        MyTab.TabPages.Add(LOG);
        MyTab.TabPages.Add(TABauth);



        TBsuggest.Clear();
        TBnameOBS.Clear();
        TBsurOBS.Clear();
        TBstateOBS.Clear();
        TBmoneyOBS.Clear();
        TBdayOBS.Clear();
    }
// ฟังก์ชั่นเปลี่ยนแปลงค่าเงิน จาก string เป็น int
private string ConvertCredit(bool doorstate, string credit)
    {
        string money;

        if (doorstate)
        {
            money = credit;
            int newvalue = int.Parse(credit);

        }
        else
        {
            int newvalue = int.Parse(credit);
            if (newvalue == 0)
            {
                money = "0";
            }
            else
            {
                newvalue = newvalue - 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            money = newvalue.ToString();
        }
    }
    return money;
}
// ฟังก์ชั่นเปลี่ยน สถานะ
private string ChangeState(bool nowstate)
{
    string updateState;
    if (nowstate)
    {
        updateState = "off";
    }
    else
    {
        updateState = "on";
    }

    return updateState;
}

//////////////////////////////////////////////

// โปรแกรม เก็บ สถานะการใช้งาน เข้าออกประตู
private void setLog(string rfid, string money, string name, string surname, string date, string time)
{
    tr = Conn.BeginTransaction();
    try
    {

        sb.Remove(0, sb.Length);
        sb.Append("INSERT INTO log (rfid,money,name,surname,date,time)");
        sb.Append(" VALUES (@rfid,@money,@name,@surname,@date,@time)");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            string sqlAdd;
            sqlAdd = sb.ToString();

            com.CommandText = sqlAdd;
            com.CommandType = CommandType.Text;
            com.Connection = Conn;
            com.Transaction = tr;
            com.Parameters.Clear();
            com.Parameters.Add("@rfid", SqlDbType.NChar).Value = rfid;
            com.Parameters.Add("@money", SqlDbType.NChar).Value = money;
            com.Parameters.Add("@name", SqlDbType.NChar).Value = name;
            com.Parameters.Add("@surname", SqlDbType.NChar).Value = surname;
            com.Parameters.Add("@date", SqlDbType.NChar).Value = date;
            com.Parameters.Add("@time", SqlDbType.NChar).Value = time;
            com.ExecuteNonQuery();

            tr.Commit();


        }
        catch
        {
            MessageBox.Show("เกิดข้อผิดพลาดระหว่างอัพเดตข้อมูล", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            tr.Rollback();
        }

        ShowLog();
        FormatLog();
    }
    // ส่งค่าวันที่ออกมา
    private string DateFunction()
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return DateTime.Now.ToLongDateString();
}
// ส่งค่าเวลาปัจจุบันออกมา
private string TimeFunction()
{
    string TimeInString = "";
    int hour = DateTime.Now.Hour;
    int min = DateTime.Now.Minute;
    int sec = DateTime.Now.Second;

    TimeInString = (hour < 10) ? "0" + hour.ToString() : hour.ToString();
    TimeInString += ":" + ((min < 10) ? "0" + min.ToString() : min.ToString());
    TimeInString += ":" + ((sec < 10) ? "0" + sec.ToString() : sec.ToString());
    return TimeInString;
}
// ฟังก์ชันแสดง สถานะการใช้งาน
private void ShowLog()
{
    sb.Remove(0, sb.Length);
    sb.Append("SELECT rfid,money,name,surname,date,time");
    sb.Append(" FROM log");
    sb.Append(" ORDER BY date,time DESC");
    string sqlshow = sb.ToString();

    SqlDataReader dr;
    DataTable dt;

    com = new SqlCommand();
    com.CommandType = CommandType.Text;
    com.CommandText = sqlshow;
    com.Connection = Conn;
    dr = com.ExecuteReader();
    if (dr.HasRows)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dt = new DataTable();
        dt.Load(dr);
        DGVlog.DataSource = dt;
    }
    else
    {
        DGVlog.DataSource = null;

    }
    dr.Close();
}
// ฟังก์ชัน สร้างรูปแบบการแสดง สถานะการใช้งาน
private void FormatLog()
{
    if (DGVlog.RowCount > 0)
    {
        DGVlog.Columns[0].HeaderText = "RFID";
        DGVlog.Columns[1].HeaderText = "money";
        DGVlog.Columns[2].HeaderText = "Name";
        DGVlog.Columns[3].HeaderText = "Surname";
        DGVlog.Columns[4].HeaderText = "Date";
        DGVlog.Columns[5].HeaderText = "Time";

        DGVlog.Columns[0].Width = 70;
        DGVlog.Columns[1].Width = 60;
        DGVlog.Columns[2].Width = 100;
        DGVlog.Columns[3].Width = 100;
        DGVlog.Columns[4].Width = 100;
        DGVlog.Columns[5].Width = 60;
    }
}

////////////////// topup new //////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// เหตุการณ์เมื่อกดปุ่ม ตรวจสอบยอดเงิน
private void BTbeforetopup_Click(object sender, EventArgs e)
{
   GetTagRFID("topup");
   if (TBrfidTOPUP.Text == "")
   {
      LBTOPUP.Items.Clear();
      LBTOPUP.Items.Add(" เกิดปัญหาในการอ่านการ์ด RFID กรุณาลองใหม่ ");
      TBrfidTOPUP.Focus();
   }
   else
   {
      GetDatabeforeTopUp();
      TBmoneyNEW.Text = "";
      if (clickTopup)
      {
         DataFunctionLock(true);
      }
   }
}
// ฟังก์ชั่น อ่านข้อมูลก่อนการเติมเงิน
private void GetDatabeforeTopUp()
{
   try
   {
      StringBuilder sb = new StringBuilder();
      sb.Remove(0, sb.Length);
      sb.Append("SELECT rfid,name,surname,phone,number,date");
      sb.Append(" FROM mainDB");
      sb.Append(" WHERE (rfid=@rfid)");
      string sqlpass = sb.ToString();

      SqlCommand com = new SqlCommand();
      SqlDataReader dr;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
com.CommandType = CommandType.Text;
com.CommandText = sqlpass;
com.Connection = Conn;
com.Parameters.Clear();
com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidTOPUP.Text.Trim();
dr = com.ExecuteReader();
if (dr.HasRows)
{
   dr.Read();
   TBnameTOPUP.Text = dr.GetString(dr.GetOrdinal("name"));
   TBsurTOPUP.Text = dr.GetString(dr.GetOrdinal("surname"));
   TBtelTOPUP.Text = dr.GetString(dr.GetOrdinal("phone"));
   TBmoneyOLD.Text = dr.GetString(dr.GetOrdinal("number")); // show tb old
   setNumber = dr.GetString(dr.GetOrdinal("number"));
   //datefortopup = dr.GetString(dr.GetOrdinal("date"));

   //functionLockTOPUP(false);
   clickTopup = false; /////
   LBTOPUP.Items.Clear();
   LBTOPUP.Items.Add("เช็คเรียบร้อยครับ");

}
else
{
   LBTOPUP.Items.Clear();
   LBTOPUP.Items.Add("เกิดปัญหาในการเชื่อมต่อฐานข้อมูล");
   TBrfidTOPUP.Clear();
   TBrfidTOPUP.Focus();
   clickTopup = false;
   return;
  }
  dr.Close();
}
catch
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        LBTOPUP.Items.Clear();
        LBTOPUP.Items.Add("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูลได้");
    }
}

// ฟังก์ชั่น รับค่าจากฐานข้อมูล
private void GetDataForTopUp()
{
    try
    {
        StringBuilder sb = new StringBuilder();
        sb.Remove(0, sb.Length);
        sb.Append("SELECT rfid,name,surname,phone,number,date");
        sb.Append(" FROM mainDB");
        sb.Append(" WHERE (rfid=@rfid)");
        string sqlpass = sb.ToString();

        SqlCommand com = new SqlCommand();
        SqlDataReader dr;
        com.CommandType = CommandType.Text;
        com.CommandText = sqlpass;
        com.Connection = Conn;
        com.Parameters.Clear();
        com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidTOPUP.Text.Trim();
        dr = com.ExecuteReader();
        if (dr.HasRows)
        {
            dr.Read();
            TBnameTOPUP.Text = dr.GetString(dr.GetOrdinal("name"));
            TBsurTOPUP.Text = dr.GetString(dr.GetOrdinal("surname"));
            TBtelTOPUP.Text = dr.GetString(dr.GetOrdinal("phone"));
            TBmoneyOLD.Text = dr.GetString(dr.GetOrdinal("number"));
            setNumber = dr.GetString(dr.GetOrdinal("number"));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            datefortopup = dr.GetString(dr.GetOrdinal("date"));

            //functionLockTOPUP(false);
            //clickTopup = false; /////
            LBTOPUP.Items.Clear();
            //LBTOPUP.Items.Add("เช็คเรียบร้อยครับ");เก็บข้อมูลเรียบร้อย

        }
        else
        {
            LBTOPUP.Items.Clear();
            LBTOPUP.Items.Add("เกิดปัญหาในการเชื่อมต่อฐานข้อมูล ทำให้ไม่มีข้อมูลมาแสดงเพื่อ
ประมวลผล");
            TBrfidTOPUP.Clear();
            TBrfidTOPUP.Focus();
            clickTopup = false;
            return;
        }
        dr.Close();
    }
    catch
    {
        LBTOPUP.Items.Clear();
        LBTOPUP.Items.Add("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูลได้");
    }
}
// แปลงค่าค่าเงินใน การเติมเงิน
private void MoneyForTopup(string m)
{
    int newvalue = int.Parse(m);
    int value = int.Parse(setNumber);
    value = value + newvalue;
    setNumber = value.ToString();
    return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
// ฟังก์ชันเติมเงิน
private void functionTopupNew()
{

    tr = Conn.BeginTransaction();
    try
    {
      sb.Remove(0, sb.Length);
      sb.Append("UPDATE mainDB");
      sb.Append(" SET number=@number,date=@date");
      sb.Append(" WHERE (rfid=@rfid)");
      string sqltop;
      sqltop = sb.ToString();

      com.CommandText = sqltop;
      com.CommandType = CommandType.Text;
      com.Connection = Conn;
      com.Transaction = tr;
      com.Parameters.Clear();
      com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value = TBrfidTOPUP.Text.Trim();
      com.Parameters.Add("@number", SqlDbType.NVarChar).Value = setNumber;
      com.Parameters.Add("@date", SqlDbType.NChar).Value =
UpdateExpireDate(datefortopup);

      int result;
      result = com.ExecuteNonQuery();
      if (result == 0)
      {
        tr.Rollback();
        LBTOPUP.Items.Clear();
        LBTOPUP.Items.Add("ไม่สามารถเติมเงินได้ เพราะขั้นตอนการส่งข้อมูลไปที่ฐานข้อมูล
มีปัญหา ");
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
        {
            tr.Commit();
            LBTOPUP.Items.Clear();
            LBTOPUP.Items.Add("เพิ่มจำนวนเงินเรียบร้อยแล้ว ");
            TBmoneyNEW.Text = setNumber;

        }
    }
    catch (Exception ex)
    {
        tr.Rollback();
        LBTOPUP.Items.Clear();
        LBTOPUP.Items.Add("เกิดข้อผิดพลาด เนื่องจาก " + ex.Message);
    }


}
// ฟังก์ชันส่งข้อมูลไปเก็บไว้ในการ์ด rfid ขั้นตอนการเติมเงินแบบใหม่
private void GetTagRFIDaaa(string m)
{

    ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;

    _ReadFlag = true;

    while (_ReadFlag)
    {
        Application.DoEvents();

        int device_address = 0;
        byte[] atq = new byte[65];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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[] buffer = 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, buffer))
{


  TBrfidTOPUP.Text = READTAG(buffer);

  _ReadFlag = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
this.Beep1(device_address);

//Application.DoEvents();

Thread.Sleep(3);

//ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);

Thread.Sleep(3);


if (TBrfidTOPUP.Text == "")
{
   LBTOPUP.Items.Clear();
   LBTOPUP.Items.Add(" เกิดปัญหาในการอ่านการ์ด RFID กรุณาลองใหม่ ");
   TBrfidTOPUP.Focus();
}
else
{
   GetDataForTopUp();
   MoneyForTopup(m);

}

};

sector_number = 0;
block_number = 2;

rfidm1_result = Mifare.MF_Auth(device_address, KeyModeTypes.KeyA, uid,
(sector_number * 4) + block_number);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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);
byte[] temp = System.Text.Encoding.Default.GetBytes(setNumber);
byte[] buffer2 = new byte[16];
for (int i = 0; i < 16; ++i)
{
   if (i < temp.Length)
   {
      buffer2[i] = temp[i];
   }
   else
   {
      buffer2[i] = 0;
   }
};
if (ResultCodeTypes.CommandOK == Mifare.MF_Write(device_address,
(byte)((sector_number * 4) + block_number), 1, buffer2))
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            setLogTopup(TBrfidTOPUP.Text.Trim(), TBnameTOPUP.Text.Trim(),
TBsurTOPUP.Text.Trim(), System.DateTime.Now.ToLongTimeString(), m,
System.DateTime.Now.ToShortDateString());
            functionTopupNew();


            Application.DoEvents();

            Thread.Sleep(3);

            ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);

            Thread.Sleep(1);


        }

        Application.DoEvents();
    };
}
// ฟังก์ชันในปุ่ม เติมเงิน 250
private void bt200_Click(object sender, EventArgs e)
{
    GetTagRFIDaaa("15");
}
// ฟังก์ชันในปุ่ม เติมเงิน 400
private void bt400_Click(object sender, EventArgs e)
{
    GetTagRFIDaaa("20");
}
// ฟังก์ชันในปุ่ม เติมเงิน 500
private void bt500_Click(object sender, EventArgs e)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
   GetTagRFIDaaa("30");
}
// เหตุการณ์ในปุ่ม แก้ไขข้อมูลการเติมเงินที่ผิดพลาด
private void BTcheckTopupNow_Click(object sender, EventArgs e)
{
   DGVet.DataSource = null;
   GetTagRFID("edittopup");
   if (TBedittopupRFID.Text == "")
   {
      MessageBox.Show("เกิดปัญหาระหว่างอ่านการ์ด RFID");
      return;
   }
   else
   {
      GetDataEditTopUp();
      functionLockbuttonEditTopup(true);
      ShowDVGEditTopup();
      FormatDGVed();

   }
}

string dateedittopup;
// เก็บข้อมูลในขั้นตอน แก้ขอข้อมูลการเติมเงินที่ผิดพลาด
private void GetDataEditTopUp()
{
   try
   {
      StringBuilder sb = new StringBuilder();
      sb.Remove(0, sb.Length);
      sb.Append("SELECT rfid,name,surname,phone,number,date");
      sb.Append(" FROM mainDB");
      sb.Append(" WHERE (rfid=@rfid)");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        string sqlpass = sb.ToString();

        SqlCommand com = new SqlCommand();
        SqlDataReader dr;
        com.CommandType = CommandType.Text;
        com.CommandText = sqlpass;
        com.Connection = Conn;
        com.Parameters.Clear();
        com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value =
TBedittopupRFID.Text.Trim();
        dr = com.ExecuteReader();
        if (dr.HasRows)
        {
          dr.Read();
          TBedittopupName.Text = dr.GetString(dr.GetOrdinal("name"));
          TBedittopupSurname.Text = dr.GetString(dr.GetOrdinal("surname"));
          TBedittopupTel.Text = dr.GetString(dr.GetOrdinal("phone"));
          TBshowTU.Text = dr.GetString(dr.GetOrdinal("number"));
          dateedittopup = dr.GetString(dr.GetOrdinal("date"));
        }
        else
        {
          MessageBox.Show("ไม่มีข้อมูลในฐานข้อมูล");
          return;
        }
        dr.Close();
      }
      catch
      {
        LBTOPUP.Items.Clear();
        LBTOPUP.Items.Add("ไม่สามารถเชื่อมต่อข้อมูลจากฐานข้อมูลได้");
      }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/// ////// ฟังก์ชั่น ในหน้า topup
// เก็บ log การเติมเงิน
private void setLogTopup(string rfid, string name, string surname, string time,string money,string
date)
{

  string primarykey = rfid + date + time;
  tr = Conn.BeginTransaction();

  try
  {
    sb.Remove(0, sb.Length);
    sb.Append("INSERT INTO logtopup (rfid,name,surname,time,money,date,prikey)");
    sb.Append(" VALUES (@rfid,@name,@surname,@time,@money,@date,@prikey)");
    string sqlAdd;
    sqlAdd = sb.ToString();

    com.CommandText = sqlAdd;
    com.CommandType = CommandType.Text;
    com.Connection = Conn;
    com.Transaction = tr;
    com.Parameters.Clear();
    com.Parameters.Add("@rfid", SqlDbType.NChar).Value = rfid;
    com.Parameters.Add("@name", SqlDbType.NChar).Value = name;
    com.Parameters.Add("@surname", SqlDbType.NChar).Value = surname;
    com.Parameters.Add("@time", SqlDbType.NChar).Value = time;
    com.Parameters.Add("@money", SqlDbType.NChar).Value = money;
    com.Parameters.Add("@date", SqlDbType.NChar).Value = date;
    com.Parameters.Add("@prikey", SqlDbType.NChar).Value = primarykey;
    com.ExecuteNonQuery();

    tr.Commit();
  }
  catch
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            MessageBox.Show("เกิดข้อผิดพลาดระหว่างอัพเดตข้อมูล", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            tr.Rollback();
        }

    }
    // เหตุการณ์ในปุ่มตรวจสอบการเติมเงิน
    private void BTlogTopup_Click(object sender, EventArgs e)
    {
        MyTab.TabPages.Clear();
        MyTab.TabPages.Add(TABeditTopup);

    }
    // เหตุการณ์ในปุ่มกลับไปหน้าเติมเงิน
    private void BTbacktoTopup_Click(object sender, EventArgs e)
    {
        MyTab.TabPages.Clear();
        MyTab.TabPages.Add(TABtopup);
        MyTab.TabPages.Add(TABregister);
        MyTab.TabPages.Add(TABedit);
        MyTab.TabPages.Add(TABauth);
    }
    // ฟังก์ชั่น lock ปุ่มในขั้นตอนการแก้ไขการเติมเงิน
    private void functionLockbuttonEditTopup(bool b)
    {
        BTdec200.Enabled = b;
        BTdec400.Enabled = b;
        BTdec500.Enabled = b;
        BTcheckTopupNow.Enabled = !b;
    }
    // ฟังก์ชั่น ลดค่าเงิน
    private string MoneyDECEditTopup(string dec,string nowmoney)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int newvalue = int.Parse(dec);
        int value = int.Parse(nowmoney);
        value = value - newvalue;
        nowmoney = value.ToString();
        return nowmoney;
    }
    // เหตุการณ์ในปุ่ม ลดเงิน 250
    private void BTdec200_Click(object sender, EventArgs e)
    {
        functionedittopupfull("15");
    }
    // ฟังก์ชั่นเปลี่ยนค่าเงิน ในขั้นตอนการแก้ไขการเติมเงิน
    private void functionedittopupfull(string number)
    {
        string mn = MoneyDECEditTopup(number, TBshowTU.Text.Trim());
        functionEditTopupNew(mn);

GetTagRFIDeditTopup(mn);//SetDataToRFID(TBedittopupName.Text.Trim(),TBshowTU.Text.Trim())
;
        functionLockbuttonEditTopup(false);
    }
    // ฟังก์ชั่น เก็บข้อมูลลงในฐานข้อมูล
    private void functionEditTopupNew(string mn)
    {

        tr = Conn.BeginTransaction();
        try
        {
            sb.Remove(0, sb.Length);
            sb.Append("UPDATE mainDB");
            sb.Append(" SET number=@number,date=@date");
            sb.Append(" WHERE (rfid=@rfid)");
            string sqltop;
            sqltop = sb.ToString();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            com.CommandText = sqltop;
            com.CommandType = CommandType.Text;
            com.Connection = Conn;
            com.Transaction = tr;
            com.Parameters.Clear();
            com.Parameters.Add("@rfid", SqlDbType.NVarChar).Value =
TBedittopupRFID.Text.Trim() ;
            com.Parameters.Add("@number", SqlDbType.NVarChar).Value = mn;
            com.Parameters.Add("@date", SqlDbType.NChar).Value =
UpdateExpireDateForEditTopup(dateedittopup);//////////////////////

            int result;
            result = com.ExecuteNonQuery();
            if (result == 0)
            {
               tr.Rollback();
               MessageBox.Show("ไม่สามารถเติมเงินได้ เพราะขั้นตอนการส่งข้อมูลไปที่ฐานข้อมูลมี
ปัญหา ");
            }
            else
            {
               tr.Commit();

               TBshowTU.Text = mn;
            }
         }
         catch (Exception ex)
         {
            tr.Rollback();
         }

      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ฟังก์ชั่น แสดงข้อมูล log การเติมเงิน
private void ShowDVGEditTopup()
{
   sb.Remove(0, sb.Length);
   sb.Append("SELECT rfid,money,date,time");
   sb.Append(" FROM logtopup");
   sb.Append(" WHERE (rfid=@rfid)");
   sb.Append(" ORDER BY date DESC,time DESC");
   string sqlshow = sb.ToString();

   SqlDataReader dr;
   DataTable dt;

   com = new SqlCommand();
   com.CommandType = CommandType.Text;
   com.CommandText = sqlshow;
   com.Connection = Conn;
   com.Parameters.Clear();
   com.Parameters.Add("@rfid", SqlDbType.NChar).Value = TBedittopupRFID.Text.Trim();
   dr = com.ExecuteReader();
   if (dr.HasRows)
   {
      dt = new DataTable();
      dt.Load(dr);
      DGVet.DataSource = dt;

   }
   else
   {
      DGVet.DataSource = null;

   }
   dr.Close();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ฟังก์ชั่น สร้าง format ในการแสดง log
private void FormatDGVed()
{
   if (DGVet.RowCount > 0)
   {
      DGVet.Columns[0].HeaderText = "RFID";
      DGVet.Columns[1].HeaderText = "CREDIT";
      DGVet.Columns[2].HeaderText = "DATE";
      DGVet.Columns[3].HeaderText = "TIME";



      DGVet.Columns[0].Width = 75;
      DGVet.Columns[1].Width = 60;
      DGVet.Columns[2].Width = 75;
      DGVet.Columns[3].Width = 75;


   }

}
// ฟังก์ชันรับค่า จากการ์ด rfid
private void GetTagRFIDeditTopup(string money)
{

   ResultCodeTypes rfidm1_result = ResultCodeTypes.CommandOK;


   _ReadFlag = true;


   while (_ReadFlag)
   {
      Application.DoEvents();


      int device_address = 0;
      byte[] atq = new byte[65];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            continue;
        };

        byte sector_number = 0;

        byte block_number = 2;

        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);

        byte[] temp = System.Text.Encoding.Default.GetBytes(money);
        byte[] buffer2 = new byte[16];
        for (int i = 0; i < 16; ++i)
        {
            if (i < temp.Length)
            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            buffer2[i] = temp[i];
          }
          else
          {
            buffer2[i] = 0;
          }
        };

        if (ResultCodeTypes.CommandOK == Mifare.MF_Write(device_address,
(byte)((sector_number * 4) + block_number), 1, buffer2))
        {

          _ReadFlag = false;

          this.Beep1(device_address);

          Application.DoEvents();

          Thread.Sleep(3);

          ISO14443A.MF_Halt(device_address, HaltModeTypes.StarndardHalt);

          Thread.Sleep(1);

        }

        Application.DoEvents();
      };
    }
    // เหตุการณ์ ในปุ่ม ลดเงิน 400
    private void BTdec400_Click(object sender, EventArgs e)
    {
      functionedittopupfull("20");
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // เหตุการณ์ ในปุ่ม ลดเงิน 500
    private void BTdec500_Click(object sender, EventArgs e)
    {
       functionedittopupfull("30");
    }
    // ฟังก์ชั่น ลดเดือน
    private string UpdateExpireDateForEditTopup(string OldExpireDate)
    {
       string[] dateSplit = OldExpireDate.Split(' ');
       string d = dateSplit[0];
       string m = dateSplit[1];
       string y = dateSplit[2];

       Month month_enum = (Month)Enum.Parse(typeof(Month), m, false);
       if ((int)month_enum == 12)
       {
          int newyear = int.Parse(y);
          newyear++;
          y = newyear.ToString();
       }
       int outputint = ((int)month_enum - 1) % 12;
       Month newm = (Month)outputint;
       string output = string.Format("{0} {1} {2}", d, newm.ToString(), y);
       return output;
    }

  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดที่ใช้ในการทดลองควบคุมการเปิดปิดประตูและแสดงผลที่จอแอลซีดี

```
#include "D:\lcd_telecom\nueng_project\rfid.h"
#define LCD_TYPE 2
#define use_portb_lcd TRUE
#include <lcd.c>

#define P1 PIN_C1                    //P is entrance
#define P2 PIN_C2
#define P3 PIN_C3
#define P4 PIN_C4

#define Q1 PIN_C5                    //Q is exit
#define Q2 PIN_A0
#define Q3 PIN_A1
#define Q4 PIN_A2


#define ON(x) output_high(x)
#define OFF(x) output_low(x)

int1 motor_left_p (int8 step)
{
  int8 i;
  for(i=1;i<=step/4;i++)
  {
    OFF(P4);
    ON(P3);                  //3
    delay_ms(20);

    ON(P2);                  //32
    delay_ms(20);

    OFF(P3);                 //2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    delay_ms(20);

    ON(P1);              //21
    delay_ms(20);

    OFF(P2);             //1
    delay_ms(20);

    ON(P4);              //14
    delay_ms(20);

    OFF(P1);             //4
    delay_ms(20);

    ON(P3);              //43
    delay_ms(20);
  }
  return(TRUE);
}

int1 motor_right_p (int8 step)
{
  int8 i;
  for(i=1;i<=step/4;i++){
    OFF(P4);
    ON(P1);
    delay_ms(20);    //1

    ON(P2);          //12
    delay_ms(20);

    OFF(P1);         //2
    delay_ms(20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ON(P3);         //23
    delay_ms(20);

    OFF(P2);        //3
    delay_ms(20);

    ON(P4);         //34
    delay_ms(20);

    OFF(P3);        //4
    delay_ms(20);

    ON(P1);         //41
    delay_ms(20);
  }
  return(TRUE);
}


int1 motor_left_q (int8 step)
{
  int8 i;
  for(i=1;i<=step/4;i++)
  {
    OFF(Q4);
    ON(Q3);             //3
    delay_ms(20);

    ON(Q2);             //32
    delay_ms(20);

    OFF(Q3);            //2
    delay_ms(20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ON(Q1);              //21
    delay_ms(20);

    OFF(Q2);             //1
    delay_ms(20);

    ON(Q4);              //14
    delay_ms(20);

    OFF(Q1);             //4
    delay_ms(20);

    ON(Q3);              //43
    delay_ms(20);
  }
  return(TRUE);
}

int1 motor_right_q (int8 step)
{
  int8 i;
  for(i=1;i<=step/4;i++){
    OFF(Q4);
    ON(Q1);
    delay_ms(20);    //1

    ON(Q2);          //12
    delay_ms(20);

    OFF(Q1);         //2
    delay_ms(20);

    ON(Q3);          //23
    delay_ms(20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    OFF(Q2);        //3
    delay_ms(20);

    ON(Q4);         //34
    delay_ms(20);

    OFF(Q3);        //4
    delay_ms(20);

    ON(Q1);         //41
    delay_ms(20);
  }
  return(TRUE);
}

void main()
{
  int8 i;
  char buffer[4];
  setup_adc_ports(NO_ANALOGS|VSS_VDD);
  setup_adc(ADC_OFF|ADC_TAD_MUL_0);
  setup_spi(FALSE);
  setup_wdt(WDT_OFF);
  setup_timer_0(RTCC_INTERNAL);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);
  setup_comparator(NC_NC_NC_NC);
  setup_vref(FALSE);
  setup_oscillator(False);

  for(i=0;i<5;i++)
  {
    output_low(PIN_C0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  delay_ms(200);
  output_high(PIN_C0);
  delay_ms(200);
}
lcd_init();

lcd_putc("\f   BTS SYSTEM   ");
lcd_putc("\n TELECOM PROJECT ");
printf("karaoken");
output_high(PIN_B3);
output_high(PIN_A3);
delay_ms(1500);
while(1)
{
  for(i=0;i<4;i++)
    buffer[i]=getc();



  if(buffer[0]=='1')
  {
    output_low(PIN_A3);              //turn off LCD exit
    lcd_putc("\f   WELLCOME   ");
    lcd_putc("\n YOUR CREDIT ");
    for(i=1;i<4;i++)
      lcd_putc(buffer[i]);
    output_high(PIN_B3);             //turn on  LCD Entrance
    motor_right_p(12);
    delay_ms(2000);
    motor_left_p(12);
    lcd_putc("\f   BTS SYSTEM   ");
    lcd_putc("\n TELECOM PROJECT ");
    output_high(PIN_A3);             //turn on LCD exit
  }
  else if(buffer[0]=='5')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      output_low(PIN_B3);                  //turn off  LCD Entrance
      lcd_putc("\f   THANK YOU    ");
      lcd_putc("\nYOUR CREDIT ");
      for(i=1;i<4;i++)
        lcd_putc(buffer[i]);
      output_high(PIN_A3);                 //turn on LCD exit
      motor_right_q(12);
      delay_ms(2000);
      motor_left_q(12);
      lcd_putc("\f   BTS SYSTEM   ");
      lcd_putc("\n TELECOM PROJECT ");
      output_high(PIN_B3);                 //turn on LCD entranc
    }
    else if(buffer[0]=='9')
    {
      output_low(PIN_A3);                  //turn off LCD exit
      lcd_putc("\f  YOUR CREDIT  ");
      lcd_putc("\n    NOT ENOUGH   ");
      output_high(PIN_B3);                 //SHOW IN ENTRANCE
      delay_ms(2000);
      lcd_putc("\f   BTS SYSTEM   ");
      lcd_putc("\n TELECOM PROJECT ");
      output_high(PIN_A3);                 //turn on LCD exit
    }
    else
    {
      lcd_putc("\f  DATA INVALID  ");
      delay_ms(1000);
      lcd_putc("\f   BTS SYSTEM   ");
      lcd_putc("\n TELECOM PROJECT ");
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้