

ระบบควบคุมการใช้บริการรถไฟฟ้าโดยใช้การ์ด RFID

The Electric Train Service Control System using RFID Card



ปฏิญญาฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ระบบควบคุมการใช้บริการรถไฟฟ้าโดยใช้การ์ด RFID**  
**The Electric Train Service Control System using RFID Card**



**ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**  
**สาขาวิชาวิศวกรรมโทรคมนาคม**  
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**  
**ปีการศึกษา 2548**

ผ่านการตรวจชิ้นงานแล้ว  
(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์  
(ลงชื่อ).....ผู้ตรวจ

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมการใช้บริการรถไฟฟ้าโดยใช้การ์ด RFID

The Electric Train Service Control System using RFID Card

ผู้จัดทำ

1. นายชนสรณู แซ่เฮ็ง 46015010

2. นายธีรวิทย์ มานะ 46015012

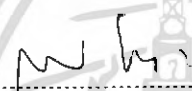
3. นายนพรัตน์ ธรรมวงษา 46015014





อาจารย์ที่ปรึกษา

(ผศ.ดร.พิเชฐ ม่วงนวล)



อาจารย์ที่ปรึกษา

(รศ.สมยศ จุณณะปิยะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบควบคุมการใช้บริการรถไฟฟ้าโดยใช้การ์ด RFID

### The Electric Train Service Control System using RFID Card

โดย นายชนสรณู แซ่เฮ็ง 46015010

นายธีรวิทย์ มานะ 46015012

นายพนรัตน์ ธรรมวงษา 46015014

อาจารย์ที่ปรึกษา ผศ.ดร.พิเชฐ ม่วงนวล

รศ.สมยศ จุณณะปิยะ

#### บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้นำเสนอการพัฒนาการคิดค่าบริการรถไฟฟ้าโดยใช้การ์ด RFID (Radio - Frequency Identification) โดยนำ RFID มาใช้ในการควบคุมการเปิดปิดประตูเข้าออกในการใช้บริการรถไฟฟ้า โดยได้แบ่งการทำงานออกเป็นสามส่วนคือ ส่วนของการรับข้อมูลจากบัตร RFID ทำการอ่านข้อมูลที่บันทึกไว้ ส่วนที่สองเป็นส่วนไมโครคอนโทรลเลอร์ควบคุมการเปิดปิดประตู และส่วนสุดท้ายเป็นส่วนของฐานข้อมูล ซึ่งจะรับข้อมูลจากบัตร สถานีต้นทาง สถานีปลายทาง ชนิดการให้บริการและนำข้อมูลที่ได้ออกมาคำนวณเพื่อสรุปค่าใช้จ่ายการให้บริการเป็นรายเดือน

#### ABSTRACT

This thesis presents the development of RFID technology for service charge of Electric Train. It is to control the entrance of user. This system has three parts, the first part is RFID card and the second part is receiver to receive the data and storage and the final part is control and management for month billing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	2
2.1 RFID คืออะไร	2
2.2 ส่วนประกอบของระบบ RFID	3
2.2.1. ฉลากอิเล็กทรอนิกส์ (Transponder/Tags)	3
2.2.1.1 แท็ก RFID ชนิดที่เป็นพาสซีฟ	4
2.2.1.2 แท็ก RFID ชนิดที่เป็นแอคทีฟ	4
2.2.2. เครื่องอ่าน (Reader)	5
2.3 คลื่นพาหะในระบบ RFID	6
2.4 หลักการทำงานของ RFID	7
2.5 ประเภทของระบบ RFID	8
2.5.1. RFID ที่จำแนกโดยขนาดของหน่วยความจำ	8
2.5.1.1 RFID ชนิด 1 บิต	8
2.5.1.2 RFID ชนิดหน่วยความจำมากกว่า 1 บิต	9
2.5.2. RFID ที่จำแนกโดยลักษณะการคล่องของสัญญาณ	9
2.5.2.1 คัมปลิงแบบปิด	9
2.5.2.2 คัมปลิงแบบรีโมท	10
2.5.2.3 ช่วงระยะไกล	10
2.5.3. RFID ที่จำแนกตามความสามารถของระบบ	11
2.5.3.1 ระบบอ่านอย่างเดียว	11
2.5.3.2 ระบบอ่านเขียน	11
2.5.3.3 ระบบไมโครโปรเซสเซอร์	11
2.6 เทคโนโลยีการเข้ารหัส/ถอดรหัสของระบบ RFID	11
2.6.1. การเข้ารหัส	11
2.6.2. การผสมข้อมูลแบบดิจิทัล	12
2.6.2.1. Amplitude Shift Keying (ASK)	13
2.6.2.2. 2 Frequency Shift Keying (2FSK)	13
2.6.2.3. 2 Phase Shift Keying (2PSK)	13
2.7 สถาปัตยกรรมของ Tags	14
2.7.1 Electronic data carriers	14
2.7.1.1. Tags ที่มีเฉพาะหน่วยความจำอย่างเดียว	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

เรื่อง	หน้า
2.7.1.2. Tags ชนิดไมโคร โปรเซสเซอร์	15
2.8 เทคโนโลยีของหน่วยความจำ	15
2.9 คุณสมบัติของระบบ RFID	16
2.10 พอร์ต RS-232C	17
2.10.1 มาตรฐาน RS-232C	18
2.10.2 ลักษณะของสัญญาณ RS-232C	18
2.10.3 การกำหนดขั้วต่อของพอร์ต RS-232C	19
2.10.4 การเชื่อมต่อมาตรฐาน RS-232C	20
2.11 การสื่อสารแบบ RS422/RS485	22
2.11.1 การเชื่อมต่อแบบ RS422	22
2.11.1.1 การเชื่อมต่อ RS422 แบบฟูลดูเพล็กซ์	22
2.11.1.2 การเชื่อมต่อ RS422 แบบซิมเพล็กซ์	22
2.11.2 การเชื่อมต่อแบบ RS485	23
2.11.2.1 ต่อกันแบบขนานทั้งระบบ	23
2.11.2.2 ต่อกันแบบ จุดต่อจุด	23
2.11.2.3 ระบบเครือข่าย RS-485	24
2.11.2.4 ฮาร์ดแวร์ ของ RS-485	24
2.11.2.5 ซอฟต์แวร์ของ RS-485	25
2.12 สถาปัตยกรรมของฐานข้อมูล	27
2.12.1 เทคโนโลยีสำหรับการเข้าถึงข้อมูลในฐานข้อมูล	29
2.12.2 เทคโนโลยีการเข้าถึงข้อมูลในแนวความคิด	32
2.13 ความรู้พื้นฐานเกี่ยวกับ MySQL	35
2.13.1 สถาปัตยกรรมของ MySQL	36
2.13.2 MySQL กับมาตรฐานภาษา SQL	37
2.13.3 ประเภทข้อมูลและตารางข้อมูลใน MySQL	39
2.13.4 การรักษาความปลอดภัย	48
2.13.5 ระดับสิทธิของผู้ใช้	49
2.13.6 ฐานข้อมูล MySQL	52
<b>บทที่ 3 การออกแบบและการสร้าง</b>	<b>55</b>
3.1 การออกแบบและการสร้างชุดการแปลงพอร์ต RS-232 เป็น RS-485	56
3.1.1 บล็อกการทำงาน	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

เรื่อง	หน้า
3.1.2 การออกแบบวงจร	57
3.1.3 การออกแบบลายวงจร	59
3.2 การออกแบบและการสร้างชุดอ่านบัตร RFID	60
3.2.1 บล็อกการทำงานของชุดอ่านบัตร RFID	60
3.2.2 การออกแบบวงจร	61
3.2.3 ข้อกำหนดในการติดต่อสื่อสารของชุดอ่านบัตร RFID	62
3.2.4 การทำงานของโปรแกรม	63
3.2.5 ฟังก์ชันการทำงานของชุดอ่านบัตร RFID	63
3.2.5.1 ฟังก์ชันการทำงานของโปรแกรมหลัก	63
3.2.5.2 ฟังก์ชันการทำงานของโปรแกรมย่อยกำหนดค่าเริ่มต้น	64
3.2.5.3 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายนอก	65
3.2.5.4 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบ LRC	66
3.2.5.5 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบพาริตี	67
3.2.5.6 ฟังก์ชันการทำงานของโปรแกรมย่อยหา LRC	67
3.2.5.7 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายใหม่เมอร์	68
3.2.5.8 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สาย การส่งข้อมูลอนุกรม	68
3.2.5.9 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สาย การรับข้อมูลอนุกรม	69
3.3 การออกแบบและการสร้างชุดควบคุมการเปิด-ปิดประตูทางเข้าออก	71
3.3.1 บล็อกการทำงานของชุดควบคุมการเปิด-ปิดประตูทางเข้าออก	71
3.3.2 การออกแบบวงจรและการทำงาน	72
3.3.3 ข้อกำหนดในการติดต่อสื่อสารของชุดควบคุมการเปิด-ปิดประตู	73
3.3.4 การทำงานของโปรแกรม	76
3.3.5 ฟังก์ชันการทำงานของชุดควบคุมการเปิด-ปิดประตูทางเข้าออก	77
3.3.5.1 ฟังก์ชันการทำงานของโปรแกรมหลัก	77
3.3.5.2 ฟังก์ชันการทำงานของโปรแกรมย่อยกำหนดค่าเริ่มต้น	78
3.3.5.3 ฟังก์ชันการทำงานของโปรแกรมย่อยแสดงหมายเลขประจำเครื่อง	79
3.3.5.4 ฟังก์ชันการทำงานของโปรแกรมย่อยกำหนดหมายเลขประจำตัวเครื่อง	80
3.3.5.5 ฟังก์ชันการทำงานของโปรแกรมย่อยแสดงความผิดพลาด	81
3.3.5.6 ฟังก์ชันการทำงานของโปรแกรมย่อยแสดงจำนวนเงิน	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

เรื่อง

หน้า

3.3.5.7	ผังการทำงานของโปรแกรมย่อยแสดงหน้าจอ LCD หลัก	83
3.3.5.8	ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายนอก	83
3.3.5.9	ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ต การรับข้อมูลอนุกรม	85
3.3.5.10	ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายใหม่เมอร์	86
3.3.5.11	ผังการทำงานของโปรแกรมย่อยตรวจสอบ LRC	87
3.3.5.12	ผังการทำงานของโปรแกรมย่อยตรวจสอบพาริตี	88
3.3.5.13	ผังการทำงานของโปรแกรมย่อยหา LRC	88
3.3.5.14	ผังการทำงานของโปรแกรมย่อยส่งข้อมูลรหัสบัตร	89
3.3.6	การออกแบบลายวงจร	89
3.4	การออกแบบฐานข้อมูลที่ใช้กับระบบการให้บริการรถไฟฟ้า	90
3.4.1	ตารางที่ใช้ในการ Log On	91
3.4.2	ตารางราคาค่าโดยสารของแต่ละสถานีไปยังสถานีปลายทางใดๆ	92
3.4.3	ตารางบอกชนิดของบัตร	93
3.4.4	ตารางเก็บจำนวนเงินคงเหลือของผู้ใช้บริการแบบเติมเงิน	93
3.4.5	ตารางเก็บยอดเงินรวมในการใช้บริการของผู้ใช้บริการรายเดือน	94
3.4.6	ตารางเก็บข้อมูลที่อยู่เพื่อใช้ในการเก็บค่าใช้บริการของผู้ใช้บริการรายเดือน	94
3.4.7	ตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของผู้ใช้บริการ แบบเติมเงิน	95
3.4.8	ตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของ ผู้บริการรายเดือน	95
3.5	การออกแบบโปรแกรมกำหนดหมายเลขประจำชุดควบคุม การเปิด-ปิดประตูทางเข้าออก	96
3.5.1	ผังการทำงานของโปรแกรมกำหนดหมายเลขประจำชุดควบคุม การเปิด-ปิดประตูทางเข้าออก	96
3.5.2	ผังการทำงานของเหตุการณ์การกดปุ่มออกจากโปรแกรม	97
3.5.3	ผังการทำงานของเหตุการณ์การกดปุ่มเปิดพอร์ต	97
3.5.4	ผังการทำงานของเหตุการณ์การกดปุ่มปิดพอร์ต	98
3.5.5	ผังการทำงานของเหตุการณ์เลือกปุ่มหมายเลข	98
3.5.6	ผังการทำงานของเหตุการณ์การกดปุ่มอำนาจหมายเลข	98
3.5.7	ผังการทำงานของเหตุการณ์การกดปุ่มกำหนดหมายเลข	99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

เรื่อง	หน้า
3.5.8 ฟังก์ชันการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ทัลสื่อสาร	99
3.6 การออกแบบโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า	100
3.6.1 ฟังก์ชันการทำงานของโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า	100
3.6.2 ฟังก์ชันการทำงานของเหตุการณ์การเลือกชนิดผู้ใช้บริการ	101
3.6.3 ฟังก์ชันการทำงานของเหตุการณ์การเลือกการให้บริการต่างๆ	101
3.6.4 ฟังก์ชันการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ทัลสื่อสาร	102
3.7 การออกแบบโปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก	103
3.7.1 ฟังก์ชันการทำงานของโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า	103
3.7.2 ฟังก์ชันการทำงานของเหตุการณ์เลือกปุ่มหมายเลขของเครื่องที่จะใช้งานบนบัส	104
3.7.3 ฟังก์ชันการทำงานของ Thread ที่ใช้คำสั่งอ่านรหัสบัตรตลอดเวลา	104
3.7.4 ฟังก์ชันการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ทัลสื่อสาร	105
<b>บทที่ 4 การทดลองและผลการทดลอง</b>	106
4.1 การทดลองชุดอ่านบัตร RFID	106
4.2 โปรแกรมกำหนดหมายเลข	108
4.3 การตั้งค่าให้กับ ODBC ของ MySQL	111
4.4 การใช้งานโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า	113
4.5 การใช้งานโปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก	115
<b>บทที่ 5 บทสรุปและวิจารณ์</b>	116
<b>ภาคผนวก</b>	
<b>บรรณานุกรม</b>	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปลูกภาพ

	หน้า
รูปที่ 2.1 แสดงวงจรเทียบเคียงของระบบ RFID	2
รูปที่ 2.2 แสดง RFID แท็กในรูปแบบต่างๆ	3
รูปที่ 2.3 แสดงบล็อกไดอะแกรมของแท็ก RFID ชนิดที่เป็นพาสซีฟ	4
รูปที่ 2.4 แสดงรูปตัวอย่างแท็ก RFID ชนิดที่เป็นแอคทีฟที่มีแบตเตอรี่ ลิเทียม 2 ก้อนอยู่ภายนอก	5
รูปที่ 2.5 แสดงโครงสร้างภายในเครื่องอ่าน	5
รูปที่ 2.6 แสดงรูปตัวอย่างเครื่องอ่านแบบต่างๆ	6
รูปที่ 2.7 แสดงความถี่ย่านที่ระบบ RFID ถูกใช้งาน	7
รูปที่ 2.8 แสดงถึงการทำงานของ RFID	7
รูปที่ 2.9 การทำงานและติดต่อสื่อสารระหว่างบัตรกับหัวอ่าน	8
รูปที่ 2.10 แสดงวงจรเทียบเคียงกับปลั๊กแบบปิด	9
รูปที่ 2.11 แสดงวงจรเทียบเคียงของการค้ำปลั๊กแบบรีโมท หรือ การค้ำปลั๊กโดยการเหนี่ยวนำ	10
รูปที่ 2.12 แสดงวงจรเทียบเคียงในช่วงระยะไกล	10
รูปที่ 2.13 แสดงการไหลของสัญญาณและข้อมูลในระบบการสื่อสารแบบดิจิทัล	11
รูปที่ 2.14 แสดงชนิดของรหัสข้อมูลที่เป็นเส้นตรง ที่ใช้ในระบบ RFID	12
รูปที่ 2.15 แสดงใน ASK ความสูงของขอคคลื่นจะเปลี่ยนสถานะอยู่ 2 สถานะ ขึ้นอยู่กับข้อมูลที่ผสมเข้ามา	13
รูปที่ 2.16 แสดง 2FSK จะเปลี่ยนแปลงความถี่ของคลื่นตามข้อมูลที่ผสมเข้ามา	13
รูปที่ 2.17 แสดง PSK จะกลับเฟสทุกครั้งเมื่อมีการเปลี่ยนสถานะของข้อมูล	13
รูปที่ 2.18 ชนิดของแท็ก RFID แบ่งตามสถาปัตยกรรมพื้นฐาน	14
รูปที่ 2.19 บล็อกไดอะแกรม ของแท็ก ชนิดมีหน่วยความจำ	14
รูปที่ 2.20 บล็อกไดอะแกรมของ แท็ก ชนิดไมโคร โปรเซสเซอร์	15
รูปที่ 2.21 การใช้งานพอร์ต RS-232C เชื่อมต่ออุปกรณ์	19
รูปที่ 2.22 การเปลี่ยนแปลงสัญญาณทีทีแอล	21
รูปที่ 2.23 แสดง การต่อสาย RS422 แบบ Full Duplex	22
รูปที่ 2.24 แสดง การต่อสาย RS422 แบบ Simplex	22
รูปที่ 2.25 แสดง การต่อสาย RS422 แบบรับหลายตัวพร้อมกัน	22
รูปที่ 2.26 แสดง การต่อสาย RS-485 แบบขนาน	23
รูปที่ 2.27 แสดง การต่อสาย RS485 แบบ จุดต่อจุด	23
รูปที่ 2.28 แสดงสถาปัตยกรรมแบบ Teleprocessing	24
รูปที่ 2.29 แสดงสถาปัตยกรรมแบบ Client/Server	25
รูปที่ 2.30 แสดงสถาปัตยกรรมแบบ Distributed Data Processing	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 2.31 แสดงการเข้าถึงการใช้เทคโนโลยีการอินเทอร์เน็ตเฟสทำให้โค้ดที่เราเขียนที่ลดลงอย่างมาก	29
รูปที่ 2.32 แสดงการใช้การอินเทอร์เน็ตเฟสระหว่างแอปพลิเคชันกับฐานข้อมูล	30
รูปที่ 2.33 แสดงสถาปัตยกรรมของ DAO	30
รูปที่ 3.34 แสดงสถาปัตยกรรมของ UDA	32
รูปที่ 2.35 แสดงสถาปัตยกรรมของ ODBC	33
รูปที่ 2.36 แสดงสถาปัตยกรรมของ OLE DB	33
รูปที่ 2.37 แสดงสถาปัตยกรรมของ ADO	34
รูปที่ 3.1 โครงสร้างการเชื่อมต่อระหว่างชุดอ่านบัตร RFID กับคอมพิวเตอร์	55
รูปที่ 3.2 โครงสร้างการเชื่อมต่อระหว่างชุดควบคุมการเปิด-ปิดประตูทางเข้าออก	56
รูปที่ 3.3 บล็อกการทำงานของชุดการแปลงพอร์ต RS-232 เป็น RS-485	56
รูปที่ 3.4 วงจรชุดการแปลงพอร์ต RS-232 เป็น RS-485	57
รูปที่ 3.5 ลายวงจรและการวางอุปกรณ์ชุดการแปลงพอร์ต RS-232 เป็น RS-485	59
รูปที่ 3.6 บล็อกการทำงานของชุดอ่านบัตร RFID	60
รูปที่ 3.7 วงจรของชุดอ่านบัตร RFID	61
รูปที่ 3.8 ผังการทำงานของโปรแกรมหลัก	63
รูปที่ 3.9 ผังการทำงานของโปรแกรมย่อยกำหนดค่าเริ่มต้น	64
รูปที่ 3.10 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตรีปต์ภายนอก	65
รูปที่ 3.11 ผังการทำงานของโปรแกรมย่อยตรวจสอบ LRC	66
รูปที่ 3.12 ผังการทำงานของโปรแกรมย่อยตรวจสอบพาริตี	67
รูปที่ 3.13 ผังการทำงานของโปรแกรมย่อยหา LRC	67
รูปที่ 3.14 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตรีปต์ไทม์เมอร์1	68
รูปที่ 3.15 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตรีปต์การส่งข้อมูลอนุกรม	69
รูปที่ 3.16 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตรีปต์การส่งข้อมูลอนุกรม	70
รูปที่ 3.17 ลายวงจรและการวางอุปกรณ์ของชุดอ่านบัตร RFID	71
รูปที่ 3.18 บล็อกการทำงานของชุดควบคุมการเปิด-ปิดประตูทางเข้าออก	71
รูปที่ 3.19 วงจรชุดควบคุมการเปิด-ปิดประตูทางเข้าออก	72
รูปที่ 3.20 ผังการทำงานของโปรแกรมหลัก	77
รูปที่ 3.21 ผังการทำงานของโปรแกรมย่อยกำหนดค่าเริ่มต้น	78
รูปที่ 3.22 ผังการทำงานของโปรแกรมย่อยแสดงหมายเลขประจำเครื่อง	79
รูปที่ 3.23 ผังการทำงานของโปรแกรมย่อยกำหนดหมายเลขประจำตัวเครื่อง	80
รูปที่ 3.24 ผังการทำงานของโปรแกรมย่อยแสดงความผิดพลาด	81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 3.25 ผังการทำงานของโปรแกรมย่อยแสดงจำนวนเงิน	82
รูปที่ 3.26 ผังการทำงานของโปรแกรมย่อยแสดงหน้าจอ LCD หลัก	83
รูปที่ 3.27 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายนอก	84
รูปที่ 3.28 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายรับข้อมูลอนุกรม	85
รูปที่ 3.29 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายเทอร์โมเมอร์	86
รูปที่ 3.30 ผังการทำงานของโปรแกรมย่อยตรวจสอบ LRC	87
รูปที่ 3.31 ผังการทำงานของโปรแกรมย่อยตรวจสอบพาริตี	88
รูปที่ 3.32 ผังการทำงานของโปรแกรมย่อยหา LRC	88
รูปที่ 3.33 ผังการทำงานของโปรแกรมย่อยส่งข้อมูลรหัสบัตร	89
รูปที่ 3.34 ลายวงจรและการวางอุปกรณ์	90
รูปที่ 3.35 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางการ Log On	91
รูปที่ 3.36 แสดงตารางที่ใช้ในการ Log On เข้าใช้ระบบฐานข้อมูล	91
รูปที่ 3.37 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางราคาค่าโดยสาร	92
รูปที่ 3.38 แสดงตารางราคาค่าโดยสารของแต่ละสถานีไปยังสถานีปลายทางใดๆ	92
รูปที่ 3.39 แสดงตารางราคาค่าโดยสารที่สถานีต้นทาง	93
รูปที่ 3.40 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางบอกชนิดของบัตร	93
รูปที่ 3.41 แสดงตารางบอกชนิดของบัตร	93
รูปที่ 3.42 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางจำนวนเงินคงเหลือของผู้ใช้บริการแบบเติมเงิน	93
รูปที่ 3.43 แสดงตารางเก็บจำนวนเงินคงเหลือของผู้ใช้บริการแบบเติมเงิน	94
รูปที่ 3.44 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางยอดเงินรวมของผู้ใช้บริการรายเดือน	94
รูปที่ 3.45 แสดงตารางเก็บยอดเงินรวมในการใช้บริการของผู้ใช้บริการรายเดือน	94
รูปที่ 3.46 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางข้อมูลที่อยู่เพื่อใช้ในการเก็บค่าใช้บริการของผู้ใช้บริการรายเดือน	94
รูปที่ 3.47 แสดงตารางเก็บข้อมูลที่อยู่เพื่อใช้ในการเก็บค่าใช้บริการของผู้ใช้บริการรายเดือน	94
รูปที่ 3.48 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางสถานะการเข้าออกในการโดยสารของผู้ใช้บริการแบบเติมเงิน	95
รูปที่ 3.49 แสดงตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของผู้ใช้บริการแบบเติมเงิน	95
รูปที่ 3.50 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางสถานะการเข้าออกในการโดยสารของผู้ใช้บริการแบบเติมเงิน	95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	<b>หน้า</b>
รูปที่ 3.51 แสดงตารางเก็บสถานะการเข้าออกในการ โดยสารแต่ละครั้งของผู้ใช้บริการรายเดือน	95
รูปที่ 3.52 ผังการทำงานของโปรแกรมหลัก	96
รูปที่ 3.53 ผังการทำงานของเหตุการณ์การกดปุ่มออกจากโปรแกรม	97
รูปที่ 3.54 ผังการทำงานของเหตุการณ์การกดปุ่มเปิดพอร์ต	97
รูปที่ 3.55 ผังการทำงานของเหตุการณ์การกดปุ่มปิดพอร์ต	98
รูปที่ 3.56 ผังการทำงานของเหตุการณ์เลือกปุ่มหมายเลข	98
รูปที่ 3.57 ผังการทำงานของเหตุการณ์การกดปุ่มอ่านหมายเลข	98
รูปที่ 3.58 ผังการทำงานของเหตุการณ์การกดปุ่มกำหนดหมายเลข	99
รูปที่ 3.59 ผังการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ตสื่อสาร	99
รูปที่ 3.60 ผังการทำงานของโปรแกรมหลัก	100
รูปที่ 3.61 ผังการทำงานของเหตุการณ์การเลือกชนิดผู้ให้บริการ	101
รูปที่ 3.62 ผังการทำงานของเหตุการณ์การเลือกการให้บริการต่างๆ	101
รูปที่ 3.63 ผังการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ตสื่อสาร	102
รูปที่ 3.64 ผังการทำงานของโปรแกรมหลัก	103
รูปที่ 3.65 ผังการทำงานของเหตุการณ์เลือกปุ่มหมายเลขของเครื่องที่จะใช้งานบนบัส	104
รูปที่ 3.66 ผังการทำงานของ Thread ที่ใช้คำสั่งอ่านรหัสบัตรตลอดเวลา	104
รูปที่ 3.67 ผังการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ตสื่อสาร	105
รูปที่ 4.1 แสดงการตั้งชื่อในการเชื่อมต่อ	106
รูปที่ 4.2 แสดงการเลือกพอร์ตในการติดต่อสื่อสาร	106
รูปที่ 4.3 แสดงการเลือกความเร็วในการติดต่อสื่อสาร	107
รูปที่ 4.4 แสดงรหัสบัตรพร้อม Flag และ LRC ที่เป็นรหัสแอสกีที่รับเข้ามาที่คอมพิวเตอร์	107
รูปที่ 4.5 แสดงรหัสบัตรที่ทาบบเข้ามาอีกใบ	108
รูปที่ 4.6 แสดง โปรแกรมที่เปิดขึ้นมา	109
รูปที่ 4.7 แสดงการเปิดพอร์ต	109
รูปที่ 4.8 แสดงการอ่านหมายเลขที่อ่านได้บนจอคอมพิวเตอร์	109
รูปที่ 4.9 แสดงการอ่านหมายเลขที่อ่านได้บนจอ LCD ของชุดควบคุม	
การเปิด-ปิดประตูทางเข้า-ทางออก	110
รูปที่ 4.10 แสดงการกำหนดหมายเลขที่กำหนดถูกต้องบนจอคอมพิวเตอร์	110
รูปที่ 4.11 แสดงการกำหนดหมายเลขที่กำหนดถูกต้องบนจอ LCD ของชุดควบคุม	
การเปิด-ปิดประตูทางเข้า-ทางออก	110
รูปที่ 4.12 การเลือก Data Source ให้กับฐานข้อมูล	111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 4.13 การเลือกฐานข้อมูลที่ใช้ในระบบรถไฟฟ้า	111
รูปที่ 4.14 ตั้งค่าให้สามารถใช้งานภาษาไทยได้	112
รูปที่ 4.15 แสดงผลการทดสอบฐานข้อมูลผ่าน ODBC	112
รูปที่ 4.16 หน้าต่าง Log On เข้าใช้โปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า	113
รูปที่ 4.17 หน้าต่าง Log On เมื่อป้อนรหัสผ่านถูกต้อง	113
รูปที่ 4.18 หน้าต่าง Log On เมื่อป้อนรหัสผ่านไม่ถูกต้อง	113
รูปที่ 4.19 แสดงการใช้งานโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า	114
รูปที่ 4.20 แสดงการเลือกหมายเลขเครื่องของโปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก	115
รูปที่ 4.21 การเปิดพอร์ตของโปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก	115



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 มาตรฐานของการใช้แรงดันไฟฟ้าในสัญญา RS-232C	19
ตารางที่ 2.2 คุณสมบัติโดยย่อของสัญญา RS-232C	21
ตารางที่ 2.3 แสดงคุณสมบัติของ EIA RS-422	26
ตารางที่ 2.4 แสดงคุณสมบัติของ EIA RS-485	26
ตารางที่ 2.5 แสดงคุณสมบัติของ EIA RS-232	27
ตารางที่ 3.1 การส่งข้อมูลของไอซี MAX485CPA	58
ตารางที่ 3.2 การรับข้อมูลของไอซี MAX485CPA	58



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันที่โลกมีการพัฒนาไปอย่างรวดเร็วทำให้มีการแข่งขันที่สูงขึ้นในทุกๆด้านรวมไปถึงการที่ทุกคนก็ต้องการความสะดวกสบายและความรวดเร็วที่เพิ่มมากขึ้นในที่นี้เราจะพูดถึงเรื่องการจราจรซึ่งปัจจุบันได้มีการนำเอาเทคโนโลยี RFID มาใช้อำนวยความสะดวกแต่ก็ยังไม่เพียงพอในการตอบสนองความต้องการของมนุษย์จากความล่าช้าในการรอซื้อตั๋วเดินทาง จากพนักงานหรือเครื่องจำหน่ายตั๋วเดินทางอัตโนมัติเกิดปัญหาในการรอซื้อตั๋วเดินทาง และบางทีขณะที่เราทำการโดยสารรถไฟฟ้าเรามีความจำเป็นที่จะลงที่สถานีอื่นที่ไม่ตรงกับตั๋วเดินทางซึ่งจะทำให้เกิดปัญหาขึ้น

เพื่อเป็นการแก้ไขปัญหาดังกล่าวเราจึงได้นำเทคโนโลยี RFID (Radio Frequency Identification) ซึ่งเป็นระบบบัตรอัจฉริยะที่มีการทำงานด้วยการอาศัยคลื่นแม่เหล็กไฟฟ้าในช่วงความถี่วิทยุสามารถเก็บข้อมูลไว้ภายในบัตรได้ซึ่งช่วยการแก้ไขปัญหาล่าช้าในการซื้อตั๋วเดินทางและเราทำการจัดทำฐานข้อมูลซึ่งจะเก็บข้อมูลในการใช้บริการรถไฟฟ้าและจัดทำเป็นใบเสร็จเก็บเงินรายเดือนบัตรที่ใช้อาจพัฒนาเป็นการใช้บัตรประชาชนได้ ซึ่ง RFID เป็นเทคโนโลยีใหม่ที่กำลังมีบทบาทและความสำคัญเพิ่มขึ้นอย่างรวดเร็ว การประยุกต์เทคโนโลยี RFID มีรูปแบบหลากหลายด้วยจุดประสงค์ที่แตกต่างกันแต่อยู่บนหลักการพื้นฐานเดียวกัน นั่นคือการใช้คลื่นความถี่วิทยุเพื่อการระบุเอกลักษณ์ของวัตถุหรือเจ้าของวัตถุที่ติดป้าย RFID แทนการระบุด้วยวิธีการอื่น ซึ่งวิธีการนี้จะช่วยอำนวยความสะดวกและเพิ่มประสิทธิภาพได้ดีกว่า

#### 1.2 วัตถุประสงค์ของการศึกษา

- 1.เพื่อศึกษาถึงกระบวนการการทำงานของระบบบัตรRFID ซึ่งเป็นบัตรอัตโนมัติ (Auto ID) ชนิดหนึ่ง
- 2.เพื่อนำไปประยุกต์ใช้ในการพัฒนาการใช้บริการรถไฟฟ้ารวมถึงการนำไปประยุกต์ใช้งานกับอุตสาหกรรมอื่นๆต่อไป
- 3.เพื่อศึกษาระบบการเชื่อมต่อระบบRFIDกับ ฐานข้อมูลให้ทำงานร่วมกัน ตรวจสอบสิทธิการใช้งาน เก็บสถิติการใช้งาน และการทำใบเสร็จเก็บเงินรายเดือน

#### 1.3 แนวความคิดในการศึกษา

เพื่อแก้ไขปัญหาดังที่กล่าวมาแล้วเราใช้ RFID Card ในการเข้าใช้บริการรถไฟฟ้าซึ่งมีการบันทึกข้อมูลสิทธิในการใช้บริการไว้ในฐานข้อมูล แล้วเครื่องอ่านจะทำการส่งสัญญาณบอกไมโครคอนโทรลเลอร์ให้เปิดประตูโดยเครื่องอ่านจะมีการบันทึกสถานีที่เข้าและสถานีที่ออกไว้ และทำการเก็บข้อมูลการใช้ที่ได้จากบัตรไว้ในฐานข้อมูล ส่วนของฐานข้อมูลจะนำสถิติการใช้งานมาทำใบเสร็จเพื่อเก็บเงินเป็นรายเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

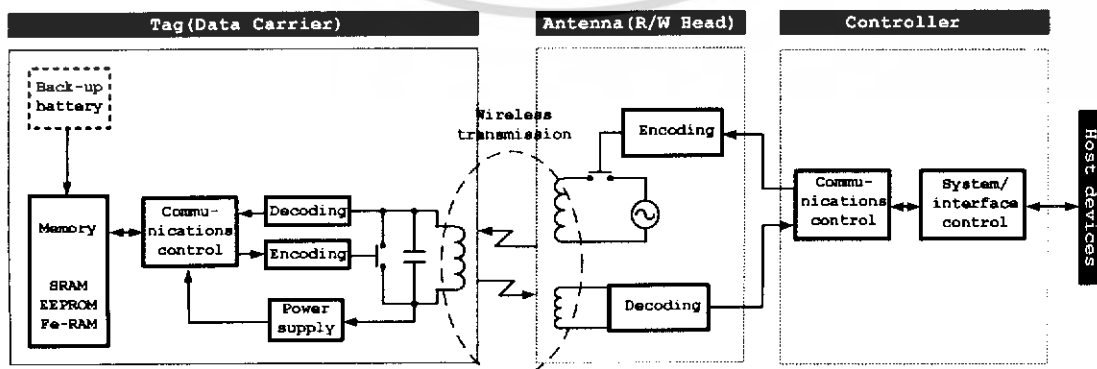
## บทที่ 2 ทฤษฎี

ในปัจจุบันนี้ระบบบ่งชี้อัตโนมัติ (Automatic Identification) หรือ Auto ID ถูกนำมาใช้และพัฒนาไปอย่างมากทั้งในภาคอุตสาหกรรม ลอจิสติกส์ กระบวนการผลิต การขนถ่ายวัตถุดิบ ฯลฯ ระบบ Auto ID จะนำมาใช้ในการจัดเก็บข้อมูล ระบุสถานะของคน สัตว์ สิ่งของ เช่น สินค้า ที่เราให้ความสนใจ

ระบบ Auto ID ที่เป็นที่รู้จักและใช้งานกันอย่างแพร่หลายที่สุดคือ ระบบบาร์โค้ด (Barcode System) ซึ่งสามารถพบเห็นได้ทั่วไปในชีวิตประจำวัน เช่น บนหีบห่อสินค้า หนังสือ หรือบนตัวสินค้า เนื่องจากมีต้นทุนต่อหน่วยที่ต่ำ ง่ายต่อการใช้งานจึงเป็นเหตุผลให้บาร์โค้ด ถูกนำมาใช้งานมากที่สุด แต่บาร์โค้ดก็มีข้อจำกัดหลายประการ ได้แก่ จัดเก็บข้อมูลได้จำกัด เสียหาย หรือมีปัญหาระหว่างการอ่านได้ง่าย ระบบ Auto ID ที่เรารู้จักรองลงมาจากบาร์โค้ดก็คือระบบสมาร์ทการ์ด (Smart Card System) เป็นระบบที่กำลังมีบทบาทอย่างรวดเร็วในปัจจุบัน เราจะพบเห็นสมาร์ทการ์ดในรูปแบบของบัตรต่างๆ เช่น บัตรชมภาพยนตร์ จิมการ์ดของโทรศัพท์เคลื่อนที่ และบัตรสมาชิกตามคลับต่างๆ โดยใช้แถบแม่เหล็ก หรือไมโครชิพในการอ่าน/เขียนข้อมูล มีข้อดีคือสามารถเก็บได้มากปลอดภัย แต่เนื่องจากเป็นแถบแม่เหล็ก วิธีการอ่านข้อมูลจากสมาร์ทการ์ดจะต้องใช้วิธีสัมผัสทำให้เกิดการสึกหรอของเครื่องอ่านเมื่อใช้ไปนานๆ ระบบ Auto ID อีกชนิดหนึ่งที่เราจะกล่าวถึงนี้เป็นระบบที่จัดข้อเสียของทั้งระบบบาร์โค้ดและสมาร์ทการ์ด เราเรียกระบบนี้ว่า **RFID (Radio Frequency Identification)** เป็นระบบ Auto ID ที่ใช้คลื่นแม่เหล็กไฟฟ้าในช่วงคลื่นความถี่วิทยุ เป็นพาหะในการสื่อสารข้อมูล ในเอกสารฉบับนี้จะกล่าวถึงระบบ RFID ตั้งแต่พื้นฐาน โครงสร้างการทำงานและการประยุกต์ใช้งานให้เป็นที่รู้จักกันเสียก่อน

### 2.1 RFID (Radio Frequency Identification System) คืออะไร

RFID จะมีลักษณะใกล้เคียงกับสมาร์ทการ์ดที่สุดคือข้อมูลจะถูกเก็บไว้ในการ์ดหรือ Tags เหมือนกันแต่ข้อแตกต่างที่เห็นได้ชัดคือการอ่าน/เขียนข้อมูลสามารถทำได้โดยไม่ต้องสัมผัสเหมือนกับสมาร์ทการ์ด ซึ่งจะอาศัยคลื่นแม่เหล็กไฟฟ้าในช่วงความถี่วิทยุ (Radio Frequency) ในการติดต่อสื่อสารจึงเป็นสาเหตุให้ RFID ได้เปรียบระบบ Auto ID ทุกชนิดและมีแนวโน้มจะถูกนำมาใช้มากขึ้นเรื่อยๆ



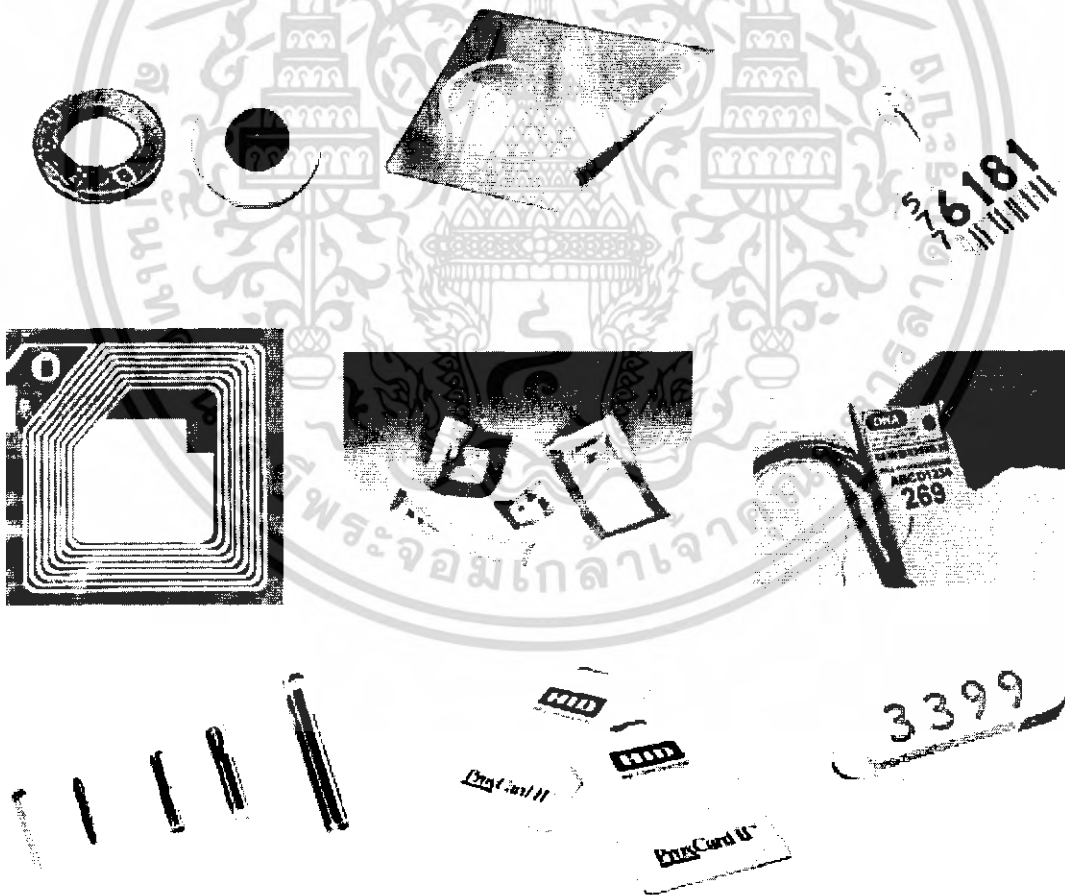
รูปที่ 2.1 แสดงวงจรเทียบเคียงของระบบ RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ส่วนประกอบของระบบ RFID

ในระบบ RFID จะมีองค์ประกอบหลักๆ อยู่ 2 ส่วนด้วยกัน ส่วนแรกคือทรานสปอนเดอร์หรือแท็ก (Transponder/Tag) ที่ใช้ติดกับวัตถุต่างๆ ที่เราต้องการ โดยแท็กที่ว่าจะบันทึกข้อมูลเกี่ยวกับวัตถุนั้นๆ เอาไว้ ส่วนที่สองก็คือเครื่องสำหรับอ่าน/เขียนข้อมูลภายในแท็ก (Interrogator/Reader) ด้วยคลื่นความถี่วิทยุ

**2.2.1. ฉลากอิเล็กทรอนิกส์ (Transponder/Tags)** หรือ Data Carrier จะเป็นส่วนของการเก็บข้อมูลและถูกติดตั้งอยู่กับวัตถุที่เราต้องการชี้แจง (Identify) โดยโครงสร้างภายในของแท็กจะประกอบด้วย 2 ส่วนใหญ่ๆ ได้แก่ ขดลวดขนาดเล็กซึ่งทำหน้าที่เป็นสายอากาศ (Antenna) สำหรับรับส่งสัญญาณคลื่นความถี่วิทยุ และสร้างพลังงานป้อนให้ส่วนของไมโครชิพ (Microchip) ที่ทำหน้าที่เก็บข้อมูลของวัตถุเช่นรหัสสินค้า โดยทั่วไปตัวแท็กอาจอยู่ในชนิดทั้งเป็น กระดาษ แผ่นฟิล์ม พลาสติก มีขนาดและรูปร่างต่างๆ กันไป ทั้งนี้ขึ้นอยู่กับวัสดุที่จะนำไปติด และมีหลายรูปแบบ เช่น ขนาดเท่าบัตรเครดิต เหรียญ กระดุม ฉลากสินค้า แอปซูล เป็นต้น แต่โดยหลักการอาจแบ่งแท็กที่มีใช้งานกันอยู่นั้นจะมีอยู่ 2 ชนิดใหญ่ๆ โดยแต่ละชนิดก็จะมี ความแตกต่างกันในแง่ของการใช้งาน ราคา โครงสร้าง และหลักการทำงานอยู่ ซึ่งจะขอกล่าวถึงและอธิบายแยกเป็นหัวข้อดังนี้

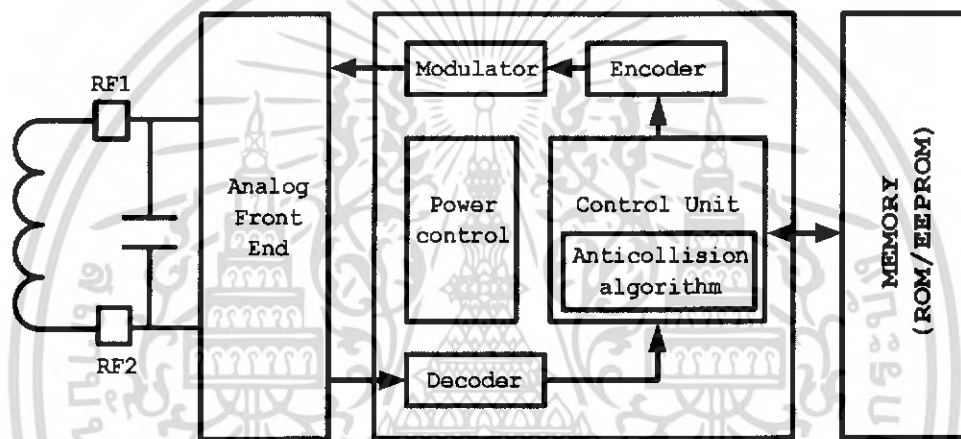


รูปที่ 2.2 แสดง RFID แท็กในรูปแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1.1 แท็ก RFID ชนิดที่เป็นพาสซีฟ

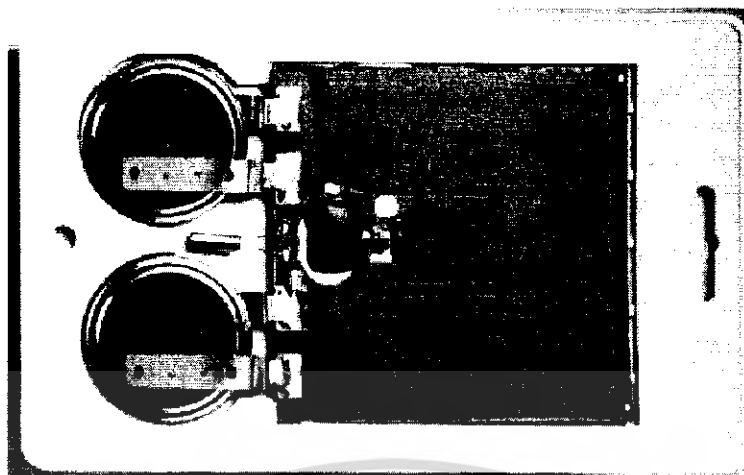
แท็กชนิดนี้ไม่ต้องอาศัยแหล่งจ่ายไฟภายนอกใดๆ เพราะภายในแท็กจะมีวงจรกำเนิดไฟฟ้าเหนี่ยวนำขนาดเล็กเป็นแหล่งจ่ายไฟในตัวอยู่ทำให้การอ่านข้อมูลทำได้ไม่ไกลมากนัก ระยะอ่านสูงสุดประมาณ 1 เมตรขึ้นอยู่กับความแรงของเครื่องส่งและคลื่นความถี่วิทยุที่ใช้ โดยปกติแท็กชนิดนี้มักมีหน่วยความจำขนาดเล็ก โดยทั่วไปประมาณ 16 ถึง 1,024 ไบต์ มีขนาดเล็กและน้ำหนักเบา ราคาต่อหน่วยต่ำ ไอซีของแท็กชนิดพาสซีฟที่มีการผลิตออกมามีทั้งขนาดและรูปร่างเป็นได้ตั้งแต่แท่งหรือแผ่นขนาดเล็กจนแทบไม่สามารถมองเห็นได้ไปจนถึงขนาดใหญ่สะดุดตา ซึ่งต่างก็มีความเหมาะสมกับชนิดงานที่แตกต่างกันโดยทั่วไปโครงสร้างภายในส่วนที่เป็นไอซีของแท็กนั้นก็จะประกอบด้วย 3 ส่วนหลักๆ ได้แก่ ส่วนของควบคุมการทำงานของภาครับส่งสัญญาณวิทยุ (Analog Front-End) ส่วนควบคุมภาคลอจิก (Digital Control Unit) ส่วนของหน่วยความจำ (Memory) ซึ่งอาจจะเป็นแบบ ROM หรือ EEPROM



รูปที่ 2.3 แสดงบล็อกโคอะแกรมของแท็ก RFID ชนิดที่เป็นพาสซีฟ

### 2.2.1.2 แท็ก RFID ชนิดที่เป็นแอคทีฟ

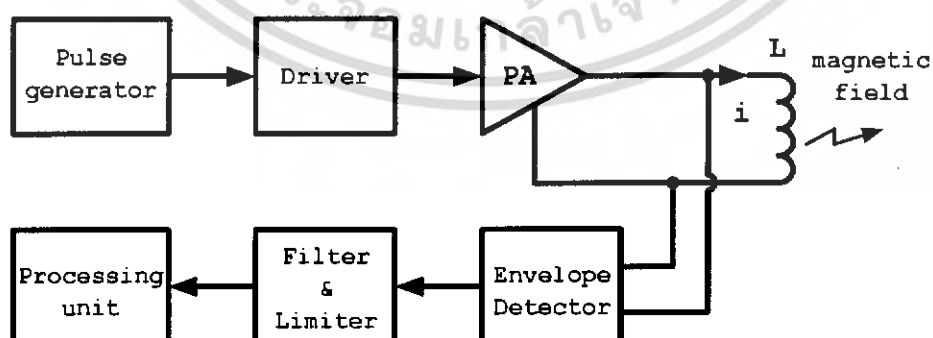
แท็กชนิดนี้จะต้องอาศัยแหล่งจ่ายไฟจากแบตเตอรี่ภายนอก เพื่อจ่ายพลังงานให้กับวงจรภายในทำงาน โดยแท็กแบบนี้สามารถมีหน่วยความจำภายในขนาดใหญ่ได้ถึง 1 เมกะไบต์ และสามารถอ่านได้ในระยะไกลสูงสุดประมาณ 10 เมตร แม้ว่าแท็กชนิดนี้จะมีข้อดีอยู่หลายข้อแต่ก็มีข้อเสียอยู่ด้วยเหมือนกัน เช่น มีราคาต่อหน่วยแพง มีขนาดค่อนข้างใหญ่ และมีระยะเวลาในการทำงานที่จำกัด นอกจากการแบ่งจากชนิดที่ว่ามาแล้วแท็กก็ยังคงแบ่งประเภทจากรูปแบบในการใช้งานได้เป็น 3 แบบ คือ แบบที่สามารถถูกอ่านและเขียนข้อมูลได้อย่างอิสระ (Read-Write), แบบเขียนได้เพียงครั้งเดียวเท่านั้น แต่อ่านได้อย่างอิสระ (Write-Once Read-Many หรือ WORM) และแบบอ่านได้เพียงอย่างเดียว (Read-Only) ด้วย เนื่องจากในการใช้งานจะนิยมใช้แท็กชนิดพาสซีฟมากกว่า



**รูปที่ 2.4** แสดงรูปตัวอย่างแท็ก RFID ชนิดที่เป็นแอคทีฟที่มีแบตเตอรี่ ลิเทียม 2 ก้อนอยู่ภายนอก

แท็กจะประกอบไปด้วย สายอากาศ (Antenna) ทำหน้าที่คล่องสัญญาณที่ส่งมาจากเครื่องอ่านและ ส่วนของไมโครชิพ ในกรณีที่ แท็กไม่มีแบตเตอรี่ในตัวอยู่นอกพื้นที่ที่มีสัญญาณจะไม่มีการทำงานเกิดขึ้น แท็ก จะทำงานก็ต่อเมื่อ แท็ก เข้ามาในพื้นที่ที่มีสัญญาณซึ่ง แท็ก จะได้รับพลังงานจากการคล่อง ของ สัญญาณสนามแม่เหล็กไฟฟ้าและสร้างแรงดันไฟฟ้าขึ้นจำนวนหนึ่ง ปริมาณเพียงพอที่จะใช้ในการทำงาน ของแท็ก

**2.2.2. เครื่องอ่าน (Reader)** จะประกอบไปด้วยภาครับ/ส่งสัญญาณวิทยุ ส่วนควบคุมและ สายอากาศ ทำหน้าที่คล่องสัญญาณกับ แท็กและส่วนของการติดต่อสื่อสารกับอุปกรณ์ควบคุมภายนอก ขึ้นอยู่กับการติดต่อสื่อสารที่เราต้องการ โดยหน้าที่ของเครื่องอ่านก็คือ การเชื่อมต่อเพื่อเขียนหรืออ่าน ข้อมูลลงในแท็ก ด้วยสัญญาณความถี่วิทยุ ภายในเครื่องอ่านจะประกอบด้วย เสาอากาศที่ทำจากขดลวด ทองแดง เพื่อใช้รับส่งสัญญาณ ภาครับและภาคส่งสัญญาณวิทยุ และวงจรควบคุมการอ่าน-เขียนข้อมูล จำพวกไมโครคอนโทรลเลอร์ และส่วนของการติดต่อกับคอมพิวเตอร์ ดังรูปที่ 2.5



**รูปที่ 2.5** แสดงโครงสร้างภายในเครื่องอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**\*หมายเหตุ** เครื่องอ่านในความหมายของเอกสารเล่มนี้ นอกจากสามารถจะอ่านข้อมูลได้แล้วยังสามารถเขียนข้อมูลได้ด้วย

โดยทั่วไปเครื่องอ่านจะประกอบด้วยส่วนประกอบหลักดังนี้

- ภาครับและส่งสัญญาณวิทยุ
- ภาครสร้างสัญญาณพาหะ
- ขดลวดที่ทำหน้าที่เป็นสายอากาศ
- วงจรจูนสัญญาณ
- หน่วยประมวลผลข้อมูล และภาคติดต่อกับคอมพิวเตอร์

โดยทั่วไปหน่วยประมวลผลข้อมูลที่อยู่ภายในเครื่องอ่านมักใช้ ไมโครคอนโทรลเลอร์ ซึ่งอัลกอริทึมที่อยู่ภายในโปรแกรม จะทำหน้าที่ถอดรหัสข้อมูล (Decoding) ที่ได้รับ และทำหน้าที่ติดต่อกับคอมพิวเตอร์ โดยลักษณะ ขนาด และรูปร่างของเครื่องอ่านจะแตกต่างกันไปตามประเภทของการใช้งาน เช่น แบบมือถือขนาดเล็กหรือ คัดสนั่ง จนถึงขนาดใหญ่เท่าประตู (Gate size) เป็นต้น ดังแสดงในรูปที่

2.6



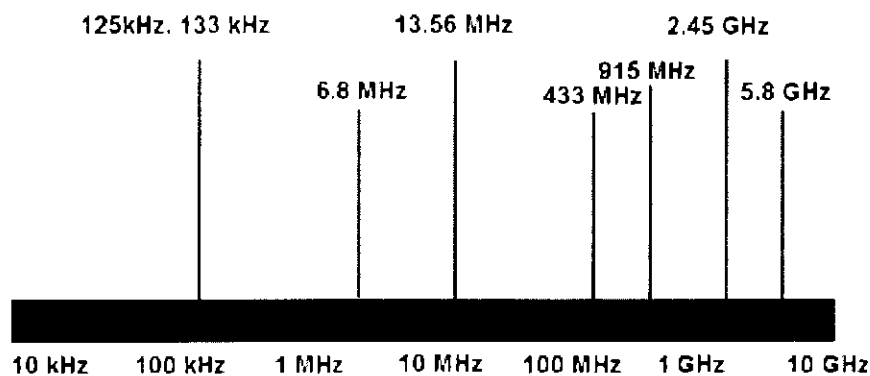
รูปที่ 2.6 แสดงรูปตัวอย่างเครื่องอ่านแบบต่างๆ

### 2.3 คลื่นพาหะในระบบ RFID

ในปัจจุบันคลื่นพาหะที่ใช้งานกันในระบบ RFID จะอยู่ในย่านความถี่ ISM (Industrial-Scientific-Medical) ซึ่งเป็นย่านความถี่ที่กำหนดในการใช้งานในเชิงอุตสาหกรรม วิทยาศาสตร์ และการแพทย์ สามารถใช้งานได้โดยไม่ตรงกับย่านความถี่ที่ใช้งานในการสื่อสารโดยทั่วไป โดยมี 3 ย่านความถี่ใช้งาน คือสำหรับคลื่นพาหะที่ใช้กันในระบบ RFID อาจแบ่งออกได้เป็น 3 ย่านหลักได้แก่

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

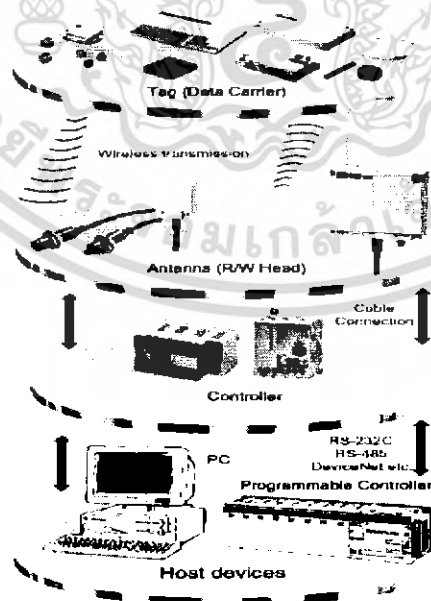


รูปที่ 2.7 แสดงความถี่ย่านที่ระบบ RFID ถูกใช้งาน

ในแง่การใช้งาน 2 ย่านความถี่แรกจะเหมาะสำหรับใช้กับงานที่มีระยะการสื่อสารข้อมูลในระยะใกล้ (LF ระยะอ่านประมาณ 10-20 เซนติเมตร และ HF ระยะอ่านประมาณ 1 เมตร) เช่น การตรวจสอบการผ่านเข้าออกพื้นที่ การตรวจหาและเก็บประวัติในสัตว์ ส่วนย่านความถี่สูงยิ่งจะถูกใช้กับงานที่มีระยะการสื่อสารข้อมูลในระยะไกล (UHF ระยะอ่านประมาณ 1-10 เมตร) เช่น ระบบเก็บค่าบริการทางด่วน และในปัจจุบันระบบ RFID กำลังถูกวิจัยและพัฒนาในย่านความถี่ไมโครเวฟที่ความถี่ 2.4 GHz และความถี่ 5.8 GHz เพื่อใช้งานที่ต้องการระยะอ่านที่ไกลกว่า 10 เมตร เป็นต้น

ในแง่ของราคาและความเร็วในการสื่อสารข้อมูล เมื่อเทียบกันแล้ว RFID ซึ่งใช้คลื่นพาหะย่านความถี่สูงเป็นระบบที่มีความเร็วในการส่งข้อมูลสูงสุดและมีราคาแพงที่สุดด้วยเช่นกัน ส่วน RFID ที่ใช้คลื่นพาหะในอีก 2 ย่านความถี่จะมีระดับราคาและความเร็วลดหลั่นกันไป

#### 2.4 หลักการทำงานของ RFID



รูปที่ 2.8 แสดงถึงการทำงานของ RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



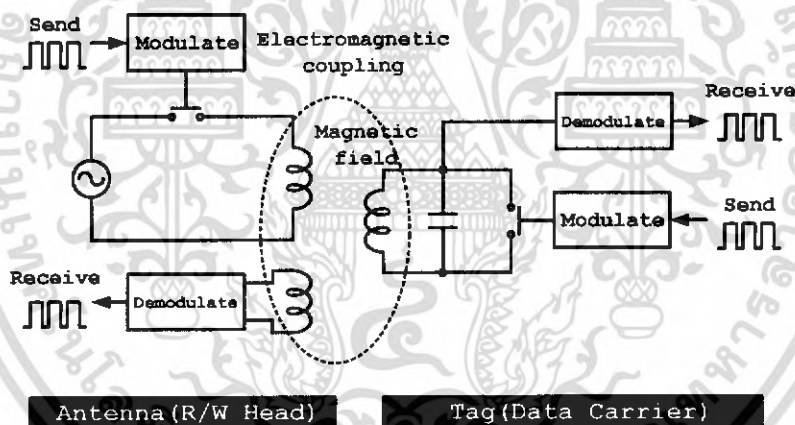
หลังบาร์โค้ดอีกที เครื่องอ่านจะออกแบบเป็นโครงสายอากาศสูงประมาณ 1-1.2 เมตร อยู่ที่ทางออกของร้านค้านั้นๆ เมื่อสินค้าที่มีแท็ก ติดอยู่ผ่านบริเวณเสาอากาศนี้ก็จะถูกตรวจจับได้จากการคลั่งสัญญาณจากสายอากาศและแท็ก จึงสามารถตรวจจับสินค้าที่ถูกนำออกไปได้แท็ก จะถูกดึงออกหรือใช้เครื่องทำลายความเป็นสนามแม่เหล็กไฟฟ้าที่เคาน์เตอร์แคชเชียร์เมื่อสินค้าถูกนำมาชำระเงิน ซึ่งเราจะพบเห็นระบบ EAS มากในซูเปอร์มาร์เก็ตหรือดีสเคาน์สโตร์ต่างๆ เช่น คาร์ฟูร์ โลตัส Big C หรือร้านขายเสื้อผ้า CD ต่างๆ เช่น TEN&CO, แมงป่อง ฯลฯ ช่วงความถี่ที่ใช้งานจะเป็นช่วงความถี่ Microwave เนื่องจากกระบวนการสื่อสารค่อนข้างไกล

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

RFID ชนิดนี้จะใช้แท็ก ที่มีไมโครชิพและหน่วยความจำเป็นส่วนประกอบสำคัญมีราคาสูงกว่าแท็ก ชนิด EAS โดยบางชนิดสามารถเก็บข้อมูลได้สูงสุดถึง 64 กิโลไบต์ ใช้งานอุตสาหกรรมหรืองานทั่วไปที่ต้องใช้ แท็ก ในการเก็บข้อมูล ในเอกสารฉบับนี้จะอ้างอิงถึงระบบ RFID ที่ใช้แท็ก ที่มีหน่วย- ความจำมากกว่า 1 บิตเท่านั้นทั้งทฤษฎีและการนำไปใช้งานยังมีอธิบายในหัวข้อต่อไป

## 2.5.2. RFID ที่จำแนกโดยลักษณะการตั้งของสัญญาณ

### 2.5.2.1 คัปปลิงแบบปิด (Close-Coupling)

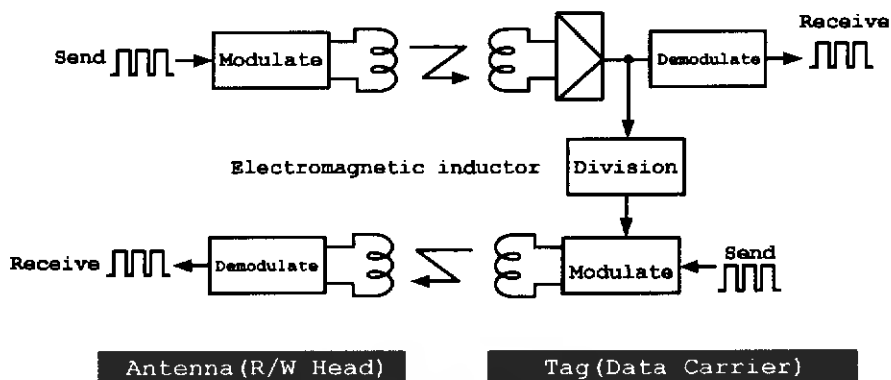


รูปที่ 2.10 แสดงวงจรเทียบเคียงคัปปลิงแบบปิด

เป็น RFID ที่มีระยะในการอ่าน/เขียนข้อมูลสั้นมากประมาณ 0-1 เซนติเมตร ดังนั้น Tags จะต้องอยู่ใกล้หรือวางอยู่บนเครื่องอ่าน Close Coupling นี้จะสามารถใช้คลื่นความถี่ได้ตั้งแต่ 0 Hz จนถึง 50 MHz เนื่องจากการทำงานของแท็กไม่อาศัยการส่งพลังงานจากการแผ่คลื่นแม่เหล็กไฟฟ้าจากเครื่องอ่านแต่อาศัยการเหนี่ยวนำเหมือนหลักการของหม้อแปลงไฟฟ้า ทำให้เกิดพลังงานที่ทำให้วงจรภายในแท็กทำงานได้ ระบบ Close-Coupling จะนิยมมาใช้งานที่ต้องการความปลอดภัยค่อนข้างสูง แต่ไม่ต้องการติดต่อได้ไกล เช่น ประตูอัตโนมัติหรือสมาร์ทการ์ดไร้สัมผัส (Contactless Smart Cards)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

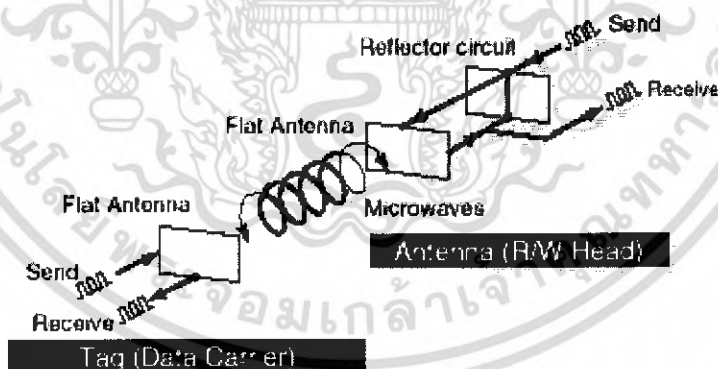
### 2.5.2.2 คัปปลิงแบบรีโมท (Remote Coupling)



รูปที่ 2.11 แสดงวงจรเทียบเคียงของการคัปปลิงแบบรีโมท หรือ การคัปปลิงโดยการเหนี่ยวนำ

เป็นระบบที่มีระยะการอ่าน/เขียนสูงถึง 1 เมตร ระบบนี้จะใช้หลักการคลื่นสัญญาณแบบ การคัปปลิงโดยอาศัยการเหนี่ยวนำของสนามแม่เหล็ก ระหว่างเครื่องอ่านกับแท็กประมาณ 90-95% ของระบบ RFID ในปัจจุบันใช้หลักการคัปปลิงแบบรีโมท นี้โดยความถี่ที่ใช้งานหลายความถี่ตั้งแต่ต่ำกว่า 135 KHz หรือ 13.56 MHz และ 27.125 MHz พลังงานไฟฟ้าจะถูกส่งโดยหลักการแผ่ของคลื่นแม่เหล็กไฟฟ้าไปให้ Tags ทำให้ Tags ได้รับพลังงานสามารถทำงานได้ ระบบคัปปลิงแบบรีโมท นี้จะพบมากในลักษณะงานอุตสาหกรรมเช่น รถยนต์ เครื่องใช้ไฟฟ้า หรือ อิเล็กทรอนิกส์

### 2.5.2.3 ช่วงระยะไกล (Long Range)



รูปที่ 2.12 แสดงวงจรเทียบเคียงในช่วงระยะไกล

ระบบนี้จะมีระยะการอ่าน/เขียนอยู่ระหว่าง 1-10 เมตรหรือบางระบบอาจสูงกว่านี้ ความถี่ที่ใช้ในระบบนี้จะเป็นย่านที่มีความถี่สูงมากหรือ ไมโครเวฟ (Microwave rang) ซึ่งปกติที่ความถี่ 2.45 GHz หรือ บางครั้งจะพบที่ 915 MHz , 5.8 GHz และ 24.125 GHz แต่การส่งพลังงานจากตัวเครื่องอ่านไปยังแท็กทำได้ยากดังนั้นแท็กที่ใช้งานจะเป็นชนิดที่มีแบตเตอรี่ในตัวซึ่งจะใช้สำหรับเป็นไฟเลี้ยงที่ทำให้ไมโครชิพทำงานและเก็บรักษาข้อมูล ลักษณะงานที่พบเห็นจะเป็นลักษณะงานที่ต้องการการสื่อสารระยะไกลเช่น ในกระบวนการผลิตรถยนต์ ระบบชำระเงินอัตโนมัติของทางด่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.3. RFID ที่จำแนกตามความสามารถของระบบ

#### 2.5.3.1 ระบบอ่านอย่างเดียว (Read Only System)

ถือว่าเป็นระบบที่เทคโนโลยีต่ำที่สุด แต่ก็มีข้อมูลซึ่งจะอยู่ในรูปของ ชิรียลนัมเบอร์ และไม่สามารถเขียนข้อมูลใหม่ลงไปได้เหมาะกับงานที่ต้องการอ่านอย่างเดียว เพื่อแยกแยะความแตกต่างของสินค้าหรือบุคคล ระบุชนิดของสินค้า หรือตู้คอนเทนเนอร์มีราคาค่าต่ำกว่า ความถี่ที่ใช้งานจะอยู่ที่ต่ำกว่า 135 KHz หรือ 2.45 GHz

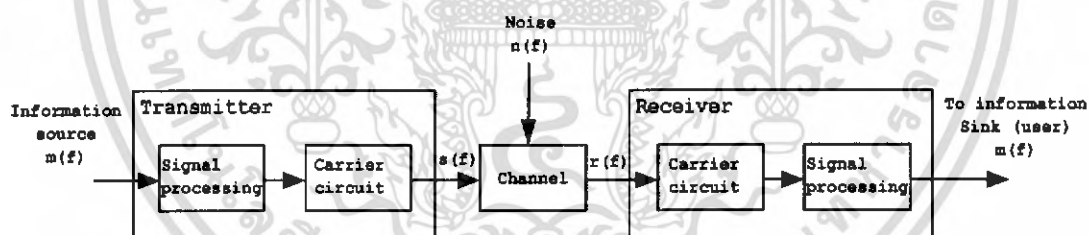
#### 2.5.3.2 ระบบอ่านเขียน (Read-Write System)

จะจัดอยู่ในระยะช่วงกลาง ของระบบ RFID แท้ก็สามารเขียนข้อมูลซ้ำได้โดยความจุจะอยู่ที่ 16 ไบต์จนถึงมากกว่า 16 กิโลไบต์ หน่วยความจำที่ใช้จะเป็นชนิด EEPROM หรือ SRAM ความถี่ที่ใช้งานจะเป็น 135 KHz , 13.56 MHz , 27.125 MHz และ 2.45 GHz

#### 2.5.3.3 ระบบไมโครโปรเซสเซอร์ (Microprocessor System)

ระบบนี้จะจัดอยู่ประเภทเทคโนโลยีสูง เพราะมีตัวไมโครโปรเซสเซอร์เป็นตัวประมวลผลในแท็ก สามารถประยุกต์ใช้งานที่หลากหลายและมีฟังก์ชันการสร้างรหัสลับ (Cryptological Functions) สามารถนำไปใช้งานที่เกี่ยวข้องกับการรักษาความปลอดภัยส่วนมากระบบนี้จะใช้ย่านความถี่ที่ 13.56 MHz หน่วยความจำที่ใช้งานจะมีขนาดตั้งแต่ร้อยละ จนถึง 16 กิโลไบต์ และหน่วยความจำจะเป็นชนิด EEPROM

## 2.6 เทคโนโลยีการเข้ารหัส/ถอดรหัสของระบบ RFID (Coding and Modulation)



รูปที่ 2.13 แสดงการไหลของสัญญาณและข้อมูลในระบบการสื่อสารแบบดิจิทัล

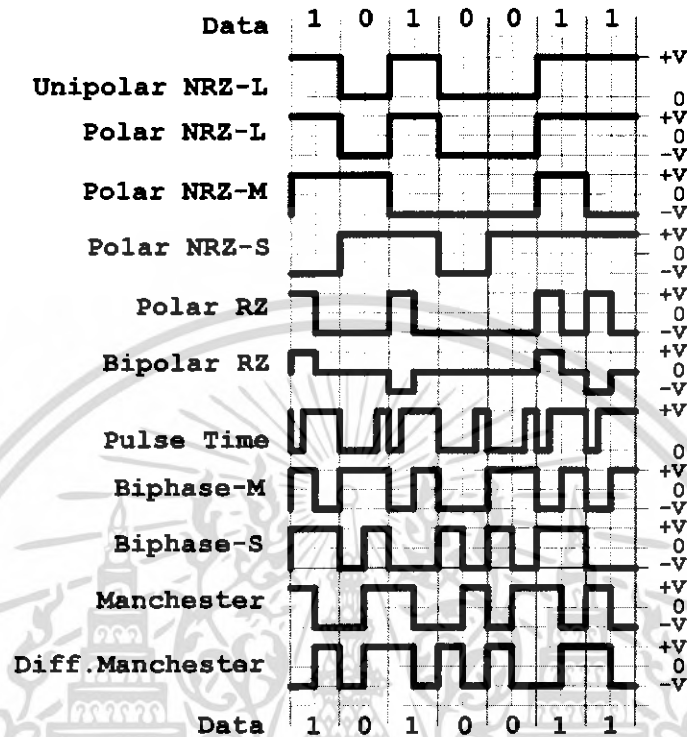
จากรูปที่ 2.13 แสดงกระบวนการสื่อสารแบบดิจิทัลซึ่ง คือการส่งข้อมูลระหว่างเครื่องอ่านกับแท็ก ในระบบ RFID นั้นเอง กระบวนการสื่อสารจะอาศัยหลักการดังนี้ การเข้ารหัสข้อมูล (Signal Coding) การผสมรหัสข้อมูลเข้ากับคลื่นพาหะ (Modulation) การส่งคลื่นสัญญาณออกไป (Transmission) การถอดข้อมูลออกจากคลื่นพาหะ (Demodulation) การแปลงรหัสสัญญาณข้อมูล (Signal decoding)

#### 2.6.1 การเข้ารหัส (Coding)

การส่งข้อมูลแบบตรงไปตรงมาจะทำให้ข้อมูลที่ส่งและรับนั้นมีความยาวเกินไปจึงมีการคิดค้นวิธีการเข้ารหัส ซึ่งการใช้รหัสจะช่วยให้การส่งและรับสั้นสุดลงและไม่ถูกรบกวนจากสิ่งรบกวนภายนอก การเข้ารหัสมักเป็นที่นิยมใช้กันอย่างกว้างขวางในการสื่อสารแบบดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบการสื่อสารแบบดิจิทัลนั้นจะใช้สัญลักษณ์ 0 กับ 1 แทนข้อมูล โดยความแตกต่างของข้อมูลจะเป็นตัวกำหนดค่า 0 กับ 1 ของแต่ละบิตข้อมูล ข้อมูลจะถูกจัดเรียงเป็นแนวนอนหรือเส้นตรง (Line Code) ซึ่งมาตรฐานของรหัสข้อมูลที่เป็นเส้นตรง จะมีหลายมาตรฐานดังนี้



รูปที่ 2.14 แสดงชนิดของรหัสข้อมูลที่เป็นเส้นตรง ที่ใช้ในระบบ RFID

จากรูปที่ 2.14 เราจะพบว่ามาตรฐานของ รหัสข้อมูลที่เป็นเส้นตรง มีอยู่หลายชนิด ความแตกต่างของมันก็คือลักษณะการแสดงค่า 0 กับ 1 ที่ให้ลักษณะสัญญาณแตกต่างกัน Line code ที่พบบ่อยและที่นิยมใช้ก็คือ NRZ และ Manchester Coding

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

**Manchester Coding :** สัญลักษณ์ “1” จะแทนด้วยช่วงขอบล่างของสัญญาณ สัญลักษณ์ “0” จะแทนด้วยช่วงขอบบนของสัญญาณ

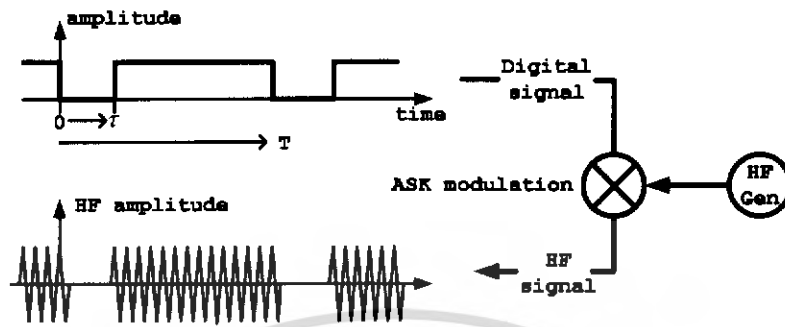
## 2.6.2 การผสมข้อมูลแบบดิจิทัล(Digital Modulation Procedure)

เนื่องจากหลักการพื้นฐานของ RFID คือใช้คลื่นแม่เหล็กไฟฟ้าในการสื่อสารรับส่งข้อมูล ดังนั้นก่อนการสื่อสารจะต้องมีกระบวนการการผสมข้อมูลที่เข้ารหัสไว้แล้วไปกับคลื่นพาหะ (Modulation) แต่เนื่องจากลักษณะข้อมูลเป็นดิจิทัลจึงต้องใช้วิธีการผสมแบบดิจิทัลซึ่งจะแตกต่างจากการผสมแบบอนาล็อกที่ใช้กันอยู่หลายระบบในปัจจุบัน เช่น FM และ AM การผสมข้อมูลแบบดิจิทัลมี 3 วิธีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.6.2.1.) Amplitude Shift Keying (ASK)**

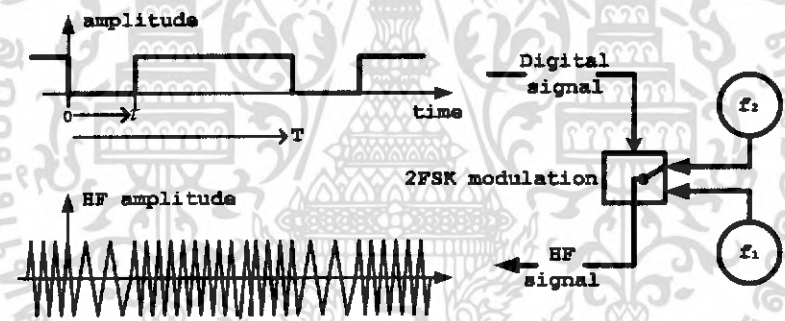
เป็นการผสมข้อมูลโดยอาศัยการเปลี่ยนแปลงของยอดคลื่นพาหะ (Amplitude) เป็นตัวแสดงลักษณะข้อมูลโดยความถี่ของคลื่นพาหะไม่เปลี่ยนแปลงดังรูปที่ 2.15



**รูปที่ 2.15** แสดงใน ASK ความสูงของยอดคลื่นจะเปลี่ยนสถานะอยู่ 2 สถานะขึ้นอยู่กับข้อมูลที่ผสมเข้ามา

**2.6.2.2.) 2 Frequency Shift Keying (2FSK)**

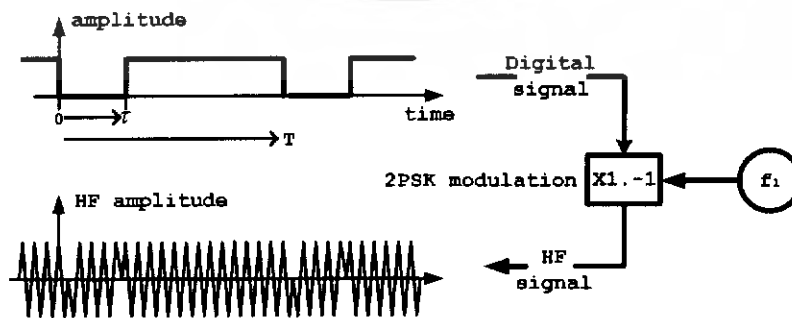
วิธีนี้จะอาศัยหลักการการเปลี่ยนแปลงความถี่ของคลื่นพาหะระหว่าง 2 ความถี่ขึ้นอยู่กับสถานะข้อมูล 0 กับ 1 โดยความสูงของยอดคลื่น (Amplitude) ไม่เปลี่ยนแปลง



**รูปที่ 2.16** แสดง 2FSK จะเปลี่ยนแปลงความถี่ของคลื่นตามข้อมูลที่ผสมเข้ามา

**2.6.2.3.) 2 Phase Shift Keying (2PSK)**

วิธีนี้จะใช้หลักการการเปลี่ยนเฟสของลูกคลื่นเป็นตรงกันข้าม (0 องศา กับ 180 องศา) เมื่อมีการเปลี่ยนสถานะของข้อมูลดังแสดงในรูปที่ 2.17

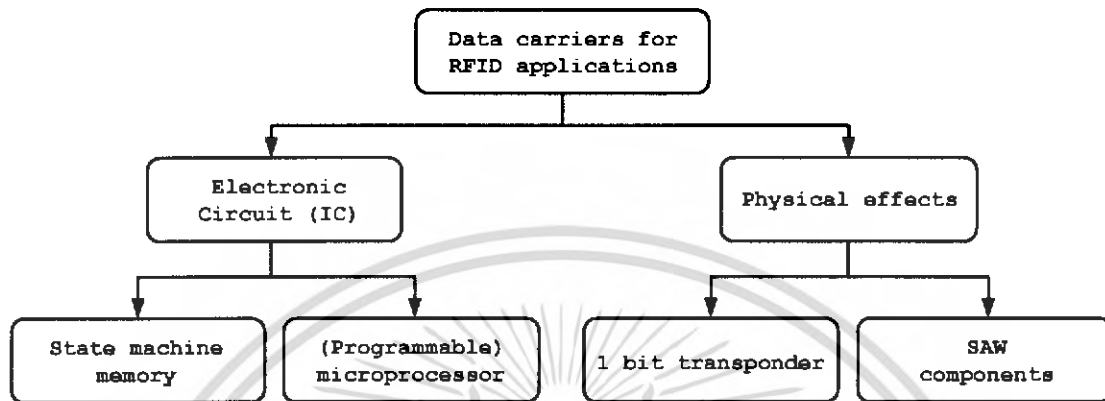


**รูปที่ 2.17** แสดง PSK จะกลับเฟสทุกครั้งเมื่อมีการเปลี่ยนสถานะของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 สถาปัตยกรรมของ Tags

แท็ก หรือ Data Carriers ที่ใช้ในระบบ RFID แบ่งตามโครงสร้างภายในได้เป็นประเภทใหญ่ๆ ได้ 2 ประเภท คือ แท็กแบบใช้วงจรรีเลย์ทรอนิกส์ (Electronic Circuit) กับแบบที่ไม่มีวงจรรีเลย์ทรอนิกส์ หรือชนิดที่ทำหน้าที่ตอบสนองต่อคลื่นสัญญาณเท่านั้น (Physical effect) โดยที่แต่ละชนิดจะแบ่งออกเป็น ชนิดย่อยตามแผนภูมิต่อไปนี้

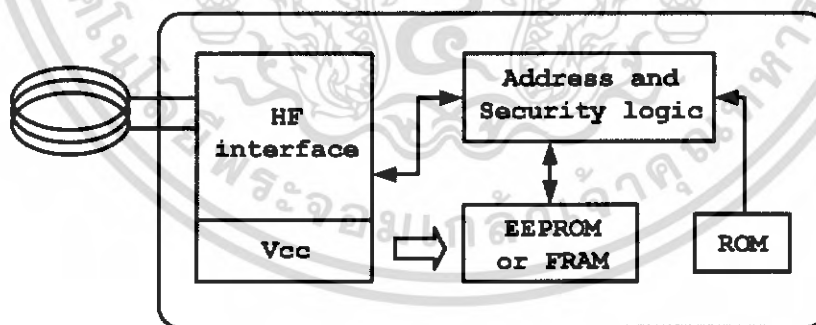


รูปที่ 2.18 ชนิดของแท็ก RFID แบ่งตามสถาปัตยกรรมพื้นฐาน

### 2.7.1 Electronic data carriers

#### 2.7.1.1. Tags ที่มีเฉพาะหน่วยความจำอย่างเดียว (Memory function Transponder)

แท็กชนิดนี้จะมีตั้งแต่ชนิดที่ใช้อ่านอย่างเดียวจนถึงมีความสามารถสูง(High end Transponder) โดยหน่วยความจำที่ใช้จะมีตั้งแต่ RAM, ROM, EEPROM หรือ FRAM และส่วนของ การเชื่อมต่อทาง HF สำหรับสร้างพลังงานเพื่อใช้งานในแท็ก



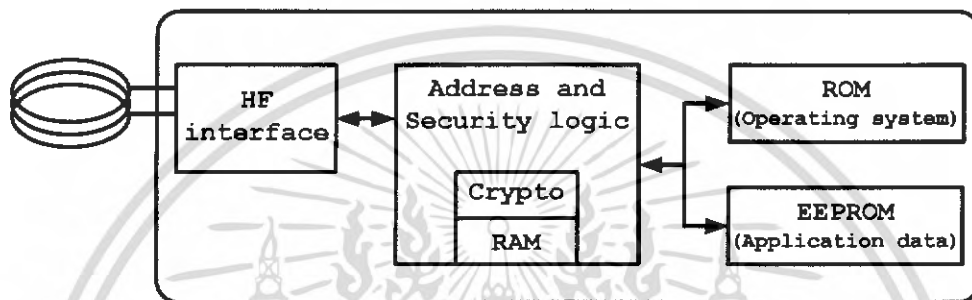
รูปที่ 2.19 บล็อกไดอะแกรมของแท็ก ชนิดมีหน่วยความจำ

จากบล็อกไดอะแกรมในรูปที่ 2.19 ส่วนของ การเชื่อมต่อทาง HF จะมีวงจรถูกตั้งสัญญาณเรดดิไฟเออร์ สัญญาณนาฬิกาและวงจรผสม/ถอดข้อมูล (Modulation/Demodulation) วงจรของการเชื่อมต่อทางความถี่สูง(HF) จะเป็นส่วนที่ใช้ในการติดต่อสื่อสารกับเครื่องอ่านและเป็นวงจร ในการสร้างพลังงานที่เป็นแรงดันไฟฟ้า (VCC) ขึ้นมาเพื่อใช้ในการทำงานของแท็ก ในส่วนของแอดเดรส และซีเคียวริตี้ลอจิก จะเป็นส่วนในการคำนวณทางตรรก (logic) ในส่วนนี้จะมีหัวใจสำคัญคือ สเตทแมชชีน ทำหน้าที่ประมวลผลในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณเหมือนไมโครโปรเซสเซอร์แต่มีขีดจำกัดในการทำงาน ไม่สามารถทำฟังก์ชันที่ซับซ้อนได้ และในส่วนนี้ยังทำหน้าที่ในการติดต่อกับหน่วยความจำเพื่ออ่านหรือเขียนข้อมูลที่ได้รับมาจากเครื่องอ่านได้ นอกจากนี้ยังมีฟังก์ชันรักษาข้อมูลความลับ (Security) โดยสามารถกำหนดรหัสส่วนตัวเพื่อป้องกันไม่ให้ผู้ที่ไม่ได้รับอนุญาตสามารถอ่านข้อมูลจากแท็กได้

### 2.7.1.2. Tags ชนิดไมโครโปรเซสเซอร์ (Microprocessor transponder)

Tags ชนิดนี้จะมีไมโครโปรเซสเซอร์เป็นหัวใจในการประมวลผลสามารถทำฟังก์ชันที่ซับซ้อนได้ ดังนั้นจึงสามารถถูกนำไปประยุกต์ใช้งานที่กว้างขวางและเป็นแท็กชนิดที่ใช้กันมากที่สุดในปัจจุบัน



รูปที่ 2.20 บล็อกไดอะแกรมของแท็ก ชนิดไมโครโปรเซสเซอร์

จากบล็อกไดอะแกรมในรูปที่ 2.20 ส่วนประกอบสำคัญจะมี การเชื่อมต่อทาง HF, CPU และพื้นที่ใช้งาน, Operation system ซึ่งใช้หน่วยความจำ ROM และหน่วยความจำที่ใช้เก็บข้อมูล

## 2.8 เทคโนโลยีของหน่วยความจำ

### RAM (Random Access Memory)

เป็นหน่วยความจำที่ใช้เก็บข้อมูลชั่วคราวข้อมูลจะสูญหายเมื่อไม่มีไฟเลี้ยง ดังนั้นหน่วยความจำชนิดนี้จึงจำเป็นต้องมีแบตเตอรี่สำรองอยู่ในแท็กด้วย ข้อดีของหน่วยความจำชนิดนี้คือสามารถเก็บข้อมูลได้มาก เวลาในการอ่าน/เขียนข้อมูลสั้น แท็กที่ใช้หน่วยความจำที่ ต้องมีแบตเตอรี่จะเรียกอีกชื่อหนึ่งว่า แท็กชนิดที่เป็นแอคทีฟ

### EEPROM (Electric Erasable PROM)

เป็นหน่วยความจำที่ไม่ต้องใช้แบตเตอรี่ในการสำรองไฟเพื่อรักษาข้อมูล เพราะภายในวงจรจะมีคาปาซิเตอร์ ชนิดพิเศษทำหน้าที่ชาร์ตประจุไฟฟ้าให้กับวงจรของหน่วยความจำซึ่งมีอายุยาวนานถึง 10 ปี จำนวนครั้งในการเขียนข้อมูลมากกว่า 100,000 ครั้ง แต่เวลาในการอ่าน/เขียนข้อมูลนานกว่า RAM และความจุที่น้อยกว่า RAM ด้วย

## FRAM (Ferroelectric RAM)

EEPROM มีข้อเด่นคือเป็นหน่วยความจำที่ไม่ต้องมีแบตเตอรี่แต่ข้อเสียคือใช้เวลาในการเขียนข้อมูลมากและใช้พลังงานมากในการเขียนข้อมูล FRAM จึงถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาข้อบกพร่องของ EEPROM คือสามารถเขียนข้อมูลได้เร็วกว่าใช้พลังงานน้อยกว่า การเขียนข้อมูลซ้ำได้มากกว่า 1,000,000,000 ครั้งและไม่ต้องมีแบตเตอรี่เหมือนกัน Tags ที่ใช้หน่วยความจำที่ไม่ต้องมีแบตเตอรี่เรียกว่า Passive Tags

## 2.9 คุณสมบัติของระบบ RFID

จะอธิบายถึงคุณสมบัติของระบบ RFID ซึ่งอาศัยพื้นฐานของการสื่อสารแบบในวิทยุเป็นเกณฑ์สำคัญ

### 1. อ่าน/เขียนโดยไม่ต้องสัมผัส (Contact less)

จุดเด่นข้อแรกของระบบ RFID คือ เครื่องอ่านกับ Tags สามารถสื่อสารกันได้โดยไม่ต้องสัมผัส ทำให้ไม่เกิดส่วนของการสึกหรอเหมือนการ์ดแถบแม่เหล็ก ทำให้ต้นทุนในการดูแลรักษาต่ำ อายุการใช้งานยาวนานสะดวกรวดเร็วในการใช้งาน

### 2. ทนต่อสภาพแวดล้อมและสิ่งสกปรก

ปัญหาที่เป็นอุปสรรคในการอ่าน/เขียนข้อมูลในระบบ Auto ID ที่แก้ไขลำบากก็คือ สภาพแวดล้อมในการใช้งาน เช่น ในโรงงานอุตสาหกรรมมีทั้งฝุ่นละออง น้ำมัน ระบบ Auto ID ที่มีปัญหามากที่สุดคือ ระบบบาร์โค้ด เพราะถ้าแถบบาร์โค้ดสกปรกหรือฉีกขาดก็จะไม่สามารถอ่านข้อมูลได้ หรือถ้าหน้าจอของตัวอ่านสกปรกก็มีปัญหาในการอ่านอีกเช่นกัน แต่ด้วยลักษณะเทคโนโลยีของ RFID ที่ใช้คลื่นความถี่วิทยุเป็นพาหะนำข้อมูลไปจะพบว่าปัญหาดังกล่าวจะไม่มีผลกระทบต่อระบบ RFID เลย 100% ดังนั้น RFID จึงเป็นอุปกรณ์ Auto ID ที่เหมาะสมที่สุดสำหรับการใช้งานในโรงงานอุตสาหกรรม

### 3. สามารถอ่าน/เขียนข้อมูลได้สะดวก

มีระบบ Auto ID น้อยชนิดที่สามารถอ่าน/เขียนข้อมูลได้สะดวกหรือบางระบบต้องใช้เครื่องอ่าน/เขียนแยกกันต่างหาก เช่น บาร์โค้ด ต้องมีเครื่องพิมพ์และเครื่องอ่านแยกกัน สมาร์ทการ์ดต้องนำแท็กมาสัมผัสกับวงจรอ่าน/เขียนโดยตรง แต่ระบบ RFID ตัวอ่านกับตัวเขียนข้อมูลจะอยู่ในตัวเดียวกันเพียงเปลี่ยนโหมดโดยใช้ซอฟต์แวร์เท่านั้น จึงเหมาะสำหรับงานที่ต้องอ่าน และเปลี่ยนแปลงข้อมูลอยู่ตลอดเวลา

### 4. ติดสารได้ทุกทิศทาง

เนื่องจากคุณสมบัติของคลื่นแม่เหล็กไฟฟ้าการอ่าน/เขียนในระบบ RFID จึงไม่ต้องคำนึงถึงทิศทางว่า Tags จะต้องอยู่ตรงหน้ากับเครื่องอ่านเสมอ แท็กสามารถอยู่ด้านหลัง ด้านข้าง หรือแม้กระทั่งถูกทับอยู่ แต่ถ้าเข้ามาอยู่ในพื้นที่สัญญาณแล้วก็จะสามารถอ่าน/เขียนข้อมูลได้ตามปกติ

## 5. Tags สามารถนำกลับมาใช้ใหม่ได้

ด้วยลักษณะ โครงสร้างและความสามารถในการเขียนข้อมูลซ้ำได้ทำให้แท็กสามารถนำกลับมาใช้ ในกระบวนการผลิตได้มากกว่า 100,000 ครั้งต่อ 1 แท็ก คุณสมบัติข้อนี้เป็นจุดแข็งอีกจุดหนึ่งที่ระบบ Auto ID ชนิดอื่นไม่สามารถทำได้

## 6. แท็ก RFID มีหลากหลายแบบให้ประยุกต์ใช้งาน

แท็กของระบบ RFID นั้นจะถูกออกแบบให้มีรูปร่างขนาด โครงสร้าง ความจุของหน่วยความจำ และลักษณะการใช้งานที่แตกต่างกันออกไป เช่น มีลักษณะเป็นสมาร์ทการ์ด กระดุม เหรียญ ทรงสี่เหลี่ยม หรือแม้กระทั่งเป็นแผ่นบาง ๆ เพื่อให้ผู้ใช้งานสามารถเลือกใช้ตามความต้องการ

## 7. ความสามารถในการทะลุทะลวงของสัญญาณ

คลื่นแม่เหล็กไฟฟ้าสามารถทะลุผ่านวัตถุที่ไม่ใช่โลหะหรือมีโลหะเป็นส่วนผสมอยู่ได้ เช่น พลาสติก ไม้ กระจก ไม้ ปูนซิเมนต์ ฯลฯ ดังนั้น Tags จึงสามารถถูกติดตั้งแบบฝัง หรือซ่อนลงไปใต้อุปกรณ์ที่เราต้องการได้ เช่น เราจะพบเห็นการฉีก RFID ที่มีลักษณะเป็นแท่งแก้วเล็ก ๆ เข้าไปในตัวสัตว์ การฝัง แท็กลงบนพื้นในระบบ AGV (Automatic Guide Vehicle)

## 8. ระยะการอ่านระยะไกล

ระยะในการอ่าน/เขียนข้อมูลของระบบ RFID นั้นทำได้ตั้งแต่ 0-10 เมตร ซึ่งถือว่าไกลที่สุดในบรรดา ระบบ Auto ID ที่ใช้งานอยู่ในปัจจุบันนี้ ทั้งนี้ระยะในการอ่าน/เขียนข้อมูลจะขึ้นอยู่กับกำลังส่งของ สายอากาศและช่วงความถี่ที่ใช้งาน สำหรับกำลังส่งของสายอากาศนั้นจะถูกกำหนดโดยกฎหมายของแต่ละ ประเทศทำให้ RFID ที่ผลิตในบางประเทศมีระยะในการอ่าน/เขียนต่างกันทั้งที่ความถี่ใช้งานเท่ากัน

## 9. หน่วยความจำขนาดใหญ่

หน่วยความจำที่ใช้ในระบบ RFID มีตั้งแต่ขนาด 1 บิต (EAS) จนถึงมากกว่า 8 กิโลไบต์ หน่วยความจำที่เป็น RAM จะสามารถเก็บข้อมูลได้มากกว่าหน่วยความจำ แบบ 16 บิต ในกระบวนการปฏิบัติงานสามารถบันทึกลงในแท็กได้ทั้งกระบวนการ หรือแม้กระทั่งข้อมูลส่วนบุคคลก็สามารถบันทึก ลงในแท็กได้

## 10. อ่าน/เขียนข้อมูลได้ครั้งละมากกว่า 1 แท็กพร้อมกัน

เมื่อแท็ก เข้ามาอยู่ในพื้นที่สัญญาณมากกว่า 1 แท็กพร้อมกันเครื่องอ่านสามารถอ่านข้อมูล ซึ่งมาพร้อมกันได้ทั้งหมดหรือจะสามารถเลือกอ่านเฉพาะ แท็กที่ระบุก็ได้

## 11. สามารถอ่าน/เขียนข้อมูลขณะวัตถุกำลังเคลื่อนที่

เครื่องอ่านกับ แท็กสามารถสื่อสารกันได้แม้ขณะฝ่ายใดฝ่ายหนึ่ง กำลังเคลื่อนที่โดยความเร็วของการเคลื่อนที่ขึ้นอยู่กับชนิดของการสื่อสาร หน่วยความจำและปริมาณข้อมูลที่ใช้อ่าน/เขียน

### 2.10 พอร์ต RS-232C

โดยปกติคอมพิวเตอร์จะมีพอร์ตเป็นอนุกรมเรียกชื่อกันว่า RS-232C อยู่ในตัวเองอยู่แล้ว เครื่องคอมพิวเตอร์ที่ไม่มีพอร์ตอนุกรม เช่น คอมพิวเตอร์ไอบีเอ็ม เป็นต้น จำเป็นจะต้องมีการ์ดโดยเรียกว่า ตัวปรับต่ออะซิงโครนัส (Asynchronous Communication Adapter) มาเสียบใส่พอร์ต RS-232C นี้ทำหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปะ หรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับและส่งข้อมูลในแบบอนุกรมเรียกว่า Universal Asynchronous Adapter เหตุที่มีชื่อเรียกว่าพอร์ต RS-232C ก็เนื่องจากสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ของประเทศอเมริกา (Electronic Industries Association : EIR) ได้กำหนดตามมาตรฐานอุปกรณ์การสื่อสารแบบอนุกรม ใ้ภายในได้ชื่อว่าพอร์ต RS-232C และมาตรฐานการส่งข้อมูลแบบอนุกรมได้มีหลายมาตรฐาน แต่ที่นิยมกันมากที่สุดสำหรับคอมพิวเตอร์ก็คือ พอร์ต RS-232C หน้าที่สำคัญของการสื่อสารแบบ อะซิงโครนัส คือ

### 1) รับผิดชอบของพอร์ต RS-232C

1. เปลี่ยนสัญญาณเข้ามาแบบอนุกรมให้เป็นแบบขนาน
2. ตรวจสอบความผิดพลาดของสัญญาณที่รับ
3. ตัดบิตหยุดและพาริตีบิตออก
4. ส่งสัญญาณให้โปรเซสเซอร์รู้ว่ารับสัญญาณไว้แล้ว

### 2) ฝั่งสัญญาณของพอร์ต RS-232C

1. เปลี่ยนสัญญาณแบบขนานจากโปรเซสเซอร์ส่งออกเป็นแบบอนุกรม
2. เพิ่มบิตหยุดและพาริตีบิต
3. เพิ่มสัญญาณควบคุมโมเด็มที่ต่อเชื่อม (ถ้ามี)

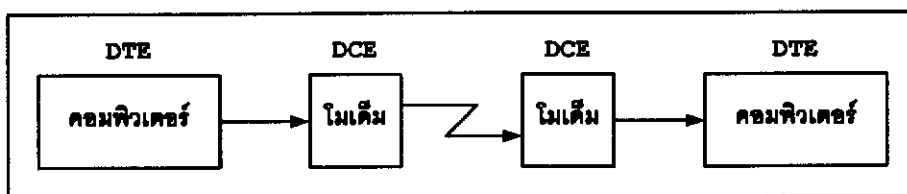
#### 2.10.1 มาตรฐาน RS-232C

มาตรฐาน RS-232C ได้จัดพิมพ์ขึ้นเมื่อปี ค.ศ 1969 โดยสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์แห่งประเทศไทย RS ย่อมาจาก Recommended Standard และส่วน 232 จะเป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ตัวอักษร C คือหมายเลขของฉบับท้ายสุดของมาตรฐาน จุดประสงค์ของมาตรฐานตัวนี้ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment : DTE) กับรูปแบบอุปกรณ์สื่อสารข้อมูล(Data Communication Equipment : DCE) สำหรับผู้ใช้คอมพิวเตอร์ DTE หมายถึง ตัวคอมพิวเตอร์ ส่วน DCE หมายถึงโมเด็ม อุปกรณ์อื่นๆ เช่น เครื่องพิมพ์ที่รับสัญญาณแบบอนุกรมอาจจะใช้ได้ทั้ง DTE และ DCE ขึ้นอยู่กับผู้ผลิต “การเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทางซึ่งต่อไปนี้จะขอเรียกว่า DTE รูปแบบอุปกรณ์สื่อสารข้อมูลซึ่งต่อไปนี้จะขอเรียกว่า DCE โดยข้อแตกต่างของ DTE และ DCE จะเห็นได้จากรูปที่ 2.21 จากภาพนี้จะเห็นได้ว่า RS-232C มีส่วนสำคัญสำหรับการสื่อสารข้อมูลระหว่างคอมพิวเตอร์ ความเร็วและระยะทางของการเชื่อมต่อพอร์ตอนุกรม RS-232C สามารถเชื่อมต่อ การถ่ายโอน ข้อมูลได้จาก 0-20,000บิตต่อวินาทีซึ่งเพียงพอสำหรับคอมพิวเตอร์ที่มีขนาดอัตราบอด 110 ถึง 9600 บอด ความยาวของสายเชื่อมต่อโดยสัญญาณตามมาตรฐานของพอร์ตอนุกรม RS-232C จำกัดแค่ 50 ฟุต ซึ่งพอสำหรับการสื่อสารคอมพิวเตอร์กับอุปกรณ์รอบนอก

#### 2.11.2 ลักษณะของสัญญาณ RS-232C

เพื่อเป็นหลักประกันว่าข้อมูลส่งออกไปยังอุปกรณ์จะถูกควบคุมอย่างถูกต้องจำเป็นต้องมีข้อตกลงกันในเรื่องของสัญญาณที่ใช้มาตรฐานของพอร์ตอนุกรม RS-232C และกำหนด ย่านของแรงดันไฟฟ้าในสัญญาณ ดังแสดงในตารางที่ 2.1 สำหรับคอมพิวเตอร์บางเครื่อง ใช้แต่สัญญาณลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 การใช้งานพอร์ต RS-232C เชื่อมต่ออุปกรณ์

ออกมาเป็นสัญญาณของพอร์ตอนุกรม RS-232C อย่างเช่น ตัวปรับต่ออะซิงโครนัสของคอมพิวเตอร์ ไอพีเอ็ม กรณีเช่นนี้ระยะทางของสายที่เชื่อมต่ออาจจะสั้นกว่า 50 ฟุต ซึ่งคั้งที่กล่าวเอาไว้เนื่องจากระดับของกราวด์จะเปลี่ยนแปลงไปอันเนื่องจากการสูญเสียไปในความต้านทานของสาย ผู้ที่เคยใช้คอมพิวเตอร์ ไอพีเอ็มอาจประสบกับปัญหาเมื่อต่อสัญญาณ RS-232C เกินกว่า 10 ฟุต แล้วใช้การไม่ได้แต่อย่างไรก็ตามพอร์ต RS-232C ของคอมพิวเตอร์ ไอพีเอ็มยังมีโอกาสให้เลือกใช้กระแสไฟฟ้า 20 มิลลิแอมแปร์ เป็นกระแสไฟฟ้าในการทำงาน

### ตารางที่ 2.1 มาตรฐานของการใช้แรงดันไฟฟ้าในสัญญาณ RS-232C

มาตรฐานของการให้แรงดันไฟฟ้า			
แรงดันไฟฟ้า	สถานภาพลอจิก	สถานภาพสัญญาณ	ฟังก์ชันในการควบคุม
บวก	0	SPACE	ON
ลบ	1	MASK	OFF

#### 2.10.3 การกำหนดขั้วต่อของพอร์ต RS-232C

ในทางฟิสิกส์มาตรฐานของพอร์ต RS-232C กำหนดขั้วต่อแบบ DB-25 ผู้ผลิตคอมพิวเตอร์อาจใช้ขั้วต่อชนิดอื่นที่นอกเหนือจาก DB-25 ยกตัวอย่าง เช่น Fujitsu F-8 IBM AT, IBM Jr เป็นต้น หัวต่อตัวเมียของขั้วต่อจะอยู่ที่ตัวโมเด็มขณะที่หัวต่อตัวผู้อยู่ที่ตัวปรับต่อ อะซิงโครนัสหรือที่ตัวคอมพิวเตอร์ อย่างไรก็ตามผู้ผลิตหลายรายไม่ได้ทำตามกฎเกณฑ์ที่กำหนดไว้ซึ่งสัญญาณต่างๆ ทำหน้าที่ดังนี้

**1. Transmit Data (TD ขาที่ 2)** เป็นสัญญาณที่ส่งออกจาก DTE ไปยังโมเด็มหรือต่อเข้าโดยตรงกับคอมพิวเตอร์ตัวอื่นหรือเครื่องพิมพ์ เมื่อไม่มีสัญญาณส่งออกสถานภาพของลอจิกที่ขาที่ 2 นี้จะมีค่าเท่ากับ "1" เทียบเท่าบิตหยุด

**2. Receive Data (RD ขาที่ 3)** เป็นทางของสัญญาณเข้าไปยังอุปกรณ์ภายนอกหรือคอมพิวเตอร์ เมื่อไม่ได้รับสัญญาณเข้ามานี้จะมีสถานภาพทางลอจิกเป็น "1"

**3. Request To Send (RTS ขาที่ 4)** ใช้สำหรับส่งสัญญาณไปยังโมเด็มหรือเครื่องพิมพ์ ซึ่งเป็นการเรียกร้องที่จะส่งสัญญาณมาทางขา 2 สัญญาณนี้ใช้คู่กับสัญญาณ CTS อุปกรณ์รับหากได้สัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณ ได้หรือยัง หากพร้อมที่จะรับก็จะส่งสัญญาณออกไปที่สาย CTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4. Clear To Send (CTS ขาที่ 5)** คังอธิบายไว้ในสัญญาณ RTS เมื่อสัญญาณนี้อยู่ในสถานะปิด (Negative Voltage หรือลอจิก “1”) หมายความว่าอุปกรณ์รับส่งบอกว่าพร้อมที่จะรับข้อมูลแล้ว

**5. Data Set Ready (DSR ขาที่ 6)** เมื่อสัญญาณสายนี้อยู่ในสถานะเปิด หรือ ลอจิก “0” เป็นการบอกคอมพิวเตอร์หรือฝ่ายส่งว่าโมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้วและพร้อมที่จะส่งได้แล้ว โมเด็มที่มีการหมุนอัตโนมัติจะส่งสัญญาณสายนี้ไปบอกให้คอมพิวเตอร์รู้ว่าต่อโทรศัพท์ได้สำเร็จแล้ว

**6. Signal Ground (SG ขาที่ 7)** SG ทำหน้าที่เป็นระดับแรงดันไฟฟ้าอ้างอิงสำหรับทุกๆ สายสัญญาณจะมีแรงดันไฟฟ้าเป็น “0” เมื่อเทียบกับสัญญาณตัวอื่น

**7. Carrier Detect (CD ขาที่ 8)** โมเด็มจะส่งสัญญาณที่อยู่สถานะเปิด ลอจิก “0” ไปบอกคอมพิวเตอร์เมื่อได้รับสัญญาณจากโมเด็มของอีกฝ่ายหนึ่ง สัญญาณนี้จะนำไปจุดไดโอดเปล่งแสงจะบอกได้ว่าได้รับสัญญาณจากโมเด็มอีกฝ่ายหนึ่งแล้ว ไดโอดเปล่งแสงจะอยู่บนหน้าปิดของโมเด็มเอง

**8. Data Terminal Ready (DTR ขาที่ 20)** คอมพิวเตอร์จะเปิดสัญญาณสายนี้ให้เปิด ลอจิก “0” เมื่อพร้อมที่จะติดต่อกับโมเด็ม โมเด็มส่วนมากไม่รายงานสถานภาพของตัวเอง CD, DSR และCTS ให้คอมพิวเตอร์รู้ถ้าหากคอมพิวเตอร์ไม่เปิดสัญญาณ DTR

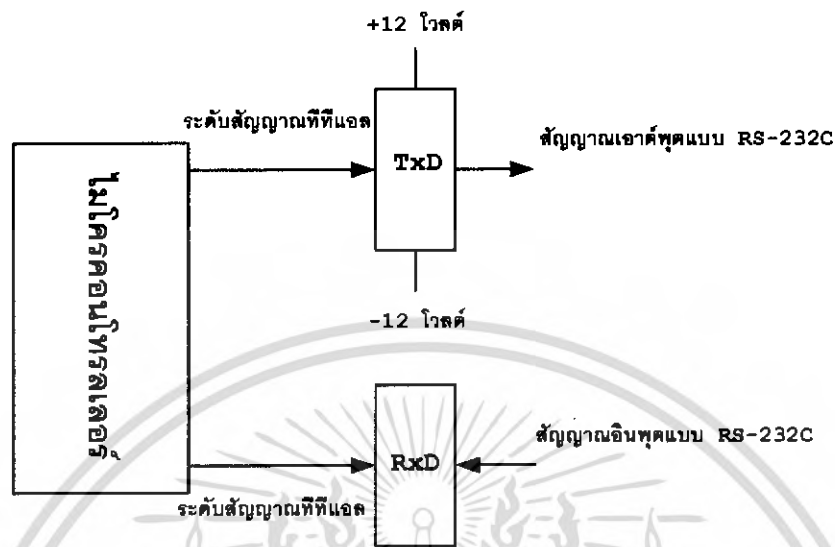
**9. Ring Indication (RI ขาที่ 22)** สัญญาณนี้ใช้ในโมเด็มที่เป็นระบบตอบรับอัตโนมัติ (Auto Answer) สัญญาณนี้จะเปิดเมื่อมีสัญญาณกระดิ่งมาและปิดระหว่างเสียงดิ่งของกระดิ่ง อาจสับสนระหว่างสถานภาพของลอจิกกับสถานภาพของสัญญาณ โดยปกติจะคุ้นเคยกับความรู้สึกที่ว่า เมื่อแรงดันไฟฟ้าเป็นบวกหรือเปิดสัญญาณลอจิกน่าจะเป็น “1” สำหรับสัญญาณต่างๆ ที่กล่าวนี้จะมีลักษณะตรงกันข้ามและเหตุที่กำหนดกฎเกณฑ์ออกมาอย่างนี้ ก็เพราะว่าแต่เดิมนั้นการติดต่อกันทางโทรเลขการทำงานของสัญญาณจะต้องครบวงจรทั้งฝ่ายส่งและฝ่ายรับ เมื่อลอจิกเป็น “0” หรือขณะที่ไม่มีอะไรส่งควรจะมีสัญญาณทางไฟฟ้าครบวงจรอยู่ตลอดเวลา จะได้ว่าวงจรไม่ขาดระหว่างทาง ตรงส่วนไหนควรจะรู้ว่าครบวงจรอยู่ตลอดเวลา ก็โดยการให้ค่าแรงดันไฟฟ้าที่ฝ่ายส่ง ดังนั้นที่สัญญาณไฟฟ้าที่มีศักย์ไฟฟ้าเป็นบวกใช้เป็นลอจิก “0”

#### 2.10.4 การเชื่อมต่อมาตรฐาน RS-232C

ในการเชื่อมต่อแบบอนุกรมเข้ากับอุปกรณ์คอมพิวเตอร์ต่างๆ เช่น คอมพิวเตอร์มักจะกำหนดในการเชื่อมต่อตามมาตรฐาน RS-232C ทั้งนี้เพื่อให้มีการใช้งานเส้นสัญญาณหรือรูปแบบของตัวเชื่อมต่อที่สอดคล้องกันและจะได้ลดปัญหาการเข้ากันไม่ได้ระหว่างสัญญาณของอุปกรณ์ที่ มาเชื่อมต่อทั้งสองด้านให้น้อยลง เนื่องจากระดับแรงดันไฟฟ้าที่ใช้และการแทนความหมายของระดับลอจิกตามมาตรฐานนี้แตกต่างไปจากที่ใช้งานกันในระบบดิจิทัลทั่วไป โดยระดับสัญญาณของพอร์ต RS-232C เป็นแบบสองขั้ว (Bipolar) ที่ระดับแรงดันไฟฟ้าทางด้านลบช่วง -3 โวลต์ถึง -15 โวลต์ ซึ่งจะแทนค่าลอจิก “0” และแรงดันไฟฟ้าทางบวกช่วง +3 โวลต์ ถึง +15 โวลต์ แทนค่าลอจิก “1” ดังนั้นจะเห็นได้ว่ามีความจำเป็น ต้องเพิ่มเติมอุปกรณ์หรือวงจรพิเศษเข้าไป เพื่อเปลี่ยนระดับแรงดันไฟฟ้าจากระบบอยู่ในช่วง +3 โวลต์ ถึง +5 โวลต์ ซึ่งได้จากขาสัญญาณของตัวไมโครคอนโทรลเลอร์ซึ่งเป็นระดับแรงดันไฟฟ้าที่สูงกว่าค่า +3 โวลต์หรือต่ำกว่า -3 โวลต์จะเห็นว่าระดับสัญญาณแบบทีทีแอล (Transistor Transistor Logic :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TTL) จากขาสัญญาณ TxD (Transmitted Data) และ RxD (Received Data) ของไมโครคอนโทรลเลอร์ จะต้องถูกปรับเปลี่ยนไปให้เป็นระดับสัญญาณ RS-232C ก่อนหลังจากนั้นที่จะทำการส่งออกไปในสายสัญญาณต่อไป ลักษณะโดยย่อของสัญญาณ RS-232C ดังรูปที่ 2.22



รูปที่ 2.22 การเปลี่ยนแปลงสัญญาณทีทีแอล

ตารางที่ 2.2 คุณลักษณะโดยย่อของสัญญาณ RS-232C

Driver output logic levels with 3 Kbyte to 7 k load	$15 \text{ Volt} > O_o > 5 \text{ Volt}$ $-5 \text{ Volt} > O_o > -15 \text{ Volt}$
Driver output Voltage when open circuit	$V_o < 25 \text{ Volt}$
<b>Driver output impedance with Power off</b>	<b><math>R_o &gt; 300 \text{ ohms}</math></b>
Output short circuit current	$I_o < 0.5 \text{ A}$
Driver slew rate	$dv / dt < 30 \text{ v/s}$
Receiver input impedance	$7 \text{ kbyte} > R_{in} > 3k$
Receiver input voltage	+15 compatible with driver
Receiver output with open circuit input	MARK
Receiver output with +3 Volt input	SPACE
Receiver output with -3 Volt input	MARK
+15	LOGIC 0 = SPACE, CONTROL ON
+5	Noise Margin
+3	Transition Region
-3	Noise Margin
-5	LOGIC 1 = MAR kbyte
-15	CONTROL OFF

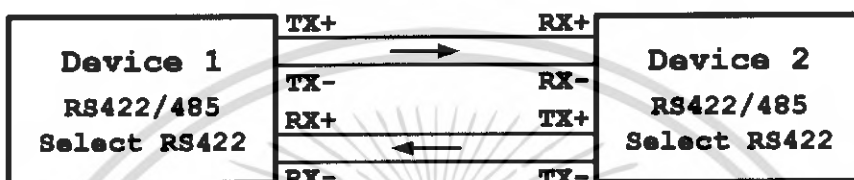
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับกรใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.11 การสื่อสารแบบ RS422/RS485**

การสื่อสารแบบ RS422/485 ทำได้หลายแบบ ซึ่งมีความต่างกัน

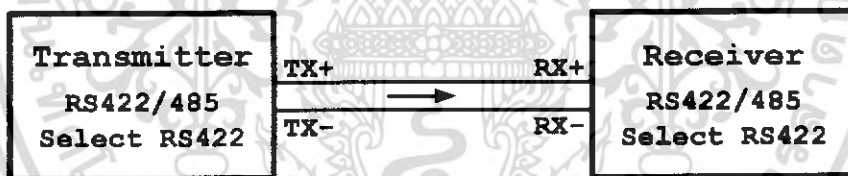
**2.11.1 การเชื่อมต่อแบบ RS422**

**2.11.1.1 การเชื่อมต่อ RS422 แบบฟูลดูเพล็กซ์** เป็นการรับส่งข้อมูลแบบ 2 ทิศทาง สามารถรับส่งได้พร้อมกันตลอดเวลา โดยแต่ละชุดจะมีสายสัญญาณชุดละ 2 คู่ (4 เส้น) ลักษณะการรับส่ง คล้ายการพูดคุยโทรศัพท์ โดยต่อสัญญาณจากวงจรรับเข้ากับวงจรส่งของทั้ง 2 ฝ่าย แต่ในการรับส่งแบบนี้ จะใช้กับอุปกรณ์แบบ “Point-to-Point” คือมีตัวต้นทางและปลายทางอย่างละ 1 ตัว เหตุผลที่ใช้การสื่อสารแบบ RS422 ก็เพื่อเพิ่มระยะทางการรับส่งให้ได้ไกลมากขึ้น โดยไม่ต้องดัดแปลงแก้ไขโปรแกรม

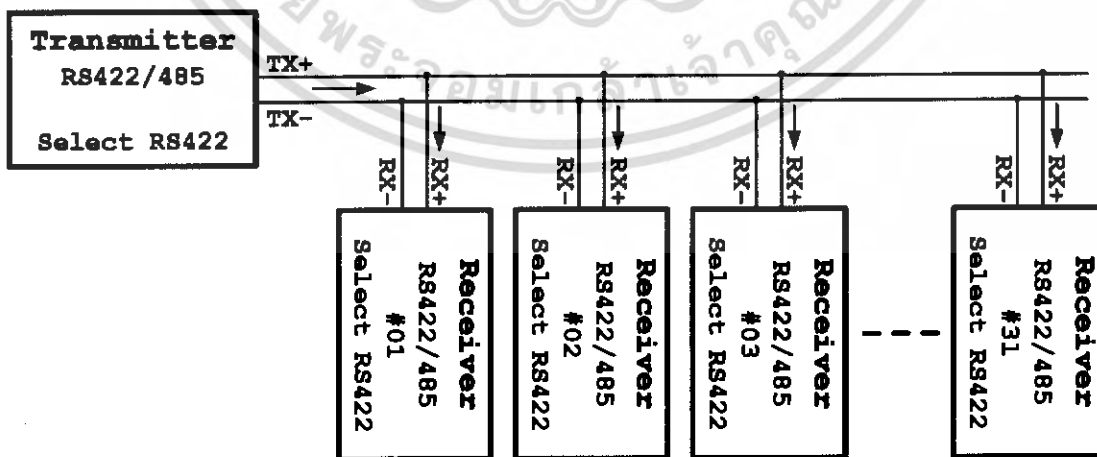


**รูปที่ 2.23** แสดงการต่อสาย RS422 แบบฟูลดูเพล็กซ์

**2.11.1.2 การเชื่อมต่อ RS422 แบบซิมเพล็กซ์** เป็นการรับส่งข้อมูลแบบทิศทางเดียว โดยกำหนดทิศทางไว้คงที่ โดยทิศทางนั้นอาจเป็นรับเข้าอย่างเดียวหรือส่งออกอย่างเดียว ซึ่งวิธีนี้จะใช้สายสัญญาณ 1 คู่ (2 เส้น) ข้อดีที่พิเศษคือ วงจรทางด้านภาคส่ง 1 ชุด สามารถต่อเข้ากับวงจรทางด้านรับได้มากถึง 32 ชุด โดยใช้สายเพียงคู่เดียว



**รูปที่ 2.24** แสดงการต่อสาย RS422 แบบซิมเพล็กซ์

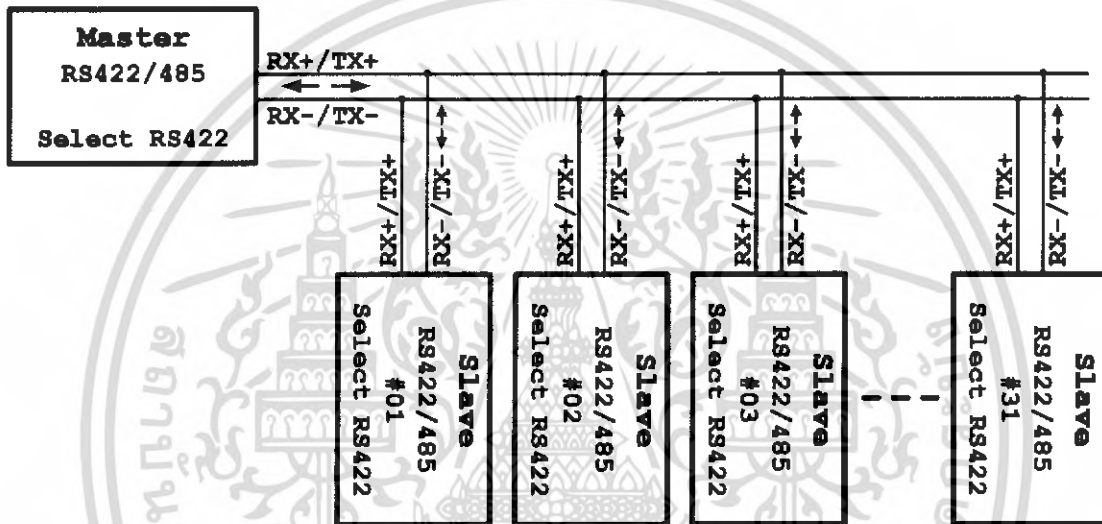


**รูปที่ 2.25** แสดงการต่อสาย RS422 แบบรับหลายตัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

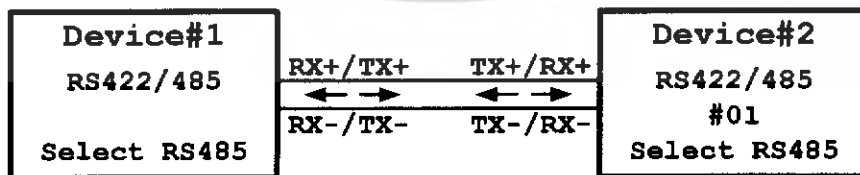
**2.11.2 การเชื่อมต่อแบบ RS485**

**2.11.2.1 ต่อกันแบบขนานทั้งระบบ** วิธีการต่อแบบนี้ต้องมีการจัดอันดับความสำคัญของอุปกรณ์ที่อยู่ในระบบ คือ ให้มีอุปกรณ์ทำหน้าที่เป็นตัวแม่ (Master) 1 ตัว สำหรับทำหน้าที่เป็นตัวกลางในการรับส่งข้อมูลเองทั้งหมด หรือเป็นเพียงการจัดลำดับการรับส่งให้กับอุปกรณ์ตัวอื่นๆ ส่วนตัวลูก (Slave) ในครั้งแรกต้องกำหนดให้เป็นการรับข้อมูลไว้ตลอดเวลา ตัวแม่จะทำหน้าที่ค้นหาตัวลูกทีละตัวเพื่อตรวจสอบว่ามีข้อมูลที่ต้องการส่งหรือไม่ ถ้ามีต้องส่งถึงตัวใด แล้วตัวแม่จึงสั่งให้อุปกรณ์ตัวที่ระบุนั้นคอยรับข้อมูลโดยตรงเอง แล้วตัวแม่จะเปลี่ยนทิศทางมาเป็นรับข้อมูลเพื่อปล่อยสายสัญญาณให้ว่าง เพื่อที่อุปกรณ์ทั้งสองตัวที่ได้รับอนุญาตจะได้รับส่งข้อมูลกันไป เมื่อทั้งคู่ทำการรับส่งข้อมูลกันเสร็จแล้ว ตัวลูกต้องส่งคำสั่งมาบอกตัวแม่ให้ทราบว่าเป็นเสร็จแล้ว เพื่อตัวแม่จะได้ ค้นหาตัวอื่นต่อไป



รูปที่ 2.26 แสดงการต่อสาย RS-485 แบบขนาน

**2.11.2.2 ต่อกันแบบ จุดต่อจุด (Point-to-Point)** วิธีนี้ใช้รับส่งข้อมูลระหว่างอุปกรณ์ 2 ตัว ใช้สายสัญญาณ 1 คู่ โดยอุปกรณ์ทั้ง 2 ตัวต้องควบคุมทิศทางให้เป็นทั้งรับและส่ง โดยการสื่อสารต้องมีข้อกำหนดการรับส่งข้อมูลด้วย ซึ่งอุปกรณ์ทั้ง 2 ตัวนี้ต้องผลัดกันรับผลัดกันส่งข้อมูล ไม่สามารถรับส่งในเวลาเดียวกันได้



รูปที่ 2.27 แสดงการต่อสาย RS485 แบบ จุดต่อจุด

### 2.11.2.3 ระบบเครือข่าย RS-485

- สามารถต่ออุปกรณ์ในระบบได้ไม่เกิน 32 ตัว
- ติดต่อกันแบบ ฮาล์ฟดูเพล็กซ์ (ฮาล์ฟดูเพล็กซ์ คือการสื่อสารที่สามารถส่งได้ 2 ทิศทางแต่คนละเวลาห้ามส่งพร้อมกัน )
- ระบบสมดุล (Balanced system ) ใช้สายสัญญาณ 1 คู่ และ สายกราวด์ 1 เส้น
- ติดต่อกันระยะทางไม่เกิน 1,200 เมตร หรือ 4,000 ฟุต
- ความยาวของระบบและจำนวนโหนดสามารถขยายได้โดยตัวทวนสัญญาณ หรือ Repeater

### 2.11.2.4 ฮาร์ดแวร์ ของ RS-485

ระบบเครือข่าย RS-485 สามารถติดต่อกันผ่านสาย 2 เส้น ( โดยทั่วไปใช้สายคู่ตีเกลียว หรือ twisted pair ) สัญญาณข้อมูลที่ส่งจะถูกวัดความแตกต่างของสัญญาณที่ตัวรับ ซึ่งมีข้อดีคือ สามารถส่งข้อมูลในระยะไกลและกำจัดปัญหาสัญญาณรบกวนได้ดี ลักษณะการเชื่อมต่อของ RS-485 สามารถทำได้ใน 2 ลักษณะคือ 2 สาย ( two-wire ) และ 4 สาย ( four-wire ) ในระบบ 2 สาย ตัวส่งและตัวรับของแต่ละอุปกรณ์ต่อเข้ากับสายคู่ตีเกลียว 1 เส้น ระบบ 4 สาย ตัวส่งของพอร์ตแม่ ( master port ) ต่อเข้ากับตัวรับของตัวลูก ( slave port ) ทุกตัวในระบบผ่านสายคู่ตีเกลียวเส้นหนึ่ง และตัวส่งของตัวลูกทุกตัวจะต่อเข้ากับตัวรับของตัวแม่ผ่านสายคู่ตีเกลียวอีกเส้นหนึ่ง การติดต่อกันในระบบอุปกรณ์ทุกตัวจะต้องถูกกำหนดตำแหน่ง ( address ) ตัวแม่ติดต่อกับตัวลูกทุกตัวโดยตรงแต่ตัวลูกแต่ละตัวจะติดต่อกับตัวแม่ได้เท่านั้นไม่สามารถติดต่อกันเองได้ และในการติดต่อนั้นตัวลูกจะติดต่อกับตัวแม่ได้ที่ละตัว เมื่อตัวลูกตัวใดตัวหนึ่งกำลังติดต่อกับตัวแม่อยู่ ตัวลูกที่เหลือจะไม่สามารถติดต่อกับตัวแม่ได้และจะอยู่ในสถานะที่เรียกว่า ค่าความต้านทานสูง หรือ สามสถานะ (Tri-State) ซึ่ง ฮาร์ดแวร์ บางผลิตภัณฑ์สามารถจัดการสถานะดังกล่าวได้โดยอัตโนมัติ แต่สำหรับบางกรณีจำเป็นต้องใช้ซอฟต์แวร์เป็นตัวจัดการ (ถ้าอุปกรณ์ RS-485ถูกควบคุมผ่าน RS-232 พอร์ตอนุกรม การจัดการเกี่ยวกับ สามสถานะ จะถูกจัดการด้วย RTS handshake line ) ผลจากการทำงานแบบ สามสถานะ ทำให้ต้องเสียเวลาระยะเวลาหนึ่งระหว่างการสิ้นสุดการส่งข้อมูลและการทำ สามสถานะ ซึ่งช่วงเวลาดังกล่าวเรียกว่า เทิร์นอราวดีเลย์ (turn-around delay) ในช่วงเวลานี้จะไม่มีตัวลูกตัวใดสามารถส่งข้อมูลได้ ซึ่งมีผลกับระบบแบบ 2 สาย แต่ไม่มีผลกับระบบแบบ 4 สาย ในทางอุดมคติแล้วระยะเวลาดังกล่าวควรมีค่าเท่ากับระยะของความยาว 1 ตัวอักษรที่อัตราการส่งข้อมูล ( baud rate ) นั้น เช่นที่บิตเรท 9,600 bps ควรมีค่าเทิร์นอราวดีเลย์ เท่ากับ 1 ms. ซึ่งผู้ผลิตควรบอกค่าเทิร์นอราวดีเลย์ของอุปกรณ์ไว้ด้วย

### ข้อดี-ข้อเสีย ของการต่อระบบแบบ 2 สาย และ 4 สาย

	2 สาย	4 สาย
<b>ข้อดี</b>	1. ราคาในการเดินสายถูก 2. แต่ละโหนดสามารถติดต่อกันได้โดยตรง	1. ไม่ต้องคำนึงถึง turn around delay 2. เนื่องจากเป็นการติดต่อแบบแม่และลูก ลูกแต่ละตัวไม่สามารถติดต่อกันได้ ทำให้สามารถต่ออุปกรณ์ที่มีโปรโตคอลต่างกันเข้าในระบบเดียวกันได้
<b>ข้อเสีย</b>	1. การติดต่อต้องคำนึงถึง turn around delay	

**ข้อสังเกต** ในระบบ 2 สาย หมายถึง สายสัญญาณ 2 เส้นและสายกราวด์ 1 เส้น สำหรับระบบ 4 สาย หมายถึง สายสัญญาณ 4 เส้นและสายกราวด์ 1 เส้น

#### 2.11.2.5 ซอร์ฟแวร์ของ RS-485

ซอร์ฟแวร์ ของระบบ RS-485 ต้องจัดการในการกำหนดตำแหน่ง, เทิร์นอราวติเลย์ และ tri-state การกำหนด เทิร์นอราวติเลย์ ต้องกำหนดให้เหมาะสม การกำหนด เทิร์นอราวติเลย์ ที่สั้นไปหรือยาวไปอาจมีผลต่อระบบ นอกจากนี้ ซอร์ฟแวร์ ดังกล่าวต้องสามารถจัดการเกี่ยวกับ สามสถานะได้ การเชื่อมต่อแบบเครือข่าย Multi-drop 485 ตามข้อกำหนดของ EIA RS-485 จะบอกสายข้อมูลเป็น A และ B แต่ก็มีผู้ผลิตจำนวนมากแทนด้วย + และ - ซึ่งโดยมากแล้วในการเชื่อมต่อ มักต่อ - เข้ากับ A และ ต่อ + เข้ากับ B การสลับขั้วกันไม่ทำความเสียหายต่ออุปกรณ์ แต่จะทำให้ไม่สามารถติดต่อกันได้ การเชื่อมต่อระบบ สายกราวด์เป็นสายที่มีความสำคัญ ทำให้แต่ละโหนดมีกราวด์เดียวกัน ( Common ground ) ระบบที่มีการติดต่อในระยะทางมากกว่า 100 เมตร หรือ 1,000 ฟุต จะมีความแตกต่างกันในสายกราวด์ ซึ่งระบบ RS-485 ยินยอมให้แตกต่างกันได้ในช่วง -7 V ถึง +12 V ถ้าแตกต่างกันมากกว่าช่วงดังกล่าวจะทำให้ข้อมูลหรือพอร์ตเสียหายได้ เราสามารถแก้ไขปัญหานี้ได้โดยการใช้ตัวแยกทางแสง ( Optical isolator )

ตารางที่ 2.3 แสดงคุณสมบัติของ EIA RS-422

EIA RS-422 Specification Summary				
Parameter	Condition	Min	Max	Unit
Driver Output Voltage			10	v
Open Circuit			-10	v
Driver Output Voltage	Pload = 100 ohms	3		v
Loaded		-2		v
Driver Output Resistance	A to B		100	ohms
Driver Output Short-circuit Current	Per Output to common		±150	mA
Driver Output Rise Time	Rt = 100 ohms		10	% of bit width
Driver Common Mode Voltage	Rt = 100 ohms		±3	v
Receiver Sensitivity	Vcm ≤ ±5v		±200	mV
Receiver Common Mode Voltage		-7	7	v
Receiver Input Resistance		4000		ohms
Difference Receiver Voltage	Operational:		±10	v
	Withstand:		±12	v

ตารางที่ 2.4 แสดงคุณสมบัติของ EIA RS-485

EIA RS-485 Specification Summary				
Parameter	Condition	Min	Max	Unit
Driver Output Voltage		1.5	6	v
Open Circuit		-1.5	-6	v
Driver Output Voltage	Pload = 100 ohms	1.5	6	v
Loaded		-1.5	-6	v
Driver Output Short-circuit Current	Per Output to +12v or -7v		±150	mA
Driver Output Rise Time	Rload = 54 ohms Cload = 50pF		30	% of bit width
Driver Common Mode Voltage	Rload = 54 ohms		3	v
Receiver Sensitivity	-7v ≤ Vcm ≤ 12v		±200	mV
Receiver Common Mode Voltage		-7	12	v
Receiver Input Resistance		12k		ohms
Difference Receiver Voltage	Operational:		±10	v
	Withstand:		±12	v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 2.5 แสดงคุณสมบัติของ EIA RS-232

EIA RS-232 Specification Summary				
Parameter	Condition	Min	Max	Unit
Driver Output Voltage Open Circuit			25	v
Driver Output Voltage Loaded	3 Kohms $\leq$ RL $\leq$ 7 Kohms	5	15	v
Driver Output Resistance Power off	$-2v \leq V_o \leq 2v$		300	ohm
Driver Output Short-circuit Current			500	mA
Driver Output Slew Rate			30	v/ussec
Maximum Load Capacitance			2500	pF
Receiver Sensitivity	$-5v \leq V_{in} \leq 12v$		+200	mV
Receiver Input Resistance	$3v \leq V_{in} \leq 25v$	3000	7000	ohm
Receiver Input Threshold Output = Mark		-3		v
Output = Space			3	v

### 2.12 สถาปัตยกรรมของฐานข้อมูล

สถาปัตยกรรมของฐานข้อมูลแบ่งตาม โครงสร้างทางกายภาพได้เป็นสองชนิดคือ เมนเฟรมคอมพิวเตอร์(Mainframe Computer) และคอมพิวเตอร์ส่วนบุคคล (Personal Computer) แต่ถ้าเราแบ่งตามสถาปัตยกรรมของฐานข้อมูลตามจำนวนผู้ที่เราได้เป็นสองแบบด้วยกัน คือระบบฐานข้อมูลที่มีผู้ใช้เพียงคนเดียวและระบบฐานข้อมูลที่มีผู้ใช้หลายคนและในที่นี้เราจะพิจารณาการแบ่งสถาปัตยกรรมของฐานข้อมูลตามจำนวนผู้ใช้นี้

#### - ระบบฐานข้อมูลผู้ใช้งานเดียว

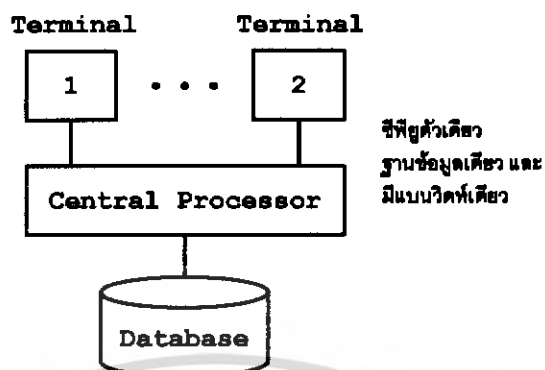
โดยปกติแล้วจะเป็นคอมพิวเตอร์ส่วนบุคคลมากกว่า ข้อมูลของฐานข้อมูลจะอยู่ในฮาร์ดดิสก์ ระบบจัดการฐานข้อมูลเป็น โปรแกรมเดี่ยวๆและฐานข้อมูลมีขนาดไม่ใหญ่ ระบบนี้สิ้นเปลืองทรัพยากรที่ว่า ต้องรอคำสั่งจากผู้ใช้งานเดียว

#### - ระบบฐานข้อมูลผู้ใช้งานหลายคน

ระบบนี้สามารถใช้ได้มากกว่าหนึ่งคนในเวลาเดียวกัน ทำให้มีประสิทธิภาพเนื่องจากระบบฐานข้อมูลสามารถตอบสนองต่อผู้ใช้หลายคน โดยมีรูปแบบที่ชัดเจนได้หลายรูปแบบ เช่น Teleprocessing, Client/Server และ Distributed Data Processing เป็นต้น ดังนี้

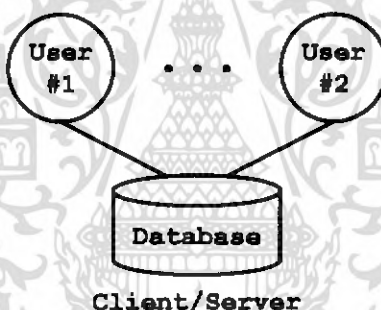
-- **Teleprocessing** ระบบนี้มีประสิทธิภาพสูงและต่อกับเครื่องคอมพิวเตอร์ที่ปลายทาง แต่ระบบนี้มีข้อเสียคือ ใช้ซีพียูตัวเดียวในการทำงานทั้งหมด ดังนั้นซีพียูต้องมีประสิทธิภาพสูง และเมื่อมีผู้ใช้งานเพิ่มมากขึ้นเรื่อยๆ ประสิทธิภาพจะลดลงอย่างมาก นอกจากนี้ยังขึ้นอยู่กับแบนด์วิดท์ (ความสามารถในการรับส่งข้อมูลระหว่างต้นทางกับปลายทาง) ซึ่งต่อระหว่างผู้กับซีพียู ถ้าแบนด์วิดท์แคบ ประสิทธิภาพ

ก็ลดลง และเมื่อเพิ่มผู้ใช้อีก แบนด์วิดท์ของแต่ละคนที่ได้ก็น้อยลงอีก ระบบนี้มีใช้ในเมนเฟรมคอมพิวเตอร์ ซึ่งมีราคาสูง



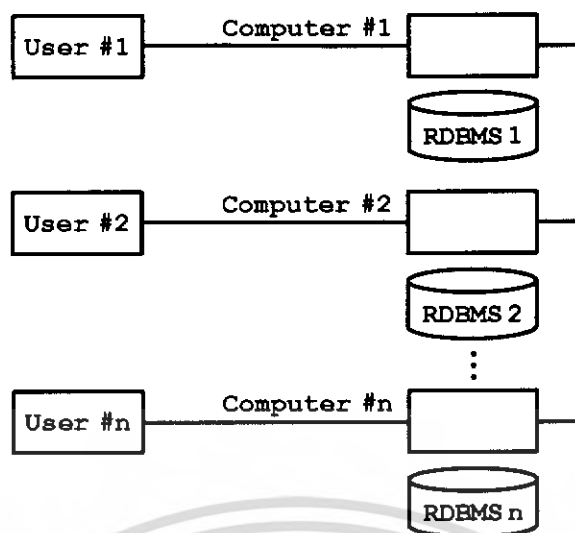
รูปที่ 2.28 แสดงสถาปัตยกรรมแบบ Teleprocessing

-- **Client/Server** ในระบบนี้ซีพียูที่อยู่ส่วนกลางเรียกว่าเซิร์ฟเวอร์ ต่อกับเครื่องคอมพิวเตอร์หลายๆเครื่องที่เรียกว่าไคลเอ็นต์ (Client)



รูปที่ 2.29 แสดงสถาปัตยกรรมแบบ Client/Server

-- **Distributed Data Processing** ระบบนี้ใช้เซิร์ฟเวอร์หลายๆเครื่องและผู้ใช้หลายคน ซึ่งเซิร์ฟเวอร์หลายๆเครื่องนี้จะเป็นการเชื่อมต่อฐานข้อมูลจากเครื่องคอมพิวเตอร์ต่างๆให้เป็นฐานข้อมูลเดียวกันการมีข้อมูลบนเซิร์ฟเวอร์แต่ละเครื่องช่วยลดเวลาการติดต่อระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ได้ ผลที่ได้คือมีประสิทธิภาพสูงแต่ข้อเสียคือต้องมีการควบคุมอย่างระมัดระวังเพื่อให้แน่ใจว่าข้อมูลในเซิร์ฟเวอร์แต่ละตัวจะสอดคล้องกันกับคอมพิวเตอร์เซิร์ฟเวอร์อื่นๆ ไม่ว่าจะเป็นการอัปเดต การเพิ่ม และการลบข้อมูล เป็นต้น ระบบนี้แสดงดังรูป



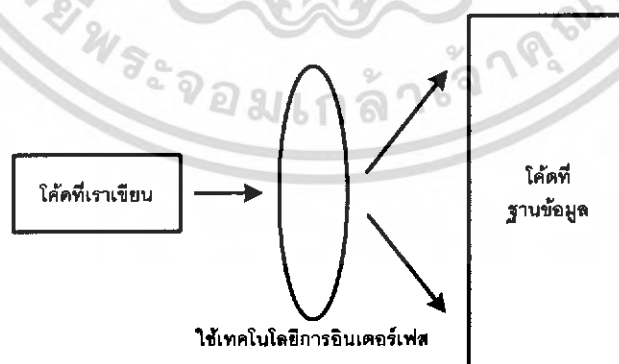
รูปที่ 2.30 แสดงสถาปัตยกรรมแบบ Distributed Data Processing

เมื่อมีระบบฐานข้อมูลอยู่ที่ เซิร์ฟเวอร์ หรือเครื่องคอมพิวเตอร์ส่วนบุคคลแล้วจำเป็นต้องมีเทคโนโลยีสำหรับการเข้าถึงข้อมูลในฐานข้อมูลอยู่ที่ไคลเอ็นต์ เทคโนโลยีสำหรับการเข้าถึงข้อมูลนี้เรียกว่าเทคโนโลยีการอินเทอร์เน็ตเฟส

### 2.12.1 เทคโนโลยีสำหรับการเข้าถึงข้อมูลในฐานข้อมูล

เนื่องจากฐานข้อมูลคือซอฟต์แวร์ชิ้นหนึ่งที่มีความซับซ้อนมาก (เทคโนโลยีการอินเทอร์เน็ตเฟสกับฐานข้อมูลมีความซับซ้อนน้อยกว่าฐานข้อมูล) มีผลให้การเขียนโปรแกรมเพื่อการติดต่อกับฐานข้อมูลด้วยอินเทอร์เน็ตเฟสแบบธรรมดาเป็นงานเขียนค่อนข้างซับซ้อนมาก แต่ถ้าใช้เทคโนโลยีการอินเทอร์เน็ตเฟสกับฐานข้อมูลแล้ว นักพัฒนาโปรแกรมสามารถเขียนโปรแกรมด้วยโค้ดที่มีจำนวนลดลง (โค้ดนี้อยู่ที่ฐานข้อมูล) ไปมากทีเดียว

การอินเทอร์เน็ตเฟสกับฐานข้อมูลที่ดัดนั้นเหมือนกับเลนส์ที่ขยายโค้ดของเราให้มีขนาดใหญ่ขึ้นดังรูป

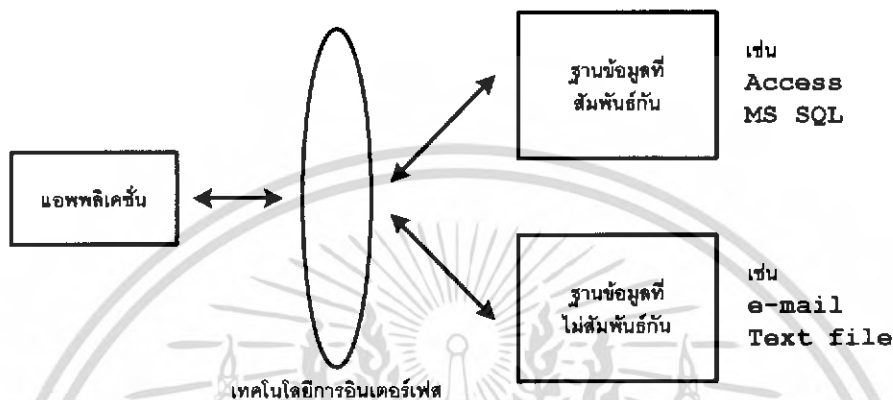


รูปที่ 2.31 แสดงการเข้าถึงการใช้เทคโนโลยีการอินเทอร์เน็ตเฟสทำให้โค้ดที่เราเขียนที่ลดลงอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมเพื่อสื่อสารกับฐานข้อมูลด้วยวิธีการธรรมดา นอกจากจะมีความซับซ้อนแล้ว ยังมีผลให้แอปพลิเคชันที่ได้ไม่ยืดหยุ่นต่อการใช้งานและมีข้อจำกัด รวมทั้งถ้าเราเปลี่ยนไปใช้กับฐานข้อมูลอื่น ๆ เราต้องแก้ไขโค้ดและสูญเสียเวลาไปมาก

เทคโนโลยีการอินเทอร์เน็ตเฟสได้ให้รูปแบบการอินเทอร์เน็ตเฟสแบบเดียวสำหรับการสื่อสารกับฐานข้อมูลที่แตกต่างกัน เทคโนโลยีการอินเทอร์เน็ตเฟสสมัยใหม่ทำให้เราเขียนโค้ดได้อย่างง่าย และสามารถทำกระบวนการซับซ้อนกับฐานข้อมูลชนิดใดๆ ก็ได้

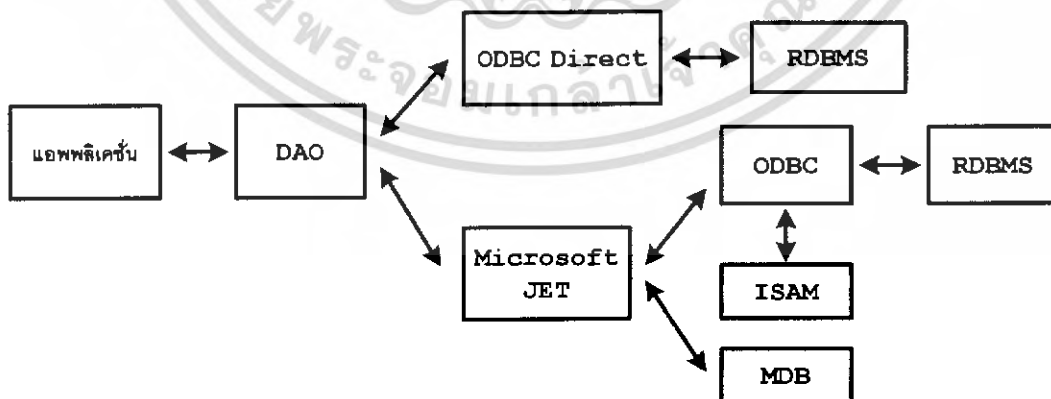


**รูปที่ 2.32** แสดงการใช้การอินเทอร์เน็ตเฟสระหว่างแอปพลิเคชันกับฐานข้อมูล

เทคโนโลยีการอินเทอร์เน็ตเฟสกับฐานข้อมูลมีด้วยกันหลาย ๆ ชนิด เช่น ODBC (Open Database Connectivity), RDO (Remote Data Objects), DAO (Data Access Object), OLE (Object-Linking and Embedding Database) และ ADO (ActiveX Data Object)

คำถามมีอยู่ว่าเมื่อไรและที่ไหนควรจะใช้เทคโนโลยีการอินเทอร์เน็ตเฟสเหล่านี้ และใช้ชนิดไหน การที่จะตอบคำถามเหล่านี้ได้ เราต้องเรียนรู้แต่ละเทคโนโลยีก่อน ดังนี้

**1. DAO (Data Access Object)**



**รูปที่ 2.33** แสดงสถาปัตยกรรมของ DAO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี ค.ศ.1995 บริษัทไมโครซอฟท์ได้แนะนำ DAO สำหรับ Microsoft Jet Database Engine ซึ่งเป็นเซตของการอินเตอร์เฟซแบบ COM Automation โดย DAO ทำให้เราสามารถใส่ภาษาโปรแกรมคอมพิวเตอร์เข้าถึงและจัดการข้อมูลในฐานข้อมูลได้

DAO เข้าถึงฐานข้อมูลด้วยโมเดลออบเจกต์ แทนที่จะเป็นการเรียกกลุ่มฟังก์ชันใน DLL (Dynamic Link Library) โดยการใช้โมเดลออบเจกต์ทำให้การเรียกดู หรือกระทำข้อมูลในฐานข้อมูลได้ง่าย นอกจากนี้แล้วออบเจกต์โมเดล DAO ได้รวมคลาสต่าง ๆ และออบเจกต์นี้ยังห่อหุ้มด้วยโครงสร้างต่างๆ ของฐานข้อมูล Access เช่น ตาราง คิวรี อินเด็กซ์ และอื่น ๆ อีกนั่นหมายความว่า DAO เราแก้ไขโครงสร้างของฐานข้อมูล Access ได้โดยตรง โดยปราศจากการใช้คำสั่ง SQL สเตทเม้นท์ได้

### DAO สามารถติดต่อกับฐานข้อมูลได้ 2 แบบ คือ

- **Microsoft Jet** สามารถเข้าถึงข้อมูลในฐานข้อมูล Access ได้โดยตรง และ Microsoft Jet นี้ต่อกับ ODBC เพื่อติดต่อกับฐานข้อมูล ISAM (Indexed-Sequential Access Model) เช่น FoxPro, Paradox หรือ Lotus 1-2-3 และฐานข้อมูล RDBMS อื่น ๆ

- **ODBC Direct** ติดต่อกับฐานข้อมูล RDBMS อื่น ๆ โดยไม่ผ่าน Microsoft Jet

### คุณลักษณะเด่นของ DAO เมื่อใช้ Microsoft Jet

Microsoft Jet มีฟังก์ชันมากมาย ซึ่งมีคุณลักษณะเด่นดังต่อไปนี้

- การจัดการกับข้อมูล (Data Manipulation) สามารถที่จะเพิ่ม อัปเดต และลบข้อมูลได้
- Join กับตารางในฐานข้อมูลอื่นได้ (Heterogeneous Join) เราสามารถทำการ Join กับตารางในฐานข้อมูลอื่น ๆ เช่น สามารถ Join ตารางใน Access กับตารางใน MS SQL Sever ได้
- การทำซ้ำ (Replication) Microsoft Jet อนุญาตให้นักพัฒนาสามารถทำสำเนาฐานข้อมูลได้ ดังนั้นผู้ใช้ที่อยู่ต่าง ๆ กัน สามารถทำงานกับฐานข้อมูลที่เกือบปีมา ณ เวลาเดียวกันและใช้ร่วมกัน หรือทำซิงค์โครไนซ์กัน เช่น เมื่อคนที่ 1 อัปเดตข้อมูล คนอื่น ๆ ที่อยู่อีกที่หนึ่งก็จะเห็นข้อมูลถูกอัปเดตด้วย

## 2. RDO (Remote Data Objects)

RDO ใช้โมเดลออบเจกต์ในการเข้าถึงข้อมูล ODBC โมเดลการ โปรแกรมมิ่ง RDO คล้ายกับโมเดลของ DAO ยกเว้นว่ามีจุดประสงค์สำหรับการเข้าถึงฐานข้อมูล SQL มากกว่าที่จะเป็นข้อมูล ISAM ดังนั้น RDO จึงเน้นไปทางด้าน Stored Procedure มากกว่า

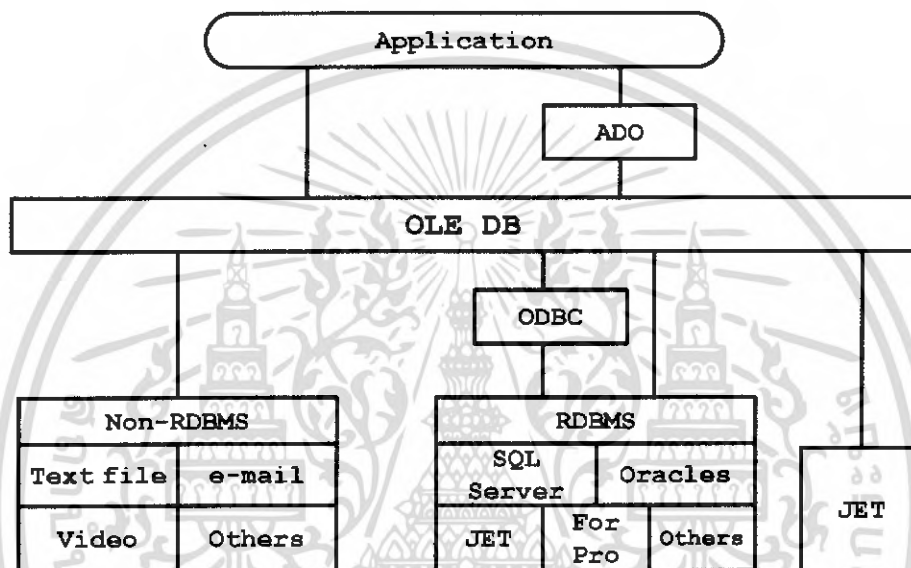
RDO ถูกพัฒนาขึ้นเพื่อให้นักพัฒนาโปรแกรมภาษา Visual Basic มีความสามารถเข้าถึงข้อมูล ODBC ได้โดยไม่ต้องเขียนโค้ด ODBC API ด้วยตนเอง ดังนั้น RDO จึงถูกจัดให้เป็นอินเตอร์เฟซในระดับสูง

นอกจากนี้แล้ว RDO ถูกออกแบบให้เอาข้อดีของฐานข้อมูล SQL Server ซึ่งก็คือการใช้คิวรีที่ซับซ้อน

### 2.1.2.2 เทคโนโลยีการเข้าถึงข้อมูลในแนวความคิด Universal Data Access (UDA)

UDA เป็นสถาปัตยกรรมใหม่ของไมโครซอฟท์ที่ทำให้การเข้าถึงข้อมูลเป็นไปได้อย่างมีประสิทธิภาพ ทั้งแบบฐานข้อมูลที่มีความสัมพันธ์กันและไม่สัมพันธ์กัน (Relational and Nonrelational) และได้หลายแพลตฟอร์ม ด้วยการนำ UDA มาใช้ เราสร้างหรือเขียนโปรแกรมได้ง่าย รวมทั้งสามารถใช้กับเครื่องมือและภาษาโปรแกรมมิ่งของทุกวันนี้ได้ทั้งหมด

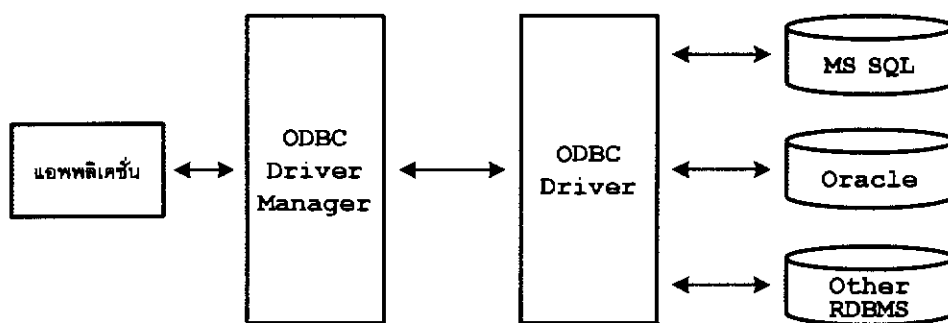
UDA เป็นระบบเปิดและไม่จำเป็นต้องใช้ผลิตภัณฑ์ของผู้ผลิตฐานข้อมูลรายเดียว ในความจริงแล้ว UDA สามารถใช้กับแพลตฟอร์มฐานข้อมูลที่มีอยู่ทุกวันนี้ได้ทั้งหมด UDA ประกอบด้วยเทคโนโลยีหลัก ๆ 3 เทคโนโลยีด้วยกัน ดังรูป



รูปที่ 3.34 แสดงสถาปัตยกรรมของ UDA

จากรูปสมมติว่าข้อมูลอยู่ใน RDBMS ซึ่งมีไคลเอนต์ ODBC และไม่มี OLE DB Provider แอปพลิเคชันของเราสามารถใช้ ODBC ติดต่อกับข้อมูลในฐานข้อมูล หรือแอปพลิเคชัน เราสามารถใช้ OLE DB Consumer ติดต่อผ่าน OLE DB Provider ไปยังไคลเอนต์ ODBC จึงจะสามารถติดต่อกับข้อมูลในฐานข้อมูลได้ (Data Source) ที่มี OLE DB Provider แอปพลิเคชันของเราใช้ OLE DB Consumer เพื่อติดต่อโดยตรงกับ OLE DB Provider โดยไม่ผ่านไคลเอนต์ ODBC (ดูรูปสถาปัตยกรรม OLE DB ประกอบ)

**1. ODBC (Open Database Connectivity)**



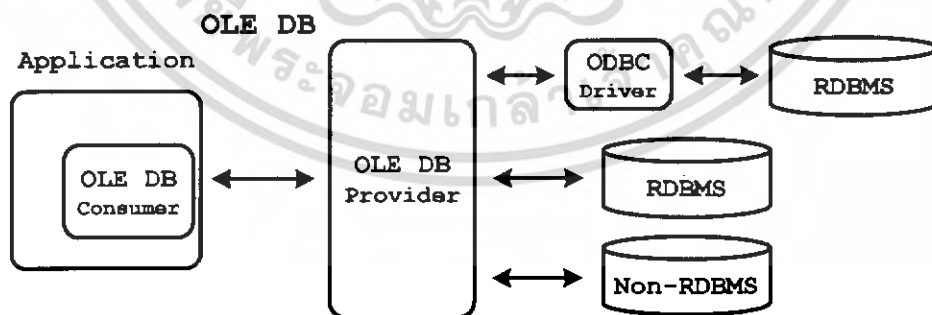
**รูปที่ 2.35** แสดงสถาปัตยกรรมของ ODBC

ODBC สร้างเมื่อปลายทศวรรษ 1980 และช่วงต้น 1990 เพื่อให้การเขียนติดต่อโปรแกรมติดต่อกับฐานข้อมูลแบบสัมพันธ์ใช้การอินเตอร์เฟสมีรูปแบบเดียวกัน ODBC ได้ให้ API สำหรับแอปพลิเคชันที่จะติดต่อกับฐานข้อมูลต่าง ๆ โดยผ่านไควร์เฟเวอร์ ODBC

เมื่อเปรียบเทียบกับกรอินเตอร์เฟสอื่น ๆ แล้ว ODBC API ถูกจัดเป็นกลุ่มการอินเตอร์เฟสระดับล่าง (Low Level Interface) ดังนั้น ODBC API ทำให้แอปพลิเคชันสามารถกระทำกับฐานข้อมูลในระดับล่าง ซึ่ง ODBC นั้นได้รับความยอมรับให้เป็นมาตรฐานในการอินเตอร์เฟสกับฐานข้อมูลแบบสัมพันธ์

**2. OLE DB (Object-Linking and Embedding Database)**

OLE DB เป็นวิธีการเข้าถึงข้อมูลชนิดใหม่จากไมโครซอฟท์ และเป็นเซตของ COM (Component Object Model) ซึ่งประกอบด้วยชุดของการอินเตอร์เฟสที่เป็นมาตรฐาน OLE DB ถูกออกแบบมาเพื่อเข้าถึงข้อมูลเดต้าซอสได้หลากหลายชนิดโดยไม่คำนึงถึงฐานข้อมูลว่าจะสัมพันธ์กันหรือไม่ ความยืดหยุ่นนี้ทำให้ผู้พัฒนาใช้ลักษณะเด่นของ OLE DB ได้อย่างมีประสิทธิภาพ OLE DB จะแบ่งเป็น Consumer และ Provider



**รูปที่ 2.36** แสดงสถาปัตยกรรมของ OLE DB

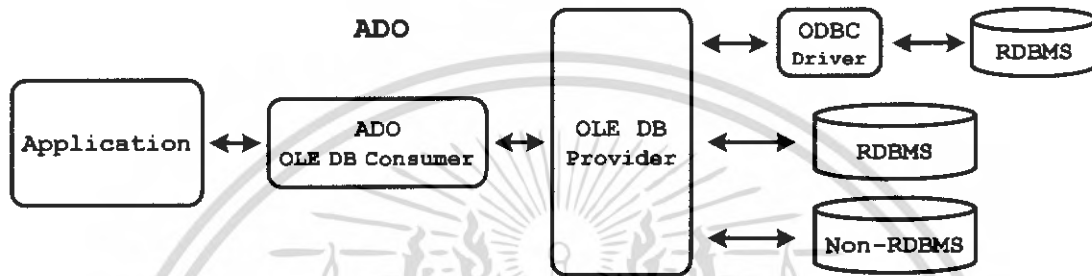
จากรูปด้าน Consumer นี้ แสดงถึงผู้ใช้ที่ต้องการสร้างแอปพลิเคชัน ส่วน Provider จะเป็นการจัดวางข้อมูลให้อยู่ในรูปแบบตารางและส่งไปยังผู้ใช้ ทำให้ผู้ใช้สามารถดึงข้อมูลจากฐานข้อมูลเพื่อกระทำกับข้อมูลเหล่านั้น เช่น การอัปเดตข้อมูล การลบข้อมูล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- OLE DB Consumer สามารถเข้าถึงข้อมูลจาก Provider ใด ๆ ก็ได้ เนื่องจาก Provider เป็น ออบเจ็กต์ COM ดังนั้น Consumer สามารถเข้าถึงข้อมูลโดยใช้ภาษาใด ๆ ก็ได้ เช่น Visual C++, ภาษา Java เป็นต้น

- OLE DB Provider จะจัดวางข้อมูลต่าง ๆ ให้อยู่ในรูปแบบตาราง ทำให้ Consumer สามารถอ่าน ข้อมูลรูปแบบนั้นเป็นแบบเรคคอร์ดเซต หรือกระทำกับข้อมูลนั้น ฉะนั้น Provider สามารถเป็นได้ทั้งแบบ ง่าย ๆ หรือแบบซับซ้อน โดยขึ้นอยู่กับข้อมูลนั้น ๆ

### 3. ADO (ActiveX Data Object)



รูปที่ 2.37 แสดงสถาปัตยกรรมของ ADO

ADO เป็นการอินเทอร์เฟซข้อมูลในระดับสูง (Hi-Level Interface) ADO มีประสิทธิภาพสูงและมีความเชื่อถือได้ในการเข้าถึงข้อมูล ADO ถูกออกแบบมาให้ง่ายต่อการอินเทอร์เฟซระดับแอปพลิเคชันกับ OLE DB ที่มีประสิทธิภาพสูง ไม่ว่าจะเป็นฐานข้อมูลแบบสัมพันธ์กันหรือแบบไม่สัมพันธ์กัน เช่น E-Mail, File Systems, Text, Graphic และอื่น ๆ

ADO ใช้การอินเทอร์เฟซแบบ Dual (Dual Interface) (ซึ่งเป็นผลจากการรวมคุณลักษณะเด่นของ Remote Data Object และ DAO ไว้ด้วยกัน) อินเทอร์เฟซนี้ทำให้ ADO สามารถใช้ภาษาสคริปต์ได้ และ ADO ยังมีคุณลักษณะเด่นในการทำคิวรีและการจัดการข้อมูล เช่น การใช้คำสั่ง SQL สเตทเม้นท์ นอกจากนี้แล้ว ADO ยังได้ให้การคิวรีแบบ Asynchronous ที่อนุญาตให้เราควบคุมการคิวรีที่จะถูกส่งกลับมาซึ่งแอปพลิเคชันก่อนที่คิวรีจะสำเร็จ การทำอย่างนี้อนุญาตให้เราให้สิทธิแก่ผู้ใช้สามารถยกเลิกหรือกระทำต่อการคิวรีที่ใช้เวลานาน เป็นต้น

#### ในการติดต่อกับฐานข้อมูลด้วยการใช้เทคโนโลยีการอินเทอร์เฟซต่าง ๆ สามารถสรุปได้ดังนี้

**OLE DB** มีประสิทธิภาพดีที่สุดในเมื่อเทียบกับการอินเทอร์เฟซแบบอื่น ๆ และใช้เวลาในการเข้าถึงข้อมูลน้อยกว่า ADO สำหรับการเขียนโปรแกรมแล้ว ภาษา Visual C++ จะเหมาะสมที่สุด แต่อย่างไรก็ตามการเขียนโค้ดเพื่อติดต่อกับฐานข้อมูลทั้งแบบสัมพันธ์กันและแบบไม่สัมพันธ์กันจะมีความซับซ้อนพอสมควร

**ADO** มีประสิทธิภาพดีมากในการเขียนโค้ดเพื่อติดต่อกับฐานข้อมูลทั้งแบบสัมพันธ์กันและแบบที่ไม่สัมพันธ์กัน ในการเขียนโค้ดสามารถใช้โปรแกรมภาษา Visual Basic หรือภาษาสคริปต์ได้ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VBScript และ JavaScript แต่ไม่เหมาะสมกับภาษา Visual C++ นอกจากนี้ ADO ยังมีฟังก์ชันการใช้งานมากเมื่อเปรียบเทียบกับฟังก์ชันที่อินเทอร์เฟซอื่นมี

ทั้ง OLE DB และ ADO มีประสิทธิภาพที่ดีกว่าการใช้เทคโนโลยีแบบอื่น ๆ ด้วยเหตุนี้ทางผู้ผลิตเทคโนโลยีทั้งสองนี้จึงได้วางไว้เป็นมาตรฐานในการเข้าถึงข้อมูลได้ทุกชนิด และข้ามแพลตฟอร์มได้

**DAO** มีประสิทธิภาพดีสำหรับการใช้กับฐานข้อมูล Access การเขียนโค้ดง่ายและใช้ได้กับหลาย ๆ ภาษา

**RDO** ใช้ในการติดต่อข้อมูลที่ตั้งอยู่ Local และตั้งอยู่ระยะไกล เหมาะสำหรับฐานข้อมูล SQL Server โดยที่ RDO นี้ถูกออกแบบมาสำหรับการใช้ภาษา Visual Basic โดยเฉพาะ

**ODBC** มีประสิทธิภาพในการติดต่อกับฐานข้อมูลที่สัมพันธ์กันได้ดี ODBC เป็นการอินเทอร์เฟซในระดับล่าง จึงทำให้การควบคุมข้อมูลกับฐานข้อมูลในระดับล่าง ได้ดีเมื่อเปรียบเทียบกับ ADO ที่เป็นการอินเทอร์เฟซในระดับสูง

## 2.13 ความรู้พื้นฐานเกี่ยวกับ MySQL

MySQL (อ่านว่า “มาย-เอส-คิว-แอล”) จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมกันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกของอินเทอร์เน็ต สาเหตุเพราะว่า MySQL เป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูง เป็นทางเลือกใหม่จากผลิตภัณฑ์ระบบจัดการฐานข้อมูลในปัจจุบัน ที่มักจะเป็นการผูกขาดของผลิตภัณฑ์เพียงไม่กี่ตัว นักพัฒนาระบบฐานข้อมูลที่เคยใช้ MySQL ต่างยอมรับในความสามารถความเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมหาศาล ทั้งยังสนับสนุนการใช้งานบนระบบปฏิบัติการมากมาย ไม่ว่าจะเป็น Unix, OS/2, Mac OS หรือ Windows ก็ตาม นอกจากนี้ MySQL ยังสามารถใช้งานร่วมกับ Web Development Platform ทั้งหมด ไม่ว่าจะเป็น C, C++, Java, Perl, PHP, Python, Tcl หรือ ASP ก็ตาม ดังนั้นจึงไม่เป็นที่น่าแปลกใจเลยที่ว่า ทำไม MySQL จึงได้รับความนิยมอย่างมากในปัจจุบัน และมีแนวโน้มสูงยิ่งขึ้นต่อไปในอนาคต

MySQL จัดเป็นซอฟต์แวร์ประเภท Open Source Software สามารถดาวน์โหลด Source Code ต้นฉบับได้จากอินเทอร์เน็ต โดยไม่เสียค่าใช้จ่ายใดๆ การแก้ไขก็สามารถทำได้ตามความต้องการ MySQL ยึดถือสิทธิบัตรตาม GPL (GNU General Public License) ซึ่งเป็นข้อกำหนดของซอฟต์แวร์ประเภทนี้ส่วนใหญ่ โดยจะเป็นการชี้แจงว่า สิ่งใดทำได้ หรือทำไม่ได้สำหรับการใช้งานในกรณีต่างๆ ทั้งนี้ถ้าต้องการข้อมูลเพิ่มเติม หรือรายละเอียดของ GPL สามารถหาข้อมูลได้จากเว็บไซต์ <http://www.gnu.org/>

MySQL ได้รับการยอมรับและทดสอบเรื่องของความรวดเร็วในการใช้งาน โดยจะมีการทดสอบและเปรียบเทียบกับผลิตภัณฑ์ทางด้านฐานข้อมูลอื่นอยู่เสมอ มีการพัฒนาอย่างต่อเนื่อง โดยเริ่มตั้งแต่เวอร์ชันแรกๆ ที่ยังไม่ค่อยมีความสามารถมากนัก มาจนถึงทุกวันนี้ MySQL ได้รับการพัฒนาให้มีความสามารถมากยิ่งขึ้น รองรับข้อมูลจำนวนมหาศาล สามารถใช้งานหลายผู้ใช้ได้พร้อมๆ กัน (Multi-user) มีการออกแบบให้สามารถแต่งงานออก เพื่อช่วยการทำงานให้เร็วยิ่งขึ้น รองรับข้อมูลจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มหาศาล เพื่อช่วยการทำงานเร็วยิ่งขึ้น (Multi-threaded) วิธีและการเชื่อมต่อที่ดีขึ้น การกำหนดสิทธิและการรักษาความปลอดภัยของข้อมูลมีความรัดกุมน่าเชื่อถือยิ่งขึ้น เครื่องมือหรือโปรแกรมสนับสนุนทั้งของตัวเองและของผู้พัฒนาอื่นๆ มีมากยิ่งขึ้น นอกจากนี้สิ่งหนึ่งที่สำคัญคือ “MySQL ได้รับการพัฒนาไปในแนวทางตามข้อกำหนดมาตรฐาน SQL ดังนั้น เราสามารถใช้คำสั่ง SQL ในการทำงาน MySQL ได้” นักพัฒนาที่ใช้ SQL มาตรฐานอยู่แล้ว ไม่ต้องศึกษาคำสั่งเพิ่มเติม แต่อาจจะต้องเรียนรู้ถึง รูปแบบและข้อจำกัดบางอย่างโดยเฉพาะ ทั้งนี้ทั้งนั้น ทางทีมงานผู้พัฒนา MySQL มีเป้าหมายอย่างชัดเจนที่จะพัฒนาให้ MySQL มีความสามารถสนับสนุนตามข้อกำหนด SQL92 มากที่สุด และจะพัฒนาให้เป็นไปตามข้อกำหนด SQL99 ต่อไป

ทุกวันนี้มีการนำ MySQL ไปใช้ในระบบต่างๆ มากมาย ไม่ว่าจะเป็นระบบเล็กๆ ที่มีจำนวนตารางข้อมูลน้อย มีความสัมพันธ์ของข้อมูลในแต่ละตารางไม่ซับซ้อน เช่น ระบบฐานข้อมูลบุคคลในแผนกเล็กๆ ไปจนถึงระบบจัดการข้อมูลขนาดใหญ่ ที่ประกอบด้วยตารางข้อมูลมากมาย มีความสัมพันธ์ของข้อมูลในแต่ละตารางซับซ้อน เช่น ระบบสต็อกสินค้า ระบบบัญชีเงินเดือน เป็นต้น โดยเฉพาะอย่างยิ่งในปัจจุบัน มีการใช้ MySQL เป็น Database Server เพื่อการทำงานสำหรับ Web Database Application ในโลกของอินเทอร์เน็ตมากขึ้น สิ่งนี้เป็นคำตอบว่า “ทำไมคุณต้องเรียนรู้และใช้งาน MySQL ?”

### 2.13.1 สถาปัตยกรรมของ MySQL

สถาปัตยกรรม หรือ โครงสร้างภายในของ MySQL ก็คือ การออกแบบการทำงานในลักษณะของ Client/Server นั่นเอง ซึ่งประกอบด้วยส่วนหลักๆ 2 ส่วน คือ ส่วนของผู้ให้บริการ (Server) และ ส่วนของผู้ใช้บริการ (Client) โดยในแต่ละส่วนจะมีโปรแกรมสำหรับการทำงานตามหน้าที่ของตน

ส่วนของผู้ให้บริการ หรือ Server จะเป็นส่วนที่ทำหน้าที่บริหารจัดการระบบฐานข้อมูลในที่นี้ก็หมายถึงตัว MySQL Server นั่นเอง และเป็นที่ยึดเก็บข้อมูลทั้งหมด ข้อมูลที่เก็บไว้นี้มีข้อมูลที่จะใช้สำหรับการทำงานกับระบบฐานข้อมูล และข้อมูลที่เกิดจากการที่ผู้ใช้แต่ละคนสร้างขึ้นมา

ส่วนของผู้ใช้บริการ หรือ Client ก็คือผู้ใช้นั่นเอง โดยโปรแกรมสำหรับใช้งานในส่วนนี้ได้แก่ MySQL Client, Access, Web Development Platform ต่างๆ (เช่น Java, Perl, PHP, ASP เป็นต้น)

#### หลักการทำงานในลักษณะ Client/Server มีดังนี้

1. ที่ฝั่งของ Server จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอคอยการร้องขอการให้บริการจาก Client
2. เมื่อมีการร้องขอการให้บริการเข้ามา Server จะทำการตรวจสอบตามวิธีการของตน เช่น อาจจะมีการให้ผู้ให้บริการระบุชื่อและรหัสผ่าน และสำหรับ MySQL สามารถกำหนดได้ว่าจะอนุญาตหรือปฏิเสธ Client ใดๆ ในระบบที่จะเข้าใช้บริการอีกด้วย ซึ่งจะได้แสดงรายละเอียดในเรื่องต่อไป
3. ถ้าผ่านการตรวจสอบ Server ก็จะอนุมัติการให้บริการแก่ Client ที่ร้องขอการให้บริการนั้นๆ ต่อไปและถ้าในกรณีที่ไม่ได้รับการอนุมัติ Server ก็จะส่งข่าวสารความผิดพลาดแจ้งกลับไป Client ที่ร้องขอการให้บริการนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client หรือ Server อาจอยู่บนเครื่องเดียวกัน หรือแยกเครื่องกันก็ได้ ทั้งนี้ขึ้นอยู่กับลักษณะการทำงาน หรือการกำหนดของผู้บริหารระบบ ตามปกติถ้าเป็นการทำงานลักษณะ Web-based มีการใช้ฐานข้อมูลขนาดไม่ใหญ่นัก ตัว MySQL Server และ Client มักจะอยู่บนเครื่องเดียวกัน โดยเครื่องคอมพิวเตอร์ดังกล่าวจะต้องมีทรัพยากรเพื่อการทำงาน เช่น เนื้อที่ฮาร์ดดิสก์, RAM มากพอสมควร แต่สำหรับการทำงานจริง (Real-world Application) ก็มักจะแยก Client และ Server ออกเป็นคนละเครื่องกัน และสามารถรองรับงานได้ดีมากกว่า ดังนั้น ผู้บริหารระบบ หรือผู้กำหนดนโยบายสำหรับการทำงานเครือข่าย จะต้องคำนึงถึงเรื่องที่เกี่ยวข้องเหล่านี้ให้ดี เพื่อที่จะทำให้ระบบมีการทำงานบริการ ให้บริการแก่ผู้ใช้ได้อย่างมีประสิทธิภาพ และข้อมูลมีความปลอดภัยมากที่สุด

### 2.13.2 MySQL กับมาตรฐานภาษา SQL

จุดมุ่งหมายหนึ่งของทีมผู้พัฒนา MySQL ต้องการทำให้ MySQL เป็น DB Server ตามข้อกำหนด ANSI SQL โดยในขั้นแรกนี้อ้างอิงตาม ANSI SQL92 เป็นหลัก ความสามารถต่างๆ โดยส่วนใหญ่ครอบคลุมการใช้งาน อาจจะมีบางสิ่งที่ยังไม่สามารถทำได้ในปัจจุบัน หรือบางสิ่งก็มีแผนจะเพิ่มเติมเข้าไป ซึ่งได้กล่าวถึงรายละเอียดไปแล้วข้างต้น แต่ในส่วนนี้เราจะมาดูสำหรับความสามารถบางอย่างที่ MySQL เพิ่มเติมเข้ามาสำหรับการใช้งานนอกเหนือจากข้อกำหนด ANSI SQL ดังนี้

- ฟิลด์ประเภท MEDIUMINT, SET, ENUM และ BLOB รวมทั้ง TEXT ต่างๆ
- การกำหนดคุณลักษณะของฟิลด์ เช่น AUTO\_INCREMENT, BINARY, NULL, UNSIGNED และ ZEROFILL
- การเปรียบเทียบสตริงเป็นแบบ Case Insensitive การจัดเรียงข้อมูลใช้รูปแบบภาษา ISO-8859-1 (Latin1) เป็นค่า default
- ชื่อ database และชื่อตารางข้อมูลจะ Case Sensitive (โดยเฉพาะ Unix Platform)
- สามารถใช้คำสั่งของระบบปฏิบัติการในการสำรอง เปลี่ยนชื่อ ย้าย ลบ หรือคัดลอกตารางข้อมูลซึ่งเกี่ยวข้องกับไฟล์ .MYD, .MYI และ .frm ได้โดยตรง
- MySQL ไม่สนับสนุน Tablespace
- โอเปอเรเตอร์ LIKE สามารถใช้กับ Numeric Columns ได้
- การใช้ INTO OUTFILE และ STRAIGHT\_JOIN ในคำสั่ง SELECT
- สามารถใช้ SQL\_SMALL\_RESULT option ในคำสั่ง SELECT ได้
- คำสั่ง EXPLAIN SELECT เพื่อให้แสดงรายละเอียดของการเชื่อมกันของตารางข้อมูล
- สามารถใช้ TEMPORARY หรือ IF NOT EXISTS สำหรับการ CREATE TABLE ได้
- สามารถทำการ CHANGE ชื่อคอลัมน์, DROP ชื่อคอลัมน์ หรือ DROP INDEX, IGNORE หรือ RENAME ในการใช้คำสั่ง ALTER TABLE ได้
- สามารถทำการ RENAME TABLE ได้
- สามารถ ADD, ALTER, DROP หรือ CHANGE clauses หลายครั้งในคำสั่ง ALTER TABLE
- สามารถทำการ DROP TABLE ด้วยคีย์เวิร์ด IF EXISTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถทำการครอป (Drop) ตารางข้อมูลหลายๆ ตารางได้ด้วยคำสั่ง DROP TABLE เพียงครั้ง

เดียว

- การใช้ LIMIT clause ในคำสั่ง DELETE
- การใช้ DELAYED clause ในคำสั่ง INSERT และ REPLACE
- การใช้ LOW\_PRIORITY clause ในคำสั่ง INSERT, REPLACE, DELETE และ UPDATE
- การใช้งาน LOAD DATA INFILE ซึ่งไวชากรณัใช้ได้กับรูปแบบการ โหลดคดาต้า (Load Data)

ของ Oracle

- คำสั่ง ANALYZE TABLE, CHECK TABLE, OPTIMIZE TABLE และ REPAIR TABLE
- คำสั่ง SHOW
- คำสั่ง SET OPTION
- ฟังก์ชันหลายฟังก์ชันที่จำเป็น เช่น ฟังก์ชันสำหรับทำงานสตริง ซึ่งสามารถใช้ได้ทันที เนื่องจากมีชื่อและการทำงานที่เหมือนกันของ ANSI SQL และ ODBC
- สามารถใช้ || แทน OR และ && แทน AND ซึ่งเหมือนกับในภาษา C แต่ไม่สามารถใช้ในการเชื่อมสตริงได้ ซึ่ง MySQL ให้ใช้ฟังก์ชัน CONCAT() แทน
- คำสั่งในการ CREATE DATABASE หรือ DROP DATABASE
- การใช้โอเปอเรเตอร์ % โดยจะให้ผลลัพธ์เดียวกันการใช้ฟังก์ชัน MOD() ตัวอย่าง เช่น  $M \% N = \text{MOD}(M,N)$  เป็นต้น
- โอเปอเรเตอร์ =, <>, <=, <, >=, <<, >>, <=>, AND, OR หรือ LIKE สามารถใช้ในการเปรียบเทียบคอลัมน์ในการใช้คำสั่ง SELECT ได้ ทั้งนี้จะอยู่ทางด้านหน้า FROM ดังตัวอย่างเช่น SELECT id = 12345 AND name='somchai' FROM person;
- ฟังก์ชัน LAST\_INSERT\_ID()
- ฟังก์ชัน BIT\_COUNT(), CASE, ELT(), FROM\_DATS(), FORMAT(), IF(), PASSWORD(), ENCRYPT(), md5(), ENCODE(), DECODE(), PERIOD\_ADD(), PERIOD\_DIFF(), TO\_DAYS(), หรือ WEEKDAY()
- ฟังก์ชัน TRIM() เพื่อการตัดส่วนของสตริง ซึ่งใน ANSI SQL จะตัดได้เพียงหนึ่งตัวอักษรเท่านั้น
- GROUP BY function STD(), BIT\_OR() และ BIT\_AND()
- การมใช้คำสั่ง REPLACE แทนการใช้คำสั่ง DELETE ตามด้วยคำสั่ง INSERT
- คำสั่ง FLUSH flush\_option
- การใช้ตัวแปร sql statement ด้วย := ตัวอย่างเช่น SELECT @asum:=SUM (total), @cnt:=COUNT(\*), @asum/@cnt AS avg FROM person;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.13.3 ประเภทข้อมูลและตารางข้อมูลใน MySQL

ในส่วนนี้จะกล่าวถึงรายละเอียดของประเภทข้อมูลที่สามารถกำหนดให้แต่ละฟิลด์ ซึ่งสำหรับระบบฐานข้อมูล MySQL จะมีส่วนที่สนับสนุนกับมาตรฐาน และส่วนที่แตกต่าง เพิ่มเติมจากมาตรฐาน และประเภทตารางข้อมูลที่สามารถเลือกใช้ได้ใน MySQL

การศึกษาเรียนรู้ถึงประเภทข้อมูลใน MySQL จะทำให้เราเข้าใจ และสามารถทำการกำหนดประเภทของข้อมูลให้ถูกต้องเหมาะสมตามจุดประสงค์ หรือตามนโยบายการทำงานของเรา มิฉะนั้นแล้ว อาจก่อให้เกิดความเสียหาย หรือทำให้เกิดข้อจำกัดในเรื่องข้อมูลของการทำงานได้

การศึกษาในเรื่องของประเภทข้อมูลใน MySQL ก็มีจุดประสงค์เช่นเดียวกัน ซึ่งจะได้ศึกษาในรายละเอียดต่อไป

#### ประเภทข้อมูลใน MySQL

##### หลักการพิจารณาเลือกใช้ประเภทข้อมูล ดังนี้

1. **ประเภทข้อมูลหลัก** ได้แก่ ตัวเลข, ตัวอักษร, วันที่และเวลา, ข้อมูลไบนารี และอื่นๆ สำหรับข้อมูลประเภทตรรกะ MySQL ยังไม่สนับสนุนในเวอร์ชันปัจจุบัน แต่จะสนับสนุนในเร็วๆ นี้

2. **ในแต่ละประเภทข้อมูล จะมีประเภทย่อยๆ ลงไปอีก** เพื่อให้ผู้ใช้เลือกใช้ประเภทข้อมูลที่เหมาะสมเพราะนอกจากจะทำงานได้ถูกต้องแล้ว ยังเป็นการช่วยประหยัดเนื้อที่การเก็บข้อมูลอีกด้วย ซึ่งต่อไปเราจะได้ศึกษาเพิ่มเติมว่าในแต่ละประเภทข้อมูลเป็นอย่างไร มีขอบเขตข้อมูลอะไรบ้าง ใช้เนื้อที่ในการเก็บข้อมูลเท่าไร

3. **การเลือกใช้ประเภทข้อมูล** หากมีความต้องการที่จะสนับสนุนหรือใช้มาตรฐาน เช่น ODBC หรือต้องการ port ข้อมูลไปใช้ในระบบอื่น ควรเลือกใช้ประเภทข้อมูลที่สนับสนุนกับมาตรฐานเท่านั้น เพื่อป้องกันความผิดพลาดที่อาจจะเกิดขึ้นต่อไปในอนาคต

4. **ฟิลด์ที่ไม่มีค่าหรือไม่มีแนวโน้มจะเกี่ยวข้องกับการคำนวณ** ควรเลือกใช้ข้อมูลประเภทตัวอักษร เพราะจะประหยัดเนื้อที่ในการจัดเก็บจริง

5. **การเลือกใช้ประเภทของข้อมูลในแต่ละฟิลด์** นอกจากจะดูความเป็นไปได้ในปัจจุบันแล้ว จะต้องคำนึงและดูแนวโน้มความเป็นไปของข้อมูล หรือปริมาณของข้อมูลต่อไปในอนาคตด้วย เพื่อป้องกันข้อจำกัดของข้อมูลเมื่อมีการขยายตัวเพิ่มขึ้น

6. **การเลือกใช้ข้อมูลประเภทวันที่และเวลาใน MySQL จะต้องเลือกใช้อย่างระมัดระวัง** เนื่องจากมีข้อจำกัดและรายละเอียดที่แตกต่างไปจากมาตรฐาน เช่น จะสามารถใช้ข้อมูลเริ่มต้นได้ที่ปี ค.ศ. 1000 เท่านั้น เป็นต้น

##### ประเภทข้อมูลใน MySQL แบ่งออกเป็นกลุ่มได้ ดังนี้

#### 1. ประเภทข้อมูลสำหรับตัวเลข

ไว้สำหรับเก็บข้อมูลตัวเลข ซึ่งอาจจะใช้ในการคำนวณ หรือการจัดเรียงข้อมูลเปรียบเทียบกันในฟิลด์นั้นๆ ประกอบด้วยประเภทข้อมูลย่อยๆ ได้แก่ จำนวนเต็ม, จำนวนทศนิยม, จำนวนจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**1.1 TINYINT(M) [UNSIGNED] [ZEROFILL]**

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -128 ถึง 127 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 255 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 1 ไบต์

**1.2 SMALLINT(M) [UNSIGNED] [ZEROFILL]**

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -32768 ถึง 32767 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 65535 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 2 ไบต์

**1.3 MEDIUMINT(M) [UNSIGNED] [ZEROFILL]**

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -8388608 ถึง 8388607 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 16777215 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 3 ไบต์

**1.4 INT(M) [UNSIGNED] [ZEROFILL]****หรือ INTEGER(M) [UNSIGNED] [ZEROFILL]**

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -2147483648 ถึง 2147483647 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 4294967295 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 4 ไบต์

**1.5 BIGINT(M) [UNSIGNED] [ZEROFILL]**

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -9223372036854775808 ถึง 9223372036854775807 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 18446744073709551615 ข้อมูลชนิดนี้ ใช้เนื้อที่เก็บข้อมูลขนาด 8 ไบต์

**1.6 FLOAT(precision) [ZEROFILL]**

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย precision เป็นค่าความละเอียดทศนิยม ซึ่งแบ่งเป็นชนิด single (มีค่าตั้งแต่ 0-24) และแบบ double (ตั้งแต่ 25-53) ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 4 หรือ 8 ไบต์ โดยแบบ single จะใช้ 4 ไบต์ และแบบ double จะใช้ 8 ไบต์ ชนิดข้อมูลประเภทนี้สำหรับกรณีใช้กับ ODBC มาตรฐาน

**1.7 FLOAT(M,D) [ZEROFILL]**

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -3.402823466E+38 ถึง -1.175494351E-38, 0 และ 1.175494351E-39 ถึง 3.402823466E+38 ค่า M เป็นจำนวนหลักที่ต้องการแสดงผล และ D เป็นจำนวนจุดทศนิยม ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 4 ไบต์

**1.8 DOUBLE(M,D) [ZEROFILL]****หรือ DOUBLE PRECISION(M,D) [ZEROFILL]****หรือ REAL(M,D) [ZEROFILL]**

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -1.7976931348623157E+308 ถึง -2.2250738585072014E-308, 0 และ 2.2250738585072014E-308 ถึง 1.7976931348623157E+308 ค่า M เป็นจำนวนหลักที่ต้องการแสดงผล และ D เป็นจำนวนจุดทศนิยม ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูล ขนาด 8 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.9 DECIMAL[(M[,D])] [ZEROFILL]

#### หรือ NUMERIC[M,D] [ZEROFILL]

ข้อมูลชนิดตัวเลขชนิด unpacked คืออนุญาตให้สามารถเก็บข้อมูลตัวอักษรเข้าไปด้วย โดยตัวอักษรหนึ่งตัวแทนแต่ละหลัก สามารถใช้ตัวเลขได้เท่ากับแบบ DOUBLE ใช้เนื้อที่เก็บข้อมูลเท่ากับ M ไบต์ (D+2, ถ้า  $M < D$ )

## 2. ประเภทข้อมูลสำหรับวันที่และเวลา

### 2.1 DATE

ข้อมูลชนิดวันที่ ตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 ถึง 31 ธันวาคม ค.ศ. 9999 การแสดงผลวันที่อยู่ในรูปแบบ 'YYYY-MM-DD' ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 3 ไบต์

### 2.2 DATETIME

ข้อมูลชนิดวันที่และเวลา ตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 เวลา 00:00:00 ถึง 31 ธันวาคม ค.ศ. 9999 เวลา 23:59:59 การแสดงผลวันที่และเวลาอยู่ในรูปแบบ 'YYYY-MM-DD HH:MM:SS' ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 8 ไบต์

### 2.3 TIMESTAMP[M]

ค่า timestamp เป็นค่าตัวเลขที่นับจำนวนวินาทีตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 เวลา 00:00:00 เป็นต้นมาโดยมีจุดสิ้นสุดประมาณ ค.ศ. 2037 การแสดงผลค่า timestamp อยู่ในรูปแบบ YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD หรือ YYMMDD ค่า M คือจำนวนตัวเลขที่บรรจุ ซึ่งอาจจะเป็น 14, 12, 8 หรือ 6 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 4 ไบต์

### 2.4 TIME

ข้อมูลประเภทเวลา สามารถเป็นได้ตั้งแต่ '-838:59:59' ถึง '838:59:59' แสดงผลในรูปแบบ HH:MM:SS ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 3 ไบต์

### 2.5 YEAR[(2/4)]

ข้อมูลประเภทปี ค.ศ. โดยสามารถเลือกว่าจะใช้แบบ 2 หรือ 4 หลัก (ค่าโดยปริยายจะเป็น 4 หลัก) ถ้าเลือกใช้แบบ 4 หลัก จะใช้ได้ตั้งแต่ปี ค.ศ. 1901 ถึง 2155 ถ้าเลือกใช้แบบ 2 หลัก จะใช้ได้ตั้งแต่ปี ค.ศ. 1970 ถึง 2069 แสดงในรูปแบบ 70 ถึง 69 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 1 ไบต์

## 3. ประเภทข้อมูลสำหรับตัวอักษร

ข้อมูลประเภทตัวอักษรถือว่าเป็นสิ่งที่ใช้กันมากที่สุด เพราะแทบจะครอบคลุมข้อมูลที่เป็นไปได้ทั้งหมดการจัดเรียงข้อมูลจะเป็นในลักษณะไม่คำนึงตัวใหญ่ตัวเล็ก (case-insensitive) ซึ่งเมื่อเปรียบเทียบกับข้อมูลประเภทไบนารี (BLOB) แล้วจะพบว่าคล้ายกันมาก จะแตกต่างกันที่ BLOB จะมีการจัดเรียงข้อมูลเป็น caseinsensitive เท่านั้น แต่ข้อมูลประเภทตัวอักษรก็มีรายละเอียดเพิ่มเติมไปอีก โดยมีรายละเอียดดังนี้

### 3.1 [NATIONAL] CHAR(M) [BINARY]

ข้อมูลประเภทสตริงที่จำกัดขนาดความกว้าง (ไม่สามารถปรับขนาดได้) โดยขนาดความกว้างเป็นได้ตั้งแต่ 1 ถึง 255 ตัวอักษร ตามปกติเมื่อมีการเรียงข้อมูล จะเป็นลักษณะ case-sensitive คือคำนึงถึงตัวเล็กตัวใหญ่การระบุชนิดข้อมูลย่อว่าเป็น BINARY จะเป็นลักษณะไม่คำนึงตัวเล็กตัวใหญ่ คำนึงเมื่อมีการเรียงข้อมูลก็จะให้ผลต่างจาก CHAR ธรรมดา การเก็บข้อมูลก็ใช้ไปตตามจำนวนตัวอักษรที่ระบุ

### 3.2 [NATIONAL] VARCHAR(M) [BINARY]

ข้อมูลประเภทนี้ก็คล้ายกับแบบ CHAR ต่างกันตรงที่ VARCHAR จะสามารถปรับขนาดตามข้อมูลที่เก็บในฟิลด์ ขนาดความกว้างอยู่ตั้งแต่ 1 ถึง 255 การเก็บข้อมูลจะเท่ากับของข้อมูลจริงในฟิลด์ๆ + 1 ไบต์

### 3.3 TINYTEXT

ข้อมูลประเภทนี้สามารถใช้ความกว้างข้อมูลได้สูงสุด 255 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 1 ไบต์

### 3.4 TEXT

ข้อมูลประเภทนี้สามารถใช้ความกว้างข้อมูลได้สูงสุด 65,535 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 2 ไบต์

### 3.5 MEDIUMTEXT

ข้อมูลประเภทนี้สามารถใช้ความกว้างข้อมูลได้สูงสุด 16,777,215 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 3 ไบต์

### 3.6 LONGTEXT

ข้อมูลประเภทนี้สามารถใช้ความกว้างข้อมูลได้สูงสุด 4,294,967,295 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 4 ไบต์

### 3.6 ENUM('value1', 'value2', ...)

ข้อมูลประเภทระบุเฉพาะค่าที่ต้องการ หรือถ้าไม่มีจะให้ค่า NULL สามารถกำหนดค่าได้ถึง 65,535 ค่า และใช้เนื้อที่เก็บข้อมูลตามจำนวนค่าที่ระบุ

### 3.7 SET('value1', 'value2', ...)

ข้อมูลประเภทเซต ประกอบด้วยข้อมูลตั้งแต่ไม่มีค่า หรือมีค่าตามสมาชิกที่กำหนด สามารถมีจำนวนสมาชิกในเซตได้ทั้งสิ้น 64 ตัว และใช้เนื้อที่เก็บข้อมูลตามจำนวนสมาชิกที่ระบุ

## 4. ประเภทข้อมูลสำหรับไบนารี (BLOB: Binary Large Object)

จากที่กล่าวไปแล้วถึงความคล้ายกันของแต่ละข้อมูลประเภทตัวอักษรและข้อมูลประเภทไบนารี คำนึงประเภทข้อมูลย่อและการเก็บข้อมูลจึงไม่แตกต่างกัน แต่ตามปกติเรามักใช้ BLOB สำหรับเก็บข้อมูลที่ไม่มีเป็นภาษามนุษย์ ข้อมูลไบนารี ได้แก่ รูปภาพ, ไฟล์ข้อมูล หรือข้อมูลที่ประกอบด้วยตัวอักษรพิเศษ ข้อมูลที่กำหนดด้วยรหัสควบคุม เป็นต้น

#### 4.1 TINYBLOB

สำหรับข้อมูลไบนารี สามารถใช้ความกว้างข้อมูลได้สูงสุด 255 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 1 ไบต์

#### 4.2 BLOB

สำหรับข้อมูลไบนารี สามารถใช้ความกว้างข้อมูลได้สูงสุด 65,535 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 2 ไบต์

#### 4.3 MEDIUMBLOB

สำหรับข้อมูลไบนารี สามารถใช้ความกว้างข้อมูลได้สูงสุด 16,777,215 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 3 ไบต์

#### 4.4 LONGBLOB

สำหรับข้อมูลไบนารี สามารถใช้ความกว้างข้อมูลได้สูงสุด 4,294,967,295 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 4 ไบต์

### 2.13.3 คำสั่งต่างๆ ใน MySQL

ในบทนี้จะได้เรียนรู้เกี่ยวกับคำสั่งต่างๆ ใน MySQL ซึ่งเป็นการรวมคำสั่งทั้งหมด ไม่ว่าจะเป็นคำสั่งที่ใช้สำหรับการกำหนด (Data Definition Language), คำสั่งที่ใช้สำหรับทำงานข้อมูล (Data Manipulation Language) และคำสั่งที่ใช้สำหรับการควบคุมการทำงาน (Data Control Language)

#### 1. คำสั่ง CREATE DATABASE

```
CREATE DATABASE demo
```

เป็นคำสั่งสร้าง database ชื่อ db\_name ขึ้นมาใหม่ ในกรณีที่มี database นั้นอยู่แล้ว ก็จะปรากฏข่าวสารแสดงความผิดพลาด (Error Message) Can't create database 'demo'. Database exists

#### 2. คำสั่ง DROP DATABASE

```
DROP DATABASE demo
```

เป็นคำสั่งลบ database ชื่อ db\_name ออกจากระบบ

#### 3. คำสั่ง CREATE TABLE

```
CREATE TABLE phonebook (id INT (4) NOT NULL
AUTO_INCREMENT, fname VARCHAR (35), lname VARCHAR (35),
email VARCHAR (50), office VARCHAR (200), province CHAR (2),
PRIMARY KEY(id), UNIQUE(id), INDEX(id))
```

โดยทั่วไปขนาดของตารางข้อมูลหนึ่งๆ จะมีขนาดไม่เกิน 2G หรือ 4G ขึ้นอยู่กับระบบปฏิบัติการ เราสามารถทำให้ประหยัดเนื้อที่ได้อีกโดยการบีบอัด (PACK) ข้อมูลคีย์นี้ ซึ่งสามารถให้บีบอัดโดย กำหนดค่า PACK\_KEYS=1 การบีบอัดนี้จะทำให้การทำงานช้าลง แต่ก็ประหยัดเนื้อที่ขึ้น และตามปกติถ้า ผู้ใช้ไม่ได้กำหนดให้บีบอัด ข้อมูลที่เป็นสตริงหรือตัวอักษรก็จะได้รับการบีบอัดข้อมูลอยู่แล้ว สำหรับ ข้อมูลตัวเลข ค่าการทำงานปกติจะไม่รับการบีบอัด เว้นแต่กำหนดการบีบอัดด้วยการใช้ UNION จะใช้ กรณีที่ต้องการรวมตารางข้อมูลหลายๆ ตารางเข้าด้วยกันเป็นอันเดียว ซึ่งจะทำงานกับตารางประเภท MERGE เท่านั้น

#### 4. คำสั่ง ALTER TABLE

```
ALTER TABLE phonebook ADD tel VARCHAR(30)
```

ALTER TABLE เป็นคำสั่งสำหรับการปรับเปลี่ยนแก้ไขโครงสร้างของตารางข้อมูล เช่น การ เพิ่ม-ลบฟิลด์, การสร้าง-ลบคีย์, การเปลี่ยนประเภทข้อมูลของฟิลด์, การเปลี่ยนชื่อฟิลด์ เป็นต้น

#### 5. คำสั่ง RENAME TABLE

```
RENAME TABLE phonebook to phone
```

เป็นคำสั่งเพื่อเปลี่ยนชื่อตารางข้อมูลเป็นชื่อใหม่

#### 6. คำสั่ง DROP TABLE

```
DROP TABLE phonebook
```

เป็นคำสั่งเพื่อให้ลบตารางข้อมูลออกจากระบบ สำหรับออปชัน RESTRICT และ CASCADE ยัง ไม่สามารถงานได้ในปัจจุบัน เนื่องจากเป็นความสามารถที่มีแผนจะรองรับในอนาคต สำหรับการ ใช้คำสั่ง DROP TABLE จะต้องให้ความระมัดระวังในการใช้งานเป็นอย่างยิ่ง โดยเฉพาะการทำงานกับตารางข้อมูล ประเภท nontransaction เพราะจะทำให้การ commit การทำงาน โดยอัตโนมัติ

#### 7. คำสั่ง OPTIMIZE TABLE

```
OPTIMIZE TABLE phonebook
```

เป็นคำสั่งสำหรับจัดระเบียบข้อมูลให้เป็นระเบียบ หรือการทำดีแฟร็กเมนต์นั่นเองนั่นเอง เพราะ ถ้าตารางข้อมูลมีการใช้งานมานาน มีการใช้งานที่ทำให้ข้อมูลมีการเปลี่ยนแปลงความยาว หรือเปลี่ยน ขนาดของเนื้อที่เก็บข้อมูล โดยเฉพาะกับข้อมูลประเภทที่บีบอัดไปตามข้อมูล เช่น VARCHAR, BLOB หรือ TEXT การใช้คำสั่ง OPTIMIZE TABLE ก็จะช่วยให้มีการจัดระเบียบข้อมูลให้ดียิ่งขึ้น หากปล่อย

ระยะเวลาหลายๆ จะเกิดช่องว่างระหว่างข้อมูล การอ่านข้อมูลของสื่อจะไม่ราบรื่น ซ้ำ เนื่องจากเสียเวลาในการกระโดดข้ามช่องว่างนั้น นั่นเอง

## 8. คำสั่ง DELETE

```
DELETE FROM phonebook WHERE id=4
```

เป็นคำสั่งสำหรับลบข้อมูลจากตาราง โดยสามารถกำหนดเงื่อนไขสำหรับการลบได้ ซึ่งถ้าไม่ได้ระบุเงื่อนไข จะเป็นการลบข้อมูลทั้งหมดออกจากตาราง และถ้าอยู่ใน AUTOCOMMIT mode (กรณีปกติทั่วไป) จะไม่สามารถทำการยกเลิกการลบครั้งนี้ได้

## 9. คำสั่ง TRUNCATE

```
TRUNCATE phonebook
```

เป็นคำสั่งที่ให้ผลเช่นเดียวกับ DELETE FROM table\_name แต่มีข้อแตกต่างดังนี้

- จะทำการครอปตารางข้อมูลก่อน แล้วทำการสร้างใหม่ ซึ่งจะให้ผลที่เร็วกว่าการลบตารางทั้งหมด
- การทำงานเป็น non-transaction ไม่สามารถทำการยกเลิกได้

## 10. คำสั่ง SELECT

```
SELECT id, fname FROM phonebook ORDER BY fname DESC
```

เป็นคำสั่งสำหรับการแสดงผล หรือการดึงข้อมูลจากตาราง โดยสามารถกำหนดเงื่อนไข รูปแบบ และวิธีการ ได้อย่างหลากหลาย การดึงข้อมูลสามารถระบุว่าจะเลือกจากตารางเดียว หรือหลายตารางก็ได้

การใช้ LIMIT rows ก็จะช่วยกำหนดให้ทำการดึงข้อมูลจากตาราง ได้ไม่เกินจำนวนแถว rows ที่ระบุ

การใช้คำสั่งคีย์เวิร์ดตามหลัง SELECT จะต้องใช้ให้ถูกต้อง มีบางคีย์เวิร์ดที่ต้องใช้เรียงลำดับกัน เช่น HAVING clause ต้องใช้ตามหลัง GROUP BY clause และใช้ก่อน ORDER BY clause เป็นต้น

ตามปกติหากไม่ได้ระบุวิธีการจัดเรียงข้อมูล ก็จะเป็นการเรียงจากน้อยไปมาก (Ascending) หรือคีย์เวิร์ด ASC แต่ถ้าต้องการเรียงจากมากไปน้อย (Descending) ให้ใช้คีย์เวิร์ด DESC

การใช้คีย์เวิร์ด HAVING จะใช้ในกรณีร่วมกับ GROUP BY ซึ่งก็เป็นการกำหนดเงื่อนไขในการดึงข้อมูล ด้วยคำสั่ง SELECT

```
SELECT id,fname FROM phonebook GROUP BY fname HAVING id>2
```

## 11. คำสั่ง INSERT

```
INSERT INTO phonebook (id, fname, lname, email, office, province)
```

```
VALUES (6, 'sineenat', 'phradmali', 'sineenat@health.moph.go.th', 'ศทตจ.', '12')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในหน่วยงานเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ผ่านการพิจารณา  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคำสั่งสำหรับการเพิ่มแถวข้อมูล หรือระเบียบใหม่เข้าตารางข้อมูล

## 12. คำสั่ง UPDATE

```
UPDATE phonebook SET office= 'ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร' WHERE id=6
```

เป็นคำสั่งเพื่อทำการเปลี่ยนแปลงข้อมูลในตารางให้เป็นตามที่ต้องการ

## 13. คำสั่ง USE

```
USE demo
```

USE เป็นคำสั่งสำหรับเลือกใช้ database SHOW TABLES แสดงตารางที่อยู่ใน database demo ทั้งหมด

## 14. คำสั่ง FLUSH

```
FLUSH TABLES phonebook
```

เป็นคำสั่งที่ใช้สำหรับการลบหน่วยความจำภายใน (Internal Cache) ที่ใช้งาน ทั้งนี้การใช้คำสั่ง FLUSH จะอนุญาตเฉพาะ user ที่มีสิทธิ์ (Privilege) เท่านั้น

**Flush\_option สามารถระบุได้ดังนี้**

**HOSTS** ทำการล้าง hose cache table เพื่อปรับปรุงข้อมูลของ host หรือ client ที่คอนเน็คต์เข้าใช้งาน ซึ่งอาจมีการเปลี่ยนแปลง IP address หรืออื่นๆ โดยเมื่อทำการ flush เรียบร้อยแล้ว ก็จะอนุญาตให้ host นั้นๆ ทำการคอนเน็คต์เข้ามาใหม่อีกครั้งหนึ่ง

**LOGS** ปิดและเปิด log files ใหม่

**PRIVILEGES** ทำการ reload สิทธิต่างๆ ของ user จาก grant tables ของ MySQL Server

**TABLES** ปิด tables ทั้งหมด หากตารางใดยังถูกเปิดอยู่ ก็จะถูก force ให้ปิดลง

## 15. คำสั่ง SHOW

```
SHOW DATABASES
```

เป็นคำสั่งสำหรับแสดงรายละเอียด (Information) ของ databases, tables, columns หรือ สถานะต่างๆของ Server

### แสดงรายละเอียดของ Databases, Tables, Columns และ Indexes

**SHOW DATABASES** แสดงรายชื่อ databases ทั้งหมดในระบบฐานข้อมูล

**SHOW TABLES** แสดงรายชื่อ tables ทั้งหมดของ current database

**SHOW OPEN TABLES** แสดงรายชื่อ tables ที่ถูกเปิดอยู่ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>SHOW COLUMNS</b>	แสดงรายละเอียดฟิลด์ทั้งหมดใน table นั้น (ให้ผลลัพธ์เช่นเดียวกับคำสั่ง DESCRIBE)
<b>SHOW FIELDS</b>	ให้ผลลัพธ์เช่นเดียวกับ SHOW COLUMNS
<b>SHOW INDEX</b>	แสดงรายละเอียดว่ามีดัชนีอะไรบ้างสำหรับ table นั้นๆ
<b>SHOW KEYS</b>	ให้ผลลัพธ์เช่นเดียวกับ SHOW INDEX

## 16. คำสั่ง GRANT และ REVOKE

```
GRANT ALL PRIVILEGES ON demo TO user1@localhost WITH GRANT OPTION
```

```
REVOKE ALL PRIVILEGES ON demo FROM user1@localhost
```

- คำสั่ง **GRANT** เป็นคำสั่งสำหรับการกำหนดสิทธิ์หรือความสามารถใดๆ ให้กับ user
- คำสั่ง **REVOKE** หรือคำสั่งที่ตรงกันข้ามกับ GRANT คือเป็นการถอนหรือยกเลิกสิทธิ์ใดๆ จาก

user

การใช้คำสั่ง GRANT หรือ REVOKE จะอนุญาตเฉพาะผู้บริหารระบบในการกำหนดสิทธิ์ต่างๆ สำหรับผู้ใช้

**MySQL จะแบ่งสิทธิ์ (Privilege) เป็น 4 ระดับ คือ**

### 1. Global level

ระดับบนสุด ครอบคลุมทุก database รายละเอียดสำหรับสิทธิ์ของระดับนี้จะถูกเก็บไว้ในตาราง mysql.user

### 2. Database level

ระดับ database ครอบคลุมเฉพาะใน database อันใดอันหนึ่ง เช่น ความสามารถในการกระทำใดๆ ภายใน database นั้นๆ รายละเอียดสำหรับสิทธิ์ของระดับนี้ จะถูกเก็บไว้ในตาราง mysql.db และ mysql.host

### 3. Table level

ระดับ table ครอบคลุมเฉพาะภายในตารางนั้นๆ เช่น ความสามารถในการกระทำกับ table นั้นๆ สามารถทำการ select, insert, delete, update ได้หรือไม่ เป็นต้น รายละเอียดสำหรับสิทธิ์ของระดับนี้ จะถูกเก็บอยู่ในตาราง mysql.tables\_priv

### 4. Column level

ระดับ column ครอบคลุมเฉพาะฟิลด์ใดฟิลด์หนึ่งภายในตารางข้อมูล รายละเอียดสำหรับสิทธิ์ของระดับนี้ จะถูกเก็บอยู่ในตาราง mysql.columns\_priv

**รายละเอียด priv\_type มีดังต่อไปนี้**

ALL PRIVILEGES	FILE	RELOAD
ALTER	INDEX	SELECT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CREATE	INSERT	SHUTDOWN
DELETE	PROCESS	UPDATE
DROP	USAGE	

- เราสามารถใช้ ALL หรือ ALL PRIVILEGES ได้อย่างใดอย่างหนึ่ง

- USAGE มีความหมายเท่ากับ no privileges โดยใช้กรณีที่เราต้องการ create user และกำหนดให้มีสิทธิ์เป็น no privileges

- การถอนสิทธิ์การ GRANT ของ user ใดๆ ให้ใช้ priv\_type เป็น GRANT OPTION

**ตัวอย่าง:** REVOKE GRANT OPTION ON ... FROM ...;

- การใช้ priv\_type สำหรับ table: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, GRANT, INDEX และ ALTER

- การใช้ priv\_type สำหรับฟิลด์ SELECT, INSERT, UPDATE

- การกำหนดสิทธิ์ระดับ database ใช้ ON \*.\*

- สำหรับ user\_name สามารถระบุได้ถึงโฮสต์ที่ใช้งาน เช่น mysql@localhost หรือ root@'%'

### 2.13.4 การรักษาความปลอดภัย

#### การกำหนดสิทธิของผู้ใช้งาน

หลักการที่ MySQL ใช้ เรียกว่า Access Control Lists (ACLs) จะมีการตรวจสอบสิทธิผู้ใช้งานว่าแต่ละคนมีสิทธิจะทำอะไรได้บ้าง กับฐานข้อมูลใดบ้าง ซึ่งสิทธิเหล่านี้จะถูกบันทึกเอาไว้ในตารางชื่อ mysql

ข้อควรกระทำและไม่ควรกระทำในการป้องกันหรือรักษาความปลอดภัยของข้อมูล

#### 1. ห้ามอนุญาตให้ใครก็ตาม ที่ไม่ใช่ผู้ดูแลระบบ มีสิทธิเข้าไปเรียกดูหรือใช้งานตารางชื่อ mysql

เพราะ mysql เป็นแหล่งที่เก็บข้อมูลเกี่ยวกับสิทธิต่าง ๆ

#### 2. อย่างกำหนดสิทธิให้กับผู้ใช้งานใด ๆ มากเกินความจำเป็น

ผู้ใช้แต่ละคนควรจะได้รับสิทธิแตกต่างกัน และควรมีสิทธิจำกัดเป็นกรณี ๆ ไป เช่น หากเป็นผู้ใช้ทั่ว ๆ ไป ควรจะมีสิทธิใช้งานแค่ฐานข้อมูลใดฐานข้อมูลหนึ่ง หรือ อาจจะมีสิทธิแค่ตารางใดตารางหนึ่งในแต่ละฐานข้อมูล

#### ระดับสิทธิการเรียกใช้ข้อมูลมี 4 ระดับคือ

- 2.1 ระดับโฮสต์ (host level)
- 2.2 ระดับฐานข้อมูล (db level)
- 2.3 ระดับตารางข้อมูล (table level)
- 2.4 ระดับคอลัมน์ข้อมูล (column level)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. ระวังผู้ใช้ที่ชื่อ root

root คือ ผู้สร้างฐานข้อมูล มีสิทธิต่าง ๆ มาก ข้อแนะนำว่าอย่าให้มีผู้ใช้ชื่อ root อยู่ในสารบบของ MySQL โดยเราสามารถกำหนดผู้ใช้นี้ขึ้นมาให้มีสิทธิเทียบเท่า root และอีกประการคือ รายชื่อผู้ใช้ของ MySQL กับของ Server ที่ใช้งาน จะเป็นคนละชุดกัน

### 4. ระวังการตั้งรหัสผ่านของผู้ใช้

การตั้งรหัสผ่านควรหลีกเลี่ยงคำศัพท์ทั่วไปที่สามารถค้นหาได้จากดิกชันนารี, หลีกเลี่ยงชื่อเล่น หรือ ชื่อสถานที่ที่ผู้บุกรุกอาจจะคาดเดาได้

### 5. หากไม่จำเป็นต้องใช้ Telnet ก็ปิดไปดีกว่า

ในกรณีที่ Server ของเราสามารถรองรับการทำงานของโปรแกรม Telnet แต่ในทางปฏิบัติไม่มีความจำเป็นที่ต้องอนุญาตให้ใครมาใช้โปรแกรมนี้ ก็ควรยกเลิกการทำงาน

### 6. อย่าเชื่อมือผู้ใช้งานนัก

เพราะหากผู้ใช้เผลอไปใช้คำสั่ง Drop database mysql ถ้าเราพัฒนาโปรแกรมขึ้นมาใช้ ควรจะมีระบบตรวจสอบคำสั่งนี้ โดยห้ามใช้คำสั่งนี้อย่างเด็ดขาด

## 2.13.5 ระดับสิทธิของผู้ใช้

### 1. ระดับโฮสต์ (host level) คือ ระดับนี้จะจำกัดสิทธิว่า ผู้ใช้สามารถติดต่อเข้ามายัง MySQL ได้

จากที่ไหนบ้าง หรือจาก โฮสต์อะไรทั้งนี้เพราะถ้า MySQL ของเราอยู่ในเครื่องคอมพิวเตอร์ที่เชื่อมโยงกับเครื่องอื่น ๆ หรือเชื่อมโยงกับเครือข่าย Internet การติดต่อสื่อสารก็ย่อมกระทำจากที่ไหนก็ได้ในโลก ผ่านทางโพรโทคอล TCP/IP

ลักษณะการตรวจสอบสิทธิการใช้งาน เมื่อเครื่องลูกข่ายที่ร้องขอติดต่อเข้ามา จะต้องแจ้งให้ทราบว่ามีมาจากไหน ชื่อโฮสต์อะไร หมายเลข IP Address เท่าไร เมื่อ MySQL รับทราบการร้องขอ ก็จะนำข้อมูลที่ได้รับแจ้งไป ตรวจสอบว่าโฮสต์นั้นมีสิทธิหรือไม่หรือมีสิทธิอะไรบ้าง

ลักษณะชื่อโฮสต์ที่สามารถกำหนดได้

ลักษณะชื่อโฮสต์ที่สามารถกำหนดได้	หมายความว่าติดต่อมาจาก...
Localhost	เครื่องเดียวกันกับ MySQL server
%	เครื่องไหนก็ได้
%.cc-moph	เครื่องใดก็ตามที่อยู่ในโดเมน cc-moph
203.157%	เครื่องใดก็ตามที่มี IP address เริ่มจาก 203.157

**ตัวอย่าง** กำหนดสิทธิให้กับผู้ใช้ที่ชื่อว่า train ก็ระบุในลักษณะ train@host เช่น อนุญาตให้ train สามารถติดต่อมายัง MySQL ได้จากเครื่องหมายเลข 203.157.4.27 ก็ระบุเป็น train@203.157.4.27 ดังนั้น ถ้าติดต่อ MySQL โดยใช้สคริปต์ PHP ก็ต้องเขียนคำสั่งในรูปแบบดังนี้

```
mysql_connect("MySQLHost","train","รหัสผ่าน");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งกำหนดสิทธิการใช้งาน GRANT มีรูปแบบดังนี้

```
GRANT สิทธิต่างๆ ON ชื่อฐานข้อมูล.ชื่อตาราง TO ชื่อผู้ใช้งาน@โฮสต์[IDENTIFILE BY 'รหัสผ่าน']
```

```
GRANT ALL ON *.* TO train@203.157.4.27 IDENTIFILE BY 'aBcXyz';
```

สิทธิต่าง คือ สิทธิ (privilege) ที่จะสามารถทำได้เช่น การเรียกดูข้อมูล (SELECT), การลบข้อมูล (DELETE) หรือ จะระบุสิทธิต่าง ๆ เป็น ALL ก็ได้ซึ่งหมายความว่าสามารถทำได้ทุกอย่าง

## 2. ระดับฐานข้อมูล (db level)

ทั้งนี้ขึ้นอยู่กับว่ามีการระบุสิทธิต่าง ๆ ตามหลังคำสั่ง GRANT ถ้าระบุคำว่า ALL ก็หมายความว่าให้สิทธิในกรณีเรียกใช้ทุกคำสั่ง เช่น

```
GRANT SELECT ON *.* TO User1@localhost IDENTIFILE BY 'Password';
```

หมายความว่า อนุญาตให้ user1 สามารถใช้คำสั่ง SELECT สำหรับเรียกดูข้อมูลทุกตารางทุกฐานข้อมูล

## 3. ระดับตารางข้อมูล (table level)

สิทธิระดับนี้จะสัมพันธ์กับสิทธิระดับฐานข้อมูล คือ ถ้าจะให้สิทธิในการใช้คำสั่งกับทุก ๆ ตารางในฐานข้อมูลก็ระบุด้วยเครื่องหมาย \* เช่น

```
GIS.* หมายถึงอนุญาตให้มีสิทธิใช้คำสั่งกับทุกตารางที่มีอยู่ในฐานข้อมูลชื่อ GIS
```

```
GRANT SELECT ON db1.table1 TO User1@localhost IDENTIFILE BY 'aBcXyz';
```

หมายความว่าอนุญาตให้ user1 เรียกใช้เฉพาะคำสั่ง SELECT เพื่อดึงข้อมูลจากตารางชื่อ table1 ซึ่งอยู่

## 4. ระดับคอลัมน์ข้อมูล (Column level)

เป็นสิทธิระดับที่ลดลงไปจากสิทธิระดับอื่น ๆ อีก โดยมีความสัมพันธ์กับสิทธิระดับฐานข้อมูล และระดับตารางด้วย เพียงแค่วิธีการกำหนดสิทธิต้องระบุในตารางที่ชื่อ column\_priv ซึ่งตารางนี้อยู่ในฐานข้อมูลmysql อีกที ฉะนั้นจึงต้องใช้คำสั่ง SQL ส่งไปตรง ๆ ที่ตาราง column\_priv มิใช่กำหนดโดยการ ใช้คำสั่ง GRANT เหมือนกับสิทธิระดับอื่น ๆ

**สิทธิต่าง ๆ หรือ privilege** หมายความว่า เมื่อเราอนุญาตให้ผู้ใช้เข้ามายังฐานข้อมูล, ตารางข้อมูล, หรือเข้ามายัง MySQL ได้แล้ว สามารถใช้คำสั่ง SQL ทำอะไรได้บ้าง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิทธิต่างๆ	การทำงาน	ตัวอย่างการใช้งาน
SELECT	เรียกดูรายการข้อมูล	SELECT * FROM table1
INSERT	เพิ่มรายการข้อมูล	INSERT INTO table1 ...
UPDATE	ปรับปรุงแก้ไขรายการข้อมูล	UPDATE table1 SET ...
DELETE	ลบรายการข้อมูล	DELETE FROM table1
ALTER	แก้ไขโครงสร้างตารางข้อมูล	ALTER TABLE table1 ADD INDEX index1(name)
CREATE	สร้างตารางหรือฐานข้อมูลขึ้นมาใหม่	CREATE DATABASE database1 CREATE TABLE table1
DROP	ลบตารางหรือฐานข้อมูล	DROP DATABASE database1 DROP TABLE table1
RELOAD	สั่งให้ server เริ่มค่าต่างๆ ใหม่	FLUSH HOST (ปลดล็อกในกรณีที่ server มีการล็อก ป้องกันการใช้งาน บางอย่าง เช่น ป้องกันมิให้มีผู้ใช้งาน พร้อมกันมาเกิดกำหนด)
SHUTDOWN	อนุญาตให้สั่งปิดการทำงานของ server	Mysqldadmin shutdown
PROCESS	ใช้คำสั่งพิเศษกับ server เช่น SHOW	SHOW PROCESSLIST (ขอให้แสดง รายการ โพรเซสที่กำลังทำงานอยู่)
FILE	อ่าน - เขียน ไฟล์ที่ server ได้	LOAD DATA INFILE 'data.txt' INTO TABLE table1 (โหลดข้อมูลจากไฟล์ชื่อ data.txt ไปบันทึกลงในตารางข้อมูล table1)

เมื่อคุณ Install MySQL เสร็จ สิ่งแรกที่คุณควรทำ ก็คือการกำหนด password ใหม่ให้กับ root user โดยวิธีการ ดังนี้

```
shell> mysql -u root
mysql> use mysql;
mysql> UPDATE user SET Password=PASSWORD('new_password') WHERE user='root';
mysql> FLUSH PRIVILEGES;
```

### วิธีการกำหนด password

จากตัวอย่างในหัวข้อที่ผ่านมาจะเห็นได้ว่าเมื่อคุณ INSERT หรือ UPDATE password คุณต้องใช้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

function PASSWORD() เพื่อเข้ารหัส password เนื่องจากการกระทำเช่นนั้น ทำให้ password ถูกเก็บลง table ในแบบที่ถูกเข้ารหัส เพื่อความปลอดภัยของข้อมูล หากคุณลืมใช้ function PASSWORD() , password ก็จะถูกเก็บในรูปแบบของตัวหนังสือ (plaintext)

### 2.13.6 ฐานข้อมูล MySQL

MySQL มองไฟล์เคอร์เนลใน C:\mysql\data เป็นฐานข้อมูล

สำหรับไฟล์ทั้งหลายที่อยู่ในโฟลเดอร์ย่อยๆ นั้น ก็คือ ไฟล์ที่อยู่ในฐานข้อมูลนั้น โดยจำแนกชนิดของไฟล์ตามนามสกุล 3 ชนิดได้แก่

**MYD** หมายถึง ไฟล์ที่ใช้เก็บข้อมูล ไฟล์นี้ก็คือตารางข้อมูลนั่นเอง เช่น phonebook.MYD หมายถึง ตารางข้อมูลชื่อ phonebook

**MYI** หมายถึง ไฟล์ดัชนี หรือ index file เพื่อใช้เก็บค่าข้อมูลจากคอลัมน์ที่มีสถานะเป็นคีย์หรือดัชนี (index) และเก็บค่าพอยน์เตอร์ (pointer) ด้วย เวลาค้นหาข้อมูล โปรแกรม MySQL จะค้นหาจากไฟล์นี้แทนที่จะค้นหาจากตาราง (ไฟล์ .MYD) ตรงๆ เพราะไฟล์ดัชนีมีขนาดเล็กกว่า มีวิธีการจัดเก็บข้อมูลที่เอื้อต่อการเรียกค้นหามากกว่า จึงค้นได้เร็วกว่า

เมื่อได้ค่าที่ต้องการแล้ว MySQL จะอาศัยพอยน์เตอร์อ้างอิงไปยังตำแหน่งแถวรายการข้อมูลที่เกี่ยวข้องกับดัชนี ซึ่งอยู่ในไฟล์ข้อมูลจริงหรือ MYD อีกที

เวลาที่มีการแก้ไขข้อมูล หรือลบข้อมูล รวมทั้งเมื่อมีการเพิ่มรายการข้อมูลใหม่เข้าไป ทำให้ค่าดัชนีมีการเปลี่ยนแปลง ข้อมูลในไฟล์ดัชนีก็จะถูกจัดเรียงลำดับใหม่ตามข้อมูลจริงที่เปลี่ยนไปด้วย

**frm** หมายถึง ไฟล์ที่เก็บรายละเอียดโครงสร้างของตารางข้อมูล ว่าประกอบด้วยคอลัมน์อะไรบ้าง และเป็นคอลัมน์ชนิดไหน ดังนั้นสมมติว่าเราสร้างตารางข้อมูลขึ้นมาหนึ่งตาราง เช่น ตารางชื่อ phonebook.MYD phonebook.MYI และ phonebook.frm โดยแต่ละไฟล์จะทำหน้าที่ตามที่อธิบายไปแล้ว

#### Reserved Word:

MySQL มีคำที่สงวนไว้ ซึ่งคุณไม่สามารถนำคำเหล่านี้ไปตั้งเป็นชื่อตารางหรือ column ได้

action	add	aggregate	all
alter	after	and	as
asc	avg	avg_row_length	auto_increment
between	bigint	bit	binary
blob	bool	both	by
cascade	case	char	character
change	chcek	checksum	column
columns	comment	constraint	create
cross	current_date	current_time	date
datetime	day	day_hour	day_minute
day_second	dayofmonth	dayofweek	dayofyear

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dec	decimal	default	delayed
delay_key_write	delete	desc	describe
distinct	distinctrow	double	drop
end	else	escape	escaped
enclosed	enum	explain	exists
fields	file	first	float
float4	float8	flush	foreign
from	for	full	function
global	grant	grants	group
having	heap	high_priority	hour
hour_minute	hour_second	hosts	identified
ignore	in	index	infile
inner	insert	insert_id	int
integer	interval	int1	int2
int3	int4	int8	into
if	is	isam	join
key	keys	kill	last_insert_id
leading	left	length	like
lines	limit	load	local
lock	logs	long	longblob
longtext	low_priority	max	max_rows
match	mediumblob	mediumtext	mediumint
middleint	min_rows	minute	minute_second
modify	month	monthname	myisam
natural	numeric	no	not
null	on	optimize	option
optionally	or	order	outer
outfile	pack_keys	partial	password
precision	primary	procedure	process
processlist	privileges	read	real
references	reload	returns	revoke
rlike	row	rows	second
select	set	show	shutdown
smallint	soname	sql_big_tables	sql_big_selects

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sql_log_off	sql_log_update	sql_select_limit	sql_low_priority_updates
sql_small_result	sql_big_result	sql_warnings	straight_join
tables	temporary	terminated	test
then	time	timestamp	tinyblob
tinytext	tinyint	trailing	to
type	use	using	unique
unlock	unsigned	update	usage
values	varchar	variables	varying
varbinary	with	write	when
where	year	year_month	zerofill



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### การออกแบบและการสร้าง

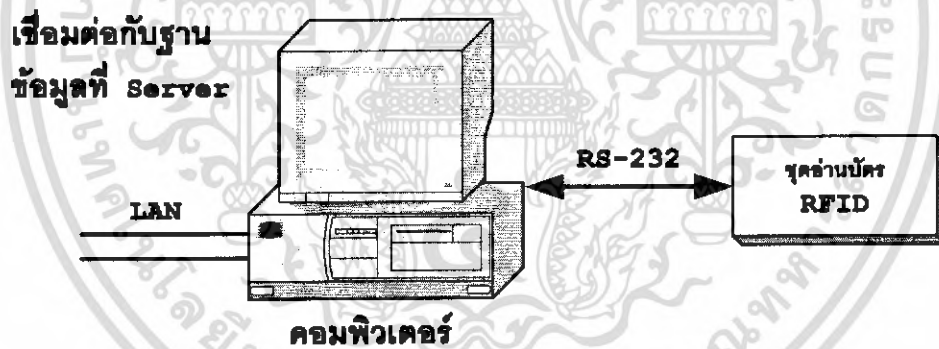
การออกแบบและการสร้างระบบควบคุมการใช้บริการรถไฟฟ้าโดยใช้บัตรRFID นั้นจะประกอบไปด้วย 8 ส่วนหลักด้วยกัน คือ

1. ชุดการแปลงพอร์ต RS-232 เป็น RS-485
2. ชุดอ่านบัตร RFID
3. ชุดควบคุมการเปิด-ปิดประตูทางเข้าออก
4. ระบบฐานข้อมูลของระบบการใช้บริการรถไฟฟ้า
5. โปรแกรมกำหนดหมายเลขประจำชุดควบคุมการเปิด-ปิดประตูทางเข้าออก
6. โปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า
7. โปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก

#### โครงสร้างของระบบการเชื่อมต่อ

- การเชื่อมต่อระหว่างชุดอ่านบัตร RFID กับคอมพิวเตอร์

มีผังการทำงานดังนี้



รูปที่ 3.1 โครงสร้างการเชื่อมต่อระหว่างชุดอ่านบัตร RFID กับคอมพิวเตอร์

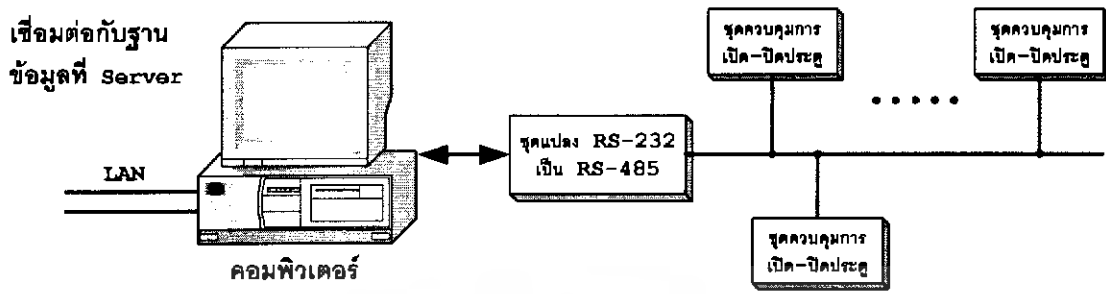
การเชื่อมต่อระหว่างชุดอ่านบัตร RFID กับคอมพิวเตอร์ เชื่อมต่อกันด้วยการใช้มาตรฐาน RS-232 โดยที่ชุดอ่านบัตรจะมีโมโครคอนโทรลเลอร์ในการประมวลผลการทำงาน โดยจะอ่านค่ารหัสประจำตัวบัตรที่ได้เพื่อส่งให้คอมพิวเตอร์ด้วยมาตรฐาน RS-232 เพื่อนำไปใช้กับโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า เช่น การเพิ่มหรือลบข้อมูลผู้ใช้บริการทั้งแบบรายเดือนและแบบเติมเงินในฐานข้อมูล หรือ การเติมเงินของผู้ใช้บริการแบบเติมเงิน เป็นต้น

การเชื่อมต่อกับฐานข้อมูลจะทำการเชื่อมต่อผ่านระบบเครือข่ายท้องถิ่น (Local Area Network : LAN) โดยฐานข้อมูลที่ใช้เขียนด้วยโปรแกรม MySQL ซึ่งเป็นโปรแกรมฐานข้อมูล Server โดยผ่าน ODBC Connector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเชื่อมต่อระหว่างชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

มีผังการทำงานดังนี้



รูปที่ 3.2 โครงสร้างการเชื่อมต่อระหว่างชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

การทำงาน คอมพิวเตอร์จะเชื่อมต่อกับชุดการแปลงพอร์ท RS-232 เป็น RS-485 จากนั้นจะเชื่อมต่อเข้ากับชุดควบคุมการเปิด-ปิดประตูทางเข้าออก ด้วยมาตรฐาน RS-485 เพื่อติดต่อสื่อสารกันได้ในลักษณะการเชื่อมต่อแบบบัส (Bus) ซึ่งสามารถติดต่อสื่อสารกันได้หลายๆ โหนด ส่วนการเชื่อมต่อกับฐานข้อมูลจะทำการเชื่อมต่อผ่านทาง ODBC Connector

ชุดควบคุมการเปิด-ปิดประตูทางเข้าออก จะใช้ในการอ่านรหัสประจำตัวบัตรRFID แล้วทำการส่งข้อมูลผ่านโปรแกรมบนคอมพิวเตอร์ เพื่อติดต่อกับฐานข้อมูลในการตรวจสอบสิทธิ และเก็บข้อมูลการเข้า-ออกที่สถานีต่างๆแล้วส่งคำสั่ง เพื่อควบคุมการเปิด-ปิดประตูให้กับชุดควบคุมการเปิด-ปิดประตูทางเข้าออกให้ทำงานตามคำสั่งที่กำหนด

3.1 การออกแบบและการสร้างชุดการแปลงพอร์ท RS-232 เป็น RS-485

3.1.1 บล็อกการทำงาน



รูปที่ 3.3 บล็อกการทำงานของชุดการแปลงพอร์ท RS-232 เป็น RS-485

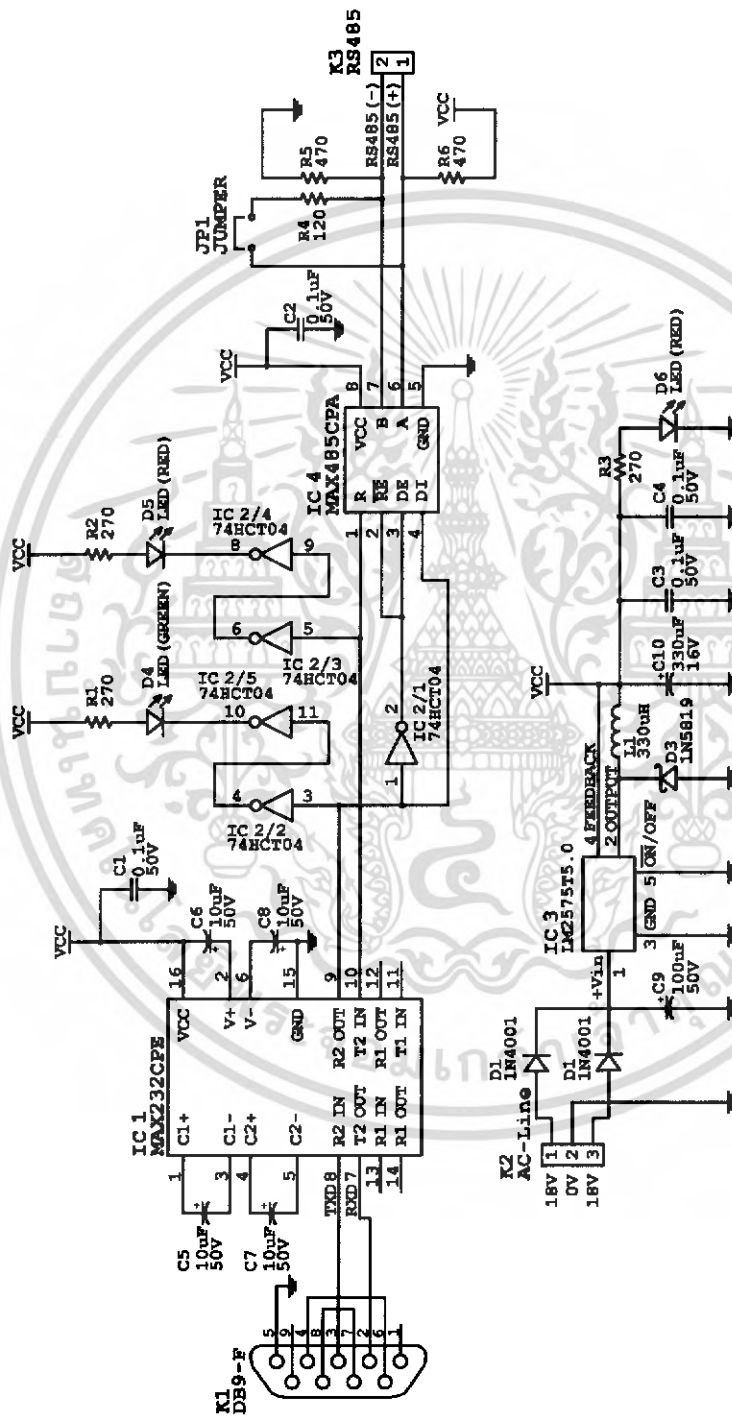
จากรูปบล็อกการทำงานของชุดการแปลงพอร์ท RS-232 เป็น RS-485 เป็นชุดที่ทำการแปลงระบบการรับ-ส่งข้อมูลอนุกรมตามมาตรฐาน RS-232 ให้มีการติดต่อสื่อสารในระยะไกลและสามารถเชื่อมต่อได้แบบ Point-To-Point เท่านั้น เพื่อแปลงเป็นระบบรับ-ส่ง ข้อมูลอนุกรมตาม มาตรฐาน RS-485 ให้มีการสื่อสารได้ในระยะไกลและสามารถเชื่อมต่อได้แบบ Multi-drop ซึ่งมีโครงสร้างการเชื่อมต่อแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัส โดยใช้สายส่งสัญญาณเพียง 2 เส้น สามารถรับ-ส่งข้อมูลได้ระยะทางประมาณ 1 กิโลเมตร และมีการรับ-ส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์ (Half-Duplex)

ชุดการแปลงพอร์ท RS-232 เป็น RS-485 เป็นชุดทำการเชื่อมต่อกับชุดควบคุมการเปิด-ปิดประตูทางเข้าออก กับคอมพิวเตอร์

3.1.2 การออกแบบวงจร



รูปที่ 3.4 วงจรชุดการแปลงพอร์ท RS-232 เป็น RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป ชุดการแปลงพอร์ท RS-232 เป็น RS-485 การรับ-ส่งข้อมูล จากพอร์ท RS-232 จากคอมพิวเตอร์จะถูกแปลงข้อมูลที่เป็นระดับสัญญาณมาตรฐาน RS-232 เป็นระดับลอจิกโดยผ่าน IC1 เบอร์ MAX 232CPE สัญญาณจากขา TxD ของพอร์ท RS-232 มีระดับ  $\pm 12V$  เข้าที่ขา R2 IN ออกเป็นสัญญาณระดับลอจิกที่ขา R2 OUT ของ IC1 ส่วนข้อมูลที่ได้รับเข้าทางขา RxD ของคอมพิวเตอร์จะได้รับสัญญาณระดับลอจิกจากขา T2 IN ออกเป็นสัญญาณระดับ  $\pm 12V$  ที่ขา T2 OUT ของ IC1

เมื่อแปลงข้อมูลเป็นระดับลอจิกแล้วจะทำการแปลงข้อมูลระดับลอจิกนี้ไปเป็นข้อมูลอนุกรมตามมาตรฐาน RS-485 ที่ IC4 เบอร์ MAX485CPA โดยการควบคุมการรับส่งข้อมูลผ่าน IC4 จะถูกควบคุมโดยใช้ IC2/1 เบอร์ 74HCT04 เป็น NOT Gate เป็นตัวกำหนดทิศทางการรับส่งข้อมูล

**ตารางที่ 3.1** การส่งข้อมูลของไอซี MAX485CPA

อินพุต			เอาต์พุต	
$\overline{RE}$	DE	DI	A	B
X	1	1	1	0
X	1	0	0	1
0	0	X	High-Z	High-Z
1	0	X	High-Z	High-Z

**ตารางที่ 3.2** การรับข้อมูลของไอซี MAX485CPA

อินพุต			เอาต์พุต
$\overline{RE}$	DE	A-B	RO
0	0	$\geq +0.2V$	1
0	0	$\leq -0.2V$	0
0	0	Input open	1
1	0	X	High-Z

X=Don't Care

High-Z=High Impedance

การส่งข้อมูลจากไอซี MAX232CPE ไปยังไอซี MAX485CPA จะใช้ IC2/1 เป็นตัวควบคุมการส่งข้อมูลโดยเมื่อต้องการส่งข้อมูลลอจิก "0" เอาท์พุตของ IC2/1 จะเป็นลอจิก "1"

จากตารางที่ 3.1 เมื่อ  $\overline{RE}$  และ DE เป็นลอจิก "1" ทั้งสองขา จะเป็นการเลือกให้ ไอซี MAX485CPA ทำหน้าที่ส่งข้อมูล โดยข้อมูลลอจิก "0" ที่รอที่ขา DI จะส่งออก ไอซีที่ขา A เป็น "0" และขา B เป็น "1" เมื่อส่งข้อมูลลอจิก "1" เอาท์พุตของ IC2/1 จะเป็น "0"

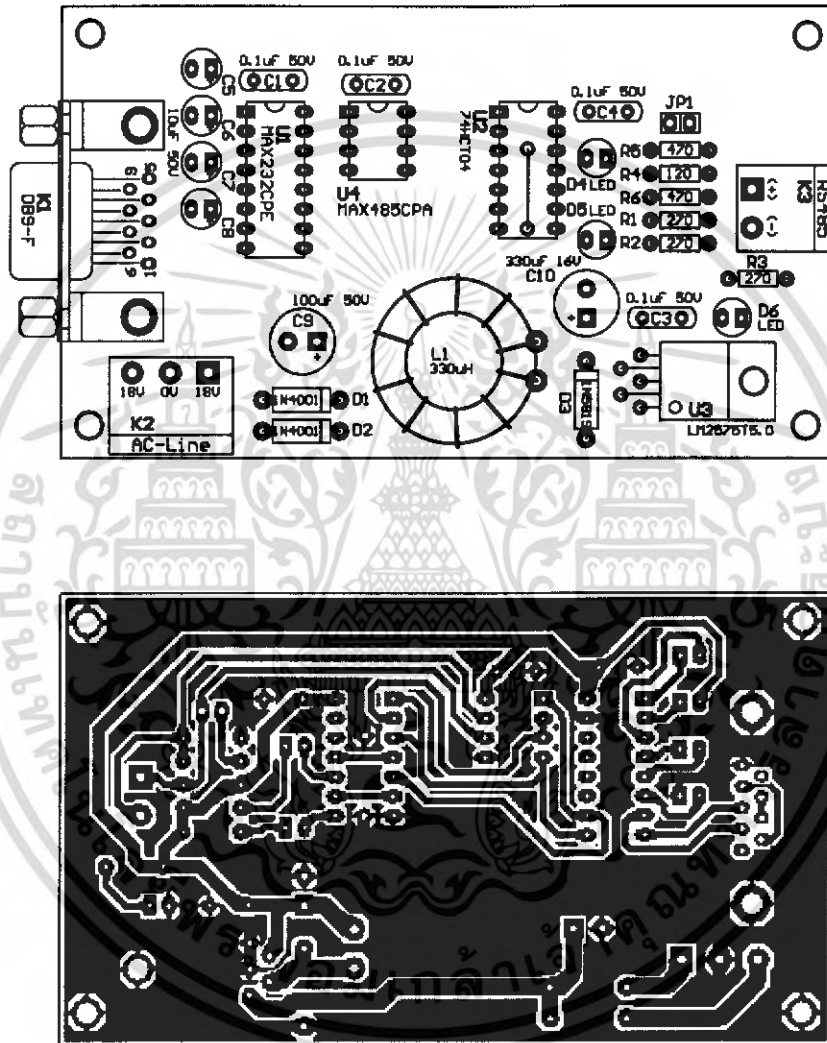
จากตารางที่ 3.1 เมื่อ  $\overline{RE}$  และ DE เป็น "0" ทั้งคู่จะทำให้ทั้งขา A และ B มีอิมพีแดนซ์สูง แต่ที่ขา A(+) ของ IC4 มีการพูลอัพไว้ที่ +5V และขา B(-) ของ IC4 จะถูกพูลดาวน์ไว้ที่ 0V จึงทำให้เอาท์พุตที่ออกที่ขา A เป็น "1" และที่ขา B เป็น "0" ทำให้การรับส่งข้อมูลเป็นแบบอัตโนมัติ

ส่วนการรับข้อมูลที่ไอซี MAX232CPE จากไอซี MAX485CPA นั้นเนื่องจากการสื่อสารแบบฮาล์ฟดูเพล็กซ์ จึงไม่มีการรับส่งข้อมูลเวลาเดียวกันเมื่อต้องการรับข้อมูลที่ขา TxD อยู่ในสภาวะ Idle เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือลอจิก “1” ออกที่ขา R2O สัญญาณที่ออก IC2/1 เป็น “0” จากตารางที่ 3.2 เห็นว่า  $\overline{RE}$  และ DE เป็น “0” ทั้งคู่ จะเป็นการรับส่งข้อมูลของไอซี MAX485CPA ไม่ว่าข้อมูลที่รับเข้ามาจะเป็นอย่างไร ที่ขา A-B จะได้ เอาท์พุทออกที่ RO ของไอซี MAX485CPA

ภาคจ่ายไฟของชุดการแปลงพอร์ท RS-232 เป็น RS-485 ใช้ไอซีเบอร์ LM2575 T5.0 ในการรักษาระดับแรงดันไฟตรงที่เรกติไฟรี่ได้ให้คงที่อยู่ที่ +5V ซึ่งเป็นไอซี DC-to-DC Converter

### 3.1.3 การออกแบบลายวงจร

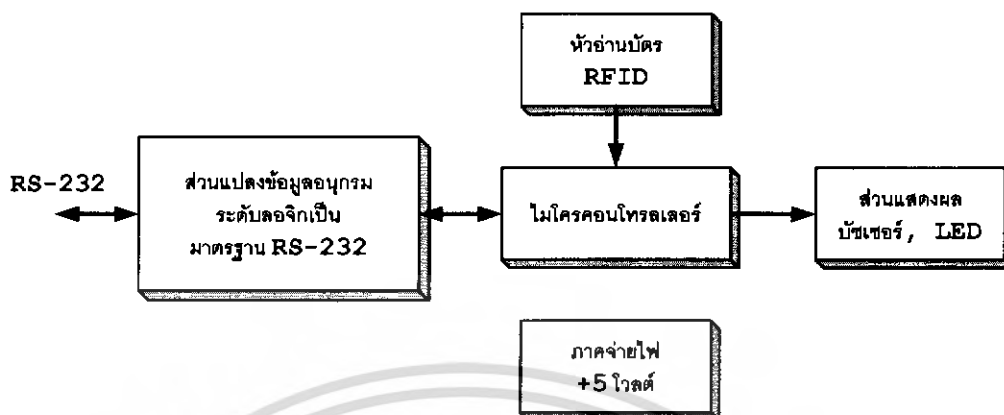


รูปที่ 3.5 ลายวงจรและการวางอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 การออกแบบและการสร้างชุดอ่านบัตร RFID

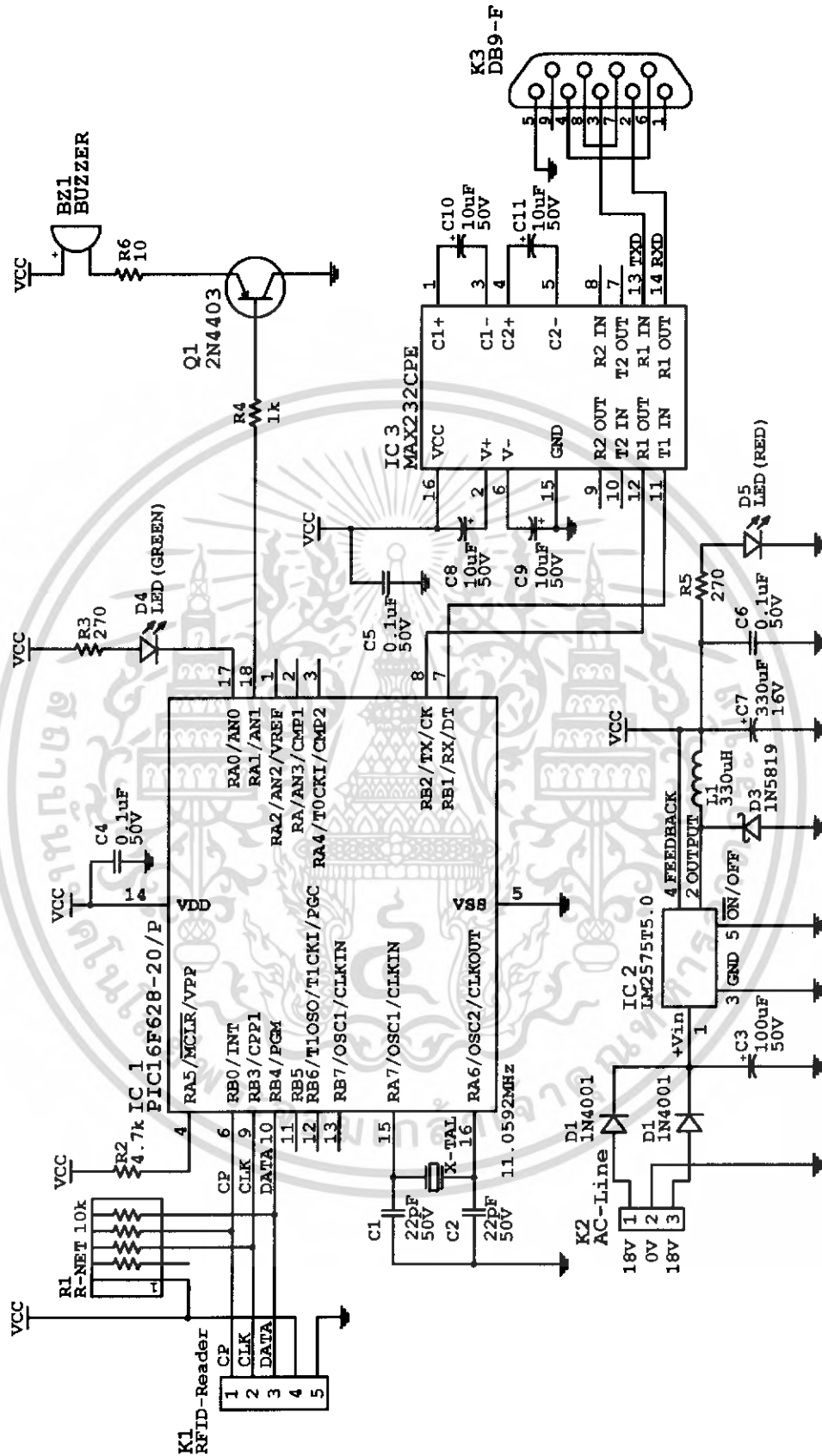
#### 3.2.1 บล็อกการทำงานของชุดอ่านบัตร RFID



**รูปที่ 3.6** บล็อกการทำงานของชุดอ่านบัตร RFID

ชุดอ่านบัตร RFID เป็นชุดที่ใช้ในการอ่านรหัสประจำตัวบัตรแล้วส่งข้อมูลไปยังคอมพิวเตอร์ ด้วยมาตรฐานการเชื่อมต่อแบบ RS-232 มีส่วนแสดงผลคือ บัชเชอร์ และ LED ให้ทราบผลการทำงานของชุดอ่านบัตร RFID หัวอ่านบัตร RFID ใช้ชุดหัวอ่านของ PROMAG รุ่น GP-8 โดยมีรูปแบบการรับส่งข้อมูลแบบ Magstripe เหมือนที่ใช้ในเครื่องอ่านบัตรแถบแม่เหล็ก

3.2.2 การออกแบบวงจร



รูปที่ 3.7 วงจรของชุดอ่านบัตร RFID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจร ไมโครคอนโทรลเลอร์ที่จะใช้ของ Microchip เบอร์ PIC 16F628-20/P ในการประมวลผลต่างๆของชุดอ่านบัตร RFID

หัวอ่านบัตร RFID มีอยู่ 3 ขาด้วยกันคือ ขา DATA ต่ออยู่ที่ PORT RB.4 ขา CLK ต่ออยู่ที่ PORT RB.3 ขา CP ต่ออยู่ที่ PORT RB.0 ของ IC1 คือ PIC 16F628-20/P ส่วนแสดงผลบัชเซอร์จะต่ออยู่ที่ขา RA1 ส่วน LED ต่ออยู่ที่ RA0

การรับส่งข้อมูลใช้ไอซี MAX 232CPE ในการแปลงระดับของสัญญาณข้อมูลระดับลอจิกไปเป็นระดับสัญญาณตามมาตรฐาน RS-232 ไปยังขั้วต่อ K3 คือ DB9 ตัวเมีย

ภาคจ่ายไฟของชุดอ่านบัตร RFID ใช้ไอซีเบอร์ LM2575 T5.0 ในการรักษาระดับแรงดันไฟตรง 24 V ที่เรกติไฟร์ไว้ให้คงที่อยู่ที่ +5V ซึ่งเป็นไอซี DC-to-DC Converter

### 3.2.3 ข้อกำหนดในการติดต่อสื่อสาร (Communication Protocol) ของชุดอ่านบัตร RFID

การเชื่อมต่อของชุดอ่านบัตร RFID จะเชื่อมต่อกับคอมพิวเตอร์ด้วยมาตรฐาน RS-232 โดยชุดอ่านบัตร RFID นี้ในการอ่านรหัสบัตรเพื่อใช้จัดการระบบบริหารผู้ใช้บริการที่ห้องจำหน่ายตั๋วเดินทาง โดยชุดอ่านบัตร RFID จะทำการเชื่อมต่อกับคอมพิวเตอร์เพียงตัวเดียว

รายละเอียดของ โปรโตคอลของชุดอ่านบัตร RFID มีรูปแบบของข้อมูลในการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับชุดอ่านบัตร RFID เป็นรหัส ASCII ทั้งหมดประกอบด้วย

1 byte	14 byte	1 byte	1 byte	1 byte
ไบต์เริ่มต้น	รหัสบัตร 14 หลัก	LRC (HIGH)	LRC (LOW)	ไบต์จบ

ไบต์เริ่มต้น คือ ไบต์แรกที่ทำให้การบอกจุดเริ่มต้นในการส่งรหัสบัตรจากชุดอ่านบัตร RFIDมายังคอมพิวเตอร์ โดยใช้เครื่องหมาย@ (40H) เป็นตัวเริ่มต้นการส่งข้อมูล

รหัสบัตร 14 หลัก เป็นรหัสประจำตัวของบัตรซึ่งเป็นเลขฐานสิบ 14 หลัก ที่อ่านได้จากบัตรโดยชุดอ่านบัตร RFID

LRC HIGH และ LRC LOW (Longitude Redundancy Check) เป็นไบต์ที่ใช้ในการตรวจสอบความผิดพลาดของข้อมูลที่ทำการส่งไป คอมพิวเตอร์ตรวจสอบได้ที่สองไบต์นี้ โดยเป็น LRC เลขฐานสิบหก ที่มาจากการบวกสะสมของข้อมูลและทำการแปลงเป็นรหัส ASCII อีกที

ไบต์จบ คือ ไบต์ที่บอกจุดสิ้นสุดของข้อมูลรหัสบัตรที่ส่งไปโดยจะใช้เครื่องหมาย # (23H) เป็นตัวปิดท้าย

ส่วนของคอมพิวเตอร์เมื่อได้รับรหัสบัตรแล้วจะทำการส่งรหัส Acknowledgement เป็นรหัส A(41H) เพื่อบอกการรับข้อมูลถูกต้อง ชุดอ่านบัตร RFID จะทำการแสดงผลที่บัชเซอร์จะมีเสียงดังขึ้นและ LED จะติดประมาณ 3 วินาที และดับไปเพียงครั้งเดียว ส่วน Negative Acknowledgement จะให้รหัสใดส่งไปก็ได้ที่ไม่ใช่ A(41H) เพื่อบอกว่าชุดอ่านบัตร RFID มีความผิดพลาดเกิดขึ้น โดยชุดอ่านบัตร RFID จะทำการแสดงผลด้วยเสียงบัชเซอร์สั้นๆ 2 ครั้ง แต่หากมีความขัดข้องในการติดต่อสื่อสารที่ชุดอ่านบัตร RFID จะทำการแสดงผลที่บัชเซอร์จะมีเสียงดังขึ้นสั้นๆ 3 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

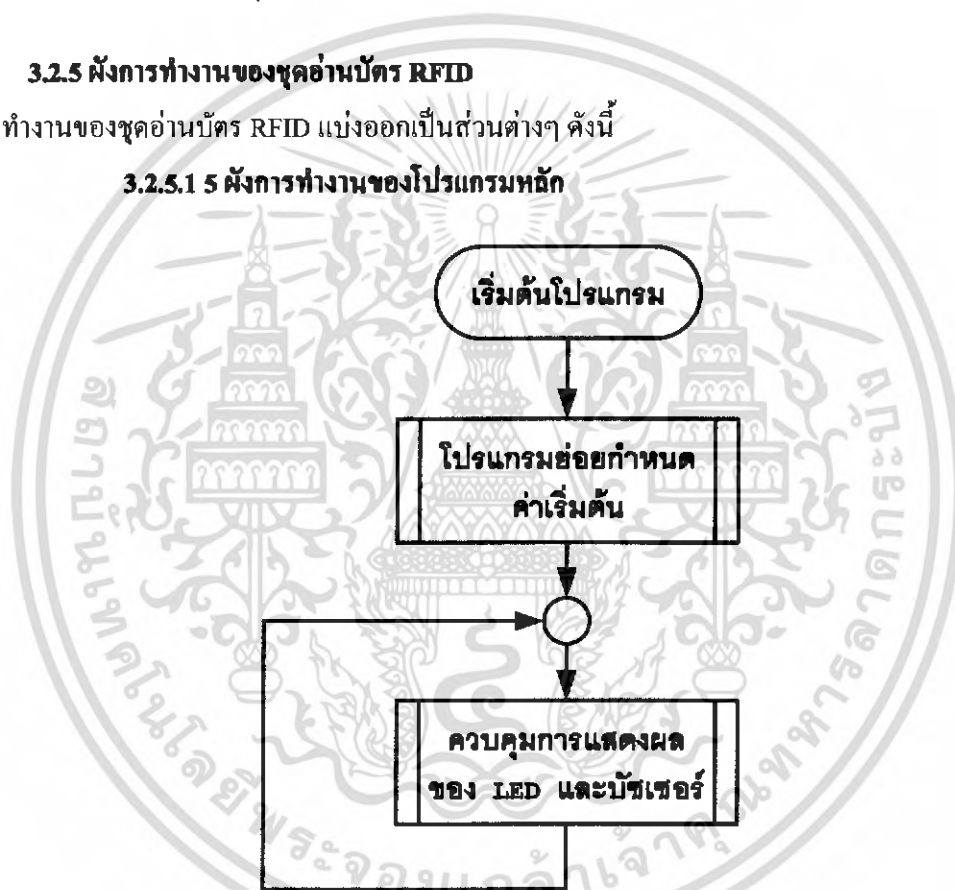
### 3.2.4 การทำงานของโปรแกรม

การทำงานของชุดอ่านบัตร RFID จะอ่านรหัสบัตรที่ได้จากหัวอ่านด้วยการบริการอินเทอร์เน็ตจากภายนอก เมื่อตรวจสอบความถูกต้องของข้อมูลเรียบร้อยแล้วจะทำการส่งข้อมูลรหัสบัตรให้คอมพิวเตอร์ ตามข้อกำหนด เมื่อคอมพิวเตอร์ได้รับรหัสบัตรที่ถูกต้องแล้วจะส่งรหัส A(41H) คือ Acknowledgement ให้ชุดอ่านบัตร RFID ได้รับความตอบสนองโดยการเปิดเสียงบี๊เซอร์และ LED ติดประมาณ 3 วินาที แต่ถ้าข้อมูลไม่ถูกต้องคอมพิวเตอร์จะทำการส่งรหัสอะไรก็ได้ที่ไม่ใช่ Acknowledgement เพื่อแสดง Negative Acknowledgement ให้กับชุดอ่านบัตร RFID แสดงผลด้วยเสียงบี๊เซอร์สั้นๆ 2 ครั้ง แต่ถ้าคอมพิวเตอร์ไม่ส่งข้อมูลเนื่องจากความขัดข้องนานเกิน 3 วินาที ชุดอ่านบัตร RFID จะทำการแสดงผลด้วยเสียงบี๊เซอร์สั้นๆ 3 ครั้ง

### 3.2.5 ผังการทำงานของชุดอ่านบัตร RFID

ผังการทำงานของชุดอ่านบัตร RFID แบ่งออกเป็นส่วนต่างๆ ดังนี้

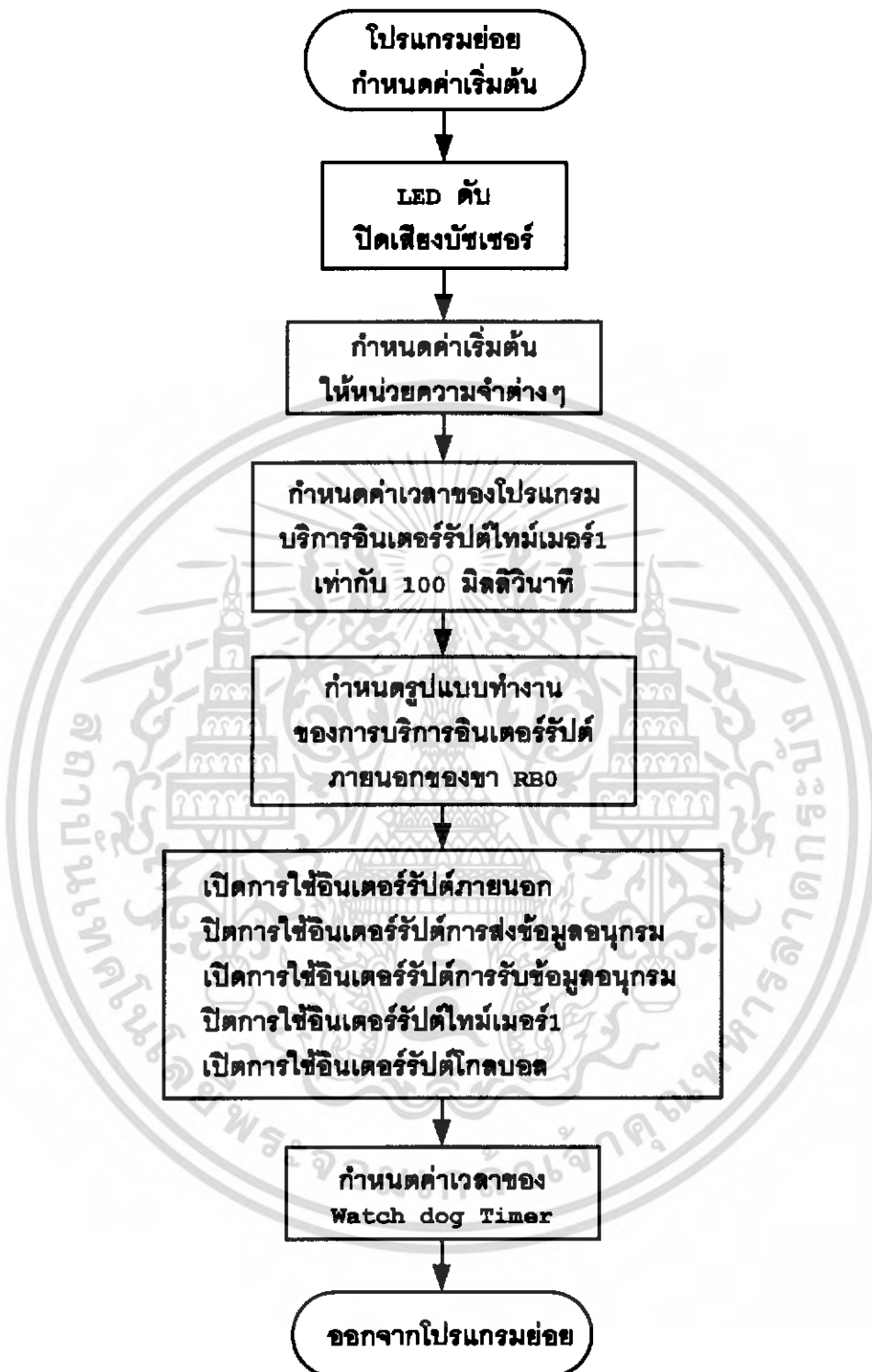
#### 3.2.5.1 5 ผังการทำงานของโปรแกรมหลัก



รูปที่ 3.8 ผังการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.5.2 ผังการทำงานของโปรแกรมย่อยกำหนดค่าเริ่มต้น

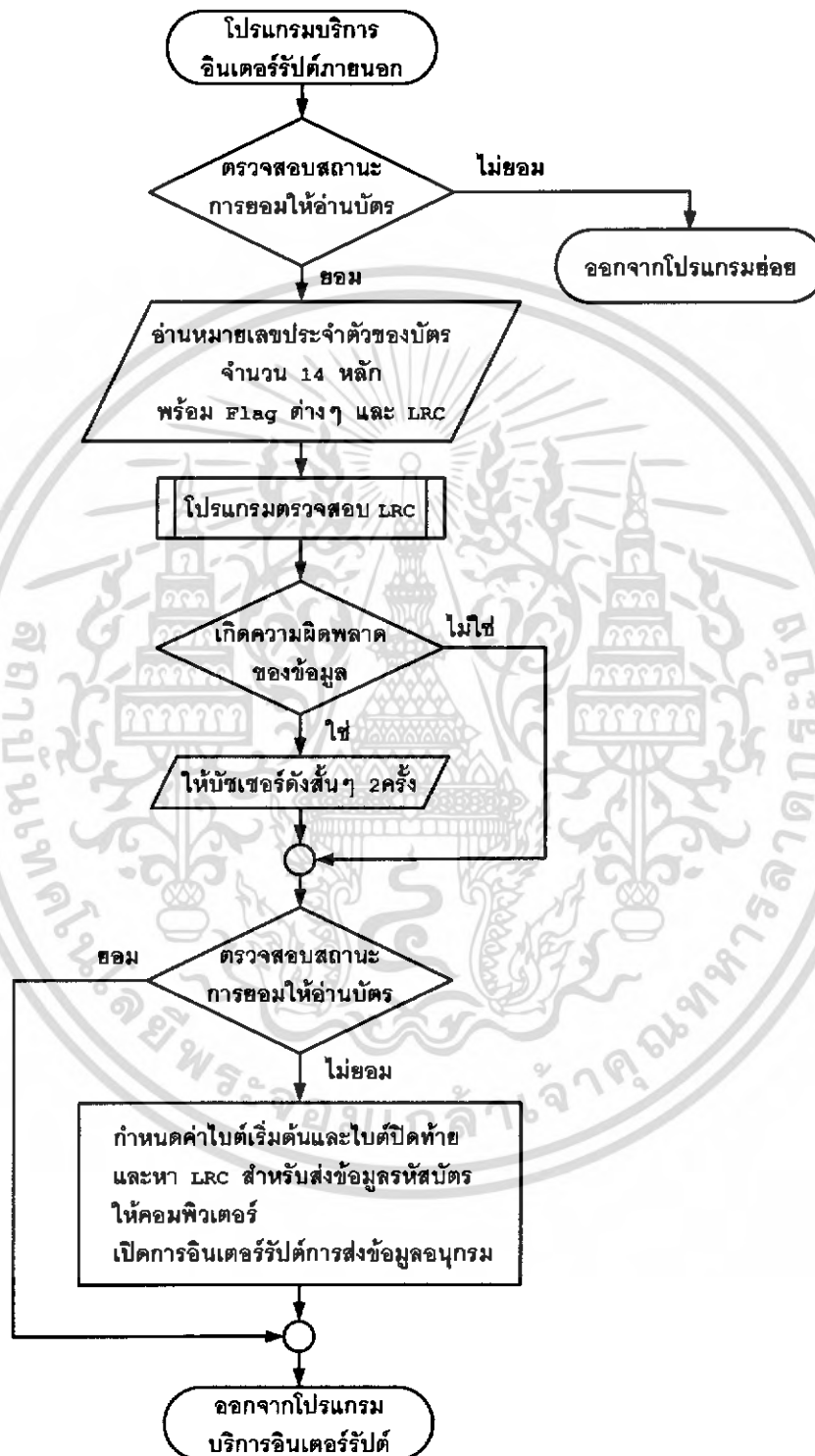


รูปที่ 3.9 ผังการทำงานของโปรแกรมย่อยกำหนดค่าเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.5.3 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายภายนอก

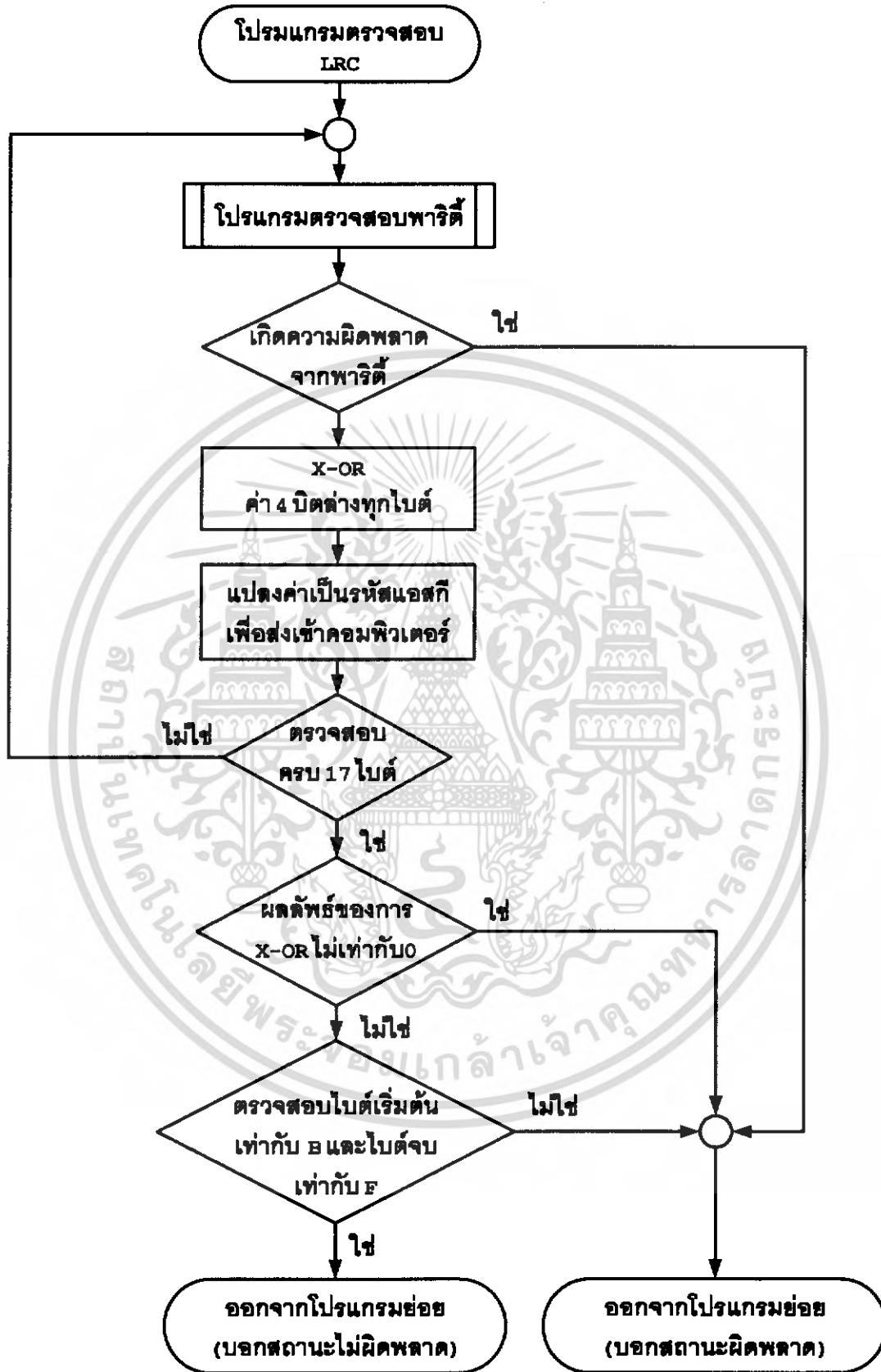
เป็นโปรแกรมใช้ในการอ่านข้อมูลรหัสบัตรจากหัวอ่านบัตร RFID เมื่อมีข้อมูลส่งจากหัวอ่านก็จะได้รับบริการอินเทอร์เน็ตไร้สายภายนอกนี้ แล้วทำการอ่านข้อมูลรหัสบัตรจากหัวอ่าน RFID



รูปที่ 3.10 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2.5.4 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบ LRC

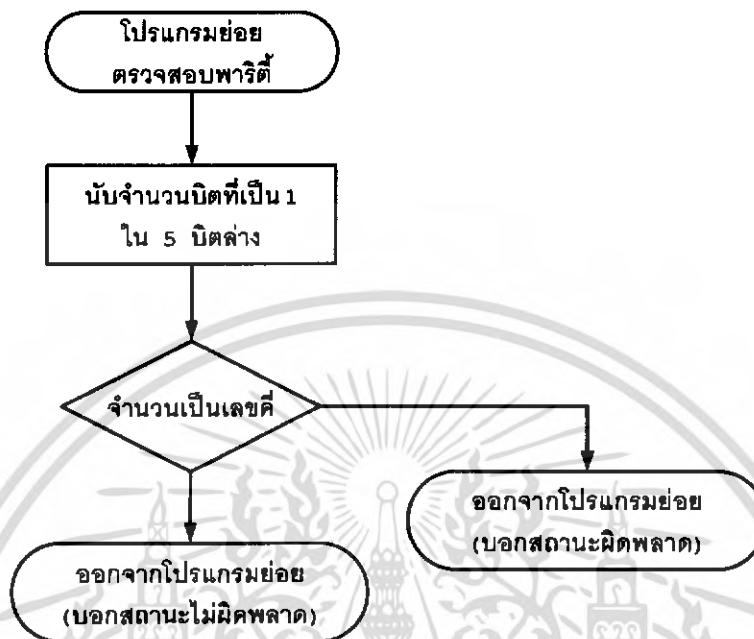


รูปที่ 3.11 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบ LRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.5.5 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบพาริตี

ในการส่งข้อมูลรหัสบิตจากหัวอ่านบิตจะส่งพาริตีเพื่อใช้ตรวจสอบความผิดพลาด โดยใช้ พาริตีคี่ มีฟังก์ชันการทำงานดังนี้



รูปที่ 3.12 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบพาริตี

### 3.2.5.6 ฟังก์ชันการทำงานของโปรแกรมย่อยหา LRC

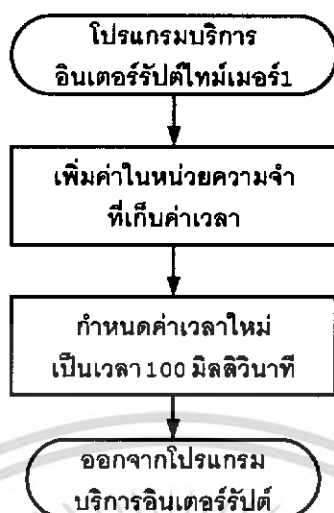
เป็นโปรแกรมใช้ในการหา LRC เพื่อใช้ในการตรวจสอบความผิดพลาดของข้อมูลที่ส่งให้คอมพิวเตอร์



รูปที่ 3.13 ฟังก์ชันการทำงานของโปรแกรมย่อยหา LRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

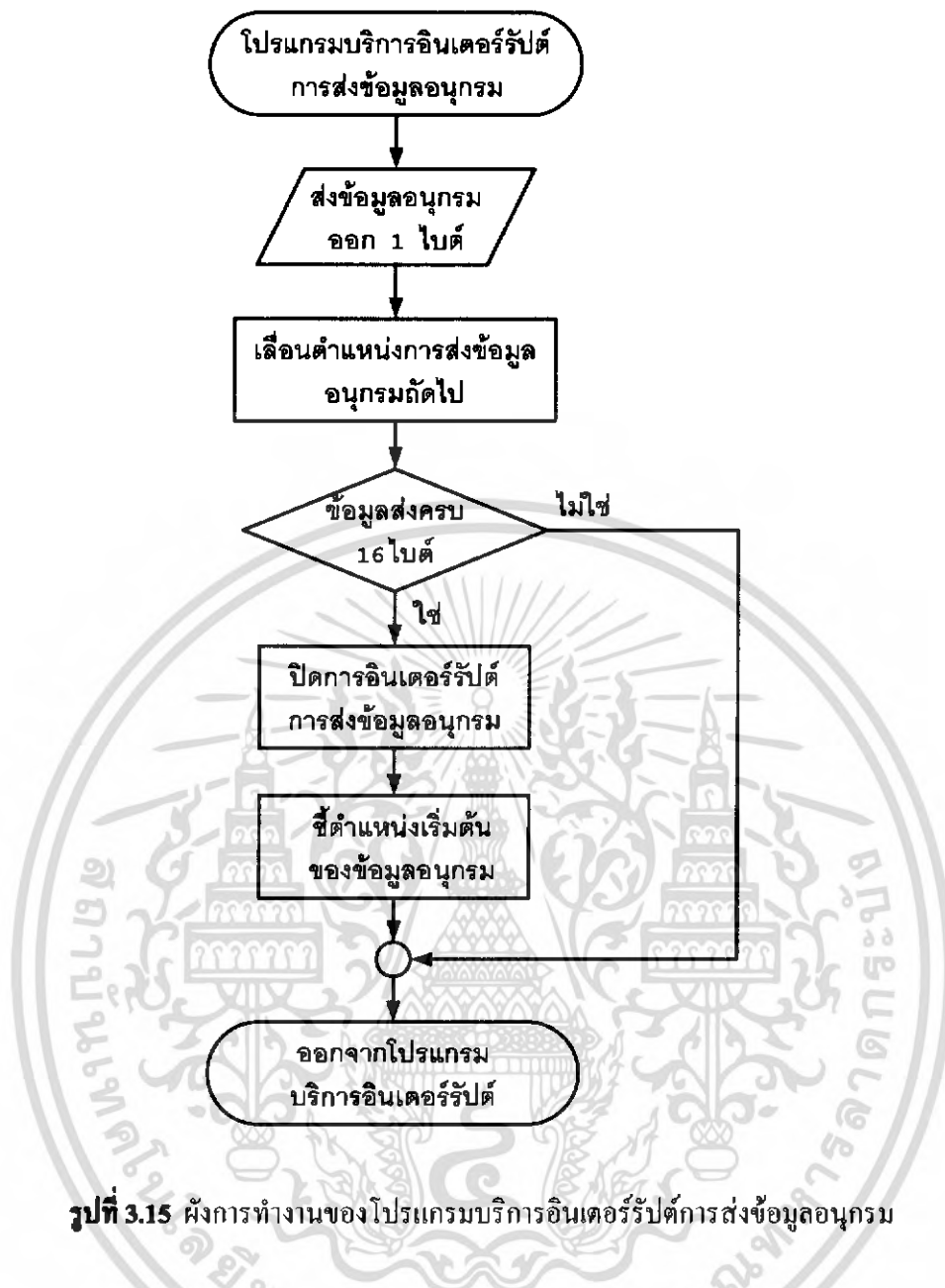
### 3.2.5.7 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้ทีเอ็มเมอร์



### รูปที่ 3.14 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้ทีเอ็มเมอร์

### 3.2.5.8 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตการส่งข้อมูลอนุกรม

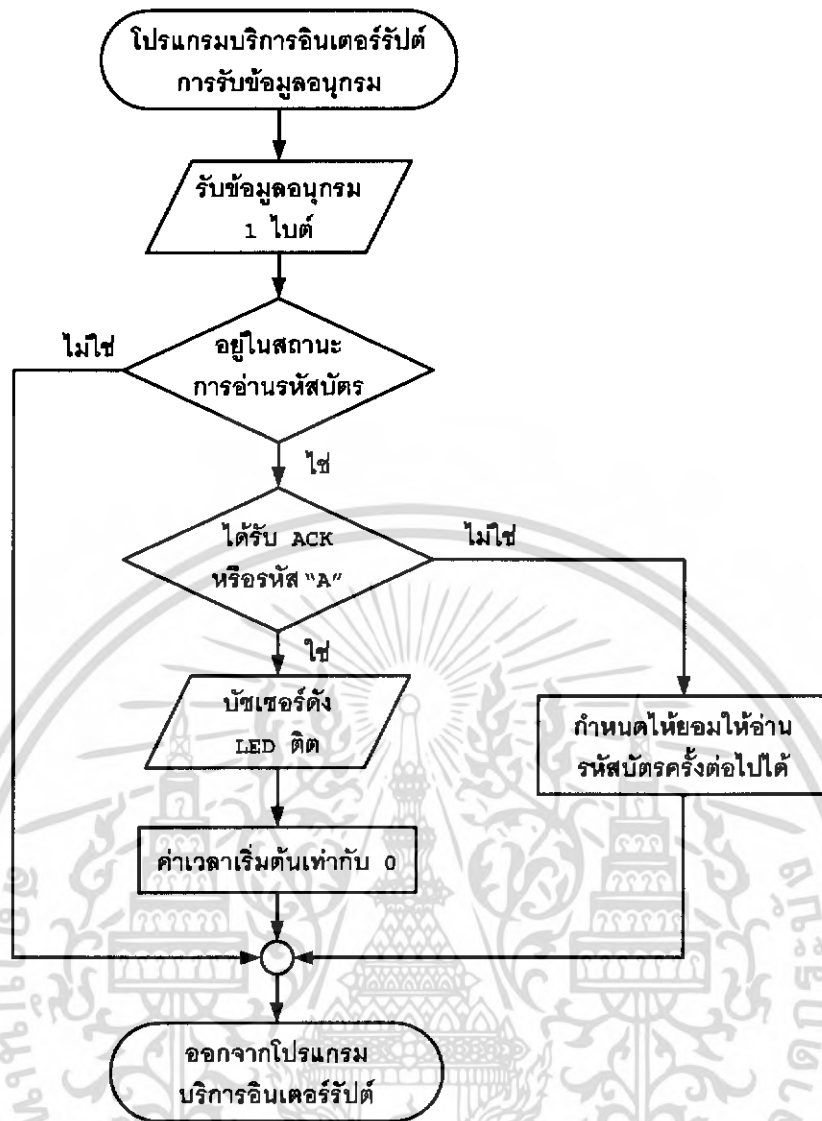
โปรแกรมบริการอินเทอร์เน็ตการส่งข้อมูลอนุกรม ทำงานเมื่อมีการเปิดการใช้บริการอินเทอร์เน็ตเพื่อการส่งข้อมูลอนุกรม โดยจะมีการส่งข้อมูลอนุกรมจำนวน 18 ไบต์ คือ ไบต์เริ่มต้น 1 ไบต์ ตามด้วยรหัสบิตเป็นรหัสแอสกีจำนวน 14 ไบต์, LRC ที่เป็นรหัสแอสกีอีก 2 ไบต์ และจบด้วยไบต์จบ 1 ไบต์ โดยมีตัวชี้ตำแหน่งข้อมูลที่ส่ง เมื่อส่งเสร็จจะทำการปิดการใช้บริการ



รูปที่ 3.15 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตการส่งข้อมูลอนุกรม

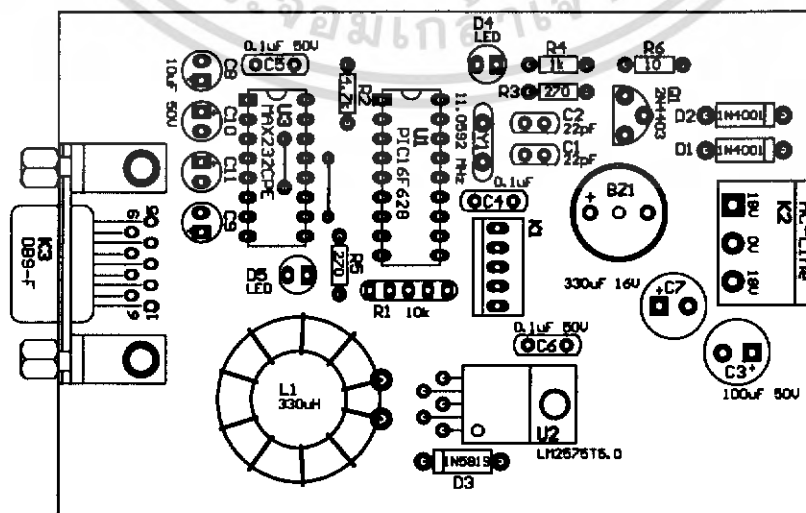
### 3.2.5.9 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตการรับข้อมูลอนุกรม

การทำงานของโปรแกรมบริการอินเทอร์เน็ตการส่งข้อมูลอนุกรม จะทำงานเมื่อมีข้อมูลเข้ามาที่รีจิสเตอร์ที่เก็บข้อมูลที่ได้รับมาจากคอมพิวเตอร์ เพื่อใช้ในการรับ Acknowledgement และ Negative Acknowledgement จากคอมพิวเตอร์ มาใช้ในการควบคุมการแสดงผลออกที่ บัชเชอร์และ LED

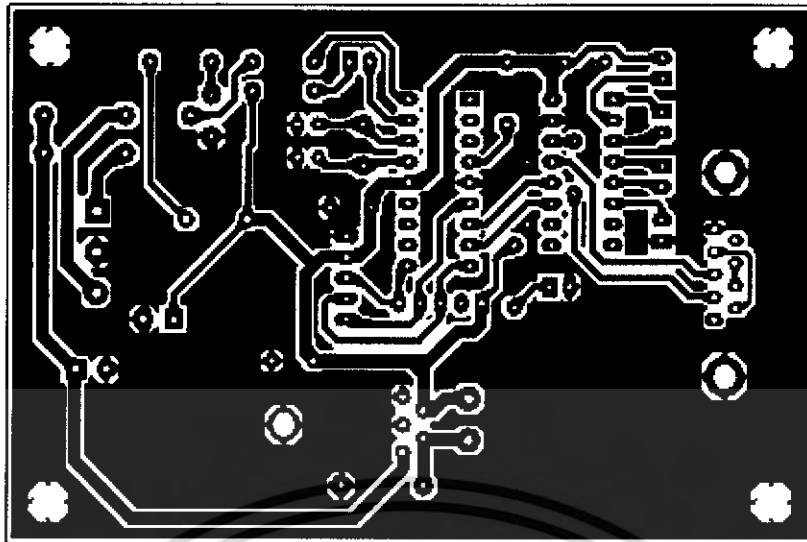


รูปที่ 3.16 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตการส่งข้อมูลอนุกรม

การออกแบบตาข่ายวงจร



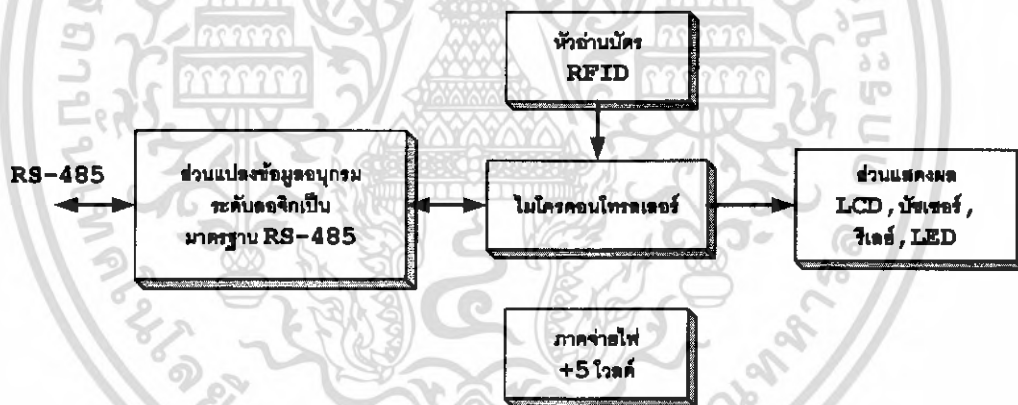
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 ลายวงจรและการวางอุปกรณ์

### 3.3 การออกแบบและการสร้างชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

#### 3.3.1 บล็อกการทำงานชุดควบคุมการเปิด-ปิดประตูทางเข้าออก



รูปที่ 3.18 บล็อกการทำงานชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

ชุดควบคุมการเปิด-ปิดประตูทางเข้าออก ประกอบไปด้วยไมโครคอนโทรลเลอร์ที่ใช้ในการประมวลผลการทำงานต่างๆของโปรแกรมตามคำสั่งของคอมพิวเตอร์ที่ส่งผ่านการเชื่อมต่อตามมาตรฐาน RS-485

ส่วนหัวอ่านบัตรจะทำงานเหมือนกับหัวข้อ 3.2.1 เรื่องชุดอ่านบัตร RFID

การแสดงผลจะแสดงผลด้วย LCD เพื่อแสดงผลการทำงานต่างๆ ที่ได้รับคำสั่งจากคอมพิวเตอร์ และแสดงค่าจำนวนเงินของผู้ใช้บริการ ส่วน LED จะแสดงผลในขณะที่มีการเปิด-ปิดประตูบัสเซอร์จะส่งเสียงดังตามการทำงานของไมโครคอนโทรลเลอร์ ส่วนรีเลย์ใช้ในการควบคุมการเปิด-ปิดประตูโดยสามารถนำไปต่อใช้งานกับประตูเปิด-ปิดอัตโนมัติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ชุดควบคุมการเปิด-ปิดประตูทางเข้าออก ใช้ไมโครคอนโทรลเลอร์ของ Microchip เบอร์ PIC 16F873A-I/SP ในการประมวลผลการทำงานทั้งหมด โดยเชื่อมต่อกับคอมพิวเตอร์ผ่านชุดแปลง RS-232 เป็น RS-485 ด้วยมาตรฐานการเชื่อมต่อ RS-485 ด้วยไอซี MAX 485CPA โดยขาส่งข้อมูลของไอซี PIC 16F873A-I/SP ต่ออยู่ที่ขา RC6 และขารับข้อมูลต่ออยู่ที่ขา RC7 โดยเลือกโหมดการส่งข้อมูลอนุกรมด้วยขา RC5 เป็นขาเลือกการทำงานในการส่งข้อมูลผ่านไอซี MAX 485CPA มีจัมเปอร์ JP1 ในการทำให้สายส่งมีอิมพีแดนซ์ที่สมดุลเมื่อสายส่งมีระยะทางไกล โดยจัมพ์ที่คั่นทางกับปลายทาง

การเชื่อมต่อกับหัวอ่านบัตร RFID มี 3 ขา คือ

ขา DATA จะต่ออยู่ที่ขา RC1

ขา CLK จะต่ออยู่ที่ขา RC0

ขา CP จะต่ออยู่ที่ขา RB0, ของไอซี PIC 16F873A-I/SP

ซึ่งจะใช้การอินเทอร์รัปต์ภายนอกในการส่งข้อมูลรหัสบัตรส่งให้ไมโครคอนโทรลเลอร์

ส่วนของการแสดงผลจะใช้ LCD แสดงผลคำสั่งที่ส่งมาจากคอมพิวเตอร์ โดยเชื่อมต่อโปรแกรมการใช้ LCD แบบ 4 บิต ที่ขา D4-D7 ของ LCD ซึ่งต่อกับขา RB4-RB7 ขา EN ของ LCD ต่อกับขา RB1, ขา RS ของ LCD ต่อกับขา RB2 และขา R/W ของ LCD ต่อกับขา RB3 ของไอซี PIC 16F873A-I/SP

บัสเซอร์ต่ออยู่ที่ขา RA2 ของไอซี PIC 16F873A-I/SP ใช้ส่งเสียงเตือนผู้ให้บริการได้รับรู้ผลการทำงานของระบบการเปิด-ปิดประตูทางเข้าออก มี LED ในการแสดงผลต่ออยู่ที่ขา RA0 รีเลย์ต่ออยู่ที่ขา RA1 ของไอซี PIC 16F873A-I/SP ซึ่งสามารถต่อสัญญาณควบคุมการเปิด-ปิดประตูโดยผ่านเทอร์มินอล K4 ได้

ภาคจ่ายไฟให้กับวงจร ใช้ไอซีเบอร์ LM2575 T5.0 ในการรักษาระดับแรงดันไฟตรง 24 V ที่เรกติไฟร์ได้ให้คงที่อยู่ที่ +5V ซึ่งเป็นไอซี DC-to-DC Converter

### 3.3.3 ข้อกำหนดในการติดต่อสื่อสาร (Communication Protocol) ของชุดควบคุมการเปิด-ปิดประตู

การเชื่อมต่อระหว่างคอมพิวเตอร์กับชุดควบคุมการเปิด-ปิดประตู มีมาตรฐานการเชื่อมต่อ คือมาตรฐาน RS-485 ซึ่งสามารถเชื่อมต่อกันได้หลายๆเครื่องตามข้อมูลของไอซีที่ใช้ จะสามารถเชื่อมต่อได้สูงถึง 32 เครื่องพร้อมกันรวมกับคอมพิวเตอร์ด้วยอีก 1 เครื่อง แต่เนื่องจากไม่มีความจำเป็นต้องการหมายเลขประจำตัวเครื่องมากถึง 32 เครื่อง จึงกำหนดให้มีการใช้หมายเลขประจำเครื่องอยู่ระหว่าง 0-9 เท่านั้น โดยหมายเลขประจำตัวเครื่องหมายเลข 0 นั้นจะเป็นหมายเลขประจำตัวเครื่องของคอมพิวเตอร์ ซึ่งเป็นตัวแม่ (Master) ใช้ในการ Broadcast ข้อมูลคำสั่งต่างๆ ให้กับตัวลูก (Slave)

กำหนดให้มีจำนวนของเครื่องที่ประตูทางเข้าและประตูทางออกมีสูงสุดได้ 4 เครื่องด้วยกัน โดยกำหนดหมายเลขประจำตัวเครื่องได้ดังนี้

หมายเลข 1-4 ใช้เป็นหมายเลขประจำตัวเครื่องของเครื่องชุดควบคุมการเปิด-ปิดประตูทางเข้าของสถานี

หมายเลข 6-9 ใช้เป็นหมายเลขประจำตัวเครื่องของเครื่องชุดควบคุมการเปิด-ปิดประตูทางออกของสถานี

ส่วนหมายเลข 5 จะไม่นำไปใช้งาน

รายละเอียดของ Protocol

ข้อมูลที่ใช้ในการติดต่อกับเครื่องจะเป็นรหัส ASCII ทั้งหมด และการส่งแต่ละคำสั่งประกอบด้วย

1 ไบต์	1 ไบต์	n ไบต์	1 ไบต์
ไบต์เริ่มต้น	หมายเลขประจำตัวเครื่อง	คำสั่ง	ไบต์จบ

ไบต์เริ่มต้น คือ ไบต์แรกที่ยกกว่าคำสั่งเริ่มชุดคำสั่ง โดยจะใช้เครื่องหมาย @ (40H) เป็นไบต์ที่บอกจุดเริ่มต้นคำสั่ง

หมายเลขประจำเครื่อง คือ หมายเลขที่ใช้อ้างอิงถึงชุดควบคุมการเปิด-ปิดประตูทางเข้า - ออก

คำสั่ง คือ คำสั่งที่ใช้สั่งชุดควบคุมการเปิด-ปิดประตูให้ส่งข้อมูลกลับมายังคอมพิวเตอร์

ไบต์จบ คือ ไบต์ที่บอกว่สิ้นสุดชุดคำสั่ง โดยจะใช้ # (23H) ในตาราง ASCII เป็นตัวปิดท้าย

รายละเอียดของคำสั่งทั้ง 7 คำสั่ง พร้อมตัวอย่าง

1. คำสั่งที่ใช้ในการอ่านหมายเลขประจำตัวเครื่อง (Address) ใช้ในกรณีที่ต้องการเปลี่ยนหมายเลขประจำตัวชุดควบคุมการเปิด-ปิดประตู โดยต้องต่อเพียงชุดเดียวเท่านั้น

คำสั่งนี้จะใช้หมายเลขประจำตัวชุดควบคุมการเปิด-ปิดประตูหมายเลข 0 เท่านั้นเพื่อใช้ในการอ่านรหัสประจำตัวชุดควบคุมการเปิด-ปิดประตู เพื่อใช้ในการเปลี่ยนรหัสใหม่เท่านั้น คำสั่งจะใช้รหัส ASCII คือ 1 (31H)

@	0	1	#
---	---	---	---

การตอบกลับจะขึ้นต้นด้วยไบต์เริ่มต้นคือ & (26H) และตามด้วย ADDR> ตามด้วยหมายเลขประจำตัวเครื่อง แล้วจบด้วยไบต์จบคือ \$ (24H) เช่นอ่านหมายเลขประจำตัวเครื่องได้หมายเลข 1

&	A	D	D	>	1	\$
---	---	---	---	---	---	----

2. คำสั่งที่ใช้ในการเขียนหมายเลขประจำตัวชุดควบคุมการเปิด-ปิดประตู โดยต้องต่อเครื่องไว้เครื่องเดียวเท่านั้น

คำสั่งจะขึ้นต้นด้วยรหัส 2 (32H) ตามด้วยหมายเลขประจำตัวเครื่องใหม่ที่จะเปลี่ยน แล้วจบด้วย # (23H)

@	0	2	7	#
---	---	---	---	---

การตอบกลับจะขึ้นต้นด้วย & (26H) ตามด้วย ADDR> ตามด้วย OK แล้วจบด้วย \$ (24H)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

&	A	D	D	>	O	K	\$
---	---	---	---	---	---	---	----

3. คำสั่งที่ใช้ในถาดควบคุมการเปิด-ปิดประตู ว่าต้องการส่งข้อมูลรหัสบัตรมาหรือไม่ คำสั่งจะใช้รหัส 3(33H)

@	2	3	#
---	---	---	---

การตอบกลับจะมี 2 กรณีด้วยกัน คือ

-- กรณีที่ไม่มีข้อมูลรหัสบัตรส่งกลับมา จะตอบด้วยรหัส &(26H) ตามด้วยรหัสเครื่องที่ส่งกลับมาแล้วตามด้วย ,(2CH) ตามด้วย ID> แล้วตามด้วย NO แล้วจบด้วย \$(24H)

&	2	,	I	D	>	N	O	\$
---	---	---	---	---	---	---	---	----

-- กรณีที่มีข้อมูลรหัสบัตรส่งกลับมา จะตอบด้วย &(26H) ตามด้วยรหัสเครื่องที่ส่งกลับมาแล้วตามด้วย ,(2CH) ตามด้วยรหัสบัตร 14 หลัก แล้วตามด้วย LRC HIGH,LRC LOW แล้วจบด้วย \$(24H)

&	2	,	I	D	>	0	0	0	0	0	1	2	3	4	5	6	7	8	9	LRC (HIGH)	LRC (LOW)	\$
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------------	-----------	----

4. คำสั่งที่ใช้ส่งข้อมูลจำนวนเงินของบัตรแบบเติมเงิน (Prepaid) ไปแสดงผลที่จอ LCD ของชุดควบคุมการเปิด-ปิดประตู คำสั่งมีรหัส 4(34H) ตามด้วยจำนวนเงินที่เหลือของบัตรแบบเติมเงิน เช่น มีเงินจำนวน 125 หน่วย

@	7	4	0	1	2	5	#
---	---	---	---	---	---	---	---

คำสั่งนี้ไม่มีการตอบกลับมายังที่ตัวคอมพิวเตอร์

5. คำสั่งที่ใช้ส่งข้อมูลจำนวนเงินของบัตรแบบรายเดือน (Postpaid) ไปแสดงผลที่จอ LCD ของชุดควบคุมการเปิด-ปิดประตู คำสั่งมีรหัส 5(35H) ตามด้วยจำนวนเงินรวมของบัตรแบบรายเดือน เช่น สมมติมียอดรวม 1200 หน่วย

@	8	5	I	2	0	0	#
---	---	---	---	---	---	---	---

คำสั่งนี้ไม่มีการตอบกลับมายังที่ตัวคอมพิวเตอร์

6. คำสั่งที่ใช้บอกว่าการส่งข้อมูลรหัสบัตรไปยังคอมพิวเตอร์มีความผิดพลาดในการสื่อสารเกิดขึ้น ด้วยการตรวจสอบ LRC ชุดควบคุมการเปิด-ปิดประตูจะแสดงผลที่ LCD ว่า Communication Error โดยจะใช้รหัสคำสั่ง 6(36H)

@	4	6	#
---	---	---	---

7. คำสั่งที่ใช้บอกว่ามีผิดพลาดจากระบบฐานข้อมูล ไม่ว่าจะเป็ความผิดพลาดจากการเชื่อมต่อ หรือ จำนวนเงินที่ไม่พอจะเข้าใช้บริการของบัตรแบบเติมเงิน โดยจะใช้รหัสคำสั่ง 7(37H)

@	1	7	#
---	---	---	---

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ความผิดพลาดต่างๆที่ส่งไปยังคอมพิวเตอร์ของชุดควบคุมการเปิด-ปิดประตู

1.

&	2	,	E	R	R	=	1	S
---	---	---	---	---	---	---	---	---

ความผิดพลาดที่เกิดจากการไม่สามารถกำหนดหมายเลขประจำชุดควบคุมการเปิด-ปิดประตูใหม่ได้เนื่องจากไม่สามารถเขียนข้อมูลลงหน่วยความจำได้

2.

&	3	,	E	R	R	=	2	S
---	---	---	---	---	---	---	---	---

ความผิดพลาดที่เกิดจากการตั้งหมายเลขประจำชุดควบคุมการเปิด-ปิดประตูไม่อยู่ในช่วงที่กำหนดไว้

3.

&	6	,	E	R	R	=	3	S
---	---	---	---	---	---	---	---	---

ความผิดพลาดที่เกิดจากการส่งคำสั่งที่ไม่มีในข้อกำหนดหรือคำสั่งที่ไม่สมบูรณ์ในการติดต่อสื่อสารกันระหว่างชุดควบคุมการเปิด-ปิดประตูกับ คอมพิวเตอร์

#### 3.3.4 การทำงานของโปรแกรม

ชุดควบคุมการเปิด-ปิดประตูทางเข้าออกสถานีรถไฟฟ้ามหานครเมื่อเปิดเครื่องเริ่มทำงานขึ้นมาจะแสดงหมายเลขรุ่นของ โปรแกรม ต่อด้วยการแสดงหมายเลขประจำตัวของเครื่องและทิศทางการติดตั้ง (ทางเข้า-ทางออก) ซึ่งสามารถโปรแกรมใหม่ได้ ตามข้อกำหนดการเชื่อมต่อ โดยใช้โปรแกรมกำหนดหมายเลขประจำชุดควบคุมการเปิด-ปิดประตูทางเข้าออกบนเครื่องคอมพิวเตอร์ สำหรับให้ผู้ที่ทำการติดตั้งกำหนดหมายเลขประจำตัวเครื่องใหม่ได้

เมื่อแสดงหมายเลขเครื่องเสร็จแล้วโปรแกรมจะรอเข้าสู่การทำงานของโปรแกรมหลักประมาณ 3 วินาที เมื่อเข้าสู่โปรแกรมหลักแล้วจะแสดงผลที่ LCD ออกมาเป็นข้อความแสดงตามทิศทางการเข้าออกที่ได้กำหนดไว้ให้กับเครื่องนั้นๆ

การรับส่งข้อมูลกับคอมพิวเตอร์จะใช้ตามข้อกำหนดการติดต่อสื่อสารที่ได้กำหนดไว้ในหัวข้อ 3.3.3 โดยจะคอยตรวจสอบคำสั่งที่ส่งมาจากคอมพิวเตอร์ว่าตรงตามหมายเลขของตัวเองหรือไม่ถ้าใช่ก็จะปฏิบัติงานตามคำสั่งนั้น แต่ถ้าไม่ตรงก็จะละทิ้งคำสั่งนั้นไป

ชุดควบคุมการเปิด-ปิดประตูทางเข้าออก เมื่อได้รับการอินเตอร์รัปต์จากหัวอ่านบัตรข้อมูลจนครบจะทำการตรวจสอบข้อมูลและรอคำสั่งขอรหัสบัตรจากคอมพิวเตอร์ถ้าคอมพิวเตอร์ไม่ส่งคำสั่งมาเกิน 2 วินาที เครื่องจะแสดงผลที่ LCD ว่าเกิดความผิดพลาดจากการสื่อสารและมีเสียงบี๊บเซอร์ดังสั้นๆ 3 ครั้ง

การอ่านรหัสจากบัตร RFID เมื่ออ่านได้แล้วส่งให้คอมพิวเตอร์ ถ้าเกิดความผิดพลาดจากการส่งข้อมูลจากการตรวจสอบ LRC ก็จะแสดงว่าเกิดความผิดพลาดจากการสื่อสารเช่นกัน แต่ถ้าเกิดจากการติดต่อกับฐานข้อมูลก็แสดงข้อความออกที่จอ LCD ว่าเกิดปัญหาจากฐานข้อมูล

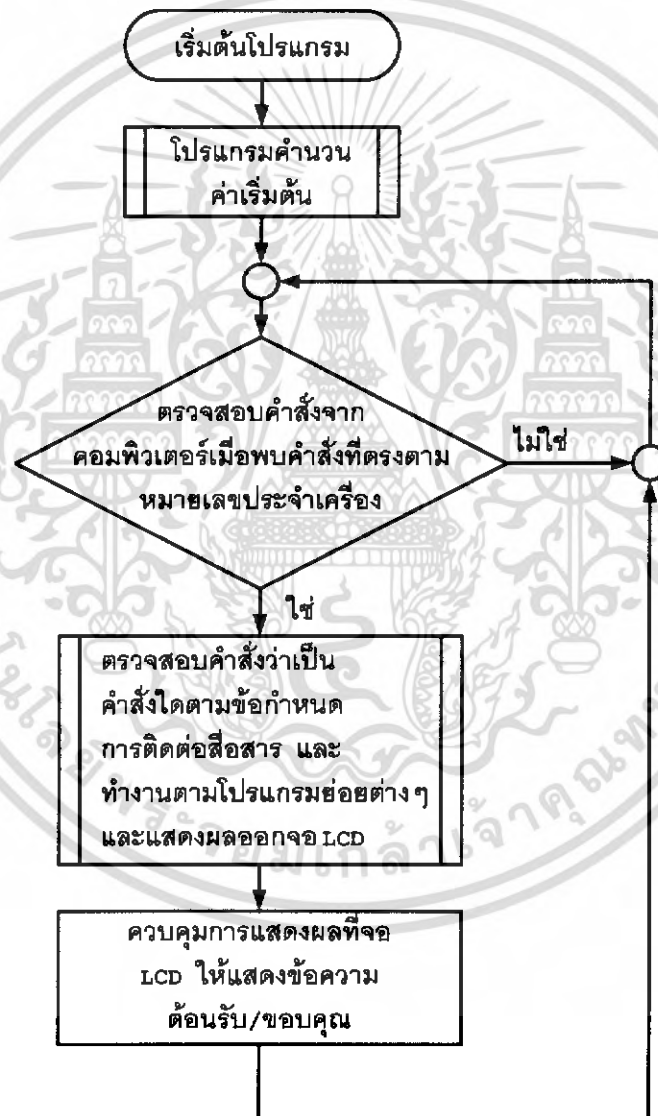
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านรหัสบาร์อาจเกิดความผิดพลาดขึ้นได้ในบางกรณี เช่น นำบาร์ผ่านหัวอ่านเร็วไปทำให้รับข้อมูลได้ไม่ครบ เป็นต้น

ชุดควบคุมการเปิด-ปิดประตูทางเข้าออก ถ้าเป็นชุดทางเข้า LCD จะแสดงข้อความจำนวนเงินก่อนการให้บริการ แต่ถ้าเป็นด้านออกจะมีการอ่านรหัสบาร์ที่ถูกต้องและค้นหาข้อมูลปรับปรุงจำนวนเงินในฐานข้อมูลเสร็จแล้วก็จะแสดงผลออกมาที่ LCD

### 3.3.5 ฟังก์ชันการทำงานของชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

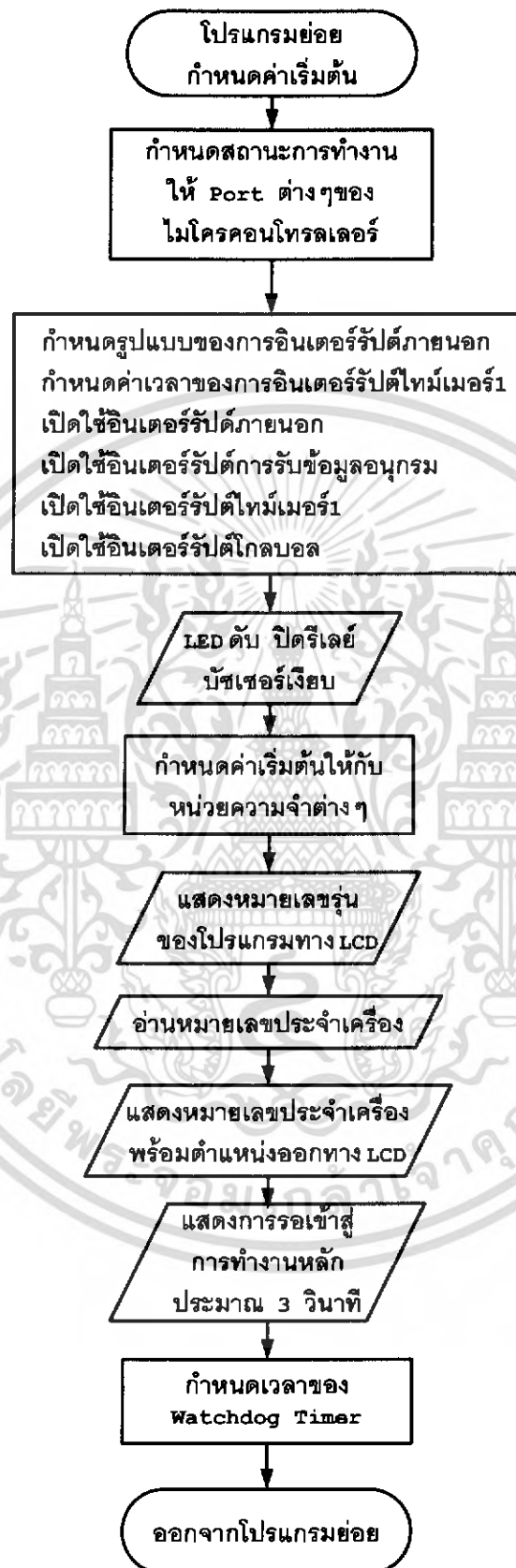
#### 3.3.5.1 ฟังก์ชันการทำงานของโปรแกรมหลัก



รูปที่ 3.20 ฟังก์ชันการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

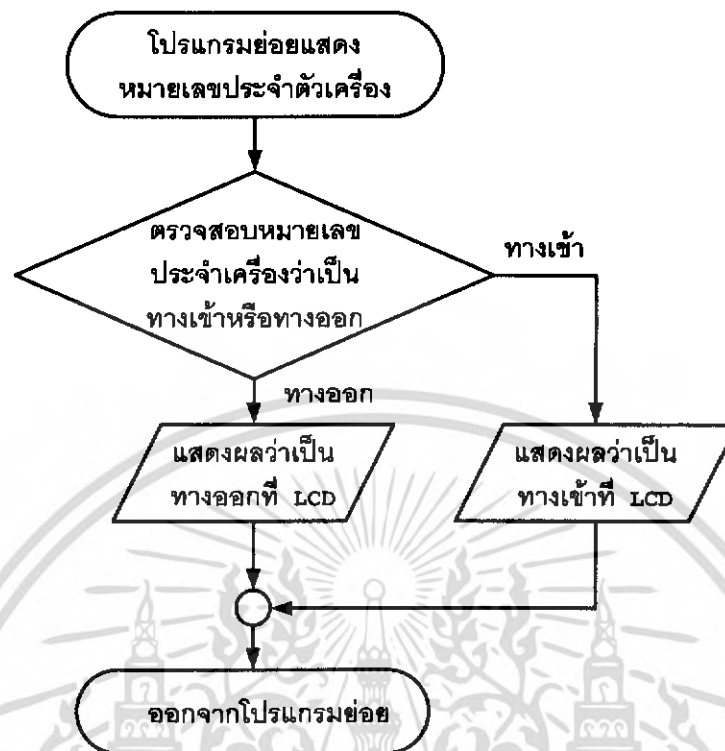
### 3.3.5.2 ฟังก์ชันการทำงานของโปรแกรมย่อยกำหนดค่าเริ่มต้น



**รูปที่ 3.21** ฟังก์ชันการทำงานของโปรแกรมย่อยกำหนดค่าเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

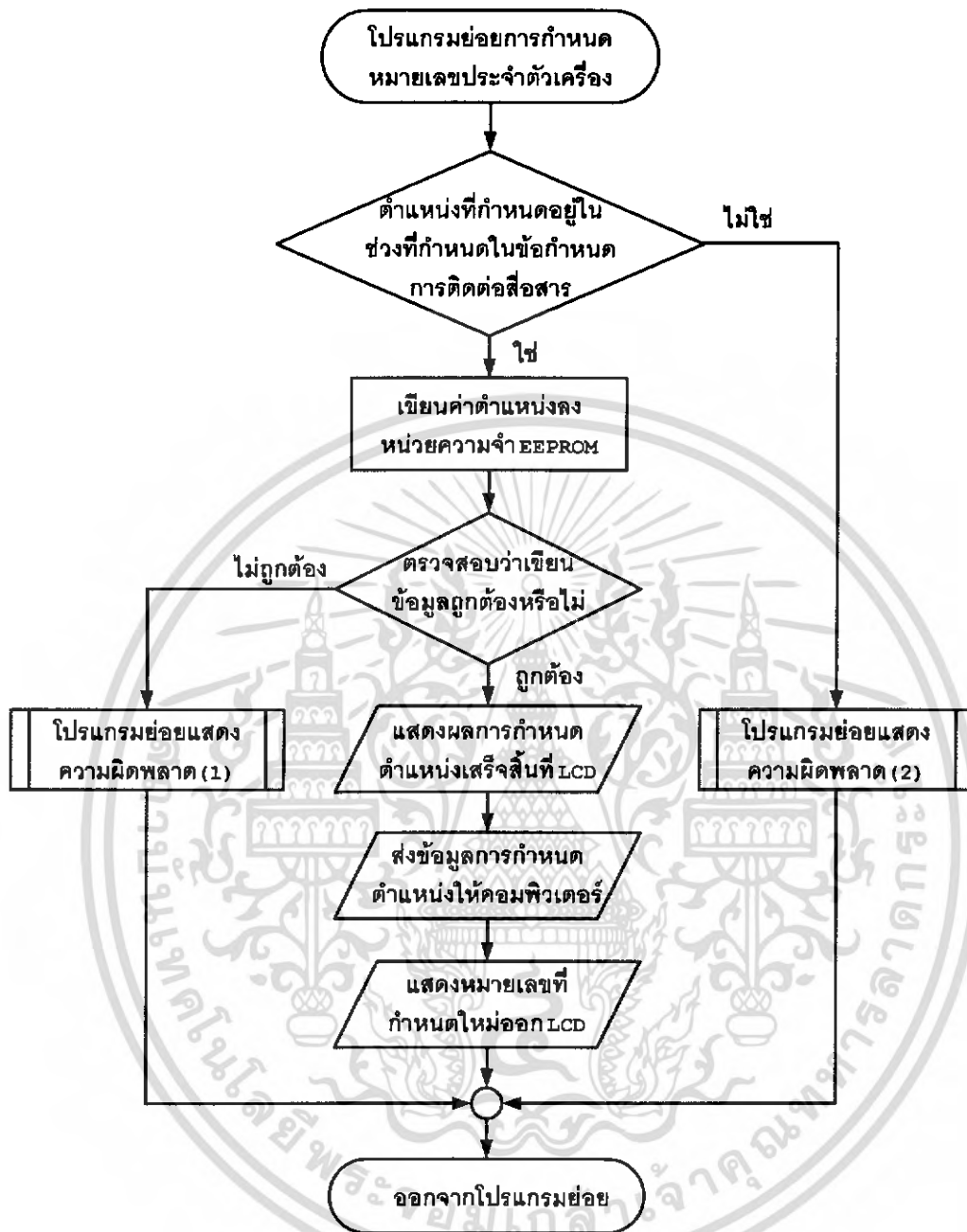
### 3.3.5.3 ฟังก์ชันการทำงานของโปรแกรมย่อยแสดงหมายเลขประจำเครื่อง



รูปที่ 3.22 ฟังก์ชันการทำงานของโปรแกรมย่อยแสดงหมายเลขประจำเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5.4 ฟังก์ชันการทำงานของโปรแกรมย่อยกำหนดหมายเลขประจำตัวเครื่อง

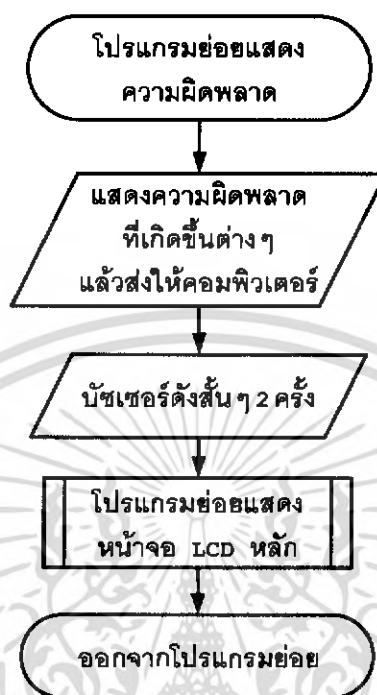


รูปที่ 3.23 ฟังก์ชันการทำงานของโปรแกรมย่อยกำหนดหมายเลขประจำตัวเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5.5 ผังการทำงานของโปรแกรมย่อยแสดงความผิดพลาด

โปรแกรมย่อยจะแสดงข้อความที่เกิดจากความผิดพลาดต่างๆที่เกิดขึ้น ไม่ว่าจะจากการสื่อสารหรือจากระบบฐานข้อมูลก็ตามจะนำไปแสดงผลที่จอ LCD และส่งให้คอมพิวเตอร์

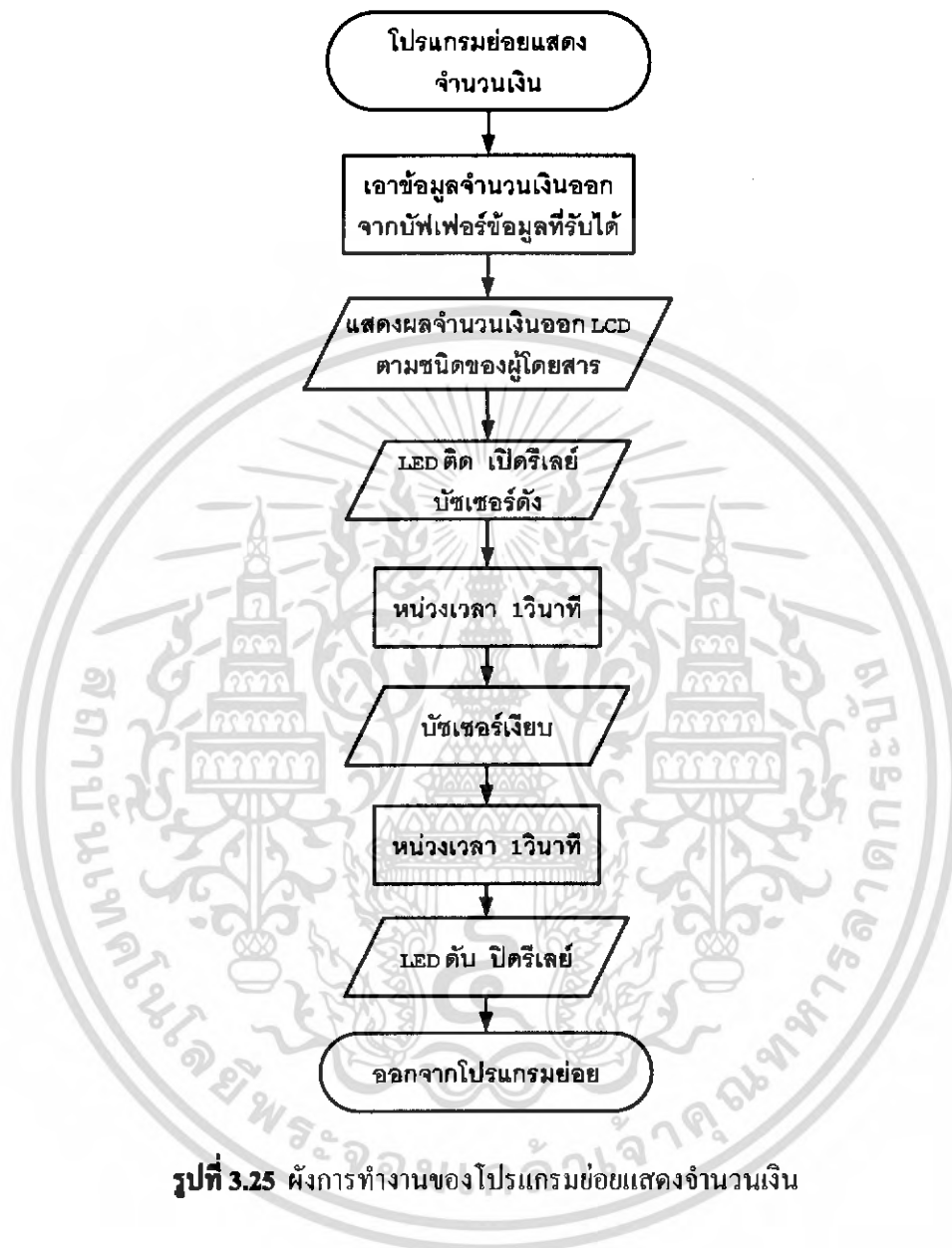


รูปที่ 3.24 ผังการทำงานของโปรแกรมย่อยแสดงความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5.6 ผังการทำงานของโปรแกรมย่อยแสดงจำนวนเงิน

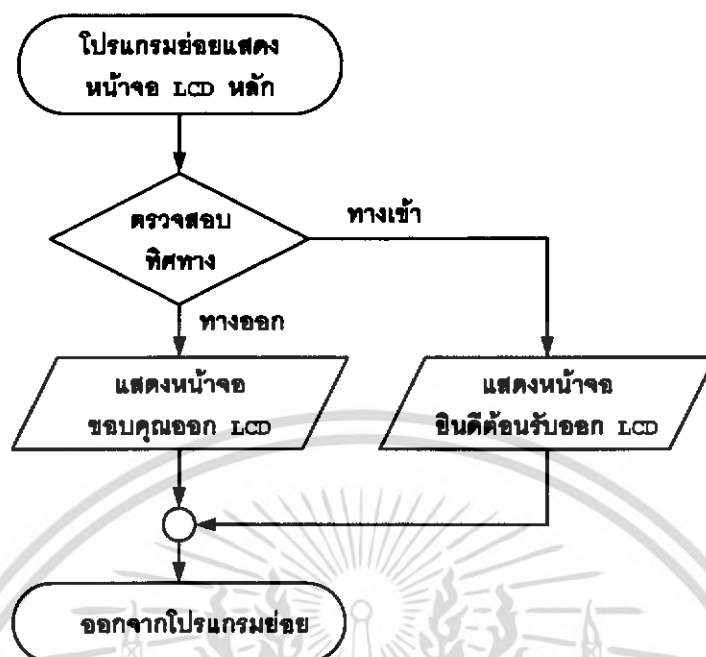
โปรแกรมย่อยแสดงจำนวนเงิน จะแสดงจำนวนเงินของผู้โดยสารรถไฟฟ้ที่ช่องทางออก โดยแสดงข้อมูลจำนวนเงินของบัตรทั้งแบบเติมเงินและแบบรายเดือนที่ได้รับจากคอมพิวเตอร์



รูปที่ 3.25 ผังการทำงานของโปรแกรมย่อยแสดงจำนวนเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

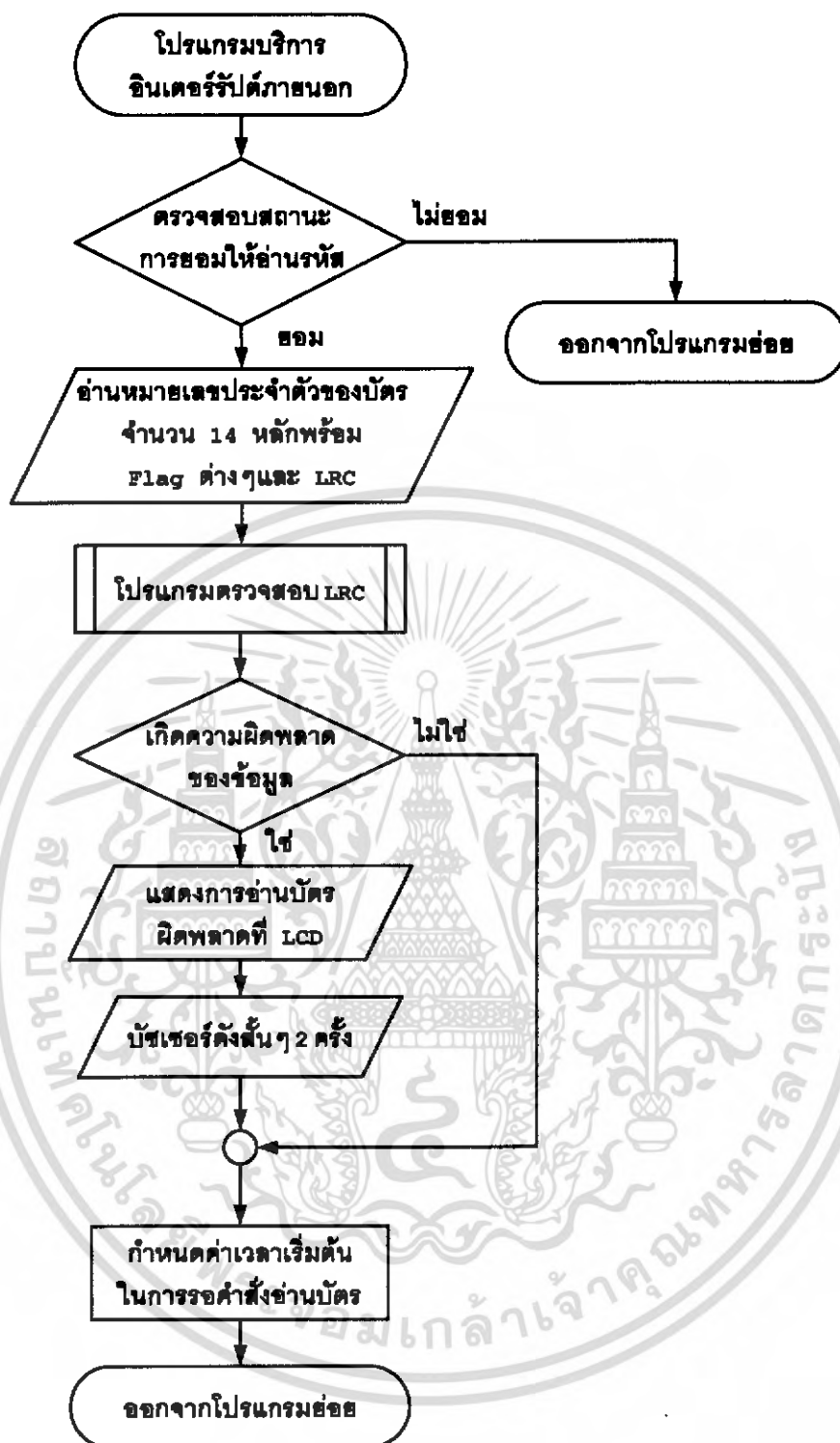
### 3.3.5.7 ฟังก์ชันการทำงานของโปรแกรมย่อยแสดงหน้าจอ LCD หลัก



รูปที่ 3.26 ฟังก์ชันการทำงานของโปรแกรมย่อยแสดงหน้าจอ LCD หลัก

### 3.3.5.8 ฟังก์ชันการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้สายภายนอก

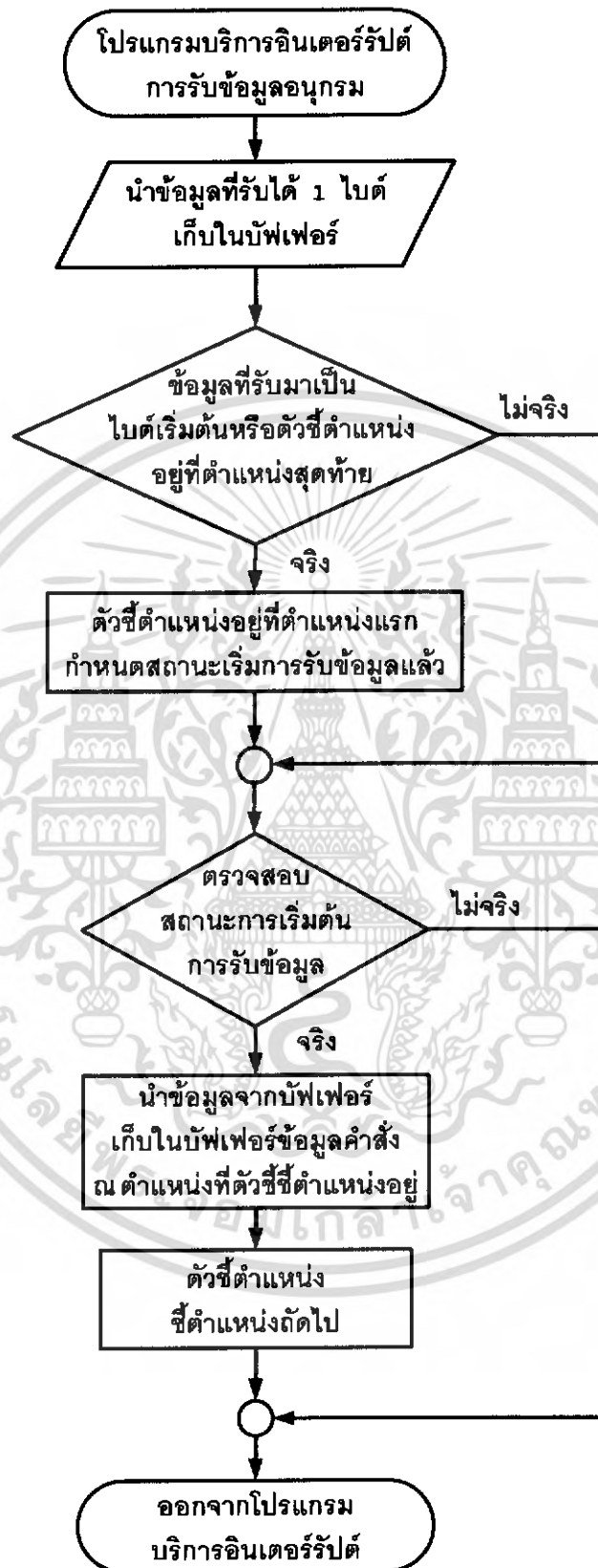
โปรแกรมบริการอินเทอร์เน็ตไร้สายภายนอก จะใช้ในการอ่านข้อมูลรหัสบัตรจากหัวอ่านบัตร RFID เมื่อมีข้อมูลส่งจากหัวอ่านบัตรก็จะได้รับการบริการอินเทอร์เน็ตไร้สายภายนอกแล้วทำการอ่านข้อมูลรหัสบัตรที่ได้ 14 หลัก



รูปที่ 3.27 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

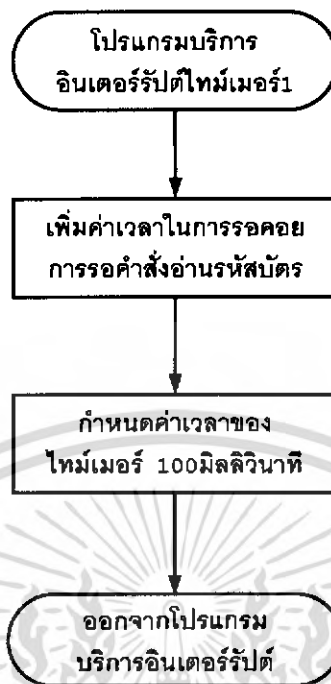
### 3.3.5.9 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตรับข้อมูลอนุกรม



รูปที่ 3.28 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตการรับข้อมูลอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

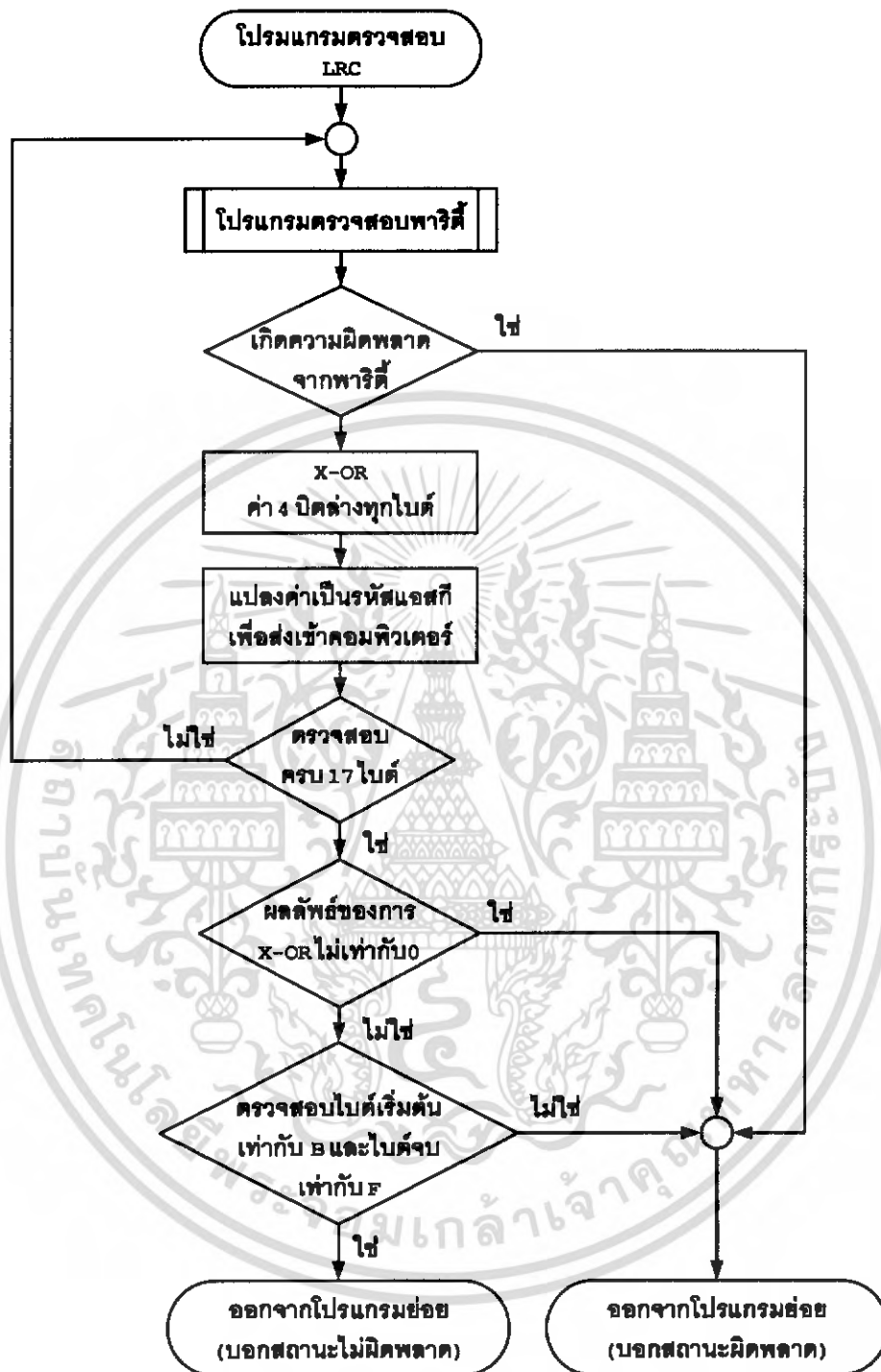
### 3.3.5.10 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้ทีโหมเมอร์1



รูปที่ 3.29 ผังการทำงานของโปรแกรมบริการอินเทอร์เน็ตไร้ทีโหมเมอร์1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.3.5.11 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบ LRC

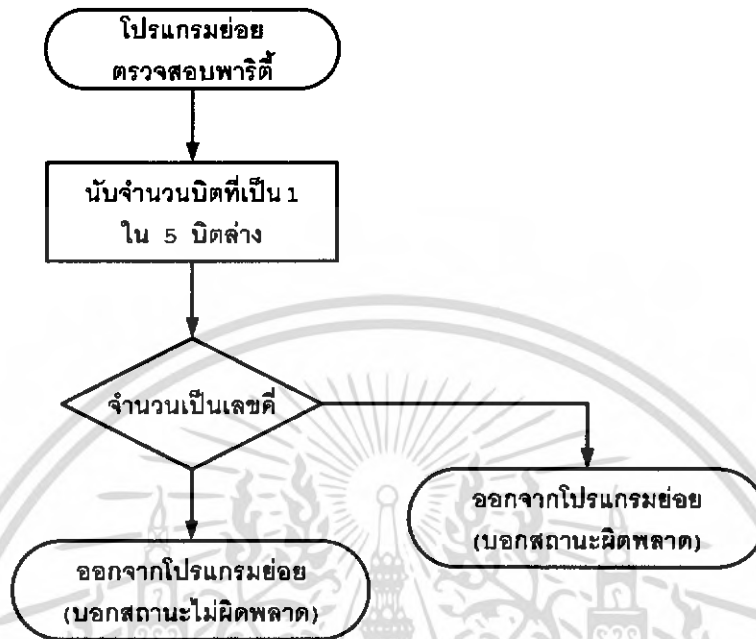


รูปที่ 3.30 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบ LRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5.12 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบพาริตี

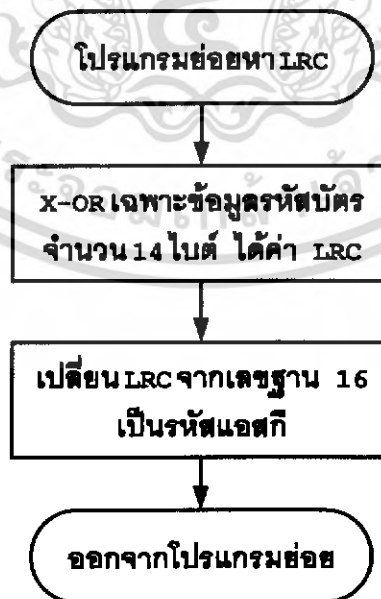
ในการส่งข้อมูลรหัสบัตรจากหัวอ่านบัตรจะส่งพาริตีเพื่อใช้ตรวจสอบความผิดพลาด โดยใช้พาริตีที่มีฟังก์ชันการทำงานดังนี้



รูปที่ 3.31 ฟังก์ชันการทำงานของโปรแกรมย่อยตรวจสอบพาริตี

### 3.3.5.13 ฟังก์ชันการทำงานของโปรแกรมย่อยหา LRC

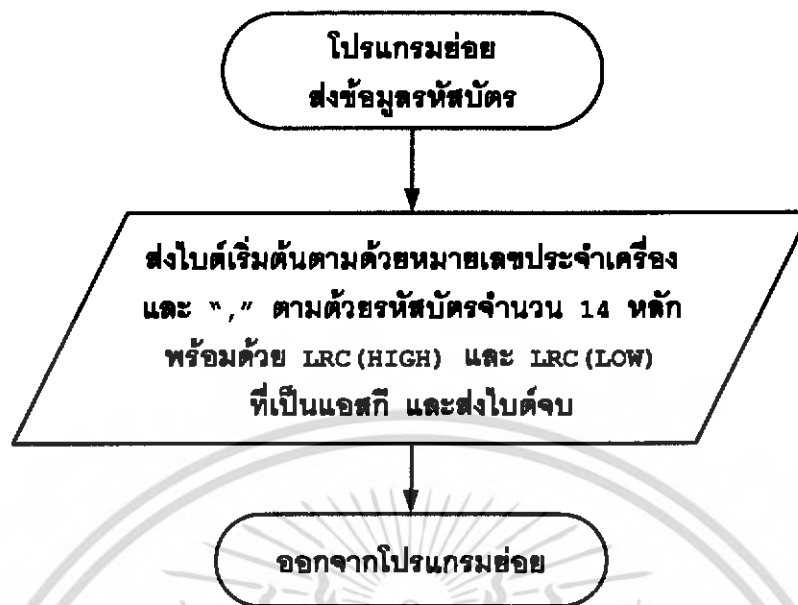
เป็นโปรแกรมใช้ในการหา LRC เพื่อใช้ในการตรวจสอบความผิดพลาดของข้อมูลที่ส่งให้คอมพิวเตอร์



รูปที่ 3.32 ฟังก์ชันการทำงานของโปรแกรมย่อยหา LRC

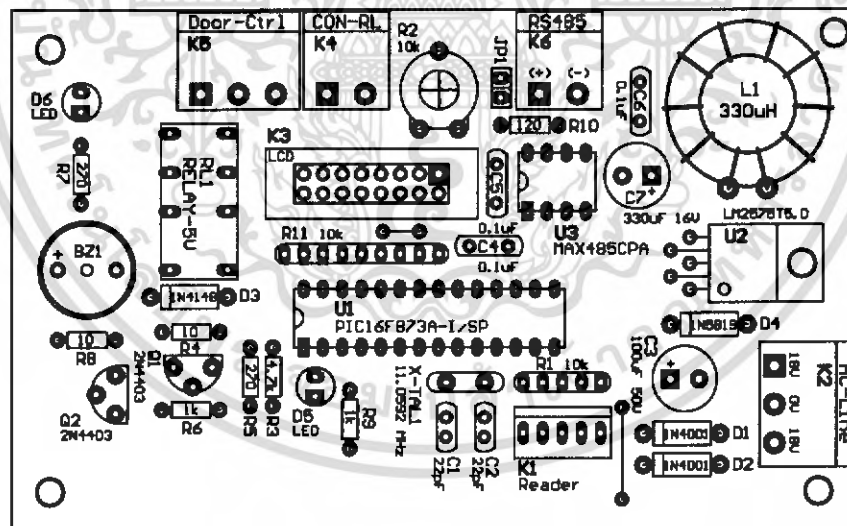
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงพาณิชย์เท่านั้น เมื่อผู้ผู้เช่าเห็นว่าไม่คุ้มค่าหรือไม่กว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5.14 ฟังก์ชันการทำงานของโปรแกรมย่อยส่งข้อมูลรหัสบัตร

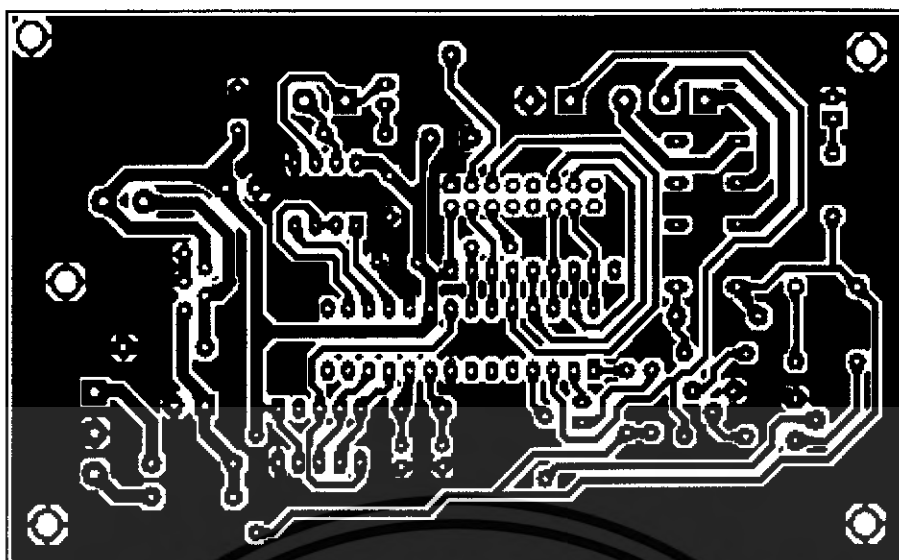


รูปที่ 3.33 ฟังก์ชันการทำงานของโปรแกรมย่อยส่งข้อมูลรหัสบัตร

### 3.3.6 การออกแบบลายวงจร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.34 ลายวงจรและการวางอุปกรณ์

### 3.4 การออกแบบฐานข้อมูลที่ใช้กับระบบการให้บริการรถไฟฟ้า

โปรแกรมระบบฐานข้อมูลที่ใช้กับระบบการให้บริการรถไฟฟ้า ใช้โปรแกรม MySQL ในการเขียนและทำการเชื่อมต่อกับโปรแกรมต่างๆ ผ่าน ODBC Connector การสร้างตารางของฐานข้อมูลจะใช้โปรแกรม SQLyog Enterprise ซึ่งเป็น GUI ที่ติดต่อกับ MySQL เข้ามาช่วยในการสร้างตารางฐานข้อมูลซึ่งประกอบด้วย

- ตารางใช้ในการ Log On เข้ามาใช้ระบบฐานข้อมูล
- ตารางราคาค่าโดยสารของแต่ละสถานีไปยังสถานีปลายทางใดๆ
- ตารางบอกชนิดของบัตร
- ตารางเก็บจำนวนเงินคงเหลือของผู้ใช้บริการแบบเติมเงิน
- ตารางเก็บยอดเงินรวมในการใช้บริการของผู้ใช้บริการรายเดือน
- ตารางเก็บข้อมูลที่อยู่เพื่อใช้ในการเก็บค่าใช้บริการของผู้ใช้บริการรายเดือน
- ตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของผู้ใช้บริการแบบเติมเงิน
- ตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของผู้ใช้บริการรายเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1 ตารางที่ใช้ในการ Log On

ตารางที่ใช้ในการ Log On เข้ามาใช้ระบบฐานข้อมูล มีด้วยกัน 4 필ด์คือ

Field Name	Datatype	Len	Default	PK?	Binary?	Not Null?	Unsigned?	Auto Incr?	Zerofill?
st_index	int	2		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
StationID	char	2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Station	char	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Password	char	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

รูปที่ 3.35 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางการ Log On

st_index	StationID	Station	Password
6	N2	พญาไท (N2)	n2
7	N1	ราชเทวี (N1)	n1
8	CS	สยาม (CS)	c๑
9	E1	ชิดลม (E1)	e1
10	E2	เพลินจิต (E2)	e2
11	E3	นานา (E3)	e3
12	E4	อโศก (E4)	e4
13	E5	พร้อมพงษ์ (E5)	e5
14	E6	ทองหล่อ (E6)	e6
15	E7	เอกมัย (E7)	e7
16	E8	พระโขนง (E8)	e8
17	E9	อ่อนนุช (E9)	e9
18	W1	สนามกีฬาแห่งชาติ (W1)	w1
19	S1	ราชดำริ (S1)	s1
20	S2	ศาลาแดง (S2)	s2
21	S3	ช่องนนทรี (S3)	s3
22	S5	สุรศักดิ์ (S5)	s5
23	S6	สะพานตากสิน (S6)	s6

รูปที่ 3.36 แสดงตารางที่ใช้ในการ Log On เข้ามาใช้ระบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.2 ตารางราคาค่าโดยสารของแต่ละสถานีไปยังสถานีปลายทางใดๆ

ตารางราคาค่าโดยสารของแต่ละสถานีไปยังสถานีปลายทางใดๆ มีด้วยกัน 3 ฟิลด์คือ

Field Name	Datatype	Len	Default	PK?	Binary?	Not Null?	Unsigned?	Auto Incr?	Zerofill?
StationID	char	2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Station	char	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Price	int	3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

รูปที่ 3.37 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางราคาค่าโดยสาร

StationID	Station	Price
N8	หมอซิด (N8)	30
N7	สะพานควาย (N7)	30
N6	อารีย์ (N5)	25
N4	สนามเป้า (N4)	20
N3	อนุสาวรีย์ชัยสมรภูมิ (N3)	20
N2	พญาไท (N2)	15
N1	ราชเทวี (N1)	10
E1	ชิดลม (E1)	10
E2	เพลินจิต (E2)	15
E3	นานา (E3)	20
E4	อโศก (E4)	20
E5	พหลโยธิน (E5)	25
E6	ทองหล่อ (E6)	25
E7	เอกมัย (E7)	30
E8	พระโขนง (E8)	30
E9	อ่อนนุช (E9)	35
W1	สนามกีฬาแห่งชาติ (W1)	10
S1	ราชดำริ (S1)	10
S2	ศาลาแดง (S2)	15
S3	ช่องนนทรี (S3)	20
S5	สุรศักดิ์ (S5)	25
S6	สะพานตากสิน (S6)	25

รูปที่ 3.38 แสดงตารางราคาค่าโดยสารของแต่ละสถานีไปยังสถานีปลายทางใดๆ

ตารางราคาค่าโดยสารนี้ซื้อตารางจะเป็นสถานีต้นทางและข้อมูลในตารางจะเป็นค่าโดยสารของสถานีปลายทางใดๆ ซึ่งทำการสร้างตารางตามสถานีที่มีในปัจจุบัน (2549) คือ 23 สถานี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

price\_cs  
price\_e1  
price\_e2  
price\_e3  
price\_e4  
price\_e5  
price\_e6  
price\_e7  
price\_e8  
price\_e9  
price\_n1  
price\_n2  
price\_n3  
price\_n4  
price\_n5  
price\_n7  
price\_n8  
price\_s1  
price\_s2  
price\_s3  
price\_s5  
price\_s6  
price\_w1

รูปที่ 3.39 แสดงตารางราคาโดยสารที่สถานีต้นทาง

### 3.4.3 ตารางบอกชนิดของบัตร

Field Name	Datatype	Len	Default	PK?	Binary?	Not Null?	Unsigned?	Auto Incr?	Zerofill?
Index	int	5		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Card_ID	char	14		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Type	char	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

รูปที่ 3.40 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางบอกชนิดของบัตร

Index	Card_ID	Type
1	00016791130760	postpaid
2	00016791108905	prepaid
3	00016791108863	prepaid

รูปที่ 3.41 แสดงตารางบอกชนิดของบัตร

### 3.4.4 ตารางเก็บจำนวนเงินคงเหลือของผู้ใช้บริการแบบเติมเงิน

Field Name	Datatype	Len	Default	PK?	Binary?	Not Null?	Unsigned?	Auto Incr?	Zerofill?
Card_ID	char	14		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
credit	int	4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

รูปที่ 3.42 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางจำนวนเงินคงเหลือของผู้ใช้บริการ

แบบเติมเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Card_ID	credit
00018791108905	150
00018791108863	150

รูปที่ 3.43 แสดงตารางเก็บจำนวนเงินคงเหลือของผู้ใช้บริการแบบเติมเงิน

### 3.4.5 ตารางเก็บยอดเงินรวมในการใช้บริการของผู้ใช้บริการรายเดือน

Field Name	Datatype	Len	Default	PK?	Binary?	Not Null?	Unsigned?	Auto Incr?	Zerofill?
Card_ID	char	14		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
amount	int	4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

รูปที่ 3.44 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางยอดเงินรวมของผู้ใช้บริการรายเดือน

Card_ID	amount
00018791130780	65

รูปที่ 3.45 แสดงตารางเก็บยอดเงินรวมในการใช้บริการของผู้ใช้บริการรายเดือน

### 3.4.6 ตารางเก็บข้อมูลที่อยู่เพื่อใช้ในการเก็บค่าบริการของผู้ใช้บริการรายเดือน

Field Name	Datatype	Len	Default	PK?	Binary?	Not Null?	Unsigned?	Auto Incr?	Zerofill?
Customer_ID	int	8		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Card_ID	char	14		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FirstName	char	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LastName	char	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Personal_ID	char	13		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Address	char	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
City	char	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zip-code	char	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Telephone	char	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

รูปที่ 3.46 แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางข้อมูลที่อยู่เพื่อใช้ในการเก็บค่าบริการของผู้ใช้บริการรายเดือน

Customer_ID	Card_ID	FirstName	LastName	Personal_ID	Address	City	Zip-code	Telephone
1	00018791130780	นายศกานี	สกุลโพธิ์	1234567890123	1 ถนนพหลโยธิน 2 แขวงท่าช้าง กรุงเทพมหานคร	10000	10000	021234567

รูปที่ 3.47 แสดงตารางเก็บข้อมูลที่อยู่เพื่อใช้ในการเก็บค่าบริการของผู้ใช้บริการรายเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.7 ตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของผู้ใช้บริการแบบเติมเงิน

Field Name	Datatype	Len	Default	PK?	Binary?	Not Null?	Unsigned?	Auto Incr?	Zerofill?
order	int	5		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
source	char	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
in_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
destination	char	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
out_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
price	int	3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**รูปที่ 3.48** แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางสถานะการเข้าออกในการโดยสารของผู้ใช้บริการแบบเติมเงิน

order	source	in_time	destination	out_time	price
1	นมชิต (N8)	2006-03-20 16:26:18	เจมัม (E7)	2006-03-20 16:26:23	40
2	นานา (E3)	2006-03-20 16:27:01	ช่องนนทรี (S3)	2006-03-20 16:27:17	25

**รูปที่ 3.49** แสดงตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของผู้ใช้บริการแบบเติมเงิน

### 3.4.8 ตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของผู้บริการรายเดือน

Field Name	Datatype	Len	Default	PK?	Binary?	Not Null?	Unsigned?	Auto Incr?	Zerofill?
order	int	5		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
source	char	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
in_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
destination	char	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
out_time	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
price	int	3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**รูปที่ 3.50** แสดงการกำหนดชนิดข้อมูลที่ใช้ในแต่ละฟิลด์ของตารางสถานะการเข้าออกในการโดยสารของผู้ใช้บริการแบบเติมเงิน

order	source	in_time	destination	out_time	price
1	นมชิต (N8)	2006-03-20 16:00:19	'สุรศักดิ์ (S5)'	2006-03-20 16:01:00	40
2	สุรศักดิ์ (S5)	2006-03-20 16:04:33	สนามกีฬาแห่งชาติ (W1)	2006-03-20 16:05:00	25

**รูปที่ 3.51** แสดงตารางเก็บสถานะการเข้าออกในการโดยสารแต่ละครั้งของผู้บริการรายเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

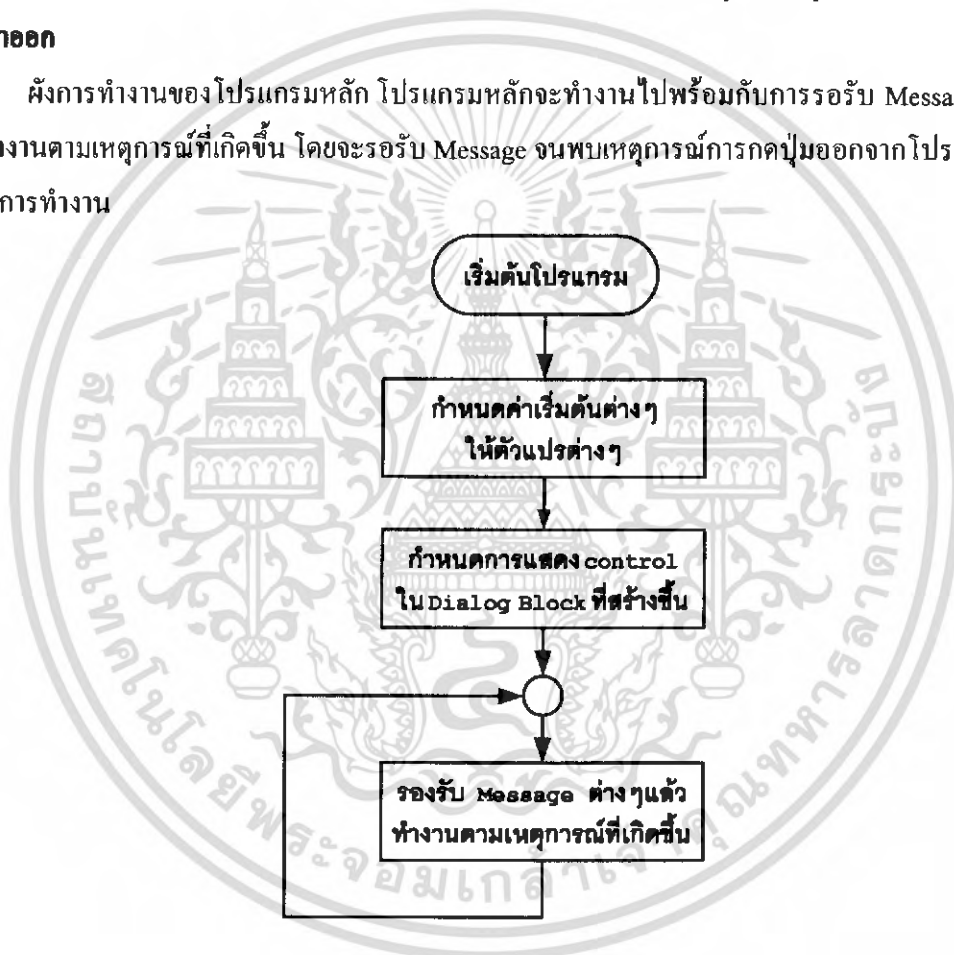
### 3.5 การออกแบบโปรแกรมกำหนดหมายเลขประจำชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

โปรแกรมกำหนดหมายเลขประจำชุดควบคุมการเปิด-ปิดประตูทางเข้าออก ใช้ในการกำหนดหมายเลขประจำเครื่องโดยมีเงื่อนไขการกำหนดหมายเลขประจำเครื่องตามหัวข้อ 3.3.3 เพื่อให้ผู้ติดตั้งเครื่องสามารถปรับเปลี่ยนหมายเลขเครื่องได้ให้ทำงานได้ไม่ว่าจะเป็นประตูทางเข้าหรือประตูทางออก

การเชื่อมต่อชุดควบคุมการเปิด-ปิดประตูทางเข้าออก กับคอมพิวเตอร์โดยใช้โปรแกรมนี้มีเงื่อนไขว่าจะต้องต่ออยู่ในระบบบัส RS-485 อยู่เพียงตัวเดียวเท่านั้น เพราะการเปลี่ยนหมายเลขนี้จะใช้หมายเลข 0 ในการเปลี่ยนทุกๆหมายเลข เป็นการ Broadcast ซึ่งมีอยู่ 2 คำสั่ง ตามข้อกำหนดในการสื่อสารที่กำหนดไว้

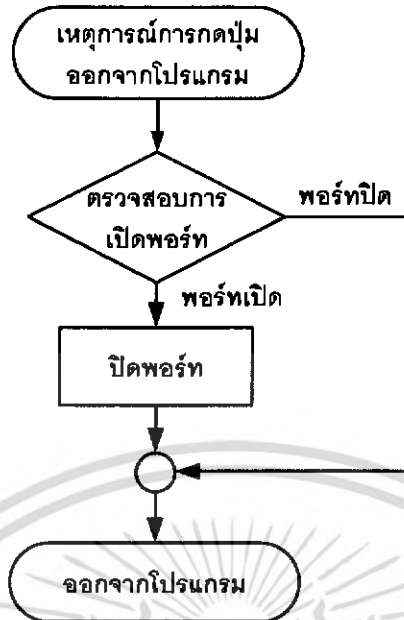
#### 3.5.1 ผังการทำงานของโปรแกรมกำหนดหมายเลขประจำชุดควบคุมการเปิด-ปิดประตูทางเข้าออก

ผังการทำงานของโปรแกรมหลัก โปรแกรมหลักจะทำงานไปพร้อมกับการรอรับ Message ต่างๆ แล้วทำงานตามเหตุการณ์ที่เกิดขึ้น โดยจะรอรับ Message จนพบเหตุการณ์การกดปุ่มออกจากโปรแกรมจึงยกเลิกการทำงาน



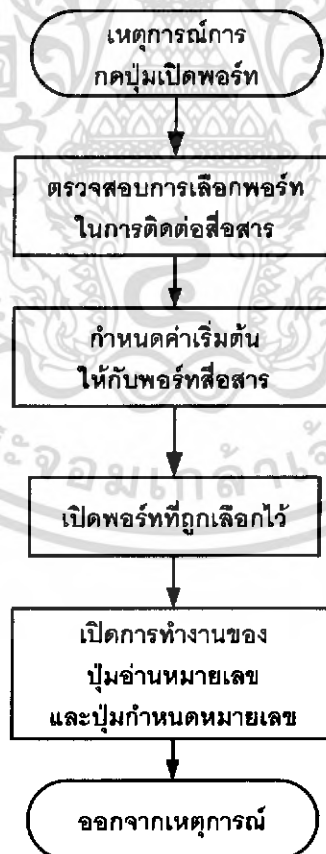
รูปที่ 3.52 ผังการทำงานของโปรแกรมหลัก

### 3.5.2 ผังการทำงานของเหตุการณ์การกดปุ่มออกจากโปรแกรม



รูปที่ 3.53 ผังการทำงานของเหตุการณ์การกดปุ่มออกจากโปรแกรม

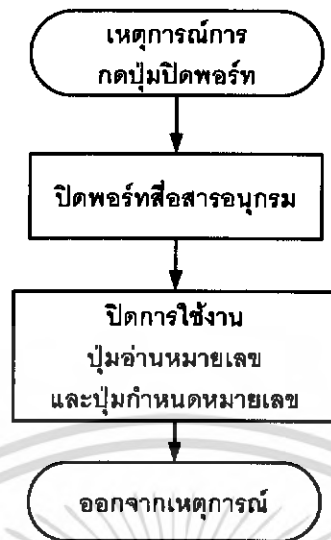
### 3.5.3 ผังการทำงานของเหตุการณ์การกดปุ่มเปิดพอร์ต



รูปที่ 3.54 ผังการทำงานของเหตุการณ์การกดปุ่มเปิดพอร์ต

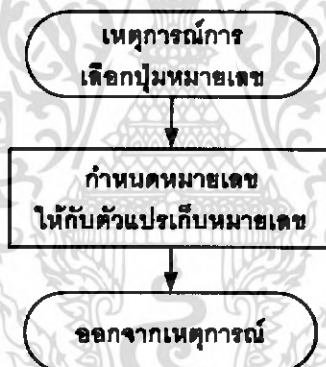
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.4 ผังการทำงานของเหตุการณ์การกดปุ่มปิดพอร์ท



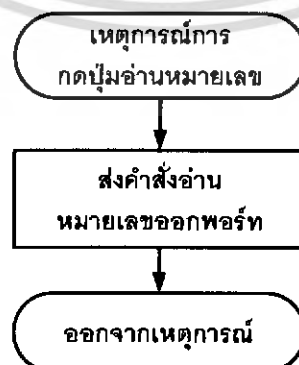
รูปที่ 3.55 ผังการทำงานของเหตุการณ์การกดปุ่มปิดพอร์ท

### 3.5.5 ผังการทำงานของเหตุการณ์เลือกปุ่มหมายเลข



รูปที่ 3.56 ผังการทำงานของเหตุการณ์เลือกปุ่มหมายเลข

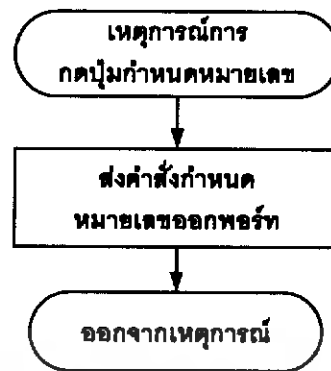
### 3.5.6 ผังการทำงานของเหตุการณ์การกดปุ่มอ่านหมายเลข



รูปที่ 3.57 ผังการทำงานของเหตุการณ์การกดปุ่มอ่านหมายเลข

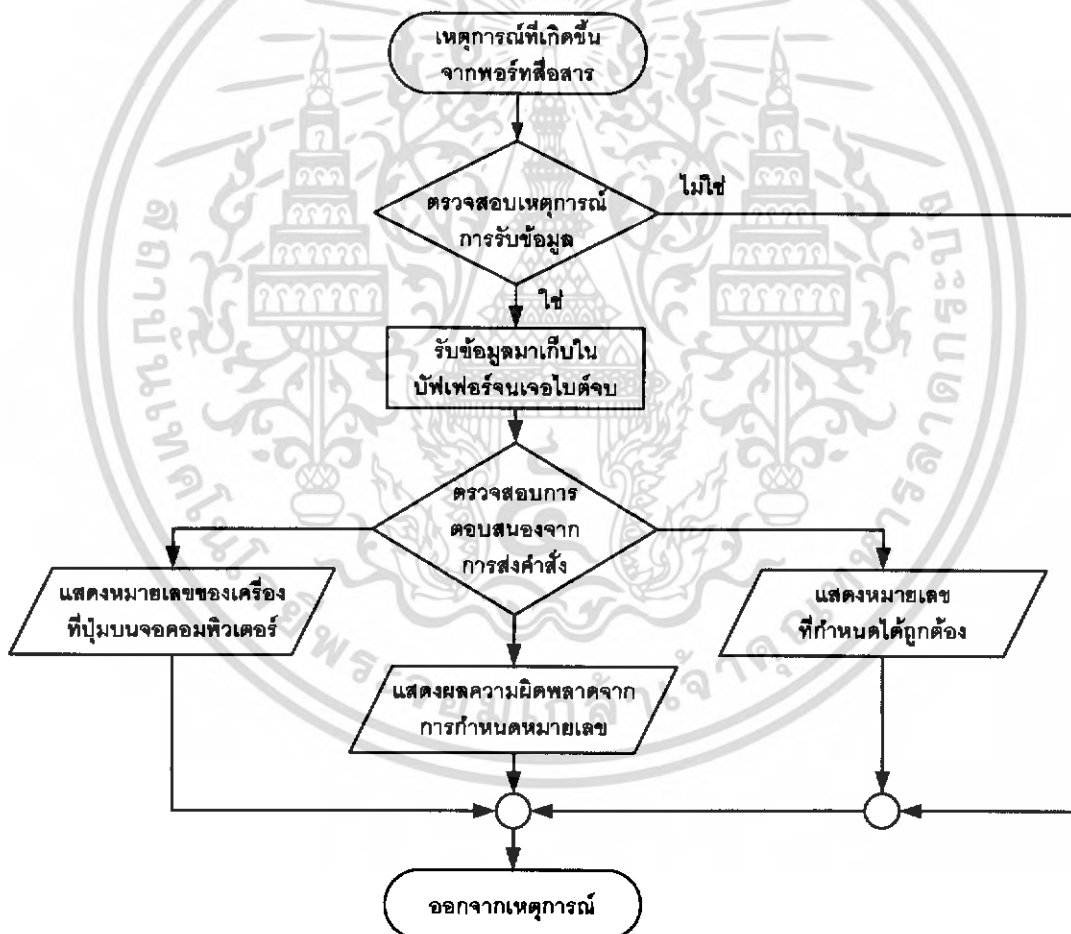
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.7 ผังการทำงานของเหตุการณ์การกดปุ่มกำหนดหมายเลข



รูปที่ 3.58 ผังการทำงานของเหตุการณ์การกดปุ่มกำหนดหมายเลข

### 3.5.8 ผังการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ตสื่อสาร



รูปที่ 3.59 ผังการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ตสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

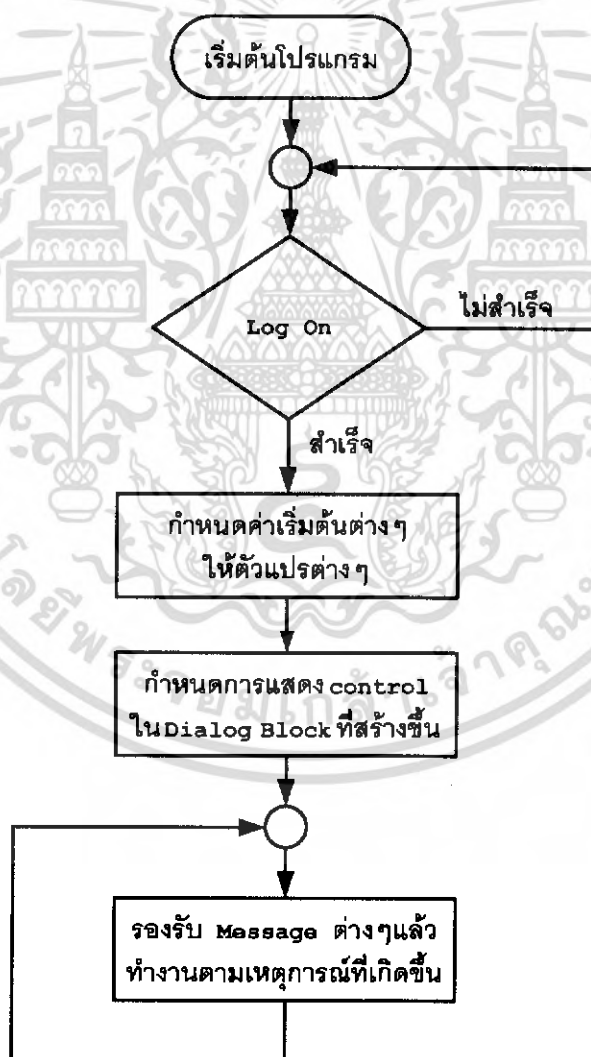
### 3.6 การออกแบบโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า

โปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า เป็นโปรแกรมที่ใช้ในการจัดการเกี่ยวกับระบบฐานข้อมูลของผู้ใช้บริการ โดยจะรับรหัสบบบัตรโดยสารเพื่อนำข้อมูลรหัสบัตรนั้นมาจัดการลงในฐานข้อมูลที่สร้างขึ้นมารองรับทั้งผู้โดยสารแบบเติมเงินและแบบรายเดือน

โปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้านี้ จะเชื่อมต่อกับชุดอ่านบัตร RFID ที่พอร์ต RS-232 ซึ่งจะประจำอยู่ที่ห้องจำหน่ายตั๋วโดยสาร

#### 3.6.1 ผังการทำงานของโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า

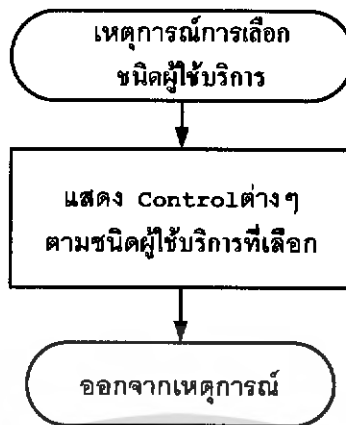
ผังการทำงานของโปรแกรมหลัก เริ่มแรกเมื่อเราเปิดโปรแกรมขึ้นมา เราจะต้องทำการ Log On เข้าสู่ระบบให้ตรงกับสถานีที่เครื่องนั้นประจำอยู่ เมื่อการ Log On สำเร็จแล้ว จะเข้าโปรแกรมหลักจะทำงานไปพร้อมกับการรอรับ Message ต่างๆแล้วทำงานตามเหตุการณ์ที่เกิดขึ้น โดยจะรอรับ Message จนพบเหตุการณ์การกดปุ่มออกจากโปรแกรมจึงยกเลิกการทำงาน



รูปที่ 3.60 ผังการทำงานของโปรแกรมหลัก

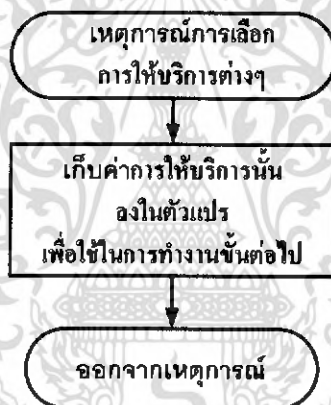
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6.2 ผังการทำงานของเหตุการณ์การเลือกชนิดผู้ใช้บริการ



รูปที่ 3.61 ผังการทำงานของเหตุการณ์การเลือกชนิดผู้ใช้บริการ

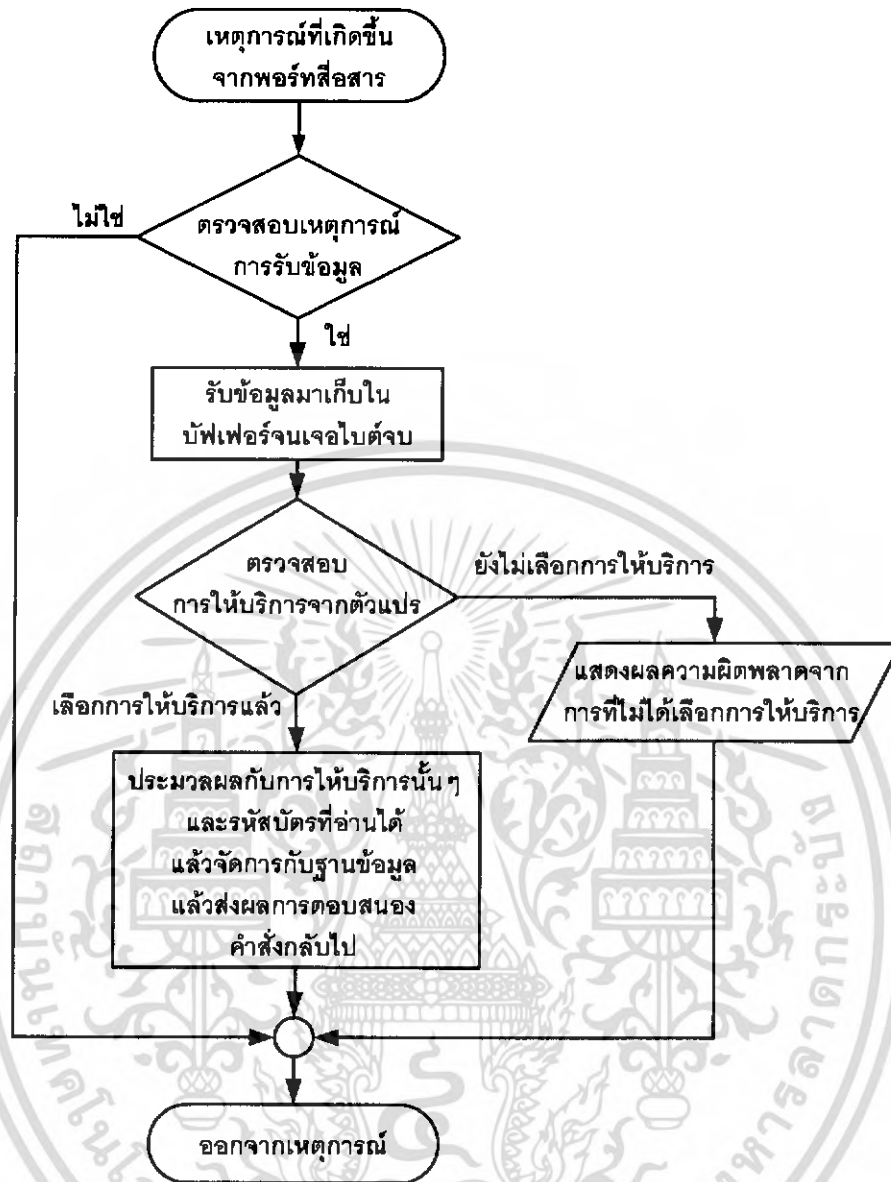
### 3.6.3 ผังการทำงานของเหตุการณ์การเลือกการให้บริการต่างๆ



รูปที่ 3.62 ผังการทำงานของเหตุการณ์การเลือกการให้บริการต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6.4 ผังการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ตสื่อสาร



รูปที่ 3.63 ผังการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ตสื่อสาร

ส่วนผังการทำงานของเหตุการณ์ต่างๆ เช่น การกดปุ่มออกจากโปรแกรม, การกดปุ่มเปิดพอร์ต และการกดปุ่มปิดพอร์ต จะคล้ายกับในหัวข้อ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

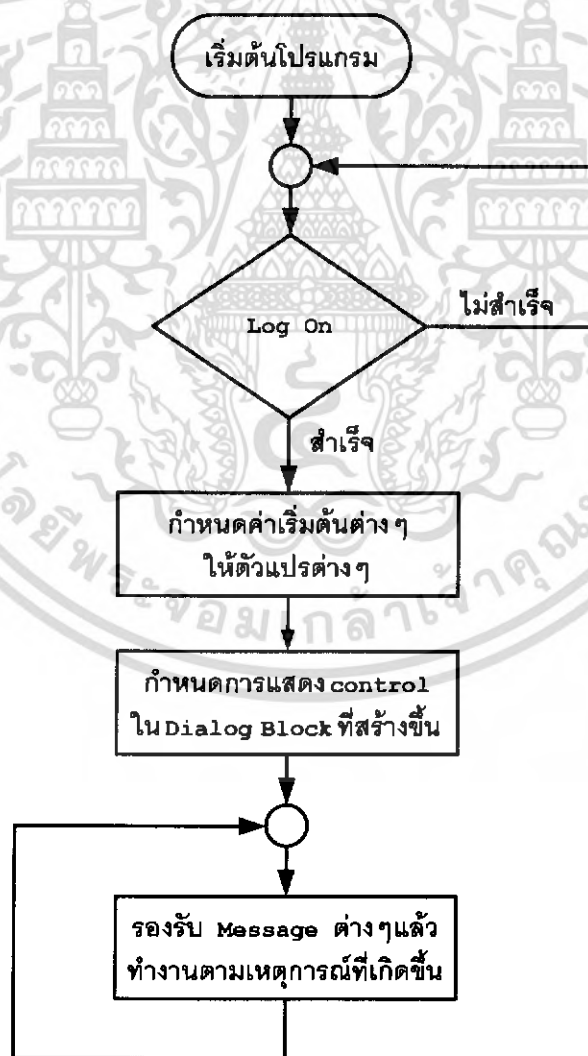
### 3.7 การออกแบบโปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก

โปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก เป็นโปรแกรมที่ใช้ในการควบคุมการเปิด-ปิดประตูทางเข้าออกของสถานีรถไฟฟ้าต่างๆ

โปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก จะเชื่อมต่อกับชุดควบคุมการเปิด-ปิดประตูทางเข้าออกเป็นระบบบัสที่มีมาตรฐาน RS-485 โดยผ่านชุดการแปลงพอร์ท RS-232 เป็น RS-485 เพื่อแปลงให้สามารถรองรับเป็นระบบบัสได้ ทำให้สามารถเชื่อมต่อกับตัวลูกได้หลายตัว โปรแกรมจะแสดงผลจำนวนเงินทั้งทางเข้าและทางออกจากสถานี โดยผ่านการประมวลผลที่คอมพิวเตอร์เพื่อจัดการกับระบบฐานข้อมูล

#### 3.7.1 ผังการทำงานของโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า

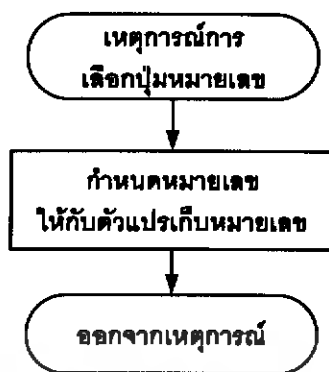
ผังการทำงานของโปรแกรมหลัก เริ่มแรกเมื่อเราเปิดโปรแกรมขึ้นมา เราจะต้องทำการ Log On เข้าสู่ระบบให้ตรงกับสถานีที่เครื่องนั้นประจำอยู่ เมื่อการ Log On สำเร็จแล้ว จะเข้าโปรแกรมหลักจะทำงานไปพร้อมกับการรอรับ Message ต่างๆแล้วทำงานตามเหตุการณ์ที่เกิดขึ้น โดยจะรอรับ Message จนพบเหตุการณ์การกดปุ่มออกจากโปรแกรมจึงยกเลิกการทำงาน



รูปที่ 3.64 ผังการทำงานของโปรแกรมหลัก

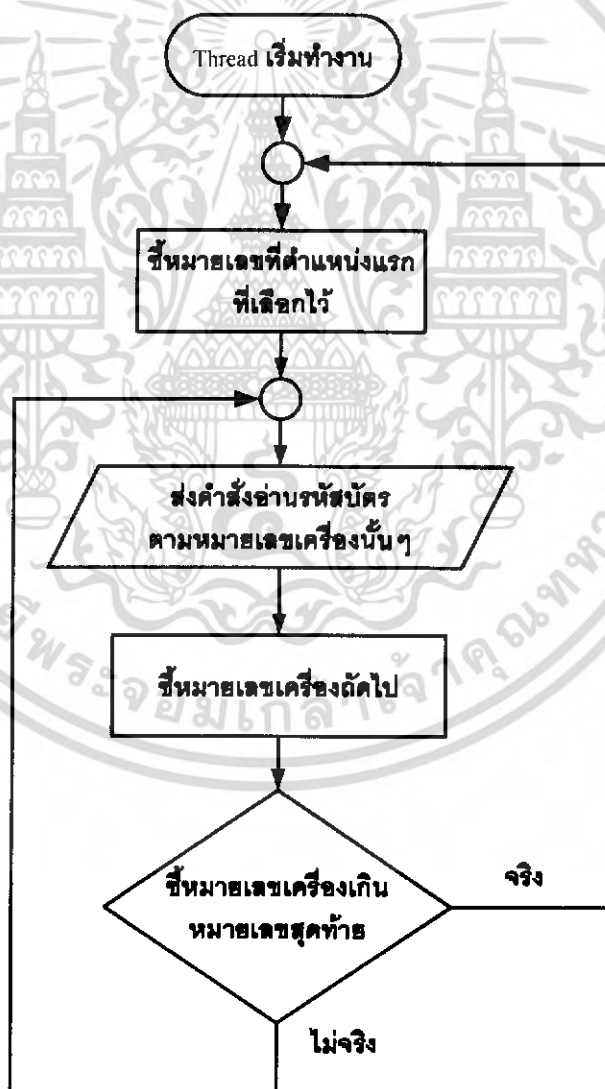
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.7.2 ฟังก์ชันการทำงานของเหตุการณ์เลือกปุ่มหมายเลขของเครื่องที่จะใช้งานบนบัตร



รูปที่ 3.65 ฟังก์ชันการทำงานของเหตุการณ์เลือกปุ่มหมายเลขของเครื่องที่จะใช้งานบนบัตร

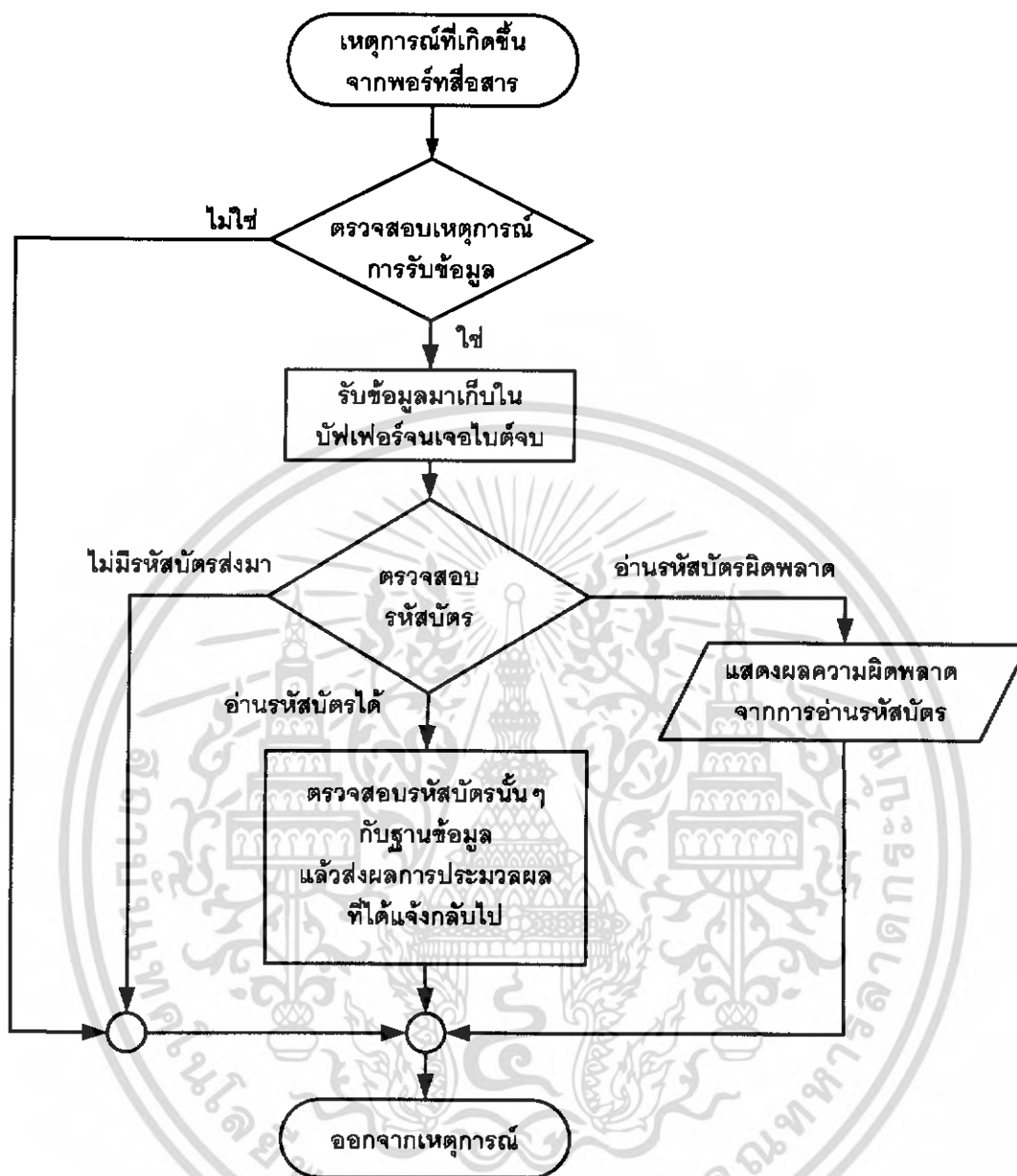
### 3.7.3 ฟังก์ชันการทำงานของ Thread ที่ใช้คำสั่งอ่านรหัสบัตรตลอดเวลา



รูปที่ 3.66 ฟังก์ชันการทำงานของ Thread ที่ใช้คำสั่งอ่านรหัสบัตรตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.7.4 ฟังก์ชันการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ทสื่อสาร



รูปที่ 3.67 ฟังก์ชันการทำงานของเหตุการณ์การที่เกิดขึ้นจากพอร์ทสื่อสาร

ส่วนฟังก์ชันการทำงานของเหตุการณ์ต่างๆ เช่น การกดปุ่มออกจากโปรแกรม, การกดปุ่มเปิดพอร์ท และการกดปุ่มปิดพอร์ท จะคล้ายกับในหัวข้อ 3.5 แต่จะมี Thread เพิ่มเข้ามา เพื่อใช้ในการส่งคำสั่งไปอ่านรหัสบัตร โดยสารจากชุดควบคุมการเปิด-ปิดประตู ตลอดเวลา เมื่อทำการกดปุ่มเปิดพอร์ทแล้วจะทำการเปิดใช้งาน Thread และเมื่อทำการกดปุ่มปิดพอร์ทก็จะทำการปิดใช้งาน Thread ด้วย

## บทที่ 4

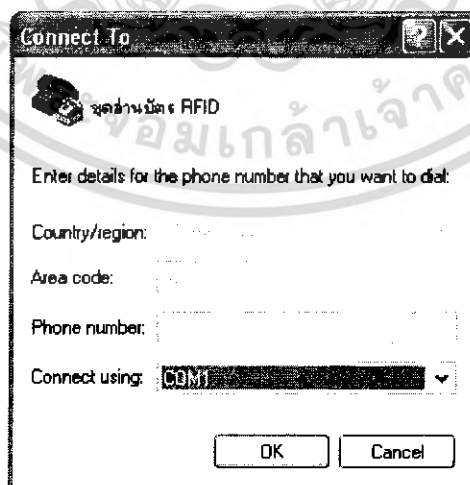
### การทดลองและผลการทดลอง

#### 4.1 การทดลองชุดอ่านบัตร RFID

- 1) เปิดโปรแกรม HyperTerminal
- 2) ตั้งชื่อในการเชื่อมต่อดังรูป 4.1
- 3) เลือกพอร์ทในการติดต่อสื่อสารดังรูป 4.2
- 4) เลือกความเร็วในการติดต่อสื่อสารดังรูป 4.3
- 5) เมื่อเข้าโปรแกรมแล้ว ทำการทาบบัตร RFID กับชุดอ่านบัตร จะเห็นข้อมูลรหัสบัตรพร้อม Flag และ LRC ที่เป็นรหัสแอสกีดังรูป 4.4
- 6) ลองทาบบัตรอีกใบจะเห็นข้อมูลรหัสบัตรที่ทาบอีกใบดังรูป 4.5

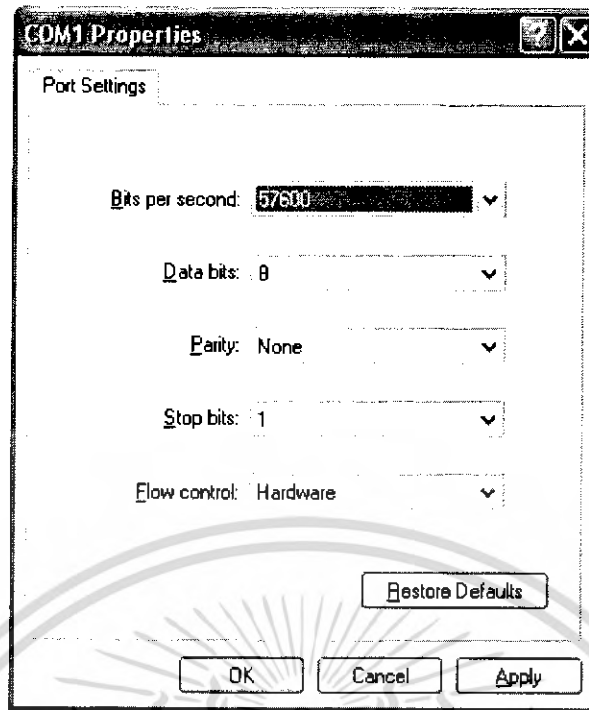


รูปที่ 4.1 แสดงการตั้งชื่อในการเชื่อมต่อ

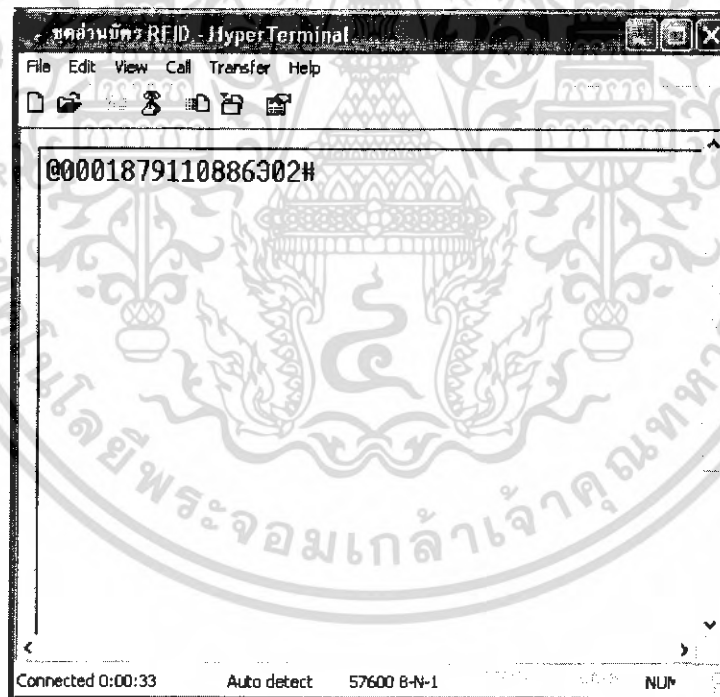


รูปที่ 4.2 แสดงการเลือกพอร์ทในการติดต่อสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

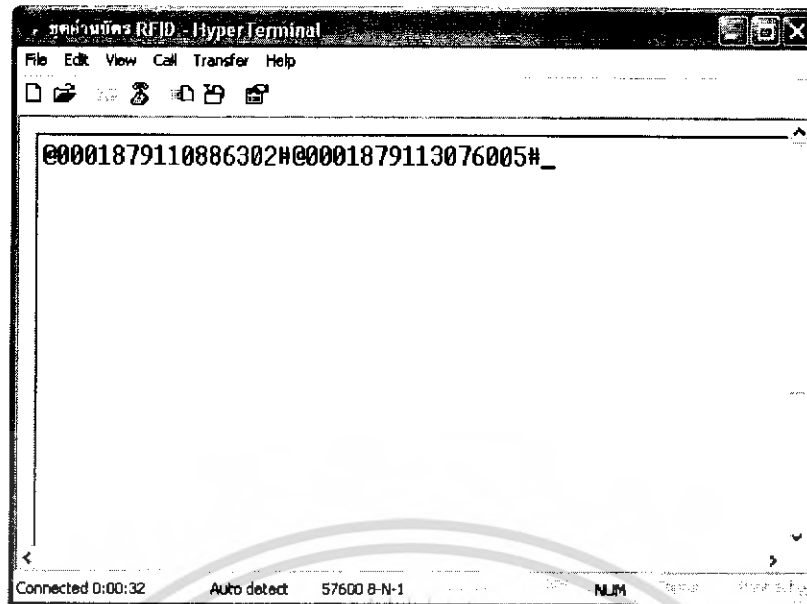


รูปที่ 4.3 แสดงการเลือกความเร็วในการติดต่อสื่อสาร



รูปที่ 4.4 แสดงรหัสบัตรพร้อม Flag และ LRC ที่เป็นรหัสแอสกีที่รับเข้ามาที่คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 4.5** แสดงรหัสบัตรที่ทานเข้ามาอีกใบ

#### 4.2 โปรแกรมกำหนดหมายเลข (Address)

โปรแกรมกำหนดหมายเลขใช้ในการกำหนดหมายเลขประจำเครื่องให้กับชุดควบคุมการเปิด-ปิดประตูทางเข้า-ทางออก โดยขั้นตอนการทดลองจะต้องต่อชุดควบคุมการเปิด-ปิดประตูทางเข้า-ทางออกกับชุดแปลง RS-232 เป็น RS-485 เพียงชุดเดียวเท่านั้น

##### ลำดับขั้นตอนการทดลอง

- 1.) ทำการเปิดโปรแกรม แสดงดังรูป 4.6
- 2.) ทำการกดปุ่มเปิดพอร์ต แสดงดังรูป 4.7
- 3.) เมื่อต้องการอ่านค่าหมายเลขประจำเครื่อง ให้กดปุ่มกดอ่านหมายเลข แสดงผลที่จอคอมพิวเตอร์เป็นหมายเลขเครื่องที่เราอ่านดังรูป 4.8 และที่ชุดควบคุมการเปิด-ปิดประตูทางเข้า-ทางออก ดังรูป 4.9
- 4.) เมื่อต้องการกำหนดหมายเลขประจำตัวเครื่องใหม่ ให้เลือกหมายเลขที่ต้องการและกดปุ่มกำหนดหมายเลขที่คอมพิวเตอร์จะแสดงผล ดังรูป 4.10 และที่ชุดควบคุมการเปิด-ปิดประตูทางเข้า-ทางออก ดังรูป 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**โปรแกรมกำหนดหมายเลข (address)**

ทางเข้า (Entrance)	ทางออก (Exit)	คำสั่ง (Command)	พอร์ตสื่อสาร
<input type="radio"/> หมายเลข 1	<input type="radio"/> หมายเลข 6	<input type="text" value="จำนวนหมายเลข"/>	COM1
<input type="radio"/> หมายเลข 2	<input type="radio"/> หมายเลข 7	<input type="text" value="กำหนดหมายเลข"/>	<input type="button" value="เปิดพอร์ต"/>
<input type="radio"/> หมายเลข 3	<input type="radio"/> หมายเลข 8	<input type="button" value="ออกจากโปรแกรม"/>	<input type="button" value="ปิดพอร์ต"/>
<input type="radio"/> หมายเลข 4	<input type="radio"/> หมายเลข 9		

\* Address ตำแหน่งหมายเลข 0 และ 5 ส่วนไว้ ไม่จำเป็นต้องใช้งานทั่วไป  
หมายเลขทางเข้าจะอยู่ระหว่าง 1 - 4 และทางออกจะอยู่ระหว่าง 6 - 9

รูปที่ 4.6 แสดงโปรแกรมที่เปิดขึ้นมา

**โปรแกรมกำหนดหมายเลข (address)**

ทางเข้า (Entrance)	ทางออก (Exit)	คำสั่ง (Command)	พอร์ตสื่อสาร
<input type="radio"/> หมายเลข 1	<input type="radio"/> หมายเลข 6	<input type="button" value="จำนวนหมายเลข"/>	<input type="text" value="COM1"/>
<input type="radio"/> หมายเลข 2	<input type="radio"/> หมายเลข 7	<input type="button" value="กำหนดหมายเลข"/>	<input type="button" value="เปิดพอร์ต"/>
<input type="radio"/> หมายเลข 3	<input type="radio"/> หมายเลข 8	<input type="button" value="ออกจากโปรแกรม"/>	<input type="button" value="ปิดพอร์ต"/>
<input type="radio"/> หมายเลข 4	<input type="radio"/> หมายเลข 9		

\* Address ตำแหน่งหมายเลข 0 และ 5 ส่วนไว้ ไม่จำเป็นต้องใช้งานทั่วไป  
หมายเลขทางเข้าจะอยู่ระหว่าง 1 - 4 และทางออกจะอยู่ระหว่าง 6 - 9

รูปที่ 4.7 แสดงการเปิดพอร์ต

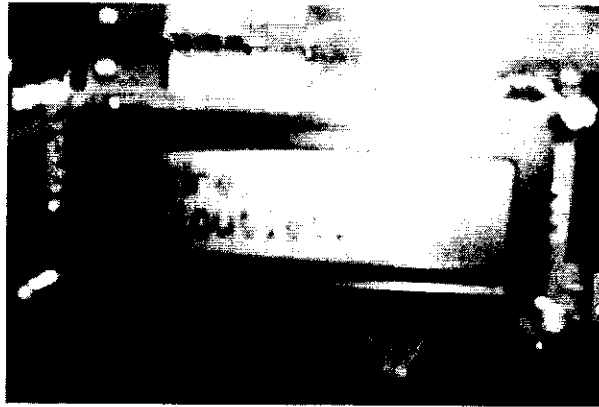
**โปรแกรมกำหนดหมายเลข (address)**

ทางเข้า (Entrance)	ทางออก (Exit)	คำสั่ง (Command)	พอร์ตสื่อสาร
<input type="radio"/> หมายเลข 1	<input type="radio"/> หมายเลข 6	<input type="button" value="จำนวนหมายเลข"/>	<input type="text" value="COM1"/>
<input type="radio"/> หมายเลข 2	<input checked="" type="radio"/> หมายเลข 7	<input type="button" value="กำหนดหมายเลข"/>	<input type="button" value="เปิดพอร์ต"/>
<input type="radio"/> หมายเลข 3	<input type="radio"/> หมายเลข 8	<input type="button" value="ออกจากโปรแกรม"/>	<input type="button" value="ปิดพอร์ต"/>
<input type="radio"/> หมายเลข 4	<input type="radio"/> หมายเลข 9		

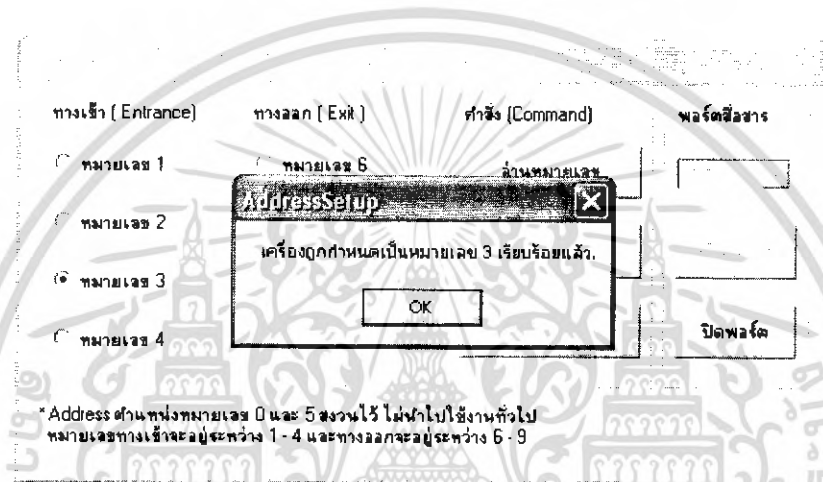
\* Address ตำแหน่งหมายเลข 0 และ 5 ส่วนไว้ ไม่จำเป็นต้องใช้งานทั่วไป  
หมายเลขทางเข้าจะอยู่ระหว่าง 1 - 4 และทางออกจะอยู่ระหว่าง 6 - 9

รูปที่ 4.8 แสดงการอ่านหมายเลขที่อ่านได้บนจอกอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 4.9** แสดงการอ่านหมายเลขที่อ่านได้บนจอ LCD ของชุดควบคุมการเปิด-ปิดประตูทางเข้า-ทางออก



**รูปที่ 4.10** แสดงการกำหนดหมายเลขที่กำหนดถูกต้องบนจอคอมพิวเตอร์

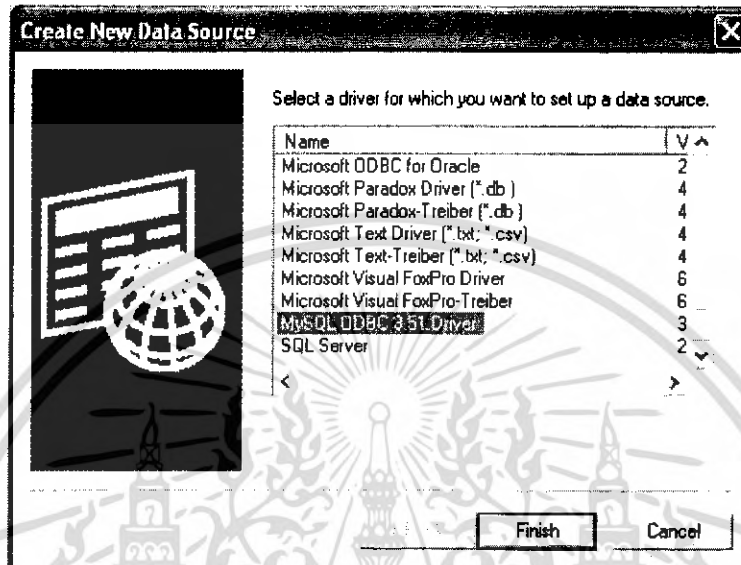


**รูปที่ 4.11** แสดงการกำหนดหมายเลขที่กำหนดถูกต้องบนจอ LCD ของชุดควบคุมการเปิด-ปิดประตูทางเข้า-ทางออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

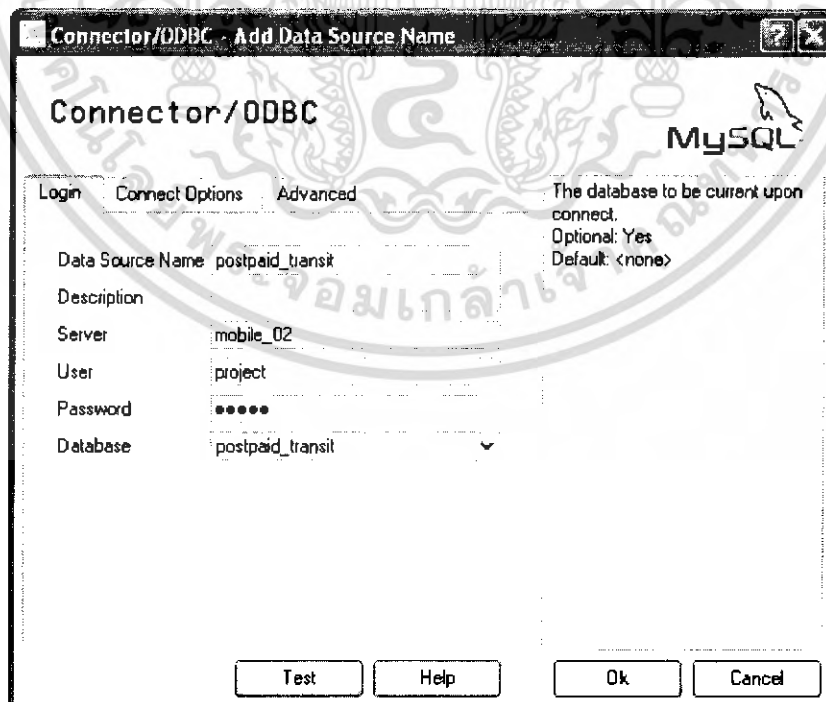
### 4.3 การตั้งค่าให้กับ ODBC ของ MySQL

- 1) ไปที่ Control Panel
- 2) ไปที่ Administrative Tools
- 3) คับเบิ้ลคลิกที่ Data Sources (ODBC)
- 4) เลือก MySQL ODBC 3.51 Driver ดังรูป 4.12



รูปที่ 4.12 การเลือก Data Source ให้กับฐานข้อมูล

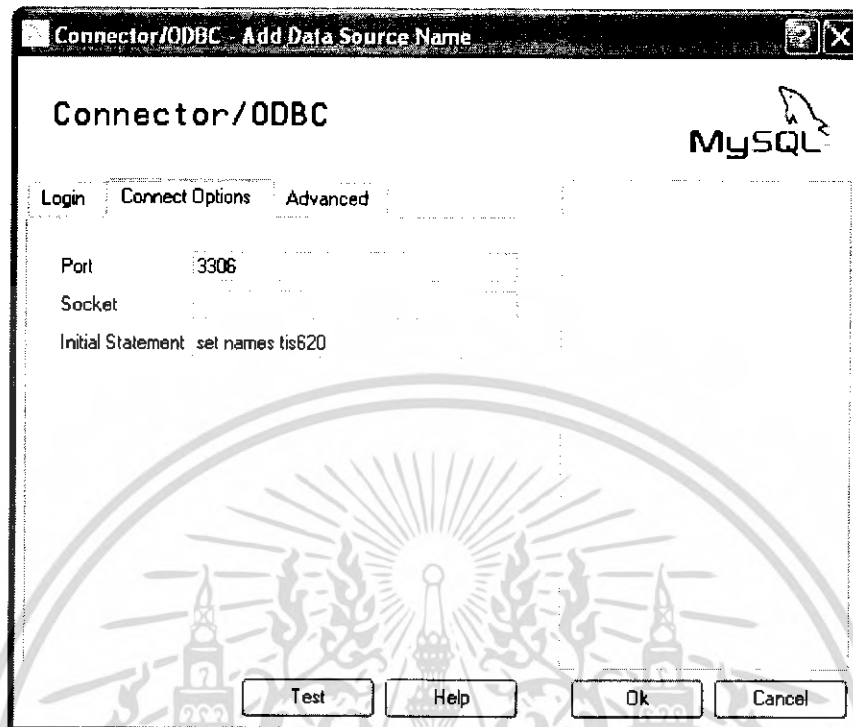
- 5) พิมพ์ค่าต่างๆ ลงไป ดังรูป 4.13



รูปที่ 4.13 การเลือกฐานข้อมูลที่ใช้ในระบบรถไฟฟ้า

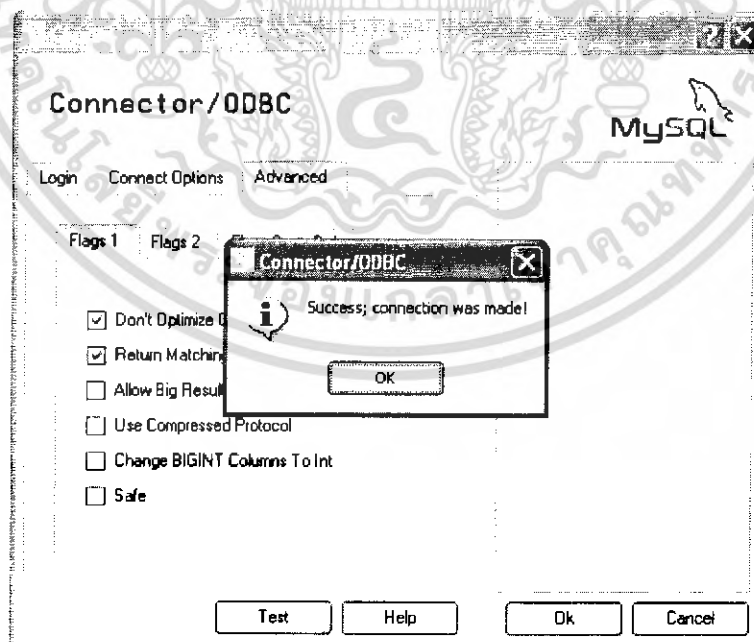
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 6) พิมพ์ค่า Port ที่ใช้ในการติดต่อสื่อสารกับ MySQL และค่า Initial Statement เพื่อให้สามารถใช้งานอักษรภาษาไทยได้ ดังรูป 4.14



รูปที่ 4.14 ตั้งค่าให้สามารถใช้งานภาษาไทยได้

- 7) ทำการทดสอบ โดยคลิกที่ปุ่ม Test ดังรูป 4.15 เมื่อผ่านแล้วให้คลิก Ok



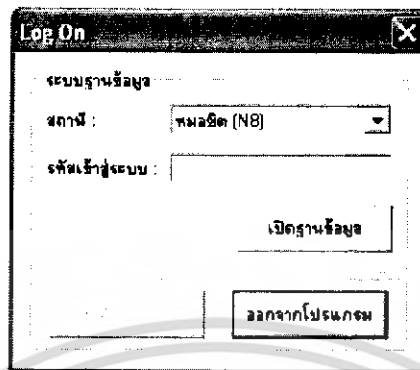
รูปที่ 4.15 แสดงผลการทดสอบฐานข้อมูลผ่าน ODBC

- 8) ทำแบบนี้กับทุกๆ ฐานข้อมูลที่ใช้ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 การใช้งานโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า

- 1) ทำการเชื่อมต่อชุดอ่านบัตร RFID กับคอมพิวเตอร์
- 2) เปิดโปรแกรมมาดังรูป 4.16



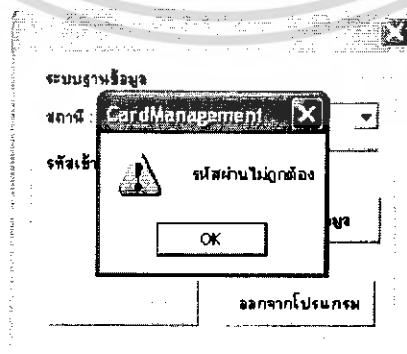
รูปที่ 4.16 หน้าต่าง Log On เข้าใช้โปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า

- 3) เลือกสถานีที่ติดตั้งเครื่องและป้อนรหัสผ่าน แล้วทำการคลิกที่ปุ่ม เปิดฐานข้อมูล ถ้ารหัสถูกต้อง จะแสดงดังรูป 4.17 แล้วคลิกปุ่ม เข้าสู่โปรแกรม



รูปที่ 4.17 หน้าต่าง Log On เมื่อป้อนรหัสผ่านถูกต้อง

ถ้ารหัสผ่านไม่ถูกต้องแสดงดังรูป 4.18



รูปที่ 4.18 หน้าต่าง Log On เมื่อป้อนรหัสผ่านไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) ทำการเปิดพอร์ต โดยคลิกที่ปุ่ม เปิดพอร์ต และเลือกชนิดของผู้ใช้บริการ แล้วคลิกเลือกวิธีการให้บริการดังรูป 4.19

รูปที่ 4.19 แสดงการใช้งานโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า

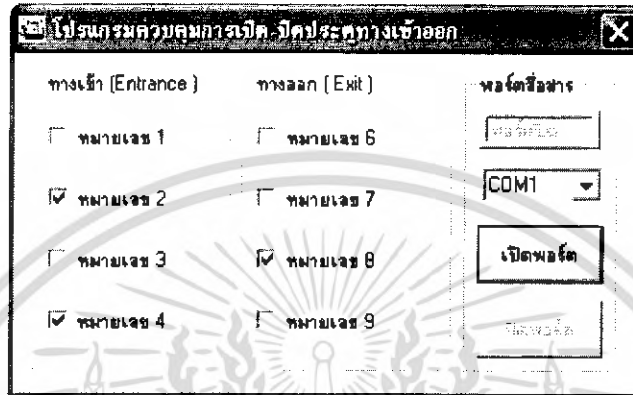
- 5) ทำการทาบบัตรลงบนชุดอ่านบัตร RFID โปรแกรมก็จะทำตามการให้บริการที่ได้เลือกไว้

ผลการทดลองของการให้บริการต่างๆ โปรแกรมสามารถทำตามได้ทุกๆ การให้บริการ โดยดูผลจากโปรแกรม SQLyog Enterprise

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

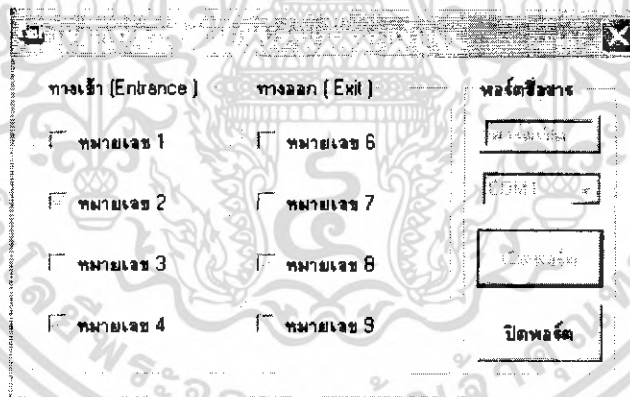
#### 4.5 การใช้งานโปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก

- 1) ทำการเชื่อมต่อชุดการแปลงพอร์ท RS-232 เป็น RS-485 กับคอมพิวเตอร์ที่พอร์ท RS-232 และทำการเชื่อมต่อชุดควบคุมการเปิด-ปิดประตูทางเข้าออก กับชุดการแปลงพอร์ท RS-232 เป็น RS-485
- 2) เปิดโปรแกรมจะปรากฏหน้าต่าง Log On เช่นเดียวกับหัวข้อ 4.4 ทำการเปิดฐานข้อมูล
- 3) ทำการเลือกหมายเลขเครื่องที่ทำการศึกษาติดตั้งทั้งหมด ดังรูป 4.20



รูปที่ 4.20 แสดงการเลือกหมายเลขเครื่องของโปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก

- 4) ทำการเปิดพอร์ต ดังรูป 4.21 จะเห็นได้ว่าไม่สามารถเลือกหมายเลขเครื่องได้



รูปที่ 4.21 การเปิดพอร์ตของโปรแกรมควบคุมการเปิด-ปิดประตูทางเข้าออก

- 5) ทำการทาบับตรลงบนชุดควบคุมการเปิด-ปิดประตูทางเข้าออก ถ้ามีข้อมูลในฐานข้อมูลคอมพิวเตอร์ก็จะประมวลผลแล้วส่งผลการตอบสนองกลับไป เนื่องจากระบบการเชื่อมต่อเป็นระบบบัสจึงจัดลำดับการให้บริการเรียงหมายเลขไป ซึ่งผลการทดสอบสามารถทำงานพร้อมกับหลายๆเครื่องได้ เราไม่สามารถสั่งเกตุลำดับการทำงานนั้นได้ทันที

ผลการทดลองของการเข้า-ออกสถานี จากการทดสอบสามารถทำงานได้ถูกต้อง โดยดูผลจากโปรแกรม SQLyog Enterprise

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5 สรุปผลและวิจารณ์

ระบบควบคุมการใช้บริการรถไฟฟ้าโดยใช้การ์ด RFID มีการทำงานของระบบโดยรวม เริ่มจากการรับค่าจากบัตรRFIDโดยตัวอ่านจะนำค่าที่ได้ผ่านไมโครคอนโทรลเลอร์ส่งไปตรวจสอบกับฐานข้อมูลในคอมพิวเตอร์โดยผ่านพอร์ต RS 485 ที่ทำการแปลงเป็นพอร์ต RS232 เพื่อต่อเข้ากับคอมพิวเตอร์ ถ้ามีข้อมูลจะสั่งให้ไมโครคอนโทรลเลอร์แสดงผลเพื่อแทนการเปิดประตูด้วยเสียง LED และการสับสวิทช์รีเลย์ ฐานข้อมูลจะทำการเก็บข้อมูลสถานีต้นทางและสถานีปลายทางไว้เพื่อทำการคำนวณค่าใช้บริการ ซึ่งการทำงานของส่วนฮาร์ดแวร์ทั้งด้านเข้าและออกจากการใช้บริการจะทำงานเหมือนกันแต่จะมีโปรแกรมที่ทำงานแตกต่างกันคือส่วนการเข้าใช้จะเก็บสถานีเข้า ส่วนด้านออกจะเก็บข้อมูลสถานีออกและนำข้อมูลไปคำนวณค่าใช้บริการ โดยได้ทำการแบ่งเป็นการใช้บริการแบบรายเดือนและแบบเติมเงิน

ในการทำโครงการนี้ เราใช้ไมโครคอนโทรลเลอร์ในการควบคุมการทำงาน และใช้โปรแกรมที่ทำการควบคุมที่แตกต่างกันในการควบคุมการทำงานของระบบแบ่งเป็นจุดเข้าหรือจุดออก โดยทำการเชื่อมต่อกับฐานข้อมูลทำการเก็บข้อมูลทั้งหมด

## บรรณานุกรม

- บุรุษนา สีสาส์วัฒนกุล และอมรพันธุ์ คำอรธร. **การสร้างระบบฐานข้อมูลด้วย PL/SQL&Oracle Developer**. กรุงเทพฯ : ด้านสหราชอาณาจักรพิมพ์.มิถุนายน.2545
- สาริต ชัยวิวัฒน์ตระกูล. **เทคนิค MySQL ให้เต็มประสิทธิภาพ**. กรุงเทพฯ : HN Group.2547
- กิตติภูมิ วรพัทธ์. **MySQL ตามครอบครัวอัครवाद**. กรุงเทพฯ : HN Group.2544
- บุรุษนา สีสาส์วัฒนกุล. **คู่มือการเขียนโปรแกรม Microsoft Advance Visual C++ Version 6.0**. กรุงเทพฯ : ชักเซสมิเคีย
- บุรุษนา สีสาส์วัฒนกุล. **คู่มือการเขียนโปรแกรมและใช้งาน Visual C++ Version 6.0**. กรุงเทพฯ : อินโฟเพรส กรกฎาคม.2544
- เทคโนโลยี RFID นวัตกรรมแห่งการเพิ่มผลผลิต**. Omron Electronics
- “RFID for Smart Logistics”, Logistics Asia 2004 Conference, Identify Limited, Bangkok, Thailand, 2004
- “RFID Technology of OMRON”. [Online].Available : <http://www.omron-ap.co.th>
- “RFID Frequency Option for Passive Chip Tags”. [Online].Available : <http://www.IDtechex.com>
- “RFID Present”. [Online].Available : <http://www.nectec.com>
- “RFID :Radio Frequency Identification”. [Online].Available : <http://www.rfidjournal.com>
- “GP-8 Reader”. [Online].Available : <http://www.promag.com>
- “RS485 Connection”. [Online].Available : <http://www.micro4dev.com>
- “การสื่อสารแบบ RS422/RS485”. [Online].Available : <http://www.electoday.com>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <16F628.h>

#FUSES WDT           //Watch Dog Timer
#FUSES HS            //High speed Osc (> 4mhz)
#FUSES NOPUT        //No Power Up Timer
#FUSES NOPROTECT    //Code not protected from reading
#FUSES NOBROWNOUT   //No brownout reset
#FUSES MCLR         //Master Clear pin enabled
#FUSES NOLVP        //No low voltage prgming, B3(PIC16) or
B5(PIC18) used for I/O
#FUSES NOCPD        //No EE protection

```

```

#use delay(clock=11059200,RESTART_WDT)
#use rs232(baud=57600,parity=N,xmit=PIN_B2,rcv=PIN_B1,bits=8)

```

```

#define byte_start '@'
#define byte_stop '#'
#define ACK 'A'

```

```

#byte PORTA=0x05
#byte TRISA=0x85
#byte PORTB=0x06
#byte TRISB=0x86

```

```

#bit CARD_CLOCK = PORTB.3
#bit CARD_DATA = PORTB.4
#bit CARD_PRESENT = PORTB.0
#bit BUZZER = PORTA.1
#bit LED = PORTA.0

```

```

char CARD_ID[18];
unsigned int id_count, Next_Tx, Rx_data;
unsigned int time;
int1 ID_status, beep_enable, time_out;

```

```

void initial(void);
void Gen_LRC(void);
int1 Chk_LRC(void);
int1 Chk_parity(unsigned int p_data);

```

```

#int_EXT
void Read_Card()
{
    unsigned int data = 0, bit_count = 5;
    int1 data_temp, bit_start = 1;

```

```

    if(!ID_status)
    {
        id_count = 0;
        do
        {
            while(CARD_CLOCK);
            delay_us(2);
            data_temp = CARD_DATA;
            if(bit_start)
                if(!data_temp) bit_start = 0;
            if(!bit_start)
            {
                data >>= 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(data_temp) data |= 0x80;
        bit_count--;
        if(bit_count == 0)
        {
            data >>= 3;
            data = ~data;
            CARD_ID[id_count] = data & 0x1F;
            id_count++;
            bit_count = 5;
            data = 0;
        }
    }
    while(!CARD_CLOCK);
}while(id_count < 17);

if(Chk_LRC())
{
    ID_status = 0;
    beep_enable = 1;
    time = 0;
}
else ID_status = !bit_start;
if(ID_status)
{
    CARD_ID[0] = byte_start;
    CARD_ID[17] = byte_stop;
    Gen_LRC();
    enable_interrupts(INT_TBE);
}
}
}

#INT_TBE
void Tx(void)
{
    putc(CARD_ID[Next_Tx]);
    Next_Tx++;
    if(Next_Tx > 17)
    {
        disable_interrupts(INT_TBE);
        Next_Tx = 0;
        time_out = 0;
        time = 0;
    }
}

#INT_RDA
void Rx(void)
{
    Rx_data = getc();

    if(ID_status)
    {
        if(Rx_data == ACK)
        {
            BUZZER = 0;
            LED = 0;
        }
        else
            ID_status = 0;
        beep_enable = 1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        time = 0;
    }
}

#INT_TIMER1
void timer_100ms(void)
{
    time++;
    set_timer1(0x7900);
}

void main()
{
    if ( restart_cause() == NORMAL_POWER_UP)
        reset_cpu();
    delay_ms(100);
    initial();

    while(TRUE)
    {
        restart_wdt();
        if(time == 30)
        {
            if(ID_status && !beep_enable)
            {
                ID_status = 0;
                beep_enable = 1;
                time_out = 1;
                time = 0;
            }
        }
        if(beep_enable)
        {
            if(ID_status)
            {
                if(time > 10)
                {
                    ID_status = 0;
                    beep_enable = 0;
                    LED = 1;
                }
                else if(time > 5) BUZZER = 1;
            }
            else
            {
                if(time % 2)
                    BUZZER = 0;
                else
                {
                    BUZZER = 1;
                    if(time_out)
                    {
                        if(time == 6)
                        {
                            time_out = 0;
                            beep_enable = 0;
                        }
                    }
                    else if(time == 4)
                        beep_enable = 0;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
  }
}

void initial(void)
{
  TRISA = 0x00;
  TRISB = 0xFF;
  BUZZER = 1;
  LED = 1;
  id_count = 0;
  Next_Tx = 0;
  time = 0;
  ID_status = 0;
  beep_enable = 0;
  time_out = 0;
  setup_timer_1( T1_INTERNAL | T1_DIV_BY_8 );
  set_timer1(0x7900);
  ext_int_edge(H_TO_L);
  enable_interrupts(INT_EXT);
  disable_interrupts(INT_TBE);
  enable_interrupts(INT_RDA);
  enable_interrupts(INT_TIMER1);
  enable_interrupts(GLOBAL);
  setup_wdt(WDT_72MS);
}

int1 Chk_LRC(void)
{
  unsigned int sum = 0;

  for(id_count = 0; id_count < 17; id_count++)
  {
    if(Chk_parity(CARD_ID[id_count]))
      return 1;
    CARD_ID[id_count] &= 0x0F;
    sum ^= CARD_ID[id_count];
    if(CARD_ID[id_count] < 10)
      CARD_ID[id_count] += '0';
    else
      CARD_ID[id_count] += 'A' - 10;
  }
  if(sum != 0) return 1;
  if(CARD_ID[0] != 'B' || CARD_ID[15] != 'F') return 1;
  return 0;
}

int1 Chk_parity(unsigned int p_data)
{
  unsigned int bit_count, n_bit = 0;
  int1 p_flag;

  if(p_data == 0) return 1;
  for(bit_count = 0; bit_count < 5; bit_count++)
  {
    p_flag = p_data & 0x01;
    p_data >>= 1;
    if(p_flag) n_bit++;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if((n_bit % 2) == 0) return 1;
    return 0;
}

void Gen_LRC(void)
{
    unsigned char lrc = 0;

    for(id_count = 1; id_count < 15; id_count++)
        lrc ^= CARD_ID[id_count];

    CARD_ID[15] = lrc >> 4;
    CARD_ID[16] = lrc & 0x0f;

    if(CARD_ID[15] < 10)
        CARD_ID[15] = CARD_ID[15] + '0';
    else
        CARD_ID[15] = CARD_ID[15] + 'A' - 10;
    if(CARD_ID[16] < 10)
        CARD_ID[16] = CARD_ID[16] + '0';
    else
        CARD_ID[16] = CARD_ID[16] + 'A' - 10;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**โปรแกรมชุดควบคุมการเปิด-ปิดประตูทางเข้า-ออกสถาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <16F873A.h>
#include <delay(clock=11059200,RESTART_WDT)>
#include <rs232(baud=57600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,enable=PIN_C5,ERRORS)>

#include "lcd2x16.c"

#define FUSES_WDT //Watch Dog Timer
#define FUSES_HS //High speed Osc (> 4mhz)
#define FUSES_NOPUT //No Power Up Timer
#define FUSES_NOPROTECT //Code not protected from reading
#define FUSES_NODEBUG //No Debug mode for ICD
#define FUSES_BROWNOUT //Reset when brownout detected
#define FUSES_NOLVP //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#define FUSES_NOCPD //No EE protection
#define FUSES_NOWRT //Program memory not write protected

#define version "1.0"
#define address_mem 0x20
#define max_tx 15
#define max_rx 8
#define byte_start '@'
#define byte_stop '#'
// define command
#define R_Add '1'
#define W_Add '2'
#define R_ID '3'
#define W_md '4'
#define W_mm '5'
#define LRC_e '6'
#define DB_e '7'

#define prepaid 0
#define postpaid 1

#define PORTA = 0x05
#define TRISA = 0x85
#define PORTB = 0x06
#define TRISB = 0x86
#define PORTC = 0x07
#define TRISC = 0x87

#define CARD_CLOCK = PORTC.0
#define CARD_DATA = PORTC.1
#define CARD_PRESENT = PORTB.0
#define LED = PORTA.0
#define RELAY = PORTA.1
#define BUZZER = PORTA.2
void initial(void);
void clear_buffer(void);
unsigned char read_memory(int address);
void write_memory(int address, unsigned char value);
void show_address(void);
void set_address(void);
void show_dberror(void);
void show_welcome(void);
void show_thank(void);
void show_money(int type);
void show_on(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void show_error(unsigned err);
int1 chk_lrc(void);
void gen_lrc(void);
void sent_packet(void);
int1 chk_parity(unsigned int p_data);
void get_money(void);
void short_beep(void);

unsigned char lrc_high, lrc_low;
unsigned char money[5];
unsigned char CARD_ID[17];
unsigned char data_packet[max_tx];
unsigned char receive_buffer[max_rx];
unsigned char Slave_address;
unsigned char Next_Rx, id_count;
unsigned char time_out;
unsigned char Rx_data;
int1 ID_status, Ent_Ext;
int1 Set_version, Active_process, start_byte;

char const welcome[] = "\f Welcome to sky\n train station.";
char const set_add[] = "\fSetting address\ncomplete.";
char const ver_str[] = "Version ";
char const start_str[] = "\fStarting >>> ";
char const thank[] = "Thank you.";
char const address_str[] = "\faddress : ";
char const direction[] = "\fndirection : ";
char const error1[] = "\fCan't config\naddress.";
char const error2[] = "\fRange of Address\nhas error.";
char const error3[] = "\fCommand not\nfound.";
char const error4[] = "\fCommunication\nerror.";
char const read_error[] = "\fReading card\nerror.";
char const db_error[] = "\fDatabase has\nproblem.";

#int_EXT
void Read_Card()
{
    unsigned char data = 0, bit_count = 5;
    int1 data_temp, bit_start = 1;

    if(!ID_status)
    {
        id_count = 0;
        do
        {
            while(CARD_CLOCK);
            delay_us(2);
            data_temp = CARD_DATA;
            if(bit_start)
                if(!data_temp) bit_start = 0;
            if(!bit_start)
            {
                data >>= 1;
                if(data_temp) data |= 0x80;
                bit_count--;
                if(bit_count == 0)
                {
                    data >>= 3;
                    data = ~data;
                    CARD_ID[id_count] = data & 0x1F;
                    id_count++;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        bit_count = 5;
        data = 0;
    }
    }
    while(!CARD_CLOCK);
}while(id_count < 17);

if(chk_lrc())
{
    ID_status = 0;
    show_error(0);
}
else ID_status = !bit_start;
if(ID_status)
    time_out = 0;
}
}

#INT_RDA
void Rx(void)
{
    Rx_data = getc();
    if(Rx_data == byte_start || Next_Rx == max_rx) {
        Next_Rx = 0;
        start_byte = 1;
    }
    if(start_byte) {
        receive_buffer[Next_Rx] = Rx_data;
        Next_Rx++;
    }
}

#INT_TIMER1
void timer_100ms(void)
{
    time_out++;
    set_timer1(0x7900);
}

void main()
{
    if(restart_cause() == NORMAL_POWER_UP)
        reset_cpu();
    initial();
    while(TRUE)
    {
        restart_wdt();
        if(!Active_process) {
            if(Rx_data == byte_stop && receive_buffer[0] == byte_start) {
                if(receive_buffer[1] == Slave_address || receive_buffer[1]
== '0')
                    Active_process = 1;
                start_byte = 0;
            }
        }
        if(Active_process) {
            switch(receive_buffer[2]) {
                case R_Add : if(receive_buffer[1] == '0') {
                    printf("&ADD>%c$",Slave_address);
                    short_beep(); delay_ms(500);
                    show_address();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    } break;
    case W_Add :   if(receive_buffer[1] == '0') {set_address();
} break;
    case R_ID :   if(ID_status) {
                    gen_lrc();
                    sent_packet();
                    ID_status = 0;
                }
                else printf("&%c,ID>NO$",Slave_address);
                break;
    case W_md :   show_money(prepaid);    break;
    case W_mm :   show_money(postpaid);   break;
    case DB_e :   show_dberror();        break;
    case LRC_e :   show_error(4);        break;
    default :     show_error(3);         break;
}
clear_buffer();
Active_process = 0;
}
if(ID_status && time_out == 20) {
    ID_status = 0;
    Active_process = 0;
    show_error(4);
}
}
}

```

```

void initial(void)
{
    unsigned char Count;
    TRISA = 0b00000000;
    TRISB = 0b00000001;
    TRISC = 0b11000011;

    setup_timer_1( T1_INTERNAL | T1_DIV_BY_8 );
    ext_int_edge(H_TO_L);
    set_timer1(0x7900);
    enable_interrupts(INT_EXT);
    enable_interrupts(INT_RDA);
    enable_interrupts(INT_TIMER1);
    enable_interrupts(GLOBAL);

    LED = 1;
    RELAY = 1;
    BUZZER = 1;
    Next_Rx = 0;
    id_count = 0;
    ID_status = 0;
    Set_version = 0;
    start_byte = 0;
    Active_process = 0;
    clear_buffer();
    LCD_Init();
    LCD_GotoXY(5,1);
    LCD_PutS("SkyTrain");
    LCD_GotoXY(3,2);
    LCD_PutS(ver_str);
    LCD_PutS(version);
    Slave_address = read_memory(address_mem);
    delay_ms(500);
    if(Slave_address < '5')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Ent_Ext = 0;
else
    Ent_Ext = 1;
show_address();
delay_ms(1500);
Count = 4;
do
{
    Count--;
    printf(LCD_PutS,"%s%u", start_str,Count);
    delay_ms(1000);
} while(Count > 0);
short_beep();
show_on();
setup_wdt(WDT_1152MS);
}

void clear_buffer(void)
{
    receive_buffer[0] = 0;
    receive_buffer[1] = 0;
    receive_buffer[2] = 0;
    receive_buffer[3] = 0;
    receive_buffer[4] = 0;
    receive_buffer[5] = 0;
    receive_buffer[6] = 0;
    receive_buffer[7] = 0;
}

unsigned char read_memory(int address)
{
    return read_eeprom(address);
}

void write_memory(int address, unsigned char value)
{
    write_eeprom(address, value);
}

void show_address(void)
{
    unsigned char way[14];

    if(Ent_Ext)
        strcpy(way, "\nis Exit.");
    else
        strcpy(way, "\nis Entrance.");
    printf(LCD_PutS, "%s%c%s", address_str, Slave_address, way);
    delay_ms(800);
}

void show_dberror(void)
{
    LCD_PutS(db_error);
    short_beep();
    delay_ms(100);
    short_beep();
    delay_ms(800);
    show_on();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void set_address(void)
{
    unsigned char tmp_add;

    tmp_add = receive_buffer[3];
    if(tmp_add > '0' && tmp_add < ':') {
        if(tmp_add != '5') {
            write_memory(address_mem , tmp_add);
            if(tmp_add == read_memory(address_mem)) {
                Slave_address = tmp_add;
                LCD_PutS(set_add);
                short_beep();
                printf("&ADD>OKS");
                if(Slave_address < '5')
                    Ent_Ext = 0;
                else
                    Ent_Ext = 1;
                delay_ms(800);
                show_address();
                delay_ms(1000);
                show_on();
            }
            else show_error(1);
        }
        else show_error(2);
    }
    else show_error(2);
}

void show_welcome(void)
{
    LCD_PutS(welcome);
}

void show_thank(void)
{
    LCD_Clear();
    LCD_GotoXY(4,1);
    LCD_PutS(thank);
}

void show_money(int1 cus_type)
{
    get_money();
    if(cus_type == prepaid)
        printf(LCD_PutS, "\fYou credit : \n%s Baht", money);
    if(cus_type == postpaid)
        printf(LCD_PutS, "\fYou amount : \n%s Baht", money);
    LED = 0;
    RELAY = 0;
    BUZZER = 0;
    delay_ms(1000);
    BUZZER = 1;
    delay_ms(1000);
    LED = 1;
    RELAY = 1;
    show_on();
}

void show_on(void)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(Ent_Ext)
        show_thank();
    else
        show_welcome();
}

void show_error(unsigned err)
{
    switch(err) {
        case 0 : LCD_PutS(read_error); break;
        case 1 : LCD_PutS(error1); printf("&%c,ERR=1$",Slave_address);
break;
        case 2 : LCD_PutS(error2); printf("&%c,ERR=2$",Slave_address);
break;
        case 3 : LCD_PutS(error3); printf("&%c,ERR=3$",Slave_address);
break;
        case 4 : LCD_PutS(error4); break;
        default : break;
    }
    short_beep();
    delay_ms(100);
    short_beep();
    delay_ms(1000);
    show_on();
}

int1 chk_lrc(void)
{
    unsigned int sum = 0;

    for(id_count = 0; id_count < 17; id_count++)
    {
        if(Chk_parity(CARD_ID[id_count]))
            return 1;
        CARD_ID[id_count] &= 0x0F;
        sum ^= CARD_ID[id_count];
    }
    if(sum != 0) return 1;
    if(CARD_ID[0] != 0x0b || CARD_ID[15] != 0x0f) return 1;
    return 0;
}

int1 chk_parity(unsigned int p_data)
{
    unsigned int bit_count, n_bit = 0;
    int1 p_flag;

    if(p_data == 0) return 1;
    for(bit_count = 0; bit_count < 5; bit_count++)
    {
        p_flag = p_data & 0x01;
        p_data >>= 1;
        if(p_flag) n_bit++;
    }
    if((n_bit % 2) == 0) return 1;
    return 0;
}

void sent_packet(void)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

putc('&'); putc(Slave_address); putc(',');
for(id_count = 1; id_count < 15; id_count++)
{
    if(CARD_ID[id_count] < 10)
        data_packet[id_count] = CARD_ID[id_count] + '0';
    else
        data_packet[id_count] = CARD_ID[id_count] + 'A' - 10;
    putc(data_packet[id_count]);
}
putc(lrc_high); putc(lrc_low); putc('$');
}

```

```

void gen_lrc(void)
{
    unsigned char lrc = 0;

    for(id_count = 1; id_count < 15; id_count++)
        lrc ^= CARD_ID[id_count];

    lrc_high = lrc >> 4;
    lrc_low = lrc & 0x0f;

    if(lrc_high < 10)
        lrc_high = lrc_high + '0';
    else
        lrc_high = lrc_high + 'A' - 10;
    if(lrc_low < 10)
        lrc_low = lrc_low + '0';
    else
        lrc_low = lrc_low + 'A' - 10;
}

```

```

void get_money(void)
{
    money[0] = receive_buffer[3];
    money[1] = receive_buffer[4];
    money[2] = receive_buffer[5];
    money[3] = receive_buffer[6];
    money[4] = 0;
}

```

```

void short_beep(void)
{
    BUZZER = 0;
    delay_ms(100);
    BUZZER = 1;
}

```

```

/*
** lcd2x16.c -- LCD 2x16
**
** Version 1.00
**
**
*/

```

```

////////// config address //////////
struct LCD_Address_Pin
{
    BOOLEAN unused;
    BOOLEAN EN;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    BOOLEAN RS;
    BOOLEAN RW;
    int DATA : 4;
} LCD_Address;

#byte LCD_Address = 0x06
#byte LCD_TRIS = 0x86

//////////////////// Mode //////////////////////////////////////
#define Command 0x00
#define Ascii 0x01
////////////////////

//////////////////// Command //////////////////////////////////////
#define Clear 0x01
#define Home 0x02
#define DisplayOff 0x08
#define DisplayOn 0x0c
#define CursorOff 0x0c
#define CursorBlink 0x0d
#define CursorOn 0x0e
#define CursorLeft 0x10
#define CursorRight 0x14
////////////////////

//////////////////// Macro //////////////////////////////////////
#define LCD_Clear() LCD_Write(Command, Clear)
#define LCD_Home() LCD_Write(Command, Home)
////////////////////

BYTE LCD_Line = 1;

void LCD_Init(void);
BYTE LCD_Read(void);
void LCD_Write_Nibble(BYTE Data);
void LCD_Write(BYTE Mode, BYTE LCD_Data );
void LCD_PutS(char Str);
void LCD_GotoXY(BYTE x, BYTE y);

void LCD_Init(void)
{
    BYTE Count;

    LCD_TRIS = 0x01;
    LCD_Address.Data = 0x00;
    LCD_Address.RS = 0;
    LCD_Address.RW = 0;
    LCD_Address.EN = 0;
    delay_ms(15);
    for(Count = 1; Count <= 3; ++Count)
    {
        LCD_Write_Nibble(3);
        delay_ms(5);
    }
    LCD_Write_Nibble(2);
    LCD_Write(Command, 0x28);
    LCD_Write(Command, DisplayOn);
    LCD_Clear();
    LCD_Write(Command, 0x06);
    LCD_Home();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/*
** LCD_Read - Read one byte of data to the LCD
*/
BYTE LCD_Read(void)
{
    BYTE LCD_Data;

    LCD_TRIS = 0xf1;
    LCD_Address.RW = 1;
    delay_cycles(1);
    LCD_Address.EN = 1;
    LCD_Data = LCD_Address.Data;
    LCD_Data <<= 4;
    LCD_Address.EN = 0;
    delay_cycles(1);
    LCD_Address.EN = 1;
    delay_cycles(1);
    LCD_Data |= LCD_Address.Data;
    LCD_Address.EN = 0;
    LCD_TRIS = 0x01;
    return LCD_Data;
}

void LCD_Write_Nibble(BYTE Data)
{
    LCD_Address.DATA = Data;
    delay_cycles(1);
    LCD_Address.EN = 1;
    delay_cycles(1);
    LCD_Address.EN = 0;
}

/*
** LCD_Write - Write one byte of command or data to the LCD
*/
void LCD_Write(BYTE Mode, BYTE LCD_Data)
{
    LCD_Address.RS = 0;
    if (Mode == Command)
        while (LCD_Read() > 0x7f);
    LCD_Address.RS = Mode;
    delay_cycles(1);
    LCD_Address.RW = 0;
    delay_cycles(1);
    LCD_Address.EN = 0;
    LCD_Write_Nibble(LCD_Data >> 4);
    LCD_Write_Nibble(LCD_Data & 0x0f);
    if (Mode == Command && (LCD_Data == Clear || LCD_Data == Home))
        delay_ms(2);
    else
        delay_us(40);
}

/*
** LCD_LCD_PutS: Display a string.
*/
void LCD_PutS(char Str)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch (Str)
{
    case '\f' : LCD_Clear();
                LCD_Line = 1;                break;
    case '\n' : LCD_GotoXY(1, ++LCD_Line);    break;
    case '\b' : LCD_Write(Command, 0x10);    break;
    default   : LCD_Write(Ascii, Str);        break;
}
}

void LCD_GotoXY(BYTE x, BYTE y)
{
    BYTE Address;

    switch(y)
    {
        case 1 : Address=0x80; break;
        case 2 : Address=0xc0; break;
    }
    Address += x - 1;
    LCD_Line = y;
    LCD_Write(Command, Address);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โปรแกรมกำหนดหมายเลขประจำตัวคณาจารย์การเปิด-ปิดประตูทางเข้าออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//AddressSetupDlg.h : header file
//
//{{AFX_INCLUDES()}

//}}AFX_INCLUDES

#ifdef _AFX_
#ifndef(AFX_ADDRESSSETUPDLG_H__04BC625F_7F22_463A_B0BE_C29190192331__INCLUDED_)
#define AFX_ADDRESSSETUPDLG_H__04BC625F_7F22_463A_B0BE_C29190192331__INCLUDED_

#define comEvReceive 2

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
#include "serialcomm.h"

////////////////////////////////////
//CAddressSetupDlg dialog

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

//Dialog Data
    #define AFX_DATA(CAddressSetupDlg)
    enum { IDD = IDD_ADDRESSSETUP_DIALOG };
    CMSComm m_SerialComm;
    CString m_port;
    #undef AFX_DATA

//ClassWizard generated virtual function overrides
    #define AFX_VIRTUAL(CAddressSetupDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV
support
    #undef AFX_VIRTUAL

//Implementation
protected:
    HICON m_hIcon;
//Generated message map functions
    #define AFX_MSG(CAddressSetupDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    afx_msg void OnReadAdd();
    afx_msg void OnWriteAdd();
    afx_msg void OnAddress1();
    afx_msg void OnAddress2();
    afx_msg void OnAddress3();
    afx_msg void OnAddress4();
    afx_msg void OnAddress6();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

afx_msg void OnAddress7();
afx_msg void OnAddress8();
afx_msg void OnAddress9();
afx_msg void OnOnCommMscomm1();
afx_msg void OnOpenPort();
afx_msg void OnClosePort();
virtual void OnCancel();
DECLARE_EVENTSINK_MAP()
#}}AFX_MSG
DECLARE_MESSAGE_MAP()
private:
void UnRdBtn();
void SetAdd9();
void SetAdd8();
void SetAdd7();
void SetAdd6();
void SetAdd4();
void SetAdd3();
void SetAdd2();
void SetAdd1();
unsigned int port;
char Address; ///
char Command[10];
char Addr[10]; ///
CString RxData;
BOOL Start_Rx;
};

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

#endif //
!defined(AFX_ADDRESSSETUPDLG_H_04BC625F_7F22_463A_B0BE_C29190192331__INCLUDED_)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// AddressSetupDlg.cpp : implementation file
//

#include "stdafx.h"
#include "AddressSetup.h"
#include "AddressSetupDlg.h"

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

////////////////////////////////////
////
// CAddressSetupDlg dialog

CAddressSetupDlg::CAddressSetupDlg(CWnd* pParent /*=NULL*/)
: CDialog(CAddressSetupDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CAddressSetupDlg)
    m_port = _T("COM1");
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in
Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    Address = '0';
    port = 1;
    RxData = _T("");
    Start_Rx = FALSE;
}

void CAddressSetupDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAddressSetupDlg)
    DDX_Control(pDX, IDC_MSCOMM1, m_SerialComm);
    DDX_CBString(pDX, IDC_PORT, m_port);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAddressSetupDlg, CDialog)
    //{{AFX_MSG_MAP(CAddressSetupDlg)
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_READ_ADD, OnReadAdd)
    ON_BN_CLICKED(IDC_WRITE_ADD, OnWriteAdd)
    ON_BN_CLICKED(IDC_ADDRESS1, OnAddress1)
    ON_BN_CLICKED(IDC_ADDRESS2, OnAddress2)
    ON_BN_CLICKED(IDC_ADDRESS3, OnAddress3)
    ON_BN_CLICKED(IDC_ADDRESS4, OnAddress4)
    ON_BN_CLICKED(IDC_ADDRESS6, OnAddress6)
    ON_BN_CLICKED(IDC_ADDRESS7, OnAddress7)
    ON_BN_CLICKED(IDC_ADDRESS8, OnAddress8)
    ON_BN_CLICKED(IDC_ADDRESS9, OnAddress9)
    ON_BN_CLICKED(IDC_OPEN_PORT, OnOpenPort)
    ON_BN_CLICKED(IDC_CLOSE_PORT, OnClosePort)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////
////
// CAddressSetupDlg message handlers

BOOL CAddressSetupDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Set the icon for this dialog. The framework does this
    automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon
    // TODO: Add extra initialization here
    GetDlgItem(IDC_OPEN_PORT)->EnableWindow(TRUE);
    GetDlgItem(IDC_CLOSE_PORT)->EnableWindow(FALSE);
    GetDlgItem(IDC_PORT)->EnableWindow(TRUE);
    GetDlgItem(IDC_READ_ADD)->EnableWindow(FALSE);
    GetDlgItem(IDC_WRITE_ADD)->EnableWindow(FALSE);
    return TRUE; // return TRUE unless you set the focus to a
control
}

// If you add a minimize button to your dialog, you will need the code
below
// to draw the icon. For MFC applications using the document/view
model,
// this is automatically done for you by the framework.

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

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

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

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

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

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void CAddressSetupDlg::OnReadAdd()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    sprintf(Command, "@01#");
    m_SerialComm.SetOutput(COLEVariant(Command));
}

void CAddressSetupDlg::OnWriteAdd()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    sprintf(Command, "@02%c#", Address);
    m_SerialComm.SetOutput(COLEVariant(Command));
}

void CAddressSetupDlg::OnAddress1()
{
    // TODO: Add your control notification handler code here
    Address = '1';
}

void CAddressSetupDlg::OnAddress2()
{
    // TODO: Add your control notification handler code here
    Address = '2';
}

void CAddressSetupDlg::OnAddress3()
{
    // TODO: Add your control notification handler code here
    Address = '3';
}

void CAddressSetupDlg::OnAddress4()
{
    // TODO: Add your control notification handler code here
    Address = '4';
}

void CAddressSetupDlg::OnAddress6()
{
    // TODO: Add your control notification handler code here
    Address = '6';
}

void CAddressSetupDlg::OnAddress7()
{
    // TODO: Add your control notification handler code here
    Address = '7';
}

void CAddressSetupDlg::OnAddress8()
{
    // TODO: Add your control notification handler code here
    Address = '8';
}

void CAddressSetupDlg::OnAddress9()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    // TODO: Add your control notification handler code here
    Address = '9';
}

BEGIN_EVENTSINK_MAP(CAddressSetupDlg, CDialog)
//{{AFX_EVENTSINK_MAP(CAddressSetupDlg)
    ON_EVENT(CAddressSetupDlg, IDC_MSCOMM1, 1 /* OnComm */,
OnOnCommMscmm1, VTS_NONE)
//}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()

void CAddressSetupDlg::OnOnCommMscmm1()
{
    // TODO: Add your control notification handler code here
    if(m_SerialComm.GetCommEvent() == comEvReceive) {
        VARIANT SData;
        CString TempData;
        SData = m_SerialComm.GetInput();
        TempData = SData.bstrVal;
        RxData += TempData;
        if(!Start_Rx && TempData == '&')
            Start_Rx = TRUE;
        if(Start_Rx) {
            if(TempData == '$') {
                strcpy(Addr, RxData);
                if(strncmp(Addr, "&ADD>", 5) == 0 && Addr[6] ==
'S') {
                    switch(Addr[5] ) {
                        case '1' : SetAdd1(); break;
                        case '2' : SetAdd2(); break;
                        case '3' : SetAdd3(); break;
                        case '4' : SetAdd4(); break;
                        case '6' : SetAdd6(); break;
                        case '7' : SetAdd7(); break;
                        case '8' : SetAdd8(); break;
                        case '9' : SetAdd9(); break;
                        default : CString PAdd;
                            PAdd.Format("หมายเลข %c ไม่อยู่ในช่วงกำหนด\n โปรดทำการกำหนด
หมายเลขใหม่,", Addr[5]);
                            AfxMessageBox(PAdd); break;
                    }
                }
            }
            else if(Addr[5] == 'O' && Addr[6] == 'K' ) {
                CString PAdd;
                PAdd.Format("เครื่องถูกกำหนดเป็นหมายเลข %c เรียบร้อยแล้ว.",
Address);
                MessageBox(PAdd);
            }
            else {
                char AddErr, err[5], error;
                AddErr = Addr[1];
                err[0] = Addr[2];
                err[1] = Addr[3];
                err[2] = Addr[4];
                err[3] = Addr[5];
                err[4] = Addr[6];
                err[5] =NULL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        error = Addr[7];
        if(strcmp(err, ",ERR=") == 0) {
            CString PAdd;
            switch(error) {
                case '1' :
                    AfxMessageBox("ไม่สามารถกำหนดหมายเลขได้ เนื่องจากหน่วยความจำซ้ำชุด."); break;
                case '2' :
                    PAdd.Format("เครื่องหมายเลข %C ไม่สามารถกำหนดเป็นหมายเลข %C ได้",
                    AddErr, Address);
                    PAdd.Format("%s\n          เนื่องจากหมายเลข %C ไม่อยู่ในช่วงที่กำหนด",
                    (CString) PAdd, Address);
                    AfxMessageBox(PAdd); break;
                default : break;
            }
        }
        RxData = _T("");
        Start_Rx = FALSE;
    }
}
UpdateData(FALSE);
}

void CAddressSetupDlg::OnOpenPort()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    if(strcmp(m_port, "COM1") == 0)
        port = 1;
    else if(strcmp(m_port, "COM2") == 0)
        port = 2;
    else if(strcmp(m_port, "COM3") == 0)
        port = 3;
    else if(strcmp(m_port, "COM4") == 0)
        port = 4;
    m_SerialComm.SetCommPort(port);
    m_SerialComm.SetSettings("57600,N,8,1");
    m_SerialComm.SetInputLen(1);
    m_SerialComm.SetInputMode(0); // Rx Text
    m_SerialComm.SetRThreshold(1); // return RxEvent
    m_SerialComm.SetPortOpen(TRUE);
    GetDlgItem(IDC_READ_ADD)->EnableWindow(TRUE);
    GetDlgItem(IDC_WRITE_ADD)->EnableWindow(TRUE);
    GetDlgItem(IDC_OPEN_PORT)->EnableWindow(FALSE);
    GetDlgItem(IDC_CLOSE_PORT)->EnableWindow(TRUE);
    GetDlgItem(IDC_PORT)->EnableWindow(FALSE);
}

void CAddressSetupDlg::OnClosePort()
{
    // TODO: Add your control notification handler code here
    m_SerialComm.SetPortOpen(FALSE);
    GetDlgItem(IDC_READ_ADD)->EnableWindow(FALSE);
    GetDlgItem(IDC_WRITE_ADD)->EnableWindow(FALSE);
    GetDlgItem(IDC_OPEN_PORT)->EnableWindow(TRUE);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        GetDlgItem(IDC_CLOSE_PORT)->EnableWindow(FALSE);
        GetDlgItem(IDC_PORT)->EnableWindow(TRUE);
        UnRdBtn();
    }

void CAddressSetupDlg::SetAdd1()
{
    CButton* AddBtn;
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));
    AddBtn->SetCheck(TRUE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));
    AddBtn->SetCheck(FALSE);
}

void CAddressSetupDlg::SetAdd2()
{
    CButton* AddBtn;
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));
    AddBtn->SetCheck(TRUE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));
    AddBtn->SetCheck(FALSE);
}

void CAddressSetupDlg::SetAdd3()
{
    CButton* AddBtn;
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));
    AddBtn->SetCheck(TRUE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        AddBtn->SetCheck(FALSE);
        AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));
        AddBtn->SetCheck(FALSE);
        AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));
        AddBtn->SetCheck(FALSE);
    }

```

```

void CAddressSetupDlg::SetAdd4()

```

```

{
    CButton* AddBtn;
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));
    AddBtn->SetCheck(TRUE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));
    AddBtn->SetCheck(FALSE);
}

```

```

void CAddressSetupDlg::SetAdd6()

```

```

{
    CButton* AddBtn;
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));
    AddBtn->SetCheck(TRUE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));
    AddBtn->SetCheck(FALSE);
}

```

```

void CAddressSetupDlg::SetAdd7()

```

```

{
    CButton* AddBtn;
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        AddBtn->SetCheck(FALSE);
        AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));
        AddBtn->SetCheck(TRUE);
        AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));
        AddBtn->SetCheck(FALSE);
        AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));
        AddBtn->SetCheck(FALSE);
    }

```

```

void CAddressSetupDlg::SetAdd8()
{

```

```

    CButton* AddBtn;
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));
    AddBtn->SetCheck(TRUE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));
    AddBtn->SetCheck(FALSE);
}

```

```

void CAddressSetupDlg::SetAdd9()
{

```

```

    CButton* AddBtn;
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));
    AddBtn->SetCheck(FALSE);
    AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));
    AddBtn->SetCheck(TRUE);
}

```

```

void CAddressSetupDlg::OnCancel()
{

```

```

    // TODO: Add extra cleanup here
    if(m_SerialComm.GetPortOpen())
        m_SerialComm.SetPortOpen(FALSE);
    CDialog::OnCancel();
}

```

```

void CAddressSetupDlg::UnRdBtn()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CButton* AddBtn;  
AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS1));  
AddBtn->SetCheck(FALSE);  
AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS2));  
AddBtn->SetCheck(FALSE);  
AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS3));  
AddBtn->SetCheck(FALSE);  
AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS4));  
AddBtn->SetCheck(FALSE);  
AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS6));  
AddBtn->SetCheck(FALSE);  
AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS7));  
AddBtn->SetCheck(FALSE);  
AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS8));  
AddBtn->SetCheck(FALSE);  
AddBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_ADDRESS9));  
AddBtn->SetCheck(FALSE);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**โปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// logondb.H : Declaration of the Clogondb class

#ifndef __LOGONDB_H_
#define __LOGONDB_H_

class ClogondbAccessor
{
public:
    int m_stindex;
    TCHAR m_StationID[3];
    TCHAR m_Station[26];
    TCHAR m_Password[9];

BEGIN_COLUMN_MAP(ClogondbAccessor)
    COLUMN_ENTRY(1, m_stindex)
    COLUMN_ENTRY(2, m_StationID)
    COLUMN_ENTRY(3, m_Station)
    COLUMN_ENTRY(4, m_Password)
END_COLUMN_MAP()

DEFINE_COMMAND(ClogondbAccessor, _T(""))

    // You may wish to call this function if you are inserting a
    // record and wish to
    // initialize all the fields, if you are not going to explicitly
    // set all of them.
    void ClearRecord()
    {
        memset(this, 0, sizeof(*this));
    }
};

class Clogondb : public CCommand<CAccessor<ClogondbAccessor> >
{
public:
    HRESULT Open()
    {
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        return OpenRowset();
    }
    HRESULT OpenDataSource()
    {
        HRESULT hr;
        CDataSource db;
        CDBPropSet dbinit(DBPROPSET_DBINIT);

        dbinit.AddProperty(DBPROP_AUTH_PASSWORD, OLESTR("std3t"));
        dbinit.AddProperty(DBPROP_AUTH_USERID, OLESTR("project"));
        dbinit.AddProperty(DBPROP_INIT_DATASOURCE,
        OLESTR("skytrain"));
        dbinit.AddProperty(DBPROP_INIT_PROMPT, (short)4);
        dbinit.AddProperty(DBPROP_INIT_LCID, (long)1054);
        hr = db.Open(_T("MSDASQL"), &dbinit);
        if (FAILED(hr))
            return hr;
    }
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        m_session.Close();///
        return m_session.Open(db);
    }
HRESULT OpenRowset()
{
    // Set properties for open
    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);

    return CCommand<CAccessor<ClogondbAccessor>
>::Open(m_session, "SELECT * FROM logon_db", &propset);
}
HRESULT SearchDB(CString m_SearchDB) ///All Created
{
    HRESULT hr;
    hr = OpenDataSource();

    // Set properties for open
    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);
    propset.AddProperty(DBPROP_CANSCROLLBACKWARDS,true); ///
    propset.AddProperty(DBPROP_IMMOBILEROWS,true); ///
    hr = CCommand<CAccessor<ClogondbAccessor> >::Open(m_session,
m_SearchDB + "'", &propset);
    if (FAILED(hr))
        return hr;
    return MoveNext();
}
CSession m_session;
};
#endif // __LOGONDB_H__

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// CardType.H : Declaration of the CCardType class

#ifndef __CARDTYPE_H_
#define __CARDTYPE_H_

class CCardTypeAccessor
{
public:
    LONG m_Index;
    TCHAR m_CardID[15];
    TCHAR m_Type[11];

BEGIN_COLUMN_MAP(CCardTypeAccessor)
    COLUMN_ENTRY(1, m_Index)
    COLUMN_ENTRY(2, m_CardID)
    COLUMN_ENTRY(3, m_Type)
END_COLUMN_MAP()

DEFINE_COMMAND(CCardTypeAccessor, _T(" \
SELECT \
    Index, \
    Card_ID, \
    Type \
FROM type_of_card"))

    // You may wish to call this function if you are inserting a
    // record and wish to
    // initialize all the fields, if you are not going to explicitly
    // set all of them.
    void ClearRecord()
    {
        memset(this, 0, sizeof(*this));
    }
};

class CCardType : public CCommand<CAccessor<CCardTypeAccessor>>
{
public:
    HRESULT Open()
    {
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        return OpenRowset();
    }
    HRESULT OpenDataSource()
    {
        HRESULT hr;
        CDataSource db;
        CDBPropSet dbinit(DBPROPSET_DBINIT);

        dbinit.AddProperty(DBPROP_AUTH_PASSWORD, OLESTR("std3t"));
        dbinit.AddProperty(DBPROP_AUTH_USERID, OLESTR("project"));
        dbinit.AddProperty(DBPROP_INIT_DATASOURCE,
        OLESTR("skytrain"));
        dbinit.AddProperty(DBPROP_INIT_PROMPT, (short)4);
        dbinit.AddProperty(DBPROP_INIT_LCID, (long)1054);
        hr = db.Open(_T("MSDASQL"), &dbinit);
    }
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (FAILED(hr))
            return hr;
        m_session.Close(); ///
        return m_session.Open(db);
    }
    HRESULT OpenRowset()
    {
        // Set properties for open
        CDBPropSet propset(DBPROPSET_ROWSET);
        propset.AddProperty(DBPROP_IRowsetChange, true);
        propset.AddProperty(DBPROP_UPDATABILITY, DBPROPVAL_UP_CHANGE
| DBPROPVAL_UP_INSERT | DBPROPVAL_UP_DELETE);
        return CCommand<CAccessor<CCardTypeAccessor>
>::Open(m_session, NULL, &propset);
    }
    HRESULT MySQLCommand(CString Cmd)
    {
        // Set properties for open
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        CCommand<CNoAccessor, CNoRowset> Cmd1;
        return Cmd1.Open(m_session, Cmd, NULL, NULL, DBGUID_DBSQL,
false);
    }
    CSession m_session;
};
#endif // __CARDTYPE_H_

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// prepaidcredit.H : Declaration of the Cprepaidcredit class

#ifndef __PREPAIDCREDIT_H_
#define __PREPAIDCREDIT_H_

class CprepaidcreditAccessor
{
public:
    TCHAR m_CardID[15];
    LONG m_credit;

BEGIN_COLUMN_MAP(CprepaidcreditAccessor)
    COLUMN_ENTRY(1, m_CardID)
    COLUMN_ENTRY(2, m_credit)
END_COLUMN_MAP()

DEFINE_COMMAND(CprepaidcreditAccessor, _T(""))

    // You may wish to call this function if you are inserting a
    // record and wish to
    // initialize all the fields, if you are not going to explicitly
    // set all of them.
    void ClearRecord()
    {
        memset(this, 0, sizeof(*this));
    }
};

class Cprepaidcredit : public CCommand<CAccessor<CprepaidcreditAccessor>
>
{
public:
    HRESULT Open()
    {
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        return OpenRowset();
    }
    HRESULT OpenDataSource()
    {
        HRESULT hr;
        CDataSource db;
        CDBPropSet dbinit(DBPROPSET_DBINIT);

        dbinit.AddProperty(DBPROP_AUTH_PASSWORD, OLESTR("std3t"));
        dbinit.AddProperty(DBPROP_AUTH_USERID, OLESTR("project"));
        dbinit.AddProperty(DBPROP_INIT_DATASOURCE,
OLESTR("skytrain"));
        dbinit.AddProperty(DBPROP_INIT_PROMPT, (short)4);
        dbinit.AddProperty(DBPROP_INIT_LCID, (long)1054);
        hr = db.Open(_T("MSDASQL"), &dbinit);
        if (FAILED(hr))
            return hr;
        m_session.Close(); ///
        return m_session.Open(db);
    }
    HRESULT OpenRowset()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    // Set properties for open
    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);
    propset.AddProperty(DBPROP_UPDATABILITY, DBPROPVAL_UP_CHANGE
| DBPROPVAL_UP_INSERT | DBPROPVAL_UP_DELETE);
    propset.AddProperty(DBPROP_CANSCROLLBACKWARDS,true); ///
    propset.AddProperty(DBPROP_IMMOBILEROWS,true); ///

    return CCommand<CAccessor<CprepaidcreditAccessor>
>::Open(m_session, "SELECT * FROM prepaid_credit", &propset);
}
HRESULT MySQLCommand(CString Cmd)
{
    // Set properties for open
    HRESULT hr;

    hr = OpenDataSource();
    if (FAILED(hr))
        return hr;
    CCommand<CNoAccessor, CNoRowset> Cmd1;
    return Cmd1.Open(m_session, Cmd, NULL, NULL, DBGUID_DBSQL,
false);
}
HRESULT SearchCredit(CString m_CardID) ///All Created
{
    HRESULT hr;
    hr = OpenDataSource();

    // Set properties for open
    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);
    propset.AddProperty(DBPROP_CANSCROLLBACKWARDS,true); ///
    propset.AddProperty(DBPROP_IMMOBILEROWS,true); ///
    hr = CCommand<CAccessor<CprepaidcreditAccessor>
>::Open(m_session,
"SELECT * FROM prepaid_credit WHERE Card_ID = '" +
m_CardID + "'", &propset);
    if (FAILED(hr))
        return hr;
    return MoveNext();
}
CSession m_session;
};

#endif // __PREPAIDCREDIT_H_
// PrepaidTransit.H : Declaration of the CPrepaidTransit class

#ifndef __PREPAIDTRANSIT_H_
#define __PREPAIDTRANSIT_H_

class CPrepaidTransitAccessor
{
public:
    LONG m_order;
    TCHAR m_source[26];
    DBTIMESTAMP m_intime;
    TCHAR m_destination[26];
    DBTIMESTAMP m_outtime;
    LONG m_price;
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BEGIN_COLUMN_MAP(CPrepaidTransitAccessor)
    COLUMN_ENTRY(1, m_order)
    COLUMN_ENTRY(2, m_source)
    COLUMN_ENTRY(3, m_intime)
    COLUMN_ENTRY(4, m_destination)
    COLUMN_ENTRY(5, m_outtime)
    COLUMN_ENTRY(6, m_price)
END_COLUMN_MAP()

DEFINE_COMMAND(CPrepaidTransitAccessor, _T(""))

    // You may wish to call this function if you are inserting a
    // record and wish to
    // initialize all the fields, if you are not going to explicitly
    // set all of them.
    void ClearRecord()
    {
        memset(this, 0, sizeof(*this));
    }
};

class CPrepaidTransit : public
CCommand<CAccessor<CPrepaidTransitAccessor> >
{
public:
    HRESULT Open(CString Card_ID)
    {
        HRESULT hr;
        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;
        return OpenRowset(Card_ID);
    }
    HRESULT OpenDataSource()
    {
        HRESULT hr;
        CDataSource db;
        CDBPropSet dbinit(DBPROPSET_DBINIT);

        dbinit.AddProperty(DBPROP_AUTH_PASSWORD, OLESTR("std3t"));
        dbinit.AddProperty(DBPROP_AUTH_USERID, OLESTR("project"));
        dbinit.AddProperty(DBPROP_INIT_DATASOURCE,
OLESTR("prepaid_transit"));
        dbinit.AddProperty(DBPROP_INIT_PROMPT, (short)4);
        dbinit.AddProperty(DBPROP_INIT_LCID, (long)1054);
        hr = db.Open(_T("MSDASQL"), &dbinit);
        if (FAILED(hr))
            return hr;
        m_session.Close(); ///
        return m_session.Open(db);
    }
    HRESULT OpenRowset(CString Card_ID)
    {
        // Set properties for open
        CDBPropSet propset(DBPROPSET_ROWSET);
        propset.AddProperty(DBPROP_IRowsetChange, true);
        propset.AddProperty(DBPROP_UPDATABILITY, DBPROPVAL_UP_CHANGE
| DBPROPVAL_UP_INSERT | DBPROPVAL_UP_DELETE);
        propset.AddProperty(DBPROP_CANSCROLLBACKWARDS, true); ///

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

propset.AddProperty(DBPROP_IMMOBILEROWS,true); ///

        return          CCommand<CAccessor<CPrepaidTransitAccessor>
>::Open(m_session, "SELECT * FROM id_" + Card_ID, &propset);
    }
    HRESULT MySQLCommand(CString Cmd)
    {
        // Set properties for open
        HRESULT          hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        CCommand<CNoAccessor, CNoRowset> Cmd1;
        return Cmd1.Open(m_session, Cmd, NULL, NULL, DBGUID_DBSQL,
false);
    }
    CSession    m_session;
};

#endif // __PREPAIDTRANSIT_H_

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// postpaidinfo.H : Declaration of the Cpostpaidinfo class

#ifndef __POSTPAIDINFO_H_
#define __POSTPAIDINFO_H_

class CpostpaidinfoAccessor
{
public:
    LONG m_CustomerID;
    TCHAR m_CardID[15];
    TCHAR m_FirstName[21];
    TCHAR m_LastName[21];
    TCHAR m_PersonalID[14];
    TCHAR m_Address[101];
    TCHAR m_City[21];
    TCHAR m_Zipcode[6];
    TCHAR m_Telephone[21];

BEGIN_COLUMN_MAP(CpostpaidinfoAccessor)
    COLUMN_ENTRY(1, m_CustomerID)
    COLUMN_ENTRY(2, m_CardID)
    COLUMN_ENTRY(3, m_FirstName)
    COLUMN_ENTRY(4, m_LastName)
    COLUMN_ENTRY(5, m_PersonalID)
    COLUMN_ENTRY(6, m_Address)
    COLUMN_ENTRY(7, m_City)
    COLUMN_ENTRY(8, m_Zipcode)
    COLUMN_ENTRY(9, m_Telephone)
END_COLUMN_MAP()

DEFINE_COMMAND(CpostpaidinfoAccessor, _T(""))

    // You may wish to call this function if you are inserting a
    // record and wish to
    // initialize all the fields, if you are not going to explicitly
    // set all of them.
    void ClearRecord()
    {
        memset(this, 0, sizeof(*this));
    }
};

class Cpostpaidinfo : public CCommand<CAccessor<CpostpaidinfoAccessor> >
{
public:
    HRESULT Open()
    {
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        return OpenRowset();
    }
    HRESULT OpenDataSource()
    {
        HRESULT hr;
        CDataSource db;
        CDBPropSet dbinit(DBPROPSET_DBINIT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        dbinit.AddProperty(DBPROP_AUTH_PASSWORD, OLESTR("std3t"));
        dbinit.AddProperty(DBPROP_AUTH_USERID, OLESTR("project"));
        dbinit.AddProperty(DBPROP_INIT_DATASOURCE,
OLESTR("skytrain"));
        dbinit.AddProperty(DBPROP_INIT_PROMPT, (short)4);
        dbinit.AddProperty(DBPROP_INIT_LCID, (long)1054);
        hr = db.Open(_T("MSDASQL"), &dbinit);
        if (FAILED(hr))
            return hr;
        m_session.Close(); ///
        return m_session.Open(db);
    }
    HRESULT OpenRowset()
    {
        // Set properties for open
        CDBPropSet propset(DBPROPSET_ROWSET);
        propset.AddProperty(DBPROP_IRowsetChange, true);
        propset.AddProperty(DBPROP_UPDATABILITY, DBPROPVAL_UP_CHANGE
| DBPROPVAL_UP_INSERT | DBPROPVAL_UP_DELETE);

        return CCommand<CAccessor<CpostpaidinfoAccessor>
>::Open(m_session, "SELECT * FROM postpaid_info", &propset);
    }
    HRESULT MySQLCommand(CString Cmd)
    {
        // Set properties for open
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        CCommand<CNoAccessor, CNoRowset> Cmd1;
        return Cmd1.Open(m_session, Cmd, NULL, NULL, DBGUID_DBSQL,
false);
    }
    HRESULT SearchCusID(CString m_CardID) ///All Created
    {
        HRESULT hr;
        hr = OpenDataSource();

        // Set properties for open
        CDBPropSet propset(DBPROPSET_ROWSET);
        propset.AddProperty(DBPROP_IRowsetChange, true);
        propset.AddProperty(DBPROP_CANSCROLLBACKWARDS, true); ///
        propset.AddProperty(DBPROP_IMMOBILEROWS, true); ///
        hr = CCommand<CAccessor<CpostpaidinfoAccessor>
>::Open(m_session,
"SELECT * FROM postpaid_info WHERE Card_ID = '" +
m_CardID + "'", &propset);
        if (FAILED(hr))
            return hr;
        return MoveNext();
    }
    CSession m_session;
};

#endif // __POSTPAIDINFO_H_

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// postpaidamount.H : Declaration of the Cpostpaidamount class

#ifndef __POSTPAIDAMOUNT_H_
#define __POSTPAIDAMOUNT_H_

class CpostpaidamountAccessor
{
public:
    TCHAR m_CardID[15];
    LONG m_amount;

BEGIN_COLUMN_MAP(CpostpaidamountAccessor)
    COLUMN_ENTRY(1, m_CardID)
    COLUMN_ENTRY(2, m_amount)
END_COLUMN_MAP()

DEFINE_COMMAND(CpostpaidamountAccessor, _T(""))

    // You may wish to call this function if you are inserting a
    // record and wish to
    // initialize all the fields, if you are not going to explicitly
    // set all of them.
    void ClearRecord()
    {
        memset(this, 0, sizeof(*this));
    }
};

class Cpostpaidamount : public CCommand<CAccessor<CpostpaidamountAccessor> >
{
public:
    HRESULT Open()
    {
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        return OpenRowset();
    }
    HRESULT OpenDataSource()
    {
        HRESULT hr;
        CDataSource db;
        CDBPropSet dbinit(DBPROPSET_DBINIT);

        dbinit.AddProperty(DBPROP_AUTH_PASSWORD, OLESTR("std3t"));
        dbinit.AddProperty(DBPROP_AUTH_USERID, OLESTR("project"));
        dbinit.AddProperty(DBPROP_INIT_DATASOURCE,
OLESTR("skytrain"));
        dbinit.AddProperty(DBPROP_INIT_PROMPT, (short)4);
        dbinit.AddProperty(DBPROP_INIT_LCID, (long)1054);
        hr = db.Open(_T("MSDASQL"), &dbinit);
        if (FAILED(hr))
            return hr;
        m_session.Close(); ///
        return m_session.Open(db);
    }
    HRESULT OpenRowset()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    // Set properties for open
    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);
    propset.AddProperty(DBPROP_UPDATABILITY, DBPROPVAL_UP_CHANGE
| DBPROPVAL_UP_INSERT | DBPROPVAL_UP_DELETE);

    return CCommand<CAccessor<CpostpaidamountAccessor>
>::Open(m_session, "SELECT * FROM postpaid_amount", &propset);
}
HRESULT MySQLCommand(CString Cmd)
{
    // Set properties for open
    HRESULT hr;

    hr = OpenDataSource();
    if (FAILED(hr))
        return hr;

    CCommand<CNoAccessor, CNoRowset> Cmd1;
    return Cmd1.Open(m_session, Cmd, NULL, NULL, DBGUID_DBSQL,
false);
}
HRESULT SearchAmount(CString m_CardID) //All Created
{
    HRESULT hr;
    hr = OpenDataSource();

    // Set properties for open
    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);
    propset.AddProperty(DBPROP_CANSCROLLBACKWARDS, true); //
    propset.AddProperty(DBPROP_IMMOBILEROWS, true); //
    hr = CCommand<CAccessor<CpostpaidamountAccessor>
>::Open(m_session,
"SELECT * FROM postpaid_amount WHERE Card_ID = '" +
m_CardID + "'", &propset);
    if (FAILED(hr))
        return hr;
    return MoveNext();
}
CSession m_session;
};

#endif // __POSTPAIDAMOUNT_H_

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// PostpaidTransit.H : Declaration of the CPostpaidTransit class

#ifndef __POSTPAIDTRANSIT_H_
#define __POSTPAIDTRANSIT_H_

class CPostpaidTransitAccessor
{
public:
    LONG m_order;
    TCHAR m_source[26];
    DBTIMESTAMP m_intime;
    TCHAR m_destination[26];
    DBTIMESTAMP m_outtime;
    LONG m_price;

BEGIN_COLUMN_MAP(CPostpaidTransitAccessor)
    COLUMN_ENTRY(1, m_order)
    COLUMN_ENTRY(2, m_source)
    COLUMN_ENTRY(3, m_intime)
    COLUMN_ENTRY(4, m_destination)
    COLUMN_ENTRY(5, m_outtime)
    COLUMN_ENTRY(6, m_price)
END_COLUMN_MAP()

DEFINE_COMMAND(CPostpaidTransitAccessor, _T(""))

    // You may wish to call this function if you are inserting a
    // record and wish to
    // initialize all the fields, if you are not going to explicitly
    // set all of them.
    void ClearRecord()
    {
        memset(this, 0, sizeof(*this));
    }
};

class CPostpaidTransit : public
CCommand<CAccessor<CPostpaidTransitAccessor> >
{
public:
    HRESULT Open(CString Card_ID)
    {
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        return OpenRowset(Card_ID);
    }
    HRESULT OpenDataSource()
    {
        HRESULT hr;
        CDataSource db;
        CDBPropSet dbinit(DBPROPSET_DBINIT);

        dbinit.AddProperty(DBPROP_AUTH_PASSWORD, OLESTR("std3t"));
        dbinit.AddProperty(DBPROP_AUTH_USERID, OLESTR("project"));
        dbinit.AddProperty(DBPROP_INIT_DATASOURCE,
        OLESTR("postpaid_transit"));
        dbinit.AddProperty(DBPROP_INIT_PROMPT, (short)4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        dbinit.AddProperty(DBPROP_INIT_LCID, (long)1054);
        hr = db.Open(_T("MSDASQL"), &dbinit);
        if (FAILED(hr))
            return hr;
        m_session.Close(); ///
        return m_session.Open(db);
    }
HRESULT OpenRowset(CString Card_ID)
{
    // Set properties for open
    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);
    propset.AddProperty(DBPROP_UPDATABILITY, DBPROPVAL_UP_CHANGE
| DBPROPVAL_UP_INSERT | DBPROPVAL_UP_DELETE);
    propset.AddProperty(DBPROP_CANSCROLLBACKWARDS,true); ///
    propset.AddProperty(DBPROP_IMMOBILEROWS,true); ///

    return CCommand<CAccessor<CPostpaidTransitAccessor>
>::Open(m_session,"SELECT * FROM id_" + Card_ID, &propset);
}
HRESULT MySQLCommand(CString Cmd)
{
    // Set properties for open
    HRESULT hr;

    hr = OpenDataSource();
    if (FAILED(hr))
        return hr;

    CCommand<CNoAccessor, CNoRowset> Cmd1;
    return Cmd1.Open(m_session, Cmd, NULL, NULL, DBGUID_DBSQL,
false);
}
CSession m_session;
};
#endif // __POSTPAIDTRANSIT_H_

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#if
!defined(AFX_LOGON_H__9D050037_5D5C_4BBB_8738_6658B1BC8C74__INCLUDED_)
#define AFX_LOGON_H__9D050037_5D5C_4BBB_8738_6658B1BC8C74__INCLUDED_

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

#include "logondb.H" ///
// LogOn.h : header file
//

////////////////////////////////////
/////
// CLogOn dialog

class CLogOn : public CDialog
{
// Construction
public:
    CLogOn(CWnd* pParent = NULL); // standard constructor
    Clogondb m_logondb; ///
    CString StationName;
    CString StationID;
// Dialog Data
    //{AFX_DATA(CLogOn)
    enum { IDD = IDD_LOG_ON };
    CComboBox m_StationName;
    //}AFX_DATA

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

// Implementation
protected:

    // Generated message map functions
    //{AFX_MSG(CLogOn)
    afx_msg void OnConnectDb();
    virtual BOOL OnInitDialog();
    //}AFX_MSG
    DECLARE_MESSAGE_MAP()
private:
    void AddStation();
    BOOL m_Connect;
    BOOL m_AddStation;
};

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

#endif
!defined(AFX_LOGON_H__9D050037_5D5C_4BBB_8738_6658B1BC8C74__INCLUDED_)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// LogOn.cpp : implementation file
//

#include "stdafx.h"
#include "CardManagement.h"
#include "LogOn.h"

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

////////////////////////////////////
////
// CLogOn dialog

CLogOn::CLogOn(CWnd* pParent /*=NULL*/)
    : CDialog(CLogOn::IDD, pParent)
{
   //{{AFX_DATA_INIT(CLogOn)
    //}}AFX_DATA_INIT
    m_Connect = FALSE;
    m_AddStation = FALSE;
}

void CLogOn::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CLogOn)
    DDX_Control(pDX, IDC_STATION_NAME, m_StationName);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CLogOn, CDialog)
   //{{AFX_MSG_MAP(CLogOn)
    ON_BN_CLICKED(IDC_CONNECT_DB, OnConnectDb)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
////
// CLogOn message handlers

BOOL CLogOn::OnInitDialog()
{
    CDialog::OnInitDialog();

    // TODO: Add extra initialization here
    GetDlgItem(IDOK)->EnableWindow(FALSE);
    m_AddStation = TRUE;
    AddStation();
    return TRUE; // return TRUE unless you set the focus to a control
                // EXCEPTION: OCX Property Pages should return FALSE
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CLogOn::OnConnectDb()
{
    // TODO: Add your control notification handler code here
    CString PwdDB;
    GetDlgItemText(IDC_STATION_ID, StationName);
    GetDlgItemText(IDC_PASSWORD, PwdDB);
    if(m_Connect == FALSE && StationName != _T("") && PwdDB !=
_T("")) {
        if (m_logondb.Open() == S_OK) {
            if( m_logondb.MoveNext() == S_OK) {
                m_logondb.Close();
                m_logondb.SearchDB(StationName);
                if(PwdDB.Collate(m_logondb.m_Password) == 0) {
                    m_Connect = TRUE;
                    StationID = m_logondb.m_StationID;
                    GetDlgItem(IDOK)->EnableWindow(TRUE);
                    GetDlgItem(IDC__CONNECT_DB)-
>EnableWindow(FALSE);
                    GetDlgItem(IDC_STATION_ID)-
>EnableWindow(FALSE);
                    GetDlgItem(IDC_PASSWORD)-
>EnableWindow(FALSE);
                }
            }
            else {
                SetDlgItemText(IDC_PASSWORD, "");
                AfxMessageBox("รหัสผ่านไม่ถูกต้อง");
            }
            m_logondb.Close();
        }
        else
        {
            AfxMessageBox(_T("ไม่สามารถเปิดใช้งานข้อมูลได้"), MB_OK);
            m_Connect = FALSE;
        }
    }
}

void CLogOn::AddStation()
{
    if (m_logondb.Open() == S_OK) {
        while(m_AddStation) {
            if(m_logondb.MoveNext() == S_OK)
                m_StationName.AddString(m_logondb.m_Station);
            else {
                m_logondb.Close();
                m_AddStation = FALSE;
            }
        }
        m_StationName.SetCurSel(0);
    }
    else
    {
        AfxMessageBox(_T("ไม่สามารถเชื่อมต่อฐานข้อมูลได้เนื่องจากระบบเครือข่ายขัดข้อง"), MB_OK);
        GetDlgItem(IDC_CONNECT_DB)->EnableWindow(FALSE);
        m_Connect = FALSE;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#ifndef INC_STABCTRL
#define INC_STABCTRL

#include <afxtempl.h>

class CStabCtrl : public CTabCtrl
{
public:
    CStabCtrl();
public:
    virtual ~CStabCtrl();
    INT m_nPrevSel;
    virtual BOOL AttachControlToTab(CWnd * _pControl,INT _nTabNum);
    virtual SetCurSel( int nItem );
    // Generated message map functions
protected:
    //{{AFX_MSG(CStabCtrl)
    afx_msg BOOL OnSelchange(NMHDR* pNMHDR, LRESULT* pResult);
    afx_msg void OnDestroy();
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
    afx_msg void OnEnable( BOOL bEnable );
    //}}AFX_MSG

    DECLARE_MESSAGE_MAP()
private:
    class CStabPage
    {
    private:
        CList <CWnd *, CWnd *> m_ControlList;
    public:
        CStabPage();
        ~CStabPage();

        BOOL AttachControl (CWnd * _pControl);
        BOOL ShowWindows ( INT _nCmdShow );
        BOOL EnableWindows ( BOOL _bEnable );
    };

    class CPageToControlsMap : public CMap <INT, INT&,CStabPage *,
CStabPage *>
    {
    public:
        CPageToControlsMap( );
        ~CPageToControlsMap( );

        BOOL AttachControl(INT _nTabNum,CWnd * _pControl);
        BOOL ShowWindows ( INT _nCurrPage, INT _nCmdShow );
        BOOL EnableWindows ( INT _nCurrPage, BOOL _bEnable );
        void RemoveAll( );
    };
    //INT m_nPrevSel;
    CPageToControlsMap m_TabPagesMap;
};
#endif // INC_STABCTRL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////
////
// STabCtrl.cpp : implementation file
//

#include "stdafx.h"
#include "STabCtrl.h"

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

CSTabCtrl::CSTabPage::CSTabPage()
{

}

CSTabCtrl::CSTabPage::~CSTabPage()
{
    m_ControlList.RemoveAll();
}

BOOL CSTabCtrl::CSTabPage::AttachControl (CWnd * _pControl)
{
    return (BOOL) m_ControlList.AddTail (_pControl);
}

BOOL CSTabCtrl::CSTabPage::ShowWindows ( INT _nCmdShow )
{
    POSITION _rPos = m_ControlList.GetHeadPosition();
    CWnd * _pCtrl = NULL;

    while(_rPos)
    {
        _pCtrl = m_ControlList.GetNext(_rPos);

        ::ShowWindow(_pCtrl -> GetSafeHwnd(), _nCmdShow );

        // cannot use because using this call with the MS WebBrowser
        // control destroys the control's window.
        // _pCtrl -> ShowWindow(_nCurrSel == _rKey);
    }

    return TRUE;
}

BOOL CSTabCtrl::CSTabPage::EnableWindows ( BOOL _bEnable )
{
    POSITION _rPos = m_ControlList.GetHeadPosition();
    CWnd * _pCtrl = NULL;

    while(_rPos)
    {
        _pCtrl = m_ControlList.GetNext(_rPos);

        ::EnableWindow(_pCtrl -> GetSafeHwnd(), _bEnable );
    }

    return TRUE;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

CSTabCtrl::CPageToControlsMap::CPageToControlsMap( )
{
}

CSTabCtrl::CPageToControlsMap::~CPageToControlsMap( )
{
    CPageToControlsMap::RemoveAll();
}

BOOL CSTabCtrl::CPageToControlsMap::AttachControl( INT    _nTabNum, CWnd    *
_pControl)
{
    CSTabPage * _pTabPage = NULL;

    if(!Lookup(_nTabNum, _pTabPage) || !_pTabPage)
    {
        _pTabPage = new CSTabPage();
        SetAt(_nTabNum, _pTabPage);
    }

    if(_pTabPage -> AttachControl(_pControl))
    {
        // make sure control is hidden...
        ::ShowWindow(_pControl -> GetSafeHwnd(), SW_HIDE );
        return TRUE;
    }
    else
    {
        return FALSE;
    }
}

BOOL CSTabCtrl::CPageToControlsMap::EnableWindows ( INT _nCurrPage, BOOL
_bEnable )
{
    CList <CWnd *, CWnd *> * _pCtrlList = NULL;
    CSTabPage * _pTabPage = NULL;

    if(Lookup(_nCurrPage, _pTabPage) && _pTabPage)
    {
        return _pTabPage -> EnableWindows ( _bEnable );
    }
    else
    {
        return FALSE;
    }
}

BOOL CSTabCtrl::CPageToControlsMap::ShowWindows ( INT _nCurrPage, INT
_nCmdShow )
{
    CList <CWnd *, CWnd *> * _pCtrlList = NULL;
    CSTabPage * _pTabPage = NULL;

    if(Lookup(_nCurrPage, _pTabPage) && _pTabPage)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return _pTabPage -> ShowWindows ( _nCmdShow );
    }
    else
    {
        return FALSE;
    }
}

void CTabCtrl::CPageToControlsMap::RemoveAll( )
{
    POSITION _rPos = GetStartPosition();
    CTabPage * _pValue = NULL;
    INT _rKey;

    while(_rPos)
    {
        GetNextAssoc(_rPos, _rKey, _pValue);
        delete _pValue;
    }

    CMap <INT, INT&,CTabPage *, CTabPage *>::RemoveAll();
}

////////////////////////////////////
////
// CTabCtrl

CTabCtrl::CTabCtrl()
{
    m_nPrevSel = -1;
}

CTabCtrl::~CTabCtrl()
{
}

BEGIN_MESSAGE_MAP(CTabCtrl, CTabCtrl)
    //{{AFX_MSG_MAP(CTabCtrl)
    ON_NOTIFY_REFLECT_EX(TCN_SELCHANGE, OnSelchange)
    ON_WM_DESTROY()
    ON_WM_CREATE()
    ON_WM_ENABLE()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
////
// CTabCtrl message handlers

int CTabCtrl::SetCurSel( int nItem )
{
    int _nRetVal = CTabCtrl::SetCurSel(nItem);

    NMHDR _nDummyNMHDR;
    LRESULT _nDummyLRESULT;

    OnSelchange (&_nDummyNMHDR, &_nDummyLRESULT);

    return _nRetVal;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BOOL CTabCtrl::AttachControlToTab(CWnd * _pControl,
                                  INT _nTabNum)
{
    if(_nTabNum >= GetItemCount())
    {
        ASSERT(FALSE);
        return FALSE;
    }

    m_TabPagesMap.AttachControl (_nTabNum, _pControl);

    return TRUE;
}

void CTabCtrl::OnEnable( BOOL bEnable )
{
    CTabCtrl::OnEnable(bEnable);

    INT _nTabPageCount (GetItemCount());

    for(INT i = 0; i < _nTabPageCount; i ++)
    {
        m_TabPagesMap.EnableWindows (i, bEnable);
    }
}

BOOL CTabCtrl::OnSelchange(NMHDR* pNMHDR, LRESULT* pResult)
{
    // TODO: Add your control notification handler code here
    INT _nCurrSel = GetCurSel();

    // hide Previous pages controls...
    m_TabPagesMap.ShowWindows (m_nPrevSel, SW_HIDE);

    // show current pages controls.
    m_TabPagesMap.ShowWindows (_nCurrSel, SW_SHOW);

    m_nPrevSel = _nCurrSel;

    *pResult = 0;

    return FALSE; // allow control to handle as well.
}

int CTabCtrl::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    m_nPrevSel = -1;
    return 0;
}

void CTabCtrl::OnDestroy()
{
    CTabCtrl::OnDestroy();

    // clean up map.
    m_TabPagesMap.RemoveAll();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#if
!defined(AFX_EDITNUM_H__5A7E06EE_85A6_40CD_B87E_2EB2D00A1A4B__INCLUDED_)
#define AFX_EDITNUM_H__5A7E06EE_85A6_40CD_B87E_2EB2D00A1A4B__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// EditNum.h : header file
//

////////////////////////////////////
/////
// CEditNum window

class CEditNum : public CEdit
{
// Construction
public:
    CEditNum();

// Attributes
public:

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CEditNum)
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CEditNum();

    // Generated message map functions
protected:
    //{{AFX_MSG(CEditNum)
    afx_msg void OnChar(UINT nChar, UINT nRepCnt, UINT nFlags);
    afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
    afx_msg void OnSetFocus(CWnd* pOldWnd);
    //}}AFX_MSG

    DECLARE_MESSAGE_MAP()
private:
    BOOL m_Focus;
};

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

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

#endif //
!defined(AFX_EDITNUM_H__5A7E06EE_85A6_40CD_B87E_2EB2D00A1A4B__INCLUDED_)

```

// EditNum.cpp : implementation file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//

#include "stdafx.h"
#include "CardManagement.h"
#include "EditNum.h"

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

////////////////////////////////////
////
// CEditNum

CEditNum::CEditNum()
{
}

CEditNum::~CEditNum()
{
}

BEGIN_MESSAGE_MAP(CEditNum, CEdit)
    //{AFX_MSG_MAP(CEditNum)
    ON_WM_CHAR()
    ON_WM_LBUTTONDOWN()
    ON_WM_SETFOCUS()
    //}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
////
// CEditNum message handlers

void CEditNum::OnSetFocus(CWnd* pOldWnd)
{
    CEdit::OnSetFocus(pOldWnd);

    // TODO: Add your message handler code here
    m_Focus = TRUE;
    //Force another WM_LBUTTONDOWN
    PostMessage(WM_LBUTTONDOWN, 0, 0);
}

void CEditNum::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    if(TRUE == m_Focus)
    {
        SetSel(0, -1);
        m_Focus = FALSE;
    }
    else
        CEdit::OnLButtonDown(nFlags, point);
}

void CEditNum::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// TODO: Add your message handler code here and/or call default
//Backspace
if(nChar>='0' && nChar<='9' || 8==nChar)
    CEdit::OnChar(nChar, nRepCnt, nFlags);
else
    MessageBeep(0xFFFFFFFF);
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// CardManagementDlg.h : header file
//
//{{AFX_INCLUDES()}
//}}AFX_INCLUDES
#if
!defined(AFX_CARDMANAGEMENTDLG_H__2395A474_7804_4D59_AC0F_5F5EBA6F1DA5__
INCLUDED_)
#define
AFX_CARDMANAGEMENTDLG_H__2395A474_7804_4D59_AC0F_5F5EBA6F1DA5__INCLUDED_

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

#include "postpaidinfo.H"
#include "postpaidamount.H"
#include "prepaidcredit.H"
#include "CardType.H"
#include "PrepaidTransit.H"
#include "PostpaidTransit.H"
#include "resource.h"
#include "serialcomm.h"
#include "STabCtrl.h"
#include "EditNum.h"

#define comEvReceive 2
#define PrePaidIndex 0
#define PostPaidIndex 1
////////////////////////////////////
////
// CCardManagementDlg dialog

class CCardManagementDlg : public CDialog
{
// Construction
public:
    CCardManagementDlg(CWnd* pParent = NULL); // standard constructor
    Cpostpaidinfo      postpaidinfo;
    Cpostpaidamount    postpaidamount;
    Cprepaidcredit      prepaidcredit;
    CCardType          CardType;
    CPrepaidTransit    PrepaidTran;
    CPostpaidTransit   PostpaidTran;

// Dialog Data
    {{{AFX_DATA(CCardManagementDlg)
    enum { IDD = IDD_CARDMANAGEMENT_DIALOG };
    CStatic      m_help_pre;
    CStatic      m_help_post;
    CStatic      m_static_pre;
    CEditNum     m_show_credit;
    CStatic      m_s_show_credit;
    CEditNum     m_credit_refill;
    CStatic      m_s_credit_refill;
    CStatic      m_cus_ser_pre;
    CButton      m_cancel_ent_pre;
    CButton      m_chk_credit;
    CButton      m_refill;
    CButton      m_cus_del_pre;
    CButton      m_cus_add_pre;
    CStatic      m_static_post;
    }}}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CStatic      m_s_amount;
CEdit m_show_amount;
CStatic m_cus_data_post;
CStatic      m_cus_ser_post;
CButton      m_cus_del_post;
CButton      m_cus_add_post;
CButton      m_cancel_ent_post;
CButton      m_chk_amount;
CStatic      m_s_zipcode;
CStatic      m_s_phone;
CStatic      m_s_per_id;
CStatic      m_s_last_name;
CStatic      m_s_first_name;
CStatic      m_s_cus_id;
CStatic      m_s_city;
CStatic      m_s_address;
CEditNum     m_phone;
CEditNum     m_per_id;
CEdit m_last_name;
CEditNum     m_zipcode;
CEdit m_address;
CEdit m_first_name;
CEditNum     m_cus_id;
CEdit m_city;
CString      m_port;
CTabCtrl     m_TabCtrl;
CMSComm      m_SerialComm;
UINT m_cusid;
CString      m_pers_id;
CString      m_firstname;
CString      m_lastname;
CString      m_addr;
CString      m_city2;
CString      m_zipcode2;
CString      m_phone2;
UINT m_amount;
UINT m_credit_refill2;
UINT m_show_credit2;

//}}AFX_DATA

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

// Implementation
protected:
HICON m_hIcon;
// Generated message map functions
//{{AFX_MSG(CCardManagementDlg)
virtual BOOL OnInitDialog();
afx_msg void OnPaint();
afx_msg HCURSOR OnQueryDragIcon();
afx_msg void OnOpenPort();
afx_msg void OnClosePort();
virtual void OnCancel();
afx_msg void OnOnCommMscomm1();
afx_msg void OnCusAddPost();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

afx_msg void OnCusDelPost();
afx_msg void OnChkAmount();
afx_msg void OnCancelEntPost();
afx_msg void OnCusAddPre();
afx_msg void OnCusDelPre();
afx_msg void OnRefill();
afx_msg void OnChkCredit();
afx_msg void OnCancelEntPre();
afx_msg void OnSelchangeTypeTab(NMHDR* pNMHDR, LRESULT* pResult);
DECLARE_EVENTSINK_MAP()
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
private:
    BOOL Check_LRC(CString Data);
    unsigned int port;
    unsigned int process_pre;
    unsigned int process_post;
    CString StrRx_Buffer;
    BOOL Start_Rx;
    BOOL StartProcess;
public:
    void ClearDataCusPost();
    void Cancel_Ent_Post();
    void Cancel_Ent_Pre();
    void Show_Amount();
    void Show_Credit();
    void PrePaid_Refill();
    void Delete_PostPaid();
    void Delete_PrePaid();
    void Create_PreTable();
    void Create_PostTable();

    void Close_All();
    void Open_Cus_Info();
    void Close_Cus_Info();
    CString StationID;
    CString StationName;
    CString Card_ID;
};

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

#endif
!defined(AFX_CARDMANAGEMENTDLG_H__2395A474_7804_4D59_AC0F_5F5EBA6F1DA5__
INCLUDED_)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// CardManagementDlg.cpp : implementation file
//

#include "stdafx.h"
#include "CardManagement.h"
#include "CardManagementDlg.h"
///
#include "LogOn.h"

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

////////////////////////////////////
////
// CCardManagementDlg dialog

CCardManagementDlg::CCardManagementDlg(CWnd* pParent /*=NULL*/)
: CDialog(CCardManagementDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CCardManagementDlg)
    m_port = _T("COM1");
    m_cusid = 0;
    m_pers_id = _T("");
    m_firstname = _T("");
    m_lastname = _T("");
    m_addr = _T("");
    m_city2 = _T("");
    m_zipcode2 = _T("");
    m_phone2 = _T("");
    m_amount = 0;
    m_credit_refill2 = 0;
    m_show_credit2 = 0;
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in
Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    process_pre = 1;
    process_post = 1;
    port = 1;
    StartProcess = FALSE;
    Start_Rx = FALSE;
    StrRx_Buffer = _T("");
}

void CCardManagementDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CCardManagementDlg)
    DDX_Control(pDX, IDC_HELP_PRE, m_help_pre);
    DDX_Control(pDX, IDC_HELP_POST, m_help_post);
    DDX_Control(pDX, IDC_STATIC_PRE, m_static_pre);
    DDX_Control(pDX, IDC_SHOW_CREDIT, m_show_credit);
    DDX_Control(pDX, IDC_S_SHOW_CREDIT, m_s_show_credit);
    DDX_Control(pDX, IDC_CREDIT_REFILL, m_credit_refill);
    DDX_Control(pDX, IDC_S_CREDIT_REFILL, m_s_credit_refill);
    DDX_Control(pDX, IDC_CUS_SER_PRE, m_cus_ser_pre);
    DDX_Control(pDX, IDC_CANCEL_ENT_PRE, m_cancel_ent_pre);
    DDX_Control(pDX, IDC_CHK_CREDIT, m_chk_credit);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DDX_Control(pDX, IDC_REFILL, m_refill);
DDX_Control(pDX, IDC_CUS_DEL_PRE, m_cus_del_pre);
DDX_Control(pDX, IDC_CUS_ADD_PRE, m_cus_add_pre);
DDX_Control(pDX, IDC_STATIC_POST, m_static_post);
DDX_Control(pDX, IDC_S_AMOUNT, m_s_amount);
DDX_Control(pDX, IDC_SHOW_AMOUNT, m_show_amount);
DDX_Control(pDX, IDC_CUS_DATA_POST, m_cus_data_post);
DDX_Control(pDX, IDC_CUS_SER_POST, m_cus_ser_post);
DDX_Control(pDX, IDC_CUS_DEL_POST, m_cus_del_post);
DDX_Control(pDX, IDC_CUS_ADD_POST, m_cus_add_post);
DDX_Control(pDX, IDC_CANCEL_ENT_POST, m_cancel_ent_post);
DDX_Control(pDX, IDC_CHK_AMOUNT, m_chk_amount);
DDX_Control(pDX, IDC_S_ZIPCODE, m_s_zipcode);
DDX_Control(pDX, IDC_S_PHONE, m_s_phone);
DDX_Control(pDX, IDC_S_PER_ID, m_s_per_id);
DDX_Control(pDX, IDC_S_LAST_NAME, m_s_last_name);
DDX_Control(pDX, IDC_S_FIRST_NAME, m_s_first_name);
DDX_Control(pDX, IDC_S_CUS_ID, m_s_cus_id);
DDX_Control(pDX, IDC_S_CITY, m_s_city);
DDX_Control(pDX, IDC_S_ADDRESS, m_s_address);
DDX_Control(pDX, IDC_PHONE, m_phone);
DDX_Control(pDX, IDC_PER_ID, m_per_id);
DDX_Control(pDX, IDC_LAST_NAME, m_last_name);
DDX_Control(pDX, IDC_ZIPCODE, m_zipcode);
DDX_Control(pDX, IDC_ADDRESS, m_address);
DDX_Control(pDX, IDC_FIRST_NAME, m_first_name);
DDX_Control(pDX, IDC_CUS_ID, m_cus_id);
DDX_Control(pDX, IDC_CITY, m_city);
DDX_CBString(pDX, IDC_PORT, m_port);
DDX_Control(pDX, IDC_TYPE_TAB, m_TabCtrl);
DDX_Control(pDX, IDC_MSCOMM1, m_SerialComm);
DDX_Text(pDX, IDC_CUS_ID, m_cusid);
DDV_MinMaxUInt(pDX, m_cusid, 0, 100000);
DDX_Text(pDX, IDC_PER_ID, m_pers_id);
DDV_MaxChars(pDX, m_pers_id, 13);
DDX_Text(pDX, IDC_FIRST_NAME, m_firstname);
DDV_MaxChars(pDX, m_firstname, 20);
DDX_Text(pDX, IDC_LAST_NAME, m_lastname);
DDV_MaxChars(pDX, m_lastname, 20);
DDX_Text(pDX, IDC_ADDRESS, m_addr);
DDV_MaxChars(pDX, m_addr, 100);
DDX_Text(pDX, IDC_CITY, m_city2);
DDV_MaxChars(pDX, m_city2, 20);
DDX_Text(pDX, IDC_ZIPCODE, m_zipcode2);
DDV_MaxChars(pDX, m_zipcode2, 5);
DDX_Text(pDX, IDC_PHONE, m_phone2);
DDV_MaxChars(pDX, m_phone2, 20);
DDX_Text(pDX, IDC_SHOW_AMOUNT, m_amount);
DDV_MinMaxUInt(pDX, m_amount, 0, 10000);
DDX_Text(pDX, IDC_CREDIT_REFILL, m_credit_refill2);
DDV_MinMaxUInt(pDX, m_credit_refill2, 0, 1000);
DDX_Text(pDX, IDC_SHOW_CREDIT, m_show_credit2);
DDV_MinMaxUInt(pDX, m_show_credit2, 0, 10000);
//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CCardManagementDlg, CDialog)
//{{AFX_MSG_MAP(CCardManagementDlg)
ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_BN_CLICKED(IDC_OPEN_PORT, OnOpenPort)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ON_BN_CLICKED(IDC_CLOSE_PORT, OnClosePort)
ON_BN_CLICKED(IDC_CUS_ADD_POST, OnCusAddPost)
ON_BN_CLICKED(IDC_CUS_DEL_POST, OnCusDelPost)
ON_BN_CLICKED(IDC_CHK_AMOUNT, OnChkAmount)
ON_BN_CLICKED(IDC_CANCEL_ENT_POST, OnCancelEntPost)
ON_BN_CLICKED(IDC_CUS_ADD_PRE, OnCusAddPre)
ON_BN_CLICKED(IDC_CUS_DEL_PRE, OnCusDelPre)
ON_BN_CLICKED(IDC_REFILL, OnRefill)
ON_BN_CLICKED(IDC_CHK_CREDIT, OnChkCredit)
ON_BN_CLICKED(IDC_CANCEL_ENT_PRE, OnCancelEntPre)
ON_NOTIFY(TCN_SELCHANGE, IDC_TYPE_TAB, OnSelchangeTypeTab)
//})AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
////
// CCardManagementDlg message handlers

BOOL CCardManagementDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

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

    // TODO: Add extra initialization here
    /// Add
    CLogOn LogonDlg;
    int res_logon = LogonDlg.DoModal();
    if(res_logon == IDOK)
    {
        StationName = LogonDlg.StationName;
        StationID = LogonDlg.StationID;
    }
    else if(res_logon == IDCANCEL)
        CCardManagementDlg::EndDialog(res_logon);
    StationID.MakeLower();
    SetDlgItemText(IDC_PORT_STATUS, "พอร์ตปิด");
    GetDlgItem(IDC_OPEN_PORT)->EnableWindow(TRUE);
    GetDlgItem(IDC_CLOSE_PORT)->EnableWindow(FALSE);
    GetDlgItem(IDC_PORT)->EnableWindow(TRUE);

    //////////////////////////////////////
    ////
    // STabCtrl initialization

    PSTR pszTabItems[] =
    {
        "ผู้ให้บริการบัตรเครดิตเงิน",
        "ผู้ให้บริการบัตรเครดิตเงิน",
        NULL
    };

    TC_ITEM tcItem;

    for(INT i = 0; pszTabItems[i] != NULL; i++)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        tcItem.mask = TCIF_TEXT;
        tcItem.pszText = pszTabItems[i];
        tcItem.cchTextMax = strlen(pszTabItems[i]);
        m_TabCtrl.InsertItem(i,&tcItem);
    }
    Close_Cus_Info();
    GetDlgItem(IDC_CREDIT_REFILL)->EnableWindow(FALSE);
    // attach controls to tabs pages.
    /// PrePaid    /// m_TabCtrl.AttachControlToTab(&, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cus_ser_pre, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cus_add_pre, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cus_del_pre, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_refill, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_chk_credit, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cancel_ent_pre, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_static_pre, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_credit_refill, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_credit_refill, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_show_credit, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_show_credit, PrePaidIndex);
    m_TabCtrl.AttachControlToTab(&m_help_pre, PrePaidIndex);
    ///
    /// PostPaid
    m_TabCtrl.AttachControlToTab(&m_cus_id, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_per_id, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_first_name, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_last_name, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_address, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_city, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_zipcode, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_phone, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_cus_id, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_per_id, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_first_name, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_last_name, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_address, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_city, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_zipcode, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_phone, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cus_add_post, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cus_del_post, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_chk_amount, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cancel_ent_post, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cus_ser_post, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_cus_data_post, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_static_post, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_s_amount, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_show_amount, PostPaidIndex);
    m_TabCtrl.AttachControlToTab(&m_help_post, PostPaidIndex);

    ///
    // initialize tab to first page.
    m_TabCtrl.SetCurSel(PrePaidIndex);

    // end of STabCtrl initialization
    //////////////////////////////////////
    ///

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

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// If you add a minimize button to your dialog, you will need the code
below
// to draw the icon. For MFC applications using the document/view
model,
// this is automatically done for you by the framework.
```

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

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

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

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

```
// The system calls this to obtain the cursor to display while the user
drags
```

```
// the minimized window.
HCURSOR CCardManagementDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}
```

```
void CCardManagementDlg::OnOpenPort()
{
    // TODO: Add your control notification handler code here
    ClearDataCusPost();
    Close_All();
    UpdateData(TRUE);
    m_credit_refill2 = 0;
    m_cusid = 0;
    m_amount = 0;
    m_show_credit2 = 0;
    UpdateData(FALSE);
    UpdateData(TRUE);
    if(strcmp(m_port, "COM1") == 0)
        port = 1;
    else if(strcmp(m_port, "COM2") == 0)
        port = 2;
    else if(strcmp(m_port, "COM3") == 0)
        port = 3;
    else if(strcmp(m_port, "COM4") == 0)
        port = 4;
    m_SerialComm.SetCommPort(port);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

m_SerialComm.SetSettings("57600,N,8,1");
m_SerialComm.SetInputLen(1);
m_SerialComm.SetInputMode(0); // Rx Text
m_SerialComm.SetRThreshold(1); // return RxEvent
m_SerialComm.SetPortOpen(TRUE);
SetDlgItemText(IDC_PORT_STATUS, "พอร์ตเปิด");
GetDlgItem(IDC_OPEN_PORT)->EnableWindow(FALSE);
GetDlgItem(IDC_CLOSE_PORT)->EnableWindow(TRUE);
GetDlgItem(IDC_PORT)->EnableWindow(FALSE);
}

void CCardManagementDlg::OnClosePort()
{
    // TODO: Add your control notification handler code here
    m_SerialComm.SetPortOpen(FALSE);
    SetDlgItemText(IDC_PORT_STATUS, "พอร์ตปิด");
    GetDlgItem(IDC_OPEN_PORT)->EnableWindow(TRUE);
    GetDlgItem(IDC_CLOSE_PORT)->EnableWindow(FALSE);
    GetDlgItem(IDC_PORT)->EnableWindow(TRUE);
    ClearDataCusPost();
    Close_All();
    UpdateData(TRUE);
    m_credit_refill2 = 0;
    m_cusid = 0;
    m_amount = 0;
    m_show_credit2 = 0;
    UpdateData(FALSE);
}

void CCardManagementDlg::OnCancel()
{
    // TODO: Add extra cleanup here
    if(m_SerialComm.GetPortOpen())
        m_SerialComm.SetPortOpen(FALSE);
    CDialog::OnCancel();
}

BEGIN_EVENTSINK_MAP(CCardManagementDlg, CDialog)
    {{{AFX_EVENTSINK_MAP(CCardManagementDlg)
        ON_EVENT(CCardManagementDlg, IDC_MSCOMM1, 1 /* OnComm */,
OnOnCommMscmm1, VTS_NONE)
        }}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()

void CCardManagementDlg::OnOnCommMscmm1()
{
    // TODO: Add your control notification handler code here
    if(m_SerialComm.GetCommEvent() == comEvReceive) {
        VARIANT SData;
        CString TempData;
        SData = m_SerialComm.GetInput();
        TempData = SData.bstrVal;
        StrRx_Buffer += TempData;
        if(!Start_Rx && TempData == '@' )
            Start_Rx = TRUE;
        if(Start_Rx) {
            if(TempData == '#') {
                if(Check_LRC(StrRx_Buffer.Mid(1, 16)))
                {
                    StrRx_Buffer = StrRx_Buffer.Mid(1, 14);
                    if(StartProcess)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            m_SerialComm.SetOutput(COLEVariant("A"));
            StartProcess = FALSE;
            Card_ID = StrRx_Buffer;

            if( m_TabCtrl.m_nPrevSel ==
PrePaidIndex )
            {
                switch(process_pre) {
                    case 1 :
Create_PreTable(); break;
                    case 2 :
Delete_PrePaid(); break;
                    case 3 :
PrePaid_Refill(); break;
                    case 4 : Show_Credit();
break;
                    case 5 :
Cancel_Ent_Pre(); break;
                    default : break;
                }
            }
            else
            {
                switch(process_post) {
                    case 1 :
Create_PostTable(); break;
                    case 2 :
Delete_PostPaid(); break;
                    case 3 : Show_Amount();
break;
                    case 4 :
Cancel_Ent_Post(); break;
                    default : break;
                }
            }
            else
            {
                m_SerialComm.SetOutput(COLEVariant("Z"));
                AfxMessageBox("กรุณาเลือกรูปแบบการให้บริการ!");
            }
            Close_All();
            StrRx_Buffer = _T("");
        }
        Start_Rx = FALSE;
    }
}
}
}

```

```

BOOL CCardManagementDlg::Check_LRC(CString Data)

```

```

{
    char LRC_HIGH, LRC_LOW, temp_data;
    temp_data = Data.GetAt(14);
    if(temp_data < 'A')
        LRC_HIGH = temp_data - '0';
    else
        LRC_HIGH = temp_data - 'A' +10;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp_data = Data.GetAt(15);
if(temp_data < 'A')
    LRC_LOW = temp_data - '0';
else
    LRC_LOW = temp_data - 'A' +10;
temp_data = 0;
for(int i = 0; i < 14; i++)
    temp_data ^= Data.GetAt( i );
if( (temp_data >> 4) == LRC_HIGH && (temp_data & 0x0f ) ==
LRC_LOW)
    return TRUE;
return FALSE;
}

```

```

void CCardManagementDlg::OnCusAddPost()
{
    // TODO: Add your control notification handler code here
    process_post = 1;
    UpdateData(TRUE);
    m_cusid = 0;
    m_amount = 0;
    UpdateData(FALSE);
    StartProcess = TRUE;
    ClearDataCusPost();
    Open_Cus_Info();
}

```

```

void CCardManagementDlg::OnCusDelPost()
{
    // TODO: Add your control notification handler code here
    process_post = 2;
    UpdateData(TRUE);
    m_cusid = 0;
    m_amount = 0;
    UpdateData(FALSE);
    StartProcess = TRUE;
    ClearDataCusPost();
    Close_Cus_Info();
}

```

```

void CCardManagementDlg::OnChkAmount()
{
    // TODO: Add your control notification handler code here
    process_post = 3;
    UpdateData(TRUE);
    m_cusid = 0;
    m_amount = 0;
    UpdateData(FALSE);
    StartProcess = TRUE;
    ClearDataCusPost();
    Close_Cus_Info();
}

```

```

void CCardManagementDlg::OnCancelEntPost()
{
    // TODO: Add your control notification handler code here
    process_post = 4;
    UpdateData(TRUE);
    m_cusid = 0;
    m_amount = 0;
    UpdateData(FALSE);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        StartProcess = TRUE;
        ClearDataCusPost();
        Close_Cus_Info();
    }

void CCardManagementDlg::OnCusAddPre()
{
    // TODO: Add your control notification handler code here
    process_pre = 1;
    StartProcess = TRUE;
    GetDlgItem(IDC_CREDIT_REFILL)->EnableWindow(FALSE);
    UpdateData(TRUE);
    m_credit_refill2 = 0;
    m_show_credit2 = 0;
    UpdateData(FALSE);
}

void CCardManagementDlg::OnCusDelPre()
{
    // TODO: Add your control notification handler code here
    process_pre = 2;
    StartProcess = TRUE;
    GetDlgItem(IDC_CREDIT_REFILL)->EnableWindow(FALSE);
    UpdateData(TRUE);
    m_credit_refill2 = 0;
    m_show_credit2 = 0;
    UpdateData(FALSE);
}

void CCardManagementDlg::OnRefill()
{
    // TODO: Add your control notification handler code here
    process_pre = 3;
    StartProcess = TRUE;
    GetDlgItem(IDC_CREDIT_REFILL)->EnableWindow(TRUE);
    UpdateData(TRUE);
    m_credit_refill2 = 0;
    m_show_credit2 = 0;
    UpdateData(FALSE);
}

void CCardManagementDlg::OnChkCredit()
{
    // TODO: Add your control notification handler code here
    process_pre = 4;
    StartProcess = TRUE;
    GetDlgItem(IDC_CREDIT_REFILL)->EnableWindow(FALSE);
    UpdateData(TRUE);
    m_credit_refill2 = 0;
    m_show_credit2 = 0;
    UpdateData(FALSE);
}

void CCardManagementDlg::OnCancelEntPre()
{
    // TODO: Add your control notification handler code here
    process_pre = 5;
    StartProcess = TRUE;
    GetDlgItem(IDC_CREDIT_REFILL)->EnableWindow(FALSE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        UpdateData(TRUE);
        m_credit_refill2 = 0;
        m_show_credit2 = 0;
        UpdateData(FALSE);
    }

void CCardManagementDlg::OnSelchangeTypeTab(NMHDR* pNMHDR, LRESULT*
pResult)
{
    // TODO: Add your control notification handler code here
    ClearDataCusPost();
    Close_All();
    UpdateData(TRUE);
    m_credit_refill2 = 0;
    m_cusid = 0;
    m_amount = 0;
    m_show_credit2 = 0;
    UpdateData(FALSE);
    *pResult = 0;
}

void CCardManagementDlg::Close_Cus_Info()
{
    GetDlgItem(IDC_PER_ID)->EnableWindow(FALSE);
    GetDlgItem(IDC_FIRST_NAME)->EnableWindow(FALSE);
    GetDlgItem(IDC_LAST_NAME)->EnableWindow(FALSE);
    GetDlgItem(IDC_ADDRESS)->EnableWindow(FALSE);
    GetDlgItem(IDC_CITY)->EnableWindow(FALSE);
    GetDlgItem(IDC_ZIPCODE)->EnableWindow(FALSE);
    GetDlgItem(IDC_PHONE)->EnableWindow(FALSE);
}

void CCardManagementDlg::Open_Cus_Info()
{
    GetDlgItem(IDC_PER_ID)->EnableWindow(TRUE);
    GetDlgItem(IDC_FIRST_NAME)->EnableWindow(TRUE);
    GetDlgItem(IDC_LAST_NAME)->EnableWindow(TRUE);
    GetDlgItem(IDC_ADDRESS)->EnableWindow(TRUE);
    GetDlgItem(IDC_CITY)->EnableWindow(TRUE);
    GetDlgItem(IDC_ZIPCODE)->EnableWindow(TRUE);
    GetDlgItem(IDC_PHONE)->EnableWindow(TRUE);
}

void CCardManagementDlg::Close_All()
{
    CButton* TempBtn;
    StartProcess = FALSE;
    TempBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_CUS_ADD_PRE));
    TempBtn->SetCheck(FALSE);
    TempBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_CUS_DEL_PRE));
    TempBtn->SetCheck(FALSE);
    TempBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_REFILL));
    TempBtn->SetCheck(FALSE);
    TempBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_CHK_CREDIT));
    TempBtn->SetCheck(FALSE);
    TempBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_CANCEL_ENT_PRE));
    TempBtn->SetCheck(FALSE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        TempBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_CUS_ADD_POST));
        TempBtn->SetCheck(FALSE);
        TempBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_CUS_DEL_POST));
        TempBtn->SetCheck(FALSE);
        TempBtn = reinterpret_cast<CButton*> (GetDlgItem(IDC_CHK_AMOUNT));
        TempBtn->SetCheck(FALSE);
        TempBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_CANCEL_ENT_POST));
        TempBtn->SetCheck(FALSE);
        GetDlgItem(IDC_CREDIT_REFILL)->EnableWindow(FALSE);
        Close_Cus_Info();
}

```

```

void CCardManagementDlg::Create_PostTable()
{
    CString Cmd;
    Cmd = "CREATE TABLE `id_";
    Cmd = Cmd + Card_ID;
    Cmd = Cmd + "` ( `order` int(5) unsigned NOT NULL
auto_increment,";
    Cmd = Cmd + "`source` char(25) default NULL, `in_time` datetime
default NULL,";
    Cmd = Cmd + "`destination` char(25) default NULL, `out_time`
datetime default NULL,";
    Cmd = Cmd + "`price` int(3) default NULL, PRIMARY KEY
(`order`))";
    Cmd = Cmd + " ENGINE=MyISAM DEFAULT CHARSET=tis620 ";
    PostpaidTran.MySqlCommand(Cmd);
    PostpaidTran.Close();
    Cmd = "insert into `skytrain`.`type_of_card` ( `Index`, `Card_ID`,
`Type` ) values ( ' ', ' ', ' '";
    Cmd = Cmd + Card_ID;
    Cmd = Cmd + " ', 'postpaid' );";
    CardType.MySqlCommand(Cmd);
    CardType.Close();
    Cmd = "insert into `skytrain`.`postpaid_amount` ( `Card_ID`,
`amount` ) values ( ' ', ' '";
    Cmd = Cmd + Card_ID;
    Cmd = Cmd + " ', '0' );";
    postpaidamount.MySqlCommand(Cmd);
    postpaidamount.Close();

    Cmd = "insert into `skytrain`.`postpaid_info` ( `Customer_ID`,
`Card_ID`, `FirstName`, `LastName`,";
    Cmd = Cmd+ "`Personal_ID`, `Address`, `City`, `Zip-code`,
`Telephone` ) values ( ' ', ' '";
    UpdateData(TRUE);
    Cmd = Cmd + Card_ID;
    Cmd = Cmd + m_firstname;
    Cmd = Cmd + m_lastname;
    Cmd = Cmd + m_pers_id;
    Cmd = Cmd + m_addr;
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ', ' ";
    Cmd = Cmd + " ' );";
    UpdateData(FALSE);
    postpaidinfo.MySqlCommand(Cmd);
    postpaidinfo.Close();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Close_All();
    }

void CCardManagementDlg::Create_PreTable()
{
    CString Cmd;
    Cmd = "CREATE TABLE `id_";
    Cmd = Cmd + Card_ID;
    Cmd = Cmd + "` ( `order` int(5) unsigned NOT NULL
auto_increment,";
    Cmd = Cmd + "`source` char(25) default NULL, `in_time` datetime
default NULL, ";
    Cmd = Cmd + "`destination` char(25) default NULL, `out_time`
datetime default NULL, ";
    Cmd = Cmd + "`price` int(3) default NULL, PRIMARY KEY
(`order`)";
    Cmd = Cmd + " ENGINE=MyISAM DEFAULT CHARSET=tis620 ";
    PrepaidTran.MySQLCommand(Cmd);
    PrepaidTran.Close();
    Cmd = "insert into `skytrain`.`type_of_card` ( `Index`, `Card_ID`,
`Type` ) values ( '', ";
    Cmd = Cmd + Card_ID;
    Cmd = Cmd + "', 'prepaid' );";
    CardType.MySQLCommand(Cmd);
    CardType.Close();
    Cmd = "insert into `skytrain`.`prepaid_credit` ( `Card_ID`,
`credit` ) values ( ";
    Cmd = Cmd + Card_ID;
    Cmd = Cmd + "', '150' );";
    prepaidcredit.MySQLCommand(Cmd);
    prepaidcredit.Close();
}

void CCardManagementDlg::Delete_PrePaid()
{
    CString Cmd;
    Cmd = "drop table `prepaid_transit`.`id_";
    Cmd = Cmd + Card_ID; Cmd = Cmd + "`";
    PrepaidTran.MySQLCommand(Cmd);
    PrepaidTran.Close();
    Cmd = "delete from `skytrain`.`type_of_card` where `Card_ID`='";
    Cmd = Cmd + Card_ID; Cmd = Cmd + "'";
    CardType.MySQLCommand(Cmd);
    CardType.Close();
    Cmd = "delete from `skytrain`.`prepaid_credit` where `Card_ID`='";
    Cmd = Cmd + Card_ID; Cmd = Cmd + "'";
    prepaidcredit.MySQLCommand(Cmd);
    prepaidcredit.Close();
}

void CCardManagementDlg::Delete_PostPaid()
{
    CString Cmd;
    Cmd = "drop table `postpaid_transit`.`id_";
    Cmd = Cmd + Card_ID; Cmd = Cmd + "`";
    PostpaidTran.MySQLCommand(Cmd);
    PostpaidTran.Close();
    Cmd = "delete from `skytrain`.`type_of_card` where `Card_ID`='";
    Cmd = Cmd + Card_ID; Cmd = Cmd + "'";
    CardType.MySQLCommand(Cmd);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CardType.Close();
        Cmd = "delete from `skytrain`.`postpaid_amount` where
`Card_ID`='";
        Cmd = Cmd + Card_ID; Cmd = Cmd + "';";
        postpaidamount.MySqlCommand(Cmd);
        postpaidamount.Close();
        Cmd = "delete from `skytrain`.`postpaid_info` where `Card_ID`='";
        Cmd = Cmd + Card_ID; Cmd = Cmd + "';";
        postpaidinfo.MySqlCommand(Cmd);
        postpaidinfo.Close();
    }

```

```

void CCardManagementDlg::PrePaid_Refill()
{
    CString Cmd, m_TempCredit;

    UpdateData(TRUE);
    m_show_credit2 = m_credit_refill2;
    m_TempCredit.Format("%u",m_credit_refill2);
    UpdateData(FALSE);

    Cmd = "update `skytrain`.`prepaid_credit` set `credit`= `credit`
+ ";
    Cmd = Cmd + m_TempCredit;
    Cmd = Cmd + "` where `Card_ID`='";
    Cmd = Cmd + Card_ID;
    Cmd = Cmd + "';";

    prepaidcredit.MySqlCommand(Cmd);
    prepaidcredit.Close();

    if (prepaidcredit.Open() == S_OK) {
        if( prepaidcredit.MoveNext() == S_OK) {
            prepaidcredit.Close();
            prepaidcredit.SearchCredit(Card_ID);
            UpdateData(TRUE);

            m_show_credit2 = prepaidcredit.m_credit;
            UpdateData(FALSE);
            prepaidcredit.Close();
        }
    }
}

```

```

void CCardManagementDlg::Show_Credit()
{
    if (prepaidcredit.Open() == S_OK)
    {
        if( prepaidcredit.MoveNext() == S_OK)
        {
            prepaidcredit.Close();
            if(prepaidcredit.SearchCredit(Card_ID) == S_OK)
            {
                m_show_credit2 = prepaidcredit.m_credit;
                UpdateData(FALSE);
            }
            prepaidcredit.Close();
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CCardManagementDlg::Show_Amount()
{
    if (postpaidamount.Open() == S_OK && postpaidinfo.Open() == S_OK)
    {
        if( postpaidamount.MoveNext() == S_OK &&
postpaidinfo.MoveNext() == S_OK)
        {
            postpaidamount.Close();
            postpaidinfo.Close();
            if(postpaidamount.SearchAmount(Card_ID) == S_OK
&& postpaidinfo.SearchCusID(Card_ID) == S_OK)
            {
                m_amount = postpaidamount.m_amount;
                m_cusid = postpaidinfo.m_CustomerID;
                m_pers_id = postpaidinfo.m_PersonalID;
                m_firstname = postpaidinfo.m_FirstName;
                m_lastname = postpaidinfo.m_LastName;
                m_addr = postpaidinfo.m_Address;
                m_city2 = postpaidinfo.m_City;
                m_zipcode2 = postpaidinfo.m_Zipcode;
                m_phone2 = postpaidinfo.m_Telephone;
                UpdateData(FALSE);
            }
            postpaidamount.Close();
            postpaidinfo.Close();
        }
    }
}

void CCardManagementDlg::Cancel_Ent_Pre()
{
    BOOL Cancel_Transit;
    unsigned long order;

    PrepaidTran.Open(Card_ID);
    PrepaidTran.MoveLast();
    if(StationName.Collate(PrepaidTran.m_source) == 0)
    {
        if(strlen(PrepaidTran.m_destination) == 0)
        {
            Cancel_Transit = TRUE;
            order = PrepaidTran.m_order;
        }
        else
            Cancel_Transit = FALSE;
    }
    PrepaidTran.Close();
    if(Cancel_Transit )
    {
        CString Cmd;
        CString TmpOrder;

        TmpOrder.Format("%d", order);
        Cmd = "update `prepaid_transit`.`id_";
        Cmd = Cmd + Card_ID;
        Cmd = Cmd + "` set `destination`=";
        Cmd = Cmd + StationName;
        Cmd = Cmd + "', `out_time` = now() , `price`='0' where
`order` = ";
        Cmd = Cmd + TmpOrder;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Cmd = Cmd + "'";
        PostpaidTran.MySqlCommand(Cmd);
        PrepaidTran.Close();
        MessageBox("การยกเลิกข้อผิดพลาดจากเข้าประตูเสร็จสิ้น.");
    }
    else
        MessageBox("ไม่พบข้อผิดพลาดจากเข้าประตู.");
}

void CCardManagementDlg::Cancel_Ent_Post()
{
    BOOL Cancel_Transit;
    unsigned long order;

    PostpaidTran.Open(Card_ID);
    PostpaidTran.MoveLast();
    if(StationName.Collate(PostpaidTran.m_source) == 0)
    {
        if(strlen(PostpaidTran.m_destination) == 0)
        {
            Cancel_Transit = TRUE;
            order = PostpaidTran.m_order;
        }
        else
            Cancel_Transit = FALSE;
    }
    PostpaidTran.Close();
    if(Cancel_Transit)
    {
        CString Cmd;
        CString TmpOrder;

        TmpOrder.Format("%d",order);
        Cmd = "update `postpaid_transit`.`id_";
        Cmd = Cmd + Card_ID;
        Cmd = Cmd + "` set `destination`= '";
        Cmd = Cmd + StationName;
        Cmd = Cmd + "', `out_time` = now(), `price`='0' where";
        Cmd = Cmd + TmpOrder;
        Cmd = Cmd + "'";
        PostpaidTran.MySqlCommand(Cmd);
        PostpaidTran.Close();
        MessageBox("การยกเลิกข้อผิดพลาดจากเข้าประตูเสร็จสิ้น.");
    }
    else
        MessageBox("ไม่พบข้อผิดพลาดจากเข้าประตู.");
}

void CCardManagementDlg::ClearDataCusPost()
{
    SetDlgItemText(IDC_PER_ID, _T(""));
    SetDlgItemText(IDC_FIRST_NAME, _T(""));
    SetDlgItemText(IDC_LAST_NAME, _T(""));
    SetDlgItemText(IDC_ADDRESS, _T(""));
    SetDlgItemText(IDC_CITY, _T(""));
    SetDlgItemText(IDC_ZIPCODE, _T(""));
    SetDlgItemText(IDC_PHONE, _T(""));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// price.H : Declaration of the Cprice class

#ifndef __PRICE_H_
#define __PRICE_H_

class CpriceAccessor
{
public:
    TCHAR m_StationID[3];
    TCHAR m_Station[26];
    LONG m_Price;

BEGIN_COLUMN_MAP(CpriceAccessor)
    COLUMN_ENTRY(1, m_StationID)
    COLUMN_ENTRY(2, m_Station)
    COLUMN_ENTRY(3, m_Price)
END_COLUMN_MAP()

DEFINE_COMMAND(CpriceAccessor, _T(""))

    // You may wish to call this function if you are inserting a
    // record and wish to
    // initialize all the fields, if you are not going to explicitly
    // set all of them.
    void ClearRecord()
    {
        memset(this, 0, sizeof(*this));
    }
};

class Cprice : public CCommand<CAccessor<CpriceAccessor>>
{
public:
    HRESULT Open()
    {
        HRESULT hr;

        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;

        return OpenRowset();
    }
    HRESULT OpenDataSource()
    {
        HRESULT hr;
        CDataSource db;
        CDBPropSet dbinit(DBPROPSET_DBINIT);

        dbinit.AddProperty(DBPROP_AUTH_PASSWORD, OLESTR("std3t"));
        dbinit.AddProperty(DBPROP_AUTH_USERID, OLESTR("project"));
        dbinit.AddProperty(DBPROP_INIT_DATASOURCE,
        OLESTR("skytrain"));
        dbinit.AddProperty(DBPROP_INIT_PROMPT, (short)4);
        dbinit.AddProperty(DBPROP_INIT_LCID, (long)1054);
        hr = db.Open(_T("MSDASQL"), &dbinit);
        if (FAILED(hr))
            return hr;
        m_session.Close(); ///
        return m_session.Open(db);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HRESULT OpenRowset ()
{
    return
    >::Open(m_session);
}
HRESULT SearchPrice(CString Command)  ///All Created
{
    HRESULT          hr;
    hr = OpenDataSource();

    // Set properties for open
    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);
    propset.AddProperty(DBPROP_UPDATABILITY, DBPROPVAL_UP_CHANGE
| DBPROPVAL_UP_INSERT | DBPROPVAL_UP_DELETE);
    propset.AddProperty(DBPROP_CANSCROLLBACKWARDS,true); ///
    propset.AddProperty(DBPROP_IMMOBILEROWS,true); ///
    hr = CCommand<CAccessor<CpriceAccessor> >::Open(m_session,
Command, &propset);
    if (FAILED(hr))
        return hr;
    return MoveNext();
}
CSession m_session;
};
#endif // __PRICE_H__

```

**\*หมายเหตุ**

logondb.H, CardType.H, prepaidcredit.H, PrepaidTransit.H, postpaidamount.H, PostpaidTransit.H, LogOn.h และ LogOn.cpp ถัดจากในโปรแกรมการจัดการระบบบัตรโดยสารรถไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// DoorControlDlg.h : header file
//
//{{AFX_INCLUDES()
#include "serialcomm.h"
//}}AFX_INCLUDES
#if
!defined(AFX_DOORCONTROLDLG_H__C371737F_B017_4160_9769_B110A16C9F3B__INC
LUDED_)
#define
AFX_DOORCONTROLDLG_H__C371737F_B017_4160_9769_B110A16C9F3B__INCLUDED_

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

#include "postpaidamount.H"
#include "prepaidcredit.H"
#include "CardType.H"
#include "PrepaidTransit.H"
#include "PostpaidTransit.H"
#include "price.H"

#define comEvReceive 2
#define PrePaidIndex 0
#define PostPaidIndex 1
#define MaxPrice 35
////////////////////////////////////
////
// CDoorControlDlg dialog

class CDoorControlDlg : public CDialog
{
// Construction
public:
    CDoorControlDlg(CWnd* pParent = NULL);    // standard constructor
    Cprice m_price;
    Cpostpaidamount m_postpaidamount;
    Cprepaidcredit m_prepaidcredit;
    CCardType m_CardType;
    CPrepaidTransit m_PrepaidTran;
    CPostpaidTransit m_PostpaidTran;
// Dialog Data
//{{AFX_DATA(CDoorControlDlg)
enum { IDD = IDD_DOORCONTROL_DIALOG };
    CMSComm m_SerialComm;
    CString m_port;
//}}AFX_DATA

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

// Implementation
protected:
    HICON m_hIcon;
    CWinThread* SThread; ///
// Generated message map functions
//{{AFX_MSG(CDoorControlDlg)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

virtual BOOL OnInitDialog();
afx_msg void OnPaint();
afx_msg HCURSOR OnQueryDragIcon();
afx_msg void OnOnCommMscmm1();
afx_msg void OnOpenPort();
afx_msg void OnClosePort();
afx_msg void OnAddress1();
afx_msg void OnAddress2();
afx_msg void OnAddress3();
afx_msg void OnAddress4();
afx_msg void OnAddress6();
afx_msg void OnAddress7();
afx_msg void OnAddress8();
afx_msg void OnAddress9();
virtual void OnCancel();
DECLARE_EVENTSINK_MAP()
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
public:
void ChkDB(CString Card_ID, char TempAddress);
void EnBtn();
void DisBtn();
static UINT SerialThread(LPVOID Param);
unsigned int port;
CString StationID;
CString StationName;
CString StrRx_Buffer;
BOOL Next_Address;
BOOL ActivateRx;
BOOL AddresStatus[10];
private:
void UnRdBtn();
BOOL Check_LRC(CString Data);
BOOL BtnOn;
};

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

#endif
#ifdef AFX_DOORCONTROLDLG_H_C371737F_B017_4160_9769_B110A16C9F3B__INC
LUDED_

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// DoorControlDlg.cpp : implementation file
//

#include "stdafx.h"
#include "DoorControl.h"
#include "DoorControlDlg.h"

#include "LogOn.h"

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

////////////////////////////////////
// CDoorControlDlg dialog

CDoorControlDlg::CDoorControlDlg(CWnd* pParent /*=NULL*/)
: CDialog(CDoorControlDlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CDoorControlDlg)
    m_port = _T("COM1");
   //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in
Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    port = 1;
    StrRx_Buffer = _T("");
    ActivateRx = FALSE;
    Next_Address = FALSE;
    BtnOn = TRUE;
}

void CDoorControlDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CDoorControlDlg)
    DDX_Control(pDX, IDC_MSCOMM1, m_SerialComm);
    DDX_CBString(pDX, IDC_PORT, m_port);
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CDoorControlDlg, CDialog)
   //{{AFX_MSG_MAP(CDoorControlDlg)
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_OPEN_PORT, OnOpenPort)
    ON_BN_CLICKED(IDC_CLOSE_PORT, OnClosePort)
    ON_BN_CLICKED(IDC_ADDRESS1, OnAddress1)
    ON_BN_CLICKED(IDC_ADDRESS2, OnAddress2)
    ON_BN_CLICKED(IDC_ADDRESS3, OnAddress3)
    ON_BN_CLICKED(IDC_ADDRESS4, OnAddress4)
    ON_BN_CLICKED(IDC_ADDRESS6, OnAddress6)
    ON_BN_CLICKED(IDC_ADDRESS7, OnAddress7)
    ON_BN_CLICKED(IDC_ADDRESS8, OnAddress8)
    ON_BN_CLICKED(IDC_ADDRESS9, OnAddress9)
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////
////
// CDoorControlDlg message handlers

BOOL CDoorControlDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

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

    // TODO: Add extra initialization here
    /// Add เสง
    for(int Count = 0; Count < 10; Count++)
        AddressStatus[Count] = FALSE;

    CLogon LogonDlg;
    int res_logon = LogonDlg.DoModal();
    if(res_logon == IDOK)
    {
        StationID = LogonDlg.StationID;
        StationName = LogonDlg.StationName;
    }
    else if(res_logon == IDCANCEL)
        CDoorControlDlg::EndDialog(res_logon);

    SetDlgItemText(IDC_STATUS, "พอหัดปิด");
    GetDlgItem(IDC_OPEN_PORT)->EnableWindow(TRUE);
    GetDlgItem(IDC_CLOSE_PORT)->EnableWindow(FALSE);
    GetDlgItem(IDC_PORT)->EnableWindow(TRUE);
    UnRdBtn();
    SThread = (CWinThread*)AfxBeginThread(SerialThread,
        this, THREAD_PRIORITY_NORMAL, 0,
        CREATE_SUSPENDED, NULL); ///

    SThread->m_bAutoDelete=FALSE; ///
    return TRUE; // return TRUE unless you set the focus to a
control
}

// If you add a minimize button to your dialog, you will need the code
// below
// to draw the icon. For MFC applications using the document/view
// model,
// this is automatically done for you by the framework.

```

```

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

        SendMessage(WM_ICONERASEBKGD, (LPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อผู้ญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

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

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

BEGIN_EVENTSINK_MAP(CDoorControlDlg, CDialog)
    //{{AFX_EVENTSINK_MAP(CDoorControlDlg)
    ON_EVENT(CDoorControlDlg, IDC_MSCOMM1, 1, /* OnComm */,
    OnOnCommMscmm1, VTS_NONE)
    //}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()

void CDoorControlDlg::OnOnCommMscmm1()
{
    // TODO: Add your control notification handler code here
    if(m_SerialComm.GetCommEvent() == comEvReceive)
    {
        VARIANT SData;
        CString TempData;
        CString TempChkID;
        CString Command;
        char TempAddress;
        char ChkStrart;

        SData = m_SerialComm.GetInput();
        TempData = SData.bstrVal;
        StrRx_Buffer += TempData;

        if(TempData.Collate("&") == 0 && ActivateRx)
            StrRx_Buffer = "&";
        if(TempData.Collate("$") == 0)
        {
            ActivateRx = FALSE;

            ChkStrart = StrRx_Buffer.GetAt(0);
            TempAddress = StrRx_Buffer.GetAt(1);
            if(ChkStrart == '&')
            {
                if(StrRx_Buffer.Right(7).Collate(",ID>NO$") !=
0)
                {
                    TempChkID = StrRx_Buffer.Right(17);
                    if(Check_LRC(TempChkID))
                    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        DisBtn();
        ActivateRx = TRUE;
        SThread->ResumeThread(); ///
    }

void CDoorControlDlg::OnClosePort()
{
    // TODO: Add your control notification handler code here
    ActivateRx = FALSE;
    BtnOn = TRUE;
    EnBtn();
    SThread->SuspendThread();
    m_SerialComm.SetPortOpen(FALSE);
    SetDlgItemText(IDC_STATUS, "พอร์ตปิด");
    GetDlgItem(IDC_OPEN_PORT)->EnableWindow(TRUE);
    GetDlgItem(IDC_CLOSE_PORT)->EnableWindow(FALSE);
    GetDlgItem(IDC_PORT)->EnableWindow(TRUE);
}

UINT CDoorControlDlg::SerialThread(LPVOID Param)
{
    CDoorControlDlg* m_DoorControlDlg = (CDoorControlDlg*) Param;
    CString Command;

    while(TRUE)
    {
        int Count = 1;
        do
        {
            if(m_DoorControlDlg->AddressStatus[Count])
            {
                Sleep(10);
                Command.Format("@@%1d3#", Count);
                m_DoorControlDlg->
                m_SerialComm.SetOutput(COleVariant(Command));
                m_DoorControlDlg->ActivateRx = TRUE;
            }
            Sleep(2);
            Count++;
            if(Count == 5)
                Count = 6;
        } while(Count > 0 && Count < 10 );
    }
    return 0;
}

void CDoorControlDlg::UnRdBtn()
{
    CButton* AddBtn;
    for(int i = 0; i < 9; i++)
    {
        if(i != 4)
        {
            AddBtn = reinterpret_cast<CButton*>
            (GetDlgItem(IDC_ADDRESS1 + i));
            AddBtn->SetButtonStyle(BS_AUTOCHECKBOX, TRUE );
            AddBtn->SetCheck(0);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CDoorControlDlg::OnAddress1()
{
    // TODO: Add your control notification handler code here
    if(BtnOn)
    {
        CButton* AddBtn;
        AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS1));
        AddrStatus[1] = AddBtn->GetCheck();
    }
}

void CDoorControlDlg::OnAddress2()
{
    // TODO: Add your control notification handler code here
    if(BtnOn)
    {
        CButton* AddBtn;
        AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS2));
        AddrStatus[2] = AddBtn->GetCheck();
    }
}

void CDoorControlDlg::OnAddress3()
{
    // TODO: Add your control notification handler code here
    if(BtnOn)
    {
        CButton* AddBtn;
        AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS3));
        AddrStatus[3] = AddBtn->GetCheck();
    }
}

void CDoorControlDlg::OnAddress4()
{
    // TODO: Add your control notification handler code here
    if(BtnOn)
    {
        CButton* AddBtn;
        AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS4));
        AddrStatus[4] = AddBtn->GetCheck();
    }
}

void CDoorControlDlg::OnAddress6()
{
    // TODO: Add your control notification handler code here
    if(BtnOn)
    {
        CButton* AddBtn;
        AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS6));
        AddrStatus[6] = AddBtn->GetCheck();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CDoorControlDlg::OnAddress7()
{
    // TODO: Add your control notification handler code here
    if(BtnOn)
    {
        CButton* AddBtn;
        AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS7));
        AddBtn->GetCheck();
    }
}

void CDoorControlDlg::OnAddress8()
{
    // TODO: Add your control notification handler code here
    if(BtnOn)
    {
        CButton* AddBtn;
        AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS8));
        AddBtn->GetCheck();
    }
}

void CDoorControlDlg::OnAddress9()
{
    // TODO: Add your control notification handler code here
    if(BtnOn)
    {
        CButton* AddBtn;
        AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS9));
        AddBtn->GetCheck();
    }
}

void CDoorControlDlg::DisBtn()
{
    CButton* AddBtn;
    for(int i = 0; i < 9; i++)
    {
        if(i != 4)
        {
            AddBtn = reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS1 + i));
            AddBtn->SetButtonStyle(BS_3STATE, TRUE );
            if(AddBtn->GetCheck())
                AddBtn->SetCheck(2);
            else
                AddBtn->SetCheck(0);
        }
    }
}

void CDoorControlDlg::EnBtn()
{
    CButton* AddBtn;
    for(int i = 0; i < 9; i++)
    {
        if(i != 4)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        AddBtn          =          reinterpret_cast<CButton*>
(GetDlgItem(IDC_ADDRESS1 + i));
        AddBtn->SetButtonStyle(BS_AUTOCHECKBOX, TRUE );
        AddBtn->SetCheck(AddressStatus[1 + i]);
    }
}
}

void CDoorControlDlg::OnCancel()
{
    // TODO: Add extra cleanup here
    if(m_SerialComm.GetPortOpen())
    {
        SThread->SuspendThread();
        m_SerialComm.SetPortOpen(FALSE);
    }
    CDialog::OnCancel();
}

void CDoorControlDlg::ChkDB(CString Card_ID, char TempAddress)
{
    CString Card_Type, Command = _T("");

    m_CardType.SearchCardType(Card_ID);
    Card_Type = m_CardType.m_Type;
    m_CardType.Close();
    if(Card_Type.Collate("prepaid") == 0)
    {
        if(TempAddress > '0' && TempAddress < '5')
        {
            m_prepaidcredit.SearchCredit(Card_ID);
            if(m_prepaidcredit.m_credit >= MaxPrice)
            {
                Command = "insert into prepaid_transit.id_" +
Card_ID;
                Command += " ( source, in_time, destination,
out_time, price ) values ('";
                Command += StationName;
                Command += "', now(), '', '', '');"
                m_PrepaidTran.Open(Card_ID);
                m_PrepaidTran.MoveLast();
                if(strlen(m_PrepaidTran.m_source) != 0 &&
strlen(m_PrepaidTran.m_destination) == 0)
                {
                    m_PrepaidTran.Close();
                    Command.Format("@@%c7#", TempAddress);
                }
                else
                {
                    m_PrepaidTran.Close();
                    m_PrepaidTran.MySQLCommand(Command);
                    Command.Format("@@%c4%4d#", TempAddress,
m_prepaidcredit.m_credit);
                    m_PrepaidTran.Close();
                }
            }
        }
        else
            Command.Format("@@%c7#", TempAddress);
        m_prepaidcredit.Close();
    }
    else
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    CString TempSource = _T("");
    CString station_price;
    CString Order;
    m_PrepaidTran.Open(Card_ID);
    if(m_PrepaidTran.MoveLast() == S_OK)
    {
        if(strlen(m_PrepaidTran.m_source) != 0 &&
strlen(m_PrepaidTran.m_destination) != 0)
        {
            m_PrepaidTran.Close();
            Command.Format("@@%c7#", TempAddress);
        }
        else
        {
            Order.Format("%d", m_PrepaidTran.m_order);
            TempSource = m_PrepaidTran.m_source;
            m_PrepaidTran.Close();
            TempSource = TempSource.Right(3);
            TempSource = TempSource.Left(2);
            TempSource.MakeLower();
            Command = "select * from price_" +
TempSource;
            Command += " where StationID = ";
            Command += StationID;
            Command += " ";
            if(m_price.SearchPrice(Command) == S_OK)
                station_price.Format("%d",m_price.m_Price);
            m_price.Close();
            Command = "update skytrain.prepaid_credit
set credit = `credit` - '";
            Command += station_price;
            Command += "' where Card_ID = ";
            Command += Card_ID;
            Command += " '";
            m_prepaidcredit.MySqlCommand(Command);
            m_prepaidcredit.Close();
            Command = "update prepaid_transit.id_";
            Command += Card_ID;
            Command += " set `destination` = ";
            Command += StationName;
            Command += " , `out_time` = now(),
`price` = '";
            Command += station_price;
            Command += "' where `order` = ";
            Command += Order;
            Command += " '";
            m_PrepaidTran.MySqlCommand(Command );
            m_prepaidcredit.SearchCredit(Card_ID);
            Command.Format("@@%c4%4d#", TempAddress,
m_prepaidcredit.m_credit);
            m_prepaidcredit.Close();
        }
    }
    m_PrepaidTran.Close();
}
m_SerialComm.SetOutput(COleVariant(Command));
}
else if(Card_Type.Collate("postpaid") == 0)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(TempAddress > '0' && TempAddress < '5')
{
    short cur_month, prev_mount;
    Command = "insert into postpaid_transit.id_" +
Card_ID;
    Command += " ( source, in_time, destination, out_time,
price ) values ('";
    Command += StationName;
    Command += "', now(), '', '', '');"
    m_PostpaidTran.Open(Card_ID);
    m_PostpaidTran.MoveLast();
    if(strlen(m_PostpaidTran.m_source) != 0 &&
strlen(m_PostpaidTran.m_destination) == 0)
    {
        m_PostpaidTran.Close();
        Command.Format("@@%c7#", TempAddress);
    }
    else
    {
        prev_mount = m_PostpaidTran.m_intime.month;
        m_PostpaidTran.Close();
        m_PostpaidTran.MySQLCommand(Command);
        m_PostpaidTran.Close();
        m_PostpaidTran.Open(Card_ID);
        m_PostpaidTran.MoveLast();
        cur_month = m_PostpaidTran.m_intime.month;
        m_PostpaidTran.Close();
        if(cur_month != prev_mount)
        {
            CString Cmd;
            Cmd = "update skytrain.postpaid_amount set
amount='0'";
            Cmd += " where Card_ID ='";
            Cmd += Card_ID;
            Cmd += "';";
            m_postpaidamount.MySQLCommand(Cmd);
            m_postpaidamount.Close();
        }
        m_postpaidamount.SearchAmount(Card_ID);
        Command.Format("@@%c5%4d#", TempAddress,
m_postpaidamount.m_amount);
        m_postpaidamount.Close();
    }
}
else
{
    CString TempSource = _T("");
    CString station_price;
    CString Order;
    m_PostpaidTran.Open(Card_ID);
    if(m_PostpaidTran.MoveLast() == S_OK)
    {
        if(strlen(m_PostpaidTran.m_source) != 0 &&
strlen(m_PostpaidTran.m_destination) != 0)
        {
            m_PostpaidTran.Close();
            Command.Format("@@%c7#", TempAddress);
        }
        else
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Order.Format("%d",
m_PostpaidTran.m_order);
        TempSource = m_PostpaidTran.m_source;
        m_PostpaidTran.Close();
        TempSource = TempSource.Right(3);
        TempSource = TempSource.Left(2);
        TempSource.MakeLower();
        Command = "select * from price_" +
TempSource;
        Command += " where StationID = '";
        Command += StationID;
        Command += "'";
        if(m_price.SearchPrice(Command) == S_OK)
            station_price.Format("%d",m_price.m_Price);
            m_price.Close();
            Command = "update skytrain.postpaid_amount
set amount = `amount` + '";
            Command += station_price;
            Command += "' where Card_ID = '";
            Command += Card_ID;
            Command += "'";
            m_postpaidamount.MySqlCommand(Command);
            m_postpaidamount.Close();
            Command = "update postpaid_transit.id";
            Command += Card_ID;
            Command += " set `destination` = '";
            Command += StationName;
            Command += "' , `out_time` = now(),
`price` = '";
            Command += station_price;
            Command += "' where `order` = '";
            Command += Order;
            Command += "'";
            m_PostpaidTran.MySqlCommand(Command);
            m_postpaidamount.SearchAmount(Card_ID);
            Command.Format("@@%c5%4d#", TempAddress,
m_postpaidamount.m_amount);
            m_postpaidamount.Close();
        }
    }
    m_PostpaidTran.Close();
}
m_SerialComm.SetOutput(ColeVariant(Command));
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**ตารางแสดงราคาค่าใช้บริการรถไฟฟ้า(บาท)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

